U.S. Department of the Interior
Geological Survey

Calculation of transient soundings for a central induction loop system (Program TCILOOP)

by

Walter L. Anderson

Open-File Report 81-1309

1981

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 TCILOOP is designed to compute transient (time-domain) decay sounding curves over layered earth models for a central induction (CI) 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 vertical magnetic field (Hz), which can be measured with a vertical-axis coil at the center of a large circular current-loop of radius a>>0. The TDR-sounding is evaluated rapidly and accurately using Fourier and Hankel transform digital filters developed by Anderson (1975, 1979a). Optionally, a transient field response (TFR) is defined as the transient Hz-field (e.g., as measured using a cryogenic magnetometer), and may be indirectly computed by integrating the TDR-sounding. We assume the measurement system is driven by an "on-off" step current source of arbitrary current. The CI 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 TCILOOP is similar in design to the methods used by Anderson (1979b) 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 CI program (e.g., improved frequency-domain approximations and

late-time asymptote approximations). But for practical fields situations, the earlier techniques used by Anderson (1979b) are usually satisfactory, when considering the currently available time-domain electromagnetic (TDEM) hardware accuracy and dynamic range.

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 induced in a vertical-axis coil at the center of a circular loop of radius a>0 , placed on or above 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,

$$V(t)=(d/dt)v(T) = \frac{4}{\pi} C \int_0^{\infty} B \, Re[Hz(B)/DC] \cos(B^2 T) \, dB, \qquad (1)$$

where

$C = (nA)I/[2\sigma_1(a^2+z^2)^{3/2}]$,

nA= turns (n) times area (A) of the receiving loop (or coil),

I= driving current I in I*exp(iwt), $I \geq 0$ amps,

σ_1 = conductivity of layer 1,

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

a= transmitting loop radius (a>0),

z= transmitting loop elevation ($z \geq 0$),

B= induction number = a/δ,

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

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

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

t= real time (t>0 seconds),

Hz(w)= frequency-domain response function for a CI-loop in a layered halfspace [see eq.(2) below, and also Ryu et al (1970, eq.(19), p.866)],

DC= limit of Hz(0), which becomes,

DC= $a^2 I/[2(a^2+z^2)^{3/2}]$.

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

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

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 order-1 Hankel transform giving Hz(w) is of the form (see Ryu et al, 1970, eq.(19), p.866),

$$Hz(w) = aI\int_0^{\infty} h(x)\, J_1(xa)\, dx + DC, \qquad (2)$$

where $w = 2B^2/(\sigma_1 \mu_0 a^2)$, and

the complex kernel h(x) depends on the layer parameters (conductivities and thicknesses for M-layers, M>0), angular frequency w>0, and the loop elevation $z \geq 0$. (To insure convergence, the DC term has been subtracted from the integrand, and therefore must be added outside the integral.)

Instead of evaluating eq. (2) directly during the lagged convolution in eq. (1.1), we can easily replace the normalized real function Re[Hz/DC] by a suitable cubic spline function with sufficient knots per decade in w (or equivalently B) to adequately define Hz from some initial induction number B0=a/δmax to Bm=a/δmin, where δ is the skin depth in layer 1. In fact, the asymptotic values as w->0 and w->∞ can be easily incorporated into the splined-frequency response by observing the limits,

$$\lim_{w\to 0} \mathrm{Re}[Hz(w)/DC] = DC/DC = 1, \quad \text{and} \quad \lim_{w\to\infty} \mathrm{Re}[Hz(w)/DC] = 0.$$

The above procedure works very fast (since we are convolving a J1-filter with a cubic spline function) and sufficiently accurate, as long as B0 and Bm (and NB=number of B-points per decade) are adequately chosen; in practice, the default values (B0=.01,NB=8,Bm=100) 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<8 (it is not recommended that NB<5 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), 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); for a halfspace model, Hmax=a is used.]

An option to compute the TFR-sounding (parameter ISTEP=1) is provided by a time-integration of the previously computed TDR-sounding; i.e.,

$$VTFR(T)= \int_{T_0}^{T} (d/dt)\ v(T)\ dt = \int_{T_0}^{T} V(t)\ dt, \qquad (3)$$

where the integrand is given by eq. (1.1), and T0>0 is sufficiently small. This definition is stable (since integration errors are minimal) and works well as long as T0 is close to zero-time. Equation (3) is evaluated rapidly for each T by replacing the integrand by a cubic spline function, which should be defined at a suitable T-spacing (see parameters T0,NT, and TM below). The relationship between normalized time T and real time t (sec.) are given by the formulas,

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

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

$$\lim_{\substack{t\to 0 \\ z=0}} [V(t)/C]=3, \quad \text{and} \quad \lim_{\substack{t\to\infty \\ z\geq 0}} [V(t)/C]=0,$$

for any horizontally layered earth model. It turns out that the limit at t=0 represents the transient decay in the first layer of any conductivity, while the analytic limit at t=∞ has completely decayed. For very 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, the field logistics may prohibit very large-sized loops. The equipment signal-to-noise ratio may also constrain the effective loop radius to use (note that a square loop of area R*R can be used, and is equivalent to a circular loop of radius a, i.e., $\pi a^2=R^2$).

After considering these practical field and equipment 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 the initial limit at t=0 (i.e., V(0)/C=3.0=Order(10^0). 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, 1979a) 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 TCILOOP have been compared with a completely different central-induction (CI) program (Raiche, 1981, priv. comm.; Raiche's CI-program is similar to a coincident-loop solution 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 TCILOOP algorithm ran about 10-to-30 times faster than Raiche's CI-algorithm.

PARAMETERS REQUIRED

Parameters required by program TCILOOP are read using a FORTRAN NAMELIST simulator on the VAX (currently, VAX FORTRAN-77 Version 2.3 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 TCILOOP is as follows:

1. Title record (always required, maximum of 80-characters).
2. $PARMS --non-default parameters--$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 solutions disk file (if default IPCH=0, this file is not written).

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

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

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

DETAILED PARAMETER DEFINITIONS

$PARMS parameters (non-default 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.

Z= Loop elevation (in m.) on or above the surface (default Z=0.0). Note that Z>0 specifies the source loop is Z meters above the surface, but that the central induction receiver coil is assumed to be placed on the surface. For most field applications, Z=0 is normally used.

ISTEP= 0 (default) to compute the transient derivative response (TDR) sounding, which corresponds to the time-derivative of Hz when the source uses a system step driving current (e.g., when using a vertical-axis coil at the loop projected center).

ISTEP= 1 to compute the transient field response (TFR) sounding, which corresponds to the integral over time of the TDR-sounding. The TFR-sounding (ISTEP=1) is generally used when transient (stacked) data is obtained using a SQUID or cryogenic magnetometer. Note that Z=0 must be used whenever ISTEP=1.

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

B0=.01 (default) is the lower induction number for which the Hz/DC frequency response approaches 1.0 for B<B0. B0 must be given (or assumed .01 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<.01 can be used.

BM=100 (default) is the upper induction number for which the Hz/DC frequency response approaches 0.0 for B>BM. BM must be given (or assumed 100 by default) as a power of 10**n. The default value is usually adequate for most models; for more accuracy in the early-time transient, BM>100 can be used.

NB=8 (default) represents the number of induction number points per decade (log-cycle) to evaluate the pre-splined frequency response function Hz(B)/DC. In general, $5 \leq NB \leq 11$ is usually adequate for most applications (NB<5 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.

T0= Initial normalized time to compute the transient, where T0>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>T0 must be specified as a power of 10**±n.

NT= Number of normalized time points to compute per time decade (log-cycle) between T0 and TM, where NT>0 must be specified.

XNORM= Normalization factor (default 3.0) 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) and time (in sec.) in the format (2E16.8). 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 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
$PARMS M=2,SIG=.02,2,H=200, A=200,
TO=.1,NT=6,TM=100,NB=6,ISTOP=0$
MODIFIED EXAMPLE
$PARMS NB=11,A=1000,ISTOP=1$END
```

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

VAX OPERATING INSTRUCTIONS

Assuming program TCILOOP (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 and FOR013.DAT will be assumed for FOR011 and FOR013, respectively). When IPCH=0 (default), this step may be ignored.

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

```
$RUN TCILOOP !  on the USGS system, use the command:

$RUN [WANDERSON]TCILOOP
```

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 TCILOOP; also, VAX system-dependent commands and calls have been minimized in TCILOOP 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 TCILOOP, and the job is terminated.

REFERENCES

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

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

----------, 1979a, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, v.44, n.7, p.1287-1305.

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

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

PB-274-119).

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.

Ryu, J., Morrison, H.F., and Ward, S.H., 1970, Electromagnetic fields about a loop source of current: Geophysics, v. 35, n.5, p. 862-896.

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 TCILOOP on a VAX system. The corresponding output file (FOR016) is given following FOR005.

FOR005

```
TEST MODEL
$PARMS M=2,A=200,T0=.1,NT=4,TM=.1E5,
SIG=.001,.1,H=200$
```

FOR016

```
 {TCILOOP}:      TEST MODEL

 M =  2          XNORM=0.30E+01  ISTEP=  0       A= 0.2000E+03   Z= 0.0000E+00
 IOUTS =  16     T0= 0.1000E+00  NT =    4       TM= 0.1000E+05  ISTOP = 1
 IOUT =    6     B0= 0.1000E-01  NB =    8       BM= 0.1000E+03  EPS= 0.10E-09
 IPCH= 0

 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(T0:TM)    TIME(SEC)     TRANS         TRANS(NORM)  NORM*XNORM

   0.10000E+00  0.25133E-05  0.27783E+01  0.10000E+01  0.30000E+01
   0.17783E+00  0.44693E-05  0.19695E+01  0.70887E+00  0.21266E+01
   0.31623E+00  0.79477E-05  0.96931E+00  0.34888E+00  0.10467E+01
   0.56234E+00  0.14133E-04  0.33341E+00  0.12000E+00  0.36001E+00
   0.10000E+01  0.25133E-04  0.77318E-01  0.27829E-01  0.83487E-01
   0.17783E+01  0.44693E-04  0.14846E-01  0.53434E-02  0.16030E-01
   0.31623E+01  0.79477E-04  0.48775E-02  0.17556E-02  0.52667E-02
   0.56234E+01  0.14133E-03  0.26530E-02  0.95490E-03  0.28647E-02
   0.10000E+02  0.25133E-03  0.15801E-02  0.56872E-03  0.17062E-02
   0.17783E+02  0.44693E-03  0.92273E-03  0.33212E-03  0.99636E-03
   0.31623E+02  0.79477E-03  0.50698E-03  0.18248E-03  0.54743E-03
   0.56234E+02  0.14133E-02  0.25649E-03  0.92321E-04  0.27696E-03
   0.10000E+03  0.25133E-02  0.11779E-03  0.42397E-04  0.12719E-03
   0.17783E+03  0.44693E-02  0.48810E-04  0.17568E-04  0.52705E-04
   0.31623E+03  0.79477E-02  0.18303E-04  0.65878E-05  0.19763E-04
   0.56234E+03  0.14133E-01  0.62686E-05  0.22563E-05  0.67688E-05
   0.10000E+04  0.25133E-01  0.19880E-05  0.71555E-06  0.21466E-05
   0.17783E+04  0.44693E-01  0.59315E-06  0.21350E-06  0.64049E-06
   0.31623E+04  0.79477E-01  0.17011E-06  0.61227E-07  0.18368E-06
   0.56234E+04  0.14133E+00  0.49036E-07  0.17650E-07  0.52949E-07
   0.10000E+05  0.25133E+00  0.14360E-07  0.51686E-08  0.15506E-07



$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=          11.42 SEC. (    0 MIN. 11.42 SEC.)
CPU_TIME=          4.94 SEC. (    0 M.  4.94 S.)   CPU % = 43.27%
BUF.I/O_COUNT=       37
DIR.I/O_COUNT=        0
PAGE_FAULTS=         62
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

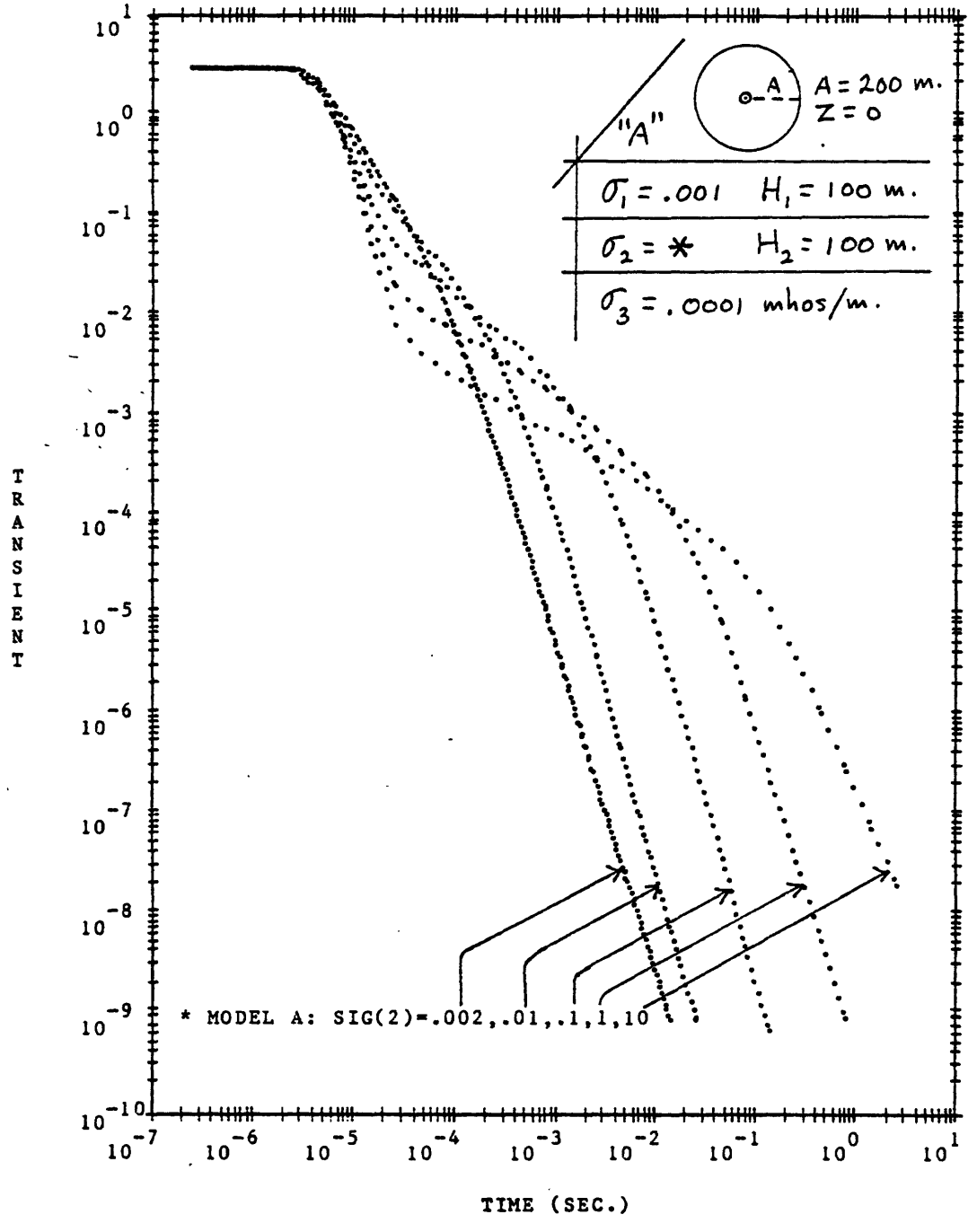
Appendix 3.-- Some sounding curve example plots

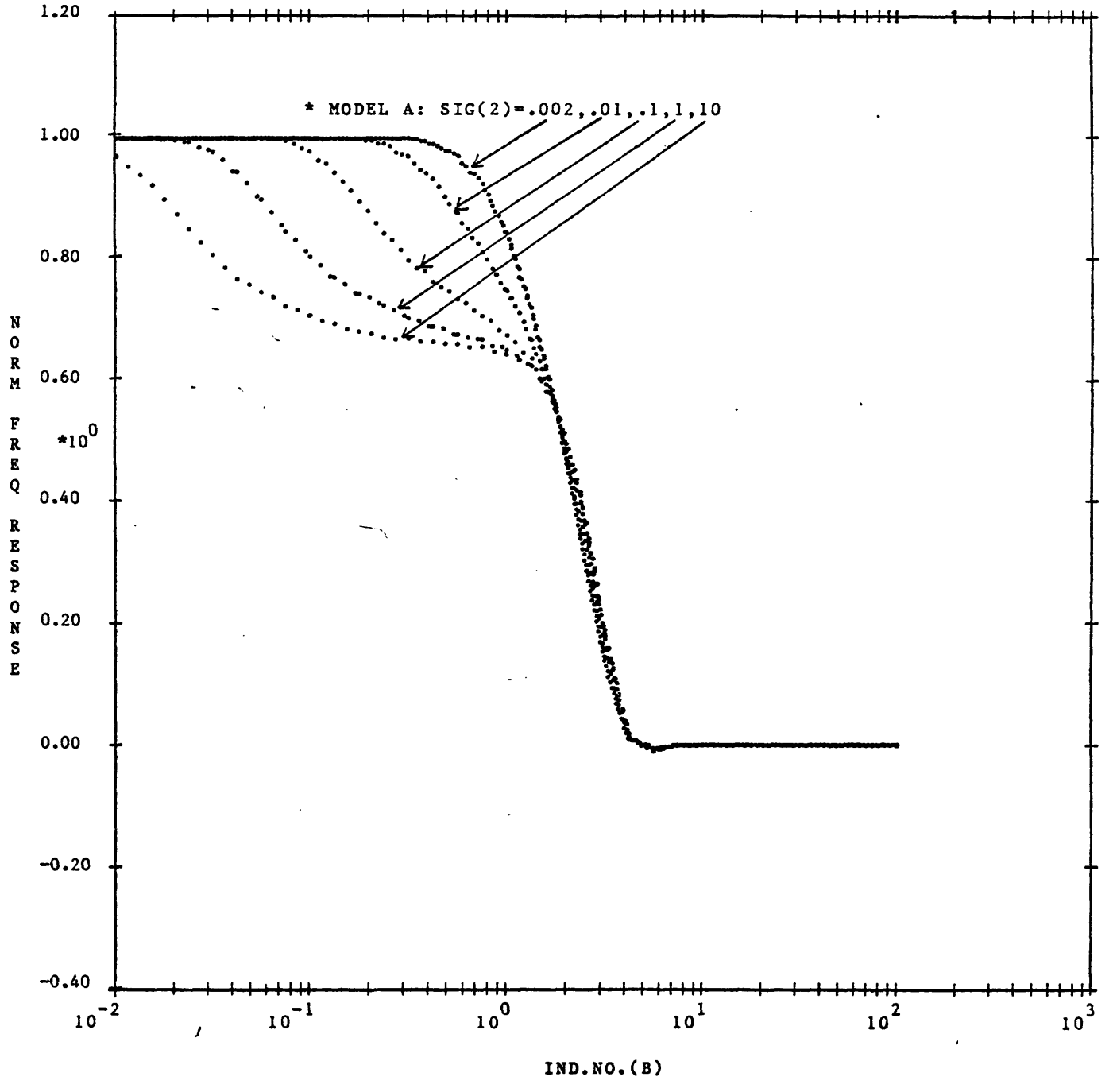
The attached plots were produced (after using IPCH>1) on an AJ-832 terminal for several layered models, and curve families, by varying certain model parameters. The beginning of each model (denoted "A","B",...) is indicated by a "model-figure insert" drawn on the unnormalized TRANSIENT* curve plot, followed on a successive page with the corresponding normalized FREQUENCY response for the given model. The notation used is, hopefully, self-explainatory. (Note that the TFR-soundings in model "S" corresponds to the TDR-soundings in model "A".)

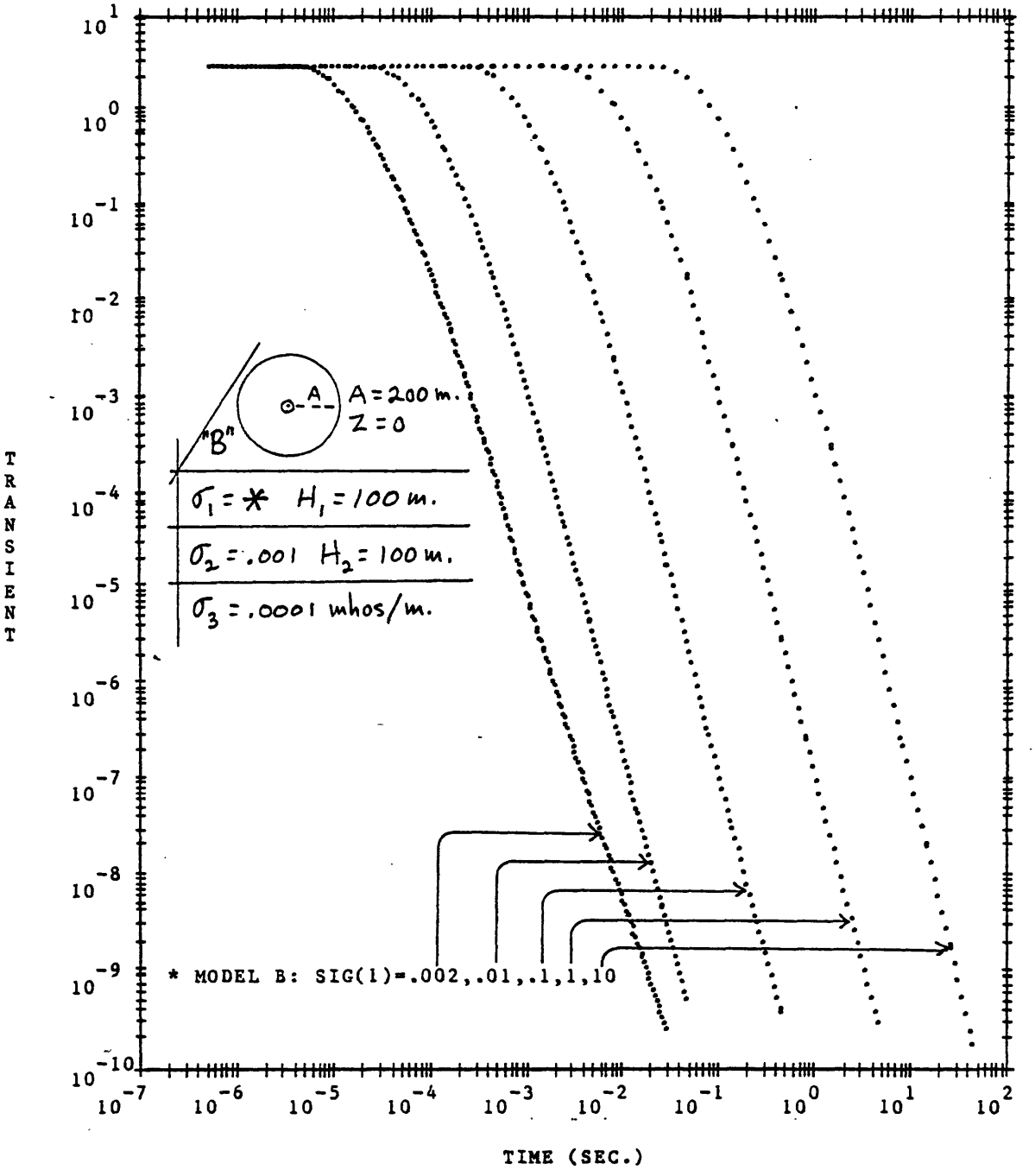
Much of the discussion in the INTRODUCTION and SUMMARY OF CALCULATIONS sections are illustrated in these plots. For example, referring to model "K", we observe that as the radius A decreases, the first deflection in each transient curve becomes progressively lower in magnitude at about 0.1 seconds. In 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.

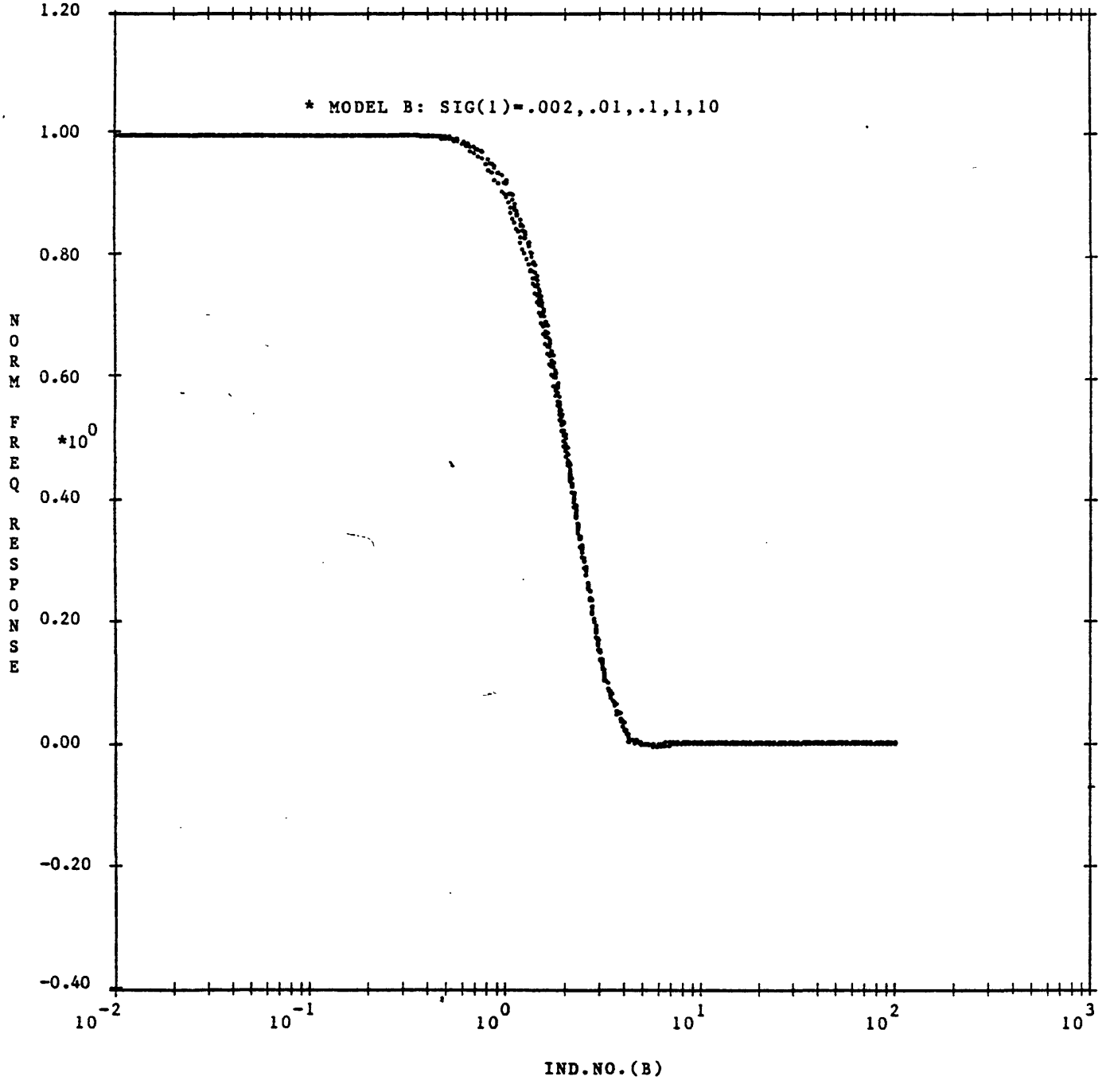
---

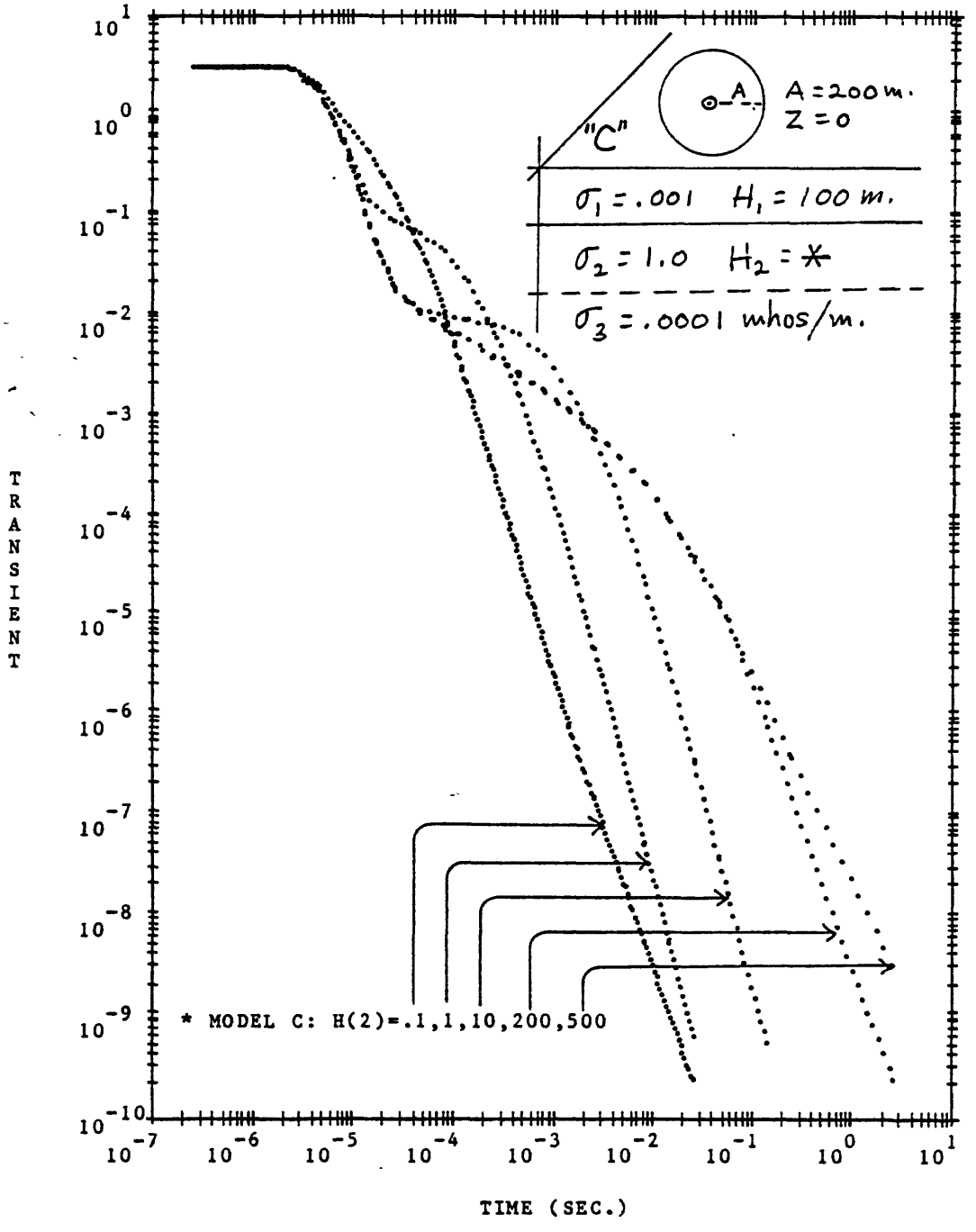
* The term "TRANSIENT" used in these plots refer to a TDR-sounding.

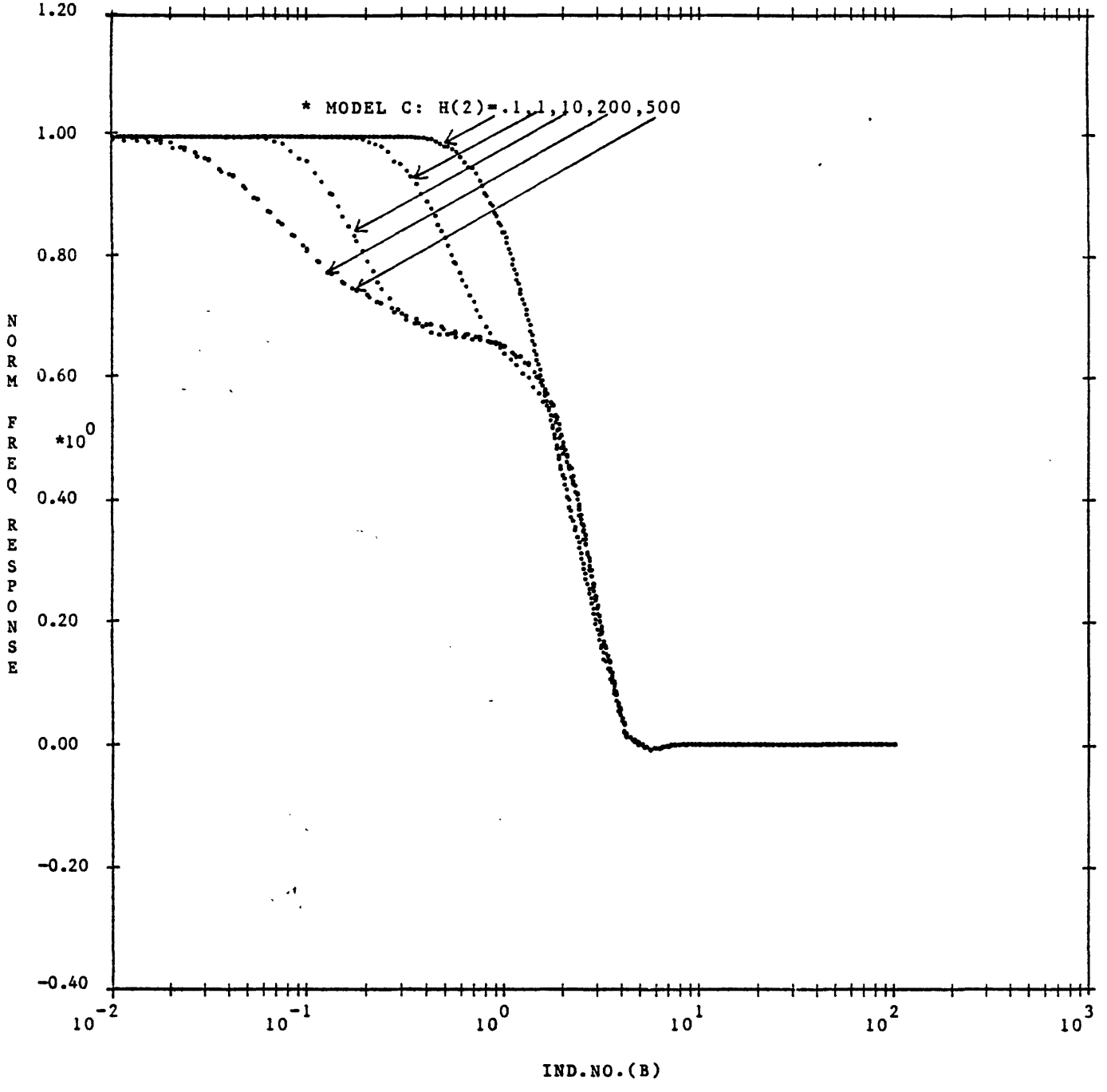

"A"
A
A = 200 m.
Z = 0
σ_1 = .001 H_1 = 100 m.
σ_2 = * H_2 = 100 m.
σ_3 = .0001 mhos/m.
* MODEL A: SIG(2)=.002,.01,.1,1,10
TRANSIENT
TIME (SEC.)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1}

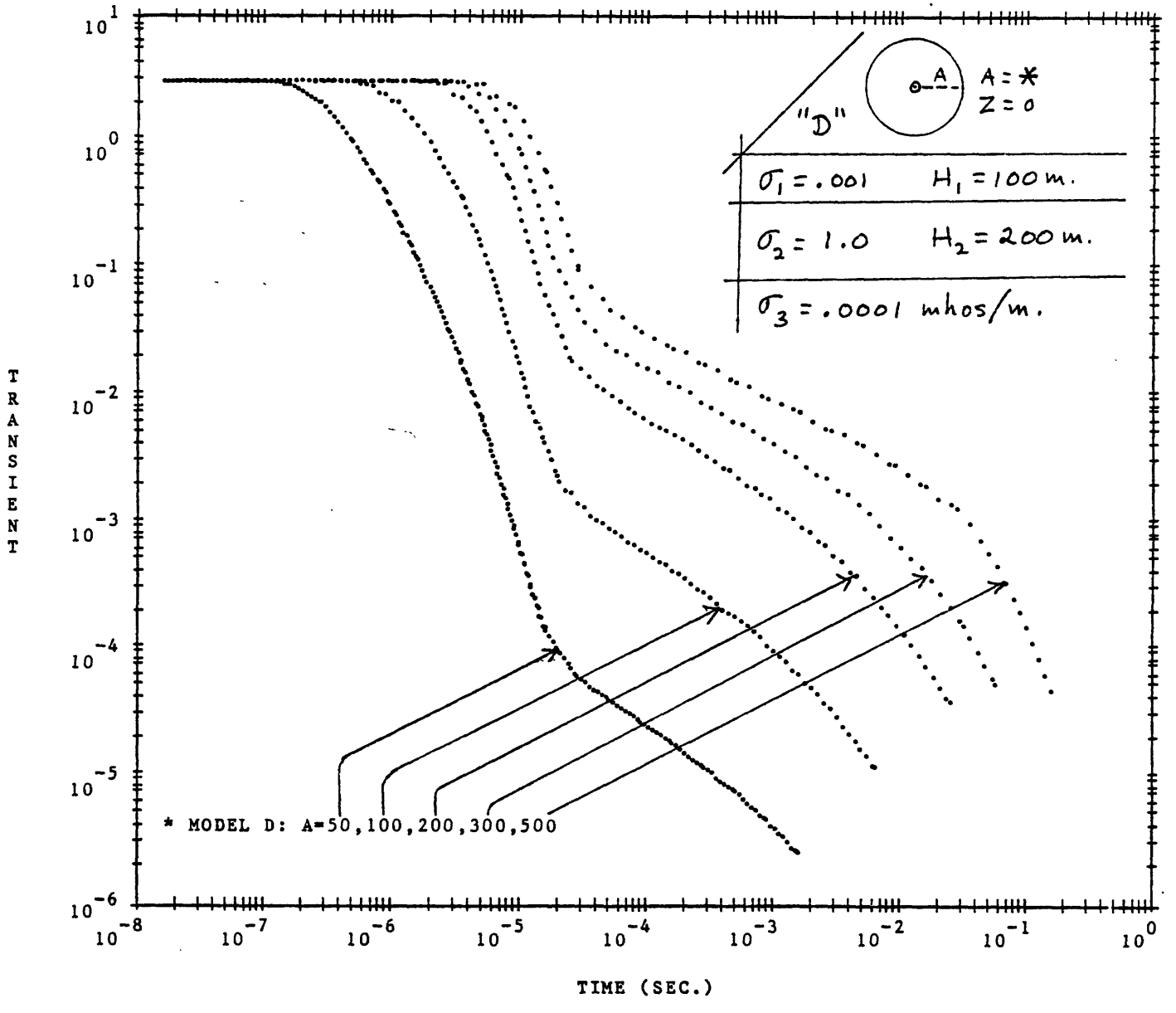

* MODEL A: SIG(2)=.002,.01,.1,1,10
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

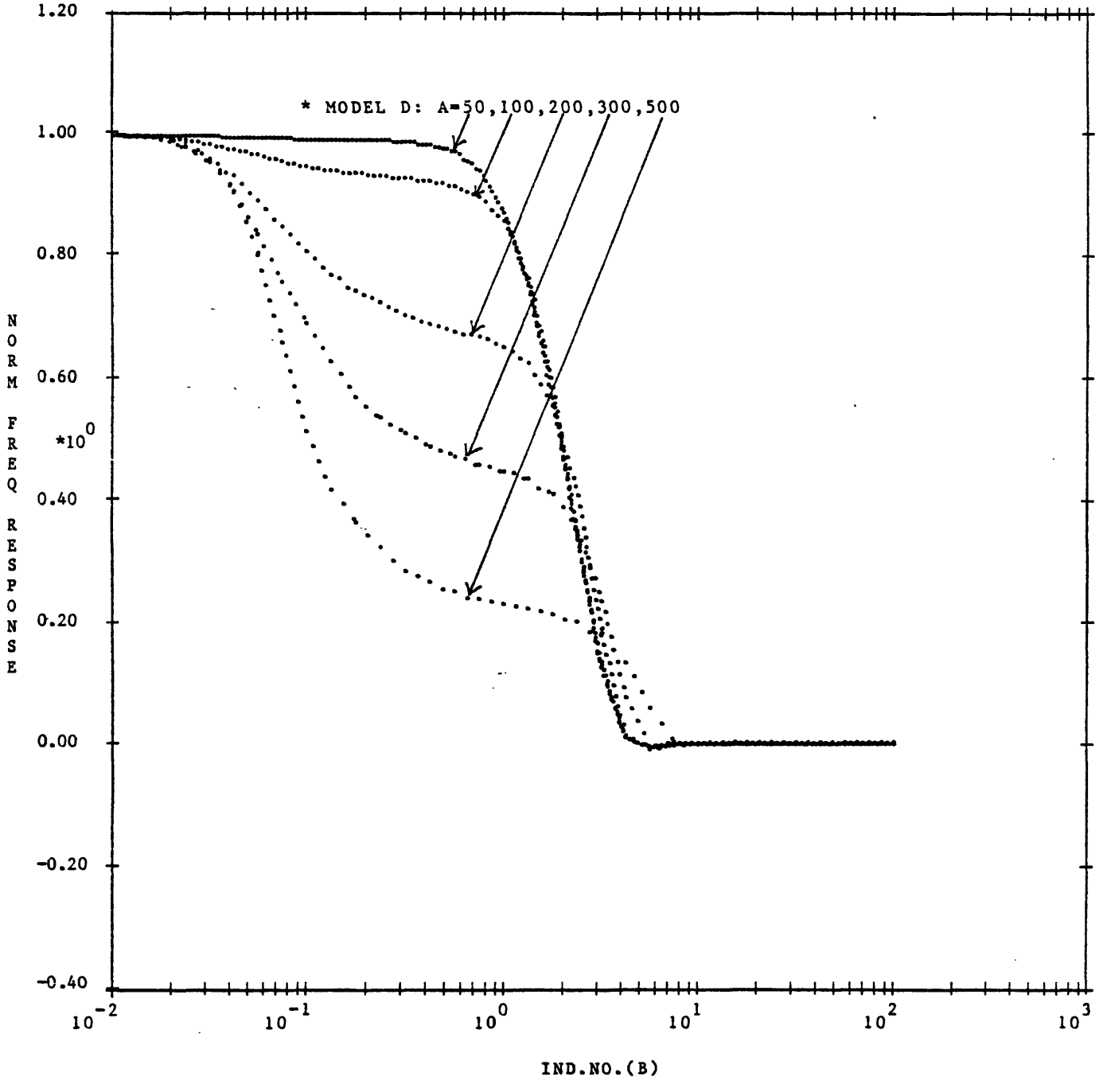

TRANSIENT
TIME (SEC.)
A
A = 200 m.
Z = 0
"B"
σ1 = * H1 = 100 m.
σ2 = .001 H2 = 100 m.
σ3 = .0001 mhos/m.
* MODEL B: SIG(1)=.002,.01,.1,1,10

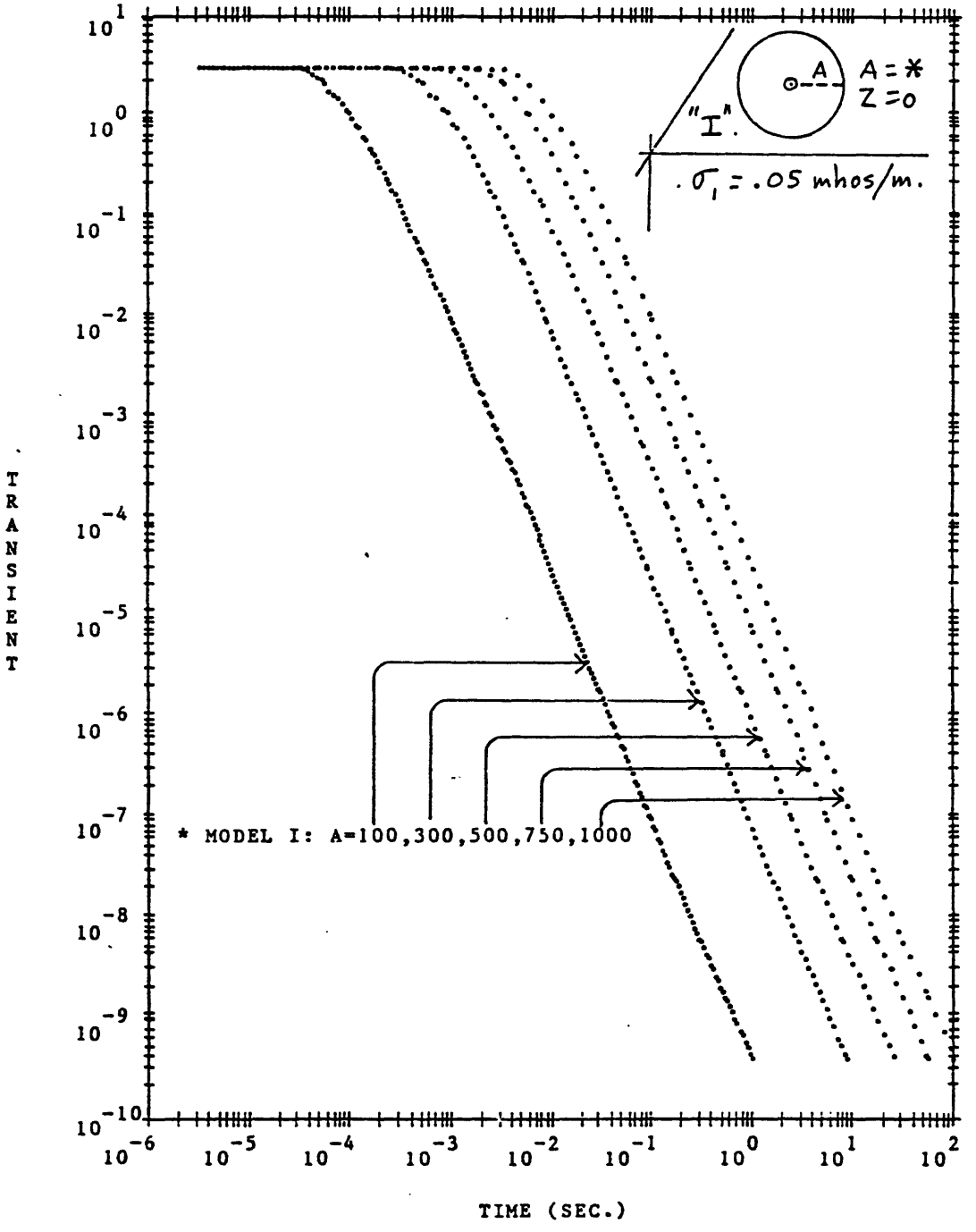

* MODEL B: SIG(1)=.002,.01,.1,1,10
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

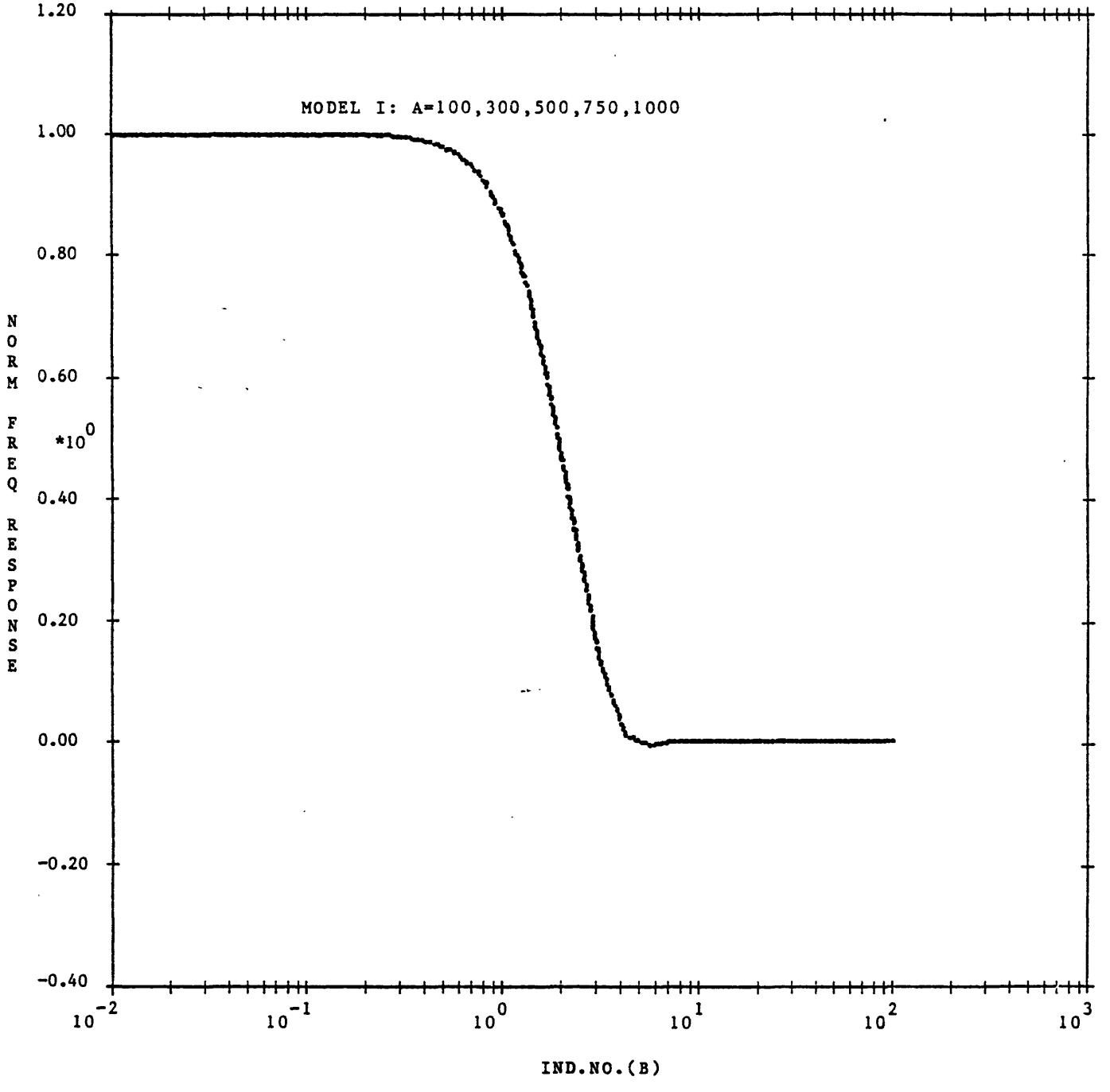

"C"
A
A = 200 m.
Z = 0
σ1 = .001 H1 = 100 m.
σ2 = 1.0 H2 = ∞
σ3 = .0001 mhos/m.
* MODEL C: H(2)=.1,1,10,200,500
TRANSIENT
TIME (SEC.)

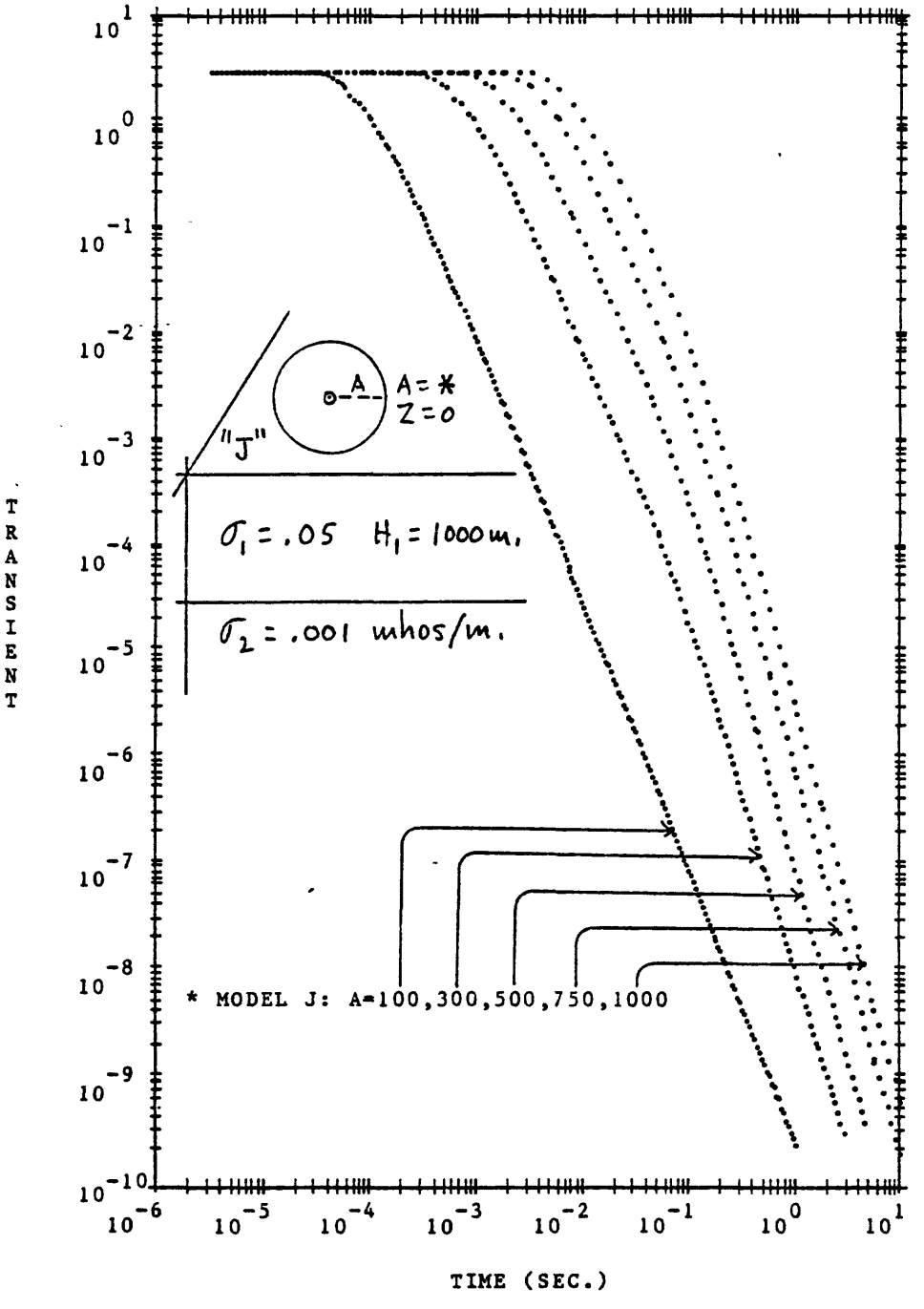

* MODEL C: H(2)=.1,1,10,200,500
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

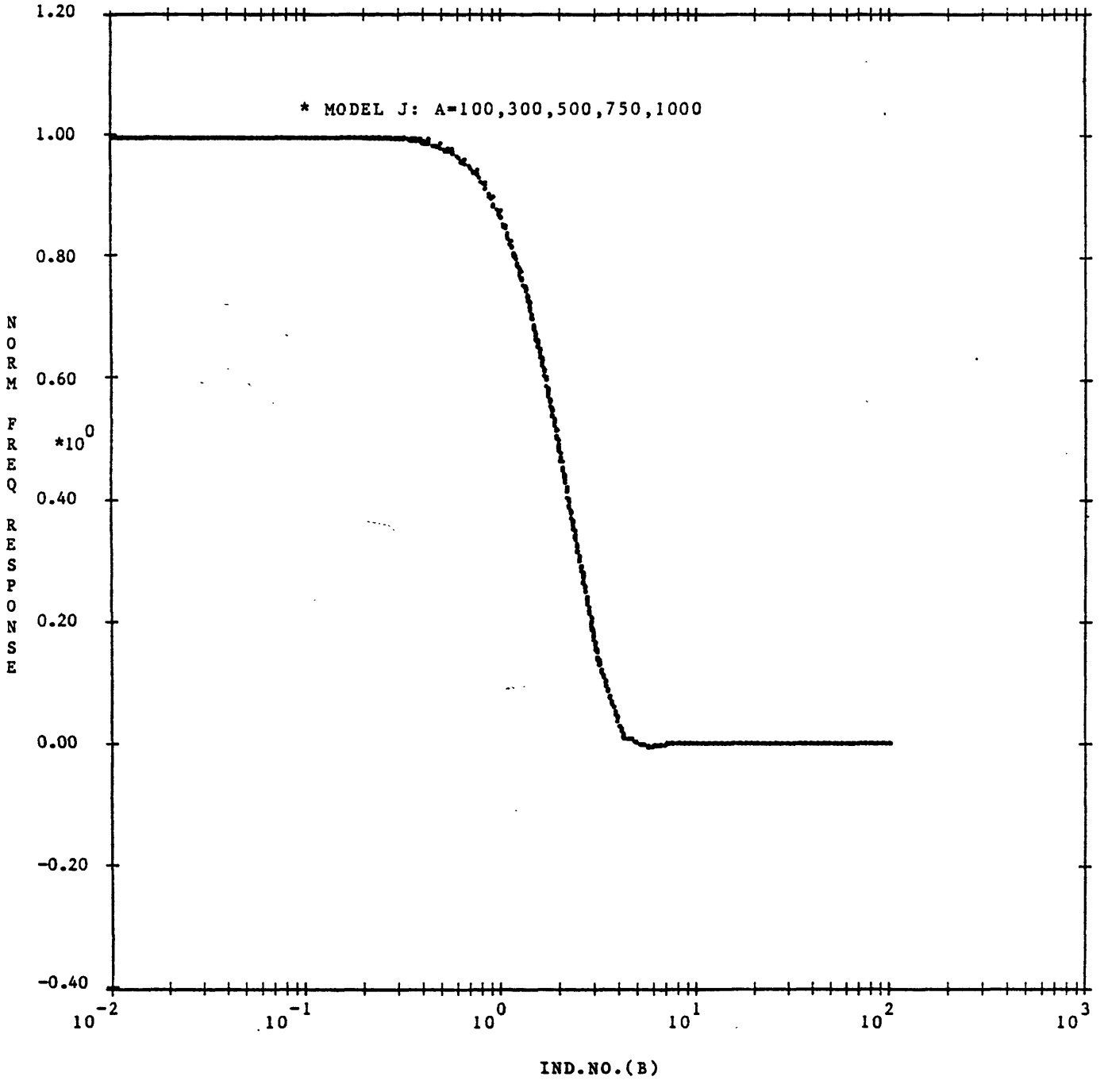

"D"
A
A = *
Z = 0
σ1 = .001 H1 = 100 m.
σ2 = 1.0 H2 = 200 m.
σ3 = .0001 mhos/m.
* MODEL D: A=50,100,200,300,500
TRANSIENT
TIME (SEC.)

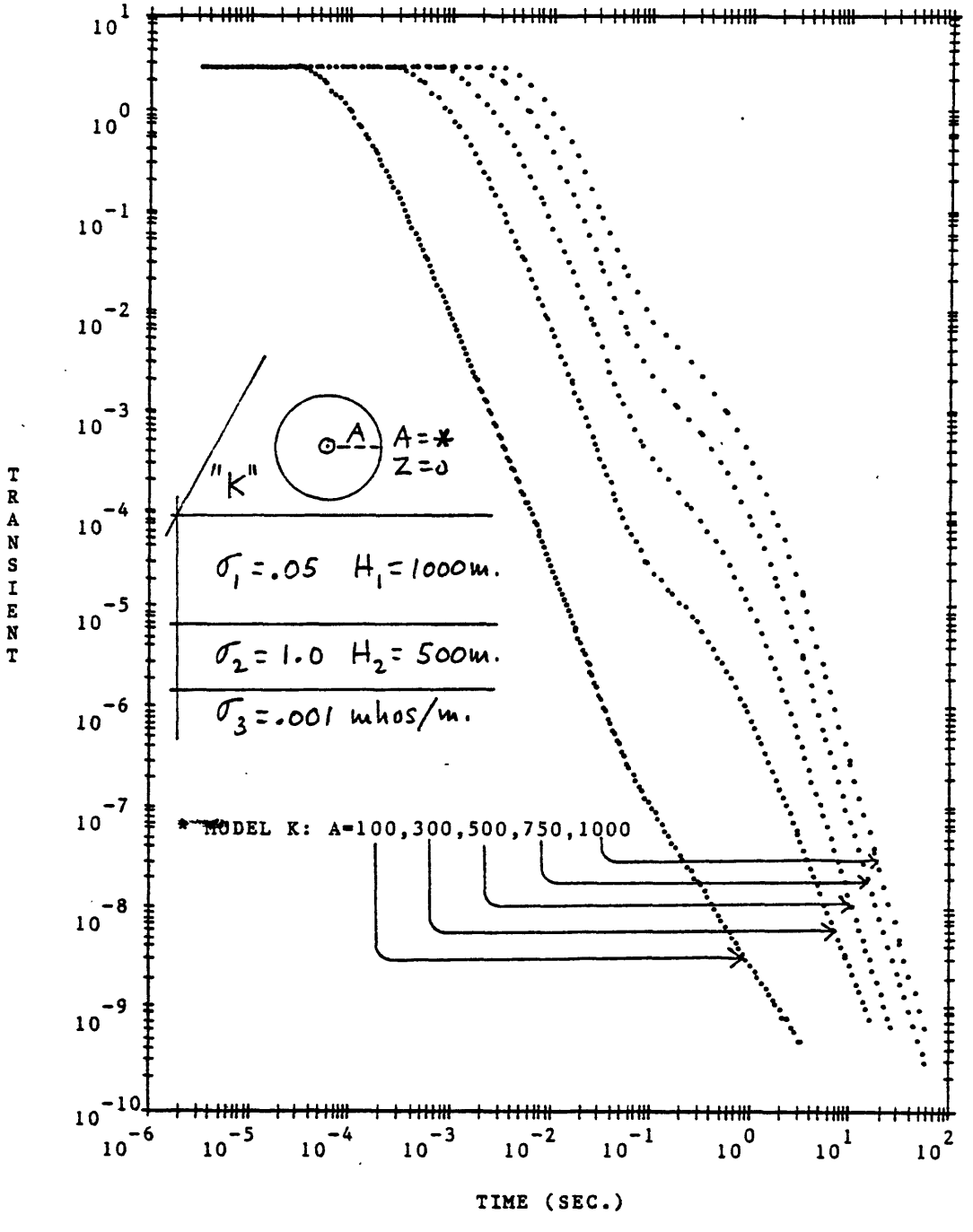

* MODEL D: A=50,100,200,300,500
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

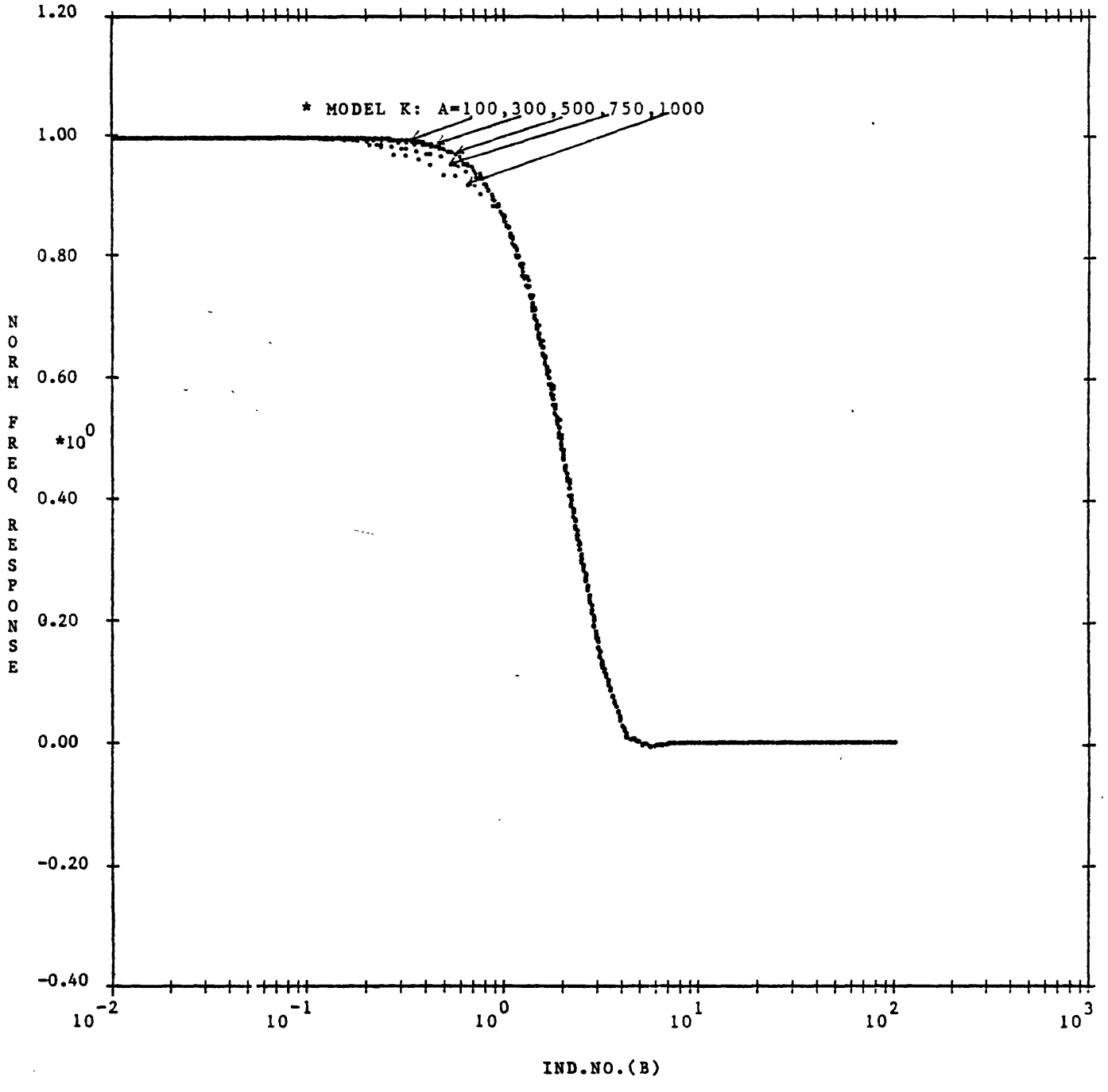

TRANSIENT
TIME (SEC.)
A
A = *
Z = 0
"I"
σ_1 = .05 mhos/m.
* MODEL I: A=100,300,500,750,1000

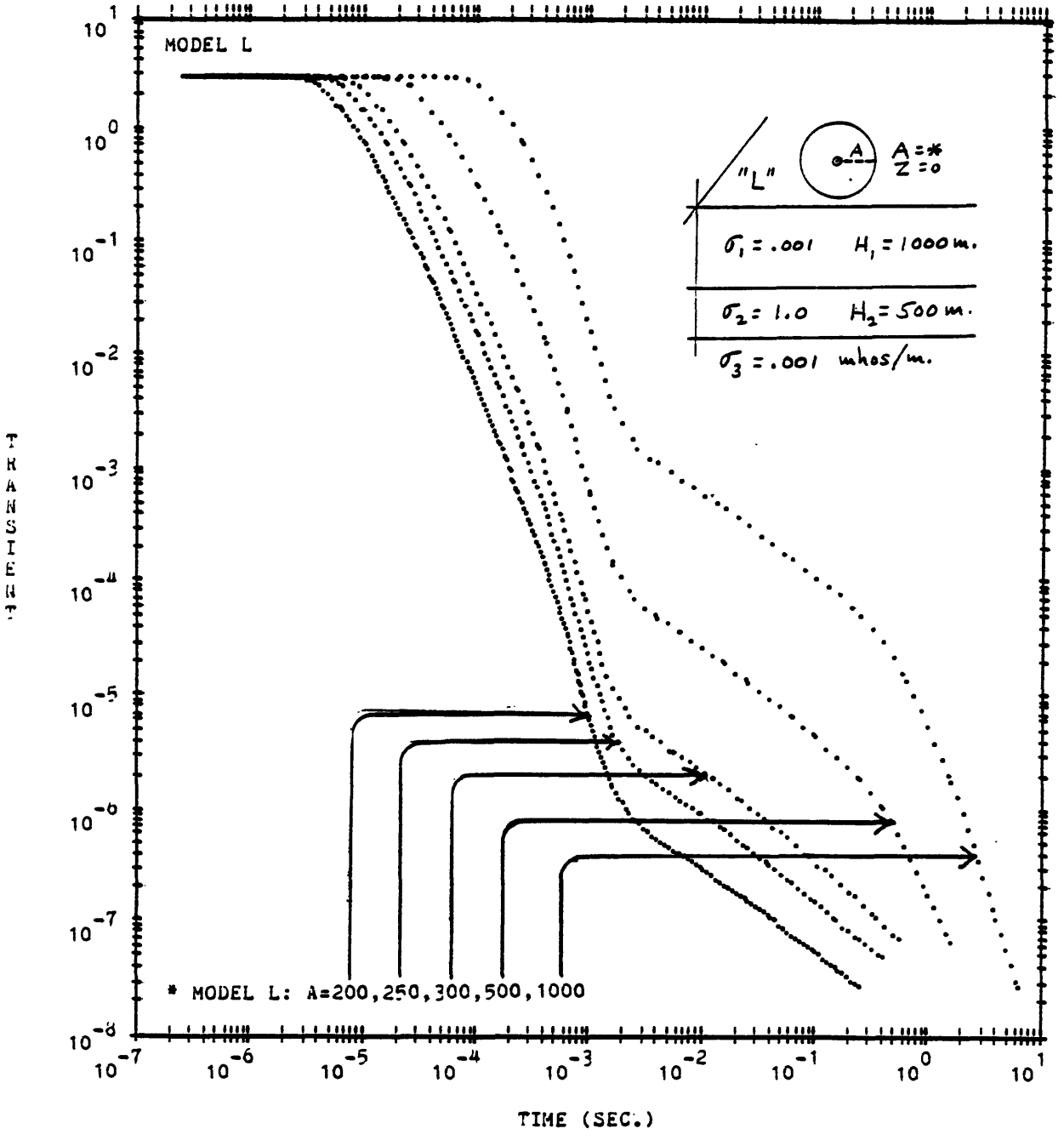

MODEL I: A=100,300,500,750,1000
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

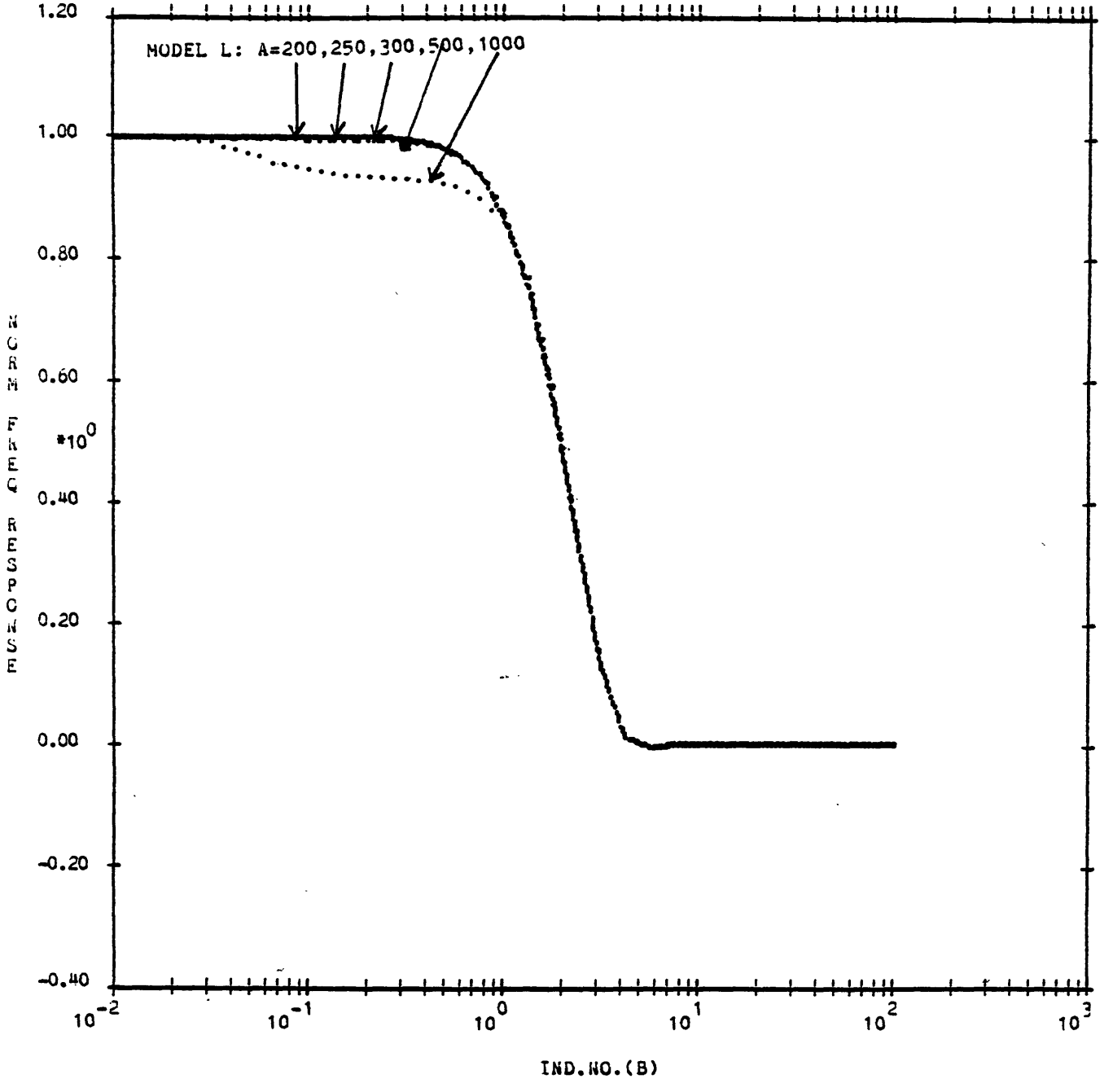

TRANSIENT
TIME (SEC.)
"J"
A
A = *
Z = 0
σ1 = .05 H1 = 1000m.
σ2 = .001 mhos/m.
* MODEL J: A=100,300,500,750,1000

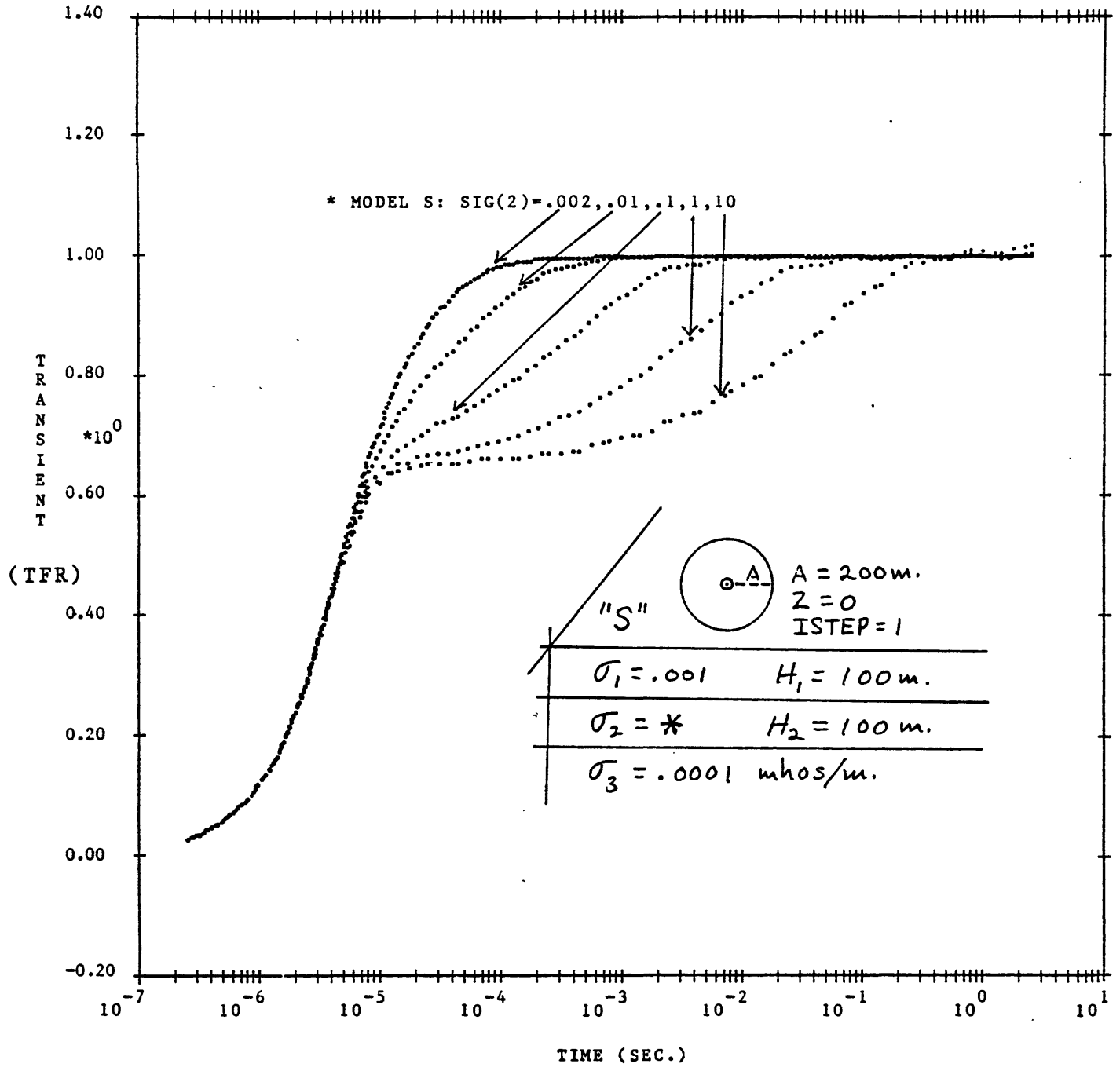

* MODEL J: A=100,300,500,750,1000
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

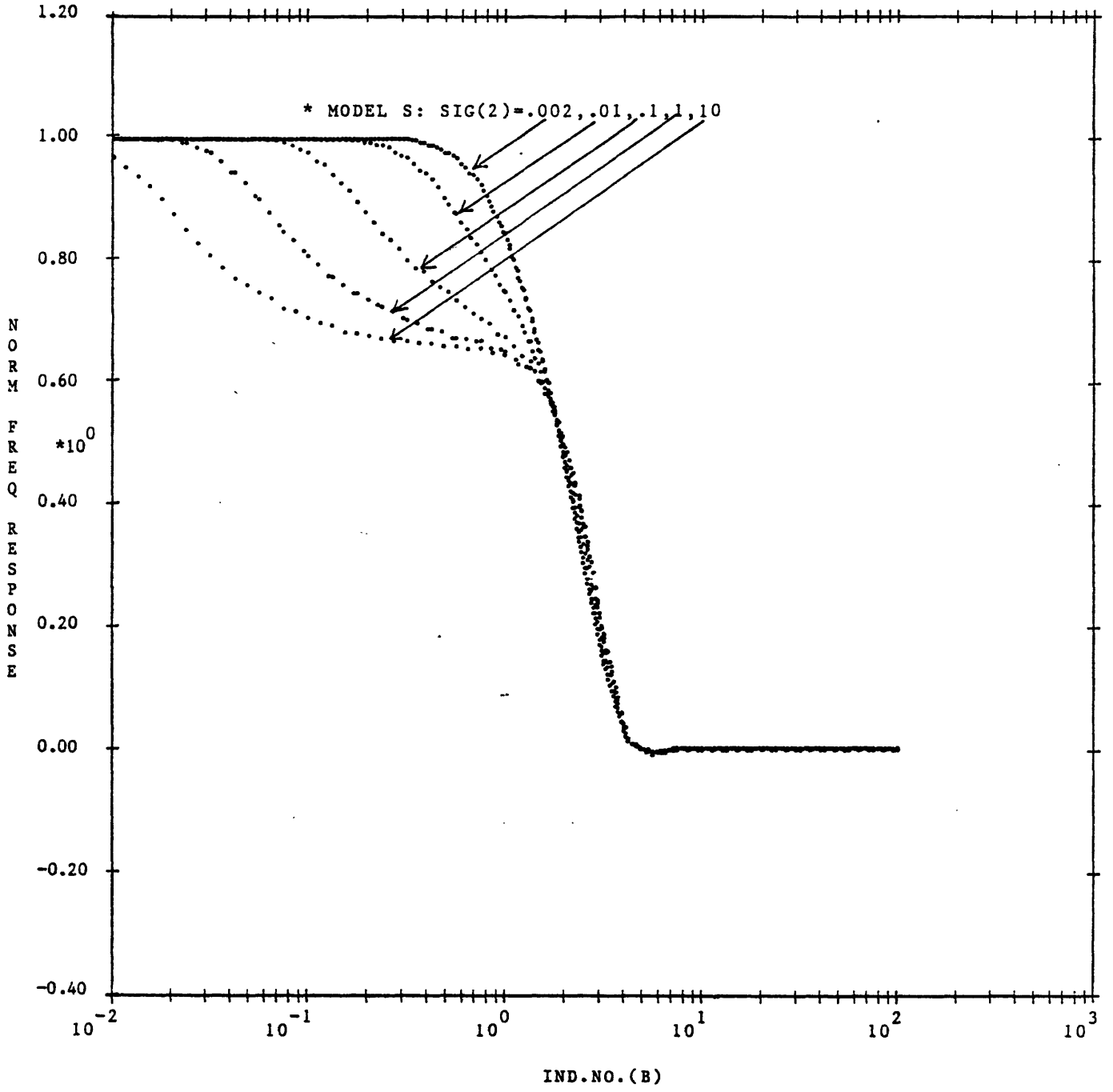

TRANSIENT
TIME (SEC.)
"K"
A
A=*
Z=0
σ1=.05 H1=1000m.
σ2=1.0 H2=500m.
σ3=.001 mhos/m.
* MODEL K: A=100,300,500,750,1000

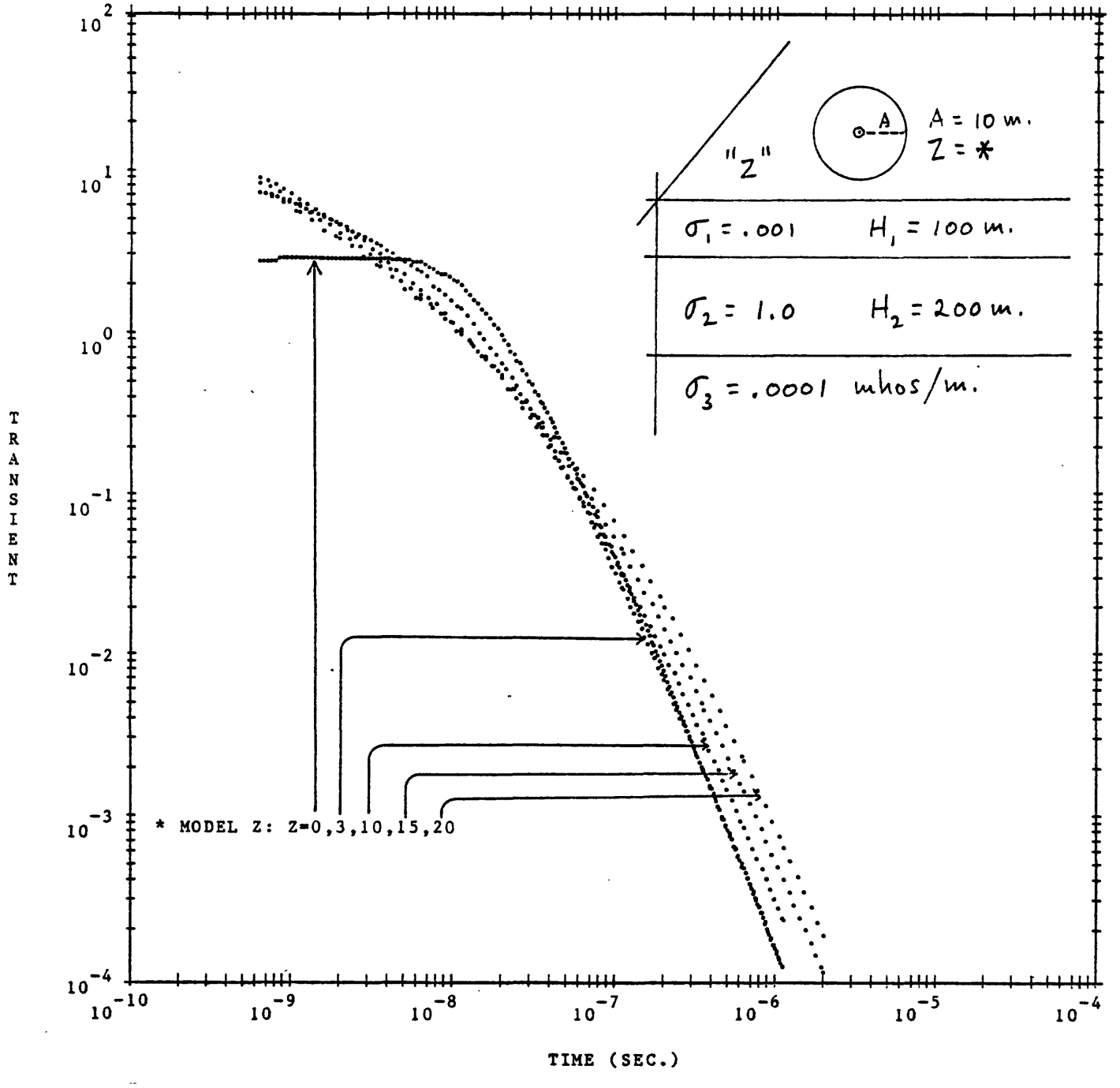

* MODEL K: A=100,300,500,750,1000
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)

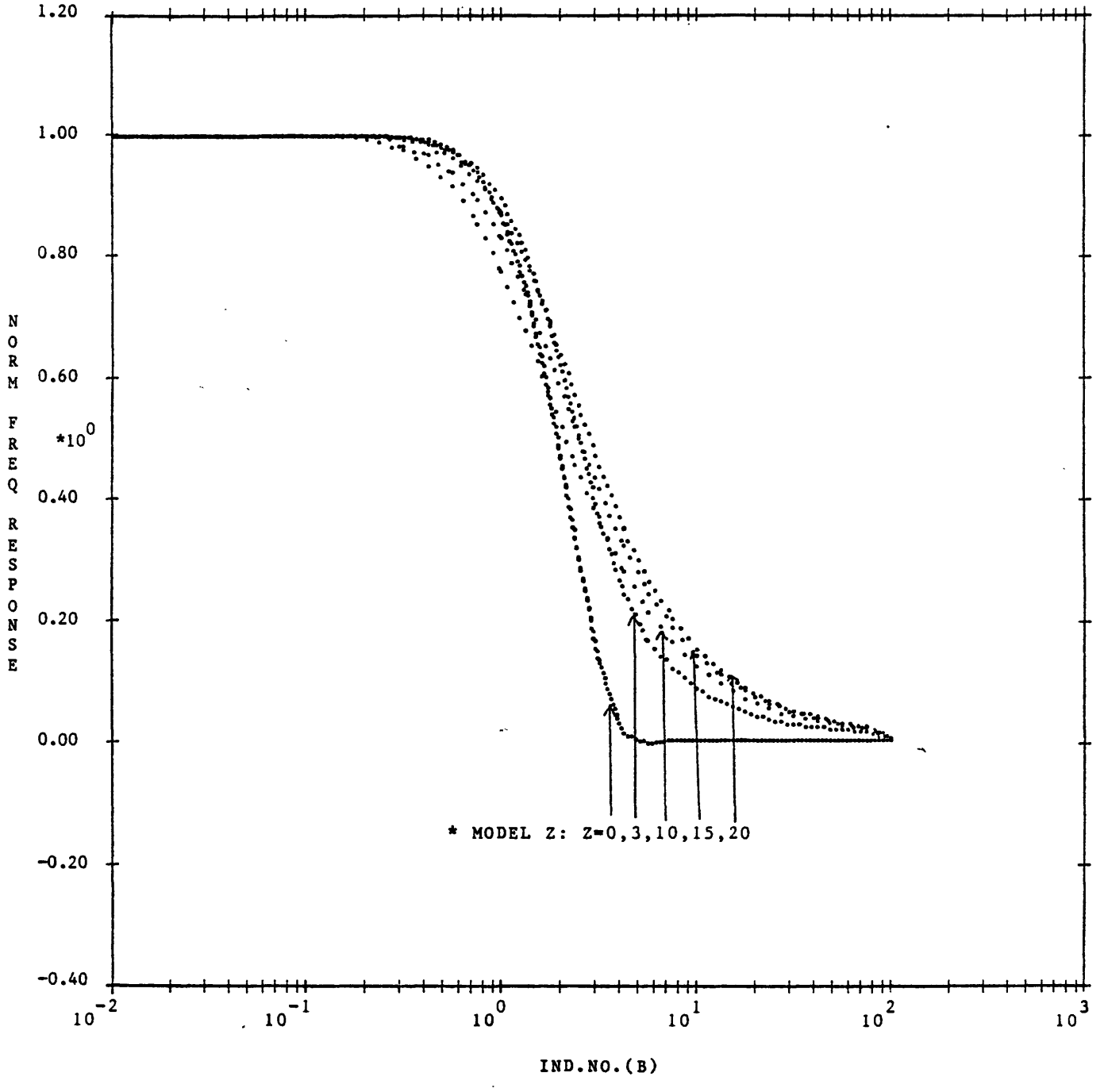

MODEL L
"L"
A
A=*
Z=0
σ1 = .001 H1 = 1000 m.
σ2 = 1.0 H2 = 500 m.
σ3 = .001 mhos/m.
* MODEL L: A=200,250,300,500,1000
TRANSIENT
TIME (SEC.)
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1


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


* MODEL S: SIG(2)=.002,.01,.1,1,10
TRANSIENT
(TFR)
*10^0
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
TIME (SEC.)
"S"
A
A = 200m.
Z = 0
ISTEP = 1
σ1 = .001 H1 = 100 m.
σ2 = * H2 = 100 m.
σ3 = .0001 mhos/m.


* MODEL S: SIG(2)=.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
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)


"Z"
A
A = 10 m.
Z = *
σ1 = .001
H1 = 100 m.
σ2 = 1.0
H2 = 200 m.
σ3 = .0001 mhos/m.
* MODEL Z: Z=0,3,10,15,20
TRANSIENT
TIME (SEC.)
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-10
10^-9
10^-8
10^-7
10^-6
10^-5
10^-4


1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-2
10^-1
10^0
10^1
10^2
10^3
IND.NO.(B)
* MODEL Z: Z=0,3,10,15,20

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 "TCILOOP.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 {TCILOOP}: TRANSIENT SOUNDING FOR CENTRAL INDUCTION LOOP  {10/21/81}  00000010
C  FORWARD SOLUTIONS, WHERE CIRCULAR LOOP HAS RADIUS A>0.0 AND ELEVATION00000020
C  Z>0 (FOR CURRENT LOOP IN AIR) OR Z=0 (FOR CURRENT LOOP ON GROUND).   00000030
C  THE TRANSIENT FIELD (CENTRAL INDUCTION) IS ASSUMED MEASURED AT THE   00000040
C  LOOP CENTER, BUT AT THE SURFACE OF THE EARTH.                        00000050
C                                                                       00000060
C  BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00000070
C                                                                       00000080
C--REFS:  RYU, ET.AL., 1970, GEOPHYSICS, V.35, N.5, P.862-896.          00000090
C  ANDERSON,W.L., 1975, NTIS REPORT PB-242-800.                         00000100
C  ANDERSON,W.L., 1979, GEOPHYSICS, V.44, NO. 7, P.1287-1305.           00000110
C  ANDERSON,W.L., 1979, USGS OPEN-FILE REPT. 79-590.                    00000120
C                                                                       00000130
C  NOTE THAT NORMALIZED TIME (TAU) IS USED FROM T0 TO TM, WHERE         00000140
C  TIME=0.5*TAU*SIG1*(FOURPI*E-7)*A**2  (TIME IN SEC.)                  00000150
C  I.E., TAU=(2.0*TIME)/(SIG1*FOURPI*E-7*A**2).                         00000160
C  IPCH=1 OPTION (DEFAULT 0) WILL WRITE FILE10 WITH                     00000170
C  (TRANS,TIME) IN FORMAT (2E16.8).                                     00000180
C  IPCH>1 WILL WRITE FILE10 (AS ABOVE), AND ALSO WILL WRITE             00000190
C  FILE11 AND FILE13 FOR POSSIBLE PLOTTING PURPOSES (LATER IF DESIRED). 00000200
C                                                                       00000210
C--CALLS RFLAGS, HZLOOP (& INTEG1 IF ISTEP=1) TO COMPUTE THE TRANSIENT  00000220
C  USING LAGGED-CONVOLUTION IN TIME (DEPENDING ON NB OPTION--SEE DOC.)  00000230
C  AND DIRECT OR SPLINED FREQ FUNCTION IN (B0,BM)--MIN,MAX IND.NUMBER.  00000240
C  NOTE: FREQ.FUNCT/DC=1.0 IS ASSUMED IF B<B0 AND =0.0 IF B>BM, WHERE   00000250
C  DEFAULT B0=.01, BM=100 IS USUALLY ADEQUATE FOR MOST MODELS.          00000260
C                                                                       00000270
      CHARACTER*80 TITLE                                               00000280
      REAL SIG(10),H(10),DER(2), T(200),V(200)                         00000290
      COMPLEX K2(10),KS1,C4,ZA,ZAC4                                    00000300
```

```
      EXTERNAL HZLOOP                                                   00000310
      COMMON/PASS/ZAC4,ANORM,CURI,DC,SIG,B0,BM,SIG1,EPS,ISTEP           00000320
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00000330
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                                00000340
C**                                                                     00000350
C** SEE CALL NAMELIST SIMULATOR FOR THE VAX                            00000360
C**                                                                     00000370
C**       NAMELIST/PARMS/M,SIG,H,A,Z,EPS,ISTEP,                         00000380
C**      1 B0,BM,NB,T0,TM,NT,XNORM,IOUT,IOUTS,IPCH,ISTOP                00000390
      COMMON/NAME_LIST/M_,SIG_(10),H_(9),A_,Z_,EPS_,ISTEP_,             00000400
     1 B0_,BM_,NB,T0,TM,NT,XNORM,IOUT,IOUTS,IPCH,ISTOP                  00000410
      DATA DER/2*0.0/                                                   00000420
C--PRESET                                                               00000430
      IPCH=0                                                            00000440
      B0_=.01                                                           00000450
      NB=8                                                              00000460
      BM_=100.                                                          00000470
      ISTEP_=0                                                          00000480
      M_=1                                                              00000490
      XNORM=3.0                                                         00000500
      DO 10 I=1,9                                                       00000510
      SIG(I)=0.0                                                        00000520
10    H_(I)=0.0                                                         00000530
      SIG(10)=0.0                                                       00000540
      A_=0.0                                                            00000550
      Z_=0.0                                                            00000560
      EPS_=.1E-9                                                        00000570
      ISTOP=1                                                           00000580
      IOUTS=16                                                          00000590
      T0=0.0                                                            00000600
      NT=0                                                              00000610
      TM=0.0                                                            00000620
      IN=5                                                              00000630
      IOUT=6                                                            00000640
20    READ(IN,30,END=999) TITLE                                         00000650
30    FORMAT(A)                                                         00000660
      CALL SETTIME                                                      00000670
C**      READ(IN,PARMS,END=999)                                         00000680
      CALL NAMELIST(IN,'$PARMS',*999)                                   00000690
      M=M_                                                              00000700
      DO 35 I=1,9                                                       00000710
      SIG(I)=SIG_(I)                                                    00000720
35    H(I)=H_(I)                                                        00000730
      SIG(10)=SIG_(10)                                                  00000740
      EPS=EPS_                                                          00000750
      B0=B0_                                                            00000760
      BM=BM_                                                            00000770
      ISTEP=ISTEP_                                                      00000780
      A=A_                                                              00000790
      Z=Z_                                                              00000800
      CALL NONBLANK(TITLE,NONBLK)                                       00000810
      IF(IOUT.GT.0)                                                     00000820
     2  WRITE(6,40)                                                     00000830
     3 TITLE,M,XNORM,ISTEP,A,Z,                                         00000840
     1 IOUTS,T0,NT,TM,ISTOP,IOUT,B0,NB,BM,EPS,IPCH,SIG,H                00000850
40    FORMAT('1{TCILOOP}:',6X,A<NONBLK>//                               00000860
     2 5H M = ,I2,10X,6HXNORM=,E8.2,2X,6HISTEP=,I3,7X,                  00000870
```

```
     3 2HA=,E11.4,3X,2HZ=,E11.4/                                                00000880
     4 9H IOUTS = ,I3,5X,3HTO=,E11.4,2X,5HNT = ,I4,7X,3HTM=,E11.4,2X,           00000890
     5 8HISTOP = ,I1/7H IOUT =,I5,                                              00000900
     6 5X,3HBO=,E11.4,2X,5HNB = ,I4,7X,3HBM=,E11.4,2X,4HEPS=,                   00000910
     7 E9.2/6H IPCH=,I2//                                                       00000920
     8 6H SIG =,5E12.4/6X,5E12.4//                                              00000930
     9 6H   H =,5E12.4/6X,5E12.4)                                               00000940
      IF(IOUTS.GT.0) WRITE(IOUTS,40)                                            00000950
     & TITLE,M,XNORM,ISTEP,A,Z,                                                 00000960
     1 IOUTS,TO,NT,TM,ISTOP,IOUT,BO,NB,BM,EPS,IPCH,SIG,H                        00000970
      IF(NT.LE.0) CALL ERRMSG('NT<=0',1,IOUT,IOUTS)                             00000980
      IF(M.LT.1.OR.M.GT.10) CALL ERRMSG('M<1 OR M>10',4,IOUT,IOUTS)             00000990
      IF(A.LE.0.0) CALL ERRMSG('A<=0',2,IOUT,IOUTS)                             00001000
      IF(ISTEP.LT.0.OR.ISTEP.GT.1)                                              00001010
     & CALL ERRMSG('ISTEP<0 OR >1',3,IOUT,IOUTS)                                00001020
      IF(BO.LE.0.0.OR.BM.LE.BO)                                                 00001030
     & CALL ERRMSG('BO<=0 OR BM<=BO',3,IOUT,IOUTS)                              00001040
      IF(TO.LE.0.0.OR.TM.LE.TO)                                                 00001050
     & CALL ERRMSG('TO<=0 OR TM<=TO',3,IOUT,IOUTS)                              00001060
      IF(Z.LT.0.0)CALL ERRMSG('Z<0',2,IOUT,IOUTS)                               00001070
      IF(Z.GT.0.0.AND.ISTEP.EQ.1)                                               00001080
     1 CALL ERRMSG('Z>0 AND ISTEP=1',1,IOUT,IOUTS)                              00001090
      IF(SIG(1).LE.0.0)CALL ERRMSG('SIG(1)<=0',3,IOUT,IOUTS)                    00001100
C--PRESET SOME CONSTANTS                                                        00001110
      R=0.0                                                                     00001120
      AA=A*A                                                                    00001130
      SIG1=SIG(1)                                                               00001140
      TCON=6.28318531E-7*SIG1*AA                                                00001150
      ZA=CMPLX(A,0.0)                                                           00001160
      IF(M.EQ.1) THEN                                                           00001170
        HMAX=A                                                                  00001180
      ELSE                                                                      00001190
        CALL MINMAX(H,M-1,TEM,HMAX)                                             00001200
      ENDIF                                                                     00001210
      ANORM=A/HMAX                                                              00001220
      CURI=.3183098861/AA                                                       00001230
      C4=CMPLX(A/(2.0*SQRT(AA+Z*Z)**3),0.0)                                     00001240
      ZAC4=ZA*C4                                                                00001250
      DC=A*CURI*REAL(C4)                                                        00001260
      ISPLN=0                                                                   00001270
      IF(NB.GT.0.AND.NB.LT.12) ISPLN=1                                          00001280
      IF(ISPLN.EQ.0) GO TO 49                                                   00001290
C--GET PRE-SPLINED FREQ. FUNCTION (0<NB<12 OPTION)                              00001300
      DB=EXP(2.30258509/FLOAT(NB))                                              00001310
      BMTEST=0.5*(BM+BM*DB)                                                     00001320
      NS=0                                                                      00001330
      TEM=BO/DB                                                                 00001340
      ISPLN=0                                                                   00001350
46    TEM=TEM*DB                                                                00001360
      IF(TEM.GE.BMTEST) GO TO 47                                                00001370
      NS=NS+1                                                                   00001380
      IF(NS.GT.200)CALL ERRMSG('SPLINED NS>200',1,IOUT,IOUTS)                   00001390
      XS(NS)=TEM                                                                00001400
      YS(NS)=HZLOOP(TEM*TEM)                                                    00001410
      GO TO 46                                                                  00001420
47      CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,V)                            00001430
C  WRITE FILE11 IF IPCH>1 (FOR LATER PLOTTING--IF DESIRED)                      00001440
```

```
      IF(IPCH.GT.1) WRITE(11,1000) TITLE(1:40),NS,(XS(I),YS(I),I=1,NS)  00001450
1000  FORMAT('3'/'IND.NO.(B)'/'NORM FREQ RESPONSE'/A/I/(2G16.8))        00001460
      ISPLN=1                                                           00001470
49    NEW=1                                                             00001480
      DT=EXP(2.30258509/FLOAT(NT))                                      00001490
      TMTEST=0.5*(TM+TM*DT)                                             00001500
      IT=0                                                              00001510
      TEM=T0/DT                                                         00001520
      IF(IOUT.GT.0) WRITE(IOUT,50)                                      00001530
50    FORMAT('0',4X,'TAU(T0:TM)',3X,'TIME(SEC)',4X,'TRANS',8X,          00001540
     &'TRANS(NORM)',2X,'NORM*XNORM'/)                                   00001550
      IF(IOUTS.GT.0) WRITE(IOUTS,50)                                    00001560
      LATE=0                                                            00001570
60    TEM=TEM*DT                                                        00001580
      IF(TEM.GE.TMTEST) GO TO 80                                        00001590
      TIME=TCON*TEM                                                     00001600
      IF(LATE.EQ.1) THEN                                                00001610
        CALL APROX1(TEM,TRANS)                                          00001620
      ELSE                                                              00001630
C--GET TRANSIENT IMPULSE RESPONSE VIA LAGGED CONVOLUTION IN TIME.       00001640
          TRANS=.63661977*RFLAGS(0,HZLOOP,EPS,0.5*T0,TMTEST,TEM,NEW)    00001650
          NEW=0                                                         00001660
        IF(TRANS.LT.1.E-7) THEN                                         00001670
        IF(IT.LT.3)                                                     00001680
     1     CALL ERRMSG('IT<3--CANNOT CALL APROX0',1,IOUT,IOUTS)         00001690
        CALL APROX0(IT,T,V)                                             00001700
        CALL APROX1(TEM,TRANS)                                          00001710
        LATE=1                                                          00001720
        ENDIF                                                           00001730
      ENDIF                                                             00001740
      IT=IT+1                                                           00001750
      IF(IT.GT.200)CALL ERRMSG('IT>200--NT,TM TOO BIG',1,IOUT,IOUTS)    00001760
      T(IT)=TEM                                                         00001770
      V(IT)=TRANS                                                       00001780
      IF(ISTEP.EQ.1) GO TO 60                                           00001790
      IF(IT.EQ.1) TRANS1=TRANS                                          00001800
      TNORM=TRANS/TRANS1                                                00001810
      TXNORM=TNORM*XNORM                                                00001820
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM          00001830
70    FORMAT(1X,5E13.5)                                                 00001840
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM        00001850
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                            00001860
100   FORMAT(2E16.8)                                                    00001870
      GO TO 60                                                          00001880
80    IF(ISTEP.EQ.0) GO TO 82                                           00001890
C--GET STEP RESPONSE AS INTEGRAL OVER TIME OF IMPULSE RESPONSE.         00001900
      CALL INTEG1(IT,T,V,3.0)                                           00001910
      TRANS1=V(IT)                                                      00001920
      DO 81 I=1,IT                                                      00001930
      TEM=T(I)                                                          00001940
      TRANS=V(I)                                                        00001950
      TIME=TCON*TEM                                                     00001960
      TNORM=TRANS/TRANS1                                                00001970
      TXNORM=TNORM*XNORM                                                00001980
      IF(IOUT.GT.0)WRITE(IOUT,70)TEM,TIME,TRANS,TNORM,TXNORM            00001990
      IF(IOUTS.GT.0)WRITE(IOUTS,70)TEM,TIME,TRANS,TNORM,TXNORM          00002000
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                            00002010
```

```
81     CONTINUE                                                          00002020
82     IF(IOUTS.GT.0) WRITE(IOUTS,90)                                    00002030
90     FORMAT(129X)                                                      00002040
       CALL CPUTIME(IOUT,IOUTS)                                          00002050
C  WRITE FILE13 IF IPCH>1 (FOR LATER PLOTTING--IF DESIRED)               00002060
       IF(IPCH.GT.1) THEN                                                00002070
         WRITE(13,2000) TITLE(1:40)                                      00002080
2000   FORMAT('3'/'TIME (SEC.)'/'TRANSIENT'/A)                           00002090
       DO I=1,IT                                                         00002100
         II=I                                                            00002110
         IF(V(I).LT.1.E-7) GO TO 2001                                    00002120
       ENDDO                                                             00002130
2001   WRITE(13,2002) II,(TCON*T(J),V(J),J=1,II)                         00002140
2002   FORMAT(I/(2G16.8))                                                00002150
       ENDIF                                                             00002160
       IF(ISTOP.NE.1) GO TO 20                                           00002170
999    CALL EXIT                                                         00002180
       END                                                               00002190
       SUBROUTINE APROX0(IT,T,V)                                         00002200
C--LATE TIME APPROXIMATION INITIALIZATION WHEN 1ST COMPUTED TRANS<1E-7    00002210
C  AND 2<IT<201 (REQUIRED).                                              00002220
C                                                                        00002230
       SAVE                                                              00002240
       DIMENSION A(201),B(201),C(201),D(2),T(1),V(1),W1(201),W2(201),    00002250
      1 TLOG(201),VLOG(201)                                              00002260
       DATA D/2*0.0/                                                     00002270
       DO 10 I=1,IT                                                      00002280
         TLOG(I)=ALOG(T(I))                                              00002290
10     VLOG(I)=ALOG(V(I))                                                00002300
       NT=IT+1                                                           00002310
       TLOG(NT)=87.498234                                                00002320
       VLOG(NT)=-87.498234                                               00002330
       CALL SPLIN1(NT,0.0,TLOG,VLOG,A,B,C,0,D,W1,W2)                     00002340
       RETURN                                                            00002350
C** ENTRY APROX1(TEM,TRANS)                                              00002360
       ENTRY APROX1(TEM,TRANS)                                           00002370
       AT=ALOG(TEM)                                                      00002380
       IF(AT.GT.87.498234) THEN                                          00002390
         TRANS=0.0                                                       00002400
         RETURN                                                          00002410
       ENDIF                                                             00002420
       CALL SPOINT(NT,TLOG,VLOG,A,B,C,AT,TT)                             00002430
       TRANS=EXP(TT)                                                     00002440
       RETURN                                                            00002450
       END                                                               00002460
       REAL FUNCTION HZLOOP(B2)                                          00002470
C--COSINE-TRANSFORM KERNEL FOR CENTRAL INDUCTION LOOP WITH               00002480
C  A>0,R=0, AND Z>=0.0.                                                  00002490
C                                                                        00002500
       REAL SIG(10),H(10),Z                                              00002510
       COMPLEX ZHANKS,ZAC4,K2(10),KS1,ZFLD                               00002520
       COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                                00002530
       COMMON/PASS/ZAC4,ANORM,CURI,DC,SIG,B0,BM,SIG1,EPS,ISTEP           00002540
       COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00002550
       EXTERNAL F3ZH                                                     00002560
       B=SQRT(B2)                                                        00002570
       IF(B.LT.B0) GO TO 3                                               00002580
```

```
      IF(B.GT.BM) GO TO 4                                              00002590
      IF(ISPLN.EQ.0) GO TO 10                                          00002600
C--ISPLN=1 (0<NB<12 OPTION) INTERPOLATE PRE-SPLINED FREQ. FUNCTION     00002610
      CALL SPOINT(NS,XS,YS,AS,BS,CS,B,HZLOOP)                          00002620
      RETURN                                                           00002630
10    F=(B/A)**2/(39.47841762E-7*SIG1)                                 00002640
      KS1=CMPLX(0.0,-7.895683523E-6*F)                                 00002650
      DO 1 I=1,M                                                       00002660
1     K2(I)=KS1*CMPLX(SIG(I),0.0)                                      00002670
      ZFLD=ANORM*ZHANKS(1,ANORM,F3ZH,EPS,LL,1) + ZAC4                  00002680
      ZFLD=CMPLX(CURI,0.0)*ZFLD                                        00002690
      HZLOOP=REAL(ZFLD)/DC                                             00002700
      RETURN                                                           00002710
3     HZLOOP=1.0                                                       00002720
      RETURN                                                           00002730
4     HZLOOP=0.0                                                       00002740
      RETURN                                                           00002750
      END                                                              00002760
      COMPLEX FUNCTION F3ZH(X)                                         00002770
C--KERNEL FOR HANKEL TRANSFORM IN CURLOOP WHEN R=0.0 AND Z>=0.0        00002780
C  SCALED BY HMAX STORED IN COMMON/MODEL/                              00002790
C                                                                      00002800
      COMPLEX Z1,Z0,K2(10),KS1,HALF                                    00002810
      REAL H(10),Z                                                     00002820
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                               00002830
      DATA HALF/(0.5,0.0)/                                             00002840
      Y=X/HMAX                                                         00002850
      CALL RECUR(Y,Z1,Z0)                                              00002860
      F3ZH=CMPLX(Y,0.0)*(Z1/(Z0+Z1)-HALF)                              00002870
      IF(Z.GT.0.0) F3ZH=F3ZH*CMPLX(EXP(-Y*Z),0.0)                      00002880
      RETURN                                                           00002890
      END                                                              00002900
      SUBROUTINE RECUR(Y,Z1,Z0)                                        00002910
C--BACKWARD RECURRENCE FOR COMPLEX IMPEDANCES Z1,Z0 GIVEN ARGUMENT     00002920
C     Y(=X/HMAX) AND MODEL PARAMETERS IN COMMON/MODEL/                 00002930
C                                                                      00002940
      REAL H(10),Z                                                     00002950
      COMPLEX Z1,Z0,K2(10),KS1,ONE,ZZ,X2,U                             00002960
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M                               00002970
      DATA ONE/(1.0,0.0)/                                              00002980
      X2=CMPLX(Y*Y,0.0)                                                00002990
      Z0=KS1/CMPLX(Y,0.0)                                              00003000
      Z1=KS1/CSQRT(X2-K2(M))                                           00003010
      IF(M.EQ.1) GO TO 20                                              00003020
      J=M-1                                                            00003030
10    U=CSQRT(X2-K2(J))                                                00003040
      ZZ=KS1/U                                                         00003050
      U=CEXP(CMPLX(-2.0*H(J),0.0)*U)                                   00003060
      U=(ONE-U)/(ONE+U)                                                00003070
      Z1=ZZ*((Z1+ZZ*U)/(ZZ+Z1*U))                                      00003080
      IF(J.EQ.1) GO TO 20                                              00003090
      J=J-1                                                            00003100
      GO TO 10                                                         00003110
20    RETURN                                                           00003120
      END                                                              00003130
      SUBROUTINE NAMELIST(IUNIT,NAME,*)                                00003140
C                                                                      00003150
```

```
C {NAMELIST INPUT ON VAX-11/780} VIA "CALL NAMELIST" {VERSION: 12/10/80}00003160
C                                                                      00003170
C--A SIMULATED 'NAMELIST/NAME/' PROCESSOR FOR VAX-11 FORTRAN-77 TO     00003180
C  IMPLEMENT "CALL NAMELIST(IUNIT,'$NAME',*EOF)" ON VAX, WHICH         00003190
C  IS SIMILAR TO "READ(IUNIT,NAME,END=EOF)" ON MOST LARGE SYSTEMS.     00003200
C                                                                      00003210
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.          00003220
C                                                                      00003230
C--THIS IS A SUBSET OF THE ACTUAL NAMELIST/NAME/ AVAILABLE ON          00003240
C  MOST LARGE MAIN-FRAME SYSTEMS. CURRENT OPTIONS ARE:                 00003250
C                                                                      00003260
C  (1) ALL VARNAM'S ARE RESTRICTED TO 1 TO 6 CHAR'S (ALP,NUM, AND '_') 00003270
C      BUT MUST BEGIN WITH AN ALP CHAR (E.G., A3_, BVAR, C_2, ETC.)    00003280
C  (2) ONLY VARIABLE TYPES REAL*4 *8 (NAMTYP=1) AND INTEGER*2 *4       00003290
C      (NAMTYP=0). SEE C==== EXAMPLE STATEMENTS FOR NAMTYP BELOW =====. 00003300
C      {NOTE: COMPLEX,LOGICAL, OR CHARACTER VARIABLE TYPES ARE "NOT"   00003310
C      CODED IN THIS VERSION.}                                         00003320
C  (3) MAX. 60 VARNAM'S ALLOWED IN NAMELIST (FOR ALL '$NAMES' USED).   00003330
C  (4) MAX. NUMBER FIELD (FLOAT OR FIXED) IS 20 CHAR WIDE, WHERE       00003340
C      BLANK CHAR'S ARE IGNORED, AND TYPE CONVERSION IS AUTOMATIC.     00003350
C      FLOAT NUMBERS WITH OPTIONAL E+XX OR D-XX AND WITH OR WITHOUT '.' 00003360
C      IN THE MANTISSA IS ALLOWED (E.G., 123E-3, .123D+02, -3.14, ETC.).00003370
C  (5) PARTIAL ARRAY'S ALLOWED; E.G., A(10)=25.1,                      00003380
C      AND B=1,3.2,...                                                 00003390
C  (6) REPEAT FACTORS ALLOWED; E.G., C=2*1,3,..                        00003400
C  (7) ONLY 1-DIM ARRAYS ALLOWED WITH MAX SIZE 99999.                  00003410
C  (8) THE NAMELIST '$NAME' MUST BE 2 TO 7 CHAR'S, AND MUST BEGIN WITH 00003420
C      A "$" CHAR (E.G., '$P', '$PARMS', ETC.);  ALSO, THE FIRST CHAR IN00003430
C      IFILE MAY BEGIN IN COL. 1 BUT LESS THAN COL. 72 (BUFFER IS 80).  00003440
C      LINES IN IFILE MAY BE CONTINUED TO COL. 1 ON NEXT LINE, AND      00003450
C      TERMINATE THE NAMELIST BY "$[END]"--THE "END" IS OPTIONAL. E.G., 00003460
C                                                                      00003470
C      $PARMS A=1,B=2.3,7*1,C(3)=-.123E-10,                            00003480
C      D=1800, E=5*20$END                                              00003490
C      $NEXNAM F=123, G=-10,C(2)=15.02 $                               00003500
C      ...END-OF-IFILE...                                              00003510
C  (9) ABOUT 98% OF ALL THE POSSIBLE ERRORS ARE DETECTED AND AN        00003520
C      ERROR MESSAGE IS PRINTED ON UNIT 06, FOLLOWED BY CALL EXIT.     00003530
C      {NOTE: WATCH OUT FOR THE REMAINING 2% UNDETECTED ERRORS!}       00003540
C                                                                      00003550
C--SUBROUTINES CALLED:                                                 00003560
C                                                                      00003570
C  DECODEIX, DECODEX, AND NONBLANK.                                    00003580
C                                                                      00003590
C--USAGE:                                                              00003600
C                                                                      00003610
C  1. MODIFY FILE 'INCLNAMES.FOR' AS REQUIRED (USE ANY EDITOR).        00003620
C     (SEE C==== EXAMPLE STATEMENTS BELOW =====.)                      00003630
C  2. RECOMPILE SUBROUTINE 'NAMELIST' WITH THE DESIRED INCLNAMES.FOR.  00003640
C  3. IN USERS CALLING PROGRAM, USE:                                   00003650
C     CALL NAMELIST(IUNIT,'$NAME',*N)  --ON VAX, WHERE N=E.O.F RETURN  00003660
C     STATEMENT LABEL.  THIS SIMULATES ON VAX:                         00003670
C      'READ(IUNIT,NAME,END=N)' ON SYSTEMS WITH NAMELIST/NAME/...      00003680
C                                                                      00003690
C**********************************************************************00003700
C                                                                      00003710
      CHARACTER*(*) NAME                                               00003720
```

```
      CHARACTER*1 C(47),BUFI                                            00003730
      CHARACTER*6 VARNAM                                                00003740
      CHARACTER*20 NUMFLD                                               00003750
      CHARACTER*80 BUF                                                  00003760
C                                                                       00003770
C=======================================================================00003780
C====== THE USER MUST CHANGE THE FOLLOWING STATEMENTS FOR THE SPECIFIC  00003790
C====== NAMELIST VARIABLES DESIRED (E.G., USE TECO OR EDT, ETC.)========00003800
C====== DIMENSION NO_NAM VARIABLES TO AGREE WITH CHANGED DATA STATEMENTS00003810
C==                                                                     00003820
C==ON VAX USE THE FOLLOWING INCLUDE STATEMENT (OPTIONALLY, USE /LIST):  00003830
C==                                                                     00003840
C>>   INCLUDE 'INCLNAMES.FOR/NOLIST'                                   00003850
C                                                                       00003860
C========================= INCLNAM12.FT ================================00003870
C=================== FOR USE IN CALL NAMELIST ==========================00003880
C NORMALLY, ONE SHOULD COPY 'INCLNAM12.FT' TO 'INCLNAMES.FT'; THEN      00003890
C   EDIT 'INCLNAMES.FT' AS DESIRED FOR USERS CALL NAMELIST. NOTE THAT   00003900
C   ONE MUST RECOMPILE 'NAMELIST.FT' WITH USERS CALLING PROGRAM,        00003910
C   WHERE 'NAMELIST.FT' CONTAINS THE FOLLOWING STATEMENT:               00003920
C                                                                       00003930
C     INCLUDE 'INCLNAMES.FT/LIST'                                       00003940
C=======================================================================00003950
C                                                                       00003960
C***********************************************************************00003970
C  THIS IS "$PARMS INPUT" FOR PROGRAMS "TCILOOP" AND "TCOLOOP"          00003980
C***********************************************************************00003990
C                                                                       00004000
      COMMON/NAME_LIST/V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,                  00004010
     * V11,V12,V13,V14,V15,V16,V17,V18                                  00004020
      INTEGER V1,V7,V10,V13,V15,V16,V17,V18                             00004030
      DIMENSION V1(1),V2(10),V3(9),V4(1),                               00004040
     * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                           00004050
     * V11(1),V12(1),V13(1),V14(1),V15(1),                             00004060
     * V16(1),V17(1),V18(1),V19(1),V20(1),                             00004070
     * V21(1),V22(1),V23(1),V24(1),V25(1),                             00004080
     * V26(1),V27(1),V28(1),V29(1),V30(1),                             00004090
     * V31(1),V32(1),V33(1),V34(1),V35(1),                             00004100
     * V36(1),V37(1),V38(1),V39(1),V40(1),                             00004110
     * V41(1),V42(1),V43(1),V44(1),V45(1),                             00004120
     * V46(1),V47(1),V48(1),V49(1),V50(1),                             00004130
     * V51(1),V52(1),V53(1),V54(1),V55(1),                             00004140
     * V56(1),V57(1),V58(1),V59(1),V60(1)                              00004150
      DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                        00004160
      CHARACTER*6 NAM(60)                                               00004170
      DATA NAM/'M','SIG','H','A','Z','EPS','ISTEP',                     00004180
     1 'BO','BM','NB','TO','TM','NT','XNORM','IOUT',                    00004190
     2 'IOUTS','IPCH','ISTOP',42*' '/                                   00004200
      DATA NAMDIM/1,10,9,15*1,42*0/                                     00004210
      DATA NAMLEN/1,3,3*1,3,5,6*2,5,4,5,4,5,42*0/                       00004220
      DATA NAMTYP/0,5*1,0,2*1,0,2*1,0,1,4*0,42*0/                       00004230
      DATA NO_NAM/18/                                                   00004240
C====== END OF INCLUDE STATEMENTS ====================================00004250
C                                                                       00004260
C==                                                                     00004270
C== FOR EXAMPLE, FILE 'INCLNAMES.FOR' MAY CONTAIN (WITHOUT "C=="):      00004280
C==                                                                     00004290
```

```
C==       COMMON/NAME_LIST/V1,V2,V3,V4                                  00004300
C==       REAL*8 V1                                                     00004310
C==       INTEGER V3                                                    00004320
C==       DIMENSION V1(1),V2(2),V3(3),V4(4),                            00004330
C==      * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                        00004340
C==      * V11(1),V12(1),V13(1),V14(1),V15(1),                          00004350
C==      * V16(1),V17(1),V18(1),V19(1),V20(1),                          00004360
C==      * V21(1),V22(1),V23(1),V24(1),V25(1),                          00004370
C==      * V26(1),V27(1),V28(1),V29(1),V30(1),                          00004380
C==      * V31(1),V32(1),V33(1),V34(1),V35(1),                          00004390
C==      * V36(1),V37(1),V38(1),V39(1),V40(1),                          00004400
C==      * V41(1),V42(1),V43(1),V44(1),V45(1),                          00004410
C==      * V46(1),V47(1),V48(1),V49(1),V50(1),                          00004420
C==      * V51(1),V52(1),V53(1),V54(1),V55(1),                          00004430
C==      * V56(1),V57(1),V58(1),V59(1),V60(1)                           00004440
C==       DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                    00004450
C==       CHARACTER*6 NAM(60)                                           00004460
C==       DATA NAM/'A','BB','ICC','DDD_4',56*' '/                       00004470
C==       DATA NAMDIM/1,2,3,4,56*0/                                     00004480
C==       DATA NAMLEN/1,2,3,5,56*0/                                     00004490
C==       DATA NAMTYP/2*1,0,1,56*0/                                     00004500
C==       DATA NO_NAM/4/                                                00004510
C====== END OF EXAMPLE INCLUDE STATEMENTS ==============================00004520
C                                                                       00004530
C***********************************************************************00004540
C NOTE: THE ABOVE EXAMPLE SIMULATES                                     00004550
C       'NAMELIST/NAME/A,BB,ICC,DDD_4'                                  00004560
C       'READ(IUNIT,NAME,END=EOF)'                                      00004570
C       'READ(IUNIT,ANYNAM,END=EOF)'                                    00004580
C       IN THE CALLING PROGRAM USING:                                   00004590
C       ...                                                             00004600
C       REAL*8 A                                                        00004610
C       ...                                                             00004620
C       COMMON/NAME_LIST/A,BB(2),ICC(3),DDD_4(4)                        00004630
C       ...                                                             00004640
C       CALL NAMELIST(IUNIT,'$NAME',*EOF)                               00004650
C       ...                                                             00004660
C       CALL NAMELIST(IUNIT,'$ANYNAM',*EOF)                             00004670
C       ...                                                             00004680
C***********************************************************************00004690
C                                                                       00004700
      DATA C/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',  00004710
     * 'O','P','Q','R','S','T','U','V','W','X','Y','Z','_',            00004720
     * '1','2','3','4','5','6','7','8','9','0',                        00004730
     * ' ','$','=',',','(','*',')','.','+','-'/                        00004740
      J=LEN(NAME)                                                       00004750
      IF(J.LT.2.OR.J.GT.7) THEN                                         00004760
        CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//               00004770
     1 NAME//'   (LENGTH<2 OR >7 CHAR''S)',1,6,0)                       00004780
      ENDIF                                                             00004790
      IF(NAME(1:1).NE.'$')                                              00004800
     1 CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//                00004810
     2 NAME//'   (1ST CHAR MUST BE "$" CHAR)',1,6,0)                    00004820
C--INITIALIZE                                                           00004830
      INAME=0                                                           00004840
10    READ(IUNIT,11,END=99991,ERR=99992) BUF                            00004850
-11   -FORMAT(A80)                                                      00004860
```

```
      IF(INAME.EQ.1) GO TO 20                                          00004870
C--LOOK FOR "$NAME"                                                    00004880
        I=INDEX(BUF,NAME)                                              00004890
      IF(I.EQ.0) GO TO 10                                              00004900
      INAME=1                                                          00004910
      ICOL=I+J                                                         00004920
      JNAM=0                                                           00004930
      ILEN=0                                                           00004940
      VARNAM=' '                                                       00004950
      NUMLEN=0                                                         00004960
      IELE=1                                                           00004970
      GO TO 30                                                         00004980
20    ICOL=1                                                           00004990
30    CALL NONBLANK(BUF,LENBUF)                                        00005000
C==BEGIN PARSER LOOP (THE BIG 20000 LOOP)                              00005010
      IEND=0                                                           00005020
      DO 20000 I=ICOL,LENBUF                                           00005030
         BUFI=BUF(I:I)                                                 00005040
      DO 40 IC=1,27                                                    00005050
         IF(BUFI.EQ.C(IC)) GO TO 100                                   00005060
40    CONTINUE                                                         00005070
      DO 50 IC=28,37                                                   00005080
      IF(BUFI.EQ.C(IC)) GO TO 200                                      00005090
50    CONTINUE                                                         00005100
      DO 60 IC=38,47                                                   00005110
      IC_=IC-37                                                        00005120
      IF(BUFI.EQ.C(IC)) GO TO 70                                       00005130
60    CONTINUE                                                         00005140
61    WRITE(6,66) I,BUF                                                00005150
66    FORMAT(/' {NAMELIST}: ERROR IN FOLLOWING RECORD AT COL(',I2,'):'/ 00005160
     1 1X,A80/<I>X,'^')                                                00005170
      CALL ERRMSG('ILLEGAL CHAR="'//BUFI//'" FOUND',0,6,0)             00005180
67    WRITE(6,66) I,BUF                                                00005190
      CALL ERRMSG('NUMLEN<1 IN DECODEIX    ',0,6,0)                    00005200
68    WRITE(6,66) I,BUF                                                00005210
      CALL ERRMSG('NUMLEN<1 IN DECODEX',0,6,0)                         00005220
70    GO TO (20000,72,73,74,75,76,77,78,79,79),IC_                     00005230
C--'$' CHAR                                                            00005240
72    IEND=1                                                           00005250
      IF(NUMLEN.GT.0) GO TO 798                                        00005260
      IF(JNAM.EQ.0) GO TO 99990                                        00005270
      WRITE(6,66) I,BUF                                                00005280
      CALL ERRMSG('MISPLACED "$" CHAR',0,6,0)                          00005290
C--'=' CHAR                                                            00005300
73    IEQ=1                                                            00005310
C--CHECK FOR VALID VARNAM, LENGTH ILEN, ETC.                           00005320
      IF(ILEN.LT.1) GO TO 733                                          00005330
      DO 732 J=1,NO_NAM                                                00005340
      JNAM=J                                                           00005350
      JLEN=NAMLEN(J)                                                   00005360
      IF(JLEN.NE.ILEN) GO TO 732                                       00005370
      DO 731 K=1,JLEN                                                  00005380
      IF(VARNAM(K:K).NE.NAM(JNAM)(K:K)) GO TO 732                      00005390
731   CONTINUE                                                         00005400
C--VARNAM VERIFIED OK TO PROCEED TO NUMFLD(S)                          00005410
C                                                                      00005420
      IDIM=NAMDIM(JNAM)                                                00005430
```

```
      NUMLEN=0                                                          00005440
      NDEC=0                                                            00005450
      NREP=1                                                            00005460
      NEXP=0                                                            00005470
      GO TO 20000                                                       00005480
732   CONTINUE                                                          00005490
      WRITE(6,66) I,BUF                                                 00005500
      CALL ERRMSG('ILLEGAL VARNAM='//VARNAM//' FOUND',0,6,0)            00005510
733   WRITE(6,66) I,BUF                                                 00005520
      CALL ERRMSG('MISPLACED "=" CHAR  ',0,6,0)                         00005530
C--',' CHAR                                                             00005540
74    IF(NUMLEN.GT.0) GO TO 799                                         00005550
      WRITE(6,66) I,BUF                                                 00005560
      CALL ERRMSG('MISPLACED "," CHAR',0,6,0)                           00005570
C--'(' CHAR                                                             00005580
75    IELE=0                                                            00005590
      GO TO 20000                                                       00005600
C--'*' CHAR                                                             00005610
76    IF(JNAM.EQ.0.OR.NUMLEN.LT.1.OR.NUMLEN.GT.5) GO TO 767             00005620
760   CALL DECODEIX(NUMFLD,NUMLEN,NREP,*67)                             00005630
      NUMLEN=0                                                          00005640
      IF(NREP.GT.0.AND.NREP.LE.NAMDIM(JNAM)) GO TO 20000                00005650
      WRITE(6,66) I,BUF                                                 00005660
      CALL ERRMSG('REPEAT FACTOR <1 OR >NAMDIM   ',0,6,0)               00005670
767   WRITE(6,66) I,BUF                                                 00005680
      CALL ERRMSG('REPEAT WIDTH > 5 OR MISPLACED "*" CHAR',0,6,0)       00005690
C--')' CHAR                                                             00005700
77    IF(IELE.NE.0) GO TO 772                                           00005710
      CALL DECODEIX(NUMFLD,NUMLEN,IELE,*67)                             00005720
      IF(IELE.LT.1) GO TO 773                                           00005730
      NREP=1                                                            00005740
      GO TO 20000                                                       00005750
772   WRITE(6,66) I,BUF                                                 00005760
      CALL ERRMSG('MISPLACED ")" CHAR',0,6,0)                           00005770
773   WRITE(6,66) I,BUF                                                 00005780
      CALL ERRMSG('ARRAY IELE<1 OR >NAMDIM  ',0,6,0)                    00005790
C--'.' CHAR                                                             00005800
78    IF(JNAM.EQ.0.OR.NEXP.GT.0.OR.NDEC.GT.0) GO TO 781                 00005810
      NDEC=NUMLEN+1                                                     00005820
      IF(NAMTYP(JNAM).EQ.1) GO TO 200                                   00005830
781   WRITE(6,66) I,BUF                                                 00005840
      CALL ERRMSG('MISPLACED "." CHAR',0,6,0)                           00005850
C--'-' OR '+' CHAR                                                      00005860
79    IF(IELE.GT.0.OR.NEXP.GT.0) GO TO 210                              00005870
      WRITE(6,66) I,BUF                                                 00005880
      CALL ERRMSG('MISPLACED "-" OR "+" CHAR',0,6,0)                    00005890
C--<ALP> CHAR                                                           00005900
100   IF(NUMLEN.GT.0) GO TO 209                                         00005910
      IF(ILEN.GT.0) GO TO 102                                           00005920
      IEQ=0                                                             00005930
      IELE=1                                                            00005940
102   ILEN=ILEN+1                                                       00005950
      IF(ILEN.GT.6) GO TO 101                                           00005960
      VARNAM(ILEN:ILEN)=BUFI                                            00005970
      GO TO 20000                                                       00005980
101   WRITE(6,66) I,BUF                                                 00005990
      CALL ERRMSG('VARNAM>6 CHAR''S',0,6,0)                             00006000
```

```
C--<+-NUM> CHAR                                                    00006010
200    IF(IELE.EQ.0) GO TO 210                                     00006020
       IF(IEQ.EQ.0) GO TO 102                                      00006030
       GO TO 210                                                   00006040
209    IF(BUFI.EQ.'E'.OR.BUFI.EQ.'D') THEN                         00006050
         NEXP=NUMLEN+1                                             00006060
       ELSE                                                        00006070
         GO TO 61                                                  00006080
       ENDIF                                                       00006090
210    NUMLEN=NUMLEN+1                                             00006100
       IF(NUMLEN.GT.20) GO TO 211                                  00006110
       NUMFLD(NUMLEN:NUMLEN)=BUFI                                  00006120
       GO TO 20000                                                 00006130
211    WRITE(6,66) I,BUF                                           00006140
       CALL ERRMSG('NUM FIELD>20 CHAR''S',0,6,0)                   00006150
C--PROCESS NUMBER FIELD                                            00006160
799    IDIM=IDIM-1                                                 00006170
       IF(IDIM.LT.0) GO TO 10004                                   00006180
798    IF(NEXP.GT.0) GO TO 1000                                    00006190
C--[NEXP=0]                                                        00006200
       IF(NDEC.GT.0) GO TO 899                                     00006210
C--[NEXP=0, NDEC=0]                                                00006220
       CALL DECODEIX(NUMFLD,NUMLEN,IX,*67)                         00006230
C--CONVERT IX AND STORE IN COMMON                                  00006240
800    X=IX                                                        00006250
       IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                          00006260
8000   GO TO (801,802,803,804,805,806,807,808,809,810,             00006270
     * 811,812,813,814,815,816,817,818,819,820,                    00006280
     * 821,822,823,824,825,826,827,828,829,830,                    00006290
     * 831,832,833,834,835,836,837,838,839,840,                    00006300
     * 841,842,843,844,845,846,847,848,849,850,                    00006310
     * 851,852,853,854,855,856,857,858,859,860),JNAM               00006320
801    V1(IELE)=X                                                  00006330
       GO TO 10000                                                 00006340
802    V2(IELE)=X                                                  00006350
       GO TO 10000                                                 00006360
803    V3(IELE)=X                                                  00006370
       GO TO 10000                                                 00006380
804    V4(IELE)=X                                                  00006390
       GO TO 10000                                                 00006400
805    V5(IELE)=X                                                  00006410
       GO TO 10000                                                 00006420
806    V6(IELE)=X                                                  00006430
       GO TO 10000                                                 00006440
807    V7(IELE)=X                                                  00006450
       GO TO 10000                                                 00006460
808    V8(IELE)=X                                                  00006470
       GO TO 10000                                                 00006480
809    V9(IELE)=X                                                  00006490
       GO TO 10000                                                 00006500
810    V10(IELE)=X                                                 00006510
       GO TO 10000                                                 00006520
811    V11(IELE)=X                                                 00006530
       GO TO 10000                                                 00006540
812    V12(IELE)=X                                                 00006550
       GO TO 10000                                                 00006560
813    V13(IELE)=X                                                 00006570
```

```
      GO TO 10000                                                    00006580
814   V14(IELE)=X                                                    00006590
      GO TO 10000                                                    00006600
815   V15(IELE)=X                                                    00006610
      GO TO 10000                                                    00006620
816   V16(IELE)=X                                                    00006630
      GO TO 10000                                                    00006640
817   V17(IELE)=X                                                    00006650
      GO TO 10000                                                    00006660
818   V18(IELE)=X                                                    00006670
      GO TO 10000                                                    00006680
819   V19(IELE)=X                                                    00006690
      GO TO 10000                                                    00006700
820   V20(IELE)=X                                                    00006710
      GO TO 10000                                                    00006720
821   V21(IELE)=X                                                    00006730
      GO TO 10000                                                    00006740
822   V22(IELE)=X                                                    00006750
      GO TO 10000                                                    00006760
823   V23(IELE)=X                                                    00006770
      GO TO 10000                                                    00006780
824   V24(IELE)=X                                                    00006790
      GO TO 10000                                                    00006800
825   V25(IELE)=X                                                    00006810
      GO TO 10000                                                    00006820
826   V26(IELE)=X                                                    00006830
      GO TO 10000                                                    00006840
827   V27(IELE)=X                                                    00006850
      GO TO 10000                                                    00006860
828   V28(IELE)=X                                                    00006870
      GO TO 10000                                                    00006880
829   V29(IELE)=X                                                    00006890
      GO TO 10000                                                    00006900
830   V30(IELE)=X                                                    00006910
      GO TO 10000                                                    00006920
831   V31(IELE)=X                                                    00006930
      GO TO 10000                                                    00006940
832   V32(IELE)=X                                                    00006950
      GO TO 10000                                                    00006960
833   V33(IELE)=X                                                    00006970
      GO TO 10000                                                    00006980
834   V34(IELE)=X                                                    00006990
      GO TO 10000                                                    00007000
835   V35(IELE)=X                                                    00007010
      GO TO 10000                                                    00007020
836   V36(IELE)=X                                                    00007030
      GO TO 10000                                                    00007040
837   V37(IELE)=X                                                    00007050
      GO TO 10000                                                    00007060
838   V38(IELE)=X                                                    00007070
      GO TO 10000                                                    00007080
839   V39(IELE)=X                                                    00007090
      GO TO 10000                                                    00007100
840   V40(IELE)=X                                                    00007110
      GO TO 10000                                                    00007120
841   V41(IELE)=X                                                    00007130
      GO TO 10000                                                    00007140
```

```
842     V42(IELE)=X                                                     00007150
        GO TO 10000                                                     00007160
843     V43(IELE)=X                                                     00007170
        GO TO 10000                                                     00007180
844     V44(IELE)=X                                                     00007190
        GO TO 10000                                                     00007200
845     V45(IELE)=X                                                     00007210
        GO TO 10000                                                     00007220
846     V46(IELE)=X                                                     00007230
        GO TO 10000                                                     00007240
847     V47(IELE)=X                                                     00007250
        GO TO 10000                                                     00007260
848     V48(IELE)=X                                                     00007270
        GO TO 10000                                                     00007280
849     V49(IELE)=X                                                     00007290
        GO TO 10000                                                     00007300
850     V50(IELE)=X                                                     00007310
        GO TO 10000                                                     00007320
851     V51(IELE)=X                                                     00007330
        GO TO 10000                                                     00007340
852     V52(IELE)=X                                                     00007350
        GO TO 10000                                                     00007360
853     V53(IELE)=X                                                     00007370
        GO TO 10000                                                     00007380
854     V54(IELE)=X                                                     00007390
        GO TO 10000                                                     00007400
855     V55(IELE)=X                                                     00007410
        GO TO 10000                                                     00007420
856     V56(IELE)=X                                                     00007430
        GO TO 10000                                                     00007440
857     V57(IELE)=X                                                     00007450
        GO TO 10000                                                     00007460
858     V58(IELE)=X                                                     00007470
        GO TO 10000                                                     00007480
859     V59(IELE)=X                                                     00007490
        GO TO 10000                                                     00007500
860     V60(IELE)=X                                                     00007510
        GO TO 10000                                                     00007520
C--[NEXP=0, NDEC>0]                                                     00007530
899     CALL DECODEX(NUMFLD,NUMLEN,NDEC,X,*68)                          00007540
C--CONVERT X AND STORE IN COMMON                                        00007550
900     IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                              00007560
        GO TO 8000                                                      00007570
C--[NEXP>0]                                                             00007580
1000    IF(NDEC.GT.0) GO TO 2000                                        00007590
C--[NEXP>0, NDEC=0]                                                     00007600
        CALL DECODEIX(NUMFLD,NEXP-1,IX,*67)                             00007610
        X=IX                                                            00007620
1002    J=1                                                             00007630
        DO 1001 K=NEXP+1,NUMLEN                                         00007640
        NUMFLD(J:J)=NUMFLD(K:K)                                         00007650
1001    J=J+1                                                           00007660
        CALL DECODEIX(NUMFLD,NUMLEN-NEXP,IE,*67)                        00007670
        X=X*10.**IE                                                     00007680
C** {LATER INSERT A CALL TO A OVERFLOW HANDLER, ETC.}                   00007690
        GO TO 900                                                       00007700
C--[NEXP>0, NDEC>0]                                                     00007710
```

```
2000  CALL DECODEX(NUMFLD,NEXP-1,NDEC,X,*68)                          00007720
      GO TO 1002                                                      00007730
C--NEXT IELE?                                                         00007740
10000 IELE=IELE+1                                                     00007750
      IF(IELE.GT.NAMDIM(JNAM)) GO TO 10002                            00007760
      IF(NREP.GT.1) GO TO 10003                                       00007770
10001 IF(IEND.EQ.1) GO TO 99990                                       00007780
      NUMLEN=0                                                        00007790
      NDEC=0                                                          00007800
      NEXP=0                                                          00007810
      NREP=1                                                          00007820
      ILEN=0                                                          00007830
      VARNAM=' '                                                      00007840
      GO TO 20000                                                     00007850
10002 IELE=1                                                          00007860
      GO TO 10001                                                     00007870
10003 NREP=NREP-1                                                     00007880
      IDIM=IDIM-1                                                     00007890
      IF(IDIM.GE.0) GO TO 8000                                        00007900
10004 WRITE(6,66) I,BUF                                               00007910
      CALL ERRMSG('TOO MANY ELEMENTS FOR GIVEN NAMDIM.',0,6,0)        00007920
C==END OF DO 20000   CONTINUE PARSER -OR- READ IN NEXT BUF, ETC.      00007930
20000 CONTINUE                                                        00007940
                GO TO 10                                              00007950
C--'$' CHAR (DELIMITER $[END] FOR THIS $NAME --$)                     00007960
99990 RETURN                                                          00007970
C--E.O.F. ON FILE IUNIT ENCOUNTERED.                                  00007980
99991 RETURN 1                                                        00007990
99992 CALL ERRMSG('CANNOT OPEN/READ CALL NAMELIST(IFILE,...)',1,6,0)  00008000
      END                                                             00008010
      SUBROUTINE CPUTIME(I1,I2)                                       00008020
C                                                                     00008030
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON 00008040
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).                     00008050
C                                                                     00008060
C  WILL EJECT FIRST IF I1>0 (OR I2>0).                                00008070
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).                              00008080
C                                                                     00008090
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:00008100
C                                                                     00008110
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!                      00008120
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM00008130
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.          00008140
C                                                                     00008150
      SAVE                                                            00008160
      INTEGER*4 ABSVAL(4),INCRVAL(4)                                  00008170
      CALL PROCINFO(ABSVAL,INCRVAL)                                   00008180
      TIMES=SECNDS(TIME0)                                             00008190
      MIN=TIMES/60.0                                                  00008200
      SEC=AMOD(TIMES,60.0)                                            00008210
      CPUSEC=INCRVAL(1)*.01                                           00008220
      IMIN=CPUSEC/60.0                                                00008230
      CSEC=AMOD(CPUSEC,60.0)                                          00008240
      PCPU=100.*(CPUSEC/TIMES)                                        00008250
      IF(I1.NE.0) THEN                                                00008260
        IF(I1.GT.0) THEN                                              00008270
          J=1                                                         00008280
```

```
        ELSE                                                            00008290
          J=0                                                           00008300
        ENDIF                                                           00008310
       WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,        00008320
      1 (INCRVAL(I),I=2,4)                                              00008330
60     FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',      00008340
      1 I4,' MIN.',F6.2,' SEC.)'/                                       00008350
      2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,                    00008360
      1 ' S.)   CPU % =',F6.2,'%'/                                      00008370
      3 ' BUF.I/O_COUNT=',I10/                                          00008380
      4 ' DIR.I/O_COUNT=',I10/                                          00008390
      5 ' PAGE_FAULTS=',2X,I10/                                         00008400
      6 ' ',65('$')//)                                                  00008410
       ENDIF                                                            00008420
       IF(I2.NE.0) THEN                                                 00008430
         IF(I2.GT.0) THEN                                               00008440
           J=1                                                          00008450
         ELSE                                                           00008460
           J=0                                                          00008470
         ENDIF                                                          00008480
       WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,        00008490
      1 (INCRVAL(I),I=2,4)                                              00008500
       ENDIF                                                            00008510
       RETURN                                                           00008520
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'      00008530
       ENTRY SETTIME()                                                  00008540
       TIME0=SECNDS(0.0)                                                00008550
       CALL PROCINFO(ABSVAL,INCRVAL)                                    00008560
       RETURN                                                           00008570
       END                                                              00008580
       SUBROUTINE DECODEIX(NUMFLD,NUMLEN,IX,*)                          00008590
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)                               00008600
       CHARACTER*9 FMT                                                  00008610
       CHARACTER*20 NUMFLD                                              00008620
       IF(NUMLEN.LT.1) RETURN 1                                         00008630
       IDIFF=20-NUMLEN                                                  00008640
       IF(IDIFF.EQ.0) THEN                                              00008650
         ENCODE(9,991,FMT) NUMLEN                                       00008660
       ELSE                                                             00008670
         ENCODE(9,992,FMT) NUMLEN,IDIFF                                 00008680
       ENDIF                                                            00008690
991    FORMAT('(I',I2,'    )')                                         00008700
992    FORMAT('(I',I2,',',I2,'X)')                                      00008710
       DECODE(9,FMT,NUMFLD) IX                                          00008720
       RETURN                                                           00008730
       END                                                              00008740
       SUBROUTINE DECODEX(NUMFLD,NUMLEN,NDEC,X,*)                       00008750
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)                               00008760
       CHARACTER*12 FMT                                                 00008770
       CHARACTER*20 NUMFLD                                              00008780
       IF(NUMLEN.LT.1) RETURN 1                                         00008790
       LENDEC=NUMLEN-NDEC                                               00008800
       IDIFF=20-NUMLEN                                                  00008810
       IF(IDIFF.EQ.0) THEN                                              00008820
        ENCODE(12,991,FMT) NUMLEN,LENDEC                                00008830
       ELSE                                                             00008840
        ENCODE(12,992,FMT) NUMLEN,LENDEC,IDIFF                          00008850
```

```
      ENDIF                                                              00008860
991   FORMAT('(F',I2,'.',I2,'    )')                                     00008870
992   FORMAT('(F',I2,'.',I2,',',I2,'X)')                                 00008880
      DECODE(12,FMT,NUMFLD) X                                            00008890
      RETURN                                                             00008900
      END                                                                00008910
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)                         00008920
C                                                                        00008930
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780                   00008940
C                                                                        00008950
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                   00008960
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2          00008970
C          > 0 FOR ONE BLANK LINE BEFORE.                                00008980
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).      00008990
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).      00009000
C                                                                        00009010
C  MESSAGES ARE WRITTEN IN THE FORM:                                     00009020
C                                                                        00009030
C  {ERRMSG}: _MSG_HERE_                                                  00009040
C                                                                        00009050
      CHARACTER*(*) MSG                                                  00009060
      I=LEN(MSG)                                                         00009070
      DO 1 J=1,2                                                         00009080
        IF(J.EQ.1) THEN                                                  00009090
       JUNIT=IUNIT1                                                      00009100
          ELSE                                                           00009110
       JUNIT=IUNIT2                                                      00009120
        ENDIF                                                            00009130
        IF(JUNIT.GT.0) THEN                                              00009140
          IF(ISKIP.EQ.0) THEN                                            00009150
            WRITE(JUNIT,2) MSG                                           00009160
            ELSE                                                         00009170
       WRITE(JUNIT,3) MSG                                                00009180
          ENDIF                                                          00009190
      ENDIF                                                              00009200
1     CONTINUE                                                           00009210
      CALL EXIT                                                          00009220
2     FORMAT(1X,'{ERRMSG}: ',A<I>)                                       00009230
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)                                      00009240
      END                                                                00009250
      SUBROUTINE INTEG1(N,X,Y,Y0)                                        00009260
C        THIS ROUTINE INTEGRATES A FUNCTION"S VALUES (Y                  00009270
C        AS A FUNCTION OF X) FROM 0 TO X BY CALCULATING THE CUBIC        00009280
C        SPLINE COEFFICIENTS AND INTEGRATING THE RESULTING               00009290
C        CUBIC POLYNOMIAL APPROXIMATION.  THE Y VALUES ARE               00009300
C        REPLACED BY THE INTEGRATED VALUES.                              00009310
C        Y0 IS THE VALUE OF Y AT X=0.0 (ASSUMES THAT ALL INPUT           00009320
C        X > 0).                                                         00009330
      DIMENSION X(N),Y(N)                                                00009340
      DIMENSION A(200),B(200),C(200),P(200),S(200),PS(2),X1(200),Y1(200)00009350
      DATA PS/0.0,0.0/                                                   00009360
      DO 1 I=1,N                                                         00009370
      X1(I+1)=X(I)                                                       00009380
    1 Y1(I+1)=Y(I)                                                       00009390
      X1(1)=0.0                                                          00009400
      Y1(1)=Y0                                                           00009410
      N1=N+1                                                             00009420
```

```
      CALL SPLIN1(N1,0,X1,Y1,A,B,C,0,PS,P,S)                            00009430
      Y(1)=X(1)*(Y0+X(1)*A(1)/2.+X(1)*X(1)*B(1)/3.+X(1)**3*C(1)/4.)     00009440
      N1=N-1                                                            00009450
      DO 10 I=1,N1                                                      00009460
      Z=X(I+1)-X(I)                                                     00009470
   10 Y(I+1)=Y(I)+Z*(Y1(I+1)+A(I+1)*Z/2.+B(I+1)*Z*Z/3.+C(I+1)*Z**3/4.)  00009480
      RETURN                                                            00009490
      END                                                               00009500
      SUBROUTINE MINMAX(A,N,AMIN,AMAX)                                  00009510
      DIMENSION A(1)                                                    00009520
      AMIN=A(1)                                                         00009530
      AMAX=AMIN                                                         00009540
      DO 1 I=2,N                                                        00009550
      AMIN=AMIN1(AMIN,A(I))                                             00009560
      AMAX=AMAX1(AMAX,A(I))                                             00009570
    1 CONTINUE                                                          00009580
      RETURN                                                            00009590
      END                                                               00009600
      SUBROUTINE NONBLANK(C,NB)                                         00009610
C--DETERMINE NON-BLANK CHAR LENGTH (=NB ON EXIT) OF C*(*)               00009620
C  NOTE THAT NB WILL BE IN [0,LEN(C)].                                  00009630
C                                                                       00009640
      CHARACTER*(*) C                                                   00009650
      L=LEN(C)                                                          00009660
      DO 10 I=L,1,-1                                                    00009670
        NB=I                                                            00009680
        IF(C(I:I).NE.' ') RETURN                                        00009690
10    CONTINUE                                                          00009700
      NB=0                                                              00009710
      RETURN                                                            00009720
      END                                                               00009730
      SUBROUTINE PROCINFO(ABS_VALUES,INCR_VALUES)                       00009740
C                                                                       00009750
C** SUBROUTINE TO OBTAIN ABSOLUTE AND INCREMENTAL VALUES OF PROCESS      00009760
C   PARAMETERS: CPU TIME, BUFFERED I/O COUNT, DIRECT I/O COUNT, AND     00009770
C   PAGE FAULTS.                                                        00009780
C                                                                       00009790
      IMPLICIT INTEGER*2(W),INTEGER*4(L)                                00009800
      PARAMETER (JPI$_CPUTIM = '00000407'X,                             00009810
     1 JPI$_BUFIO = '0000040C'X,JPI$_DIRIO = '0000040B'X,               00009820
     2 JPI$_PAGEFLTS= '0000040A'X)                                      00009830
      INTEGER*4 ABS_VALUES(4),INCR_VALUES(4),LCL_VALUES(4)              00009840
      COMMON/ITEMLIST/                                                  00009850
     1 W_LEN1,W_CODE1,L_ADDR1,L_LENADDR1,                               00009860
     2 W_LEN2,W_CODE2,L_ADDR2,L_LENADDR2,                               00009870
     3 W_LEN3,W_CODE3,L_ADDR3,L_LENADDR3,                               00009880
     4 W_LEN4,W_CODE4,L_ADDR4,L_LENADDR4,                               00009890
     5 W_LEN5,W_CODE5                                                   00009900
      DATA W_LEN1,W_LEN2,W_LEN3,W_LEN4,W_LEN5/5*4/                      00009910
      DATA W_CODE1/JPI$_CPUTIM/,                                        00009920
     1 W_CODE2/JPI$_BUFIO/,                                             00009930
     2 W_CODE3/JPI$_DIRIO/,                                             00009940
     3 W_CODE4/JPI$_PAGEFLTS/,                                          00009950
     4 W_CODE5/0/                                                       00009960
      DATA L_LENADDR1,L_LENADDR2,L_LENADDR3,L_LENADDR4/4*0/             00009970
      L_ADDR1=%LOC(LCL_VALUES(1))                                       00009980
      L_ADDR2=%LOC(LCL_VALUES(2))                                       00009990
```

```
      L_ADDR3=%LOC(LCL_VALUES(3))                                          00010000
      L_ADDR4=%LOC(LCL_VALUES(4))                                          00010010
C** PERFORM THE SYSTEM SERVICE CALL                                        00010020
      CALL SYS$GETJPI(,,,W_LEN1,,,)                                        00010030
C** ASSIGN THE NEW VALUES TO THE ARGUMENTS                                 00010040
      DO I=1,4                                                             00010050
        INCR_VALUES(I)=LCL_VALUES(I)-ABS_VALUES(I)                         00010060
        ABS_VALUES(I)=LCL_VALUES(I)                                        00010070
      END DO                                                               00010080
      RETURN                                                               00010090
      END                                                                  00010100
      REAL FUNCTION RFLAGS(N,FUN,TOL,T0,TM,T,NEW)                          00010110
C--FOURIER TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION                00010120
C  GIVES FOURIER COSINE OR SINE TRANSFORMS VIA RLAGF0,RLAGF1               00010130
C  REF: ANDERSON,1975,NTIS REPT. PB-242-800,P.76-87.                       00010140
C                                                                          00010150
C      N = 0 FOR COSINE TRANSFORM (VIA RLAGF0)                             00010160
C      N = 1 FOR SINE TRANSFORM (VIA RLAGF1)                               00010170
C    FUN = EXTERNAL REAL KERNEL FUNCTION.                                  00010180
C    TOL = TOLERANCE REQUESTED FOR RLAGF0 OR RLAGF1                        00010190
C     T0 = TMIN TO USE (E.G., LET T0=.5*TMIN, TMIN=TRUE)                   00010200
C     TM = TMAX TO USE (TM>T0)                                             00010210
C      T = TRANSFORM PARAMETER (T0<=T<=TM) FOR THIS CALL (NEW=1 OR 0)      00010220
C    NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS.        00010230
C    NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY.      00010240
C                                                                          00010250
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),                        00010260
     & D(2),W1(200),W2(200)                                                00010270
      EXTERNAL FUN                                                         00010280
      DATA D/2*0.0/                                                        00010290
      IF(NEW.EQ.0) GO TO 3                                                 00010300
      NT=AINT(5.*ALOG(TM/T0))+5                                            00010310
      IF(NT.GT.200)CALL ERRMSG('IN RFLAGS: NT>200   ',4,6,16)              00010320
      NT1=NT+1                                                             00010330
      X0=ALOG(T0)+.2*NT                                                    00010340
      NU=1                                                                 00010350
      DO 1 J=1,NT                                                          00010360
      I=NT1-J                                                              00010370
      X=X0-.2*J                                                            00010380
      EX=EXP(X)                                                            00010390
      ARG(I)=EX                                                            00010400
      IF(N.EQ.0) Y(I)=RLAGF0(X,FUN,TOL,L,NU)/EX                            00010410
      IF(N.NE.0) Y(I)=RLAGF1(X,FUN,TOL,L,NU)/EX                            00010420
1     NU=0                                                                 00010430
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)                         00010440
2     IF(NT.LT.0) CALL ERRMSG('IN RFLAGS: NT<0 AFTER SPLIN1  ',6,6,16)     00010450
3     IF(T.LT.T0) CALL ERRMSG('IN RFLAGS: T<T0',3,6,16)                    00010460
      IF(T.GT.TM) CALL ERRMSG('IN RFLAGS: T>TM',3,6,16)                    00010470
      CALL SPOINT(NT,ARG,Y,AR,BR,CR,T,X)                                   00010480
      RFLAGS=X                                                             00010490
      RETURN                                                               00010500
      END                                                                  00010510
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                            00010520
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.                 00010530
C                                                                          00010540
C        BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO        00010550
C                                                                          00010560
```

```
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2                              00010570
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),    00010580
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)            00010590
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).                         00010600
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).                         00010610
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)                         00010620
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.   00010630
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.             00010640
C           IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE      00010650
C              IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR           00010660
C              IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.              00010670
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.     00010680
C           P,S= WORK ARRAYS (EACH DIM=M).                               00010690
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.               00010700
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:                            00010710
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3             00010720
C       FOR I=1,2,...,M-1                                                00010730
C                                                                        00010740
C                                                                        00010750
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL                00010760
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999              00010770
      N=M-1                                                              00010780
      IF(IT.EQ.0) GO TO 20                                               00010790
C--1ST DERIVATIVE BOUNDARIES GIVEN                                       00010800
      NE=N-1                                                             00010810
      IF(H) 999,11,1                                                     00010820
C--EQUAL SPACING H .GT. 0. AND IT=1                                      00010830
    1 HH=3.0/H                                                           00010840
      DO 2 I=1,NE                                                        00010850
      B(I)=4.0                                                           00010860
      C(I)=1.0                                                           00010870
      A(I)=1.0                                                           00010880
    2 P(I)=HH*(Y(I+2)-Y(I))                                              00010890
      P(1)=P(1)-D(1)                                                     00010900
      P(NE)=P(NE)-D(2)                                                   00010910
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE                        00010920
    3 C(1)=C(1)/B(1)                                                     00010930
      P(1)=P(1)/B(1)                                                     00010940
      DO 4 I=2,NE                                                        00010950
      MUL=1.0/(B(I)-A(I)*C(I-1))                                         00010960
      C(I)=MUL*C(I)                                                      00010970
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))                                        00010980
C--OBTAIN SPLINE COEFFICIENTS                                            00010990
      A(NE+IT)=P(NE)                                                     00011000
      I=NE-1                                                             00011010
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)                                        00011020
      I=I-1                                                              00011030
       IF(I.GE.1) GO TO 5                                                00011040
      IF(IT.EQ.0) GO TO 6                                                00011050
      A(1)=D(1)                                                          00011060
      A(M)=D(2)                                                          00011070
    6 IF(H.EQ.0.) GO TO 14                                               00011080
      HH=1.0/H                                                           00011090
      DO 7 I=1,N                                                         00011100
      MUL=HH*(Y(I+1)-Y(I))                                               00011110
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))                                00011120
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))                                  00011130
```

```
      RETURN                                                            00011140
C--UNEQUAL SPACING H=0.. AND IT=1                                       00011150
   11 DO 12 I=1,N                                                       00011160
   12 S(I+1)=X(I+1)-X(I)                                                00011170
      DO 13 I=1,NE                                                      00011180
      B(I)=2.0*(S(I+1)+S(I+2))                                          00011190
      C(I)=S(I+1)                                                       00011200
      A(I)=S(I+2)                                                       00011210
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/     00011220
     $ (S(I+1)*S(I+2))                                                  00011230
      P(1)=P(1)-S(3)*D(1)                                               00011240
      P(NE)=P(NE)-S(N)*D(2)                                             00011250
      GO TO 3                                                           00011260
   14 DO 15 I=1,N                                                       00011270
      HH=1.0/S(I+1)                                                     00011280
      MUL=(Y(I+1)-Y(I))*HH**2                                           00011290
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH                                 00011300
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2                              00011310
      RETURN                                                            00011320
C--2ND DERIVATIVE BOUNDARIES GIVEN                                      00011330
   20 NE=N+1                                                            00011340
      IF(H) 999,31,21                                                   00011350
C--EQUAL SPACING H .GT. 0 AND IT=0                                      00011360
   21 HH=3.0/H                                                          00011370
      DO 22 I=2,N                                                       00011380
      B(I)=4.0                                                          00011390
      C(I)=1.0                                                          00011400
      A(I)=1.0                                                          00011410
   22 P(I)=HH*(Y(I+1)-Y(I-1))                                           00011420
      B(1)=2.0                                                          00011430
      B(NE)=2.0                                                         00011440
      C(1)=1.0                                                          00011450
      C(NE)=1.0                                                         00011460
      A(NE)=1.0                                                         00011470
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)                                    00011480
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)                                   00011490
      GO TO 3                                                           00011500
C--UNEQUAL SPACING H=0 AND IT=0                                         00011510
   31 DO 32 I=1,N                                                       00011520
   32 S(I+1)=X(I+1)-X(I)                                                00011530
      N1=N-1                                                            00011540
      DO 33 I=1,N1                                                      00011550
      B(I+1)=2.0*(S(I+1)+S(I+2))                                        00011560
      C(I+1)=S(I+1)                                                     00011570
      A(I+1)=S(I+2)                                                     00011580
   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)))/   00011590
     *      (S(I+1)*S(I+2))                                            00011600
      B(1)=2.0                                                          00011610
      B(NE)=2.0                                                         00011620
      C(1)=1.0                                                          00011630
      C(NE)=1.0                                                         00011640
      A(NE)=1.0                                                         00011650
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)                           00011660
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)                          00011670
      GO TO 3                                                           00011680
  999 M=-IABS(M)                                                        00011690
      RETURN                                                            00011700
```

```
      END                                                               00011710
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                              00011720
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y           00011730
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE       00011740
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).         00011750
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'        00011760
C  WHICH REQUIRES ONLY ONE CALL.                                        00011770
C                                                                       00011780
      DIMENSION X(1),Y(1),A(1),B(1),C(1)                                00011790
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9                              00011800
      M1=M-1                                                            00011810
      DO 1 I=1,M1                                                       00011820
      J=I                                                               00011830
      IF(XX.LE.X(I+1)) GO TO 2                                          00011840
    1 CONTINUE                                                          00011850
    9 WRITE(6,60) XX,X(1),X(M)                                          00011860
   60 FORMAT('0ERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL00011870
     * (',E16.8,',',E16.8,')')                                         00011880
      RETURN                                                            00011890
    2 Z=XX-X(J)                                                         00011900
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z                                  00011910
      RETURN                                                            00011920
      END                                                               00011930
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                       00011940
C {VAX-11/780 VERSION FORTRAN-77 (X3.9-1978); SEE NOTE(2) BELOW.}       00011950
C=======================================================================00011960
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00011970
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                                 00011980
C                                                                       00011990
C--REF: ANDERSON, W.L., 1979, GEOPHYSICS, VOL. 44, NO. 7, P. 1287-1305. 00012000
C                                                                       00012010
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF       00012020
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF        00012030
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY   00012040
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,       00012050
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.         00012060
C                                                                       00012070
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                    00012080
C                                                                       00012090
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED.  00012100
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00012110
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100012120
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).           00012130
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)   00012140
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00012150
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION           00012160
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00012170
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00012180
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH       00012190
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G)   00012200
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A  00012210
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED.    00012220
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE         00012230
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00012240
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00012250
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00012260
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS   00012270
```

```
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00012280
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY     00012290
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR   00012300
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.   00012310
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00012320
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00012330
C              USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00012340
C              EVALUATION -- SEE NOTE (1) BELOW).                        00012350
C              FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.       00012360
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION  00012370
C              FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).         00012380
C              NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,  00012390
C              NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.               00012400
C     NEW  =1  IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF      00012410
C              FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING      00012420
C              ZHANKS FOR UNRELATED KERNELS.                             00012430
C              NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE 00012440
C              FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING00012450
C              REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.        00012460
C     NEW  =0  TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED00012470
C              IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED           00012480
C              DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN  00012490
C              NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK. 00012500
C                                                                        00012510
C     ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE     00012520
C              COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN      00012530
C              THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).   00012540
C                                                                        00012550
C========================================================================00012560
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                        00012570
C     ...                                                                00012580
C     COMPLEX Z1,Z2,ZHANKS,FSAVE                                         00012590
C     COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00012600
C     EXTERNAL ZF1,ZF2                                                   00012610
C     ...                                                                00012620
C     Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                      00012630
C     DO 1 I=1,NSAVE                                                     00012640
C  C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.       00012650
C  C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)    00012660
C   1 CONTINUE                                                           00012670
C     Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                      00012680
C     ...                                                                00012690
C     END                                                                00012700
C     COMPLEX FUNCTION ZF1(G)                                            00012710
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.     00012720
C     END                                                                00012730
C     COMPLEX FUNCTION ZF2(G)                                            00012740
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.     00012750
C     END                                                                00012760
C========================================================================00012770
C--NOTES                                                                 00012780
C       (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.      00012790
C             THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS  00012800
C             EXP-UNDERFLOW TO 0.0.                                      00012810
C       (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT00012820
C             DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00012830
C             TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN     00012840
```

```
C             (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE00012850
C             SAVE Y1,ISAVE                                             00012860
C       (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT   00012870
C             IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),   00012880
C             BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.          00012890
C       (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),  00012900
C             HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS          00012910
C             IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).  00012920
C=============================================================================00012930
C                                                                       00012940
      SAVE Y1,ISAVE                                                     00012950
      COMPLEX FUN,C,CMAX,FSAVE                                          00012960
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                           00012970
      DOUBLE PRECISION E,ER,Y1,Y                                        00012980
      DIMENSION T(2),TMAX(2)                                            00012990
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),               00013000
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                        00013010
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),   00013020
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),              00013030
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                             00013040
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                               00013050
C-----E=DEXP(.2D0), ER=1.0D0/E                                          00013060
      DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/        00013070
C--J0-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)          00013080
      DATA WA0/                                                         00013090
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,      00013100
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,      00013110
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,      00013120
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,      00013130
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,      00013140
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,      00013150
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,      00013160
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,      00013170
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,      00013180
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,      00013190
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,      00013200
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,      00013210
     *-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,      00013220
     *-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,      00013230
     *-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,      00013240
     *-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,      00013250
     * 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,      00013260
     * 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,      00013270
     * 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/      00013280
      DATA WB0/                                                         00013290
     * 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,      00013300
     * 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,      00013310
     * 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,      00013320
     * 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,      00013330
     * 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,      00013340
     * 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,      00013350
     * 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,      00013360
     * 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,      00013370
     * 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,      00013380
     * 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,      00013390
     * 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,      00013400
     * 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,      00013410
```

```
     * 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,      00013420
     * 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,      00013430
     * 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,      00013440
     *-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00013450
     * 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00013460
     * 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00013470
     * 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00013480
      DATA WC0/                                                        00013490
     * 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00013500
     * 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00013510
     * 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00013520
     * 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00013530
     * 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00013540
     * 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00013550
     * 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00013560
     * 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00013570
     * 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00013580
     * 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00013590
     * 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00013600
     * 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00013610
     * 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00013620
     * 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00013630
     * 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00013640
     * 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00013650
     *-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00013660
     * 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00013670
     * 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00013680
      DATA WD0/                                                        00013690
     * 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00013700
     *-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00013710
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00013720
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00013730
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00013740
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00013750
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00013760
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00013770
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00013780
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00013790
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00013800
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00013810
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00013820
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                    00013830
C--END OF J0 FILTER WEIGHTS                                            00013840
C                                                                      00013850
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)         00013860
      DATA WA1/                                                        00013870
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,      00013880
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,      00013890
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00013900
     * 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,      00013910
     * 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,      00013920
     * 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,      00013930
     * 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,      00013940
     * 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,      00013950
     *-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,      00013960
     *-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,      00013970
     *-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,      00013980
```

```
*-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,      00013990
* 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,      00014000
* 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,      00014010
* 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,      00014020
* 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,      00014030
* 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,      00014040
* 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,      00014050
* 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/      00014060
 DATA WB1/                                                         00014070
* 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,      00014080
* 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,      00014090
* 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,      00014100
* 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,      00014110
* 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,      00014120
* 1.0021488E-07, 1.4950371E-07, 2.2303208E-07, 3.3272689E-07,      00014130
* 4.9636623E-07, 7.4049804E-07, 1.1046805E-06, 1.6480103E-06,      00014140
* 2.4585014E-06, 3.6677163E-06, 5.4714550E-06, 8.1626422E-06,      00014150
* 1.2176782E-05, 1.8166179E-05, 2.7099223E-05, 4.0428804E-05,      00014160
* 6.0307294E-05, 8.9971508E-05, 1.3420195E-04, 2.0021123E-04,      00014170
* 2.9860417E-04, 4.4545291E-04, 6.6423156E-04, 9.9073275E-04,      00014180
* 1.4767050E-03, 2.2016806E-03, 3.2788147E-03, 4.8837292E-03,      00014190
* 7.2596811E-03, 1.0788355E-02, 1.5973323E-02, 2.3612041E-02,      00014200
* 3.4655327E-02, 5.0608141E-02, 7.2827752E-02, 1.0337889E-01,      00014210
* 1.4207357E-01, 1.8821315E-01, 2.2996815E-01, 2.5088500E-01,      00014220
* 2.0334626E-01, 6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00014230
*-1.8280529E-01, 4.7014634E-01, 7.2991233E-03,-3.0614594E-01,      00014240
* 2.4781735E-01,-1.1149185E-01, 2.5985386E-02, 1.0850279E-02,      00014250
*-2.2830217E-02, 2.4644647E-02,-2.2895284E-02, 2.0197032E-02/      00014260
 DATA WC1/                                                         00014270
*-1.7488968E-02, 1.5057670E-02,-1.2953923E-02, 1.1153254E-02,      00014280
*-9.6138436E-03, 8.2952090E-03,-7.1628361E-03, 6.1882910E-03,      00014290
*-5.3482055E-03, 4.6232056E-03,-3.9970542E-03, 3.4560118E-03,      00014300
*-2.9883670E-03, 2.5840861E-03,-2.2345428E-03, 1.9323046E-03,      00014310
*-1.6709583E-03, 1.4449655E-03,-1.2495408E-03, 1.0805480E-03,      00014320
*-9.3441130E-04, 8.0803899E-04,-6.9875784E-04, 6.0425624E-04,      00014330
*-5.2253532E-04, 4.5186652E-04,-3.9075515E-04, 3.3790861E-04,      00014340
*-2.9220916E-04, 2.5269019E-04,-2.1851585E-04, 1.8896332E-04,      00014350
*-1.6340753E-04, 1.4130796E-04,-1.2219719E-04, 1.0567099E-04,      00014360
*-9.1379828E-05, 7.9021432E-05,-6.8334412E-05, 5.9092726E-05,      00014370
*-5.1100905E-05, 4.4189914E-05,-3.8213580E-05, 3.3045496E-05,      00014380
*-2.8576356E-05, 2.4711631E-05,-2.1369580E-05, 1.8479514E-05,      00014390
*-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00014400
*-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00014410
*-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00014420
*-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00014430
*-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00014440
*-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00014450
*-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00014460
 DATA WD1/                                                         00014470
*-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00014480
*-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00014490
*-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00014500
*-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00014510
*-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00014520
*-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00014530
*-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00014540
*-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00014550
```

```
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00014560
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00014570
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00014580
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00014590
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00014600
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00014610
C--END OF J1 FILTER WEIGHTS                                             00014620
C                                                                       00014630
      NONE=0                                                            00014640
      IF(NEW.EQ.0) GO TO 100                                            00014650
      NSAVE=0                                                           00014660
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                 00014670
      Y1=0.7358852661479794460D0/DBLE(B)                                00014680
  100 ZHANKS=(0.0,0.0)                                                  00014690
      CMAX=(0.0,0.0)                                                    00014700
      NF=0                                                              00014710
      Y=Y1                                                              00014720
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)    00014730
      ASSIGN 110 TO M                                                   00014740
      I=131                                                             00014750
      Y=Y*E                                                             00014760
      GO TO 200                                                         00014770
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                  00014780
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                  00014790
      I=I+1                                                             00014800
      Y=Y*E                                                             00014810
      IF(I.LE.149) GO TO 200                                            00014820
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                      00014830
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))       00014840
      CMAX=TOL*CMAX                                                     00014850
      ASSIGN 120 TO M                                                   00014860
      GO TO 200                                                         00014870
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                          00014880
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130       00014890
      I=I+1                                                             00014900
      Y=Y*E                                                             00014910
      IF(I.LE.283) GO TO 200                                            00014920
  130 Y=Y1                                                              00014930
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130                 00014940
      ASSIGN 140 TO M                                                   00014950
      I=130                                                             00014960
      GO TO 200                                                         00014970
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                           00014980
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.             00014990
     * NONE.EQ.0) GO TO 190                                            00015000
      I=I-1                                                             00015010
      Y=Y*ER                                                            00015020
      IF(I.GT.0) GO TO 200                                              00015030
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.                00015040
  190 ISAVE=1                                                           00015050
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE            00015060
      ZHANKS=ZHANKS/B                                                   00015070
      RETURN                                                            00015080
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY)    00015090
  200 G=SNGL(Y)                                                         00015100
      IF(NEW) 300,210,300                                               00015110
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                      00015120
```

```
      ISAVE0=ISAVE                                                      00015130
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                   00015140
      ISAVE=ISAVE+1                                                     00015150
      IF(ISAVE.LE.NSAVE) GO TO 220                                      00015160
      ISAVE=ISAVE0                                                      00015170
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                          00015180
      GO TO 300                                                         00015190
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.                  00015200
  240 C=FSAVE(ISAVE)                                                    00015210
      ISAVE=ISAVE+1                                                     00015220
C-----SWITCH ON ORDER N                                                 00015230
  250 IF(N) 270,260,270                                                 00015240
  260 C=C*WT0(I)                                                        00015250
      GO TO 280                                                         00015260
  270 C=C*WT1(I)                                                        00015270
  280 ZHANKS=ZHANKS+C                                                   00015280
      GO TO M,(110,120,140)                                             00015290
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)            00015300
  300 NSAVE=NSAVE+1                                                     00015310
      C=FUN(G)                                                          00015320
      NF=NF+1                                                           00015330
      FSAVE(NSAVE)=C                                                    00015340
      GSAVE(NSAVE)=G                                                    00015350
      GO TO 250                                                         00015360
      END                                                               00015370
        REAL FUNCTION RLAGF0(X,FUN,TOL,L,NEW)                           00015380
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00015390
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*COS(G*B)*DG' DEFINED AS THE   00015400
C  REAL FOURIER COSINE TRANSFORM WITH ARGUMENT X(=ALOG(B))              00015410
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND               00015420
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00015430
C                                                                       00015440
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                    00015450
C                                                                       00015460
C--PARAMETERS:                                                          00015470
C                                                                       00015480
C     * X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00015490
C               'RLAGF0' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,   00015500
C               SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,  00015510
C               THEN SUBPROGRAM 'RFOUR0' IS ADVISED FOR GENERAL USE.    00015520
C               (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).            00015530
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).     00015540
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN   00015550
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.                00015560
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE            00015570
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00015580
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,     00015590
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00015600
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,      00015610
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON       00015620
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,        00015630
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'  00015640
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00015650
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00015660
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,   00015670
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00015680
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE        00015690
```

```
C                  CONVOLUTION (L DEPENDS ON TOL AND FUN).                  00015700
C                MIN.L=24 AND MAX.L=281--WHICH COULD                        00015710
C                  OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING     00015720
C                  VERY FAST...                                             00015730
C     * NEW=     1 IS NECESSARY 1ST TIME OR BRAND NEW X.                    00015740
C                0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20           00015750
C                  IS ASSUMED INTERNALLY BY THIS ROUTINE.                   00015760
C                  NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL                 00015770
C                  STILL ASSUME X=(LAST X)-0.20 ANYWAY...                   00015780
C                  IT IS THE USERS RESPONSIBILITY TO NORMALIZE               00015790
C                  BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00015800
C                THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT         00015810
C                  TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A               00015820
C                  SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00015830
C                  ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...            00015840
C                  THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED               00015850
C                  KERNELS WILL BE USED IN THE LAGGED CONVOLUTION           00015860
C                  WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS       00015870
C                  WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)               00015880
C                                                                           00015890
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF0; THE FOURIER      00015900
C  TRANSFORM IS THEN RLAGF0/B WHICH IS TO BE COMPUTED AFTER EXIT FROM       00015910
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...              00015920
C                                                                           00015930
C--USAGE-- 'RLAGF0' IS CALLED AS FOLLOWS:                                   00015940
C     ...                                                                   00015950
C     EXTERNAL RF                                                           00015960
C     ...                                                                   00015970
C     R=RLAGF0(ALOG(B),RF,TOL,L,NEW)/B                                      00015980
C     ...                                                                   00015990
C     END                                                                   00016000
C     REAL FUNCTION RF(G)                                                   00016010
C     ...USER SUPPLIED CODE...                                              00016020
C     END                                                                   00016030
C                                                                           00016040
C--NOTES:                                                                   00016050
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM         00016060
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS           00016070
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                                  00016080
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION       00016090
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN               00016100
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,       00016110
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'         00016120
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING      00016130
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).        00016140
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                      00016150
C                  ...                                                      00016160
C                  NB=AINT(5.*ALOG(BMAX/BMIN))+1                            00016170
C                  NB1=NB+1                                                 00016180
C                  X0=ALOG(BMAX)+.2                                         00016190
C                  NEW=1                                                    00016200
C                  DO 1 J=1,NB                                              00016210
C                  I=NB1-J                                                  00016220
C                  X=X0-.2*J                                                00016230
C                  ARG(I)=EXP(X)                                            00016240
C                  ANS(I)=RLAGF0(X,RF,TOL,L,NEW)/ARG(I)                     00016250
C     1            NEW=0                                                    00016260
```

```
C                ...                                                            00016270
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR            00016280
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,          00016290
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)                  00016300
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                       00016310
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY                  00016320
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                      00016330
C     BMAX,BMIN AND BY SETTING NEW=1....                                        00016340
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00016350
C                                                                               00016360
C                                                                               00016370
        DIMENSION KEY(281),SAVE(281)                                            00016380
        DIMENSION YT(281),Y1(76),Y2(76),Y3(76),Y4(53)                           00016390
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1)),               00016400
     1 (YT(229),Y4(1))                                                          00016410
C--COS-EXTENDED FILTER WEIGHT ARRAYS:                                           00016420
      DATA Y1/                                                                  00016430
     1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14,              00016440
     2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14,              00016450
     3 1.7534103E-13, 7.9373498E-15, 2.1235658E-13, 7.9981520E-14,              00016460
     4 2.3815757E-13, 1.9714260E-13, 2.8920132E-13, 3.4161340E-13,              00016470
     5 4.0349917E-13, 5.2203885E-13, 5.9837223E-13, 7.8015306E-13,              00016480
     6 8.8911655E-13, 1.1709731E-12, 1.3165595E-12, 1.7578463E-12,              00016490
     7 1.9538564E-12, 2.6289768E-12, 2.9167697E-12, 3.9044344E-12,              00016500
     8 4.3927341E-12, 5.7526904E-12, 6.6569552E-12, 8.4555678E-12,              00016510
     9 1.0063229E-11, 1.2487964E-11, 1.5134682E-11, 1.8501488E-11,              00016520
     1 2.2720051E-11, 2.7452598E-11, 3.4025443E-11, 4.0875985E-11,              00016530
     2 5.0751668E-11, 6.1094382E-11, 7.5492982E-11, 9.1445759E-11,              00016540
     3 1.1227336E-10, 1.3676464E-10, 1.6720269E-10, 2.0423244E-10,              00016550
     4 2.4932743E-10, 3.0470661E-10, 3.7198526E-10, 4.5449934E-10,              00016560
     5 5.5502537E-10, 6.7793669E-10, 8.2810001E-10, 1.0112626E-09,              00016570
     6 1.2354800E-09, 1.5085255E-09, 1.8432253E-09, 2.2503397E-09,              00016580
     7 2.7499027E-09, 3.3569525E-09, 4.1025670E-09, 5.0077487E-09,              00016590
     8 6.1205950E-09, 7.4703399E-09, 9.1312760E-09, 1.1143911E-08,              00016600
     9 1.3622929E-08, 1.6623917E-08, 2.0324094E-08, 2.4798610E-08,              00016610
     1 3.0321709E-08, 3.6992986E-08, 4.5237482E-08, 5.5183434E-08/              00016620
      DATA Y2/                                                                  00016630
     1 6.7491070E-08, 8.2317946E-08, 1.0069271E-07, 1.2279375E-07,              00016640
     2 1.5022907E-07, 1.8316969E-07, 2.2413747E-07, 2.7322865E-07,              00016650
     3 3.3441046E-07, 4.0756197E-07, 4.9894278E-07, 6.0793233E-07,              00016660
     4 7.4443665E-07, 9.0679753E-07, 1.1107379E-06, 1.3525651E-06,              00016670
     5 1.6573073E-06, 2.0174273E-06, 2.4728798E-06, 3.0090445E-06,              00016680
     6 3.6898816E-06, 4.4879625E-06, 5.5059521E-06, 6.6935820E-06,              00016690
     7 8.2160716E-06, 9.9828691E-06, 1.2260527E-05, 1.4888061E-05,              00016700
     8 1.8296530E-05, 2.2202672E-05, 2.7305154E-05, 3.3109672E-05,              00016710
     9 4.0751046E-05, 4.9372484E-05, 6.0820947E-05, 7.3619571E-05,              00016720
     1 9.0780005E-05, 1.0976837E-04, 1.3550409E-04, 1.6365676E-04,              00016730
     2 2.0227521E-04, 2.4398338E-04, 3.0197018E-04, 3.6370760E-04,              00016740
     3 4.5083748E-04, 5.4213338E-04, 6.7315347E-04, 8.0800951E-04,              00016750
     4 1.0051938E-03, 1.2041401E-03, 1.5011708E-03, 1.7942344E-03,              00016760
     5 2.2421056E-03, 2.6730676E-03, 3.3490681E-03, 3.9815050E-03,              00016770
     6 5.0028666E-03, 5.9285668E-03, 7.4730905E-03, 8.8233510E-03,              00016780
     7 1.1160132E-02, 1.3119627E-02, 1.6653199E-02, 1.9472767E-02,              00016790
     8 2.4800811E-02, 2.8793704E-02, 3.6762063E-02, 4.2228780E-02,              00016800
     9 5.3905163E-02, 6.0804660E-02, 7.7081738E-02, 8.3874501E-02,              00016810
     1 1.0377190E-01, 1.0377718E-01, 1.1892208E-01, 9.0437429E-02/              00016820
      DATA Y3/                                                                  00016830
```

```
     1 7.1685138E-02,-3.9473064E-02,-1.5078720E-01,-4.0489859E-01,      00016840
     2-5.6018995E-01,-6.8050388E-01,-1.5094224E-01, 6.6304064E-01,      00016850
     3 1.3766748E+00,-8.0373222E-01,-1.0869629E+00, 1.2812892E+00,      00016860
     4-5.0341082E-01,-4.4274455E-02, 2.0913102E-01,-1.9999661E-01,      00016870
     5 1.5207664E-01,-1.0920260E-01, 7.8169956E-02,-5.6651561E-02,      00016880
     6 4.1611799E-02,-3.0880012E-02, 2.3072559E-02,-1.7311631E-02,      00016890
     7 1.3021442E-02,-9.8085025E-03, 7.3943529E-03,-5.5769518E-03,      00016900
     8 4.2073164E-03,-3.1745026E-03, 2.3954154E-03,-1.8076122E-03,      00016910
     9 1.3640816E-03,-1.0293934E-03, 7.7682952E-04,-5.8623518E-04,      00016920
     1 4.4240399E-04,-3.3386183E-04, 2.5195025E-04,-1.9013541E-04,      00016930
     2 1.4348659E-04,-1.0828284E-04, 8.1716174E-05,-6.1667509E-05,      00016940
     3 4.6537684E-05,-3.5119887E-05, 2.6503388E-05,-2.0000904E-05,      00016950
     4 1.5093768E-05,-1.1390572E-05, 8.5959318E-06,-6.4869407E-06,      00016960
     5 4.8953713E-06,-3.6942830E-06, 2.7878625E-06,-2.1038241E-06,      00016970
     6 1.5875917E-06,-1.1980090E-06, 9.0398030E-07,-6.8208296E-07,      00016980
     7 5.1458650E-07,-3.8817581E-07, 2.9272267E-07,-2.2067921E-07,      00016990
     8 1.6623514E-07,-1.2514102E-07, 9.4034535E-08,-7.0556837E-08,      00017000
     9 5.2741581E-08,-3.9298610E-08, 2.9107255E-08,-2.1413893E-08,      00017010
     1 1.5742032E-08,-1.1498608E-08, 8.7561571E-09,-7.2959446E-09/      00017020
      DATA Y4/                                                         00017030
     1 6.8816619E-09,-8.9679825E-09, 1.4258275E-08,-1.9564299E-08,      00017040
     2 2.0235313E-08,-1.4725545E-08, 5.4632820E-09, 3.5995580E-09,      00017050
     3-9.5287133E-09, 1.1460041E-08,-1.0250532E-08, 7.4641748E-09,      00017060
     4-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,      00017070
     5 7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,      00017080
     6 1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,      00017090
     7-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,      00017100
     8-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,      00017110
     9-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,      00017120
     1-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,      00017130
     2-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,      00017140
     3-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,      00017150
     4-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,      00017160
     5-5.6047478E-14/                                                  00017170
C--$$ENDATA                                                            00017180
      IF(NEW) 10,30,10                                                 00017190
10    LAG=-1                                                           00017200
      X0=-X-30.30251236                                                00017210
      DO 20 IR=1,281                                                   00017220
20    KEY(IR)=0                                                        00017230
30    LAG=LAG+1                                                        00017240
      RLAGF0=0.0                                                       00017250
      CMAX=0.0                                                         00017260
      L=0                                                              00017270
      ASSIGN 110 TO M                                                  00017280
      I=149                                                            00017290
      GO TO 200                                                        00017300
110   CMAX=AMAX1(ABS(C),CMAX)                                          00017310
      I=I+1                                                            00017320
      IF(I.LE.170) GO TO 200                                           00017330
      IF(CMAX.EQ.0.0) GO TO 150                                        00017340
      CMAX=TOL*CMAX                                                    00017350
      ASSIGN 120 TO M                                                  00017360
      I=148                                                            00017370
      GO TO 200                                                        00017380
120   IF(ABS(C).LE.CMAX) GO TO 130                                     00017390
      I=I-1                                                            00017400
```

```
      IF(I.GT.0) GO TO 200                                              00017410
130   ASSIGN 140 TO M                                                   00017420
      I=171                                                             00017430
      GO TO 200                                                         00017440
140   IF(ABS(C).LE.CMAX) GO TO 190                                      00017450
      I=I+1                                                             00017460
      IF(I.LE.281) GO TO 200                                            00017470
      GO TO 190                                                         00017480
150   ASSIGN 160 TO M                                                   00017490
      I=1                                                               00017500
      GO TO 200                                                         00017510
160   IF(C.EQ.0.0) GO TO 170                                            00017520
      I=I+1                                                             00017530
      IF(I.LE.148) GO TO 200                                            00017540
170   ASSIGN 180 TO M                                                   00017550
      I=281                                                             00017560
      GO TO 200                                                         00017570
180   IF(C.EQ.0.0) GO TO 190                                            00017580
      I=I-1                                                             00017590
      IF(I.GE.171) GO TO 200                                            00017600
190   RETURN                                                            00017610
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)              00017620
200   LOOK=I+LAG                                                        00017630
      IQ=LOOK/282                                                       00017640
      IR=MOD(LOOK,282)                                                  00017650
      IF(IR.EQ.0) IR=1                                                  00017660
      IROLL=IQ*281                                                      00017670
      IF(KEY(IR).LE.IROLL) GO TO 220                                    00017680
210   C=SAVE(IR)*YT(I)                                                  00017690
      RLAGF0=RLAGF0+C                                                   00017700
      L=L+1                                                             00017710
      GO TO M,(110,120,140,160,180)                                     00017720
220   KEY(IR)=IROLL+IR                                                  00017730
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                             00017740
      GO TO 210                                                         00017750
      END                                                               00017760
        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)                           00017770
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00017780
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*SIN(G*B)*DG' DEFINED AS THE   00017790
C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))                00017800
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND               00017810
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00017820
C                                                                       00017830
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                    00017840
C                                                                       00017850
C--PARAMETERS:                                                          00017860
C                                                                       00017870
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00017880
C                'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,  00017890
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00017900
C                THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.   00017910
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).           00017920
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).     00017930
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN   00017940
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.                00017950
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE            00017960
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00017970
```

```
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,      00017980
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00017990
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,       00018000
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON        00018010
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,         00018020
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'   00018030
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00018040
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN  00018050
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,    00018060
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...   00018070
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE         00018080
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).                00018090
C               MIN.L=20 AND MAX.L=266--WHICH COULD                      00018100
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING   00018110
C                 VERY FAST...                                           00018120
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.                  00018130
C               0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20         00018140
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.                 00018150
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL               00018160
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...                 00018170
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE            00018180
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00018190
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT       00018200
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A             00018210
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00018220
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...          00018230
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED             00018240
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION         00018250
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS     00018260
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)             00018270
C                                                                        00018280
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER    00018290
C  TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM    00018300
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...           00018310
C                                                                        00018320
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:                                00018330
C     ...                                                                00018340
C     EXTERNAL RF                                                        00018350
C     ...                                                                00018360
C     R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B                                   00018370
C     ...                                                                00018380
C     END                                                                00018390
C     REAL FUNCTION RF(G)                                                00018400
C     ...USER SUPPLIED CODE...                                           00018410
C     END                                                                00018420
C                                                                        00018430
C--NOTES:                                                                00018440
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM      00018450
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS        00018460
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                               00018470
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION    00018480
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN            00018490
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,    00018500
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'      00018510
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING   00018520
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).      00018530
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                   00018540
```

```
C                ...                                                    00018550
C                NB=AINT(5.*ALOG(BMAX/BMIN))+1                          00018560
C                NB1=NB+1                                               00018570
C                X0=ALOG(BMAX)+.2                                       00018580
C                NEW=1                                                  00018590
C                DO 1 J=1,NB                                            00018600
C                I=NB1-J                                                00018610
C                X=X0-.2*J                                              00018620
C                ARG(I)=EXP(X)                                          00018630
C                ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)                   00018640
C     1          NEW=0                                                  00018650
C                ...                                                    00018660
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR    00018670
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,   00018680
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)           00018690
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                00018700
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY           00018710
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW               00018720
C     BMAX,BMIN AND BY SETTING NEW=1....                                 00018730
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00018740
C                                                                        00018750
C                                                                        00018760
        DIMENSION KEY(266),SAVE(266)                                     00018770
        DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38)                    00018780
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),        00018790
     1 (WT(229),W4(1))                                                   00018800
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:                                    00018810
      DATA W1/                                                           00018820
     1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12,       00018830
     2 1.7236636E-12,-1.8227727E-12, 1.9255992E-12,-2.0335514E-12,       00018840
     3 2.1473541E-12,-2.2675549E-12, 2.3946842E-12,-2.5292661E-12,       00018850
     4 2.6718110E-12,-2.8227693E-12, 2.9825171E-12,-3.1514006E-12,       00018860
     5 3.3297565E-12,-3.5179095E-12, 3.7163306E-12,-3.9256378E-12,       00018870
     6 4.1464798E-12,-4.3794552E-12, 4.6252131E-12,-4.8845227E-12,       00018880
     7 5.1582809E-12,-5.4474462E-12, 5.7530277E-12,-6.0760464E-12,       00018890
     8 6.4175083E-12,-6.7783691E-12, 7.1595239E-12,-7.5618782E-12,       00018900
     9 7.9864477E-12,-8.4344110E-12, 8.9072422E-12,-9.4067705E-12,       00018910
     1 9.9349439E-12,-1.0493731E-11, 1.1084900E-11,-1.1709937E-11,       00018920
     2 1.2370354E-11,-1.3067414E-11, 1.3802200E-11,-1.4575980E-11,       00018930
     3 1.5390685E-11,-1.6249313E-11, 1.7155934E-11,-1.8115250E-11,       00018940
     4 1.9131898E-11,-2.0209795E-11, 2.1352159E-11,-2.2561735E-11,       00018950
     5 2.3840976E-11,-2.5192263E-11, 2.6618319E-11,-2.8122547E-11,       00018960
     6 2.9709129E-11,-3.1382870E-11, 3.3149030E-11,-3.5013168E-11,       00018970
     7 3.6981050E-11,-3.9058553E-11, 4.1251694E-11,-4.3566777E-11,       00018980
     8 4.6010537E-11,-4.8590396E-11, 5.1314761E-11,-5.4193353E-11,       00018990
     9 5.7236720E-11,-6.0455911E-11, 6.3861222E-11,-6.7461492E-11,       00019000
     1 7.1265224E-11,-7.5279775E-11, 7.9512249E-11,-8.3971327E-11/       00019010
      DATA W2/                                                           00019020
     1 8.8668961E-11,-9.3621900E-11, 9.8851764E-11,-1.0438319E-10,       00019030
     2 1.1024087E-10,-1.1644680E-10, 1.2301979E-10,-1.2997646E-10,       00019040
     3 1.3733244E-10,-1.4510363E-10, 1.5330772E-10,-1.6196550E-10,       00019050
     4 1.7110130E-10,-1.8074257E-10, 1.9091922E-10,-2.0166306E-10,       00019060
     5 2.1300756E-10,-2.2498755E-10, 2.3763936E-10,-2.5100098E-10,       00019070
     6 2.6511250E-10,-2.8001616E-10, 2.9575691E-10,-3.1238237E-10,       00019080
     7 3.2994314E-10,-3.4849209E-10, 3.6808529E-10,-3.8878042E-10,       00019090
     8 4.1063982E-10,-4.3372666E-10, 4.5811059E-10,-4.8386049E-10,       00019100
     9 5.1105728E-10,-5.3977672E-10, 5.7011632E-10,-6.0215516E-10,       00019110
```

```
     1 6.3601273E-10,-6.7175964E-10, 7.0955028E-10,-7.4942601E-10,      00019120
     2 7.9161025E-10,-8.3606980E-10, 8.8317110E-10,-9.3270330E-10,      00019130
     3 9.8533749E-10,-1.0404508E-09, 1.0993731E-09,-1.1605442E-09,      00019140
     4 1.2267391E-09,-1.2942905E-09, 1.3691677E-09,-1.4429912E-09,      00019150
     5 1.5288164E-09,-1.6077524E-09, 1.7085998E-09,-1.7890471E-09,      00019160
     6 1.9129068E-09,-1.9857116E-09, 2.1491608E-09,-2.1926779E-09,      00019170
     7 2.4312660E-09,-2.3959044E-09, 2.7872500E-09,-2.5610596E-09,      00019180
     8 3.2762318E-09,-2.6082940E-09, 4.0261453E-09,-2.3560563E-09,      00019190
     9 5.3176554E-09,-1.3960161E-09, 7.7708747E-09, 1.1853546E-09,      00019200
     1 1.2760851E-08, 7.4264707E-09, 2.3342187E-08, 2.1869851E-08/      00019210
      DATA W3/                                                         00019220
     1 4.6306744E-08, 5.4631686E-08, 9.6763087E-08, 1.2823337E-07,      00019230
     2 2.0832812E-07, 2.9280540E-07, 4.5580888E-07, 6.5992437E-07,      00019240
     3 1.0056815E-06, 1.4779183E-06, 2.2284335E-06, 3.2994604E-06,      00019250
     4 4.9485823E-06, 7.3545473E-06, 1.1001083E-05, 1.6380539E-05,      00019260
     5 2.4469550E-05, 3.6469246E-05, 5.4441527E-05, 8.1176726E-05,      00019270
     6 1.2113828E-04, 1.8066494E-04, 2.6954609E-04, 4.0202288E-04,      00019280
     7 5.9969995E-04, 8.9437312E-04, 1.3338166E-03, 1.9886697E-03,      00019290
     8 2.9643943E-03, 4.4168923E-03, 6.5773518E-03, 9.7855105E-03,      00019300
     9 1.4539361E-02, 2.1558670E-02, 3.1871864E-02, 4.6903518E-02,      00019310
     1 6.8559512E-02, 9.9170152E-02, 1.4120770E-01, 1.9610835E-01,      00019320
     2 2.6192603E-01, 3.2743321E-01, 3.6407406E-01, 3.1257559E-01,      00019330
     3 9.0460168E-02,-3.6051039E-01,-8.6324760E-01,-8.1178720E-01,      00019340
     4 5.2205241E-01, 1.5449873E+00,-1.1817933E+00,-2.6759896E-01,      00019350
     5 8.0869203E-01,-6.2757149E-01, 3.4062630E-01,-1.5885304E-01,      00019360
     6 7.0472984E-02,-3.1624462E-02, 1.4894068E-02,-7.4821176E-03,      00019370
     7 4.0035936E-03,-2.2543784E-03, 1.3160358E-03,-7.8636604E-04,      00019380
     8 4.7658745E-04,-2.9125817E-04, 1.7885105E-04,-1.1012416E-04,      00019390
     9 6.7910334E-05,-4.1914054E-05, 2.5881544E-05,-1.5985851E-05,      00019400
     1 9.8751880E-06,-6.1008526E-06, 3.7692543E-06,-2.3287953E-06/      00019410
      DATA W4/                                                         00019420
     1 1.4388425E-06,-8.8899353E-07, 5.4926991E-07,-3.3937048E-07,      00019430
     2 2.0968284E-07,-1.2955437E-07, 8.0046336E-08,-4.9457371E-08,      00019440
     3 3.0557711E-08,-1.8880390E-08, 1.1665454E-08,-7.2076428E-09,      00019450
     4 4.4533423E-09,-2.7515696E-09, 1.7001092E-09,-1.0504494E-09,      00019460
     5 6.4904567E-10,-4.0102999E-10, 2.4778763E-10,-1.5310321E-10,      00019470
     6 9.4600354E-11,-5.8453314E-11, 3.6119400E-11,-2.2320056E-11,      00019480
     7 1.3793460E-11,-8.5242656E-12, 5.2675102E-12,-3.2543076E-12,      00019490
     8 2.0097689E-12,-1.2405412E-12, 7.6530538E-13,-4.7191929E-13,      00019500
     9 2.9084993E-13,-1.7923661E-13, 1.1018948E-13,-6.7885902E-14,      00019510
     1 4.2025050E-14,-2.1314731E-14/                                   00019520
C--$$ENDATA                                                           00019530
C                                                                     00019540
      IF(NEW) 10,30,10                                                00019550
10    LAG=-1                                                          00019560
      X0=-X-38.30455704                                               00019570
      DO 20 IR=1,266                                                  00019580
20    KEY(IR)=0                                                       00019590
30    LAG=LAG+1                                                       00019600
      RLAGF1=0.0                                                      00019610
      CMAX=0.0                                                        00019620
      L=0                                                             00019630
      ASSIGN 110 TO M                                                 00019640
      I=191                                                           00019650
      GO TO 200                                                       00019660
110   CMAX=AMAX1(ABS(C),CMAX)                                         00019670
      I=I+1                                                           00019680
```

```
      IF(I.LE.208) GO TO 200                                          00019690
      IF(CMAX.EQ.0.0) GO TO 150                                       00019700
      CMAX=TOL*CMAX                                                   00019710
      ASSIGN 120 TO M                                                 00019720
      I=190                                                           00019730
      GO TO 200                                                       00019740
120   IF(ABS(C).LE.CMAX) GO TO 130                                    00019750
      I=I-1                                                           00019760
      IF(I.GT.0) GO TO 200                                            00019770
130   ASSIGN 140 TO M                                                 00019780
      I=209                                                           00019790
      GO TO 200                                                       00019800
140   IF(ABS(C).LE.CMAX) GO TO 190                                    00019810
      I=I+1                                                           00019820
      IF(I.LE.266) GO TO 200                                          00019830
      GO TO 190                                                       00019840
150   ASSIGN 160 TO M                                                 00019850
      I=1                                                             00019860
      GO TO 200                                                       00019870
160   IF(C.EQ.0.0) GO TO 170                                          00019880
      I=I+1                                                           00019890
      IF(I.LE.190) GO TO 200                                          00019900
170   ASSIGN 180 TO M                                                 00019910
      I=266                                                           00019920
      GO TO 200                                                       00019930
180   IF(C.EQ.0.0) GO TO 190                                          00019940
      I=I-1                                                           00019950
      IF(I.GE.209) GO TO 200                                          00019960
190   RETURN                                                          00019970
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)            00019980
200   LOOK=I+LAG                                                      00019990
      IQ=LOOK/267                                                     00020000
      IR=MOD(LOOK,267)                                                00020010
      IF(IR.EQ.0) IR=1                                                00020020
      IROLL=IQ*266                                                    00020030
      IF(KEY(IR).LE.IROLL) GO TO 220                                  00020040
210   C=SAVE(IR)*WT(I)                                                00020050
      RLAGF1=RLAGF1+C                                                 00020060
      L=L+1                                                           00020070
      GO TO M,(110,120,140,160,180)                                   00020080
220   KEY(IR)=IROLL+IR                                                00020090
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                           00020100
      GO TO 210                                                       00020110
      END                                                             00020120
$
```