U.S. Department of the Interior
Geological Survey

Mail Stop 964
Box 25046, Federal Center
Denver, Colorado 80225

Programs TRANS_HCLOOP and TRANS_HZWIRE: Calculation of transient horizontal coplanar loop soundings and transient wire-loop soundings

by

Walter L. Anderson

OPEN-FILE REPORT 79-590

1979

CONTENTS.

By Walter L. Anderson

INTRODUCTION.

Program TRANS_HCLOOP is a general-purpose program for calculating the transient response for a ground horizontal coplanar loop-loop configuration (Frischknecht, 1967; Anderson, 1973) at the surface of a horizontally stratified earth for the quasi-static case (i.e., neglecting displacement currents). Program TRANS_HZWIRE is a similar program for calculating the transient response for a grounded dipole wire element and horizontal loop configuration on a layered earth. Fast lagged-convolution techniques for computing transient soundings (Kauahikaua and Anderson, 1977) are used in both programs. Digital filters developed by Anderson (1975, 1979) are used to evaluate Fourier and Hankel transform integrals.

Accuracy of the computed transient responses are generally good to 3-places (or better) while using the default accuracy-control parameters (see parameters eps, b0, bm, nb, and ihalf below). Greater accuracy is possible by changing some or all of these accuracy-control parameters, but at the expense of longer run times. The default parameters are usually adequate for most applications.

It should be noted that programs TRANS_HCLOOP and TRANS_HZWIRE supersede the EMTRAN programs published in Anderson (1973) for EMTRAN options n=0 and 1, respectively. Both programs listed in this report have identical parameters and operating instructions, and therefore only program TRANS_HCLOOP will be addressed below.

The transient impulse or step response soundings due to a system impulse or step driving current may be selected for any layered model (maximum of 10 layers), and computed over an arbitrary time range. Also, a specified transient shift (normalization) may be used. Both unnormalized and normalized forms are printed. The computed transient response may be written to a disk file for subsequent use.

To provide as much timely computer information as possible, this report is being released without a mathematical formulation section. The interested reader may consult the cited references for more details.

DETAIL PARAMETER DEFINITIONS.

$parms parameters (with defaults and cross-references):

m= Number of layers in the model (1<=m<=10; default m=1).
(cref: $parms sig,d).

sig()= Array of m layer conductivities (in mhos/meter, where sig(1)>0, and sig(i)>=0, i=2,...,m).
(cref: $parms m).

d()= Array of m-1 layer thicknesses (in meters, where d(i)>0,i=1,...,m-1; d ignored if m=1)
(cref: $parms m).

x0= Transmitter-receiver x-separation, where x0>=0 meters. [For program TRANS_HZWIRE, the dipole element is directed along the x-axis, centered at the origin. For program TRANS_HCLOOP, the transmitter loop is centered at the origin].

y0= Transmitter-receiver y-separation, where y0>0 meters must be given. Note the relevant separation distance is r=sqrt(x0*x0+y0*y0).

eps= Requested convolution integration tolerance used to compute all Hankel and Fourier transforms. (default .1e-5).

istep= 0 (default) for transient impulse response sounding, which corresponds to the time-derivative of Hz when the source uses a system step driving current (or Hz when the source uses a system impulse driving current).
= 1 for transient step response sounding, which corresponds to Hz when the source uses a system step driving current. Note istep=1 is generally used when the transient data is obtained using a SQUID or cryogenic magnetometer. If ihalf=1 and istep=1 is used, then the transient step response is computed as the integral over time of the impulse response (e.g., see Kauahikaua and Anderson, 1977, p.13).
(cref: $parms ihalf)

ihalf= 0 (default) to use the analytical transient for the halfspace impulse (istep=0) or step response (istep=1) separately from the secondary field numerical integrated transient (note ihalf=0 is usually more accurate than ihalf=1).
= 1 to include the halfspace field with the secondary

field to evaluate the numerical integrated transient response.
(cref: $parms istep)

b0 = .01 (default) is the lower induction number limit for which the mutual coupling Z/Z0 frequency response approaches 1.0 for b<b0. b0 must be given as a power of 10**-n.

bm = 100 (default) is the upper induction number limit for which the mutual coupling Z/Z0 frequency response approaches 0.0 for b>bm. bm must be given as a power of 10**n.

Note: b0=.01 and bm=100 are usually adequate for most applications. However, for more accuracy, b0<=.001 and/or bm>=1000 may be used. If parameter nb=0 is used, it is recommended that b0<.01 and bm>100 be used.
(cref: $parms nb).

nb = 10 (default) represents the number of induction number points per decade (log cycle) to evaluate the pre-splined frequency response function Z/Z0. In general, 5<=nb<12 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 via a lagged-convolution cosine or sine transform algorithm. When nb=0 or nb>=12, parameters b0<.01 and bm>100 should also be changed. Note nb=0 or nb>=12 is the most accurate method, but more time-consuming than using nb<12.
(cref: $parms b0,bm).

t0= Initial normalized time to compute the transient, where t0>0 must be specified as a power of 10**+-n. The normalized time (tau) and actual time (tsec in seconds) are related by the formula:
tau=(2.*tsec)/(sig(1)*fourpi*10**-7*r2), where r2=x0*x0+y0*y0.
(cref: $parms x0,y0,sig()).

tm= Maximum normalized time to compute the transient, where tm>t0 must be specified as a power of 10**+-n.
(cref: $parms t0).

nt= Number of normalized time points to compute per decade (log cycle) between t0 and tm, where nt>0 must be specified.
(cref: $parms t0,tm).

xnorm= Normalization factor (default 9.0 for TRANS_HCLOOP and 3.0 for TRANS_HZWIRE) to use at t0. Note: both the normalized and unnormalized transient response will be printed along with a normalization of 1.0 at t0 (e.g., see the test output in appendix 3).
(cref: $parms t0).

iout= 6 (default) is the primary print file unit number (usually assigned as the terminal "user output" on Multics). To suppress iout file output, set iout=0.

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

ipch= 0 (default) to ignore this option (i.e., file10 is not written).
= 1 to write file10 with the unnormalized transient response (trans) and time (tsec) in the format (2e16.8). This option may be used to generate input for other programs (e.g., plot routines, etc).

istop= 1 (default) to end the run after the current problem.
= 0 to continue the run with a new title line and changed $parms on file05. The program will continue until istop=1 is set on the last $parms or an end-of-file is encountered on file05 (if attached as a disk input file--see operating instructions below for attaching file05 on Multics).

$end [end of $parms parameters]
(note the "end" in $end may be excluded on Multics)

EXAMPLES OF INPUT PARAMETERS

```
example
$parms m=2,sig=.02,2,d=200,y0=2000,
 t0=.1,nt=6,tm=100,nb=6,istop=0$
modified example
$parms nb=10,y0=1000,istop=1$
```

MULTICS OPERATING INSTRUCTIONS.

1. Initially, one should add the following libraries (via the command "asr") to his search rules after the working directory: >udd>Emodl_inv>WAnderson>lib_em and >udd>Emodl_inv>WAnderson>lib_1.

2. Either attach "file05" to a predetermined ascii (stream) parameter file, or let file05 default to "user_input" (i.e., the user's terminal). The order of parameters and data on file05 must be given as defined in the section PARAMETERS REQUIRED above. To attach file05, type:
   io attach file05 vfile_ parameter_file_name

3. Set the underflow condition handler off by typing: set_ufl -off

4. Execute program TRANS_HCLOOP by typing: trans_hcloop
   [execute program TRANS_HZWIRE by typing: trans_hzwire]

If file05 was not attached, then the user must anticipate the required title and $parms to be typed on "user_input". Prompt messages are not printed on the terminal.

Note "file16" is a duplicate print file (normally disk on Multics), and "file06" is always the on-line terminal print file. File16 should either be deleted or dprinted to a line-printer after running program TRANS_HCLOOP. To submit the job as a batch job (called absentee on Multics), prepare step 1-4 above in a segment with .absin suffix and use the "enter_abs_request" command.

ERROR MESSAGES.

Most parameter and/or data errors are noted by self-explanatory messages appearing in the printed file(s), and the job is terminated. For example, the message "error--y0<=0" means that a violation (or omission) of the required parameter range has been committed in the $parms namelist. Check the named $parms value(s), correct, and resubmit the job.

Exponent underflow may occur when the argument is less than 10**-38 on Multics; this is ok since 0.0 replaces all underflows. To suppress the underflow messages, the command "set_ufl -off" can be used prior to executing TRANS_HCLOOP (or TRANS_HZWIRE).

REFERENCES.

Anderson, W.L., 1973, Fortran IV programs for the determination of the transient tangential electric field and vertical magnetic field about a vertical magnetic dipole for an m-layer stratified earth by numerical integration and digital linear filtering: U.S. Geol. Survey Rept. USGS-GD-73-017, avail. from U.S. Dept. Comm. NTIS, Springfield, Va. 22161 as Rept. PB-221-240, 82 p.

----------, 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geol. Survey Rept. USGS-GD-75-012, 223 p. avail. from U.S. Dept. Comm. NTIS, Springfield, Va. 22161 as Rept. PB-242-800/1WC.

----------, 1979 (in press), Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics.

Frischknecht, F.C., 1967,Fields about an oscillating magnetic dipole over a two-layer earth, and applications to ground and airborne electromagnetic surveys: Quarterly of the Colorado School of Mines, v. 62, no. 1, 326 p.

Kauahikaua, J., and Anderson, W.L., 1977, Calculation of standard transient and frequency sounding curves for a horizontal wire source of arbitrary length: U.S. Geol. Survey Rept. USGS-GD-77-007, 63 p. available from U.S. Dept. Comm. NTIS, Springfield, Va. 22161 as Rept. PB-274-119.

Appendix 1.-- Source listing

The attached subprograms are listed with beginning line numbers in the following order:

```
C--TRANS_HCLOOP: TRANSIENT HORIZONTAL COPLANAR LOOP SOUNDINGS--1/23/79. 00000010
C--TRANS_HZWIRE: TRANSIENT HZ HORIZONTAL WIRE-LOOP SOUNDINGS--1/23/79.  00001760
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00003500
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                         00003730
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                              00004930
        REAL FUNCTION RLAGF0(X,FUN,TOL,L,NEW)                           00005150
        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)                           00007540
      SUBROUTINE RECUR1(G,V1,F1)                                        00009900
      REAL FUNCTION ERF(X)                                              00010210
      REAL FUNCTION HCLOOP(B2)                                          00010340
      REAL FUNCTION HSLOOP(B2)                                          00010720
      SUBROUTINE INTEG1(N,X,Y,Y0)                                       00010800
      REAL FUNCTION RFLAGS(N,FUN,TOL,T0,TM,T,NEW)                       00011050
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                       00011460
      COMPLEX FUNCTION FG2(G)                                           00014880
      COMPLEX FUNCTION FG(G)                                            00015020
      REAL FUNCTION HZWIRE(B2)                                          00015100
      REAL FUNCTION HSWIRE(B2)                                          00015460
```

Note: Subprograms FG, HZWIRE, and HSWIRE are used only by program TRANS_HZWIRE.

Source Availability

The current version of the source code may be obtained by writing directly to the author. A magnetic tape copy of the source code will be sent to requestors to be copied and returned to the author. This method of releasing the program was selected in order to satisfy requests for the latest updated version. The magnetic tape will be recorded in the following mode (unless otherwise requested):

> Industry compatible: 9-track, unlabeled, EBCDIC mode, odd-parity, 800 bpi density, 80-character records (unblocked card images), and contained on one file.

```
C--TRANS_HCLOOP: TRANSIENT HORIZONTAL COPLANAR LOOP SOUNDINGS--1/23/79. 00000010
C  ** HONEYWELL MULTICS VERSION **                                      00000020
C  USES MUTUAL COUPLING RATIO Z/Z0 FOR SURFACE HORIZ LOOP-LOOP SYSTEM.  00000030
C                                                                       00000040
C  BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00000050
C                                                                       00000060
C--SEE DOCUMENTATION FOR TRANS_HCLOOP.                                  00000070
C  NOTE NORMALIZED TIME TAU USED FROM T0 TO TM, WHERE                   00000080
C  TIME=0.5*TAU*SIG1*(FOURPI*E-7)*RR**2  (TIME IN SEC.)                 00000090
C  I.E, TAU=(2.0*TIME)/(SIG1*FOURPI*E-7*RR**2)                          00000100
C  IPCH=1 OPTION (DEFAULT 0) WILL WRITE FILE10 WITH                     00000110
C  (TRANS,TIME) IN FORMAT (2E16.8)                                      00000120
C                                                                       00000130
C--CALLS RFLAGS, HSLOOP, AND HCLOOP TO COMPUTE THE TRANSIENT            00000140
C  USING LAGGED-CONVOLUTION IN TIME AND (DEPENDING ON NB OPTION)        00000150
C  DIRECT OR SPLINED FREQ FUNCTION DEFINED IN (B0,BM);                  00000160
C  NOTE Z/Z0=1.0 ASSUMED IF B<B0 AND Z/Z0=0.0 IF B>BM                   00000170
C  DEFAULT B0=.01, BM=100 IS USUALLY ADEQUATE FOR MOST MODELS.          00000180
C                                                                       00000190
C--FOLLOWING CHARACTER STATEMENT ONLY FOR MULTICS SYSTEM                00000200
      CHARACTER*5 TITLE                                                 00000210
      DIMENSION TITLE(16)                                               00000220
      REAL SIG(10),D(10),K(10),DD(9),DER(2),T(200),V(200)               00000230
      EXTERNAL HCLOOP,HSLOOP                                            00000240
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00000250
      COMMON/LOOP/D,RR,EPS,B0,BM,M1,IHALF                               00000260
      COMMON/MODEL/K,DD,M                                               00000270
      NAMELIST/PARMS/M,SIG,D,IPCH,XNORM,B0,BM,IHALF,                    00000280
     1 X0,Y0,EPS,IOUTS,ISTOP,T0,NT,TM,IOUT,ISTEP,NB                     00000290
C--STATEMENT FUNCTION HZ0(U) IS HALFSPACE IMPULSE RESPONSE              00000300
      HZ0(U)=9.*ERF(U)-1.128379167*U*EXP_(-U*U)*(9.+2.*U*U*(3.+2.*U*U)) 00000310
C--STATEMENT FUNCTION HZ1(U) IS HALFSPACE STEP RESPONSE                 00000320
      HZ1(U)=1.+ERF(U)*(9./(2.*U*U)-1.)-EXP_(-U*U)*.5641895835*         00000330
     * (9./U+4.*U)                                                     00000340
      DATA DER/2*0.0/                                                   00000350
C--PRESET                                                               00000360
      IHALF=0                                                           00000370
      IPCH=0                                                            00000380
      B0=.01                                                            00000390
      NB=10                                                             00000400
      BM=100.                                                           00000410
      ISTEP=0                                                           00000420
      XNORM=9.0                                                         00000430
      M=1                                                               00000440
      DO 10 I=1,10                                                      00000450
      SIG(I)=0.0                                                        00000460
10    D(I)=0.0                                                          00000470
      X0=0.0                                                            00000480
      Y0=0.0                                                            00000490
      EPS=.1E-5                                                         00000500
      ISTOP=1                                                           00000510
      IOUTS=16                                                          00000520
```

```
      T0=0.0                                                                00000530
      NT=0                                                                  00000540
      TM=0.0                                                                00000550
      IN=5                                                                  00000560
      IOUT=6                                                                00000570
20    READ(IN,30,END=999) TITLE                                             00000580
30    FORMAT(16A5)                                                          00000590
      READ(IN,PARMS,END=999)                                                00000600
      IF(IOUT.GT.0)                                                         00000610
     &  WRITE(6,40)                                                         00000620
     & TITLE,M,ISTEP,X0,Y0,NB,EPS,IOUT,XNORM,IPCH,                          00000630
     1 IOUTS,T0,NT,TM,ISTOP,B0,BM,IHALF,SIG,D                               00000640
40    FORMAT('1T R A N S _ H C L O O P  --  ',16A5//                        00000650
     &' M=',I2,10X,'ISTEP=',I3,7X,'X0=',E11.4,1X,                           00000660
     &'Y0=',E11.4/' NB=',I3,8X,'EPS=',E11.4,' IOUT=',I4,6X,                 00000670
     &'XNORM=',E9.3,2X,'IPCH=',I2/' IOUTS=',I3,5X,'T0=',E11.4,2X,'NT =',00000680
     &I5,6X,'TM=',E11.4,2X,' ISTOP=',I2/                                    00000690
     &' B0=',E10.4,' BM=',E11.4,2X,'IHALF=',I3//                            00000700
     &' SIG =',5E12.4/6X,5E12.4//                                           00000710
     &3X,'D =',5E12.4/6X,5E12.4)                                            00000720
      IF(IOUTS.GT.0)                                                        00000730
     & WRITE(IOUTS,40) TITLE,M,ISTEP,X0,Y0,NB,EPS,IOUT,XNORM,               00000740
     1 IPCH,IOUTS,T0,NT,TM,ISTOP,B0,BM,IHALF,SIG,D                          00000750
      IF(M.LT.1.OR.M.GT.10) CALL ERRMSG(20HM.LT.1.OR.M.GT.10   ,4,IOUT,     00000760
     1 IOUTS)                                                               00000770
      IF(Y0.LE.0.0)CALL ERRMSG('Y0<=0',1,IOUT,IOUTS)                        00000780
      IF(ISTEP.LT.0.OR.ISTEP.GT.1)                                          00000790
     & CALL ERRMSG('ISTEP<0 OR >1  ',3,IOUT,IOUTS)                          00000800
      IF(T0.LE.0.0.OR.TM.LE.T0)                                             00000810
     & CALL ERRMSG('T0<=0 OR TM<=T0',3,IOUT,IOUTS)                          00000820
      IF(NT.LE.0) CALL ERRMSG('NT<=0',1,IOUT,IOUTS)                         00000830
      IF(SIG(1).LE.0.0)CALL ERRMSG(15HSIG(1).LE.0.0  ,3,IOUT,IOUTS)         00000840
C--PRESET SOME CONSTANTS                                                    00000850
      IF(ISTEP.EQ.0) IPATH=0                                                00000860
      IF(ISTEP.EQ.1.AND.IHALF.EQ.0) IPATH=1                                 00000870
      IF(ISTEP.EQ.1.AND.IHALF.EQ.1) IPATH=-1                                00000880
      MM=M                                                                  00000890
      M1=M-1                                                                00000900
      XX=X0                                                                 00000910
      YY=Y0                                                                 00000920
      RR=SQRT(XX*XX+YY*YY)                                                  00000930
      SIG1=SIG(1)                                                           00000940
      DO 45 I=1,M                                                           00000950
45    K(I)=SIG(I)/SIG1                                                      00000960
      TCON=6.28318531E-7*SIG1*RR*RR                                         00000970
      ISPLN=0                                                               00000980
      IF(NB.GT.0.AND.NB.LT.12.AND.M.GT.1) ISPLN=1                           00000990
      IF(ISPLN.EQ.0) GO TO 49                                               00001000
C--GET PRE-SPLINED FREQ FUNCTION (0<NB<12 OPTION)                           00001010
      DB=EXP(2.30258509/FLOAT(NB))                                          00001020
      NS=0                                                                  00001030
      TEM=B0/DB                                                             00001040
```

```
      ISPLN=0                                                          00001050
46    TEM=TEM*DB                                                       00001060
      IF(TEM.GT.BM) GO TO 47                                           00001070
      NS=NS+1                                                          00001080
      IF(NS.GT.200) CALL ERRMSG('SPLINED NS>200.',3,6,16)              00001090
      XS(NS)=TEM                                                       00001100
      YS(NS)=HCLOOP(TEM*TEM)                                           00001110
      GO TO 46                                                         00001120
47    CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,V)                     00001130
      ISPLN=1                                                          00001140
49    NEW=1                                                            00001150
      DT=EXP(2.30258509/FLOAT(NT))                                     00001160
      IT=0                                                             00001170
      TEM=T0/DT                                                        00001180
      IF(IOUT.GT.0) WRITE(IOUT,50)                                     00001190
50    FORMAT('0',4X,'TAU(T0:TM)',3X,'TIME(SEC)',4X,'TRANS',8X,         00001200
     &'TRANS(NORM)',2X,'NORM*XNORM'/)                                  00001210
      IF(IOUTS.GT.0) WRITE(IOUTS,50)                                   00001220
60    TEM=TEM*DT                                                       00001230
      IF(TEM.GT.TM) GO TO 80                                           00001240
      TRANS=0.0                                                        00001250
      IF(M.EQ.1.AND.IHALF.EQ.0) GO TO 64                               00001260
C--GET TRANSIENT VIA RFLAGS (LAGGED-CONVOLUTION IN TIME)               00001270
      IF(IPATH.LE.0)                                                   00001280
     * TRANS=.63661977*RFLAGS(0,HCLOOP,EPS,0.5*T0,TM,TEM,NEW)          00001290
      IF(IPATH.EQ.1)                                                   00001300
     * TRANS=.63661977*RFLAGS(1,HSLOOP,EPS,0.5*T0,TM,TEM,NEW)          00001310
      NEW=0                                                            00001320
      IF(IHALF.EQ.1) GO TO 65                                          00001330
64    U=1.0/SQRT(2.*TEM)                                               00001340
      IF(IPATH.LE.0) TRANS=TRANS+HZ0(U)                                00001350
      IF(IPATH.EQ.1) TRANS=TRANS+HZ1(U)                                00001360
65    TIME=TCON*TEM                                                    00001370
      IT=IT+1                                                          00001380
      IF(IPATH.EQ.-1) GO TO 200                                        00001390
      IF(IT.EQ.1) TRANS1=TRANS                                         00001400
      TNORM=TRANS/TRANS1                                               00001410
      TXNORM=TNORM*XNORM                                               00001420
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM         00001430
70    FORMAT(1X,5E13.5)                                                00001440
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM       00001450
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                           00001460
100   FORMAT(2E16.8)                                                   00001470
      GO TO 60                                                         00001480
80    IF(IPATH.GE.0)  GO TO 82                                         00001490
C--GET STEP RESPONSE AS INTEGRAL OVER TIME OF IMPULSE RESPONSE         00001500
C  IF IPATH=-1 (I.E., IHALF=1, ISTEP=1)                                00001510
      CALL INTEG1(IT,T,V,9.0)                                          00001520
      TRANS1=V(1)                                                      00001530
      DO 81 I=1,IT                                                     00001540
      TEM=T(I)                                                         00001550
      TRANS=V(I)                                                       00001560
```

```
      TIME=TCON*TEM                                                    00001570
      TNORM=TRANS/TRANS1                                               00001580
      TXNORM=TNORM*XNORM                                               00001590
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM         00001600
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM       00001610
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                           00001620
81    CONTINUE                                                         00001630
82    IF(IOUTS.GT.0) WRITE(IOUTS,90)                                   00001640
90    FORMAT(129X)                                                     00001650
      IF(ISTOP.NE.1) GO TO 20                                          00001660
999   CALL CLOSE_FILE('-ALL')                                          00001670
      STOP                                                             00001680
C--SAVE IMPULSE RESPONSE FOR LATER STEP RESPONSE                       00001690
200   IF(IT.GT.200) CALL ERRMSG('IT>200 GEN WHEN ISTEP=1  ',5,IOUT,    00001700
     & IOUTS)                                                          00001710
      T(IT)=TEM                                                        00001720
      V(IT)=TRANS                                                      00001730
      GO TO 60                                                         00001740
      END                                                              00001750

C--TRANS_HZWIRE: TRANSIENT HZ HORIZONTAL WIRE-LOOP SOUNDINGS--1/23/79. 00001760
C  ** HONEYWELL MULTICS VERSION **                                     00001770
C  USES MUTUAL COUPLING RATIO Z/Z0 FOR SURFACE DIPOLE WIRE-LOOP SYSTEM. 00001780
C                                                                      00001790
C  BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.          00001800
C                                                                      00001810
C--SEE DOCUMENTATION FOR TRANS_HZWIRE.                                 00001820
C  NOTE NORMALIZED TIME TAU USED FROM T0 TO TM, WHERE                  00001830
C  TIME=0.5*TAU*SIG1*(FOURPI*E-7)*RR**2  (TIME IN SEC.)                00001840
C  I.E, TAU=(2.0*TIME)/(SIG1*FOURPI*E-7*RR**2)                         00001850
C  IPCH=1 OPTION (DEFAULT 0) WILL WRITE FILE10 WITH                    00001860
C  (TRANS,TIME) IN FORMAT (2E16.8)                                     00001870
C                                                                      00001880
C--CALLS RFLAGS, HSWIRE,  AND HZWIRE TO COMPUTE THE TRANSIENT          00001890
C  USING LAGGED-CONVOLUTION IN TIME AND (DEPENDING ON NB OPTION)       00001900
C  DIRECT OR SPLINED FREQ FUNCTION DEFINED IN (B0,BM);                 00001910
C  NOTE Z/Z0=1.0 ASSUMED IF B<B0 AND Z/Z0=0.0 IF B>BM                  00001920
C  DEFAULT B0=.01, BM=100 IS USUALLY ADEQUATE FOR MOST MODELS.         00001930
C                                                                      00001940
C--FOLLOWING CHARACTER STATEMENT ONLY FOR MULTICS SYSTEM               00001950
      CHARACTER*5 TITLE                                                00001960
      DIMENSION TITLE(16)                                              00001970
      REAL SIG(10),D(10),K(10),DD(9),DER(2),T(200),V(200)              00001980
      EXTERNAL HZWIRE,HSWIRE                                           00001990
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN     00002000
      COMMON/WIRE/D,RR,EPS,B0,BM,M1,IHALF                              00002010
      COMMON/MODEL/K,DD,M                                              00002020
      NAMELIST/PARMS/M,SIG,D,IPCH,XNORM,B0,BM,IHALF,                   00002030
     1 X0,Y0,EPS,IOUTS,ISTOP,T0,NT,TM,IOUT,ISTEP,NB                    00002040
C--STATEMENT FUNCTION HZ0(U) IS HALFSPACE IMPULSE RESPONSE             00002050
      HZ0(U)=3.*(ERF(U)-(1.128379167*(U+.666666667*U**3)*EXP_(-U*U)))  00002060
C--STATEMENT FUNCTION HZ1(U) IS HALFSPACE STEP RESPONSE                00002070
```

```
      HZ1(U)=1.+ERF(U)*(3./(2.*U*U)-1.)-EXP_(-U*U)*1.692568751/U          00002080
      DATA DER/2*0.0/                                                    00002090
C--PRESET                                                                00002100
      IHALF=0                                                            00002110
      IPCH=0                                                             00002120
      B0=.01                                                             00002130
      NB=10                                                              00002140
      BM=100.                                                            00002150
      ISTEP=0                                                            00002160
      XNORM=3.0                                                          00002170
      M=1                                                                00002180
      DO 10 I=1,10                                                       00002190
      SIG(I)=0.0                                                         00002200
10    D(I)=0.0                                                           00002210
      X0=0.0                                                             00002220
      Y0=0.0                                                             00002230
      EPS=.1E-5                                                          00002240
      ISTOP=1                                                            00002250
      IOUTS=16                                                           00002260
      T0=0.0                                                             00002270
      NT=0                                                               00002280
      TM=0.0                                                             00002290
      IN=5                                                               00002300
      IOUT=6                                                             00002310
20    READ(IN,30,END=999) TITLE                                          00002320
30    FORMAT(16A5)                                                       00002330
      READ(IN,PARMS,END=999)                                             00002340
      IF(IOUT.GT.0)                                                      00002350
     &  WRITE(6,40)                                                      00002360
     & TITLE,M,ISTEP,X0,Y0,NB,EPS,IOUT,XNORM,IPCH,                       00002370
     1 IOUTS,T0,NT,TM,ISTOP,B0,BM,IHALF,SIG,D                            00002380
40    FORMAT('1T R A N S _ H Z W I R E  --  ',16A5//                     00002390
     &' M=',I2,10X,'ISTEP=',I3,7X,'X0=',E11.4,1X,                        00002400
     &'Y0=',E11.4/' NB=',I3,8X,'EPS=',E11.4,' IOUT=',I4,6X,              00002410
     &'XNORM=',E9.3,2X,'IPCH=',I2/' IOUTS=',I3,5X,'T0=',E11.4,2X,'NT =',00002420
     &I5,6X,'TM=',E11.4,2X,' ISTOP=',I2/                                 00002430
     &' B0=',E10.4,' BM=',E11.4,2X,'IHALF=',I3//                         00002440
     &' SIG =',5E12.4/6X,5E12.4//                                        00002450
     &3X,'D =',5E12.4/6X,5E12.4)                                         00002460
      IF(IOUTS.GT.0)                                                     00002470
     & WRITE(IOUTS,40) TITLE,M,ISTEP,X0,Y0,NB,EPS,IOUT,XNORM,            00002480
     1 IPCH,IOUTS,T0,NT,TM,ISTOP,B0,BM,IHALF,SIG,D                       00002490
      IF(M.LT.1.OR.M.GT.10) CALL ERRMSG(20HM.LT.1.OR.M.GT.10   ,4,IOUT,  00002500
     1 IOUTS)                                                            00002510
      IF(Y0.LE.0.0)CALL ERRMSG('Y0<=0',1,IOUT,IOUTS)                     00002520
      IF(ISTEP.LT.0.OR.ISTEP.GT.1)                                       00002530
     & CALL ERRMSG('ISTEP<0 OR >1  ',3,IOUT,IOUTS)                       00002540
      IF(T0.LE.0.0.OR.TM.LE.T0)                                          00002550
     & CALL ERRMSG('T0<=0 OR TM<=T0',3,IOUT,IOUTS)                       00002560
      IF(NT.LE.0) CALL ERRMSG('NT<=0',1,IOUT,IOUTS)                      00002570
      IF(SIG(1).LE.0.0)CALL ERRMSG(15HSIG(1).LE.0.0  ,3,IOUT,IOUTS)      00002580
C--PRESET SOME CONSTANTS                                                 00002590
```

```
      IF(ISTEP.EQ.0) IPATH=0                                          00002600
      IF(ISTEP.EQ.1.AND.IHALF.EQ.0) IPATH=1                           00002610
      IF(ISTEP.EQ.1.AND.IHALF.EQ.1) IPATH=-1                          00002620
      MM=M                                                            00002630
      M1=M-1                                                          00002640
      XX=XO                                                           00002650
      YY=YO                                                           00002660
      RR=SQRT(XX*XX+YY*YY)                                            00002670
      SIG1=SIG(1)                                                     00002680
      DO 45 I=1,M                                                     00002690
45    K(I)=SIG(I)/SIG1                                                00002700
      TCON=6.28318531E-7*SIG1*RR*RR                                   00002710
      ISPLN=0                                                         00002720
      IF(NB.GT.0.AND.NB.LT.12.AND.M.GT.1) ISPLN=1                     00002730
      IF(ISPLN.EQ.0) GO TO 49                                         00002740
C--GET PRE-SPLINED FREQ FUNCTION (0<NB<12 OPTION)                     00002750
      DB=EXP(2.30258509/FLOAT(NB))                                    00002760
      NS=0                                                            00002770
      TEM=BO/DB                                                       00002780
      ISPLN=0                                                         00002790
46    TEM=TEM*DB                                                      00002800
      IF(TEM.GT.BM) GO TO 47                                          00002810
      NS=NS+1                                                         00002820
      IF(NS.GT.200) CALL ERRMSG('SPLINED NS>200.',3,6,16)             00002830
      XS(NS)=TEM                                                      00002840
      YS(NS)=HZWIRE(TEM*TEM)                                          00002850
      GO TO 46                                                        00002860
47    CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,V)                    00002870
      ISPLN=1                                                         00002880
49    NEW=1                                                           00002890
      DT=EXP(2.30258509/FLOAT(NT))                                    00002900
      IT=0                                                            00002910
      TEM=TO/DT                                                       00002920
      IF(IOUT.GT.0) WRITE(IOUT,50)                                    00002930
50    FORMAT('0',4X,'TAU(TO:TM)',3X,'TIME(SEC)',4X,'TRANS',8X,        00002940
     &'TRANS(NORM)',2X,'NORM*XNORM'/)                                 00002950
      IF(IOUTS.GT.0) WRITE(IOUTS,50)                                  00002960
60    TEM=TEM*DT                                                      00002970
      IF(TEM.GT.TM) GO TO 80                                          00002980
      TRANS=0.0                                                       00002990
      IF(M.EQ.1.AND.IHALF.EQ.0) GO TO 64                              00003000
C--GET TRANSIENT VIA RFLAGS (LAGGED-CONVOLUTION IN TIME)              00003010
      IF(IPATH.LE.0)                                                  00003020
     * TRANS=.63661977*RFLAGS(0,HZWIRE,EPS,0.5*TO,TM,TEM,NEW)         00003030
      IF(IPATH.EQ.1)                                                  00003040
     * TRANS=.63661977*RFLAGS(1,HSWIRE,EPS,0.5*TO,TM,TEM,NEW)         00003050
      NEW=0                                                           00003060
      IF(IHALF.EQ.1) GO TO 65                                         00003070
64    U=1.0/SQRT(2.*TEM)                                              00003080
      IF(IPATH.LE.0) TRANS=TRANS+HZO(U)                               00003090
      IF(IPATH.EQ.1) TRANS=TRANS+HZ1(U)                               00003100
65    TIME=TCON*TEM                                                   00003110
```

```
      IT=IT+1                                                          00003120
      IF(IPATH.EQ.-1) GO TO 200                                        00003130
      IF(IT.EQ.1) TRANS1=TRANS                                         00003140
      TNORM=TRANS/TRANS1                                               00003150
      TXNORM=TNORM*XNORM                                               00003160
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM         00003170
70    FORMAT(1X,5E13.5)                                                00003180
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM       00003190
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                           00003200
100   FORMAT(2E16.8)                                                   00003210
      GO TO 60                                                         00003220
80    IF(IPATH.GE.0)  GO TO 82                                         00003230
C--GET STEP RESPONSE AS INTEGRAL OVER TIME OF IMPULSE RESPONSE         00003240
C  IF IPATH=-1 (I.E., IHALF=1, ISTEP=1)                                00003250
      CALL INTEG1(IT,T,V,3.0)                                          00003260
      TRANS1=V(1)                                                      00003270
      DO 81 I=1,IT                                                     00003280
      TEM=T(I)                                                         00003290
      TRANS=V(I)                                                       00003300
      TIME=TCON*TEM                                                    00003310
      TNORM=TRANS/TRANS1                                               00003320
      TXNORM=TNORM*XNORM                                               00003330
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM         00003340
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM       00003350
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME                           00003360
81    CONTINUE                                                         00003370
82    IF(IOUTS.GT.0) WRITE(IOUTS,90)                                   00003380
90    FORMAT(129X)                                                     00003390
      IF(ISTOP.NE.1) GO TO 20                                          00003400
999   CALL CLOSE_FILE('-ALL')                                          00003410
      STOP                                                             00003420
C--SAVE IMPULSE RESPONSE FOR LATER STEP RESPONSE                       00003430
200   IF(IT.GT.200) CALL ERRMSG('IT>200 GEN WHEN ISTEP=1  ',5,IOUT,    00003440
     & IOUTS)                                                          00003450
      T(IT)=TEM                                                        00003460
      V(IT)=TRANS                                                      00003470
      GO TO 60                                                         00003480
      END                                                              00003490

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                  00003500
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                       00003510
C                                                                      00003520
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120              00003530
C               (USE NH-------- FORM FOR ANSI COMPATABILITY)           00003540
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24  00003550
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT. 00003560
C               IF I6.LE.0 UNIT I6 IGNORED.                            00003570
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                       00003580
C               IF I9.LE.0, UNIT I9 IGNORED.                           00003590
C--MESSAGE WRITTEN IN FORM--                                           00003600
C       /ERROR--MSG HERE                                               00003610
C                                                                      00003620
```

```
      DIMENSION MSG(30)                                                 00003630
      J=5*M5                                                            00003640
      K=J/4+MOD(J,4)                                                    00003650
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                           00003660
   10 FORMAT(/8H ERROR--,30A4)                                          00003670
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                           00003680
      CALL CLOSE_FILE('-ALL')                                           00003690
C                                                                       00003700
      STOP                                                              00003710
      END                                                               00003720

      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                         00003730
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.              00003740
C                                                                       00003750
C        BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO     00003760
C                                                                       00003770
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2                             00003780
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),   00003790
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)           00003800
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).                        00003810
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).                        00003820
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)                        00003830
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.  00003840
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.            00003850
C           IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE     00003860
C              IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR          00003870
C              IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.             00003880
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.    00003890
C           P,S= WORK ARRAYS (EACH DIM=M).                              00003900
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.              00003910
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:                           00003920
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3            00003930
C       FOR I=1,2,...,M-1                                               00003940
C                                                                       00003950
C                                                                       00003960
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL               00003970
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999             00003980
      N=M-1                                                             00003990
      IF(IT.EQ.0) GO TO 20                                              00004000
C--1ST DERIVATIVE BOUNDARIES GIVEN                                      00004010
      NE=N-1                                                            00004020
      IF(H) 999,11,1                                                    00004030
C--EQUAL SPACING H .GT. 0. AND IT=1                                     00004040
    1 HH=3.0/H                                                          00004050
      DO 2 I=1,NE                                                       00004060
      B(I)=4.0                                                          00004070
      C(I)=1.0                                                          00004080
      A(I)=1.0                                                          00004090
    2 P(I)=HH*(Y(I+2)-Y(I))                                             00004100
      P(1)=P(1)-D(1)                                                    00004110
      P(NE)=P(NE)-D(2)                                                  00004120
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE                       00004130
```

```
    3 C(1)=C(1)/B(1)                                                    00004140
      P(1)=P(1)/B(1)                                                    00004150
      DO 4 I=2,NE                                                       00004160
      MUL=1.0/(B(I)-A(I)*C(I-1))                                        00004170
      C(I)=MUL*C(I)                                                     00004180
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))                                       00004190
C--OBTAIN SPLINE COEFFICIENTS                                           00004200
      A(NE+IT)=P(NE)                                                    00004210
      I=NE-1                                                            00004220
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)                                       00004230
      I=I-1                                                             00004240
       IF(I.GE.1) GO TO 5                                               00004250
      IF(IT.EQ.0) GO TO 6                                               00004260
      A(1)=D(1)                                                         00004270
      A(M)=D(2)                                                         00004280
    6 IF(H.EQ.0.) GO TO 14                                              00004290
      HH=1.0/H                                                          00004300
      DO 7 I=1,N                                                        00004310
      MUL=HH*(Y(I+1)-Y(I))                                              00004320
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))                               00004330
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))                                 00004340
      RETURN                                                            00004350
C--UNEQUAL SPACING H=0.. AND IT=1                                       00004360
   11 DO 12 I=1,N                                                       00004370
   12 S(I+1)=X(I+1)-X(I)                                                00004380
      DO 13 I=1,NE                                                      00004390
      B(I)=2.0*(S(I+1)+S(I+2))                                          00004400
      C(I)=S(I+1)                                                       00004410
      A(I)=S(I+2)                                                       00004420
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/     00004430
     $ (S(I+1)*S(I+2))                                                  00004440
      P(1)=P(1)-S(3)*D(1)                                               00004450
      P(NE)=P(NE)-S(N)*D(2)                                             00004460
      GO TO 3                                                           00004470
   14 DO 15 I=1,N                                                       00004480
      HH=1.0/S(I+1)                                                     00004490
      MUL=(Y(I+1)-Y(I))*HH**2                                           00004500
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH                                 00004510
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2                              00004520
      RETURN                                                            00004530
C--2ND DERIVATIVE BOUNDARIES GIVEN                                      00004540
   20 NE=N+1                                                            00004550
      IF(H) 999,31,21                                                   00004560
C--EQUAL SPACING H .GT. 0 AND IT=0                                      00004570
   21 HH=3.0/H                                                          00004580
      DO 22 I=2,N                                                       00004590
      B(I)=4.0                                                          00004600
      C(I)=1.0                                                          00004610
      A(I)=1.0                                                          00004620
   22 P(I)=HH*(Y(I+1)-Y(I-1))                                           00004630
      B(1)=2.0                                                          00004640
      B(NE)=2.0                                                         00004650
```

```
      C(1)=1.0                                                          00004660
      C(NE)=1.0                                                         00004670
      A(NE)=1.0                                                         00004680
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)                                    00004690
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)                                   00004700
      GO TO 3                                                           00004710
C--UNEQUAL SPACING H=0 AND IT=0                                         00004720
   31 DO 32 I=1,N                                                       00004730
   32 S(I+1)=X(I+1)-X(I)                                                00004740
      N1=N-1                                                            00004750
      DO 33 I=1,N1                                                      00004760
      B(I+1)=2.0*(S(I+1)+S(I+2))                                        00004770
      C(I+1)=S(I+1)                                                     00004780
      A(I+1)=S(I+2)                                                     00004790
   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)))/   00004800
     *      (S(I+1)*S(I+2))                                            00004810
      B(1)=2.0                                                          00004820
      B(NE)=2.0                                                         00004830
      C(1)=1.0                                                          00004840
      C(NE)=1.0                                                         00004850
      A(NE)=1.0                                                         00004860
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)                           00004870
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)                          00004880
      GO TO 3                                                           00004890
  999 M=-IABS(M)                                                        00004900
      RETURN                                                            00004910
      END                                                               00004920

      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                              00004930
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y           00004940
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE       00004950
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).         00004960
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'        00004970
C  WHICH REQUIRES ONLY ONE CALL.                                        00004980
C                                                                       00004990
      DIMENSION X(1),Y(1),A(1),B(1),C(1)                                00005000
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9                              00005010
      M1=M-1                                                            00005020
      DO 1 I=1,M1                                                       00005030
      J=I                                                               00005040
      IF(XX.LE.X(I+1)) GO TO 2                                          00005050
    1 CONTINUE                                                          00005060
    9 WRITE(6,60) XX,X(1),X(M)                                          00005070
   60 FORMAT('0ERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL00005080
     * (',E16.8,',',E16.8,')')                                         00005090
      RETURN                                                            00005100
    2 Z=XX-X(J)                                                         00005110
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z                                  00005120
      RETURN                                                            00005130
      END                                                               00005140
```

```
      REAL FUNCTION RLAGF0(X,FUN,TOL,L,NEW)                              00005150
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE                00005160
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*COS(G*B)*DG' DEFINED AS THE     00005170
C  REAL FOURIER COSINE TRANSFORM WITH ARGUMENT X(=ALOG(B))                00005180
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND                 00005190
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....         00005200
C                                                                         00005210
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                     00005220
C                                                                         00005230
C--PARAMETERS:                                                           00005240
C                                                                         00005250
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00005260
C              'RLAGF0' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,     00005270
C              SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,    00005280
C              THEN SUBPROGRAM 'RFOUR0' IS ADVISED FOR GENERAL USE.      00005290
C              (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).              00005300
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).      00005310
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN    00005320
C                CALLING PROGRAM AND IN SUBPROGRAM FUN.                 00005330
C                THE REAL FUNCTION FUN SHOULD BE A MONOTONE             00005340
C                DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00005350
C       TOL=       REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,    00005360
C                IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00005370
C                THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,       00005380
C                TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON        00005390
C                THE FUNCTION FUN AND PARAMETER X...IN GENERAL,         00005400
C              A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'   00005410
C              BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00005420
C              RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN  00005430
C              APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,    00005440
C              ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...   00005450
C       L=       RESULTING NO. FILTER WTS. USED IN THE VARIABLE         00005460
C                CONVOLUTION (L DEPENDS ON TOL AND FUN).                00005470
C              MIN.L=24 AND MAX.L=281--WHICH COULD                      00005480
C                OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING   00005490
C                VERY FAST...                                           00005500
C     * NEW=   1 IS NECESSARY 1ST TIME OR BRAND NEW X.                  00005510
C              0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20         00005520
C                IS ASSUMED INTERNALLY BY THIS ROUTINE.                 00005530
C                NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL               00005540
C                STILL ASSUME X=(LAST X)-0.20 ANYWAY...                 00005550
C                IT IS THE USERS RESPONSIBILITY TO NORMALIZE            00005560
C                BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00005570
C              THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT       00005580
C                TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A             00005590
C                SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00005600
C                ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...          00005610
C                THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED             00005620
C                KERNELS WILL BE USED IN THE LAGGED CONVOLUTION         00005630
C                WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS     00005640
C                WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)             00005650
C                                                                       00005660
```

```
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF0; THE FOURIER   00005670
C  TRANSFORM IS THEN RLAGF0/B WHICH IS TO BE COMPUTED AFTER EXIT FROM   00005680
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...          00005690
C                                                                       00005700
C--USAGE-- 'RLAGF0' IS CALLED AS FOLLOWS:                               00005710
C     ...                                                               00005720
C     EXTERNAL RF                                                       00005730
C     ...                                                               00005740
C     R=RLAGF0(ALOG(B),RF,TOL,L,NEW)/B                                  00005750
C     ...                                                               00005760
C     END                                                               00005770
C     REAL FUNCTION RF(G)                                               00005780
C     ...USER SUPPLIED CODE...                                          00005790
C     END                                                               00005800
C                                                                       00005810
C--NOTES:                                                               00005820
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM     00005830
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS       00005840
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                              00005850
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION   00005860
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN           00005870
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,   00005880
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'     00005890
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING  00005900
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).     00005910
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                  00005920
C                 ...                                                   00005930
C                 NB=AINT(5.*ALOG(BMAX/BMIN))+1                         00005940
C                 NB1=NB+1                                              00005950
C                 X0=ALOG(BMAX)+.2                                      00005960
C                 NEW=1                                                 00005970
C                 DO 1 J=1,NB                                           00005980
C                 I=NB1-J                                               00005990
C                 X=X0-.2*J                                             00006000
C                 ARG(I)=EXP(X)                                         00006010
C                 ANS(I)=RLAGF0(X,RF,TOL,L,NEW)/ARG(I)                  00006020
C     1           NEW=0                                                 00006030
C                 ...                                                   00006040
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR    00006050
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,  00006060
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)          00006070
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.               00006080
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY          00006090
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW              00006100
C     BMAX,BMIN AND BY SETTING NEW=1....                                00006110
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00006120
C                                                                       00006130
C                                                                       00006140
        DIMENSION KEY(281),SAVE(281)                                    00006150
        DIMENSION YT(281),Y1(76),Y2(76),Y3(76),Y4(53)                   00006160
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1)),       00006170
     1 (YT(229),Y4(1))                                                  00006180
```

```
C--COS-EXTENDED FILTER WEIGHT ARRAYS:                                   00006190
      DATA Y1/                                                          00006200
     1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14,      00006210
     2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14,      00006220
     3 1.7534103E-13, 7.9373498E-15, 2.1235658E-13, 7.9981520E-14,      00006230
     4 2.3815757E-13, 1.9714260E-13, 2.8920132E-13, 3.4161340E-13,      00006240
     5 4.0349917E-13, 5.2203885E-13, 5.9837223E-13, 7.8015306E-13,      00006250
     6 8.8911655E-13, 1.1709731E-12, 1.3165595E-12, 1.7578463E-12,      00006260
     7 1.9538564E-12, 2.6289768E-12, 2.9167697E-12, 3.9044344E-12,      00006270
     8 4.3927341E-12, 5.7526904E-12, 6.6569552E-12, 8.4555678E-12,      00006280
     9 1.0063229E-11, 1.2487964E-11, 1.5134682E-11, 1.8501488E-11,      00006290
     1 2.2720051E-11, 2.7452598E-11, 3.4025443E-11, 4.0875985E-11,      00006300
     2 5.0751668E-11, 6.1094382E-11, 7.5492982E-11, 9.1445759E-11,      00006310
     3 1.1227336E-10, 1.3676464E-10, 1.6720269E-10, 2.0423244E-10,      00006320
     4 2.4932743E-10, 3.0470661E-10, 3.7198526E-10, 4.5449934E-10,      00006330
     5 5.5502537E-10, 6.7793669E-10, 8.2810001E-10, 1.0112626E-09,      00006340
     6 1.2354800E-09, 1.5085255E-09, 1.8432253E-09, 2.2503397E-09,      00006350
     7 2.7499027E-09, 3.3569525E-09, 4.1025670E-09, 5.0077487E-09,      00006360
     8 6.1205950E-09, 7.4703399E-09, 9.1312760E-09, 1.1143911E-08,      00006370
     9 1.3622929E-08, 1.6623917E-08, 2.0324094E-08, 2.4798610E-08,      00006380
     1 3.0321709E-08, 3.6992986E-08, 4.5237482E-08, 5.5183434E-08/      00006390
      DATA Y2/                                                          00006400
     1 6.7491070E-08, 8.2317946E-08, 1.0069271E-07, 1.2279375E-07,      00006410
     2 1.5022907E-07, 1.8316969E-07, 2.2413747E-07, 2.7322865E-07,      00006420
     3 3.3441046E-07, 4.0756197E-07, 4.9894278E-07, 6.0793233E-07,      00006430
     4 7.4443665E-07, 9.0679753E-07, 1.1107379E-06, 1.3525651E-06,      00006440
     5 1.6573073E-06, 2.0174273E-06, 2.4728798E-06, 3.0090445E-06,      00006450
     6 3.6898816E-06, 4.4879625E-06, 5.5059521E-06, 6.6935820E-06,      00006460
     7 8.2160716E-06, 9.9828691E-06, 1.2260527E-05, 1.4888061E-05,      00006470
     8 1.8296530E-05, 2.2202672E-05, 2.7305154E-05, 3.3109672E-05,      00006480
     9 4.0751046E-05, 4.9372484E-05, 6.0820947E-05, 7.3619571E-05,      00006490
     1 9.0780005E-05, 1.0976837E-04, 1.3550409E-04, 1.6365676E-04,      00006500
     2 2.0227521E-04, 2.4398338E-04, 3.0197018E-04, 3.6370760E-04,      00006510
     3 4.5083748E-04, 5.4213338E-04, 6.7315347E-04, 8.0800951E-04,      00006520
     4 1.0051938E-03, 1.2041401E-03, 1.5011708E-03, 1.7942344E-03,      00006530
     5 2.2421056E-03, 2.6730676E-03, 3.3490681E-03, 3.9815050E-03,      00006540
     6 5.0028666E-03, 5.9285668E-03, 7.4730905E-03, 8.8233510E-03,      00006550
     7 1.1160132E-02, 1.3119627E-02, 1.6653199E-02, 1.9472767E-02,      00006560
     8 2.4800811E-02, 2.8793704E-02, 3.6762063E-02, 4.2228780E-02,      00006570
     9 5.3905163E-02, 6.0804660E-02, 7.7081738E-02, 8.3874501E-02,      00006580
     1 1.0377190E-01, 1.0377718E-01, 1.1892208E-01, 9.0437429E-02/      00006590
      DATA Y3/                                                          00006600
     1 7.1685138E-02,-3.9473064E-02,-1.5078720E-01,-4.0489859E-01,      00006610
     2-5.6018995E-01,-6.8050388E-01,-1.5094224E-01, 6.6304064E-01,      00006620
     3 1.3766748E+00,-8.0373222E-01,-1.0869629E+00, 1.2812892E+00,      00006630
     4-5.0341082E-01,-4.4274455E-02, 2.0913102E-01,-1.9999661E-01,      00006640
     5 1.5207664E-01,-1.0920260E-01, 7.8169956E-02,-5.6651561E-02,      00006650
     6 4.1611799E-02,-3.0880012E-02, 2.3072559E-02,-1.7311631E-02,      00006660
     7 1.3021442E-02,-9.8085025E-03, 7.3943529E-03,-5.5769518E-03,      00006670
     8 4.2073164E-03,-3.1745026E-03, 2.3954154E-03,-1.8076122E-03,      00006680
     9 1.3640816E-03,-1.0293934E-03, 7.7682952E-04,-5.8623518E-04,      00006690
     1 4.4240399E-04,-3.3386183E-04, 2.5195025E-04,-1.9013541E-04,      00006700
```

```
     2 1.4348659E-04,-1.0828284E-04, 8.1716174E-05,-6.1667509E-05,      00006710
     3 4.6537684E-05,-3.5119887E-05, 2.6503388E-05,-2.0000904E-05,      00006720
     4 1.5093768E-05,-1.1390572E-05, 8.5959318E-06,-6.4869407E-06,      00006730
     5 4.8953713E-06,-3.6942830E-06, 2.7878625E-06,-2.1038241E-06,      00006740
     6 1.5875917E-06,-1.1980090E-06, 9.0398030E-07,-6.8208296E-07,      00006750
     7 5.1458650E-07,-3.8817581E-07, 2.9272267E-07,-2.2067921E-07,      00006760
     8 1.6623514E-07,-1.2514102E-07, 9.4034535E-08,-7.0556837E-08,      00006770
     9 5.2741581E-08,-3.9298610E-08, 2.9107255E-08,-2.1413893E-08,      00006780
     1 1.5742032E-08,-1.1498608E-08, 8.7561571E-09,-7.2959446E-09/      00006790
      DATA Y4/                                                         00006800
     1 6.8816619E-09,-8.9679825E-09, 1.4258275E-08,-1.9564299E-08,      00006810
     2 2.0235313E-08,-1.4725545E-08, 5.4632820E-09, 3.5995580E-09,      00006820
     3-9.5287133E-09, 1.1460041E-08,-1.0250532E-08, 7.4641748E-09,      00006830
     4-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,      00006840
     5 7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,      00006850
     6 1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,      00006860
     7-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,      00006870
     8-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,      00006880
     9-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,      00006890
     1-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,      00006900
     2-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,      00006910
     3-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,      00006920
     4-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,      00006930
     5-5.6047478E-14/                                                  00006940
C--$$ENDATA                                                            00006950
      IF(NEW) 10,30,10                                                 00006960
10    LAG=-1                                                           00006970
      X0=-X-30.30251236                                                00006980
      DO 20 IR=1,281                                                   00006990
20    KEY(IR)=0                                                        00007000
30    LAG=LAG+1                                                        00007010
      RLAGF0=0.0                                                       00007020
      CMAX=0.0                                                         00007030
      L=0                                                              00007040
      ASSIGN 110 TO M                                                  00007050
      I=149                                                            00007060
      GO TO 200                                                        00007070
110   CMAX=AMAX1(ABS(C),CMAX)                                          00007080
      I=I+1                                                            00007090
      IF(I.LE.170) GO TO 200                                           00007100
      IF(CMAX.EQ.0.0) GO TO 150                                        00007110
      CMAX=TOL*CMAX                                                    00007120
      ASSIGN 120 TO M                                                  00007130
      I=148                                                            00007140
      GO TO 200                                                        00007150
120   IF(ABS(C).LE.CMAX) GO TO 130                                     00007160
      I=I-1                                                            00007170
      IF(I.GT.0) GO TO 200                                             00007180
130   ASSIGN 140 TO M                                                  00007190
      I=171                                                            00007200
      GO TO 200                                                        00007210
140   IF(ABS(C).LE.CMAX) GO TO 190                                     00007220
```

```
      I=I+1                                                          00007230
      IF(I.LE.281) GO TO 200                                         00007240
      GO TO 190                                                      00007250
150   ASSIGN 160 TO M                                                00007260
      I=1                                                            00007270
      GO TO 200                                                      00007280
160   IF(C.EQ.0.0) GO TO 170                                         00007290
      I=I+1                                                          00007300
      IF(I.LE.148) GO TO 200                                         00007310
170   ASSIGN 180 TO M                                                00007320
      I=281                                                          00007330
      GO TO 200                                                      00007340
180   IF(C.EQ.0.0) GO TO 190                                         00007350
      I=I-1                                                          00007360
      IF(I.GE.171) GO TO 200                                         00007370
190   RETURN                                                         00007380
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)           00007390
200   LOOK=I+LAG                                                     00007400
      IQ=LOOK/282                                                    00007410
      IR=MOD(LOOK,282)                                               00007420
      IF(IR.EQ.0) IR=1                                               00007430
      IROLL=IQ*281                                                   00007440
      IF(KEY(IR).LE.IROLL) GO TO 220                                 00007450
210   C=SAVE(IR)*YT(I)                                               00007460
      RLAGF0=RLAGF0+C                                                00007470
      L=L+1                                                          00007480
      GO TO M,(110,120,140,160,180)                                  00007490
220   KEY(IR)=IROLL+IR                                               00007500
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                          00007510
      GO TO 210                                                      00007520
      END                                                            00007530

        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)                        00007540
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE            00007550
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*SIN(G*B)*DG' DEFINED AS THE 00007560
C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))             00007570
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND            00007580
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....    00007590
C                                                                    00007600
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                 00007610
C                                                                    00007620
C--PARAMETERS:                                                       00007630
C                                                                    00007640
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00007650
C                'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E., 00007660
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00007670
C                THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.  00007680
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).          00007690
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).    00007700
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN  00007710
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.               00007720
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE           00007730
```

```
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00007740
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,     00007750
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00007760
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,      00007770
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON       00007780
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,        00007790
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'  00007800
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00007810
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00007820
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,   00007830
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00007840
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE        00007850
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).               00007860
C               MIN.L=20 AND MAX.L=266--WHICH COULD                     00007870
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING  00007880
C                 VERY FAST...                                          00007890
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.                 00007900
C               0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20        00007910
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.                00007920
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL              00007930
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...                00007940
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE            00007950
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00007960
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT      00007970
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A            00007980
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00007990
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...         00008000
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED            00008010
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION        00008020
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS    00008030
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)            00008040
C                                                                       00008050
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER   00008060
C  TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM   00008070
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...          00008080
C                                                                       00008090
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:                               00008100
C     ...                                                               00008110
C     EXTERNAL RF                                                       00008120
C     ...                                                               00008130
C     R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B                                  00008140
C     ...                                                               00008150
C     END                                                               00008160
C     REAL FUNCTION RF(G)                                               00008170
C     ...USER SUPPLIED CODE...                                          00008180
C     END                                                               00008190
C                                                                       00008200
C--NOTES:                                                               00008210
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM     00008220
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS       00008230
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                              00008240
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION   00008250
```

```
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN                00008260
C     THAT THE ACTUAL NUMBFR OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,        00008270
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'          00008280
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING       00008290
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).          00008300
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                       00008310
C               ...                                                          00008320
C               NB=AINT(5.*ALOG(BMAX/BMIN))+1                                00008330
C               NB1=NB+1                                                     00008340
C               X0=ALOG(BMAX)+.2                                             00008350
C               NEW=1                                                        00008360
C               DO 1 J=1,NB                                                  00008370
C               I=NB1-J                                                      00008380
C               X=X0-.2*J                                                    00008390
C               ARG(I)=EXP(X)                                                00008400
C               ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)                         00008410
C     1         NEW=0                                                        00008420
C               ...                                                          00008430
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR         00008440
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,       00008450
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)               00008460
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                    00008470
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY               00008480
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                   00008490
C     BMAX,BMIN AND BY SETTING NEW=1....                                     00008500
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE     00008510
C                                                                            00008520
C                                                                            00008530
        DIMENSION KEY(266),SAVE(266)                                         00008540
        DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38)                        00008550
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),            00008560
     1 (WT(229),W4(1))                                                       00008570
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:                                        00008580
      DATA W1/                                                               00008590
     1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12,           00008600
     2 1.7236636E-12,-1.8227727E-12, 1.9255992E-12,-2.0335514E-12,           00008610
     3 2.1473541E-12,-2.2675549E-12, 2.3946842E-12,-2.5292661E-12,           00008620
     4 2.6718110E-12,-2.8227693E-12, 2.9825171E-12,-3.1514006E-12,           00008630
     5 3.3297565E-12,-3.5179095E-12, 3.7163306E-12,-3.9256378E-12,           00008640
     6 4.1464798E-12,-4.3794552E-12, 4.6252131E-12,-4.8845227E-12,           00008650
     7 5.1582809E-12,-5.4474462E-12, 5.7530277E-12,-6.0760464E-12,           00008660
     8 6.4175083E-12,-6.7783691E-12, 7.1595239E-12,-7.5618782E-12,           00008670
     9 7.9864477E-12,-8.4344110E-12, 8.9072422E-12,-9.4067705E-12,           00008680
     1 9.9349439E-12,-1.0493731E-11, 1.1084900E-11,-1.1709937E-11,           00008690
     2 1.2370354E-11,-1.3067414E-11, 1.3802200E-11,-1.4575980E-11,           00008700
     3 1.5390685E-11,-1.6249313E-11, 1.7155934E-11,-1.8115250E-11,           00008710
     4 1.9131898E-11,-2.0209795E-11, 2.1352159E-11,-2.2561735E-11,           00008720
     5 2.3840976E-11,-2.5192263E-11, 2.6618319E-11,-2.8122547E-11,           00008730
     6 2.9709129E-11,-3.1382870E-11, 3.3149030E-11,-3.5013168E-11,           00008740
     7 3.6981050E-11,-3.9058553E-11, 4.1251694E-11,-4.3566777E-11,           00008750
     8 4.6010537E-11,-4.8590396E-11, 5.1314761E-11,-5.4193353E-11,           00008760
     9 5.7236720E-11,-6.0455911E-11, 6.3861222E-11,-6.7461492E-11,           00008770
```

```
1 7.1265224E-11,-7.5279775E-11, 7.9512249E-11,-8.3971327E-11/      00008780
 DATA W2/                                                          00008790
1 8.8668961E-11,-9.3621900E-11, 9.8851764E-11,-1.0438319E-10,      00008800
2 1.1024087E-10,-1.1644680E-10, 1.2301979E-10,-1.2997646E-10,      00008810
3 1.3733244E-10,-1.4510363E-10, 1.5330772E-10,-1.6196550E-10,      00008820
4 1.7110130E-10,-1.8074257E-10, 1.9091922E-10,-2.0166306E-10,      00008830
5 2.1300756E-10,-2.2498755E-10, 2.3763936E-10,-2.5100098E-10,      00008840
6 2.6511250E-10,-2.8001616E-10, 2.9575691E-10,-3.1238237E-10,      00008850
7 3.2994314E-10,-3.4849209E-10, 3.6808529E-10,-3.8878042E-10,      00008860
8 4.1063982E-10,-4.3372666E-10, 4.5811059E-10,-4.8386049E-10,      00008870
9 5.1105728E-10,-5.3977672E-10, 5.7011632E-10,-6.0215516E-10,      00008880
1 6.3601273E-10,-6.7175964E-10, 7.0955028E-10,-7.4942601E-10,      00008890
2 7.9161025E-10,-8.3606980E-10, 8.8317110E-10,-9.3270330E-10,      00008900
3 9.8533749E-10,-1.0404508E-09, 1.0993731E-09,-1.1605442E-09,      00008910
4 1.2267391E-09,-1.2942905E-09, 1.3691677E-09,-1.4429912E-09,      00008920
5 1.5288164E-09,-1.6077524E-09, 1.7085998E-09,-1.7890471E-09,      00008930
6 1.9129068E-09,-1.9857116E-09, 2.1491608E-09,-2.1926779E-09,      00008940
7 2.4312660E-09,-2.3959044E-09, 2.7872500E-09,-2.5610596E-09,      00008950
8 3.2762318E-09,-2.6082940E-09, 4.0261453E-09,-2.3560563E-09,      00008960
9 5.3176554E-09,-1.3960161E-09, 7.7708747E-09, 1.1853546E-09,      00008970
1 1.2760851E-08, 7.4264707E-09, 2.3342187E-08, 2.1869851E-08/      00008980
 DATA W3/                                                          00008990
1 4.6306744E-08, 5.4631686E-08, 9.6763087E-08, 1.2823337E-07,      00009000
2 2.0832812E-07, 2.9280540E-07, 4.5580888E-07, 6.5992437E-07,      00009010
3 1.0056815E-06, 1.4779183E-06, 2.2284335E-06, 3.2994604E-06,      00009020
4 4.9485823E-06, 7.3545473E-06, 1.1001083E-05, 1.6380539E-05,      00009030
5 2.4469550E-05, 3.6469246E-05, 5.4441527E-05, 8.1176726E-05,      00009040
6 1.2113828E-04, 1.8066494E-04, 2.6954609E-04, 4.0202288E-04,      00009050
7 5.9969995E-04, 8.9437312E-04, 1.3338166E-03, 1.9886697E-03,      00009060
8 2.9643943E-03, 4.4168923E-03, 6.5773518E-03, 9.7855105E-03,      00009070
9 1.4539361E-02, 2.1558670E-02, 3.1871864E-02, 4.6903518E-02,      00009080
1 6.8559512E-02, 9.9170152E-02, 1.4120770E-01, 1.9610835E-01,      00009090
2 2.6192603E-01, 3.2743321E-01, 3.6407406E-01, 3.1257559E-01,      00009100
3 9.0460168E-02,-3.6051039E-01,-8.6324760E-01,-8.1178720E-01,      00009110
4 5.2205241E-01, 1.5449873E+00,-1.1817933E+00,-2.6759896E-01,      00009120
5 8.0869203E-01,-6.2757149E-01, 3.4062630E-01,-1.5885304E-01,      00009130
6 7.0472984E-02,-3.1624462E-02, 1.4894068E-02,-7.4821176E-03,      00009140
7 4.0035936E-03,-2.2543784E-03, 1.3160358E-03,-7.8636604E-04,      00009150
8 4.7658745E-04,-2.9125817E-04, 1.7885105E-04,-1.1012416E-04,      00009160
9 6.7910334E-05,-4.1914054E-05, 2.5881544E-05,-1.5985851E-05,      00009170
1 9.8751880E-06,-6.1008526E-06, 3.7692543E-06,-2.3287953E-06/      00009180
 DATA W4/                                                          00009190
1 1.4388425E-06,-8.8899353E-07, 5.4926991E-07,-3.3937048E-07,      00009200
2 2.0968284E-07,-1.2955437E-07, 8.0046336E-08,-4.9457371E-08,      00009210
3 3.0557711E-08,-1.8880390E-08, 1.1665454E-08,-7.2076428E-09,      00009220
4 4.4533423E-09,-2.7515696E-09, 1.7001092E-09,-1.0504494E-09,      00009230
5 6.4904567E-10,-4.0102999E-10, 2.4778763E-10,-1.5310321E-10,      00009240
6 9.4600354E-11,-5.8453314E-11, 3.6119400E-11,-2.2320056E-11,      00009250
7 1.3793460E-11,-8.5242656E-12, 5.2675102E-12,-3.2543076E-12,      00009260
8 2.0097689E-12,-1.2405412E-12, 7.6530538E-13,-4.7191929E-13,      00009270
9 2.9084993E-13,-1.7923661E-13, 1.1018948E-13,-6.7885902E-14,      00009280
1 4.2025050E-14,-2.1314731E-14/                                    00009290
```

```
C--$$ENDATA                                                          00009300
C                                                                    00009310
      IF(NEW) 10,30,10                                               00009320
10    LAG=-1                                                         00009330
      X0=-X-38.30455704                                              00009340
      DO 20 IR=1,266                                                 00009350
20    KEY(IR)=0                                                      00009360
30    LAG=LAG+1                                                      00009370
      RLAGF1=0.0                                                     00009380
      CMAX=0.0                                                       00009390
      L=0                                                            00009400
      ASSIGN 110 TO M                                                00009410
      I=191                                                          00009420
      GO TO 200                                                      00009430
110   CMAX=AMAX1(ABS(C),CMAX)                                        00009440
      I=I+1                                                          00009450
      IF(I.LE.208) GO TO 200                                         00009460
      IF(CMAX.EQ.0.0) GO TO 150                                      00009470
      CMAX=TOL*CMAX                                                  00009480
      ASSIGN 120 TO M                                                00009490
      I=190                                                          00009500
      GO TO 200                                                      00009510
120   IF(ABS(C).LE.CMAX) GO TO 130                                   00009520
      I=I-1                                                          00009530
      IF(I.GT.0) GO TO 200                                           00009540
130   ASSIGN 140 TO M                                                00009550
      I=209                                                          00009560
      GO TO 200                                                      00009570
140   IF(ABS(C).LE.CMAX) GO TO 190                                   00009580
      I=I+1                                                          00009590
      IF(I.LE.266) GO TO 200                                         00009600
      GO TO 190                                                      00009610
150   ASSIGN 160 TO M                                                00009620
      I=1                                                            00009630
      GO TO 200                                                      00009640
160   IF(C.EQ.0.0) GO TO 170                                         00009650
      I=I+1                                                          00009660
      IF(I.LE.190) GO TO 200                                         00009670
170   ASSIGN 180 TO M                                                00009680
      I=266                                                          00009690
      GO TO 200                                                      00009700
180   IF(C.EQ.0.0) GO TO 190                                         00009710
      I=I-1                                                          00009720
      IF(I.GE.209) GO TO 200                                         00009730
190   RETURN                                                         00009740
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)           00009750
200   LOOK=I+LAG                                                     00009760
      IQ=LOOK/267                                                    00009770
      IR=MOD(LOOK,267)                                               00009780
      IF(IR.EQ.0) IR=1                                               00009790
      IROLL=IQ*266                                                   00009800
      IF(KEY(IR).LE.IROLL) GO TO 220                                 00009810
```

```
210   C=SAVE(IR)*WT(I)                                                  00009820
      RLAGF1=RLAGF1+C                                                   00009830
      L=L+1                                                             00009840
      GO TO M,(110,120,140,160,180)                                     00009850
220   KEY(IR)=IROLL+IR                                                  00009860
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                             00009870
      GO TO 210                                                         00009880
      END                                                               00009890

      SUBROUTINE RECUR1(G,V1,F1)                                        00009900
C--BACKWARD RECURRENCE FOR COMPLEX V1,F1 GIVEN REAL*4 ARGUMENT G AND:   00009910
COMMON/MODEL/ PARAMETERS:                                               00009920
C     K(10) = NORMALIZED CONDUCTIVITY ARRAY (M VALUES,WHERE K(1)=1.0).  00009930
C     D(9)  = LAYER THICKNESS ARRAY (M-1 VALUES) D=2*THICKNESS/DEL.     00009940
C     M     = NUMBER LAYERS (M.GE.1.AND.M.LE.10)                        00009950
C               SPECIAL CASE WHEN M=1 (HOMOGENEOUS--D IGNORED)          00009960
C                                                                       00009970
C--NOTE: G,K,D ARE REAL*4                                               00009980
C                                                                       00009990
C                                                                       00010000
      COMMON/MODEL/K,D,M                                                00010010
      REAL*4 K(10),D(9)                                                 00010020
      COMPLEX C,VM,V1,F1,EVD,ONE                                        00010030
      DATA ONE/(1.0,0.0)/                                               00010040
      F1=ONE                                                            00010050
      G2=G*G                                                            00010060
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00010070
      IF(M.EQ.1) GO TO 2                                                00010080
      J=M-1                                                             00010090
    1 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00010100
      EVD=CEXP(-V1*D(J))                                                00010110
      C=(ONE-EVD)/(ONE+EVD)                                             00010120
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                      00010130
      IF(J.EQ.1) GO TO 3                                                00010140
      J=J-1                                                             00010150
      VM=V1                                                             00010160
      GO TO 1                                                           00010170
    2 V1=VM                                                             00010180
    3 RETURN                                                            00010190
      END                                                               00010200

      REAL FUNCTION ERF(X)                                              00010210
C                                                                       00010220
C        ERF COMPUTES THE ERROR FUNCTION TO SIX                         00010230
C        PLACE ACCURACY.                                                00010240
C                                                                       00010250
      DIMENSION A(5)                                                    00010260
      DATA A/0.254829592,-0.284496736,1.421413741,-1.453152027,         00010270
     1 1.061405429/                                                     00010280
      T=1./(1.+0.3275911*X)                                             00010290
      ERF=T*(A(1)+T*(A(2)+T*(A(3)+T*(A(4)+T*A(5)))))                    00010300
      ERF=1.-ERF*EXP_(-X*X)                                             00010310
```

```
      RETURN                                                            00010320
      END                                                               00010330

      REAL FUNCTION HCLOOP(B2)                                          00010340
C--COSINE-TRANSFORM KERNEL FOR HORIZONTAL COPLANAR LOOP-LOOP            00010350
C  TRANSIENT FAST FORWARD SOLUTION (A=0 GROUND CASE)                    00010360
C                                                                       00010370
      COMPLEX ZHANKS,ONEI,ONE1,TWO2,NINE,NINE9,EIGHTI,CB,CB2,CB3,Z      00010380
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00010390
      COMMON/LOOP/D(10),RR,EPS,B0,BM,M1,IHALF                           00010400
      COMMON/MODEL/RK(10),DD(9),MM                                      00010410
      EXTERNAL FG2                                                      00010420
      DATA ONE1/(1.,1.)/,ONEI/(0.,1.)/,TWO2/(-2.,2.)/,NINE/(9.,0.)/,    00010430
     & NINE9/(9.,9.)/,EIGHTI/(0.,8.)/                                   00010440
      B=SQRT(B2)                                                        00010450
      IF(B.LT.B0) GO TO 2                                               00010460
      IF(B.GT.BM) GO TO 3                                               00010470
      IF(ISPLN.EQ.1) GO TO 4                                            00010480
      CB=B                                                              00010490
      CB2=B2                                                            00010500
      CB3=CB*CB2                                                        00010510
      Z=(0.0,0.0)                                                       00010520
      IF(IHALF.EQ.1) Z=-ONEI*(NINE-(NINE+NINE9*CB+EIGHTI*CB2+TWO2*CB3)* 00010530
     & CEXP(-CB*ONE1))/CB2                                              00010540
      IF(MM.EQ.1) GO TO 1                                               00010550
      DEL=RR/B                                                          00010560
      DO 10 I=1,M1                                                      00010570
10    DD(I)=2.0*D(I)/DEL                                                00010580
      Z=Z+CB3*ZHANKS(0,B,FG2,EPS,LL,1)                                  00010590
1     HCLOOP=REAL(Z)                                                    00010600
      RETURN                                                            00010610
2     IF(IHALF.EQ.0) GO TO 3                                            00010620
      HCLOOP=1.0                                                        00010630
      RETURN                                                            00010640
3     HCLOOP=0.0                                                        00010650
      RETURN                                                            00010660
C--ISPLN=1 (0<NB<12 OPTION)                                             00010670
C  INTERPOLATE PRE-SPLINED FREQ FUNCTION                                00010680
4     CALL SPOINT(NS,XS,YS,AS,BS,CS,B,HCLOOP)                           00010690
      RETURN                                                            00010700
      END                                                               00010710

      REAL FUNCTION HSLOOP(B2)                                          00010720
C--SINE-TRANSFORM KERNEL FOR HORIZONTAL COPLANAR LOOP-LOOP               00010730
C  TRANSIENT STEP RESPONSE FORWARD SOLUTION                             00010740
C--CALLS FUNCTION HCLOOP (IMPULSE RESPONSE KERNEL)                      00010750
C                                                                       00010760
      HSLOOP=HCLOOP(B2)/B2                                              00010770
      RETURN                                                            00010780
      END                                                               00010790
```

```
      SUBROUTINE INTEG1(N,X,Y,Y0)                                       00010800
C        THIS ROUTINE INTEGRATES A FUNCTION"S VALUES (Y                 00010810
C        AS A FUNCTION OF X) FROM 0 TO X BY CALCULATING THE CUBIC       00010820
C        SPLINE COEFFICIENTS AND INTEGRATING THE RESULTING              00010830
C        CUBIC POLYNOMIAL APPROXIMATION.  THE Y VALUES ARE              00010840
C        REPLACED BY THE INTEGRATED VALUES.                             00010850
C        Y0 IS THE VALUE OF Y AT X=0.0 (ASSUMES THAT ALL INPUT          00010860
C        X > 0).                                                        00010870
      DIMENSION X(N),Y(N)                                               00010880
      DIMENSION A(200),B(200),C(200),P(200),S(200),PS(2),X1(200),Y1(200)00010890
      DATA PS/0.0,0.0/                                                  00010900
      DO 1 I=1,N                                                        00010910
      X1(I+1)=X(I)                                                      00010920
    1 Y1(I+1)=Y(I)                                                      00010930
      X1(1)=0.0                                                         00010940
      Y1(1)=Y0                                                          00010950
      N1=N+1                                                            00010960
      CALL SPLIN1(N1,0,X1,Y1,A,B,C,0,PS,P,S)                            00010970
      Y(1)=X(1)*(Y0+X(1)*A(1)/2.+X(1)*X(1)*B(1)/3.+X(1)**3*C(1)/4.)     00010980
      N1=N-1                                                            00010990
      DO 10 I=1,N1                                                      00011000
      Z=X(I+1)-X(I)                                                     00011010
   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.)  00011020
      RETURN                                                            00011030
      END                                                               00011040

      REAL FUNCTION RFLAGS(N,FUN,TOL,T0,TM,T,NEW)                       00011050
C--FOURIER TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION             00011060
C  GIVES FOURIER COSINE OR SINE TRANSFORMS VIA RLAGF0,RLAGF1            00011070
C  REF: ANDERSON,1975,NTIS REPT. PB-242-800,P.76-87.                    00011080
C                                                                       00011090
C      N = 0 FOR COSINE TRANSFORM (VIA RLAGF0)                          00011100
C      N = 1 FOR SINE TRANSFORM (VIA RLAGF1)                            00011110
C    FUN = EXTERNAL REAL KERNEL FUNCTION.                               00011120
C    TOL = TOLERANCE REQUESTED FOR RLAGF0 OR RLAGF1                     00011130
C     T0 = TMIN TO USE (E.G., LET T0=.5*TMIN, TMIN=TRUE)                00011140
C     TM = TMAX TO USE (TM>T0)                                          00011150
C      T = TRANSFORM PARAMETER (T0<=T<=TM) FOR THIS CALL (NEW=1 OR 0)   00011160
C    NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS.     00011170
C    NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY.   00011180
C                                                                       00011190
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),                     00011200
     & D(2),W1(200),W2(200)                                             00011210
      EXTERNAL FUN                                                      00011220
      DATA D/2*0.0/                                                     00011230
      IF(NEW.EQ.0) GO TO 3                                              00011240
      NT=AINT(5.*ALOG(TM/T0))+5                                         00011250
      IF(NT.GT.200)CALL ERRMSG('IN RFLAGS: NT>200   ',4,6,16)           00011260
      NT1=NT+1                                                          00011270
      X0=ALOG(T0)+.2*NT                                                 00011280
      NU=1                                                              00011290
      DO 1 J=1,NT                                                       00011300
```

```
      I=NT1-J                                                          00011310
      X=X0-.2*J                                                        00011320
      EX=EXP(X)                                                        00011330
      ARG(I)=EX                                                        00011340
      IF(N.EQ.0) Y(I)=RLAGF0(X,FUN,TOL,L,NU)/EX                        00011350
      IF(N.NE.0) Y(I)=RLAGF1(X,FUN,TOL,L,NU)/EX                        00011360
1     NU=0                                                             00011370
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)                     00011380
2     IF(NT.LT.0) CALL ERRMSG('IN RFLAGS: NT<0 AFTER SPLIN1  ',6,6,16) 00011390
3     IF(T.LT.T0) CALL ERRMSG('IN RFLAGS: T<T0',3,6,16)                00011400
      IF(T.GT.TM) CALL ERRMSG('IN RFLAGS: T>TM',3,6,16)                00011410
      CALL SPOINT(NT,ARG,Y,AR,BR,CR,T,X)                               00011420
      RFLAGS=X                                                         00011430
      RETURN                                                           00011440
      END                                                              00011450

      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                      00011460
C=====================================================================00011470
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00011480
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                              00011490
C                                                                    00011500
C--REF: ANDERSON, W.L., 1979 (IN PRESS), GEOPHYSICS.                 00011510
C                                                                    00011520
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF     00011530
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF     00011540
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY 00011550
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,     00011560
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.      00011570
C                                                                    00011580
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                 00011590
C                                                                    00011600
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED. 00011610
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00011620
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100011630
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).       00011640
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED) 00011650
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00011660
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION       00011670
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00011680
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00011690
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH   00011700
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G) 00011710
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A 00011720
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED. 00011730
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE     00011740
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00011750
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00011760
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS 00011770
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS 00011780
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00011790
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY 00011800
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR 00011810
```

```
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.    00011820
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER  00011830
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE  00011840
C              USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL       00011850
C              EVALUATION -- SEE NOTE (1) BELOW).                         00011860
C              FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.        00011870
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION   00011880
C              FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).          00011890
C              NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,   00011900
C              NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.                00011910
C     NEW  =1  IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF       00011920
C              FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING       00011930
C              ZHANKS FOR UNRELATED KERNELS.                              00011940
C              NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE  00011950
C              FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING 00011960
C              REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.         00011970
C     NEW  =0  TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED 00011980
C              IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED            00011990
C              DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN   00012000
C              NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK.  00012010
C                                                                         00012020
C     ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE      00012030
C              COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN       00012040
C              THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).    00012050
C                                                                         00012060
C===========================================================================00012070
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                          00012080
C      ...                                                                00012090
C      COMPLEX Z1,Z2,ZHANKS,FSAVE                                         00012100
C      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00012110
C      EXTERNAL ZF1,ZF2                                                   00012120
C      ...                                                                00012130
C      Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                      00012140
C      DO 1 I=1,NSAVE                                                     00012150
C   C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.       00012160
C   C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)    00012170
C    1 CONTINUE                                                          00012180
C      Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                      00012190
C      ...                                                                00012200
C      END                                                                00012210
C      COMPLEX FUNCTION ZF1(G)                                            00012220
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.     00012230
C      END                                                                00012240
C      COMPLEX FUNCTION ZF2(G)                                            00012250
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.     00012260
C      END                                                                00012270
C===========================================================================00012280
C--NOTES                                                                  00012290
C        (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.      00012300
C              THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS   00012310
C              EXP-UNDERFLOW TO 0.0.                                      00012320
C        (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT 00012330
```

```
C             DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00012340
C             TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN    00012350
C             (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE00012360
C             SAVE Y1,ISAVE                                             00012370
C       (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT   00012380
C             IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),   00012390
C             BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.          00012400
C       (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),  00012410
C             HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS         00012420
C             IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).  00012430
C=============================================================================00012440
C                                                                       00012450
      COMPLEX FUN,C,CMAX,FSAVE                                          00012460
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                           00012470
      DOUBLE PRECISION E,ER,Y1,Y                                        00012480
      DIMENSION T(2),TMAX(2)                                            00012490
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),               00012500
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                        00012510
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),   00012520
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),              00012530
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                             00012540
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                               00012550
C-----E=DEXP(.2D0), ER=1.0D0/E                                          00012560
      DATA E/1.2214027581601698 34 D0/,ER/.8187307530779818 59 D0/      00012570
C--J0-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)          00012580
      DATA WA0/                                                         00012590
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,      00012600
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,      00012610
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,      00012620
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,      00012630
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,      00012640
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,      00012650
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,      00012660
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,      00012670
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,      00012680
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,      00012690
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,      00012700
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,      00012710
     *-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,      00012720
     *-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,      00012730
     *-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,      00012740
     *-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,      00012750
     * 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,      00012760
     * 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,      00012770
     * 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/      00012780
      DATA WB0/                                                         00012790
     * 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,      00012800
     * 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,      00012810
     * 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,      00012820
     * 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,      00012830
     * 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,      00012840
     * 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,      00012850
```

```
     * 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,      00012860
     * 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,      00012870
     * 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,      00012880
     * 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,      00012890
     * 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,      00012900
     * 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,      00012910
     * 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,      00012920
     * 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,      00012930
     * 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,      00012940
     *-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00012950
     * 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00012960
     * 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00012970
     * 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00012980
      DATA WC0/                                                        00012990
     * 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00013000
     * 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00013010
     * 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00013020
     * 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00013030
     * 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00013040
     * 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00013050
     * 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00013060
     * 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00013070
     * 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00013080
     * 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00013090
     * 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00013100
     * 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00013110
     * 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00013120
     * 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00013130
     * 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00013140
     * 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00013150
     *-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00013160
     * 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00013170
     * 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00013180
      DATA WD0/                                                        00013190
     * 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00013200
     *-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00013210
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00013220
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00013230
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00013240
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00013250
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00013260
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00013270
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00013280
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00013290
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00013300
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00013310
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00013320
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                     00013330
C--END OF J0 FILTER WEIGHTS                                            00013340
C                                                                      00013350
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)         00013360
      DATA WA1/                                                        00013370
```

```
*-4.2129715E-16,  5.3667031E-15,-7.1183962E-15,  8.9478500E-15,      00013380
*-1.0767891E-14,  1.2362265E-14,-1.3371129E-14,  1.3284178E-14,      00013390
*-1.1714302E-14,  8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00013400
* 6.1279163E-15,-9.1102765E-15,  9.9696405E-15,-9.3649955E-15,      00013410
* 8.6009018E-15,-8.9749846E-15,  1.1153987E-14,-1.4914821E-14,      00013420
* 1.9314024E-14,-2.3172388E-14,  2.5605477E-14,-2.6217555E-14,      00013430
* 2.5057768E-14,-2.2485539E-14,  1.9022752E-14,-1.5198084E-14,      00013440
* 1.1422464E-14,-7.9323958E-15,  4.8421406E-15,-2.1875032E-15,      00013450
*-3.2177842E-17,  1.8637565E-15,-3.3683643E-15,  4.6132219E-15,      00013460
*-5.6209538E-15,  6.4192841E-15,-6.8959928E-15,  6.9895792E-15,      00013470
*-6.5355935E-15,  5.6125163E-15,-4.1453931E-15,  2.6358827E-15,      00013480
*-9.5104370E-16,  1.4600474E-16,  5.6166519E-16,  8.2899246E-17,      00013490
* 5.0032100E-16,  4.3752205E-16,  2.1052293E-15,-9.5451973E-16,      00013500
* 6.4004437E-15,-2.1926177E-15,  1.1651003E-14,  5.8415433E-16,      00013510
* 1.8044664E-14,  1.0755745E-14,  3.0159022E-14,  3.3506138E-14,      00013520
* 5.8709354E-14,  8.1475200E-14,  1.2530006E-13,  1.8519112E-13,      00013530
* 2.7641786E-13,  4.1330823E-13,  6.1506209E-13,  9.1921659E-13,      00013540
* 1.3698462E-12,  2.0447427E-12,  3.0494477E-12,  4.5501001E-12,      00013550
* 6.7870250E-12,  1.0126237E-11,  1.5104976E-11,  2.2536053E-11/      00013560
 DATA WB1/                                                           00013570
* 3.3617368E-11,  5.0153839E-11,  7.4818173E-11,  1.1161804E-10,      00013580
* 1.6651222E-10,  2.4840923E-10,  3.7058109E-10,  5.5284353E-10,      00013590
* 8.2474468E-10,  1.2303750E-09,  1.8355034E-09,  2.7382502E-09,      00013600
* 4.0849867E-09,  6.0940898E-09,  9.0913020E-09,  1.3562651E-08,      00013610
* 2.0233058E-08,  3.0184244E-08,  4.5029477E-08,  6.7176304E-08,      00013620
* 1.0021488E-07,  1.4950371E-07,  2.2303208E-07,  3.3272689E-07,      00013630
* 4.9636623E-07,  7.4049804E-07,  1.1046805E-06,  1.6480103E-06,      00013640
* 2.4585014E-06,  3.6677163E-06,  5.4714550E-06,  8.1626422E-06,      00013650
* 1.2176782E-05,  1.8166179E-05,  2.7099223E-05,  4.0428804E-05,      00013660
* 6.0307294E-05,  8.9971508E-05,  1.3420195E-04,  2.0021123E-04,      00013670
* 2.9860417E-04,  4.4545291E-04,  6.6423156E-04,  9.9073275E-04,      00013680
* 1.4767050E-03,  2.2016806E-03,  3.2788147E-03,  4.8837292E-03,      00013690
* 7.2596811E-03,  1.0788355E-02,  1.5973323E-02,  2.3612041E-02,      00013700
* 3.4655327E-02,  5.0608141E-02,  7.2827752E-02,  1.0337889E-01,      00013710
* 1.4207357E-01,  1.8821315E-01,  2.2996815E-01,  2.5088500E-01,      00013720
* 2.0334626E-01,  6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00013730
*-1.8280529E-01,  4.7014634E-01,  7.2991233E-03,-3.0614594E-01,      00013740
* 2.4781735E-01,-1.1149185E-01,  2.5985386E-02,  1.0850279E-02,      00013750
*-2.2830217E-02,  2.4644647E-02,-2.2895284E-02,  2.0197032E-02/      00013760
 DATA WC1/                                                           00013770
*-1.7488968E-02,  1.5057670E-02,-1.2953923E-02,  1.1153254E-02,      00013780
*-9.6138436E-03,  8.2952090E-03,-7.1628361E-03,  6.1882910E-03,      00013790
*-5.3482055E-03,  4.6232056E-03,-3.9970542E-03,  3.4560118E-03,      00013800
*-2.9883670E-03,  2.5840861E-03,-2.2345428E-03,  1.9323046E-03,      00013810
*-1.6709583E-03,  1.4449655E-03,-1.2495408E-03,  1.0805480E-03,      00013820
*-9.3441130E-04,  8.0803899E-04,-6.9875784E-04,  6.0425624E-04,      00013830
*-5.2253532E-04,  4.5186652E-04,-3.9075515E-04,  3.3790861E-04,      00013840
*-2.9220916E-04,  2.5269019E-04,-2.1851585E-04,  1.8896332E-04,      00013850
*-1.6340753E-04,  1.4130796E-04,-1.2219719E-04,  1.0567099E-04,      00013860
*-9.1379828E-05,  7.9021432E-05,-6.8334412E-05,  5.9092726E-05,      00013870
*-5.1100905E-05,  4.4189914E-05,-3.8213580E-05,  3.3045496E-05,      00013880
*-2.8576356E-05,  2.4711631E-05,-2.1369580E-05,  1.8479514E-05,      00013890
```

```
     *-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00013900
     *-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00013910
     *-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00013920
     *-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00013930
     *-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00013940
     *-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00013950
     *-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00013960
      DATA WD1/                                                        00013970
     *-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00013980
     *-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00013990
     *-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00014000
     *-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00014010
     *-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00014020
     *-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00014030
     *-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00014040
     *-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00014050
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00014060
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00014070
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00014080
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00014090
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00014100
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00014110
C--END OF J1 FILTER WEIGHTS                                            00014120
C                                                                      00014130
      NONE=0                                                           00014140
      IF(NEW.EQ.0) GO TO 100                                           00014150
      NSAVE=0                                                          00014160
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                00014170
      Y1=0.73588526614797944600D0/DBLE(B)                              00014180
  100 ZHANKS=(0.0,0.0)                                                 00014190
      CMAX=(0.0,0.0)                                                   00014200
      NF=0                                                             00014210
      Y=Y1                                                             00014220
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)   00014230
      ASSIGN 110 TO M                                                  00014240
      I=131                                                            00014250
      Y=Y*E                                                            00014260
      GO TO 200                                                        00014270
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                 00014280
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                 00014290
      I=I+1                                                            00014300
      Y=Y*E                                                            00014310
      IF(I.LE.149) GO TO 200                                           00014320
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                     00014330
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))      00014340
      CMAX=TOL*CMAX                                                    00014350
      ASSIGN 120 TO M                                                  00014360
      GO TO 200                                                        00014370
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                         00014380
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130      00014390
      I=I+1                                                            00014400
      Y=Y*E                                                            00014410
```

```
      IF(I.LE.283) GO TO 200                                          00014420
  130 Y=Y1                                                            00014430
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130               00014440
      ASSIGN 140 TO M                                                 00014450
      I=130                                                           00014460
      GO TO 200                                                       00014470
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                         00014480
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.           00014490
     * NONE.EQ.0) GO TO 190                                          00014500
      I=I-1                                                           00014510
      Y=Y*ER                                                          00014520
      IF(I.GT.0) GO TO 200                                            00014530
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.              00014540
  190 ISAVE=1                                                         00014550
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE          00014560
      ZHANKS=ZHANKS/B                                                 00014570
      RETURN                                                          00014580
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY)  00014590
  200 G=SNGL(Y)                                                       00014600
      IF(NEW) 300,210,300                                             00014610
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                    00014620
      ISAVE0=ISAVE                                                    00014630
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                 00014640
      ISAVE=ISAVE+1                                                   00014650
      IF(ISAVE.LE.NSAVE) GO TO 220                                    00014660
      ISAVE=ISAVE0                                                    00014670
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                        00014680
      GO TO 300                                                       00014690
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.                00014700
  240 C=FSAVE(ISAVE)                                                  00014710
      ISAVE=ISAVE+1                                                   00014720
C-----SWITCH ON ORDER N                                               00014730
  250 IF(N) 270,260,270                                               00014740
  260 C=C*WT0(I)                                                      00014750
      GO TO 280                                                       00014760
  270 C=C*WT1(I)                                                      00014770
  280 ZHANKS=ZHANKS+C                                                 00014780
      GO TO M,(110,120,140)                                           00014790
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)          00014800
  300 NSAVE=NSAVE+1                                                   00014810
      C=FUN(G)                                                        00014820
      NF=NF+1                                                         00014830
      FSAVE(NSAVE)=C                                                  00014840
      GSAVE(NSAVE)=G                                                  00014850
      GO TO 250                                                       00014860
      END                                                             00014870

      COMPLEX FUNCTION FG2(G)                                         00014880
C-- F(G)*G*G KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE                 00014890
C   GROUND CASE (A=0). NOTE:  FG2 IS USED IN T0,T1 INTEGRALS          00014900
C   VIA SUBR 'ZHANKS'.                                              00014910
C                                                                     00014920
```

```
      COMPLEX V1,F1,C,ONE,TWO                                          00014930
      DATA ONE,TWO/(1.0,0.0),(2.0,0.0)/                                00014940
      C=CMPLX(G,0.)                                                    00014950
      CALL RECUR1(G,V1,F1)                                             00014960
C//      FG2=(TWO*V1*C*C*C*(F1-ONE))/((C+V1)*(C+V1*F1))                00014970
C  ON MULTICS, REWRITE AS:                                             00014980
      FG2=TWO*V1*(C/(C+V1))*(C/(C+V1*F1))*(F1-ONE)*C                   00014990
      RETURN                                                           00015000
      END                                                              00015010

      COMPLEX FUNCTION FG(G)                                           00015020
C-- F(G)*G KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE                    00015030
C   GROUND CASE (A=0).  FG IS USED IN T2 INTEGRAL VIA SUBR 'ZHANKS'.   00015040
C                                                                      00015050
      COMPLEX FG2                                                      00015060
      FG=FG2(G)/G                                                      00015070
      RETURN                                                           00015080
      END                                                              00015090

      REAL FUNCTION HZWIRE(B2)                                         00015100
C--COSINE-TRANSFORM KERNEL FOR HZ WIRE-LOOP                            00015110
C  TRANSIENT FAST FORWARD SOLUTION (A=0 GROUND CASE)                   00015120
C                                                                      00015130
      COMPLEX ZHANKS,ONEI,TWOI,THREE,THREE3,CB,CB2,Z                   00015140
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN     00015150
      COMMON/WIRE/D(10),RR,TOL,B0,BM,M1,IHALF                          00015160
      COMMON/MODEL/RK(10),DD(9),MM                                     00015170
      EXTERNAL FG                                                      00015180
      DATA ONEI/(0.,1.)/,TWOI/(0.,2.)/,THREE/(3.,0.)/,THREE3/(3.,3.)/  00015190
      B=SQRT(B2)                                                       00015200
      IF(B.LT.B0) GO TO 2                                              00015210
      IF(B.GT.BM) GO TO 3                                              00015220
      IF(ISPLN.EQ.1) GO TO 4                                           00015230
      CB=B                                                             00015240
      CB2=B2                                                           00015250
      Z=(0.0,0.0)                                                      00015260
      IF(IHALF.EQ.1) Z=-ONEI*(THREE-(THREE+THREE3*CB+TWOI*CB2)*        00015270
     & CEXP(CMPLX(-B,-B)))/CB2                                         00015280
      IF(MM.EQ.1) GO TO 1                                              00015290
      DEL=RR/B                                                         00015300
      DO 10 I=1,M1                                                     00015310
10    DD(I)=2.0*D(I)/DEL                                               00015320
      Z=Z-CB2*ZHANKS(1,B,FG,TOL,LL,1)                                  00015330
1     HZWIRE=REAL(Z)                                                   00015340
      RETURN                                                           00015350
2     IF(IHALF.EQ.0) GO TO 3                                           00015360
      HZWIRE=1.0                                                       00015370
      RETURN                                                           00015380
3     HZWIRE=0.0                                                       00015390
      RETURN                                                           00015400
C--ISPLN=1 (0<NB<12 OPTION)                                            00015410
C  INTERPOLATE PRE-SPLINED FREQ FUNCTION                               00015420
```

```
4     CALL SPOINT(NS,XS,YS,AS,BS,CS,B,HZWIRE)                    00015430
      RETURN                                                     00015440
      END                                                        00015450

      REAL FUNCTION HSWIRE(B2)                                   00015460
C--SINE-TRANSFORM KERNEL FOR HORIZONTAL COPLANAR WIRE-LOOP        00015470
C  TRANSIENT STEP RESPONSE FORWARD SOLUTION                       00015480
C--CALLS FUNCTION HZWIRE (IMPULSE RESPONSE KERNEL)               00015490
C                                                                00015500
      HSWIRE=HZWIRE(B2)/B2                                       00015510
      RETURN                                                     00015520
      END                                                        00015530
```

Appendix 2.-- Conversion to other systems

1. All lower-case letters used for parameters and Fortran names in this report should be changed to upper-case letters for most other systems.

2. Any of the following Multics statements and/or calls should be deleted or replaced if converting to another system:

```
character*n          (replace by logical*n or delete)
call open_           (delete)
call close_          (delete)
exp_                 (replace by exp)
dexp_                (replace by dexp)
cexp_                (replace by cexp)
```

3. All Multics exp-underflow messages are suppressed and the result set to 0.0. An equivalent method should be used for other systems.

4. Subprogram ERRMSG should be changed according to the number of characters per word of the target machine (note that 4 char/word uses format A4 on the Honeywell Multics system; however, 5 char/word is assumed in the input parameter array MSG). Similar changes should be made, if necessary, to other character arrays and format statements (e.g., see subprograms TRANS_HCLOOP and TRANS_HZWIRE, arrays TITLE and FMT).

5. Multics names greater than 6-characters (e.g. trans_hcloop, trans_hzwire, etc.) should be renamed to 6 or less characters for most other systems.

6. Some systems may require statement order changes. For example, the statement functions HZ0(U) and HZ1(U) used in subprograms TRANS_HCLOOP and TRANS_HZWIRE appear "before" the DATA statement on Multics. Most systems require the statement functions to be the first executable statements of a subprogram (i.e., placed "after" the DATA statement).

Appendix 3.-- Test problem input/output listing

The following input file (file05) was used to run a test problem for program TRANS_HCLOOP on a Honeywell Multics system. The output listing (file16) follows beginning on the next page.

file05

```
test2x
$parms m=2,y0=1000,t0=.01,nt=5,tm=100,
 sig=.2,2,d=500$
```

The same input file05 was used to run program TRANS_HZWIRE on Multics. The output listing for TRANS_HZWIRE follows the output for TRANS_HCLOOP.

```
t r a n s _ h c l o o p  --  test2x

m= 2          istep=  0        x0= 0.0000e+00 y0= 0.1000e+04
nb= 10        eps= 0.1000e-05 iout=   6       xnorm=0.900e+01  ipch= 0
iouts= 16     t0= 0.1000e-01  nt =    5       tm= 0.1000e+03   istop= 1
b0=0.1000e-01 bm= 0.1000e+03  ihalf=  0

sig =  0.2000e+00  0.2000e+01  0.0000e+00  0.0000e+00  0.0000e+00
       0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00

  d =  0.5000e+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-01  0.12566e-02  0.90011e+01  0.10000e+01  0.90000e+01
 0.15849e-01  0.19916e-02  0.89980e+01  0.99965e+00  0.89968e+01
 0.25119e-01  0.31565e-02  0.90034e+01  0.10003e+01  0.90023e+01
 0.39811e-01  0.50028e-02  0.89840e+01  0.99810e+00  0.89829e+01
 0.63096e-01  0.79288e-02  0.86501e+01  0.96100e+00  0.86490e+01
 0.10000e+00  0.12566e-01  0.65378e+01  0.72633e+00  0.65370e+01
 0.15849e+00  0.19916e-01  0.28415e+01  0.31569e+00  0.28412e+01
 0.25119e+00  0.31565e-01  0.47536e+00  0.52811e-01  0.47530e+00
 0.39811e+00  0.50028e-01 -0.70071e-01 -0.77847e-02 -0.70062e-01
 0.63096e+00  0.79288e-01 -0.68712e-01 -0.76337e-02 -0.68704e-01
 0.10000e+01  0.12566e+00 -0.49168e-01 -0.54625e-02 -0.49162e-01
 0.15849e+01  0.19916e+00 -0.39093e-01 -0.43431e-02 -0.39088e-01
 0.25119e+01  0.31565e+00 -0.28908e-01 -0.32116e-02 -0.28905e-01
 0.39811e+01  0.50028e+00 -0.18966e-01 -0.21071e-02 -0.18964e-01
 0.63096e+01  0.79288e+00 -0.10977e-01 -0.12195e-02 -0.10975e-01
 0.10000e+02  0.12566e+01 -0.56448e-02 -0.62712e-03 -0.56441e-02
 0.15849e+02  0.19916e+01 -0.26225e-02 -0.29135e-03 -0.26222e-02
 0.25119e+02  0.31565e+01 -0.11203e-02 -0.12446e-03 -0.11202e-02
 0.39811e+02  0.50028e+01 -0.44700e-03 -0.49661e-04 -0.44694e-03
 0.63096e+02  0.79288e+01 -0.16876e-03 -0.18749e-04 -0.16874e-03
 0.10000e+03  0.12566e+02 -0.60367e-04 -0.67066e-05 -0.60360e-04
```

```
t r a n s _ h z w i r e  --  test2x

m= 2          istep=  0        x0= 0.0000e+00 y0= 0.1000e+04
nb= 10        eps= 0.1000e-05  iout=   6       xnorm=0.300e+01  ipch= 0
iouts= 16     t0= 0.1000e-01   nt =    5       tm= 0.1000e+03   istop= 1
b0=0.1000e-01 bm= 0.1000e+03   ihalf=  0

sig =  0.2000e+00  0.2000e+01  0.0000e+00  0.0000e+00  0.0000e+00
       0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00  0.0000e+00

  d =  0.5000e+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-01  0.12566e-02  0.29999e+01  0.10000e+01  0.30000e+01
  0.15849e-01  0.19916e-02  0.30001e+01  0.10001e+01  0.30002e+01
  0.25119e-01  0.31565e-02  0.29998e+01  0.99996e+00  0.29999e+01
  0.39811e-01  0.50028e-02  0.29999e+01  0.99999e+00  0.30000e+01
  0.63096e-01  0.79288e-02  0.29767e+01  0.99227e+00  0.29768e+01
  0.10000e+00  0.12566e-01  0.27553e+01  0.91844e+00  0.27553e+01
  0.15849e+00  0.19916e-01  0.20364e+01  0.67880e+00  0.20364e+01
  0.25119e+00  0.31565e-01  0.10907e+01  0.36357e+00  0.10907e+01
  0.39811e+00  0.50028e-01  0.45970e+00  0.15324e+00  0.45971e+00
  0.63096e+00  0.79288e-01  0.19724e+00  0.65750e-01  0.19725e+00
  0.10000e+01  0.12566e+00  0.10078e+00  0.33594e-01  0.10078e+00
  0.15849e+01  0.19916e+00  0.56026e-01  0.18676e-01  0.56028e-01
  0.25119e+01  0.31565e+00  0.30867e-01  0.10289e-01  0.30868e-01
  0.39811e+01  0.50028e+00  0.16178e-01  0.53930e-02  0.16179e-01
  0.63096e+01  0.79288e+00  0.79349e-02  0.26450e-02  0.79351e-02
  0.10000e+02  0.12566e+01  0.36262e-02  0.12088e-02  0.36263e-02
  0.15849e+02  0.19916e+01  0.15515e-02  0.51719e-03  0.15516e-02
  0.25119e+02  0.31565e+01  0.62657e-03  0.20886e-03  0.62659e-03
  0.39811e+02  0.50028e+01  0.24122e-03  0.80410e-04  0.24123e-03
  0.63096e+02  0.79288e+01  0.89345e-04  0.29783e-04  0.89348e-04
  0.10000e+03  0.12566e+02  0.32249e-04  0.10750e-04  0.32250e-04
```