U.S. DEPARTMENT OF THE INTERIOR
U.S. GEOLOGICAL SURVEY

KEY ANALYSIS SYSTEM

Volume 4 - Subroutines

Robert P. Massé

March 29, 1994

Version 1.0

Open-File Report 94-186

National Earthquake Information Center
Golden, Colorado

1994

The KEY ANALYSIS SYSTEM has been developed as part of the project:

United States National Seismograph Network: Development of Advanced Processing Techniques.

This project is funded by the Nuclear Regulatory Commission.

```
      SUBROUTINE SARCOS(ARG)
C
C*********************************************************************
C
C     COMPUTES ARC COSINE OF ARG WITH CHECK ON RANGE
C
C     ARG - ARGUMENT (IN RADIANS) - INPUT AND OUTPUT
C
C     JULY 16, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      IF(ARG)10,30,40
   10 IF(ARG.GT.-0.99998)GO TO 60
      ARG=PI
      GO TO 9000
C
   30 ARG=PI*0.5
      GO TO 9000
C
   40 IF(ARG.LT.0.99998)GO TO 60
      ARG=0.0
      GO TO 9000
C
   60 ARG=ACOS(ARG)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SCALTS(X,ISAMP,SCALE)
C
C*****************************************************************************
C
C     SCALE A TIME SERIES TO A SPECIFIED MAXIMUM VALUE
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES TO SCALE - INPUT
C     SCALE - DESIRED MAXIMUM VALUE FOR SCALED TIME SERIES - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 25, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION X(1)
C
      CALL SFS(X,ISAMP,SFX)
C
      IF(SFX.EQ.0.0)GO TO 9000
      ARG=SCALE/SFX
C
      DO 500 J=1,ISAMP
      X(J)=X(J)*ARG
  500 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE SDISP(LVRY)
C
C*********************************************************************
C
C     COMPUTE RAYLEIGH OR LOVE WAVE PHASE AND GROUP VELOCITY DISPERSION
C
C     LVRY - RAYLEIGH/LOVE WAVE FLAG - INPUT
C            = 0 RAYLEIGH WAVE
C            = 1 LOVE WAVE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 19, 1994  COLORADO
C
C*********************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
C
      DIMENSION V(1000,4),T(300),C(300),U(300),AMED(300),ENGK(300),
     * ENGP(300),HVRAT(300)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(30001),C(1)),(BX(31001),T(1))
      EQUIVALENCE (BX(32001),U(1)),(BX(33001),AMED(1))
      EQUIVALENCE (BX(34001),ENGK(1)),(BX(35001),ENGP(1))
      EQUIVALENCE (BX(36001),HVRAT(1)),(BX(40001),V(1,1))
C
      CHARACTER*10 MODNAM
      CHARACTER*40 FILENM
C
      DATA NY/'y'/
C
C     READ MODEL PARAMETERS
C
   20 WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','FILE NAME FOR MODEL',/)
      READ(IRE,70,END=9000)FILENM
   70 FORMAT(A40)
C
C     FILENM - FILE NAME FOR INPUT MODEL
C
      WRITE(IWR2,100)FILENM
  100 FORMAT(/,10X,'SELECTED FILE NAME IS ',A40,
     * //,'++ INPUT: ','IS IT CORRECT (y/n)?',/)
      READ(IRE,150,END=9000)KY
  150 FORMAT(A1)
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 20
C
      OPEN(LUN3,FILE=FILENM,STATUS='old')
C
      REWIND LUN3
C
      READ(LUN3,160,END=9000)MODNAM
```

```
  160 FORMAT(A10)
C
C     MODNAM - MODEL TITLE
C
      M=0
C
      DO 170 I=1,1000
      M=M+1
      READ(LUN3,*,END=9000)(V(M,N),N=1,4)
C
C     V(M,1) - LAYER THICKNESS (IN KILOMETERS)
C     V(M,2) - LAYER P VELOCITY (IN KM/S)
C              (NOT USED FOR LOVE WAVE COMPUTATIONS)
C     V(M,3) - LAYER S VELOCITY (IN KM/S)
C     V(M,4) - LAYER DENSITY
C
      IF(V(M,4).LE.0.0)GO TO 200
  170 CONTINUE
C
  200 MMAX=M-1
      IF(MMAX.LE.1)GO TO 7000
C
C     READ DISPERSION PARAMETERS
C
  300 WRITE(IWR2,350)
  350 FORMAT(/,'++ INPUT: ','MODENO',/)
      READ(IRE,*,END=9000)MODENO
C
C     MODENO - MODE FLAG
C              = 0 FUNDAMENTAL MODE
C              = 1 FIRST HIGHER MODE
C              = 2 SECOND HIGHER MODE
C              = n N'TH HIGHER MODE
C
      IF(MODENO.LT.0)MODENO=0
      WRITE(IWR,400)MODENO
      WRITE(IWR2,400)MODENO
  400 FORMAT(/,10X,'MODENO = ',I3,/)
C
      WRITE(IWR,420)
      WRITE(IWR2,420)
  420 FORMAT(/,10X,'DISPERSION PARAMETERS',/)
C
      WRITE(IWR2,450)
  450 FORMAT(/,'++ INPUT: ','TFIR',3X,'TLAS',3X,'DT',3X,'ITOP',/)
      READ(IRE,*,END=9000)TFIR,TLAS,DT,ITOP
C
C     TFIR - FIRST PERIOD VALUE (IN SECONDS)
C            (OR FIRST FREQUENCY VALUE)
C     TLAS - LAST PERIOD VALUE (IN SECONDS)
C            (OR LAST FREQUENCY VALUE)
C     DT - INCREMENT IN PERIOD VALUES (IN SECONDS)
C          (OR INCREMENT IN FREQUENCY VALUES)
C          = 0 SET TO 1.0
C     ITOP - PERIOD FLAG
C            = 0 INCREMENT PERIOD BY +DT
C            = 1 INCREMENT PERIOD BY *DT
C            = 2 INCREMENT FREQUENCY BY +DT AND FORM PERIOD=1/FREQUENCY
```

```
C            = 3 READ PERIOD VALUES FROM KEYBOARD
C
      IF(TFIR.LE.0.0)GO TO 7000
      IF(DT.EQ.0.0)DT=1.0
      IF(TLAS.LT.TFIR.AND.DT.GT.0.0)GO TO 7000
      WRITE(IWR,500)TFIR,TLAS,DT,ITOP
      WRITE(IWR2,500)TFIR,TLAS,DT,ITOP
  500 FORMAT(/,10X,'TFIR = ',F8.2,5X,'TLAS = ',F8.2,5X,'DT = ',F8.2,/,
     * 10X,'ITOP = ',I5,/)
C
      KMAX=(TLAS-TFIR)/DT+1.5
      IF(KMAX.GT.300)KMAX=300
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','CFIR',3X,'DC',/)
      READ(IRE,*,END=9000)CFIR,DC
C
C     CFIR - STARTING PHASE VELOCITY FOR SEARCH (IN KM/S)
C            = 0 SET TO 2.0
C     DC - INCREMENT IN PHASE VELOCITY FOR INITIAL SEARCH (IN KM/S)
C          = 0 SET TO 0.01
C
      IF(CFIR.LE.0.0)CFIR=2.0
      IF(DC.LE.0.0)DC=0.01
      WRITE(IWR,700)CFIR,DC
      WRITE(IWR2,700)CFIR,DC
  700 FORMAT(/,10X,'CFIR = ',F8.2,5X,'DC = ',F8.2,/)
C
      WRITE(IWR2,800)
  800 FORMAT(/,'++ INPUT: ','MODPR',3X,'KCDS',3X,'KWRITE',/)
      READ(IRE,*,END=9000)MODPR,KCDS,KWRITE
C
C     MODPR - MODEL LISTING FLAG
C             = 0 NO LISTING MADE
C             = 1 LISTING MADE
C     KCDS - ROOT ACCURACY FLAG
C            = 0 ACCURACY = 1E-7
C            = 1 ACCURACY = 1E-6
C            = 2 ACCURACY = 1E-8
C            = 3 ACCURACY = 1E-9
C            = 4 ACCURACY = 1E-5
C            = 5 ACCURACY = 1E-10
C            = 6 ACCURACY = 1E-4
C     KWRITE - DEBUG LISTING FLAG
C              = 0 NO LISTING MADE
C              = 1 LISTING MADE OF ROOT SEARCH
C              = 2 LISTING MADE OF ROOT SEARCH AND LAYER DIVISION
C
C     FIND LARGEST VALUE OF SHEAR VELOCITY
C
      BTAMX=ABS(V(1,3))
C
      DO 1000 M=2,MMAX
      IF(BTAMX.GE.ABS(V(M,3)))GO TO 1000
      BTAMX=ABS(V(M,3))
 1000 CONTINUE
C
C     SET UP ACCURACY CRITERIA
```

```
C
      IF(KCDS.LE.0)ACC=1E-7
      IF(KCDS.EQ.1)ACC=1E-6
      IF(KCDS.EQ.2)ACC=1E-8
      IF(KCDS.EQ.3)ACC=1E-9
      IF(KCDS.EQ.4)ACC=1E-5
      IF(KCDS.EQ.5)ACC=1E-10
      IF(KCDS.GT.6)ACC=1E-4
C
C     COMPUTE ALL PERIODS FOR DISPERSION CURVE
C
      T(1)=TFIR
      IF(ITOP.GT.0)GO TO 1200
C
      DO 1100 K=2,KMAX
      T(K)=T(K-1)+DT
 1100 CONTINUE
C
      GO TO 2000
C
 1200 IF(ITOP.GT.1)GO TO 1400
C
      DO 1300 K=2,KMAX
      T(K)=T(K-1)*DT
 1300 CONTINUE
C
      GO TO 2000
C
 1400 IF(ITOP.GT.2)GO TO 1700
C
      DO 1500 K=2,KMAX
      T(K)=T(K-1)+DT
 1500 CONTINUE
C
      DO 1600 K=1,KMAX
      T(K)=1.0/T(K)
 1600 CONTINUE
C
      GO TO 2000
C
 1700 WRITE(IWR2,1800)
 1800 FORMAT(/,'++ INPUT: ','T',/)
      READ(IRE,*,END=9000)(T(K),K=1,KMAX)
C
C     DETERMINE ALL ROOTS
C
 2000 CALL SROOT(LVRY,MODENO,KWRITE,CFIR,DC,ACC,MMAX,BTAMX,
     * KLAST,KMAX)
C
      IF(KLAST.GT.0)GO TO 2700
C
      WRITE(IWR,2200)
      WRITE(IWR2,2200)
 2200 FORMAT(/,10X,'NO ROOTS FOUND',/)
      GO TO 300
C
 2700 IF(LVRY.GT.0)GO TO 4000
C
```

```
C     COMPUTE GROUP VELOCITY FOR RAYLEIGH WAVES
C
      CALL GPVEL(KLAST,T,C,U,DT)
C
C     LISTING OF MODEL DESIRED?
C
 4000 IF(MODPR.GT.0)GO TO 5000
C
      WRITE(IWR,4200)MODNAM
      WRITE(IWR2,4200)MODNAM
 4200 FORMAT(//,10X,'MODEL PARAMETERS - ',A10,/)
C
      WRITE(IWR,4300)
      WRITE(IWR2,4300)
 4300 FORMAT(/,10X,'LAYER NO',4X,'DEPTH',4X,'THICKNESS',5X,
     * 'P VEL',5X,'S VEL',3X,'DENSITY',/)
C
      DEEP=0.0
C
      DO 4500 M=1,MMAX
      DEEP=DEEP+V(M,1)
      WRITE(IWR,4400)M,DEEP,(V(M,N),N=1,4)
      WRITE(IWR2,4400)M,DEEP,(V(M,N),N=1,4)
 4400 FORMAT(12X,I3,6X,F7.3,3X,F7.3,5X,F7.3,3X,F7.3,3X,F6.3)
 4500 CONTINUE
C
 5000 IF(LVRY.GT.0)GO TO 5500
      WRITE(IWR,5100)
      WRITE(IWR2,5100)
 5100 FORMAT(//,10X,'RAYLEIGH WAVE DISPERSION FOR FLAT EARTH MODEL',/)
      GO TO 5800
C
 5500 WRITE(IWR,5600)
      WRITE(IWR2,5600)
 5600 FORMAT(//,10X,'LOVE WAVE DISPERSION FOR FLAT EARTH MODEL',/)
C
 5800 IF(MODENO.GT.0)GO TO 6000
      WRITE(IWR,5900)
      WRITE(IWR2,5900)
 5900 FORMAT(/,10X,'FUNDAMENTAL MODE',/)
      GO TO 6200
C
 6000 WRITE(IWR,6100)MODENO
      WRITE(IWR2,6100)MODENO
 6100 FORMAT(/,10X,'HIGHER MODE ',I4,/)
C
 6200 IF(LVRY.GT.0)GO TO 6500
C
      WRITE(IWR,6300)
      WRITE(IWR2,6300)
 6300 FORMAT(/,11X,'PERIOD',6X,'PHASE VELOCITY',4X,'GROUP VELOCITY',
     * 7X,'U0/W0',/)
C
      WRITE(IWR,6400)(T(K),C(K),U(K),HVRAT(K),K=1,KLAST)
      WRITE(IWR2,6400)(T(K),C(K),U(K),HVRAT(K),K=1,KLAST)
 6400 FORMAT(10X,F7.3,8X,F9.5,9X,F9.5,10X,F7.5)
      GO TO 6900
C
```

```
 6500 WRITE(IWR,6600)
      WRITE(IWR2,6600)
 6600 FORMAT(/,11X,'PERIOD',4X,'C VEL',4X,'U VEL',3X,'KIN ENER',4X,
     * 'POT ENER',4X,'MED RESP',/)
      WRITE(IWR,6700)(T(K),C(K),U(K),ENGK(K),ENGP(K),AMED(K),K=1,KLAST)
      WRITE(IWR2,6700)(T(K),C(K),U(K),ENGK(K),ENGP(K),AMED(K),K=1,KLAST)
 6700 FORMAT(10X,F7.3,1X,F9.5,1X,F8.5,1X,E11.5,1X,E11.5,1X,E11.5)
C
 6900 CLOSE(LUN3)
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'ERROR IN INPUT PARAMETERS',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SEIPLT
C
C*********************************************************************
C
C     PLOT TIME SERIES EXECUTIVE ROUTINE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 30, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/PCKF/ICHECK,IPLFL
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*80 MSG
      CHARACTER*20 AXTIT,AYTIT,AMTIT,IBLK,AX2
C
      DATA IBLK/'                    '/,AX2/'     Time (sec)     '/
C
C     SET UP PLOT PARAMETERS
C
      CMPIN=2.54
      MODE=0
      TF=-20.0
      FSDIST=0.0
      DMAX=9.0
      ZLT=DMAX+1.50
      SCALD=1.0
      DI=1.0
      DDI=2.0
      IPF1=0
      IPF2=1
      FAC=1.0
      TZERO=0.0
      TOFSET=20.0
      AXTIT=IBLK
      AYTIT=IBLK
      AMTIT=IBLK
C
      WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','IAUTO',/)
      READ(IRE,*,END=9000)IAUTO
C
C     IAUTO - PLOT DEFAULT OPTION FLAG
C             = 0 READ IN PLOT PARAMETERS
C             = 1 PLOT ALL CHANNELS ON PRESET AXES
```

```
C
      WRITE(IWR,60)IAUTO
      WRITE(IWR2,60)IAUTO
   60 FORMAT(/,10X,'IAUTO = ',I2,/)
C
      IF(IAUTO.NE.0)GO TO 120
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','NCH',3X,'ICHAN1',/)
      READ(IRE,*,END=9000)NCH,ICHAN1
C
C     NCH - TOTAL NUMBER OF CHANNELS TO PLOT
C           = 0 SET TO NACHLS
C     ICHAN1 - FIRST CHANNEL TO PLOT
C              = 0 SET TO 1
C
  120 IF(IAUTO.NE.0)NCH=NACHLS
      IF(IAUTO.NE.0)ICHAN1=1
      IF(ICHAN1.LE.0)ICHAN1=1
      IF(NCH.LE.0)NCH=NACHLS
      NCH2=NCH+ICHAN1-1
      IF(NCH2.GT.NACHLS)NCH2=NACHLS
      WRITE(IWR,130)NCH,ICHAN1
      WRITE(IWR2,130)NCH,ICHAN1
  130 FORMAT(/,10X,'NCH = ',I4,5X,'ICHAN1 = ',I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(NCH2.LT.ICHAN1)GO TO 7000
C
      IF(IAUTO.NE.0)GO TO 170
C
      WRITE(IWR2,150)
  150 FORMAT(/,'++ INPUT: ','IFL',3X,'IHEAD',/)
      READ(IRE,*,END=9000)IFL,IHEAD
C
C     IFL - PLOT FLAG
C           = 0 PLOT ALL DATA IN NCH CHANNELS ON PRESET AXES
C           = 1 INPUT PLOT PARAMETERS FOR AXIS AND DATA
C     IHEAD - HEADER FILE
C             = 0 DON'T READ HEADER
C             = 1 READ HEADER
C
  170 IF(IAUTO.NE.0)IFL=0
      IF(IAUTO.NE.0)IHEAD=0
      WRITE(IWR,200)IFL,IHEAD
      WRITE(IWR2,200)IFL,IHEAD
  200 FORMAT(/,10X,'IFL = ',I2,5X,'IHEAD = ',I2,/)
C
      IF(IFL.NE.0)GO TO 1000
C
C     USE PRESET PARAMETERS AND PLOT ALL DATA
C
      AXTIT=AX2
C
C     FIND LONGEST TRACE
C
      TMX1=0.0
C
```

```
      DO 500 I=ICHAN1,NCH2
      TMX2=JSAMP(I)/RSAMR(I)
      IF(TMX2.LE.TMX1)GO TO 500
      TMX1=TMX2
  500 CONTINUE
C
      ITMX=TMX1/10.0
      TMAX=ITMX*10+10
      TDIZ=((TMAX+TOFSET)/9.5)/CMPIN
      ITDIZ=TDIZ/5.0
      TDIZ=ITDIZ*5+5
      SCALT=1.0/(TDIZ*CMPIN)
      IF(TMAX.LE.100.0)TI=10.0
      IF(TMAX.GT.100.0)TI=20.0
      TTI=TI*4.0
      GO TO 2000
C
C     READ TIME AXIS PARAMETERS
C
 1000 WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','TI',3X,'TTI',3X,'TMAX',3X,'TDIZ',/)
      READ(IRE,*,END=9000)TI,TTI,TMAX,TDIZ
C
C     TI - INCREMENT IN TIME VALUE FOR TICKS (IN SECONDS)
C          = 0 SET TO 20.0
C     TTI - INCREMENT IN TIME VALUE FOR LABELED TICKS (IN SECONDS)
C           (MUST BE INTEGER MULTIPLE OF TI)
C           = 0 SET TO 80
C     TMAX - MAXIMUM RELATIVE TIME VALUE TO BE PLOTTED (IN SECONDS)
C            = 0 SET TO 200.0
C     TDIZ - TRACE PLOTTING INCREMENT (IN SECONDS/CM)
C            = 0 SET TO 10.0
C
      IF(TI.LE.0.0)TI=20.0
      IF(TTI.EQ.0.0)TTI=80.0
      IF(TMAX.LE.0.0)TMAX=200.0
      IF(TDIZ.LE.0.0)TDIZ=10.0
      SCALT=1.0/(TDIZ*CMPIN)
C
C     READ DISTANCE AXIS PARAMETERS
C
      WRITE(IWR2,1500)
 1500 FORMAT(/,'++ INPUT: ','ZLT',/)
      READ(IRE,*,END=9000)ZLT
C
C     ZLT - TOTAL PLOT DISTANCE DIMENSION (IN INCHES)
C           = 0 SET TO 9.5
C
      IF(ZLT.LE.0.0)ZLT=9.5
C
C     READ PLOT PARAMETERS
C
 2000 IF(IAUTO.NE.0)GO TO 2200
C
      WRITE(IWR2,2100)
 2100 FORMAT(/,'++ INPUT: ','MODEP',3X,'FAC',/)
      READ(IRE,*,END=9000)MODEP,FAC
C
```

```
C     MODEP - MODE FOR OPENING PLOT SYSTEM
C             = 0 SET TO 3
C             = 1 IMMEDIATE MODE
C             = 2 DEFERRED MODE
C             = 3 EDIT MODE
C     FAC - PLOT SCALE FACTOR
C           = 0 SET TO 1.0
C
 2200 IF(IAUTO.NE.0)MODEP=3
      IF(IAUTO.NE.0)FAC=1.0
      IF(MODEP.LE.0)MODEP=3
      IF(FAC.LE.0.0)FAC=1.0
      IF(IHEAD.LE.0)GO TO 3000
C
      WRITE(IWR2,2500)
 2500 FORMAT(/,'++ INPUT: ','AXTIT',/)
      READ(IRE,2600,END=9000)AXTIT
 2600 FORMAT(A20)
C
      WRITE(IWR2,2700)
 2700 FORMAT(/,'++ INPUT: ','AYTIT',/)
      READ(IRE,2600,END=9000)AYTIT
C
      WRITE(IWR2,2800)
 2800 FORMAT(/,'++ INPUT: ','AMTIT',/)
      READ(IRE,2600,END=9000)AMTIT
C
C     AXTIT - TIME AXIS TITLE (20 CHARACTERS MAXIMUM))
C     AYTIT - DISTANCE AXIS TITLE (20 CHARACTERS MAXIMUM)
C     AMTIT - MAIN PLOT TITLE (20 CHARACTERS MAXIMUM)
C
 3000 WRITE(IWR,3300)
      WRITE(IWR2,3300)
 3300 FORMAT(/,10X,'PLOT PARAMETERS',/)
      WRITE(IWR,3500)MODEP,FAC
      WRITE(IWR2,3500)MODEP,FAC
 3500 FORMAT(/,10X,'MODEP = ',I2,5X,'FAC = ',F7.3,/)
      WRITE(IWR,3600)TI,TTI,TMAX,TDIZ
      WRITE(IWR2,3600)TI,TTI,TMAX,TDIZ
 3600 FORMAT(/,10X,'TI = ',F8.2,5X,'TTI = ',F7.2,5X,'TMAX = ',F7.2,/,
     * 10X,'TDIZ = ',F6.2,/)
      WRITE(IWR,3700)ZLT
      WRITE(IWR2,3700)ZLT
 3700 FORMAT(/,10X,'ZLT = ',F8.2,/)
C
      PFAC=DMAX/(NCH*2)
      IF(IPLFL.NE.0)GO TO 4000
      WIN=ZLT+2.0
C
      CALL OPNPLT(MODEP,WIN,111)
C
C     SCALE AND PLOT EACH TRACE
C
 4000 DO 6000 I=ICHAN1,NCH2
      ISAMP=JSAMP(I)
      SAMR=RSAMR(I)
      INDEX=(I-1)*LENG+1
C
```

```
C     DETERMINE MAXIMUM O-P AMPLITUDE
C
      CALL SFS(DATA(INDEX),ISAMP,SFX)
C
      WRITE(IWR2,4100)I,SFX
 4100 FORMAT(/,10X,'MAXIMUM AMPLITUDE FOR CHANNEL ',I4,' IS 'F12.3,/)
C
C     SCALE DATA
C
      IF(SFX.LT.0.000001)SFX=1.0
      PF=-PFAC/SFX
C
      DO 4500 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)*PF
 4500 CONTINUE
C
C     POSITION DATA TRACE
C
      DF=PFAC+(I-ICHAN1)*2*PFAC
C
      DO 4600 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)+DF
 4600 CONTINUE
C
      IF(I.NE.ICHAN1)GO TO 4700
C
C     PLOT AXES
C
      CALL RAXEPL(TF,TMAX,SCALT,TI,TTI,FSDIST,DMAX,SCALD,DI,DDI,
     * IPF1,IPF2,FAC,AXTIT,AYTIT,AMTIT)
C
      IPLFL=1
 4700 NP=ISAMP
C
C     CHECK FOR TOO MANY SAMPLES
C
      TM=FLOAT(ISAMP)/SAMR
      TDIF=TM-TMAX
      IF(TDIF.LE.0.0)GO TO 4800
      NP=NP-TDIF*SAMR
C
C     PLOT DATA TRACE
C
 4800 CALL SPLDAT(TF,TZERO,NP,SCALT,FSDIST,DMAX,SCALD,SAMR,
     * DATA(INDEX),FAC)
C
C     REMOVE POSITIONING TERM
C
      DO 5000 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)-DF
 5000 CONTINUE
C
C     REMOVE SCALING FACTOR
C
      DO 5100 J=1,ISAMP
```

```
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)/PF
 5100 CONTINUE
C
 6000 CONTINUE
C
C     RETURN CONTROL TO SCREEN
C
      CALL HLDPLT(5.0,1.0,MODE,MSG)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY PLOTTED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SETHD
C
C*********************************************************************
C
C     READ HEADER INFORMATION FROM THE KEYBOARD
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 24, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL TO CHANGE HEADER
C             = 0 SET TO 1
C
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','JCODE',/)
      READ(IRE,250,END=9000)KCODE
  250 FORMAT(A4)
C
C     JCODE - STATION CODE
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','JCHN',/)
      READ(IRE,500,END=9000)KC1,KC2,KC3,KC4
  500 FORMAT(4A1)
C
C     JCHN1 - STATION CHANNEL IDENTIFICATION
C     JCHN2 - STATION CHANNEL IDENTIFICATION
C     JCHN3 - STATION CHANNEL IDENTIFICATION
C     JCHN4 - STATION CHANNEL IDENTIFICATION
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','RLAT',3X,'RLON',3X,'RELEV',/)
      READ(IRE,*,END=9000)RKLAT,RKLON,RKELEV
C
C     RLAT - STATION GEOGRAPHIC LATITUDE (IN DEGREES)
C            - = S
C            = 0 USE OLD LATITUDE
C     RLON - STATION GEOGRAPHIC LONGITUDE (IN DEGREES)
```

```
C             - = W
C             = 0 USE OLD LONGITUDE
C     RELEV - STATION ELEVATION (IN KILOMETERS)
C
      WRITE(IWR2,700)
  700 FORMAT(/,'++ INPUT: ','RSAMR',/)
      READ(IRE,*,END=9000)RKSAMR
C
C     RSAMR - DATA SAMPLING RATE (IN SAMPLES/S)
C             = 0 USE OLD SAMPLING RATE
C
      IF(ICHAN.LE.0)ICHAN=1
      IF(RKLAT.EQ.0.0)RKLAT=RLAT(ICHAN)
      IF(RKLON.EQ.0.0)RKLON=RLON(ICHAN)
      IF(RKELEV.EQ.0.0)RKELEV=RELEV(ICHAN)
      IF(RKSAMR.EQ.0.0)RKSAMR=RSAMR(ICHAN)
      WRITE(IWR,800)
      WRITE(IWR2,800)
  800 FORMAT(/,10X,'NEW HEADER INFORMATION PARAMETERS',/)
      WRITE(IWR,900)ICHAN
      WRITE(IWR2,900)ICHAN
  900 FORMAT(/,10X,'ICHAN =',I4,/)
      WRITE(IWR,1000)KCODE,KC1,KC2,KC3,KC4,RKLAT,RKLON,RKELEV,RKSAMR
      WRITE(IWR2,1000)KCODE,KC1,KC2,KC3,KC4,RKLAT,RKLON,RKELEV,RKSAMR
 1000 FORMAT(/,10X,'JCODE = ',A4,3X,'JCHN = ',4A1,3X,'RLAT = ',F7.3,
     * 3X,'RLON = ',F8.3,//,10X,'RELEV = ',F7.3,5X,'RSAMR = ',F7.3,/)
C
      JCODE(ICHAN)=KCODE
      JCHN(1,ICHAN)=KC1
      JCHN(2,ICHAN)=KC2
      JCHN(3,ICHAN)=KC3
      JCHN(4,ICHAN)=KC4
      RLAT(ICHAN)=RKLAT
      RLON(ICHAN)=RKLON
      RELEV(ICHAN)=RKELEV
      RSAMR(ICHAN)=RKSAMR
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'HEADER INFORMATION NOT CHANGED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'HEADER INFORMATION SUCCESSFULLY CHANGED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SETRES
C
C*****************************************************************
C
C     SET UP COMPLEX RESPONSE FUNCTION IN MASTER FILE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 30, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/RESPON/SA0,KNP,SPOLES,KNZ,SZEROS,KFLAG
C
      DIMENSION SPOLES(30),SZEROS(20)
      DIMENSION XPOLES(30),XZEROS(20)
      DIMENSION EP(60),EZ(40)
C
      EQUIVALENCE (XPOLES(1),EP),(XZEROS(1),EZ)
C
      COMPLEX SPOLES,SZEROS
      COMPLEX XPOLES,XZEROS
C
      DATA NSTOP/'stop'/,NZERO/'ZERO'/
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','IRP',/)
      READ(IRE,*,END=9000)IRP
C
C     IRP - SELECT MASTER RESPONSE FLAG
C           = 0 READ RESPONSE POLES AND ZEROS
C           = 1 SET UP RESPONSE FROM STORED INFORMATION
C
      IF(IRP.LE.0)GO TO 5000
C
C     READ DESIRED STATION-COMPONENT
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','ICODE',/)
      READ(IRE,350,END=9000)ICODE
  350 FORMAT(A4)
C
C     ICODE - DESIRED STATION FROM 'response' FILE
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','ICOMP',/)
      READ(IRE,350,END=9000)ICOMP
C
C     ICOMP - DESIRED COMPONENT FROM 'response' FILE
C
      WRITE(IWR,500)ICODE,ICOMP
      WRITE(IWR2,500)ICODE,ICOMP
  500 FORMAT(/,10X,'ICODE = ',A4,5X,'ICOMP = ',A4,/)
C
C     READ DESIRED DATE
C
```

```
      WRITE(IWR2,600)
  600 FORMAT(/,'++INPUT: ','IYR',3X,'IMO',3X,'IDY',/)
      READ(IRE,*,END=9000)IYR,IMO,IDY
C
C     IYR - DESIRED YEAR
C     IMO - DESIRED MONTH (NUMBER)
C     IDY - DESIRED DAY
C
      IF(IYR.LE.1960)GO TO 7400
      IF(IMO.LE.0.0)GO TO 7400
      IF(IDY.LE.0.0)GO TO 7400
      WRITE(IWR,650)IYR,IMO,IDY
      WRITE(IWR2,650)IYR,IMO,IDY
  650 FORMAT(/,10X,'IYR = ',I4,5X,'IMO = ',I2,5X,'IDY = ',I2,/)
C
C     READ RESPONSE FILE
C
      OPEN(LUN3,FILE='response',STATUS='old')
C
      REWIND LUN3
C
  700 READ(LUN3,750,END=7000)KSTA,KCOMP,JBMO,JBDY,JBYR,JEMO,JEDY,JEYR,
     * SA0,KNP,KNZ,KFLAG,JNDEX
  750 FORMAT(A4,1X,A4,1X,I2,1X,I2,1X,I4,1X,I2,1X,I2,1X,I4,1X,E10.3,
     * 1X,I2,1X,I2,1X,I4,1X,A4)
C
C     KSTA - STATION CODE
C     KCOMP - INSTRUMENT COMPONENT
C     JBMO - RESPONSE BEGIN MONTH
C     JBDY - RESPONSE BEGIN DAY
C     JBYR - RESPONSE BEGIN YEAR
C     JEMO - RESPONSE END MONTH
C     JEDY - RESPONSE END DAY
C     JEYR - RESPONSE END YEAR
C     SA0 - CONSTANT RESPONSE TERM
C     KNP - NUMBER OF COMPLEX POLES
C     KNZ - NUMBER OF COMPLEX ZEROS
C     KFLAG - SPECIAL DATA FLAG
C     JNDEX - POINTER TO RESPONSE
C
      IF(KSTA.EQ.KSTOP)GO TO 7000
      IF(KSTA.NE.ICODE)GO TO 700
      IF(KCOMP.NE.ICOMP)GO TO 700
      IF(IYR.LT.JBYR)GO TO 700
      IF(IYR.GT.JEYR)GO TO 700
      IF(IYR.GT.JBYR.AND.IYR.LT.JEYR)GO TO 900
C
C     CONVERT DATES TO JULIAN DAYS
C
      CALL JUL(IYR,IMO,IDY,IDD,LEAP)
C
      CALL JUL(JBYR,JBMO,JBDY,JBB,LEAP)
C
      CALL JUL(JEYR,JEMO,JEDY,JEE,LEAP)
C
      IF(IYR.EQ.JBYR.AND.IDD.LT.JBB)GO TO 700
      IF(IYR.EQ.JEYR.AND.IDD.GT.JEE)GO TO 700
C
```

```
C     READ RESPONSE FILE UNTIL POLES AND ZEROS FOUND
C
  900 READ(LUN3,350,END=7000)KNDEX
C
C     KNDEX - RESPONSE NAME FOR POINTER
C
      IF(KNDEX.NE.JNDEX)GO TO 900
C
C     READ POLES AND ZEROS
C
      KNP2=KNP*2
      KNZ2=KNZ*2
C
      READ(LUN3,920,END=7000)(EP(I),I=1,KNP2)
  920 FORMAT(4(E10.3,1X))
C
      READ(LUN3,350,END=7000)KZERO
C
      IF(KZERO.EQ.NSTOP)GO TO 7000
      IF(KZERO.NE.NZERO)GO TO 7000
C
      READ(LUN3,920,END=7000)(EZ(I),I=1,KNZ2)
C
      REWIND LUN3
C
      CLOSE (LUN3)
C
C     TRANSFER POLES AND ZEROS FROM MASTER FILE TO COMMON/RESPON/
C
      DO 950 I=1,KNP
      SPOLES(I)=XPOLES(I)
  950 CONTINUE
C
      DO 1000 I=1,KNZ
      SZEROS(I)=XZEROS(I)
 1000 CONTINUE
C
      GO TO 5900
C
C     READ RESPONSE POLES AND ZEROS
C
 5000 WRITE(IWR2,5100)
 5100 FORMAT(/,'++ INPUT: ','SA0',/)
      READ(IRE,*,END=9000)SA0
C
C     SA0 - CONSTANT RESPONSE TERM
C
      WRITE(IWR2,5200)
 5200 FORMAT(/,'++ INPUT: ','KNP',/)
      READ(IRE,*,END=9000)KNP
C
C     KNP - NUMBER OF COMPLEX POLES
C
      WRITE(IWR2,5300)
 5300 FORMAT(/,'++ INPUT: ','SPOLES',/)
      READ(IRE,*,END=9000)(SPOLES(I),I=1,30)
C
C     SPOLES - COMPLEX POLES
```

```
C
      WRITE(IWR2,5400)
 5400 FORMAT(/,'++ INPUT: ','KNZ',/)
      READ(IRE,*,END=9000)KNZ
C
C     KNZ - NUMBER OF COMPLEX ZEROS
C
      WRITE(IWR2,5500)
 5500 FORMAT(/,'++ INPUT: ','SZEROS',/)
      READ(IRE,*,END=9000)(SZEROS(I),I=1,20)
C
C     SZEROS - COMPLEX ZEROS
C
      WRITE(IWR2,5600)
 5600 FORMAT(/,'++ INPUT: ','KFLAG',/)
      READ(IRE,*,END=9000)KFLAG
C
C     KFLAG - SPECIAL DATA FLAG
C
C     LIST RESPONSE INFORMATION
C
 5900 WRITE(IWR,6000)
      WRITE(IWR2,6000)
 6000 FORMAT(/,10X,'RESPONSE POLES',/)
      WRITE(IWR,6100)(SPOLES(K),K=1,KNP)
      WRITE(IWR2,6100)(SPOLES(K),K=1,KNP)
 6100 FORMAT(10X,E10.3,3X,E10.3,6X,E10.3,3X,E10.3)
      WRITE(IWR,6200)
      WRITE(IWR2,6200)
 6200 FORMAT(/,10X,'RESPONSE ZEROS',/)
      WRITE(IWR,6100)(SZEROS(K),K=1,KNZ)
      WRITE(IWR2,6100)(SZEROS(K),K=1,KNZ)
      WRITE(IWR,6400)SA0
      WRITE(IWR2,6400)SA0
 6400 FORMAT(/,10X,'SA0 = ',E10.3)
      WRITE(IWR,6500)KFLAG
      WRITE(IWR2,6500)KFLAG
 6500 FORMAT(/,10X,'KFLAG = ',I8,/)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'STATION-COMPONENT RESPONSE NOT AVAILABLE',/)
 7200 WRITE(IWR2,7300)
 7300 FORMAT(/,10X,'RESPONSE NOT SET UP',/)
      GO TO 9000
C
 7400 WRITE(IWR2,7500)
 7500 FORMAT(/,10X,'ERROR IN INPUT PARAMETERS',/)
      GO TO 7200
C
 8000 WRITE(IWR,8100)ICODE,ICOMP
      WRITE(IWR2,8100)ICODE,ICOMP
 8100 FORMAT(/,10X,A4,' - ',A4,/)
      WRITE(IWR,8200)
      WRITE(IWR2,8200)
 8200 FORMAT(/,10X,'RESPONSE SET UP IN MASTER FILE',/)
C
 9000 RETURN
```

```
      SUBROUTINE SETST
C
C*****************************************************************
C
C     READ START TIME OF DATA FROM THE KEYBOARD
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 24, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL TO CHANGE HEADER
C             = 0 SET TO 1
C
      IF(ICHAN.LE.0)ICHAN=1
      WRITE(IWR,150)ICHAN
      WRITE(IWR2,150)ICHAN
  150 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','JDOFY',3X,'JHOUR',3X,'JMIN',3X,'RSEC',
     * ,/)
      READ(IRE,*,END=9000)KDOFY,KHOUR,KMIN,RKSEC
C
C     JDOFY - NEW START DAY
C     JHOUR - NEW START HOUR
C     JMIN - NEW START MINUTE
C     RSEC - NEW START SECOND
C
      WRITE(IWR,300)
      WRITE(IWR2,300)
  300 FORMAT(/,10X,'NEW START TIME',/)
      WRITE(IWR,400)KDOFY,KHOUR,KMIN,RKSEC
      WRITE(IWR2,400)KDOFY,KHOUR,KMIN,RKSEC
  400 FORMAT(/,10X,'JDOFY = ',I4,3X,'JHOUR = ',I3,3X,'JMIN = ',I3,
     * 3X,'RSEC = ',F7.3,/)
C
      JDOFY(ICHAN)=KDOFY
      JHOUR(ICHAN)=KHOUR
```

```
      JMIN(ICHAN)=KMIN
      RSEC(ICHAN)=RKSEC
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'HEADER INFORMATION NOT CHANGED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'START TIME SUCCESSFULLY CHANGED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SFS(X,ISAMP,SFX)
C
C*********************************************************************
C
C     DETERMINE MAXIMUM O-P VALUE OF DATA CHANNEL
C
C     X - DATA CHANNEL - INPUT
C     ISAMP - NUMBER OF SAMPLES IN DATA CHANNEL - INPUT
C     SFX - MAXIMUM O-P ABSOLUTE VALUE OF DATA CHANNEL - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 13, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1)
C
      SFX=0.0
C
      DO 100 J=1,ISAMP
      TEMP=ABS(X(J))
      IF(TEMP.GT.SFX)SFX=TEMP
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SHIFT(X,ISAMP,NS,LENG,IERR)
C
C*********************************************************************
C
C     SHIFT A TIME SERIES
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT AND OUTPUT
C     NS - TOTAL NUMBER OF SAMPLES TO SHIFT (SKIP) - INPUT
C          > 0 SHIFT TO RIGHT (INCREASE DELAY)
C          < 0 SHIFT TO LEFT (REDUCE DELAY)
C     LENG - TOTAL BUFFER SIZE AVAILABLE - INPUT
C     IERR - ERROR STATUS FLAG - OUTPUT
C            = 0 NO ERROR
C            = 1 ERROR OCCURRED
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 7, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION X(1)
C
      IERR=0
      IF(NS.EQ.0)GO TO 9000
      IF(NS.GT.0)GO TO 2000
C
C     SHIFT DATA TO REDUCE DELAY
C
      NX=-NS
      IFP=NX+1
      ILP=ISAMP
      K=1
C
      DO 500 J=IFP,ILP
      X(K)=X(J)
      K=K+1
  500 CONTINUE
C
C     ZERO END OF TRACE
C
      IFP=ISAMP-NX+1
C
      DO 1000 J=IFP,ILP
      X(J)=0.0
 1000 CONTINUE
C
      ISAMP=ISAMP-NX
      GO TO 9000
C
C     SHIFT DATA TO INCREASE DELAY
C
 2000 NSAMP=ISAMP+NS
      IFP=1
      ILP=ISAMP
```

```
      IF(NSAMP.LE.LENG)GO TO 2200
      MN=NSAMP-LENG
      ILP=ILP-MN
      NSAMP=LENG
      IF(ILP.LE.1)GO TO 7000
C
 2200 DO 2500 J=IFP,ILP
      K=NSAMP-J+1
      L=ILP-J+1
      X(K)=X(L)
 2500 CONTINUE
C
C     ZERO BEGINNING OF TRACE
C
      DO 3000 J=1,NS
      X(J)=0.0
 3000 CONTINUE
C
      ISAMP=NSAMP
      GO TO 9000
C
 7000 IERR=1
C
 9000 RETURN
      END
```

```
      SUBROUTINE SHIFTX
C
C*****************************************************************
C
C     EXECUTIVE ROUTINE FOR SHIFTING DATA IN A CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 23, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1)
C
      COMPLEX RPOLES,RZEROS
C
C     READ SHIFTING PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',3X,'DELAY',/)
      READ(IRE,*,END=9000)ICHAN,DELAY
C
C     ICHAN - DESIRED CHANNEL TO SHIFT
C     DELAY - DELAY TIME (IN SECONDS)
C             > 0 INCREASE DELAY
C             < 0 DELETE FIRST PART OF TRACE TO ADVANCE TRACE
C
      WRITE(IWR,200)ICHAN,DELAY
      WRITE(IWR2,200)ICHAN,DELAY
  200 FORMAT(/,10X,'ICHAN = ',I4,5X,'DELAY = ',F8.3,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
      IF(ICHAN.LE.0)GO TO 7000
      ISAMP=JSAMP(ICHAN)
      NS=DELAY*RSAMR(ICHAN)
      IF(NS.EQ.0)GO TO 8000
      MNS=-NS
      IF(NS.LT.0.AND.MNS.GT.ISAMP)GO TO 7000
      INDEX=(ICHAN-1)*LENG+1
C
C     SHIFT TRACE
C
      CALL SHIFT(DATA(INDEX),ISAMP,NS,LENG,IERR)
C
      IF(IERR.EQ.1)GO TO 7000
      JSAMP(ICHAN)=ISAMP
      GO TO 8000
C
```

```
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT SHIFTED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)NS
      WRITE(IWR2,8100)NS
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY SHIFTED ',I6,' SAMPLES',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SIG1(X,ISAMP,SAMR,F,A,B,A0)
C
C*********************************************************************
C
C     CALCULATE SYNTHETIC SINE/COSINE SIGNAL COMPONENT
C
C     X - GENERATED SIGNAL - OUTPUT
C     ISAMP - NUMBER OF SAMPLES TO GENERATE - INPUT
C     SAMR - SAMPLING RATE OF GENERATED SIGNAL (IN SAMPLES/S) - INPUT
C     F - FREQUENCY OF SINE/COSINE COMPONENT - INPUT
C     A - COSINE COEFFICIENT - INPUT
C     B - SINE COEFFICIENT - INPUT
C     A0 - DC OFFSET - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 22, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION X(1)
C
      ARG=TWOPI*F/SAMR
C
      DO 500 J=1,ISAMP
      FJ=J-1
      X(J)=A*COS(ARG*FJ)+B*SIN(ARG*FJ)+A0
  500 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SIG2(X,ISAMP,IS,AMP)
C
C*****************************************************************************
C
C     CALCULATE SYNTHETIC DELTA FUNCTION SIGNAL COMPONENT
C
C     X - GENERATED SIGNAL - OUTPUT
C     ISAMP - NUMBER OF SAMPLES TO GENERATE - INPUT
C     IS - SAMPLE NUMBER TO INSERT DELTA FUNCTION - INPUT
C     AMP - AMPLITUDE OF DELTA FUNCTION - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 22, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION X(1)
C
      DO 100 J=1,ISAMP
      X(J)=0.0
  100 CONTINUE
C
      X(IS)=AMP
C
      RETURN
      END
```

```
      SUBROUTINE SIG3(X,ISAMP,SAMR,F)
C
C*********************************************************************
C
C     CALCULATE SYNTHETIC FUNCTION (t**2)*exp(2-t)*sin(2pift)
C
C     X - GENERATED SIGNAL - OUTPUT
C     ISAMP - NUMBER OF SAMPLES TO GENERATE - INPUT
C     SAMR - SAMPLING RATE OF GENERATED SIGNAL - INPUT
C     F - FREQUENCY OF SINE COMPONENT - INPUT
C
C     REFERENCES
C     KULHANEK AND KLIMA,1970,GEOPHYS. J.,21,235-242.
C     FARNBACH,1975,BSSA,65,951-962.
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 25, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION X(1)
C
      ARG2=TWOPI*F
C
      DO 500 J=1,ISAMP
      FJ=(J-1)/SAMR
      ARG1=2.0-2.0*FJ
      X(J)=FJ*FJ*EXP(ARG1)*SIN(ARG2*FJ)
  500 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SIGNAL
C
C*****************************************************************
C
C     CALCULATE SYNTHETIC TEST SIGNALS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 22, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
      COMMON/DEM/IDEM,IDEM1
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1),BX(1)
C
      COMPLEX RPOLES,RZEROS
C
C     READ SIGNAL OPTIONS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',3X,'SAMR',3X,'NSIG',3X,
     * 'KSAMP',/)
      READ(IRE,*,END=9000)ICHAN,SAMR,NSIG,KSAMP
C
C     ICHAN - CHANNEL TO ADD SYNTHETIC SIGNAL
C             = 0 SET TO 1
C             ***** CAUTION SIGNALS ARE ADDED TO CHANNEL AS IS;
C                   CHANNEL IS NOT SET TO ZERO *****
C     SAMR - SAMPLING RATE (IN SAMPLES/S)
C            = 0 SET TO 40.0
C     NSIG - NUMBER OF DIFFERENT SIGNALS TO COMPUTE AND ADD
C            TO ICHAN
C            = 0 SET TO 1
C     KSAMP - NUMBER OF SAMPLES TO SET FOR DATA CHANNEL
C             = 0 USE PRE-EXISTING NUMBER
C
      IF(ICHAN.LE.0)ICHAN=1
      IF(ICHAN.GT.NACHLS)NACHLS=ICHAN
      IF(SAMR.LE.0.0)SAMR=40.0
      IF(NSIG.LE.0)NSIG=1
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
  250 WRITE(IWR,260)
      WRITE(IWR2,260)
  260 FORMAT(/,10X,'SIGNAL PARAMETERS',/)
```

```
      WRITE(IWR,300)ICHAN,SAMR,NSIG,KSAMP
      WRITE(IWR2,300)ICHAN,SAMR,NSIG,KSAMP
  300 FORMAT(/,10X,'ICHAN = ',I6,5X,'SAMR = ',F7.3,5X,'NSIG = ',I5,
     * /,10X,'KSAMP = ',I6,/)
C
C     ZERO WORK AREA
C
  400 DO 500 I=1,IDEM1
      BX(I)=0.0
  500 CONTINUE
C
C     GENERATE ALL SIGNALS
C
      DO 6500 I=1,NSIG
 1000 WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','ISNX',/)
      READ(IRE,*,END=9000)ISNX
C
C     ISNX - SIGNAL TYPE
C             = 0 SET TO 1
C             = 1 SINE/COSINE SIGNAL
C             = 2 DELTA FUNCTION
C             = 3 SPECIAL FUNCTION # 1
C                 f(t)=(t**2)*exp(2-2t)*sin(2pift)
C
      IF(ISNX.LE.0)ISNX=1
      WRITE(IWR,1500)ISNX
      WRITE(IWR2,1500)ISNX
 1500 FORMAT(/,10X,'ISNX = ',I3,/)
C
      WRITE(IWR2,2000)
 2000 FORMAT(/,'++ INPUT: ','ISAMP',3X,'IFIR',/)
      READ(IRE,*,END=9000)ISAMP,IFIR
C
C     ISAMP - NUMBER OF SAMPLES TO GENERATE
C             = 0 SET TO LENG
C     IFIR - FIRST SAMPLE TO ADD GENERATED SIGNAL TO
C            = 0 SET TO 1
C
      IF(ISAMP.LE.0)ISAMP=LENG
      IF(IFIR.LE.0)IFIR=1
      WRITE(IWR,2100)ISAMP,IFIR
      WRITE(IWR2,2100)ISAMP,IFIR
 2100 FORMAT(/,10X,'ISAMP = ',I6,5X,'IFIR = ',I6,/)
C
      IF(IFIR.GT.LENG)GO TO 7000
      IF((ISAMP+IFIR-1).GT.LENG)GO TO 7000
C
C     READ WEIGHTING AND SCALING PARAMETERS
C
      WRITE(IWR2,2300)
 2300 FORMAT(/,'++ INPUT: ','ALPHA',3X,'SCALE',/)
      READ(IRE,*,END=9000)ALPHA,SCALE
C
C     ALPHA - EXPONENTIAL DECAY CONSTANT exp(-alpha*t)
C             = 0 NO EXPONENTIAL WEIGHTING
C     SCALE - SCALE FACTOR FOR SIGNAL (MAXIMUM VALUE FOR SIGNAL)
C             = 0 SET TO 1.0
```

```
C
      IF(SCALE.EQ.0.0)SCALE=1.0
      WRITE(IWR,2500)ALPHA,SCALE
      WRITE(IWR2,2500)ALPHA,SCALE
 2500 FORMAT(/,10X,'ALPHA = ',F10.4,5X,'SCALE = ',F10.4,/)
C
      IF(ISNX.NE.1)GO TO 3500
C
C     CALCULATE SINE/COSINE SIGNAL
C
      WRITE(IWR2,2700)
 2700 FORMAT(/,'++ INPUT: ','T',3X,'A',3X,'B',3X,'A0',/)
      READ(IRE,*,END=9000)T,A,B,A0
C
C     T - PERIOD OF SINE/COSINE COMPONENT
C     A - COSINE COEFFICIENT
C     B - SINE COEFFICIENT
C     A0 - DC OFFSET
C
      WRITE(IWR,2900)T,A,B,A0
      WRITE(IWR2,2900)T,A,B,A0
 2900 FORMAT(/,10X,'T = ',F10.4,5X,'A  = ',F10.4,5X,'B = ',F10.4,/,
     * 29X,'A0 = ',F10.4,/)
C
      F=1.0/T
C
      CALL SIG1(BX,ISAMP,SAMR,F,A,B,A0)
C
      GO TO 6000
C
 3500 IF(ISNX.NE.2)GO TO 4500
C
C     CALCULATE DELTA FUNCTION
C
      WRITE(IWR2,3600)
 3600 FORMAT(/,'++ INPUT: ','AMP',3X,'IS',/)
      READ(IRE,*,END=9000)AMP,IS
C
C     AMP - AMPLITUDE OF DELTA FUNCTION
C     IS - SAMPLE NUMBER TO PLACE DELTA FUNCTION
C
      WRITE(IWR,3700)AMP,IS
      WRITE(IWR2,3700)AMP,IS
 3700 FORMAT(/,10X,'AMP = ',F10.4,5X,'IS = ',I5,/)
C
      CALL SIG2(BX,ISAMP,IS,AMP)
C
      GO TO 6000
C
 4500 IF(ISNX.NE.3)GO TO 6500
C
C     CALCULATE SPECIAL FUNCTION # 1
C
      WRITE(IWR2,4600)
 4600 FORMAT(/,'++ INPUT: ','T',/)
      READ(IRE,*,END=9000)T
C
C     T - PERIOD OF SINE COMPONENT
```

```
C
      WRITE(IWR,4700)T
      WRITE(IWR2,4700)T
 4700 FORMAT(/,10X,'T = ',F12.5,/)
C
      F=1.0/T
C
      CALL SIG3(BX,ISAMP,SAMR,F)
C
C     ATTENUATE SIGNAL
C
 6000 IF(ALPHA.EQ.0.0)GO TO 6100
C
      CALL EXPWF(BX,ISAMP,SAMR,ALPHA)
C
 6100 IF(SCALE.EQ.1.0)GO TO 6200
C
C     SCALE SIGNAL
C
      CALL SCALTS(BX,ISAMP,SCALE)
C
 6200 INDEX=(ICHAN-1)*LENG+IFIR
C
C     ADD SIGNAL TO DATA CHANNEL
C
      DO 6300 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)+BX(J)
 6300 CONTINUE
C
 6500 CONTINUE
C
      RSAMR(ICHAN)=SAMR
      IF(KSAMP.GT.0)JSAMP(ICHAN)=KSAMP
      IF(JSAMP(ICHAN).LE.0)JSAMP(ICHAN)=ISAMP
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'SIGNAL NOT GENERATED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'SIGNAL SUCCESSFULLY GENERATED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SPECTR
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR FOURIER TRANSFORM
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 25, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1),BX(1),A(1),B(1),C(1),PH(1)
C
      EQUIVALENCE (BX(20001),A(1)),(BX(30001),B(1)),(BX(40001),C(1))
      EQUIVALENCE (BX(50001),PH(1))
C
      COMPLEX RPOLES,RZEROS
C
      ISIGN=1
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','IPOW',/)
      READ(IRE,*,END=9000)IPOW
C
C     IPOW - SPECTRA OPTION
C            = 0 DIRECT TRANSFORM TO REAL AND IMAGINARY COMPONENTS
C            = 1 INDIRECT TRANSFORM TO TIME DOMAIN
C            = 2 DIRECT TRANSFORM TO POWER SPECTRA
C                (LIMITED TO 10000 POINTS)
C            = 3 DIRECT TRANSFORM TO AMPLITUDE AND PHASE SPECTRA
C                (LIMITED TO 10000 POINTS)
C            = 4 SLOW TEST TRANSFORM
C                (LIMITED TO 10000 POINTS)
C
      WRITE(IWR,400)IPOW
      WRITE(IWR2,400)IPOW
  400 FORMAT(/,10X,'IPOW = ',I2,/)
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL NUMBER TO COMPUTE SPECTRA OF
C             > 0 MOVE ICHAN TO BX AND COMPUTE FOR BX
C             = 0 COMPUTE FOR BX
C
```

```
      WRITE(IWR,600)ICHAN
      WRITE(IWR2,600)ICHAN
  600 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,700)
  700 FORMAT(/,'++ INPUT: ','ISAMP',3X,'IFIR',3X,'SAMR',/)
      READ(IRE,*,END=9000)ISAMP,IFIR,SAMR
C
C     ISAMP - NUMBER OF SAMPLES TO MOVE TO BX
C             = 0 MOVE ALL SAMPLES
C     IFIR - FIRST SAMPLE TO MOVE
C            = 0 SET TO 1
C     SAMR - SAMPLING RATE (IN SAMPLES/S)
C            = 0 USE VALUE FOR CHANNEL
C
      IF(IFIR.LE.0)IFIR=1
      IF(ISAMP.LE.0.AND.ICHAN.EQ.0)GO TO 7000
      IF(ISAMP.LE.0)ISAMP=JSAMP(ICHAN)-IFIR+1
      IF(SAMR.LE.0.0.AND.ICHAN.EQ.0)GO TO 7000
      IF(SAMR.LE.0.0)SAMR=RSAMR(ICHAN)
      WRITE(IWR,800)ISAMP,IFIR,SAMR
      WRITE(IWR2,800)ISAMP,IFIR,SAMR
  800 FORMAT(/,10X,'ISAMP = ',I6,5X,'IFIR = ',I4,5X,'SAMR = ',F6.2,/)
C
      IF(ISAMP.LE.0.OR.SAMR.LE.0.0)GO TO 7000
C
      IF(IPOW.EQ.4)GO TO 1700
      IF(IPOW.NE.1)GO TO 1000
      ISIGN=-1
      GO TO 1300
C
 1000 WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','NSM',/)
      READ(IRE,*,END=9000)NSM
C
C     NSM - NUMBER OF POINTS TO SMOOTH SPECTRA OVER
C           (MUST BE BETWEEN 5 AND 25)
C           = 0 NO SMOOTHING
C
      WRITE(IWR,1200)NSM
      WRITE(IWR2,1200)NSM
 1200 FORMAT(/,10X,'NSM = ',I3,/)
C
 1300 IF(ICHAN.LE.0)GO TO 1500
      IFL=1
      IB1=IFIR
      IL1=IB1+ISAMP-1
      NCH1=ICHAN
      IB2=1
      NCH2=1
      IHF=0
C
C     MOVE DATA CHANNEL TO BX
C
      CALL MOVE(IFL,IB1,IL1,NCH1,IB2,NCH2,IHF)
C
```

```
C     CALCULATE TRANSFORM OF DATA IN BX
C
 1500 CALL TTOF1(BX,ISAMP,LENG,ISIGN,N2PWR,N2LEN,NH2,N2P1,NZ)
C
      WRITE(IWR,1600)N2LEN
      WRITE(IWR2,1600)N2LEN
 1600 FORMAT(/,10X,'NUMBER OF SAMPLES TRANSFORMED = ',I6,/)
      GO TO 2000
C
 1700 IF(ICHAN.LE.0)GO TO 1750
      INDEX=(ICHAN-1)*LENG+1
C
C     CALCULATE SLOW TRANSFORM OF DATA IN DATA ARRAY
C
      CALL TTOF3(DATA(INDEX),ISAMP,SAMR,LENG,A,B,C,PH,N2PWR,N2LEN,
     * NH2,N2P1,NZ)
C
      GO TO 1780
C
C     CALCULATE SLOW TRANSFORM OF DATA IN BX ARRAY
C
 1750 CALL TTOF3(BX,ISAMP,SAMR,LENG,A,B,C,PH,N2PWR,N2LEN,NH2,
     * N2P1,NZ)
C
C     LIST SPECTRAL DATA
C
 1780 CALL PRSPC3(N2LEN,NH2,SAMR,A,B,C,PH)
C
      GO TO 9000
C
 2000 IF(IPOW.EQ.1)GO TO 9000
      IF(IPOW.LE.0)GO TO 4000
      IF(IPOW.GT.3)GO TO 9000
C
C     CALCULATE AMPLITUDE AND PHASE
C
      CALL AMPPH1(BX,C,PH,NH2)
C
      IF(IPOW.NE.2)GO TO 3000
C
C     CALCULATE POWER
C
      CALL POW(C,N2LEN,NH2)
C
 3000 IF(NSM.LE.0)GO TO 4000
C
C     SMOOTH SPECTRAL DATA
C
      CALL RAVAGE(C,N2LEN,NSM)
C
 4000 WRITE(IWR2,4100)
 4100 FORMAT(/,'++ INPUT: ','IPRT',3X,'IPLOT',/)
      READ(IRE,*,END=9000)IPRT,IPLOT
C
C     IPRT - LIST OPTION
C            > 0 THEN LIST
C     IPLOT - PLOT SPECTRA OPTION
C             > 0 THEN PLOT SPECTRA
```

```
C
      WRITE(IWR,4300)IPRT,IPLOT
      WRITE(IWR2,4300)IPRT,IPLOT
 4300 FORMAT(/,10X,'IPRT = ',I2,5X,'IPLOT = ',I2,/)
C
      IF(IPRT.LE.0.AND.IPLOT.LE.0)GO TO 9000
C
      WRITE(IWR2,4800)
 4800 FORMAT(/,'++ INPUT: ','TMN',3X,'TMX',/)
      READ(IRE,*,END=9000)TMN,TMX
C
C     TMN - SHORTEST PERIOD TO PRINT AND PLOT
C           = 0 USE DEFAULT FROM SPECTRA
C     TMX - LONGEST PERIOD TO PRINT AND PLOT
C           = 0 USE DEFAULT FROM SPECTRA
C
      WRITE(IWR,5000)TMN,TMX
      WRITE(IWR2,5000)TMN,TMX
 5000 FORMAT(/,10X,'TMN = ',F10.3,5X,'TMX = ',F10.3,/)
C
      FMX=0.0
      FMN=0.0
      IF(TMX.GT.0.0)FMN=1.0/TMX
      IF(TMN.GT.0.0)FMX=1.0/TMN
C
      IF(IPRT.LE.0)GO TO 6000
C
C     LIST SPECTRAL DATA
C
      CALL PRSPC1(BX,C,PH,NH2,N2LEN,N2P1,FMN,FMX,SAMR,IPOW)
C
 6000 IF(IPLOT.LE.0)GO TO 9000
C
C     PLOT SPECTRAL DATA
C
      WRITE(IWR2,6200)
 6200 FORMAT(/,'++ INPUT: ','IFFT',/)
      READ(IRE,*,END=9000)IFFT
C
C     IFFT - FREQUENCY/PERIOD FLAG
C            = 0 PLOT FREQUENCY VALUES
C            = 1 PLOT PERIOD VALUES
C
      WRITE(IWR,6300)IFFT
      WRITE(IWR2,6300)IFFT
 6300 FORMAT(/,10X,'IFFT = ',I2,/)
C
      DF=0.0
C
      CALL PLSPEC(C,PH,NH2,N2LEN,FMN,FMX,DF,SAMR,IPOW,IFFT)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'SPECTRAL OPERATION NOT COMPLETED',/)
      GO TO 9000
```

```
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'SPECTRAL OPERATIONS SUCCESSFULLY COMPLETED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SPLDAT(TF,TZERO,NP,SCALT,FSDIST,DMAX,SCALD,
     * SAMR,S,FAC)
C
C*********************************************************************
C
C     POSITION AND PLOT A SINGLE TIME SERIES
C
C     TF - PLOT ORIGIN FOR TIME (IN SECONDS) - INPUT
C     TZERO - LOCATION OF FIRST POINT TO PLOT (IN SECONDS) - INPUT
C     NP - NUMBER OF POINTS TO PLOT - INPUT
C     SCALT - TRACE PLOTTING INCREMENT (IN INCHES/S) - INPUT
C     FSDIST - PLOT ORIGIN FOR DISTANCE - INPUT
C     DMAX - MAXIMUM DISTANCE VALUE - INPUT
C            (PLOT SIZE MUST BE 1.5 INCHES LARGER THAN THIS)
C     SCALD - DISTANCE PLOTTING INCREMENT (IN IN/KILOMETER) - INPUT
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     S - TIME SERIES - INPUT
C     FAC - PLOT SCALE FACTOR - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 18, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION S(1)
C
      IF(NP.LE.0)GO TO 9000
C
      CALL FACTOR(FAC)
C
      YLT=(DMAX-FSDIST)*SCALD+1.50
      TT=(TZERO-TF)*SCALT+0.50
C
      DO 500 I=1,NP
      FI=I-1
      T=TT+(FI/SAMR)*SCALT
      D=(S(I)-FSDIST)*SCALD
      D=YLT-D
      IF(I.GT.1)GO TO 150
C
      CALL PLOT(T,D,3)
C
      CALL PLOT(T,D,2)
C
      GO TO 500
C
  150 CALL PLOT(T,D,1)
C
  500 CONTINUE
C
C     POSITION PEN BACK AT BEGINNING OF TRACE
C
      T=TT
C
      CALL PLOT(T,D,3)
C
      CALL FACTOR(1.0)
```

```
      SUBROUTINE SPLOT(X,Y,IFIR,ILAS,INC,IYMAX,KMAX,XS,XL,YS,YL)
C
C*********************************************************************
C
C     PRINTER PLOT OF X VS Y DATA
C
C     PLOTS X IN A 0-70 FIELD VS Y IN A 0-IYMAX FIELD
C
C     X - DATA ARRAY - INPUT
C     Y - DATA ARRAY - INPUT
C     IFIR - FIRST POINT IN ARRAYS TO PLOT - INPUT
C          = 0 SET TO 1
C     ILAS - LAST POINT IN ARRAYS TO PLOT - INPUT
C     INC - INCREMENT FOR SELECTING DATA POINTS TO PLOT - INPUT
C         = 0 SET TO 1
C     IYMAX - MAXIMUM NUMBER OF LINES IN Y FIELD - INPUT
C             = 0 SET TO 55
C     KMAX - MAX/MIN FLAG - INPUT
C            = 0 PLOT ALL VALUES
C            = 1 MAXIMUM AND MINIMUM LIMITS PROVIDED
C     XS - SMALLEST X VALUE PLOTTED - INPUT OR OUTPUT
C     XL - LARGEST X VALUE PLOTTED - INPUT OR OUTPUT
C     YS - SMALLEST Y VALUE PLOTTED - INPUT OR OUTPUT
C     YL - LARGEST Y VALUE PLOTTED - INPUT OR OUTPUT
C
C     NOVEMBER 15, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION X(1),Y(1),FX(61),ZX(6)
C
      INTEGER*4 FX,TSTAR,EYE,BLNK
C
      DATA TSTAR/'*'/,EYE/'I'/,BLNK/' '/
C
      IF(IFIR.LE.0)IFIR=1
      IF(INC.LE.0)INC=1
      IF(IYMAX.LE.0)IYMAX=55
      MMAX=IYMAX+1
      IF(KMAX.GT.0)GO TO 30
C
C     DETERMINE LARGEST AND SMALLEST X AND Y VALUES
C
      XL=X(IFIR)
      XS=X(IFIR)
      YL=Y(IFIR)
      YS=Y(IFIR)
C
      DO 20 JK=IFIR,ILAS,INC
      IF(XL.LT.X(JK))XL=X(JK)
      IF(XS.GT.X(JK))XS=X(JK)
      IF(YL.LT.Y(JK))YL=Y(JK)
      IF(YS.GT.Y(JK))YS=Y(JK)
   20 CONTINUE
C
C     DETERMINE PLOT SCALE FACTORS
```

```
C
   30 XSCALE=(XL-XS)/60.0
      YSCALE=(YL-YS)/FLOAT(IYMAX)
      YHALF=YSCALE*0.5
C
      WRITE(IWR,50)
      WRITE(IWR2,50)
   50 FORMAT(/,1X,'Y VALUES ',6(10HI.........),'I')
C
      DO 1000 J=1,MMAX
C
      DO 200 I=2,60
      FX(I)=BLNK
  200 CONTINUE
C
      FX(1)=EYE
      FX(61)=EYE
C
      ZY=YSCALE*(J-1)+YS
C
      DO 500 I=IFIR,ILAS,INC
      IF(Y(I).GT.YL)GO TO 500
      IF(Y(I).LT.YS)GO TO 500
      IF(X(I).GT.XL)GO TO 500
      IF(X(I).LT.XS)GO TO 500
      IF(ABS(Y(I)-ZY).GT.YHALF)GO TO 500
      IX=((X(I)-XS)/XSCALE)+0.5
      II=IX+1
      FX(II)=TSTAR
  500 CONTINUE
C
C     PRINT LINE OF DATA
C
      WRITE(IWR,600)ZY,(FX(I),I=1,61)
      WRITE(IWR2,600)ZY,(FX(I),I=1,61)
  600 FORMAT(F9.3,1X,61A1)
C
 1000 CONTINUE
C
      WRITE(IWR,1200)
      WRITE(IWR2,1200)
 1200 FORMAT(10X,6(10HI.........),'I')
C
      DO 1500 I=1,11
      ZX(I)=10.0*FLOAT(I-1)*XSCALE+XS
 1500 CONTINUE
C
      WRITE(IWR,1600)(ZX(I),I=1,6)
      WRITE(IWR2,1600)(ZX(I),I=1,6)
 1600 FORMAT(5X,6F10.3,/)
C
      RETURN
      END
```

```
      SUBROUTINE SPLOTX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR PRINTER PLOT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 15, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1),BX(1)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'ICHAN2',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2
C
C     ICHAN1 - CHANNEL OF X DATA ARRAY
C              = 0 SET TO 1
C     ICHAN2 - CHANNEL OF Y DATA ARRAY
C              = 0 CREATE SEQUENTIAL INDEX ARRAY IN BX AS Y
C
      IF(ICHAN1.LE.0)ICHAN1=1
      WRITE(IWR,300)ICHAN1,ICHAN2
      WRITE(IWR2,300)ICHAN1,ICHAN2
  300 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','IFIR',3X,'ILAS',3X,'INC',/)
      READ(IRE,*,END=9000)IFIR,ILAS,INC
C
C     IFIR - FIRST POINT TO PLOT
C            = 0 SET TO 1
C     ILAS - LAST POINT TO PLOT
C            = 0 SET TO CHANNEL VALUE
C     INC - INCREMENT FOR SELECTING DATA POINTS TO PLOT
C           = 0 SET TO 1
C
      IF(IFIR.LE.0)IFIR=1
      IF(ILAS.LE.0)ILAS=JSAMP(ICHAN1)
      IF(INC.LE.0)INC=1
```

```
      WRITE(IWR,700)IFIR,ILAS,INC
      WRITE(IWR2,700)IFIR,ILAS,INC
  700 FORMAT(/,10X,'IFIR = ',I6,5X,'ILAS = ',I6,5X,'INC = ',I3,/)
C
      WRITE(IWR2,800)
  800 FORMAT(/,'++ INPUT: ','IYMAX',3X,'KMAX',/)
      READ(IRE,*,END=9000)IYMAX,KMAX
C
C     IYMAX - MAXIMUM NUMBER OF LINES IN Y FIELD
C     KMAX - MAX/MIN PLOT FLAG
C            = 0 PLOT ALL VALUES
C            = 1 READ IN MAXIMUM/MINIMUM VALUES
C
      WRITE(IWR,900)IYMAX,KMAX
      WRITE(IWR2,900)IYMAX,KMAX
  900 FORMAT(/,10X,'IYMAX = 'I4,5X,'KMAX = ',I2,/)
C
      IF(KMAX.LE.0)GO TO 1400
C
      WRITE(IWR2,1000)
 1000 FORMAT(/,'++ INPUT: ','XS',3X,'XL',3X,'YS',3X,'YL',/)
      READ(IRE,*,END=9000)XS,XL,YS,YL
C
C     XS - SMALLEST X VALUE TO PLOT
C     XL - LARGEST X VALUE TO PLOT
C     YS - SMALLEST Y VALUE TO PLOT
C     YL - LARGEST Y VALUE TO PLOT
C
      WRITE(IWR,1200)XS,XL,YS,YL
      WRITE(IWR2,1200)XS,XL,YS,YL
 1200 FORMAT(/,10X,'XS = ',F10.3,5X,'XL = ',F10.3,5X,/,10X,
     * 'YS = ',F10.3,5X,'YL = ',F10.3,/)
C
 1400 INDEX1=(ICHAN1-1)*LENG+IFIR
      IF(ICHAN2.LE.0)GO TO 2000
      INDEX2=(ICHAN2-1)*LENG+IFIR
C
      CALL SPLOT(DATA(INDEX1),DATA(INDEX2),IFIR,ILAS,INC,IYMAX,KMAX,
     * XS,XL,YS,YL)
C
      GO TO 9000
C
 2000 DO 2500 I=1,ILAS
      BX(I)=I
 2500 CONTINUE
C
      CALL SPLOT(DATA(INDEX1),BX,IFIR,ILAS,INC,IYMAX,KMAX,XS,XL,YS,YL)
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'ERROR IN INPUT PARAMETERS',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE SROOT(LVRY,MODENO,KWRITE,CFIR,DC,ACC,MMAX,BTAMX,KLAST,
     * KMAX)
C
C*********************************************************************
C
C     DETERMINE ROOTS OF THE PERIOD EQUATION FOR RAYLEIGH AND
C     LOVE WAVES
C
C     LVRY - RAYLEIGH/LOVE WAVE FLAG - INPUT
C            = 0 RAYLEIGH WAVE
C            = 1 LOVE WAVE
C     MODENO - MODE FLAG - INPUT
C              = 0 FUNDAMENTAL MODE
C              = 1 FIRST HIGHER MODE
C              = 2 SECOND HIGHER MODE
C              = n N'TH HIGHER MODE
C     KWRITE - DEBUG LISTING FLAG - INPUT
C              = 0 NO LISTING MADE
C              = 1 LISTING MADE OF ROOT SEARCH
C              = 2 LISTING MADE OF ROOT SEARCH AND LAYER DIVISION
C     CFIR - STARTING PHASE VELOCITY FOR SEARCH (IN KM/S) - INPUT
C     DC - INCREMENT IN PHASE VELOCITY FOR INITIAL SEARCH
C          (IN KM/S) - INPUT
C     ACC - ACCURACY CRITERIA - INPUT
C     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     BTAMX - LARGEST VALUE OF SHEAR VELOCITY (IN KM/S) - INPUT
C     KLAST - NUMBER OF ROOTS FOUND - OUTPUT
C     KMAX - NUMBER OF PERIOD VALUES - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 18, 1994  COLORADO
C
C*********************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION C(300),T(300),U(300),HVRAT(300),AMED(300),ENGK(300),
     * ENGP(300)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(30001),C(1)),(BX(31001),T(1))
      EQUIVALENCE (BX(32001),U(1)),(BX(33001),AMED(1))
      EQUIVALENCE (BX(34001),ENGK(1)),(BX(35001),ENGP(1))
      EQUIVALENCE (BX(36001),HVRAT(1))
C
      REAL*8 DELT1,DELT2,DDT,ONED,FACT
C
      K=1
      IPARMO=0
      IPDER=0
      NTRLMX=30
      ONE=1.0
      ONED=1.0
      MODE=MODENO
      ISICH=0
C
```

```
      IF(KWRITE.EQ.0)GO TO 200
      WRITE(IWR2,100)
  100 FORMAT(/,11X,'PERIOD',6X,'PHASE VELOCITY',10X,'PERIOD EQUATION',
     * /)
C
C     INITIALIZE PARAMETERS
C
  200 ILAST=0
      K2CT=0
      KNT=100
      C(1)=CFIR
      DELC=DC
      NTRLS=0
C
C     OBTAIN VALUE OF PERIOD EQUATION
C
      IF(LVRY.GT.0)GO TO 300
C
      CALL TEQUR(C(1),T(1),MMAX,LCT,HVR,KWRITE,NTRLS,DELT1)
C
      FACT=DABS(ONED/DELT1)
      DELT1=DELT1*FACT
      GO TO 400
C
  300 CALL TEQUL(C(1),T(1),MMAX,MODENO,ILAST,KWRITE,NTRLS,WVNOSQ,CSQ,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,DELT1)
C
  400 IF(NTRLS.EQ.0)GO TO 500
C
C     ERROR RETURN - TRY TO CONTINUE
C
  450 C(1)=C(1)+DC
      KNT=KNT-1
      IF(KNT.LT.0)GO TO 7200
C
  500 IF(KWRITE.EQ.0)GO TO 700
      WRITE(IWR2,600)T(1),C(1),DELT1
  600 FORMAT(10X,F7.3,6X,F12.9,7X,D20.9)
C
  700 C(1)=C(1)+DC
      IF(C(1).GT.BTAMX)GO TO 7000
      IF(C(1).LE.0.0)GO TO 7300
C
C     OBTAIN VALUE OF PERIOD EQUATION
C
      IF(LVRY.GT.0)GO TO 1100
C
      CALL TEQUR(C(1),T(1),MMAX,LCT,HVR,KWRITE,NTRLS,DELT2)
C
      DELT2=DELT2*FACT
      GO TO 1200
C
 1100 CALL TEQUL(C(1),T(1),MMAX,MODENO,ILAST,KWRITE,NTRLS,WVNOSQ,CSQ,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,DELT2)
C
 1200 IF(KWRITE.EQ.0)GO TO 1300
      WRITE(IWR2,600)T(1),C(1),DELT2
C
```

```
 1300 IF(NTRLS.NE.0)GO TO 450
C
C     DOES DELT1 HAVE THE SAME SIGN AS DELT2?
C
      IF((DELT1*DELT2).GT.0.0)GO TO 1400
      IF(MODE.LE.0)GO TO 4100
      MODE=MODE-1
 1400 DELT1=DELT2
      GO TO 700
C
C     EXTRAPOLATE TO OBTAIN NEXT TRIAL INCREMENT IN PHASE VELOCITY
C
 1500 IF(K.GE.5)GO TO 2100
      GO TO (4100,1600,1900,2000),K
C
C     OBTAIN SECOND SOLUTION TO DETERMINE SLOPE
C
 1600 IF(K2CT.GT.0)GO TO 1800
      K2CT=1
      TEMP=T(2)
      T(2)=T(1)+(T(2)-T(1))/40.0
      C(2)=C(1)
C
 1700 DELC=DC*0.01
      GO TO 2500
C
C     LINEAR INTERPOLATION FOR K=2
C
 1800 K2CT=0
      C(2)=C(2)+(C(2)-C(1))*(TEMP-T(2))/(T(2)-T(1))
      T(2)=TEMP
      GO TO 1700
C
C     USE PREVIOUS DELC FOR K=3
C
 1900 C(K)=C(2)+(C(2)-C(1))
      GO TO 1700
C
C     FIT PARABOLA TO ATTEMPTED SOLUTIONS OF PERIOD EQUATION FOR K=4
C
 2000 C(K)=C(K-1)+C(K-1)-C(K-2)
      CPAR=3.0*(C(K-1)-C(K-2))+C(K-3)
      DELC=CPAR-C(K)
      GO TO 2500
C
C     FIT CUBIC TO POINTS FOR K=5 OR GREATER
C
 2100 C(K)=3.0*(C(K-1)-C(K-2))+C(K-3)
      CCUB=4.0*(C(K-1)+C(K-3))-6.0*C(K-2)-C(K-4)
      DELC=CCUB-C(K)
C
C     OBTAIN NEXT VALUE OF PERIOD EQUATION
C
 2500 IF(LVRY.GT.0)GO TO 2600
C
      CALL TEQUR(C(K),T(K),MMAX,LCT,HVR,KWRITE,NTRLS,DELT1)
C
      DELT1=DELT1*FACT
```

```
      GO TO 2700
C
 2600 CALL TEQUL(C(K),T(K),MMAX,MODENO,ILAST,KWRITE,NTRLS,WVNOSQ,CSQ,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,DELT1)
C
 2700 ISICH=1
      IF(KWRITE.EQ.0)GO TO 2800
      WRITE(IWR2,600)T(K),C(K),DELT1
C
C     ERROR RETURN FROM RAYGIJ OR LOVAIJ?
C
 2800 IF(NTRLS.LT.1000)GO TO 3500
C
C     ERROR RETURN
C
      GO TO 7200
C
 3500 C(K)=C(K)+DELC
      IF(C(K).GT.0.0)GO TO 3600
      C(K)=(C(K)-DELC)*0.90
C
 3600 IF(LVRY.GT.0)GO TO 3700
C
      CALL TEQUR(C(K),T(K),MMAX,LCT,HVR,KWRITE,NTRLS,DELT2)
C
      DELT2=DELT2*FACT
      GO TO 4000
C
 3700 CALL TEQUL(C(K),T(K),MMAX,MODENO,ILAST,KWRITE,NTRLS,WVNOSQ,CSQ,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,DELT2)
C
 4000 IF(KWRITE.EQ.0)GO TO 4100
      WRITE(IWR2,600)T(K),C(K),DELT2
C
C     ERROR RETURN FROM RAYGIJ OR LOVAIJ?
C
      IF(NTRLS.GE.1000)GO TO 7200
C
 4100 DDT=DELT2-DELT1
      IF(DABS(DDT).GT.1E-70)GO TO 4200
      DELX=-SIGN(ONE,DELC)*DSIGN(ONE,DELT2)*DSIGN(ONE,DDT)
      DELC=ABS(DELC)*0.9*DELX
      GO TO 4300
C
 4200 DELC=-DELC*(DELT2/DDT)
 4300 C(K)=C(K)+DELC
      IF(C(K).GT.0.0)GO TO 4400
      C(K)=(C(K)-DELC)*0.90
 4400 IF((DELT1*DELT2).GE.0.0)GO TO 4500
      ISICH=0
 4500 DELT1=DELT2
      IF(C(K).GT.BTAMX)GO TO 7000
      IF(ISICH.EQ.0)GO TO 4900
      IF(NTRLS.GT.10)GO TO 4900
      IF(ABS(DELC).LT.ACC)DELC=0.0001*SIGN(ONE,DELC)
      GO TO 5000
C
C     IS INCREMENT IN PHASE VELOCITY WITHIN DESIRED ACCURACY?
```

```
C
 4900 IF(ACC.GE.ABS(DELC))GO TO 5200
 5000 NTRLS=NTRLS+1
      IF(NTRLS.LE.NTRLMX)GO TO 3600
C
      WRITE(IWR2,5100)NTRLMX,MODENO,T(K),C(K)
 5100 FORMAT(/,10X,'RESIGNED AFTER ',I3,' TRIALS FOR MODE ',I4,/,
     * 10X,'PERIOD ',F7.3,' AND PHASE VELOCITY ',F7.3,/)
      GO TO 7200
C
 5200 IF(DABS(DELT2).LE.10.0)GO TO 5300
      IF(ISICH.LE.0)GO TO 5300
      DELC=SIGN(ONE,DELC)*0.01
      GO TO 5000
C
 5300 NTRLS=0
      IF(K2CT.GT.0)GO TO 1500
      IF(LVRY.GT.0)GO TO 5500
C
C     STORE SURFACE ELLIPTICITY FOR RAYLEIGH WAVES
C
      HVRAT(K)=HVR
C
      GO TO 6000
C
C     FINAL LOVE WAVE COMPUTATIONS
C
 5500 IF(K.NE.1)GO TO 5800
      WRITE(IWR2,5600)
 5600 FORMAT(/,'++ INPUT: ','IPARMO',3X,/)
      READ(IRE,*,END=9000)IPARMO
C
C     IPARMO - LIST PARTICLE MOTION FLAG
C              = 0 DON'T LIST
C              = 1 LIST
C     IPDER - COMPUTE AND LIST PARTIAL DERIVATIVES FLAG
C             = 0 DON'T COMPUTE AND LIST
C             = 1 COMPUTE AND LIST DC/DH AND DU/DH
C             = 3 COMPUTE AND LIST DC/DBETA AND DU/DBETA
C             = 4 COMPUTE AND LIST DC/DRHO AND DU/DRHO
C
      WRITE(IWR,5700)IPARMO
      WRITE(IWR2,5700)IPARMO
 5700 FORMAT(/,10X,'IPARMO = ',I2,5X,/)
C
C     MAKE LAST PASS TO COMPUTE PARTIALS AND ENERGY INTEGRALS
C     FOR LOVE WAVES
C
 5800 ILAST=1
      IPDER=0
C
      CALL TEQUL(C(K),T(K),MMAX,MODENO,ILAST,KWRITE,NTRLS,WVNOSQ,CSQ,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,DELT2)
C
      ILAST=0
C
      U(K)=FI1L/(C(K)*FI0L)
      AMED(K)=0.5/FI1L
```

```
      ENGK(K)=WVNOSQ*CSQ*FI0L*0.5
      ENGP(K)=(WVNOSQ*FI1L+FI2L)*0.5
C
C     INCREMENT TO NEXT PERIOD VALUE FOR RAYLEIGH AND LOVE WAVES
C
 6000 K=K+1
      IF(K-KMAX)1500,1500,7200
C
 7000 WRITE(IWR2,7100)C(K)
 7100 FORMAT(/,10X,'PHASE VELOCITY OF ',F7.3,' IS GREATER THAN ',
     * 'LARGEST',/,10X,'SHEAR VELOCITY OF MODEL',/)
 7200 KLAST=K-1
      GO TO 9000
C
 7300 WRITE(IWR2,7400)C(K)
 7400 FORMAT(/,10X,'PHASE VELOCITY OF ',F8.3,' IS LESS THAN ZERO',/)
      GO TO 7200
C
 9000 RETURN
      END
```

```
      SUBROUTINE STABIT(STATUS,ISTAT)
C
C*********************************************************************
C
C     DECODES THE GIVEN STATUS VARIABLE AND RETURNS THE FOLLOWING
C     INFORMATION IN ARRAY ISTAT.
C
C     STATUS - DATA WORD CONTAINING STATUS BITS - INPUT
C     ISTAT - STATUS ARRAY - OUTPUT
C             ISTAT(1) : BYTE 10 - BIT 0 - MISSING DATA
C             ISTAT(2) :         - BIT 1 - Z SATURATED OR CLIPPED
C                                          (SET)
C             ISTAT(3) :         - BIT 2 - N-S SATURATED OR CLIPPED
C                                          (SET)
C             ISTAT(4) :         - BIT 3 - E-W SATURATED OR CLIPPED
C                                          (SET)
C             ISTAT(5) :         - BIT 4 - EVENT DETECTED (SET)
C             ISTAT(6) :         - BIT 5 - (AVAILABLE)
C             ISTAT(7) :         - BIT 6 - (AVAILABLE)
C             ISTAT(8) :         - BIT 7 - CALIBRATION ON (SET)
C             ISTAT(9) : BYTE 15 - BIT 7 - E-W CHANNEL RSTN
C                                          SHORT-PERIOD SEISMOMETER
C             ISTAT(10):         - BIT 6 - N-S CHANNEL RSTN
C                                          SHORT-PERIOD SEISMOMETER
C             ISTAT(11):         - BIT 5 - Z CHANNEL RSTN
C                                          SHORT-PERIOD SEISMOMETER
C                                A SET BIT INDICATES THE KS 36000.
C                                A UNSET BIT INDICATES THE S750.
C
C*********************************************************************
C
      DIMENSION ISTAT(11)
C
      INTEGER*4 TTWO
C
      DATA BIG/10000000000./
C
      ISTATS=(STATUS-BIG)*1.E-6+.5
      TTWO=1024
C
      DO 10 I=1,11
      ISTAT(I)=ISTATS/TTWO
      ISTATS=ISTATS-ISTAT(I)*TTWO
      TTWO=TTWO/2
   10 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE STORLT
C
C*****************************************************************************
C
C     LIST STORAGE PARAMETERS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      WRITE(IWR,100)
      WRITE(IWR2,100)
  100 FORMAT(/,10X,'STORAGE PARAMETERS',/)
      WRITE(IWR,200)NCHLS,LENG,NACHLS,IDEM,IDEM1
      WRITE(IWR2,200)NCHLS,LENG,NACHLS,IDEM,IDEM1
  200 FORMAT(/,10X,'NCHLS = ',I4,5X,'LENG = ',I6,5X,'NACHLS = ',
     * I4,//,10X,'IDEM = ',I6,5X,'IDEM1 = ',I6,/)
C
      RETURN
      END
```

```
      SUBROUTINE STRPX
C
C*****************************************************************
C
C     EXECUTIVE ROUTINE FOR SETTING THE RESPONSE FOR A CHANNEL FROM
C     THE MASTER FILE OF RESPONSES
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 13, 1994  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/RESPON/SA0,KNP,SPOLES,KNZ,SZEROS,KFLAG
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION SPOLES(30),SZEROS(20)
C
      COMPLEX RPOLES,RZEROS
      COMPLEX SPOLES,SZEROS
C
C     READ CHANNEL NUMBER
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL TO SET RESPONSE FROM MASTER FILE
C             = 0 SET TO 1
C
      IF(ICHAN.LE.0)ICHAN=1
      WRITE(IWR,200)ICHAN
      WRITE(IWR2,200)ICHAN
  200 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
C     OBTAIN MASTER RESPONSE
C
      CALL SETRES
C
C     SET UP MASTER RESPONSE IN CHANNEL HEADER
C
  800 JNP(ICHAN)=KNP
      JNZ(ICHAN)=KNZ
      JFLAG(ICHAN)=KFLAG
      RA0(ICHAN)=SA0
C
      DO 900 I=1,KNP
      RPOLES(I,ICHAN)=SPOLES(I)
  900 CONTINUE
```

```
C
      DO 1000 I=1,KNZ
      RZEROS(I,ICHAN)=SZEROS(I)
 1000 CONTINUE
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'ERROR IN INPUT PARAMETERS',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'RESPONSE NOT SET UP',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)ICHAN
      WRITE(IWR2,8100)ICHAN
 8100 FORMAT(/,10X,'MASTER RESPONSE SET UP IN CHANNEL ',I4,/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE TAP(DATA,ISAMP,PERCNT)
C
C*********************************************************************
C
C     TAPER ENDS OF ALL DATA CHANNELS
C
C     DATA - DATA ARRAY - INPUT AND OUTPUT
C     ISAMP - NUMBER OF POINTS IN TIME SERIES - INPUT
C     PERCNT - PERCENT OF TIME SERIES TO TAPER - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 28, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION DATA(1)
C
      FSAMP=ISAMP
      MTER=FSAMP*PERCNT/100.0
      IF(MTER.LE.0)GO TO 9000
      ARG=PI/(2.0*MTER)
C
C     CHECK FOR LEADING ZEROES
C
      IFP=1
C
      DO 50 J=1,ISAMP
      IF(DATA(J).NE.0.0)GO TO 60
      IFP=J+1
   50 CONTINUE
C
   60 IF(IFP.LE.3)IFP=1
      ILP=IFP+MTER-1
C
C     TAPER START OF TIME SERIES
C
      DO 100 J=IFP,ILP
      FJ=J-IFP
      DATA(J)=DATA(J)*(1.0-COS(FJ*ARG))
  100 CONTINUE
C
C     CHECK FOR TRAILING ZEROES
C
      ILP=ISAMP
C
      DO 150 J=1,ISAMP
      K=ISAMP-J+1
      IF(DATA(K).NE.0.0)GO TO 180
      ILP=K-1
  150 CONTINUE
C
  180 IFP=ILP-MTER+1
C
C     TAPER END OF TIME SERIES
```

```
C
      DO 200 J=IFP,ILP
      FJ=ILP-J
      DATA(J)=DATA(J)*(1.0-COS(FJ*ARG))
  200 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE TAPEX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR TAPERING DATA
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 19, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1)
C
      COMPLEX RPOLES,RZEROS
C
C     READ TAPER PARAMETERS
C
      WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','ICHAN',3X,'PERCNT',/)
      READ(IRE,*,END=9000)ICHAN,PERCNT
C
C     ICHAN - DESIRED CHANNEL
C             = 0 TAPER ALL CHANNELS
C     PERCNT - PERCNT OF TIME SERIES TO TAPER
C
      WRITE(IWR,100)
      WRITE(IWR2,100)
  100 FORMAT(/,10X,'TAPER PARAMETERS',/)
      WRITE(IWR,200)ICHAN,PERCNT
      WRITE(IWR2,200)ICHAN,PERCNT
  200 FORMAT(/,10X,'TAPER CHANNEL ',I4,' BY ',F7.2,' PERCENT',/)
C
      IF(PERCNT.LE.0.0)GO TO 7000
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      IF(ICHAN.LE.0)GO TO 500
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 2000
C
  500 NCH1=1
      NCH2=NACHLS
C
 2000 DO 3000 I=NCH1,NCH2
      ISAMP=JSAMP(I)
      INDEX=(I-1)*LENG+1
C
```

```
      CALL TAP(DATA(INDEX),ISAMP,PERCNT)
C
 3000 CONTINUE
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT TAPERED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY TAPERED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE TELIP(ELAT,SLAT,DIST,TC)
C
C*****************************************************************
C
C     DETERMINE CORRECTION TO P TRAVEL TIME DUE TO EARTH'S ELLIPTICITY
C
C     ELAT - EPICENTER GEOCENTRIC CO-LATITUDE (IN DEGREES) - INPUT
C     SLAT - STATION GEOCENTRIC CO-LATITUDE (IN DEGREES) - INPUT
C     DIST - DISTANCE (IN DEGREES) - INPUT
C     TC - TIME CORRECTION TO P TIME DUE TO ELLIPTICITY (TO BE
C          ADDED TO TOTAL TRAVEL TIME) (IN SECONDS) - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 27, 1993  COLORADO
C
C*****************************************************************
C
      CALL HELIP(SLAT,HS)
      CALL HELIP(ELAT,HE)
      CALL FELIP(DIST,F)
C
      TC=F*(HE+HS)
C
      RETURN
      END
```

```
      SUBROUTINE TEQUL(PHASE,PERIOD,MMAX,MODENO,ILAST,KWRITE,NTRLS,
     * WVNOSQ,CSQ,IPARMO,IPDER,FI0L,FI1L,FI2L,TEQULL)
C
C*****************************************************************
C
C     DETERMINE VALUE OF PERIOD EQUATION FOR LOVE WAVES
C
C     PHASE - TRIAL PHASE VELOCITY VALUE (IN KM/S) - INPUT
C     PERIOD - PERIOD OF RAYLEIGH WAVE (IN SECONDS) - INPUT
C     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     MODENO - MODE FLAG - INPUT
C              = 0 FUNDAMENTAL
C              = 1 FIRST HIGHER MODE
C              = 2 SECOND HIGHER MODE
C              = n  N'TH HIGHER MODE
C     ILAST - LAST ITERATION FLAG - INPUT
C             = 0 NOT LAST ITERATION
C             = 1 LAST ITERATION
C     KWRITE - DEBUG LISTING FLAG - INPUT
C              = 0 NO LISTING MADE
C              = 1 LISTING MADE OF ROOT SEARCH
C              = 2 LISTING MADE OF ROOT SEARCH AND LAYER DIVISION
C     NTRLS - NUMBER OF TRIALS TO FIND A ROOT - INPUT AND OUTPUT
C             = 1000 ERROR ENCOUNTERED
C     WVNOSQ - WAVE NUMBER SQUARED - OUTPUT
C     CSQ - PHASE VELOCITY SQUARED - OUTPUT
C     IPARMO - LIST PARTICLE MOTION FLAG - INPUT
C              = 0 DON'T LIST
C              = 1 LIST
C     IPDER - COMPUTE AND LIST PARTIAL DERIVATIVES FLAG - INPUT
C             = 0 DON'T COMPUTE AND LIST
C             = 1 COMPUTE AND LIST DC/DH AND DU/DH
C             = 3 COMPUTE AND LIST DC/DBETA AND DU/DBETA
C             = 4 COMPUTE AND LIST DC/DRHO AND DU/DRHO
C     FI0L - ENERGY INTEGRAL 0 - OUTPUT
C     FI1L - ENERGY INTEGRAL 1 - OUTPUT
C     FI2L - ENERGY INTEGRAL 2 - OUTPUT
C     TEQULL - VALUE OF PERIOD EQUATION - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 14, 1994  COLORADO
C
C*****************************************************************
C
      COMMON/IN/BX
C
      DIMENSION V(1000,4),A(2,2)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(40001),V(1,1)),(BX(50801),A(1,1))
C
      REAL*8 A,R11,R21,RR11,RR21,TEQULL,ARGK,TEQUL1,TEQUL2,DIF
C
      CCC=PHASE
      TTT=PERIOD
      WVNO=6.283185308/(CCC*TTT)
      WVNOSQ=WVNO*WVNO
```

```
      CSQ=CCC*CCC
      MODE=MODENO
      TEQUL1=0.0
      TEQUL2=0.0
      LCT=0
C
      M=1
      ARGK=CSQ/(V(M,3)*V(M,3))
      ARGK=ARGK-1.0
      MM=M
C
C     COMPUTE HASKELL MATRIX ELEMENTS FOR FIRST LAYER
C
      CALL LOVAIJ(ARGK,WVNO,TTT,MM,KWRITE,MMAX,NTRLS,ILAST,LCT)
C
      M=MM
      R11=A(1,1)
      R21=A(2,1)
C
      IF(ILAST.LE.0)GO TO 100
C
C     COMPUTE ENERGY INTEGRALS, GROUP VELOCITY, AND DERIVATIVES
C
      MM=M
C
      CALL ENGL(TTT,CCC,R11,R21,MODENO,MMAX,WVNO,WVNOSQ,ARGK,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,MM,LCT,KWRITE)
C
      M=MM
C
      GO TO 300
C
  100 IF(R11.GE.0.0)GO TO 300
  200 MODE=MODE-1
      IF(MODE.LT.0)GO TO 1000
C
  300 M=M+1
C
C     IF SHEAR VELOCITY ZERO, STOP PROCESSING LAYERS
C
      IF(V(M,3).GT.0.0)GO TO 400
      MMAX=M
      GO TO 500
C
  400 ARGK=CSQ/(V(M,3)*V(M,3))
      ARGK=ARGK-1.0
C
  500 MM=M
C
C     COMPUTE HASKELL MATRIX ELEMENTS
C
      CALL LOVAIJ(ARGK,WVNO,TTT,MM,KWRITE,MMAX,NTRLS,ILAST,LCT)
C
      M=MM
C
C     CHECK FOR HALF-SPACE
C
      IF(M.GE.MMAX)GO TO 1000
```

```
C
  600 RR11=A(1,1)*R11+A(1,2)*R21
      RR21=A(2,1)*R11+A(2,2)*R21
C
      IF(ILAST.LE.0)GO TO 700
      R11=RR11
      R21=RR21
C
C     COMPUTE ENERGY INTEGRALS
C
      MM=M
C
      CALL ENGL(TTT,CCC,R11,R21,MODENO,MMAX,WVNO,WVNOSQ,ARGK,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,MM,LCT,KWRITE)
C
      M=MM
      GO TO 800
C
C     CHECK SIGN CHANGE OF PARTICLE MOTION AGAINST DESIRED MODE
C
  700 IF((R11*RR11).GE.0.0)GO TO 800
      MODE=MODE-1
      IF(MODE.LT.0)GO TO 1000
C
C     CHECK SIZE OF PARTICLE MOTION
C
  800 IF(DABS(RR11).GT.10000.0)GO TO 1000
C
      IF(KWRITE.NE.2)GO TO 850
      WRITE(IWR2,820)RR11,RR21
  820 FORMAT(10X,'RR11 = ',D15.9,5X,'RR21 = ',D15.9)
C
  850 TEQUL2=RR21-A(1,1)*RR11
      IF(M.LE.3)GO TO 880
      DIF=DABS((TEQUL1-TEQUL2)/TEQUL1)
C
      IF(KWRITE.NE.2)GO TO 870
      WRITE(IWR2,860)TEQUL1,TEQUL2,DIF
  860 FORMAT(10X,D15.9,4X,D15.9,4X,'DIF = 'D15.9)
C
  870 IF(DIF.LT.1E-7)GO TO 1000
  880 TEQUL1=TEQUL2
C
      R11=RR11
      R21=RR21
C
C     CHECK IF LAYERS HAVE BEEN DIVIDED
C
  900 IF(LCT.LT.2)GO TO 300
      LCT=LCT-1
      GO TO 600
C
C     TEQULL IS THE PERIOD EQUATION FOR LOVE WAVES
C
 1000 TEQULL=R21-A(1,1)*R11
C
C     THIS IS HASKELL'S EQUATION 9.9
C
```

```
      IF(ILAST.LE.0)GO TO 9000
C
C     COMPUTE ENERGY INTEGRALS AND PARTIAL DERIVATIVES AFTER ROOT
C     HAS BEEN FOUND
C
      MM=M
C
      CALL ENGL(TTT,CCC,R11,R21,MODENO,MMAX,WVNO,WVNOSQ,ARGK,
     * IPARMO,IPDER,FI0L,FI1L,FI2L,MM,LCT,KWRITE)
C
      M=MM
C
 9000 RETURN
      END
```

```
      SUBROUTINE TEQUR(PHASE,PERIOD,MMAX,LCT,HVR,KWRITE,NTRLS,TEQURR)
C
C*****************************************************************
C
C     DETERMINE VALUE OF PERIOD EQUATION FOR RAYLEIGH WAVES
C
C     PHASE - TRIAL PHASE VELOCITY VALUE (IN KM/S) - INPUT
C     PERIOD - PERIOD OF RAYLEIGH WAVE (IN SECONDS) - INPUT
C     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     LCT - NUMBER OF SUBDIVISIONS TO THE CURRENT LAYER - OUTPUT
C     HVR - SURFACE ELLIPTICITY (HORIZONTAL/VERTICAL) - OUTPUT
C     KWRITE - DEBUG LISTING FLAG - INPUT
C              = 0 NO LISTING MADE
C              = 1 LISTING MADE OF ROOT SEARCH
C              = 2 LISTING MADE OF ROOT SEARCH AND LAYER DIVISION
C     NTRLS - NUMBER OF TRIALS TO FIND A ROOT - INPUT AND OUTPUT
C             = 1000 ERROR ENCOUNTERED
C     TEQURR - VALUE OF PERIOD EQUATION - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 19, 1994  COLORADO
C
C*****************************************************************
C
      COMMON/IN/BX
C
      DIMENSION P1(6),PP1(6),R12(6),R14(6),RR12(6),RR14(6),G(6,6),
     * V(1000,4)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(40001),V(1,1)),(BX(50001),P1(1))
      EQUIVALENCE (BX(50101),PP1(1)),(BX(50201),R12(1))
      EQUIVALENCE (BX(50301),R14(1)),(BX(50401),RR12(1))
      EQUIVALENCE (BX(50501),RR14(1)),(BX(50601),G(1,1))
C
      REAL*8 G,P1,R12,R14,PP1,RR12,RR14,TEQURR,ARGH,ARGK,TEQUR1,
     * TEQUR2,DIF
C
      IFST=1
      CCC=PHASE
      TTT=PERIOD
      WVNO=6.283185308/(CCC*TTT)
      WVNOSQ=WVNO*WVNO
      CSQ=CCC*CCC
      TEQUR1=0.0
      TEQUR2=0.0
C
      M=1
      GO TO 400
C
C     INCREMENT TRUE LAYER INDEX
C
  300 M=M+1
C
  400 ARGH=CSQ/(V(M,2)*V(M,2))
      ARGH=1.0-ARGH
      ARGK=0.0
```

```
      IF(V(M,3).LE.0.0)GO TO 500
      ARGK=CSQ/(V(M,3)*V(M,3))
      ARGK=1.0-ARGK
C
C     DETERMINE SUBMATRIX ELEMENTS
C
  500 MM=M
C
      CALL RAYGIJ(ARGH,ARGK,WVNO,TTT,CSQ,MM,KWRITE,MMAX,NTRLS,LCT)
C
      M=MM
C
C     CHECK IF FIRST LAYER
C
      IF(IFST.EQ.0)GO TO 700
C
C     SET UP PRODUCT FOR FIRST LAYER
C
      DO 600 I=1,6
      P1(I)=G(I,1)
      R12(I)=G(I,2)
      R14(I)=G(I,4)
  600 CONTINUE
C
      IFST=0
      GO TO 300
C
C     INITIALIZE PRODUCT ELEMENTS
C
  700 DO 800 I=1,6
      PP1(I)=0.0
      RR12(I)=0.0
      RR14(I)=0.0
  800 CONTINUE
C
C     COMPUTE PRODUCT ELEMENTS
C
      DO 1000 I=1,6
      DO 900 J=1,6
      PP1(I)=P1(J)*G(I,J)+PP1(I)
      RR12(I)=R12(J)*G(I,J)+RR12(I)
      RR14(I)=R14(J)*G(I,J)+RR14(I)
  900 CONTINUE
C
C     CHECK FOR HALF-SPACE
C
      IF(M.GE.MMAX)GO TO 1200
 1000 CONTINUE
C
 1050 TEQUR2=PP1(1)
      IF(M.LE.3)GO TO 1080
      DIF=DABS((TEQUR1-TEQUR2)/TEQUR1)
C
      IF(KWRITE.NE.2)GO TO 1070
      WRITE(IWR2,1060)TEQUR1,TEQUR2,DIF
 1060 FORMAT(10X,D15.9,3X,D15.9,3X,'DIF = ',D15.9)
C
 1070 IF(DIF.LT.1E-5)GO TO 1200
```

```
 1080 TEQUR1=TEQUR2
C
 1090 DO 1100 I=1,6
      P1(I)=PP1(I)
      R12(I)=RR12(I)
      R14(I)=RR14(I)
 1100 CONTINUE
C
C     CHECK IF LAYERS HAVE BEEN DIVIDED
C
      IF(LCT.LT.2)GO TO 300
C
C     FORM PRODUCT OF ALL SUBDIVIDED LAYER SUBMATRICES
C
      LCT=LCT-1
      GO TO 700
C
C     TEQURR IS THE PERIOD EQUATION FOR RAYLEIGH WAVES
C
 1200 TEQURR=PP1(1)
C
C     COMPUTE SURFACE ELLIPICITY - HORIZONTAL TO VERTICAL RATIO
C
      IF(V(1,3).GT.0.0)GO TO 1500
      HVR=0.0
      GO TO 9000
C
 1500 HVR=-(RR14(1)/RR12(1))
C
 9000 RETURN
      END
```

```
      SUBROUTINE TESTF
C
C*********************************************************************
C
C     GENERAL PURPOSE TESTING ROUTINE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 17, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/CAL/XXCAL
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
      COMMON/DTAZ/DELTA,DISKM,AZSE,AZES,CTT,ATT,DTT
      COMMON/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/OPTION/IOPS(1)
      COMMON/PCKF/ICHECK,IPLFL
      COMMON/RESPON/SA0,KNP,SPOLES,KNZ,SZEROS,KFLAG
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DELTA(100),DISKM(100),AZSE(100),AZES(100),CTT(100),
     * ATT(100),DTT(100)
      DIMENSION EXFLAG(10)
      DIMENSION SPOLES(30),SZEROS(20)
      DIMENSION DATA(1),BX(1)
C
      COMPLEX RPOLES,RZEROS
      COMPLEX SPOLES,SZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
C
C
C
C
 9000 RETURN
      END
```

```
      SUBROUTINE TIMC(TSEC,IHR,MIN,SEC)
C
C*****************************************************************************
C
C     CALCULATE TOTAL TIME IN SECONDS
C
C     TSEC - TOTAL TIME (IN SECONDS) - OUTPUT
C     IHR - HOURS - INPUT
C     MIN - MINUTES - INPUT
C     SEC - SECONDS - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 28, 1993  COLORADO
C
C*****************************************************************************
C
      TSEC=((IHR*60)+MIN)*60+SEC
C
      RETURN
      END
```

```
      FUNCTION TIMLPP(DISTX,DPTHX)
C
C*********************************************************************
C
C     COMPUTE THE TRAVEL TIME OF THE PHASE pP FOR A GIVEN
C     DISTANCE AND DEPTH
C
C     DISTX - DISTANCE (IN DEGREES) - INPUT
C     DPTHX - DEPTH (IN KILOMETERS) - INPUT
C     TIMLPP - TRAVEL TIME OF pP (IN SECONDS) - OUTPUT
C
C     WRITTEN BY GEOTECH ON CONTRACT TO U.S. GOVERNMENT
C
C     JANUARY 2, 1993  DALLAS
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     SEPTEMBER 3, 1993  COLORADO
C
C*********************************************************************
C
      TIMLPP=0.0
      DIST=DISTX
      DPTH=DPTHX
C
      IF(DPTH.LT.0.0)DPTH=0.0
      IF(DIST.LT.0.0)DIST=0.0
      IF(DPTH.GT.700.0)DPTH=700.0
      IF(DIST.GT.180.0)DIST=180.0
      IF(DPTH.LT.15.0)GO TO 40
      IF(DIST.GE.47.0.AND.DIST.LE.99.0)GO TO 30
      IF(DIST.GE.(17.0+(DPTH*30.0)/585.0).AND.DIST.LT.47.0)GO TO 30
      GO TO 40
C
C     EVALUATE POLYNOMIAL TO OBTAIN SURFACE REFLECTION POINT
C     DISTANCE X1 FROM EPICENTER
C
   30 X1=3.837437441E-01
     *  -(2.077023938E-02)*DIST
     *  +(1.077586036E-02)*DPTH
     *  +(4.286477915E-04)*DIST*DIST
     *  -(2.517302465E-04)*DIST*DPTH
     *  +(2.073143175E-05)*DPTH*DPTH
     *  -(2.731592749E-06)*DIST*DIST*DIST
     *  +(1.817303722E-06)*DIST*DIST*DPTH
     *  -(2.012145067E-07)*DIST*DPTH*DPTH
C
C     X2 IS DISTANCE FROM THE SURFACE REFLECTION POINT TO THE STATION
C
      X2=DIST-X1
C
C     FORM SUM OF TRAVEL TIME FROM HYPOCENTER TO SURFACE AND FROM
C     SURFACE TO SURFACE
C
      TIMLPP=PASTIM(1,X1,DPTH)+PASTIM(1,X2,0.0)
C
      GO TO 9000
```

```
C
   40 IF(DPTH.LE.30.0)TIMLPP=DPTH/6.0
C
      IF(DPTH.GT.30.0.AND.DPTH.LE.710.0)TIMLPP=DPTH/9.0
C
      TIMLPP=TIMLPP+PASTIM(1,DIST,0.0)
C
 9000 RETURN
      END
```

```
      FUNCTION TRVLTM(DELT,DEPH,NPHAS,ELAT,SLAT,ELEV)
C
C*********************************************************************
C
C     DETERMINE TRAVEL TIME FOR BODY WAVE PHASES
C
C     DELT - DISTANCE (IN DEGREES) - INPUT
C     DEPH - HYPOCENTER DEPTH (IN KILOMETERS) - INPUT
C     NPHAS - PHASE NAME - INPUT
C
C             Distance Range for Phase
C
C                 Pg    0   - 15
C                 ScP   0   - 70
C                 PcP   0   - 150
C                 pP    0.3 - 180
C                 sP    0.3 - 180
C                 PP    0.3 - 180
C                 P     0   - 180
C                 PPP   0   - 180
C                 SP    0   - 140
C                 SPP   0   - 180
C                 SKP1  70  - 180
C                 SKP2  70  - 180
C                 PKP   100 - 180
C                 PKP2  100 - 180
C                 PKKP  130 - 180
C                 PKK2  150 - 180
C                 PKK3  150 - 180
C                 PKPP  0   - 180
C                 PcS   0   - 70
C                 ScS   0   - 100
C                 SKS   60  - 180
C                 SKS2  60  - 180
C                 S     0   - 180
C                 pS    0   - 180
C                 sS    0   - 180
C                 PS    0   - 140
C                 PPS   0   - 180
C                 SS    0   - 180
C                 SSS   0   - 180
C
C     ELAT - EPICENTER GEOCENTRIC CO-LATITUDE (IN DEGREES) - INPUT
C     SLAT - STATION GEOCENTRIC CO-LATITUDE (IN DEGREES) - INPUT
C     ELEV - STATION ELEVATION (IN KILOMETERS) - INPUT
C     TRVLTM - TRAVEL TIME (IN SECONDS) - OUTPUT
C              = 0 IF OUTSIDE DISTANCE OR DEPTH RANGE FOR PHASE
C
C     WRITTEN BY GEOTECH ON CONTRACT TO U.S. GOVERNMENT
C
C     JANUARY 2, 1968  DALLAS
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     SEPTEMBER 2, 1993  COLORADO
C
C*********************************************************************
```

```
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION A1(14,7),A2(11,5),B1(14),B2(14),B3(14),B4(14),B5(14),
     * B6(14),B7(14),C1(11),C2(11),C3(11),C4(11),C5(11)
      DIMENSION NAMPHA(29)
C
      EQUIVALENCE (A1(1,1),B1(1)),(A1(1,2),B2(1)),(A1(1,3),B3(1))
      EQUIVALENCE (A1(1,4),B4(1)),(A1(1,5),B5(1)),(A1(1,6),B6(1))
      EQUIVALENCE (A1(1,7),B7(1)),(A2(1,1),C1(1)),(A2(1,2),C2(1))
      EQUIVALENCE (A2(1,3),C3(1)),(A2(1,4),C4(1)),(A2(1,5),C5(1))
C
      DATA NAMPHA/'Pg  ','ScP ','PcP ','pP  ','sP  ','PP  ','P   ',
     * 'PPP ','SP  ','SPP ','SKP1','SKP2','PKP ','PKP2','PKKP','PKK2',
     * 'PKK3','PKPP','PcS ','ScS ','SKS ','SKS2','S   ','pS  ','sS  ',
     * 'PS  ','PPS ','SS  ','SSS '/
C
      DATA B1/
     *  0.725466553E 03,  0.514220459E 03,  0.140201880E 04,
     *  0.127383594E 04,  0.127080103E 04,  0.200144141E 04,
     *  0.190918311E 04,  0.223226294E 04,  0.228185010E 04,
     *  0.242367993E 04,  0.725180664E 03,  0.935882568E 03,
     *  0.358082764E 03,  0.154438672E 04/
C
      DATA B2/
     * -0.208253324E 00, -0.976705551E-01, -0.540286255E 01,
     * -0.507266903E 01, -0.721118546E 01, -0.214745941E 02,
     *  0.918109417E-01, -0.960087109E 01, -0.399716377E 01,
     *  0.608225204E-01, -0.166538775E 00, -0.135327995E 00,
     *  0.187140656E 02, -0.371129704E 01/
C
      DATA B3/
     * -0.244061112E 00, -0.136967599E 00, -0.237183154E 00,
     * -0.240536392E 00, -0.128715932E 00, -0.130626976E 00,
     * -0.135048985E 00,  0.132529914E 00, -0.138018012E 00,
     * -0.135096133E 00, -0.136979997E 00, -0.244272411E 00,
     * -0.214915633E 00, -0.231045544E 00/
C
      DATA B4/
     *  0.760856867E-01,  0.549451187E-01,  0.624160878E-01,
     *  0.600765534E-01,  0.748699903E-01,  0.152351737E 00,
     * -0.144603997E-01,  0.886546373E-01, -0.957952440E-02,
     * -0.180495381E-01,  0.749689937E-01,  0.995805860E-01,
     * -0.919871330E-01,  0.499149263E-01/
C
      DATA B5/
     *  0.245971140E-03,  0.127155377E-03, -0.161161879E-04,
     *  0.252875761E-04, -0.346252345E-04, -0.547394663E-08,
     *  0.164725498E-04,  0.752431447E-06,  0.316956866E-04,
     *  0.152682042E-04,  0.135319438E-03,  0.237455330E-03,
     * -0.202409705E-03, -0.580301712E-04/
C
      DATA B6/
     *  0.543648930E-04,  0.322706037E-04,  0.512616534E-04,
     *  0.534502178E-04,  0.304103451E-04,  0.358167308E-04,
     *  0.302378758E-04,  0.303376291E-04,  0.299713865E-04,
     *  0.302038388E-04,  0.320242834E-04,  0.549107499E-04,
     *  0.532945269E-04,  0.521004404E-04/
```

```
C
      DATA B7/
     * -0.417994335E-03, -0.231887490E-03, -0.176572707E-03,
     * -0.124795668E-03, -0.203528529E-03, -0.298742438E-03,
     *  0.339665130E-04, -0.355799450E-03,  0.463075121E-04,
     *  0.540278270E-04, -0.408720458E-03, -0.407973304E-03,
     *  0.156304595E-03, -0.147563173E-03/
C
      DATA C1/
     * -0.373113491E-01, -0.616451502E 00, -0.271883678E 01,
     *  0.133242245E 02,  0.211756744E 02,  0.509404087E 01,
     *  0.238003373E 00, -0.282065125E 02, -0.490024567E 01,
     * -0.542202652E 00, -0.261333084E 01/
C
      DATA C2/
     * -0.387731530E-01,  0.513308227E 00,  0.394649744E 00,
     *  0.671006083E 00,  0.476140738E 00, -0.111201286E 00,
     * -0.263838731E-01,  0.589427292E 00,  0.278056383E 00,
     *  0.510750115E 00,  0.402935207E 00/
C
      DATA C3/
     *  0.944281742E-02, -0.778818503E-02, -0.150423311E-01,
     * -0.543052703E-02, -0.682960451E-02,  0.220050178E-01,
     *  0.169843696E-01, -0.116032399E-01, -0.226074122E-01,
     * -0.762423128E-02, -0.969219953E-02/
C
      DATA C4/
     *  0.457515940E-03, -0.724263518E-04, -0.273073558E-03,
     * -0.290501863E-03, -0.139695840E-03,  0.635795994E-03,
     *  0.238381312E-03, -0.678342534E-03,  0.103666983E-02,
     * -0.513249106E-04, -0.285896473E-03/
C
      DATA C5/
     * -0.875224941E-04,  0.382494327E-04,  0.608550909E-04,
     *  0.308118615E-04,  0.280966633E-04, -0.148782870E-03,
     * -0.130471817E-03, -0.567274242E-06,  0.111776128E-03,
     *  0.339051912E-04,  0.358979014E-04/
C
      IELP=0
      IELV=0
      DIST=DELT
      DEPTH=DEPH
C
C     DETERMINE NUMBER CORRESPONDING TO PHASE NAME
C
   30 DO 50 I=1,29
      NPHNM=I
      IF(NPHAS.EQ.NAMPHA(I))GO TO 90
   50 CONTINUE
C
      NPHNM=7
      WRITE(IWR,60)NPHAS
      WRITE(IWR2,60)NPHAS
   60 FORMAT(/,10X,'***** PHASE ',A4,' UNIDENTIFIED, P PHASE ',
     * 'USED *****',/)
C
   90 NP=NPHNM
C
```

```
      TRVLTM=0.0
C
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,
     * 1300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300,
     * 2400,2500,2600,2700,2800,2900,9000),NP
C
C     PHASE Pg
C
  100 IF(NPHNM.NE.1)GO TO 9000
      IELV=1
      IF(DIST.GT.15.0)GO TO 9000
      X1=SQRT(DIST*DIST+DEPTH*DEPTH)
C
      TRVLTM=X1/6.0
C
      GO TO 8000
C
C     PHASE ScP
C
  200 IND=1
      IELV=1
      IF(DIST.GT.70.0)GO TO 9000
      GO TO 3200
C
C     PHASE PcP
C
  300 IND=2
      IELV=1
      IF(DIST.GT.150.0)GO TO 9000
      GO TO 3200
C
C     PHASE pP
C
  400 IF(DIST.LT.0.3.OR.DEPTH.LT.0.5)GO TO 9000
      IELV=1
C
      TRVLTM=TIMLPP(DIST,DEPTH)
C
      GO TO 8000
C
C     PHASE sP
C
  500 JND=1
      IELV=1
      IF(DIST.LT.0.3.OR.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE PP
C
  600 JND=2
      IELV=1
      IF(DIST.LT.0.3.AND.DEPTH.GT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE P
C
  700 IELV=1
      IELP=1
```

```
      IF(DIST.LT.0.3.AND.DEPTH.GT.0.5)GO TO 9000
C
      TRVLTM=PASTIM(1,DIST,DEPTH)
C
      GO TO 8000
C
C     PHASE PPP
C
  800 JND=3
      IELV=1
      IF(DIST.LT.0.3.AND.DEPTH.GT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE SP
C
  900 JND=4
      IELV=1
      IF(DIST.GT.140.0)GO TO 9000
      GO TO 3400
C
C     PHASE SPP
C
 1000 JND=5
      IELV=1
      IF(DIST.LT.0.3.AND.DEPTH.GT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE SKP1
C
 1100 IND=3
      IELV=1
      IF(DIST.LT.70.0)GO TO 9000
      GO TO 3200
C
C     PHASE SKP2
C
 1200 IND=4
      IELV=1
      IF(DIST.LT.70.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKP
C
 1300 IND=5
      IELV=1
      IF(DIST.LT.100.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKP2
C
 1400 IND=6
      IELV=1
      IF(DIST.LT.100.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKKP1
C
 1500 IND=7
```

```
      IELV=1
      IF(DIST.GT.130.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKKP2
C
 1600 IND=8
      IELV=1
      IF(DIST.GT.150.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKKP3
C
 1700 IND=9
      IELV=1
      IF(DIST.GT.150.0)GO TO 9000
      GO TO 3200
C
C     PHASE PKPPKP
C
 1800 IND=10
      IELV=1
      GO TO 3200
C
C     PHASE PcS
C
 1900 IND=11
      IELV=2
      IF(DIST.GT.70.0)GO TO 9000
      GO TO 3200
C
C     PHASE ScS
C
 2000 IND=12
      IELV=2
      IF(DIST.GT.100.0)GO TO 9000
      GO TO 3200
C
C     PHASE SKS1
C
 2100 IND=13
      IELV=2
      IF(DIST.GT.60.0)GO TO 9000
      GO TO 3200
C
C     PHASE SKS2
C
 2200 IND=14
      IELV=2
      IF(DIST.GT.60.0)GO TO 9000
      GO TO 3200
C
C     PHASE S
C
 2300 IELV=2
      IF(DIST.LT.0.3.AND.DEPTH.LT.0.5)GO TO 9000
C
      TRVLTM=PASTIM(2,DIST,DEPTH)
```

```
C
      GO TO 8000
C
C     PHASE pS
C
 2400 JND=6
      IELV=2
      IF(DIST.LT.0.3.OR.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE sS
C
 2500 JND=7
      IELV=2
      IF(DIST.LT.0.3.OR.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE PS
C
 2600 JND=8
      IELV=2
      IF(DIST.GT.140.0)GO TO 9000
      GO TO 3400
C
C     PHASE PPS
C
 2700 JND=9
      IELV=2
      IF(DIST.LT.0.3.AND.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE SS
C
 2800 JND=10
      IELV=2
      IF(DIST.LT.0.3.AND.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
C     PHASE SSS
C
 2900 JND=11
      IELV=2
      IF(DIST.LT.0.3.AND.DEPTH.LT.0.5)GO TO 9000
      GO TO 3400
C
 3200 DO 3300 K=1,7
C
      TRVLTM=TRVLTM+A1(IND,K)*FUNC(K,DIST,DEPTH)
C
 3300 CONTINUE
C
      GO TO 8000
C
C     DETERMINE REFLECTION POINT DISTANCE X1
C
 3400 X1=0.0
C
      DO 3500 K=1,5
```

```
C
      X1=X1+A2(JND,K)*FUNC(K,DIST,DEPTH)
C
 3500 CONTINUE
C
      IF(X1.LT.0.0)X1=0.0
C
      GO TO (8000,8000,8000,8000,3700,3600,8000,3600,3700,3700,
     * 8000,8000,8000,8000,8000,8000,8000,8000,8000,8000,8000,
     * 8000,8000,3600,3700,3600,3800,3700,3700,8000),NP
C
 3600 TRVLTM=TRVLTM+PASTIM(1,X1,DEPTH)
C
      GO TO 3900
C
 3700 TRVLTM=TRVLTM+PASTIM(2,X1,DEPTH)
C
      GO TO 3900
C
 3800 X0=DIST-X1
C
      TRVLTM=TRVLTM+PASTIM(2,X0,0.0)
C
      NP=6
      DIST=X1
      GO TO 600
C
 3900 GO TO (8000,8000,8000,8000,4000,4000,8000,4100,4000,4100,
     * 8000,8000,8000,8000,8000,8000,8000,8000,8000,8000,8000,
     * 8000,8000,4200,4200,4200,8000,4200,4300,8000),NP
C
 4000 X2=DIST-X1
C
      TRVLTM=TRVLTM+PASTIM(1,X2,0.0)
C
      GO TO 8000
C
 4100 X2=(DIST-X1)/2.0
C
      TRVLTM=TRVLTM+2.0*PASTIM(1,X2,0.0)
C
      GO TO 8000
C
 4200 X2=DIST-X1
C
      TRVLTM=TRVLTM+PASTIM(2,X2,0.0)
C
      GO TO 8000
C
 4300 X2=(DIST-X1)/2.0
C
      TRVLTM=TRVLTM+2.0*PASTIM(2,X2,0.0)
C
C     CHECK FOR ELLIPTICITY CORRECTION
C
 8000 IF(IELP.EQ.0)GO TO 8500
C
      CALL TELIP(ELAT,SLAT,DIST,TC)
```

```
C
      TRVLTM=TRVLTM+TC
C
C     CHECK FOR STATION ELEVATION CORRECTION
C
 8500 IF(IELV.EQ.0)GO TO 9000
      IF(ELEV.LE.0.0)GO TO 9000
      VEL=5.3
      IF(IELV.NE.1)VEL=3.4
      TC=ELEV/VEL
C
      TRVLTM=TRVLTM+TC
C
 9000 RETURN
      END
```

```
      SUBROUTINE TTCOMP(SLAT,SLON,SELEV,ELAT,ELON,EDEPTH,NPHAS,
     * DELTA,DISKM,AZSE,AZES,TT,UVEL,PERIOD,NPER,IDF)
C
C*********************************************************************
C
C     DETERMINE BODY WAVE TRAVEL TIME FOR A STATION-EPICENTER PAIR
C     AND FOR A GIVEN PHASE
C
C     SLAT - STATION GEOGRAPHIC LATITUDE (IN DEGREES) - INPUT
C            - = S
C     SLON - STATION GEOGRAPHIC LONGITUDE (IN DEGREES) - INPUT
C            - = W
C     SELEV - STATION ELEVATION (IN KILOMETERS) - INPUT
C     ELAT - EPICENTER GEOGRAPHIC LATITUDE (IN DEGREES) - INPUT
C            - = S
C     ELON - EPICENTER GEOGRAPHIC LONGITUDE (IN DEGREES) - INPUT
C            - = W
C     EDEPTH - HYPOCENTER DEPTH (IN KILOMETERS) - INPUT
C     NPHAS - DESIRED PHASE FOR TT CALCULATION - INPUT
C     DELTA - DISTANCE (IN DEGREES) - OUTPUT
C     DISKM - DISTANCE (IN KILOMETERS) - OUTPUT
C     AZSE - AZIMUTH FROM STATION-TO-EPICENTER (IN DEGREES) - OUTPUT
C     AZES - AZIMUTH FROM EPICENTER-TO-STATION (IN DEGREES) - OUTPUT
C     TT - BODY OR SURFACE WAVE TRAVEL TIME (IN SECONDS) - OUTPUT
C          = 0 IF OUTSIDE DISTANCE OR DEPTH RANGE FOR PHASE
C     UVEL - SURFACE WAVE GROUP VELOCITY (IN KM/S) - INPUT OR OUTPUT
C     PERIOD - SURFACE WAVE PERIOD (IN SECONDS) - INPUT
C     NPER - NUMBER OF PERIOD VALUES - INPUT
C            = 0 THEN BODY WAVE TRAVEL TIME COMPUTED
C     IDF - DISTANCE FLAG
C           = 0 COMPUTE DISTANCE
C           = 1 DON'T COMPUTE DISTANCE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 7, 1993  COLORADO
C
C*********************************************************************
C
C     CONVERT COORDINATES
C
      CALL CONVRT(SLAT,SLON,SSLAT,SSLON)
C
      CALL CONVRT(ELAT,ELON,EELAT,EELON)
C
      IF(IDF.NE.0)GO TO 500
C
C     COMPUTE DISTANCE-AZIMUTH
C
      CALL DIAZ(EELAT,EELON,SSLAT,SSLON,DELTA,DISKM,AZSE,AZES)
C
  500 IF(NPER.NE.0)GO TO 2000
C
C     COMPUTE BODY WAVE TRAVEL TIME
C
 1000 TT=TRVLTM(DELTA,EDEPTH,NPHAS,EELAT,SSLAT,SELEV)
C
      GO TO 9000
```

```
C
 2000 CALL STRAV(DELTA,DISKM,UVEL,PERIOD,NPER,TT)
C
 9000 RETURN
      END
```

```
      SUBROUTINE TTOF1(Z,ISAMP,LENG,ISIGN,N2PWR,N2LEN,NH2,N2P1,NZ)
C
C*********************************************************************
C
C     FOURIER TRANSFORM OF DATA USING RCOOL
C
C     Z - TIME SERIES OR ITS TRANSFORM - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C     LENG - TOTAL NUMBER OF LOCATIONS AVAILABLE - INPUT
C     ISIGN - TRANSFORM FLAG - INPUT
C             = 1 DIRECT TRANSFORM
C             =-1 INDIRECT TRANSFORM
C     N2PWR - POWER OF 2 WHICH IS TO BE TRANSFORMED - OUTPUT
C     N2LEN - NUMBER OF POINTS FOR POWER OF TWO - OUTPUT
C     NH2 - NUMBER OF REAL COMPONENTS - OUTPUT
C     N2P1 - N2LEN+1 - OUTPUT
C     NZ - NUMBER OF ZEROS WHICH HAVE BEEN ADDED - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 28, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION Z(1)
C
C     DETERMINE POWER OF 2 PARAMETERS
C
      CALL POWER2(ISAMP,LENG,N2PWR,N2LEN,NH2,N2P1,NTOT,NZ)
C
      IF(ISIGN.LT.0)GO TO 520
      JJ=ISAMP+1
      IF(N2LEN.LT.JJ)GO TO 600
      KK=NTOT
C
C     ZERO LOCATIONS WHICH WILL BE TRANSFORMED BEYOND DATA
C
      DO 300 J=JJ,KK
      Z(J)=0.0
  300 CONTINUE
C
      GO TO 600
C
  500 IF(ISIGN.GT.0)GO TO 600
C
C     SCALE DATA FOR INVERSE TRANSFORM
C
  520 DO 550 J=2,NTOT
      Z(J)=Z(J)*0.5
  550 CONTINUE
C
C     TRANSFORM TIME SERIES
C
  600 CALL RCOOL(Z,N2PWR,ISIGN)
C
      IF(ISIGN.LT.0)GO TO 9000
C
C     SCALE TRANSFORMED DATA
```

```
C
      FAC=N2LEN*0.5
C
      DO 1000 J=1,NTOT
      Z(J)=Z(J)/FAC
 1000 CONTINUE
C
      Z(1)=Z(1)*0.5
C
 9000 RETURN
      END
```

```
      SUBROUTINE TTOF3(Z,ISAMP,SAMR,LENG,A,B,C,PH,N2PWR,N2LEN,NH2,
     * N2P1,NZ)
C
C*****************************************************************
C
C     FOURIER TRANSFORM OF DATA USING FOURT
C
C     Z - TIME SERIES - INPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     LENG - TOTAL NUMBER OF LOCATIONS AVAILABLE - INPUT
C     A - FOURIER COSINE COEFFICIENT - OUTPUT
C     B - FOURIER SINE COEFFICIENT - OUTPUT
C     C - FOURIER POWER - OUTPUT
C     PH - PHASE ANGLE (IN DEGREES) - OUTPUT
C     N2PWR - POWER OF 2 WHICH IS TO BE TRANSFORMED - OUTPUT
C     N2LEN - NUMBER OF POINTS FOR POWER OF TWO - OUTPUT
C     NH2 - NUMBER OF REAL COMPONENTS = N2LEN/2+1 - OUTPUT
C     N2P1 - N2LEN+1 - OUTPUT
C     NZ - NUMBER OF ZEROS WHICH HAVE BEEN ADDED - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 28, 1993  COLORADO
C
C*****************************************************************
C
      DIMENSION Z(1),A(1),B(1),C(1),PH(1)
C
C     DETERMINE POWER OF 2 PARAMETERS
C
      CALL POWER2(ISAMP,LENG,N2PWR,N2LEN,NH2,N2P1,NTOT,NZ)
C
C     TRANSFORM TIME SERIES
C
      CALL FOURT(Z,ISAMP,SAMR,A,B,C,PH)
C
      RETURN
      END
```

```
      SUBROUTINE UDISPX
C
C*********************************************************************
C
C     MOVING WINDOW DETERMINATION OF GROUP VELOCITY
C
C     WRITTEN BY ROBERT P. MASSE
C
C     MARCH 7, 1994  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/DEM/IDEM,IDEM1
      COMMON/DTAZ/DELTA,DISKM,AZSE,AZES,CTT,ATT,DTT
      COMMON/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DELTA(100),DISKM(100),AZSE(100),AZES(100),CTT(100),
     * ATT(100),DTT(100)
      DIMENSION EXFLAG(10)
      DIMENSION BX(1)
      DIMENSION DATA(1)
      DIMENSION VEL(300),T(133),SYNTH(17000),QUADR(17000),VT(300,133),
     * NT(133),IVT(300,133),NNT(133)
C
      EQUIVALENCE (BX(20001),VT(1,1)),(DATA(164001),SYNTH(1))
      EQUIVALENCE (DATA(182001),QUADR(1)),(BX(20001),IVT(1,1))
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      IDC=NACHLS*LENG
      IF(IDC.GT.164000)GO TO 7600
      IF(IDEM.LT.200000)GO TO 7600
C
C     READ DISPERSION CALCULATION PARAMETERS
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','UMIN',3X,'UMAX',3X,'DELU',/)
      READ(IRE,*,END=9000)UMIN,UMAX,DELU
C
C     UMIN - MINIMUM GROUP VELOCITY (IN KM/S)
C     UMAX - MAXIMUM GROUP VELOCITY (IN KM/S)
C     DELU - INCREMENT IN GROUP VELOCITY (IN KM/S)
C            = 0 SET TO 0.01
C
      IF(DELU.LE.0.0)DELU=0.01
```

```
      WRITE(IWR,550)UMIN,UMAX,DELU
      WRITE(IWR2,550)UMIN,UMAX,DELU
  550 FORMAT(/,10X,'UMIN = ',F8.3,5X,'UMAX = ',F8.3,5X,'DELU = ',
     * F8.3,/)
C
      IF(UMIN.LE.0.0.OR.UMAX.LE.0.0)GO TO 7300
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','TMIN',3X,'TMAX',3X,'NPERV',/)
      READ(IRE,*,END=9000)TMIN,TMAX,NPERV
C
C     TMIN - MINIMUM PERIOD (IN SECONDS)
C     TMAX - MAXIMUM PERIOD (IN SECONDS)
C     NPERV - NUMBER OF PERIODS TO CALCULATE FOR
C             = 0 SET TO 44
C             > 133 SET TO 133
C
      IF(NPERV.GT.133)NPERV=133
      IF(NPERV.LE.0)NPERV=44
      WRITE(IWR,650)TMIN,TMAX,NPERV
      WRITE(IWR2,650)TMIN,TMAX,NPERV
  650 FORMAT(/,10X,'TMIN = ',F9.3,5X,'TMAX = ',F9.3,5X,'NPERV = ',I4,/)
C
      IF(TMIN.LE.0.0.OR.TMAX.LE.0.0)GO TO 7000
C
      WRITE(IWR2,700)
  700 FORMAT(/,'++ INPUT: ','ALPHA',3X,'BAND',/)
      READ(IRE,*,END=9000)ALPHA,BAND
C
C     ALPHA - DEFINES SLOPE OF FILTER
C             = 0 SET TO 3.1
C     BAND - DEFINES BANDWIDTH OF FILTER
C            = 0 SET TO 0.4
C            > 1 SET TO 0.9
C
      IF(ALPHA.LE.0.0)ALPHA=3.1
      IF(BAND.LE.0.0)BAND=0.4
      IF(BAND.GE.1.0)BAND=0.9
      WRITE(IWR,750)ALPHA,BAND
      WRITE(IWR2,750)ALPHA,BAND
  750 FORMAT(/,10X,'ALPHA = ',F8.3,5X,'BAND = ',F8.3,/)
C
 1000 WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','ICHAN',3X,'IFIR',/)
      READ(IRE,*,END=9000)ICHAN,IFIR
C
C     ICHAN - DESIRED CHANNEL TO COMPUTE DISPERSION FOR
C             = 0 SET TO 1
C     IFIR - FIRST SAMPLE TO PROCESS
C            = 0 SET TO 1
C
      IF(ICHAN.LE.0)ICHAN=1
      IF(ICHAN.GT.NACHLS)GO TO 7000
      IF(IFIR.LE.0)IFIR=1
      WRITE(IWR,1200)ICHAN,IFIR
      WRITE(IWR2,1200)ICHAN,IFIR
 1200 FORMAT(/,10X,'ICHAN = ',I4,5X,'IFIR = ',I2,/)
C
```

```
      WRITE(IWR2,1300)
 1300 FORMAT(/,'++ INPUT: ','IAPH',/)
      READ(IRE,*,END=9000)IAPH
C
C     IAPH - SPECTRA OPTION FLAG
C            = 0 COMPUTE AMPLITUDE SPECTRA
C            = 1 COMPUTE PHASE SPECTRA
C
      WRITE(IWR,1350)IAPH
      WRITE(IWR2,1350)IAPH
 1350 FORMAT(/,10X,'IAPH = ',I2,/)
C
      INDEX=(ICHAN-1)*LENG+IFIR
      ISAMP=JSAMP(ICHAN)
      ISSAV=ISAMP
      SAMR=RSAMR(ICHAN)
      IF(IFIR.GT.ISAMP)GO TO 7000
      ISAMP=ISAMP-IFIR+1
C
C     DETERMINE POWER OF TWO
C
      LEN=17000
C
      CALL POWER2(ISAMP,LEN,N2PWR,N2LEN,NH2,N2P1,NTOT,NZ)
C
      FPWR=N2LEN
      IF(N2LEN.GE.ISAMP)GO TO 1380
C
      WRITE(IWR,1370)ISAMP,N2LEN
      WRITE(IWR2,1370)ISAMP,N2LEN
 1370 FORMAT(/,10X,'DATA LENGTH OF ',I6,' SAMPLES IS TOO LONG TO ',
     * 'TRANSFORM',/,10X,'ONLY ',I6,' SAMPLES TRANSFORMED',/)
      ISAMP=N2LEN
      GO TO 1410
C
 1380 WRITE(IWR,1390)ISAMP,N2LEN
      WRITE(IWR2,1390)ISAMP,N2LEN
 1390 FORMAT(/,10X,'DATA LENGTH IS ',I6,' SAMPLES',/,10X,
     * 'NUMBER OF SAMPLES TRANSFORMED IS',I6,/)
C
C     REMOVE MEAN AND LINEAR TREND OF DATA
C
 1410 IDEG=1
C
      CALL METR(DATA(INDEX),ISSAV,IDEG)
C
C     ZERO BX ARRAY
C
      DO 1420 I=1,20000
      BX(I)=0.0
 1420 CONTINUE
C
C     CALCULATE DISTANCE-AZIMUTH FOR STATION
C
      IFL=1
C
      CALL ADAZE(IFL,ICHAN,ICHAN)
C
```

```
C     CALCULATE CHANNEL START TIME AND ORIGIN TIME
C
      CALL EETIME(START,ORGT,TDIF,ICHAN,IFIR)
C
      TINC=(ISAMP-1)/SAMR
      TLDIF=TDIF+TINC
      IF(TLDIF.LE.0.0)GO TO 7300
C
C     SET UP ACTUAL MINIMUM AND MAXIMUM GROUP VELOCITIES
C
      IF(TDIF.LE.0.0)GO TO 1600
C
      VMAX=DISKM(ICHAN)/TDIF
      IF(UMAX.LE.VMAX)GO TO 1600
C
      WRITE(IWR,1500)VMAX
      WRITE(IWR2,1500)VMAX
 1500 FORMAT(/,10X,'MAXIMUM POSSIBLE GROUP VELOCITY IS ',F8.3,/)
      IU=VMAX/DELU
      UMAX=IU
      UMAX=UMAX*DELU
      GO TO 1800
C
 1600 WRITE(IWR,1650)UMAX
      WRITE(IWR2,1650)UMAX
'1650 FORMAT(/,10X,'MAXIMUM GROUP VELOCITY IS ',F8.3,/)
C
 1800 VMIN=DISKM(ICHAN)/TLDIF
      IF(VMIN.LE.UMIN)GO TO 2000
C
      WRITE(IWR,1850)VMIN
      WRITE(IWR2,1850)VMIN
 1850 FORMAT(/,10X,'MINIMUM POSSIBLE GROUP VELOCITY IS ',F8.3,/)
      IU=VMIN/DELU
      UMIN=IU
      UMIN=UMIN*DELU+DELU
      GO TO 2200
C
 2000 WRITE(IWR,2050)UMIN
      WRITE(IWR2,2050)UMIN
 2050 FORMAT(/,10X,'MINIMUM GROUP VELOCITY IS ',F8.3,/)
C
 2200 IF(UMAX.LE.UMIN)GO TO 7300
C
C     DETERMINE NUMBER OF VELOCITY VALUES REQUESTED
C
      NVEL=(UMAX-UMIN)/DELU+1.0001
      IF(NVEL.LE.300)GO TO 2400
C
      NVEL=300
      UMIN=UMAX-299.0*DELU
      WRITE(IWR,2300)UMIN
      WRITE(IWR2,2300)UMIN
 2300 FORMAT(/,10X,'TOO MANY VELOCITIES VALUES REQUESTED',/,10X,
     * 'MINIMUM GROUP VELOCITY USED IS ',F8.3,/)
C
 2400 IF(UMIN.LE.0.0)GO TO 7300
      IF(UMAX.LE.UMIN)GO TO 7300
```

```
C
C     CALCULATE ALL VELOCITIES
C
      DO 2500 I=1,NVEL
      FI=I-1
      VEL(I)=UMAX-FI*DELU
 2500 CONTINUE
C
C     SET UP ACTUAL PERIOD VALUES
C
      PMAX=FPWR/SAMR
      PMIN=2.0/SAMR
      IF(TMIN.LT.PMIN)TMIN=PMIN
      IF(TMAX.GT.PMAX)TMAX=PMAX
C
      FMAX=1.0/TMIN
      FMIN=1.0/TMAX
      DF=(FMAX-FMIN)/(NPERV-1)
      DFMIN=1.0/PMAX
      IF(DF.LT.DFMIN)DF=DFMIN
C
C     CALCULATE ALL PERIODS
C
      T(1)=TMIN
      FREQ=FMAX
C
      DO 2700 I=2,133
      II=I
      FREQ=FREQ-DF
      IF(FREQ.LE.0.0)GO TO 2800
      T(I)=1.0/FREQ
      IF(TMAX.LT.T(I))GO TO 2800
 2700 CONTINUE
C
      NPER=II
      GO TO 2820
C
 2800 NPER=II-1
C
 2820 WRITE(IWR,2830)DF,SAMR
      WRITE(IWR2,2830)DF,SAMR
 2830 FORMAT(/,10X,'INCREMENT IN FREQUENCY IS ',F10.4,/,10X,
     * 'SAMPLING RATE IS ',F7.2,/)
C
C     DETERMINE INDICES OF FOURIER COEFFICIENTS FOR PERIODS CLOSEST TO
C     REQUESTED PERIODS
C
      DO 3000 I=1,NPER
C
      DO 2850 J=2,NH2
      JJ=J
      FJ=J
      PER1=FPWR/(FJ-1.0)
      PER2=FPWR/FJ
      IF((T(I)*SAMR).GE.PER2)GO TO 2900
 2850 CONTINUE
C
 2900 IF((PER1+PER2-2.0*T(I)*SAMR).GE.0.0)GO TO 2950
```

```
      NT(I)=JJ-1
      GO TO 3000
 2950 NT(I)=JJ
 3000 CONTINUE
C
C     TRANSFER DATA TRACE TO BX ARRAY
C
      IFL=1
      IB1=IFIR
      IL1=IB1+ISAMP-1
      NCH1=ICHAN
      IB2=1
      NCH2=1
C
      CALL MOVE(IFL,IB1,IL1,NCH1,IB2,NCH2)
C
C     SCALE DATA
C
      CALL SFS(BX,ISAMP,SFX)
C
      IF(SFX.LE.5.0)GO TO 3200
      SFXX=5.0/SFX
C
      CALL CALB(BX,ISAMP,SFXX)
C
C     DIRECT TRANSFORM OF THE TIME SERIES
C
 3200 ISIGN=1
C
      CALL RCOOL(BX,N2PWR,ISIGN)
C
C     SCALE TRANSFORMED DATA
C
      FAC=N2LEN*0.5
C
      DO 3500 I=1,NTOT
      BX(I)=BX(I)/FAC
 3500 CONTINUE
C
      BX(1)=BX(1)*0.5
C
C     DETERMINING LOWEST AND HIGHEST INDICES TO BE USED IN THE INVERSE
C     TRANSFORMATION
C
      BANDL=SQRT(1.0+BAND*BAND)
      BANDU=BANDL+BAND
      BANDL=BANDL-BAND
      IBL=FPWR*BANDL/(T(NPER)*SAMR)
      IBU=FPWR*BANDU/(T(1)*SAMR)
      KMIN=IBL+1
      IF(KMIN.GT.0)GO TO 4200
C
 4000 KMIN=3
      WRITE(IWR,4100)KMIN
      WRITE(IWR2,4100)KMIN
 4100 FORMAT(/,10X,'WINDOW EXCEEDS UPPER RANGE OF PERIODS;'/,
     * 10X,'WINDOWING WILL BE INTERRUPTED AT N =',I4,/)
C
```

```
 4200 KMAX=IBU+1
      IF(KMAX.LE.NH2)GO TO 4500
C
 4300 KMAX=NH2
      WRITE(IWR,4400)KMAX
      WRITE(IWR2,4400)KMAX
 4400 FORMAT(/,10X,'WINDOW EXCEEDS LOWER RANGE OF PERIODS;',/,
     * 10X,'WINDOWING WILL BE INTERRUPTED AT N = ',I4,/)
C
C     FILTERING IN THE FREQUENCY DOMAIN
C
 4500 DO 6000 M=1,NPER
C
C     ZERO ARRAY FOR FILTERING TRANSFORM
C
      DO 4600 I=1,NTOT
      SYNTH(I)=0.0
 4600 CONTINUE
C
C     DETERMINE MINIMUM AND MAXIMUM INDICES TO FILTER
C
      KU=FPWR*BANDU/(T(M)*SAMR)
      KU=KU+1
      KP=KU
      IF(KU.LE.KMAX)GO TO 4700
      KU=KMAX
 4700 KL=FPWR*BANDL/(T(M)*SAMR)
      KL=KL-1
      KS=KL
      IF(KS.LT.1)KS=1
      IF(KL.GE.KMIN)GO TO 4800
      KL=KMIN
 4800 NL=NT(M)
C
      ZPREV=KU-KL
      ZPREV=1.0/ZPREV
      DOT=NL-KS
      DOT=1.0/DOT
C
C     FILTER FREQUENCY SPECTRA FOR GIVEN PERIOD
C
      DO 4900 L=KL,NL
      J=NH2+L
      FACT=NL-L
      FACT=FACT*DOT
      FACT=EXP(-ALPHA*FACT*FACT)*ZPREV
      SYNTH(L)=BX(L)*FACT
      SYNTH(J)=BX(J)*FACT
 4900 CONTINUE
C
      NL=NT(M)+1
      DOT=KP-NL+1
      DOT=1.0/DOT
C
      DO 5000 L=NL,KU
      J=NH2+L
      FACT=L-NL+1
      FACT=FACT*DOT
```

```
      FACT=EXP(-ALPHA*FACT*FACT)*ZPREV
      SYNTH(L)=BX(L)*FACT
      SYNTH(J)=BX(J)*FACT
 5000 CONTINUE
C
C     RETURN TO TIME DOMAIN
C
C     CREATE THE QUADRATURE SPECTRUM
C
      DO 5300 I=1,NH2
      J=NH2+I
      QUADR(I)=-SYNTH(J)
      QUADR(J)=SYNTH(I)
 5300 CONTINUE
C
C     SCALE DATA FOR INVERSE TRANSFORM
C
      DO 5400 I=2,NTOT
      SYNTH(I)=SYNTH(I)*0.5
      QUADR(I)=QUADR(I)*0.5
 5400 CONTINUE
C
C     INVERSE FOURIER TRANSFORM IN-PHASE AND QUADRATURE SPECTRA
C
      ISIGN=-1
C
      CALL RCOOL(SYNTH,N2PWR,ISIGN)
C
      CALL RCOOL(QUADR,N2PWR,ISIGN)
C
C     DETERMINATION OF ENVELOPE AMPLITUDE OR PHASE SHIFT
C
      DO 5900 N=1,NVEL
      SAP=(DISKM(ICHAN)/VEL(N)-TDIF)*SAMR
      K=SAP
      FK=K
      DSAP=SAP-FK
      IF(K.GT.1)GO TO 5500
      SYN=0.0
      QUA=0.0
      GO TO 5600
C
 5500 SYN=SYNTH(K-1)+DSAP*(SYNTH(K)-SYNTH(K-1))
      QUA=QUADR(K-1)+DSAP*(QUADR(K)-QUADR(K-1))
C
 5600 IF(IAPH.GT.0)GO TO 5700
C
      VT(N,M)=SQRT(SYN*SYN+QUA*QUA)
      GO TO 5900
C
 5700 VT(N,M)=PHASOR(SYN,QUA)
C
 5900 CONTINUE
C
 6000 CONTINUE
C
      IF(IAPH.GT.0)GO TO 6200
C
```

```
C     SCALE AMPLITUDES
C
      AMAX=0.0
C
      DO 6110 J=1,NPER
C
      DO 6100 I=1,NVEL
      IF(VT(I,J).GT.AMAX)AMAX=VT(I,J)
 6100 CONTINUE
C
 6110 CONTINUE
C
      IF(AMAX.LE.0.0)GO TO 7000
C
      DO 6150 J=1,NPER
C
      DO 6120 I=1,NVEL
      IVT(I,J)=(VT(I,J)*99.0/AMAX)+0.5
 6120 CONTINUE
C
 6150 CONTINUE
C
      GO TO 6300
C
C     SCALE PHASES
C
 6200 DO 6250 J=1,NPER
C
      DO 6220 I=1,NVEL
      IVT(I,J)=VT(I,J)/4.0+0.5
 6220 CONTINUE
C
 6250 CONTINUE
C
 6300 WRITE(IWR2,6350)UMIN,UMAX,NVEL
 6350 FORMAT(/,10X,'VELOCITY RANGE: ',F8.3,' TO ',F8.3,' WITH ',I4,
     * ' VALUES',/)
      WRITE(IWR2,6400)TMIN,TMAX,NPER
 6400 FORMAT(/,10X,'PERIOD RANGE: ',F8.3,' TO ',F8.3,' WITH ',I4,
     * ' VALUES',/)
C
      IF(IAPH.GT.0)GO TO 6520
C
      WRITE(IWR,6500)
      WRITE(IWR2,6500)
 6500 FORMAT(/,10X,'AMPLITUDE',/)
      GO TO 6560
C
 6520 WRITE(IWR,6540)
      WRITE(IWR2,6540)
 6540 FORMAT(/,10X,'PHASE/4',/)
C
C     OUTPUT FINAL DISPERSION RESULTS
C
 6560 NVL=NVEL/54
      IF((54*NVL).NE.NVEL)NVL=NVL+1
      NPL=NPER/22
      IF((22*NPL).NE.NPER)NPL=NPL+1
```

```
      NV1=1
      NV2=54
      IF(NV2.GT.NVEL)NV2=NVEL
C
      DO 6570 I=1,NPER
      NNT(I)=T(I)+0.5
 6570 CONTINUE
C
      DO 6800 I=1,NVL
      NP1=1
      NP2=22
      IF(NP2.GT.NPER)NP2=NPER
C
      DO 6700 J=1,NPL
C
      DO 6600 K=NV1,NV2
      WRITE(IWR,6580)VEL(K),(IVT(K,L),L=NP1,NP2)
      WRITE(IWR2,6580)VEL(K),(IVT(K,L),L=NP1,NP2)
 6580 FORMAT(F6.2,22I3)
 6600 CONTINUE
C
      WRITE(IWR,6620)(NNT(L),L=NP1,NP2)
      WRITE(IWR2,6620)(NNT(L),L=NP1,NP2)
 6620 FORMAT(/,6X,22I3)
      WRITE(IWR,6650)
      WRITE(IWR2,6650)
 6650 FORMAT(/)
      NP1=NP1+22
      NP2=NP2+22
      IF(NP2.GT.NPER)NP2=NPER
 6700 CONTINUE
C
      NV1=NV1+54
      NV2=NV2+54
      IF(NV2.GT.NVEL)NV2=NVEL
 6800 CONTINUE
C
      WRITE(IWR,6820)
      WRITE(IWR2,6820)
 6820 FORMAT(10X,'INDEX',7X,'PERIOD-ACTUAL',7X,'PERIOD-DISPLAY',/)
C
      DO 6850 I=1,NPER
      WRITE(IWR,6840)NT(I),T(I),NNT(I)
      WRITE(IWR2,6840)NT(I),T(I),NNT(I)
 6840 FORMAT(10X,I4,9X,F8.3,13X,I6)
 6850 CONTINUE
C
      IF(IAPH.NE.0)GO TO 9000
      IF(SFX.LT.5.0)GO TO 6900
      AMAX=AMAX*SFX*SFX/25.0
 6900 WRITE(IWR,6950)AMAX
      WRITE(IWR2,6950)AMAX
 6950 FORMAT(/,10X,'POWER LEVEL OF 99 CORRESPONDS TO ',E12.2,/)
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****'/)
```

```
      GO TO 9000
C
 7300 WRITE(IWR2,7400)
 7400 FORMAT(/,10X,'***** VELOCITIES DETERMINED ARE NOT ',
     * 'POSSIBLE *****',/)
      GO TO 9000
C
 7600 WRITE(IWR2,7700)
 7700 FORMAT(/,10X,'***** NOT ENOUGH ROOM LEFT IN DATA ARRAY FOR ',
     * 'SCRATCH *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE WRDATX
C
C*****************************************************************
C
C     EXECUTIVE ROUTINE FOR WRITE DATA TO FILE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*40 FILENM
C
      DATA NY/'y'/
C
      LUN2=10
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL TO WRITE TO FILE
C             = 0 WRITE ALL CHANNELS
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'WRITE DATA TO FILE PARAMETERS',/)
      WRITE(IWR,300)ICHAN
      WRITE(IWR2,300)ICHAN
  300 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
  350 WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT; ','FILENM',/)
      READ(IRE,500,END=9000)FILENM
  500 FORMAT(A40)
C
C     FILENM - NAME OF FILE TO CREATE FOR DATA
C
      WRITE(IWR,600)FILENM
      WRITE(IWR2,600)FILENM
  600 FORMAT(/,10X,'FILENM = ',A40,
```

```
     * //,'++ INPUT: IS IT CORRECT (y/n)?',/)
      READ(IRE,650,END=9000)KY
  650 FORMAT(A1)
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 350
C
C     WRITE CHANNEL DATA TO FILE
C
      IF(ICHAN.LE.0)GO TO 1000
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 2000
C
 1000 NCH1=1
      NCH2=NACHLS
C
 2000 DO 3000 I=NCH1,NCH2
      INDEX=(I-1)*LENG+1
C
      CALL WRFIL(DATA(INDEX),I,FILENM)
C
 3000 CONTINUE
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'FILES NOT WRITTEN',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'FILES SUCCESSFULLY WRITTEN',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE WRFIL(X,ICHAN,FILENM)
C
C*********************************************************************
C
C     WRITE DATA TO FILE
C
C     X - DATA ARRAY TO WRITE TO FILE - INPUT
C     ICHAN - CHANNEL NUMBER TO WRITE TO FILE - INPUT
C     FILENM - FILE NAME FOR WRITE - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/OPTION/IOPS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION X(1),IOPS(1),EXFLAG(10)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
      CHARACTER*40 FILENM
C
C     CHECK IF FILE LUN2 IS OPEN
C
      IF(IOPS(51).NE.0)GO TO 300
      OPEN(LUN2,FILE=FILENM,FORM='UNFORMATTED',STATUS='UNKNOWN')
      IOPS(51)=1
C
C     WRITE HEADER AND DATA
C
  300 K=ICHAN
C
      WRITE(LUN2)JCODE(K),(JCHN(I,K),I=1,4),RLAT(K),RLON(K),RELEV(K),
     * JYEAR(K),JDOFY(K),JHOUR(K),JMIN(K),RSEC(K),RSAMR(K),RA0(K),
     * JNP(K),(RPOLES(I,K),I=1,30),JNZ(K),(RZEROS(I,K),I=1,20),
     * JSAMP(K),JFLAG(K),EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,
     * IEXHR,IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
C
      ISAMP=JSAMP(K)
C
      WRITE(LUN2)(X(I),I=1,ISAMP)
C
      RETURN
      END
```

```
      SUBROUTINE XSAMP
C
C*************************************************************************
C
C     LIST NUMBER OF SAMPLES
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 26, 1993  COLORADO
C
C*************************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR,100)
      WRITE(IWR2,100)
  100 FORMAT(/,10X,'CHANNEL NUMBER',5X,'SAMPLES',5X,'SAMPLING RATE',/)
      WRITE(IWR,500)(I,JSAMP(I),RSAMR(I),I=1,NACHLS)
      WRITE(IWR2,500)(I,JSAMP(I),RSAMR(I),I=1,NACHLS)
  500 FORMAT(15X,I4,10X,I6,7X,F8.2)
C
      RETURN
      END
```

```
      SUBROUTINE XSAMP2
C
C*****************************************************************
C
C     LIST NUMBER OF SAMPLES AND CHANGE NUMBER
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 26, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',3X,'ISAMP',3X,'SEC',/)
      READ(IRE,*,END=9000)ICHAN,ISAMP,SEC
C
C     ICHAN - CHANNEL NUMBER
C             = 0 CHANGE SAMPLE NUMBER FOR ALL CHANNELS TO ISAMP
C     ISAMP - NUMBER OF SAMPLES TO CHANGE TO
C     SEC - NUMBER OF SECONDS TO CHANGE TO
C           > 0 USE SEC RATHER THAN ISAMP
C           = 0 USE ISAMP RATHER THAN SEC
C
      WRITE(IWR,1000)
      WRITE(IWR2,1000)
 1000 FORMAT(/,10X,'CHANGE SAMPLE NUMBER PARAMETERS',/)
      WRITE(IWR,1500)ICHAN,ISAMP,SEC
      WRITE(IWR2,1500)ICHAN,ISAMP,SEC
 1500 FORMAT(/,10X,'ICHAN = ',I4,5X,'ISAMP = ',I6,5X,'SEC = ',F8.2,/)
C
      IF(ISAMP.GT.LENG)ISAMP=LENG
      IF(ICHAN.GT.NCHLS)GO TO 7000
      IF(ICHAN.LE.0)GO TO 2000
C
      IF(SEC.GT.0.0)ISAMP=SEC*RSAMR(ICHAN)
      IF(ISAMP.GT.LENG)ISAMP=LENG
      JSAMP(ICHAN)=ISAMP
      GO TO 3000
C
 2000 DO 2500 I=1,NACHLS
      IF(SEC.GT.0.0)ISAMP=SEC*RSAMR(I)
      IF(ISAMP.GT.LENG)ISAMP=LENG
      JSAMP(I)=ISAMP
 2500 CONTINUE
C
 3000 WRITE(IWR,3100)
```

```
      WRITE(IWR2,3100)
 3100 FORMAT(/,10X,'CHANNEL NUMBER',5X,'SAMPLES',5X,'SAMPLING RATE',/)
      WRITE(IWR,3200)(I,JSAMP(I),RSAMR(I),I=1,NACHLS)
      WRITE(IWR2,3200)(I,JSAMP(I),RSAMR(I),I=1,NACHLS)
 3200 FORMAT(15X,I4,10X,I6,7X,F8.2)
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'NUMBER OF SAMPLES NOT CHANGED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'NUMBER OF SAMPLES SUCCESSFULLY CHANGED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE XYAX(XF,XMAX,SCALX,XI,XXI,YF,YMAX,SCALY,YI,YYI,
     * IPF1,IPF2,FAC,AXTIT,AYTIT,AMTIT,LGF,SIZ1,SIZ2,SIZ3)
C
C*********************************************************************
C
C     PLOT X-Y AXES AND LABEL
C
C     THE X AND Y AXES ARE DESIGNATED ACCORDING TO THEIR ORIENTATION
C     ON THE SCREEN. THE FINAL ORIENTATION OF THESE AXES ON PAPER IS
C     ROTATED 90 DEGREES COUNTERCLOCKWISE.
C
C     XF - X AXIS PLOT ORIGIN VALUE - INPUT
C     XMAX - MAXIMUM X VALUE - INPUT
C     SCALX - PLOTTING INCREMENT - INPUT
C           - LINEAR PLOT (INCHES/X VALUE)
C           - LOG PLOT (TOTAL X INCHES FOR ALL CYCLES)
C     XI - INCREMENT IN X VALUE FOR TICKS - INPUT
C     XXI - INCREMENT IN X VALUE FOR LABELED TICKS - INPUT
C           (MUST BE INTEGER MULTIPLE OF XI)
C     YF - Y AXIS PLOT ORIGIN VALUE - INPUT
C     YMAX - MAXIMUM Y VALUE - INPUT
C            (PLOT SIZE MUST BE 1.5 INCHES LARGER THAN THIS)
C     SCALY - PLOTTING INCREMENT - INPUT
C           - LINEAR PLOT (INCHES/Y VALUE)
C           - LOG PLOT (TOTAL Y INCHES FOR ALL CYCLES)
C     YI - INCREMENT IN Y VALUE FOR TICKS - INPUT
C     YYI - INCREMENT IN Y VALUE FOR LABELED TICKS - INPUT
C     IPF1 - PLOT FLAG FOR NUMBERS ON X AXIS - INPUT
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     IPF2 - PLOT FLAG FOR NUMBERS ON Y AXIS - INPUT
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     FAC - PLOT SCALE FACTOR - INPUT
C     AXTIT - TITLE FOR X AXIS (20 CHARACTERS MAXIMUM) - INPUT
C     AYTIT - TITLE FOR Y AXIS (20 CHARACTERS MAXIMUM) - INPUT
C     AMTIT - TITLE FOR PLOT (20 CHARACTERS MAXIMUM) - INPUT
C     LGF - FLAG FOR LOG PLOT - INPUT
C           = 0 LINEAR PLOT
C           = 1 LOG PLOT ON X AXIS
C           = 2 LOG PLOT ON Y AXIS
C           = 3 LOG PLOT ON X AND Y AXIS
C     SIZ1 - NUMBER SIZE FOR X AND Y AXES (IN INCHES) - INPUT
C            = 0 SET TO 0.14
C     SIZ2 - TITLE SIZE FOR X AND Y AXES (IN INCHES) - INPUT
C            = 0 SET TO 0.28
C     SIZ3 - SIZE FOR MAIN TITLE (IN INCHES) - INPUT
C            = 0 SET TO 0.28
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 1, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/PCKF/ICHECK,IPLFL
C
      CHARACTER*1 BCD
```

```
      CHARACTER*20 AXTIT,AYTIT,AMTIT
C
      DATA BCD/'-'/
C
      IF(IPLFL.EQ.0)GO TO 50
C
C     CLEAR SCREEN PLOT FRAME AND ADVANCE DEFERRED FRAME
C
      CALL ADVPLT
C
   50 CALL FACTOR(FAC)
C
      CALL PLOT(1.0,0.0,-3)
C
C     INITIALIZE PLOT PARAMETERS
C
      IF(SIZ1.LE.0.0)SIZ1=0.14
      IF(SIZ2.LE.0.0)SIZ2=0.28
      IF(SIZ3.LE.0.0)SIZ3=0.28
      IF(LGF.LT.0.OR.LGF.GT.4)GO TO 9000
      IF((LGF.EQ.1.OR.LGF.EQ.3).AND.XF.EQ.0.0)GO TO 9000
      IF((LGF.EQ.2.OR.LGF.EQ.3).AND.YF.EQ.0.0)GO TO 9000
C
      IEND=0
      IPASS=1
      IPF=IPF1
      IFIRNX=0
      IFIRNY=0
      ZLG=1.0
      XLT=1.50
      YLT=ABS(YMAX-YF)*SCALY+1.50
      XA=XLT
      YA=YLT
      YYLT=YLT
      FTM=XF
      FTB=XF
      FCNT=XF+XXI
      IF(LGF.EQ.0)GO TO 70
C
C     INITIALIZE LOG X PLOT PARAMETERS
C
      IF(LGF.EQ.2)GO TO 60
      XI=XF
      XXI=XF*10.0
      NCY=0
C
      CALL CYCLES(XF,XMAX,NCY,ZFT,ZFL,ZMAXT,ZMAXL,ZFCYT)
C
      IF(XF.EQ.ZFT)GO TO 55
      XF=ZFL
      XI=ZFT
      IF(XI.NE.XF)IFIRNX=1
      XXI=ZFT*10.0
   55 FCNT=XXI
      FNCX=ALOG10(XMAX/XF)
      IF(LGF.NE.3)GO TO 70
C
C     INITIALIZE LOG Y PLOT PARAMETERS
```

```
C
   60 YLT=SCALY+1.50
      YI=YF
      YYI=YF*10.0
      NCY=0
C
      CALL CYCLES(YF,YMAX,NCY,ZFT,ZFL,ZMAXT,ZMAXL,ZFCYT)
C
      IF(YF.EQ.ZFT)GO TO 65
      YF=ZFL
      YI=ZFT
      IF(YI.NE.YF)IFIRNY=1
      YYI=ZFT*10.0
   65 YA=YLT
      YYLT=YLT
      FNCY=ALOG10(YMAX/YF)
C
   70 IF((XMAX-XI).LT.XF)GO TO 9000
      IF((YMAX-YI).LT.YF)GO TO 9000
C
      CALL PLOT(XA,YA,3)
C
C---------------------------------------------------------------------
C
C     PLOT TOP X AXIS (RIGHT AXIS ON PAPER)
C
      IF(IPF.NE.0)GO TO 100
      IF(IFIRNX.NE.0)GO TO 100
C
C     PLOT X NUMBER VALUE
C
      XXA=XLT+0.02
      YYA=YLT+0.85
      IF(LGF.EQ.0.OR.LGF.EQ.2)XXA=XXA-0.10
C
      IF(FTM.LT.0.0)GO TO 80
      IF(LGF.EQ.0.OR.LGF.EQ.2)GO TO 75
C
      IJFTM=ALOG10(FTM)
      ATM=IJFTM
      SIZP=SIZ1*0.8
      CALL NUMBER(XXA,YYA,SIZP,ATM,270.0,'F5.0')
      ATM=10.0
      XXA=XXA-0.12
      YYA=YYA+0.08
      CALL NUMBER(XXA,YYA,SIZ1,ATM,270.0,'F3.0')
      GO TO 90
C
   75 IF(FTM.GE.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(FTM.GE.1.0.AND.FTM.LT.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(FTM.LT.1.0.AND.FTM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(FTM.LT.0.10.AND.FTM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(FTM.LT.0.010.AND.FTM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(FTM.LT.0.0010.AND.FTM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
```

```
     * 270.0,'F6.4')
      IF(FTM.LT.0.00010.AND.FTM.GE.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F7.5')
      IF(FTM.LT.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
      GO TO 90
C
   80 ATM=ABS(FTM)
      IF(ATM.GE.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(ATM.GE.1.0.AND.ATM.LT.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(ATM.LT.1.0.AND.ATM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(ATM.LT.0.10.AND.ATM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(ATM.LT.0.010.AND.ATM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(ATM.LT.0.0010.AND.ATM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(ATM.LT.0.0001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
C
   90 XA=XLT
      YA=YLT
C
      CALL PLOT(XA,YA,3)
C
C     PLOT X TICK MARK
C
  100 XXA=XLT+0.24
      YYA=YYLT-0.08
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,90.0,1)
C
      CALL PLOT(XA,YA,3)
C
  200 FTM=FTM+XI*ZLG
      IF(LGF.EQ.1.OR.LGF.EQ.3)XA=XLT+ALOG10(FTM/FTB)*SCALX/FNCX
      IF(LGF.EQ.0.OR.LGF.EQ.2)XA=XA+XI*SCALX
      IF(FTM.LT.XMAX)GO TO 235
      IF(FTM.EQ.XMAX)GO TO 230
      IEND=1
      IF(LGF.EQ.1.OR.LGF.EQ.3)XA=XLT+SCALX
      IF(LGF.EQ.0.OR.LGF.EQ.2)XA=XLT+ABS(XMAX-XF)*SCALX
      GO TO 235
C
  230 IEND=2
C
  235 CALL PLOT(XA,YA,2)
C
      IF(IEND.NE.0)GO TO 350
C
C     PLOT X TICK MARK
C
      XXA=XA+0.24
      YYA=YYLT-0.08
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,90.0,1)
C
      CALL PLOT(XA,YA,3)
```

```
C
      IF(FCNT.NE.0.0)GO TO 240
      IF(ABS(FTM-FCNT).GT.0.0001)GO TO 300
      GO TO 245
  240 IF(ABS((FTM-FCNT)/FCNT).GT.0.001)GO TO 300
  245 IF(IPF.NE.0)GO TO 270
C
C     PLOT X NUMBER VALUE
C
      XXA=XA+0.02
      YYA=YLT+0.85
      IF(LGF.EQ.0.OR.LGF.EQ.2)XXA=XXA-0.10
C
      IF(FTM.LT.0.0)GO TO 255
      IF(LGF.EQ.0.OR.LGF.EQ.2)GO TO 250
C
      IJFTM=ALOG10(FTM)
      ATM=IJFTM
      SIZP=SIZ1*0.8
      CALL NUMBER(XXA,YYA,SIZP,ATM,270.0,'F5.0')
      ATM=10.0
      XXA=XXA-0.12
      YYA=YYA+0.08
      CALL NUMBER(XXA,YYA,SIZ1,ATM,270.0,'F3.0')
      GO TO 260
C
  250 IF(FTM.GE.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(FTM.GE.1.0.AND.FTM.LT.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(FTM.LT.1.0.AND.FTM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(FTM.LT.0.10.AND.FTM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(FTM.LT.0.010.AND.FTM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(FTM.LT.0.0010.AND.FTM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(FTM.LT.0.00010.AND.FTM.GE.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F7.5')
      IF(FTM.LT.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
      GO TO 260
C
  255 ATM=ABS(FTM)
      IF(ATM.GE.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(ATM.GE.1.0.AND.ATM.LT.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(ATM.LT.1.0.AND.ATM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(ATM.LT.0.10.AND.ATM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(ATM.LT.0.010.AND.ATM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(ATM.LT.0.0010.AND.ATM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(ATM.LT.0.0001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
C
  260 CALL PLOT(XA,YA,3)
C
```

```
  270 IF(LGF.EQ.1.OR.LGF.EQ.3)GO TO 280
      IF(LGF.EQ.0.OR.LGF.EQ.2)FCNT=FCNT+XXI
      GO TO 300
C
  280 ZLG=ZLG*10.0
      FCNT=XXI*ZLG
C
  300 IF(FTM.LT.XMAX)GO TO 200
  350 IF(IPASS.EQ.2)GO TO 400
C
C---------------------------------------------------------------------
C
C     PLOT RIGHT Y AXIS (TOP AXIS ON PAPER)
C
      XTOP=XA
      YA=YLT
C
      CALL PLOT(XA,YA,3)
C
      YYLT=1.50
      YA=YYLT
C
      CALL PLOT(XA,YA,2)
C
C---------------------------------------------------------------------
C
C     PLOT BOTTOM X AXIS (RIGHT AXIS ON PAPER)
C
      XA=XLT
      FTM=XF
      FCNT=XF+XXI
      IEND=0
      IPASS=2
      IPF=1
      ZLG=1.0
      IF(LGF.NE.1.AND.LGF.NE.3)GO TO 360
      FCNT=XXI
C
  360 CALL PLOT(XA,YA,3)
C
      GO TO 100
C
C---------------------------------------------------------------------
C
C     PLOT LEFT Y AXIS (BOTTOM AXIS ON PAPER)
C
  400 IEND=0
      ZLG=1.0
      FTM=YF
      FTB=YF
      XA=XLT
      YA=YLT
      FCNT=YF+YYI
      IF(LGF.NE.2.AND.LGF.NE.3)GO TO 500
C
C     INITIALIZE LOG Y PARAMETER
C
      FCNT=YYI
```

```
C
  500 CALL PLOT(XA,YA,3)
C
C     PLOT Y TICK MARK
C
      XXA=XLT-0.08
      YYA=YLT-0.24
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,0.0,1)
C
      CALL PLOT(XA,YA,3)
C
      IF(IPF2.NE.0)GO TO 600
      IF(IFIRNY.NE.0)GO TO 600
C
C     PLOT Y NUMBER VALUE
C
      XXA=XA-0.40
      YYA=YA+0.10
      IF(LGF.EQ.0.OR.LGF.EQ.1)YYA=YYA+0.30
C
      IF(FTM.LT.0.0)GO TO 550
      IF(LGF.EQ.0.OR.LGF.EQ.1)GO TO 540
C
      IJFTM=ALOG10(FTM)
      ATM=IJFTM
      SIZP=SIZ1*0.8
      CALL NUMBER(XXA,YYA,SIZP,ATM,270.0,'F5.0')
      ATM=10.0
      XXA=XXA-0.12
      YYA=YYA+0.08
      CALL NUMBER(XXA,YYA,SIZ1,ATM,270.0,'F3.0')
      GO TO 570
C
  540 IF(FTM.GE.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(FTM.GE.1.0.AND.FTM.LT.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(FTM.LT.1.0.AND.FTM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(FTM.LT.0.10.AND.FTM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(FTM.LT.0.010.AND.FTM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(FTM.LT.0.0010.AND.FTM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(FTM.LT.0.00010.AND.FTM.GE.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F7.5')
      IF(FTM.LT.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
      GO TO 570
C
  550 ATM=ABS(FTM)
      IF(ATM.GE.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(ATM.GE.1.0.AND.ATM.LT.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(ATM.LT.1.0.AND.ATM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(ATM.LT.0.10.AND.ATM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
```

```
      IF(ATM.LT.0.010.AND.ATM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(ATM.LT.0.0010.AND.ATM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(ATM.LT.0.0001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
C
  570 CALL PLOT(XA,YA,3)
C
  600 FTM=FTM+YI*ZLG
      IF(LGF.EQ.2.OR.LGF.EQ.3)YA=YLT-ALOG10(FTM/FTB)*SCALY/FNCY
      IF(LGF.EQ.0.OR.LGF.EQ.1)YA=YA-YI*SCALY
      IF(FTM.LT.YMAX)GO TO 700
      IF(FTM.EQ.YMAX)GO TO 650
      IEND=1
      YA=YYLT
      GO TO 700
C
  650 IEND=2
C
  700 CALL PLOT(XA,YA,2)
C
      IF(IEND.NE.0)GO TO 2000
C
C     PLOT Y TICK MARK
C
      XXA=XA-0.08
      YYA=YA-0.24
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,0.0,1)
C
      CALL PLOT(XA,YA,3)
C
      IF(FCNT.NE.0.0)GO TO 730
      IF(ABS(FTM-FCNT).GT.0.0001)GO TO 1000
      GO TO 740
  730 IF(ABS((FTM-FCNT)/FCNT).GT.0.001)GO TO 1000
  740 IF(IPF2.NE.0)GO TO 800
C
C     PLOT Y NUMBER VALUE
C
      XXA=XA-0.40
      YYA=YA+0.10
      IF(LGF.EQ.0.OR.LGF.EQ.1)YYA=YYA+0.30
C
      IF(FTM.LT.0.0)GO TO 750
      IF(LGF.EQ.0.OR.LGF.EQ.1)GO TO 745
C
      IJFTM=ALOG10(FTM)
      ATM=IJFTM
      SIZP=SIZ1*0.8
      CALL NUMBER(XXA,YYA,SIZP,ATM,270.0,'F5.0')
      ATM=10.0
      XXA=XXA-0.12
      YYA=YYA+0.08
      CALL NUMBER(XXA,YYA,SIZ1,ATM,270.0,'F3.0')
      GO TO 770
C
  745 IF(FTM.GE.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
```

```
      IF(FTM.GE.1.0.AND.FTM.LT.100000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(FTM.LT.1.0.AND.FTM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(FTM.LT.0.10.AND.FTM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(FTM.LT.0.010.AND.FTM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(FTM.LT.0.0010.AND.FTM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(FTM.LT.0.00010.AND.FTM.GE.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F7.5')
      IF(FTM.LT.0.00001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
      GO TO 770
C
  750 ATM=ABS(FTM)
      IF(ATM.GE.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F6.0')
      IF(ATM.GE.1.0.AND.ATM.LT.10000.0)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.0')
      IF(ATM.LT.1.0.AND.ATM.GE.0.10)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.1')
      IF(ATM.LT.0.10.AND.ATM.GE.0.010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.2')
      IF(ATM.LT.0.010.AND.ATM.GE.0.0010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F5.3')
      IF(ATM.LT.0.0010.AND.ATM.GE.0.00010)CALL NUMBER(XXA,YYA,SIZ1,FTM,
     * 270.0,'F6.4')
      IF(ATM.LT.0.0001)CALL NUMBER(XXA,YYA,SIZ1,FTM,270.0,'F5.0')
C
  770 CALL PLOT(XA,YA,3)
C
  800 IF(LGF.EQ.2.OR.LGF.EQ.3)GO TO 900
      IF(LGF.EQ.0.OR.LGF.EQ.1)FCNT=FCNT+YYI
      GO TO 1000
C
  900 ZLG=ZLG*10.0
      FCNT=YYI*ZLG
C
 1000 IF(FTM.LT.YMAX)GO TO 600
C
C---------------------------------------------------------------------
C
 2000 CALL PLOT(0.0,YA,3)
C
      CALL PLOT(0.0,0.0,3)
C
C     PLOT X TITLE
C
      IF(LGF.EQ.0.OR.LGF.EQ.2)XXA=XLT+(ABS(XMAX-XF)*SCALX)/2.0-2.2
      IF(LGF.EQ.1.OR.LGF.EQ.3)XXA=XLT+SCALX/2.0-2.2
      YYA=YLT+1.00
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,SIZ2,AXTIT,0.0,20)
C
C     PLOT Y TITLE
C
```

```
      XXA=XLT-0.90
      IF(LGF.EQ.0.OR.LGF.EQ.1)YYA=YLT-(ABS(YMAX-YF)*SCALY)/2.0+2.2
      IF(LGF.EQ.2.OR.LGF.EQ.3)YYA=YLT-SCALY/2.0+2.2
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,SIZ2,AYTIT,270.0,20)
C
C     PLOT MAIN TITLE
C
      XXA=XTOP+0.30
      YYA=YYA+1.20
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,SIZ3,AMTIT,270.0,20)
C
      CALL PLOT(0.0,0.0,3)
C
      CALL FACTOR(1.0)
C
 9000 RETURN
      END
```

```
      SUBROUTINE XYDAPL(X,Y,LINE,MARK,NP,XF,XMAX,SCALX,YF,YMAX,SCALY,
     * FAC,LGF)
C
C*********************************************************************
C
C     PLOT X-Y DATA POINTS
C
C     X - X COORDINATES - INPUT
C     Y - Y COORDINATES - INPUT
C     LINE - FLAG FOR CONTINUOUS LINE - INPUT
C            = 0 NO LINE JOINING POINTS
C            = 1 LINE JOINING POINTS
C     MARK - SYMBOL CODE USED FOR DATA POINTS - INPUT
C            = 0 DON'T PLOT SYMBOL
C            = 1 PLOT X
C            = 2 PLOT O
C            = 3 PLOT *
C            = 4 PLOT +
C     NP - NUMBER OF DATA POINTS TO PLOT - INPUT
C     XF - X AXIS PLOT ORIGIN VALUE - INPUT
C     XMAX - MAXIMUM X VALUE - INPUT
C     SCALX - INCHES/X VALUE OR INCHES/LOG X VALUE PLOTTING
C             INCREMENT - INPUT
C     YF - Y AXIS PLOT ORIGIN VALUE - INPUT
C     YMAX - MAXIMUM Y VALUE - INPUT
C     SCALY - INCHES/Y VALUE OR INCHES/LOG Y VALUE PLOTTING
C             INCREMENT - INPUT
C     FAC - PLOT SCALE FACTOR - INPUT
C     LGF - FLAG FOR LOG PLOT - INPUT
C           = 0 LINEAR PLOT
C           = 1 LOG PLOT ON X AXIS
C           = 2 LOG PLOT ON Y AXIS
C           = 3 LOG PLOT ON X AND Y AXIS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 1, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1),Y(1),BCD(4)
C
      CHARACTER*1 BCD
C
      DATA BCD/'x','o','*','+'/
C
      IF(NP.LE.0)GO TO 9000
      IF(LGF.LT.0.OR.LGF.GT.4)GO TO 9000
C
      CALL FACTOR(FAC)
C
      XLT=1.50
      YLT=ABS(YMAX-YF)*SCALY+1.50
      IF(LGF.EQ.2.OR.LGF.EQ.3)YLT=SCALY+1.50
      IF(LGF.EQ.1.OR.LGF.EQ.3)FNCX=ALOG10(XMAX/XF)
      IF(LGF.EQ.2.OR.LGF.EQ.3)FNCY=ALOG10(YMAX/YF)
C
      K=1
```

```
C
      DO 1000 I=1,NP
      IF(X(I).LT.XF.OR.X(I).GT.XMAX)GO TO 1000
      IF(Y(I).LT.YF.OR.Y(I).GT.YMAX)GO TO 1000
      IF(LGF.EQ.0.OR.LGF.EQ.2)XX=ABS(X(I)-XF)*SCALX
      IF(LGF.EQ.0.OR.LGF.EQ.1)YY=ABS(Y(I)-YF)*SCALY
      IF(LGF.EQ.1.OR.LGF.EQ.3)XX=ALOG10(X(I)/XF)*SCALX/FNCX
      IF(LGF.EQ.2.OR.LGF.EQ.3)YY=ALOG10(Y(I)/YF)*SCALY/FNCY
      XX=XLT+XX
      YY=YLT-YY
C
      IF(K.NE.1)GO TO 500
C
      K=K+1
      GO TO 700
C
  500 IF(LINE.EQ.0)GO TO 700
C
      CALL PLOT(XX,YY,2)
C
      GO TO 800
C
  700 CALL PLOT(XX,YY,3)
C
      IF(MARK.LE.0)GO TO 1000
C
  800 XXX=XX-0.04
      YYY=YY-0.05
C
      CALL SYMBOL(XXX,YYY,0.14,BCD(MARK),0.0,1)
C
      CALL PLOT(XX,YY,3)
C
 1000 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE XYPLT
C
C*********************************************************************
C
C     PLOT X-Y SET OF DATA
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 23, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/PCKF/ICHECK,IPLFL
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*80 MSG
      CHARACTER*20 AXTIT,AYTIT,AMTIT,IBLK
C
      DATA IBLK/'                    '/,NY/'y'/
C
C     SET UP PLOT PARAMETERS
C
      CMPIN=2.54
      MODE=0
      AXTIT=IBLK
      AYTIT=IBLK
      AMTIT=IBLK
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','LGF',3X,'IHEAD',3X,'IPF1',3X,'IPF2',/)
      READ(IRE,*,END=9000)LGF,IHEAD,IPF1,IPF2
C
C     LGF - FLAG FOR LOG PLOT
C           = 0 LINEAR PLOT
C           = 1 LOG PLOT ON X AXIS
C           = 2 LOG PLOT ON Y AXIS
C           = 3 LOG PLOT ON X AND Y AXIS
C     IHEAD - FLAG FOR HEADER PLOT
C             = 0 DON'T PLOT HEADER
C             = 1 PLOT HEADER
C     IPF1 - PLOT FLAG FOR NUMBERS ON X AXIS
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     IPF2 - PLOT FLAG FOR NUMBERS ON Y AXIS
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C
```

```
      IF(IHEAD.LT.0.OR.IHEAD.GT.1)IHEAD=0
      IF(IPF1.LT.0.OR.IPF1.GT.1)IPF1=1
      IF(IPF2.LT.0.OR.IPF2.GT.1)IPF2=1
      WRITE(IWR,300)LGF,IHEAD,IPF1,IPF2
      WRITE(IWR2,300)LGF,IHEAD,IPF1,IPF2
  300 FORMAT(/,10X,'LGF = ',I2,5X,'IHEAD = ',I2,5X,'IPF1 = ',I2,
     * 5X,'IPF2 = ',I2,/)
C
      IF(LGF.LT.0.OR.LGF.GT.3)GO TO 7000
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','XF',3X,'XMAX',3X,'XI',3X,'XXI',3X,
     * 'XDIZ',/)
      READ(IRE,*,END=9000)XF,XMAX,XI,XXI,XDIZ
C
C     XF - X AXIS PLOT ORIGIN VALUE (SMALLEST OR MOST NEGATIVE
C          NUMBER)
C     XMAX - MAXIMUM X VALUE TO BE PLOTTED
C     XI - INCREMENT IN X VALUE FOR TICKS
C          = 0 SET TO (XMAX-XF)/10.0 FOR LINEAR PLOT
C          = 0 SET TO XF*10**CYCLE FOR LOG PLOT
C     XXI - INCREMENT IN X VALUE FOR LABELED TICKS
C           (MUST BE INTEGER MULTIPLE OF XI)
C           = 0 SET TO 2 XI FOR LINEAR PLOT
C           = 0 SET TO 10 XI FOR LOG PLOT
C     XDIZ - TRACE PLOTTING INCREMENT (IN X VALUE/CM)
C          - LINEAR PLOT (X/CM VALUE)
C            = 0 SET TO XI
C          - LOG PLOT (TOTAL X CM FOR ALL CYCLES)
C            < 1 SET TO 6 INCHES
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','YF',3X,'YMAX',3X,'YI',3X,'YYI',3X,
     * 'YDIZ',/)
      READ(IRE,*,END=9000)YF,YMAX,YI,YYI,YDIZ
C
C     YF - Y AXIS PLOT ORIGIN VALUE (SMALLEST OR MOST NEGATIVE
C          VALUE)
C     YMAX - MAXIMUM Y VALUE TO BE PLOTTED
C     YI - INCREMENT IN Y VALUE FOR TICKS
C          = 0 SET TO (YMAX-YF)/10.0 FOR LINEAR PLOT
C          = 0 SET TO YF*10**CYCLE FOR LOG PLOT
C     YYI - INCREMENT IN Y VALUE FOR LABELED TICKS
C           (MUST BE INTEGER MULTIPLE OF YI)
C           = 0 SET TO 2 YI FOR LINEAR PLOT
C           = 0 SET TO 10 YI FOR LOG PLOT
C     YDIZ - TRACE PLOTTING INCREMENT (IN Y VALUE/CM)
C          - LINEAR PLOT (Y/CM VALUE)
C            = 0 SET TO YI
C          - LOG PLOT (TOTAL Y CM FOR ALL CYCLES)
C            < 1 SET TO 6 INCHES
C
      IF(XF.GE.XMAX)GO TO 7000
      IF(YF.GE.YMAX)GO TO 7000
      IF(LGF.NE.0.AND.LGF.NE.2)GO TO 700
C
C     LINEAR PLOT FOR X AXIS
C
```

```
      IF(XI.EQ.0.0)XI=ABS(XMAX-XF)/10.0
      IF(XXI.EQ.0.0)XXI=2.0*XI
      IF(XDIZ.LE.0.0)XDIZ=XI/(0.60*CMPIN)
      SCALX=1.0/(XDIZ*CMPIN)
C
  600 IF(LGF.NE.0.AND.LGF.NE.1)GO TO 800
C
C     LINEAR PLOT FOR Y AXIS
C
      IF(YI.EQ.0.0)YI=ABS(YMAX-YF)/10.0
      IF(YYI.EQ.0.0)YYI=2.0*YI
      IF(YDIZ.LE.0.0)YDIZ=YI/(0.60*CMPIN)
      SCALY=1.0/(YDIZ*CMPIN)
      WIN=ABS(YMAX-YF)*SCALY+3.0
      IF(WIN.LT.12.5)WIN=12.5
      GO TO 850
C
C     LOG PLOT FOR X AXIS
C
  700 IF(XF.LE.0.0)XF=1.0
      XI=XF
      XXI=10.0*XI
      SCALX=XDIZ/CMPIN
      IF(SCALX.LE.1.0)SCALX=6.0
      GO TO 600
C
C     LOG PLOT FOR Y AXIS
C
  800 IF(YF.LE.0.0)YF=1.0
      YI=YF
      YYI=10.0*YI
      SCALY=YDIZ/CMPIN
      IF(SCALY.LE.1.0)SCALY=6.0
      WIN=SCALY+3.0
      IF(WIN.LT.12.5)WIN=12.5
C
  850 WRITE(IWR,900)
      WRITE(IWR2,900)
  900 FORMAT(/,10X,'PLOT PARAMETERS',/)
      WRITE(IWR,1000)XF,XMAX,XI,XXI,XDIZ
      WRITE(IWR2,1000)XF,XMAX,XI,XXI,XDIZ
 1000 FORMAT(/,10X,'XF = ',F10.2,5X,'XMAX = ',F10.2,/,10X,'XI = ',F10.2,
     * 5X,'XXI = ',F11.2,3X,'XDIZ = ',F10.2,/)
      WRITE(IWR,1100)YF,YMAX,YI,YYI,YDIZ
      WRITE(IWR2,1100)YF,YMAX,YI,YYI,YDIZ
 1100 FORMAT(/,10X,'YF = ',F10.2,5X,'YMAX = ',F10.2,/,10X,'YI = ',F10.2,
     * 5X,'YYI = ',F11.2,3X,'YDIZ = ',F10.2,/)
C
      WRITE(IWR2,2000)
 2000 FORMAT(/,'++ INPUT: ','MODEP',3X,'FAC',3X,'SIZ1',3X,'SIZ2',3X,
     * 'SIZ3',/)
      READ(IRE,*,END=9000)MODEP,FAC,SIZ1,SIZ2,SIZ3
C
C     MODEP - MODE FOR OPENING PLOT SYSTEM
C             = 0 SET TO 3
C             = 1 IMMEDIATE MODE
C             = 2 DEFERRED MODE
C             = 3 EDIT MODE
```

```
C     FAC - PLOT SCALE FACTOR
C           = 0 SET TO 1.0
C     SIZ1 - NUMBER SIZE FOR X AND Y AXES (IN INCHES)
C            = 0 SET T0 0.14
C     SIZ2 - TITLE SIZE FOR X AND Y AXES (IN INCHES)
C            = 0 SET TO 0.28
C     SIZ3 - SIZE FOR MAIN TITLE (IN INCHES)
C            = 0 SET TO 0.28
C
      IF(MODEP.LE.0)MODEP=3
      IF(FAC.LE.0.0)FAC=1.0
      WRITE(IWR,2200)MODEP,FAC,SIZ1,SIZ2,SIZ3
      WRITE(IWR2,2200)MODEP,FAC,SIZ1,SIZ2,SIZ3
 2200 FORMAT(/,10X,'MODEP = ',I2,5X,'FAC = ',F7.3,//,10X,'SIZ1 = ',
     * F7.3,5X,'SIZ2 = ',F7.3,5X,'SIZ3 = ',F7.3,/)
C
      IF(IPLFL.NE.0)GO TO 2500
C
      CALL OPNPLT(MODEP,WIN,110)
C
 2500 IF(IHEAD.EQ.0)GO TO 3000
C
      WRITE(IWR2,2700)
 2700 FORMAT(/,'++ INPUT: ','AXTIT',/)
      READ(IRE,2800,END=9000)AXTIT
 2800 FORMAT(A20)
C
      WRITE(IWR2,2900)
 2900 FORMAT(/,'++ INPUT: ','AYTIT',/)
      READ(IRE,2800,END=9000)AYTIT
C
      WRITE(IWR2,2950)
 2950 FORMAT(/,'++ INPUT: ','AMTIT',/)
      READ(IRE,2800,END=9000)AMTIT
C
C     AXTIT - X AXIS TITLE (20 CHARACTERS)
C     AYTIT - Y AXIS TITLE (20 CHARACTERS)
C     AMTIT - MAIN PLOT TITLE (20 CHARACTERS)
C
C     PLOT AXES
C
 3000 CALL XYAX(XF,XMAX,SCALX,XI,XXI,YF,YMAX,SCALY,YI,YYI,IPF1,IPF2,
     * FAC,AXTIT,AYTIT,AMTIT,LGF,SIZ1,SIZ2,SIZ3)
C
      IPLFL=1
C
C     READ PLOT PARAMETERS
C
 3100 WRITE(IWR2,3500)
 3500 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'ICHAN2',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2
C
C     ICHAN1 - DATA CHANNEL NUMBER FOR X DATA
C              = 0 SET TO 1
C     ICHAN2 - DATA CHANNEL NUMBER FOR Y DATA
C              = 0 SET TO 2
C
      IF(ICHAN1.LE.0)ICHAN1=1
```

```
      IF(ICHAN2.LE.0)ICHAN1=2
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NACHLS)GO TO 7000
      WRITE(IWR,3600)ICHAN1,ICHAN2
      WRITE(IWR2,3600)ICHAN1,ICHAN2
 3600 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      WRITE(IWR2,4000)
 4000 FORMAT(/,'++ INPUT: ','IFIR',3X,'ILAS',/)
      READ(IRE,*,END=9000)IFIR,ILAS
C
C     IFIR - FIRST DATA POINT TO PLOT
C            = 0 SET TO 1
C     ILAS - LAST DATA POINT TO PLOT
C            = 0 SET TO NUMBER OF SAMPLES IN CHANNEL
C
      IF(IFIR.LE.0)IFIR=1
      IF(ILAS.LE.0)ILAS=JSAMP(ICHAN1)
      IF(ILAS.GT.JSAMP(ICHAN1))ILAS=JSAMP(ICHAN1)
      IF(ILAS.GT.JSAMP(ICHAN2))ILAS=JSAMP(ICHAN2)
      IF(IFIR.GT.ILAS)GO TO 7000
      NP=ILAS-IFIR+1
      WRITE(IWR,4200)IFIR,ILAS
      WRITE(IWR2,4200)IFIR,ILAS
 4200 FORMAT(/,10X,'IFIR = ',I6,5X,'ILAS = ',I6,/)
C
      WRITE(IWR2,5000)
 5000 FORMAT(/,'++ INPUT: ','LINE',3X,'MARK',/)
      READ(IRE,*,END=9000)LINE,MARK
C
C     LINE - FLAG FOR CONTINUOUS LINE
C            = 0 NO LINE JOINING POINTS
C            = 1 LINE JOINING POINTS
C     MARK - SYMBOL CODE USED FOR DATA POINTS
C            = 0 DON'T PLOT SYMBOL
C            = 1 PLOT X
C            = 2 PLOT O
C            = 3 PLOT *
C            = 4 PLOT +
C
      IF(MARK.LT.0)MARK=0
      IF(MARK.GT.4)MARK=4
      WRITE(IWR,5200)LINE,MARK
      WRITE(IWR2,5200)LINE,MARK
 5200 FORMAT(/,10X,'LINE = ',I2,5X,'MARK = ',I3,/)
C
      INDEX1=(ICHAN1-1)*LENG+IFIR
      INDEX2=(ICHAN2-1)*LENG+IFIR
C
      CALL XYDAPL(DATA(INDEX1),DATA(INDEX2),LINE,MARK,NP,XF,XMAX,
     * SCALX,YF,YMAX,SCALY,FAC,LGF)
C
      WRITE(IWR2,6000)
 6000 FORMAT(/,10X,'PLOT MORE DATA CHANNELS ON SAME AXES?',//,
     * '++ INPUT: ','(y/n)',/)
      READ(IRE,6100,END=9000)KY
 6100 FORMAT(A1)
C
```

```
C     KY - y OR n (PLOT MORE DATA?)
C
      IF(NY.EQ.KY)GO TO 3100
C
C     RETURN CONTROL TO SCREEN
C
      CALL HLDPLT(5.0,1.0,MODE,MSG)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT PLOTTED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY PLOTTED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE XYPSPC(X,Y,NP,XF,XMAX,YF,YMAX)
C
C*********************************************************************
C
C     PLOT X-Y SET OF DATA
C
C     X - AMPLITUDE, POWER, OR PHASE - INPUT
C     Y - PERIOD OR FREQUENCY - INPUT
C     NP - NUMBER OF POINTS TO PLOT - INPUT
C     XF - X AXIS PLOT ORIGIN VALUE - INPUT
C     XMAX - MAXIMUM X VALUE TO BE PLOTTED - INPUT
C     YF - Y AXIS PLOT ORIGIN VALUE - INPUT
C     YMAX - MAXIMUM Y VALUE TO BE PLOTTED - INPUT
C            (PLOT SIZE MUST BE 1.5 INCHES LARGER THAN THIS)
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 6, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/PCKF/ICHECK,IPLFL
C
      DIMENSION X(1),Y(1)
C
      CHARACTER*80 MSG
      CHARACTER*20 AXTIT,AYTIT,AMTIT,IBLK
C
      DATA IBLK/'                    '/
C
C     SET UP PLOT PARAMETERS
C
      CMPIN=2.54
      MODE=0
      AXTIT=IBLK
      AYTIT=IBLK
      AMTIT=IBLK
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','LGF',3X,'IHEAD',3X,'IPF1',3X,'IPF2',/)
      READ(IRE,*,END=9000)LGF,IHEAD,IPF1,IPF2
C
C     LGF - FLAG FOR LOG PLOT
C           = 1 LOG PLOT ON X AXIS
C           = 3 LOG PLOT ON X AND Y AXIS
C     IHEAD - FLAG FOR HEADER PLOT
C             = 0 DON'T PLOT HEADER
C             = 1 PLOT HEADER
C     IPF1 - PLOT FLAG FOR NUMBERS ON X AXIS
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     IPF2 - PLOT FLAG FOR NUMBERS ON Y AXIS
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C
      IF(IHEAD.LT.0.OR.IHEAD.GT.1)IHEAD=0
      IF(IPF1.LT.0.OR.IPF1.GT.1)IPF1=1
```

```
      IF(IPF2.LT.0.OR.IPF2.GT.1)IPF2=1
      WRITE(IWR,300)LGF,IHEAD,IPF1,IPF2
      WRITE(IWR2,300)LGF,IHEAD,IPF1,IPF2
  300 FORMAT(/,10X,'LGF = ',I2,5X,'IHEAD = ',I2,5X,'IPF1 = ',I2,
     * 5X,'IPF2 = ',I2,/)
C
      IF(LGF.LT.0.OR.LGF.GT.3)GO TO 7000
C
      WRITE(IWR2,310)
  310 FORMAT(/,'++ INPUT: ','ISF',3X,'NCY',/)
      READ(IRE,*,END=9000)ISF,NCY
C
C     ISF - FLAG FOR SPECTRAL AMPLITUDE RANGE TO PLOT
C           = 0 LOG PLOT OF ENTIRE RANGE
C               (LGF CANNOT BE 0 OR 2)
C           = 1 LOG PLOT OF MAXIMUM VALUE AND NCY CYCLES DOWN
C           = 2 READ IN RANGE TO PLOT
C     NCY - NUMBER OF CYCLES TO PLOT
C           = 0 SET TO 1 IF ISF = 1
C
      IF(NCY.LE.0.AND.ISF.EQ.1)NCY=1
      WRITE(IWR,320)ISF,NCY
      WRITE(IWR2,320)ISF,NCY
  320 FORMAT(/,10X,'ISF = ',I2,5X,'NCY = ',I2,/)
C
      IF(ISF.GE.2)GO TO 390
      IF(LGF.EQ.0.OR.LGF.EQ.2)LGF=3
C
      WRITE(IWR2,330)
  330 FORMAT(/,'++ INPUT: ','XDIZ',3X,'YI',3X,'YYI',3X,'YDIZ',/)
      READ(IRE,*,END=9000)XDIZ,YI,YYI,YDIZ
C
C     XDIZ - TRACE PLOTTING INCREMENT (IN X VALUE/CM)
C          - LINEAR PLOT (CM/X VALUE)
C            = 0 SET TO XI
C          - LOG PLOT (TOTAL X CM FOR ALL CYCLES)
C            = 0 SET TO 6 INCHES
C     YI - INCREMENT IN Y VALUE FOR TICKS
C          = 0 SET TO (YMAX-YF)/10.0 FOR LINEAR PLOT
C          = 0 SET TO YF*10**CYCLE FOR LOG PLOT
C     YYI - INCREMENT IN Y VALUE FOR LABELED TICKS
C           (MUST BE INTEGER MULTIPLE OF YI)
C           = 0 SET TO 2 YI FOR LINEAR PLOT
C           = 0 SET TO 10 YI FOR LOG PLOT
C     YDIZ - TRACE PLOTTING INCREMENT (IN Y VALUE/CM)
C          - LINEAR PLOT (Y/CM VALUE)
C            = 0 SET TO YI
C          - LOG PLOT (TOTAL Y CM FOR ALL CYCLES)
C            < 1 SET TO 6 INCHES
C
      WRITE(IWR,340)XDIZ,YI,YYI,YDIZ
      WRITE(IWR2,340)XDIZ,YI,YYI,YDIZ
  340 FORMAT(/,10X,'XDIZ = ',F10.3,5X,'YI = ',F10.3,5X,'YYI = ',F10.3,
     * /,10X,'YDIZ = ',F10.3,/)
C
      IF(ISF.LE.0)GO TO 700
C
C     DETERMINE AMPLITUDE, POWER, OR PHASE PLOT ORIGIN VALUE
```

```
C
      CALL CYCLES(XF,XMAX,NCY,ZFT,ZFL,ZMAXT,ZMAXU,XF)
C
      GO TO 700
C
  390 WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','XF',3X,'XMAX',3X,'XI',3X,'XXI',3X,
     * 'XDIZ',/)
      READ(IRE,*,END=9000)XF,XMAX,XI,XXI,XDIZ
C
C     XF - X AXIS PLOT ORIGIN VALUE (SMALLEST VALUE)
C     XMAX - MAXIMUM X VALUE TO BE PLOTTED
C     XI - INCREMENT IN X VALUE FOR TICKS
C          = 0 SET TO (XMAX-XF)/10.0 FOR LINEAR PLOT
C          = 0 SET TO XF*10**CYCLE FOR LOG PLOT
C     XXI - INCREMENT IN X VALUE FOR LABELED TICKS
C           (MUST BE INTEGER MULTIPLE OF XI)
C           = 0 SET TO 2 XI FOR LINEAR PLOT
C           = 0 SET TO 10 XI FOR LOG PLOT
C     XDIZ - TRACE PLOTTING INCREMENT (IN X VALUE/CM)
C          - LOG PLOT (TOTAL X CM FOR ALL CYCLES)
C            < 1 SET TO 6 INCHES
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','YF',3X,'YMAX',3X,'YI',3X,'YYI',3X,
     * 'YDIZ',/)
      READ(IRE,*,END=9000)YF,YMAX,YI,YYI,YDIZ
C
C     YF - Y AXIS PLOT ORIGIN VALUE
C     YMAX - MAXIMUM Y VALUE TO BE PLOTTED
C     YI - INCREMENT IN Y VALUE FOR TICKS
C          = 0 SET TO (YMAX-YF)/10.0 FOR LINEAR PLOT
C          = 0 SET TO YF*10**CYCLE FOR LOG PLOT
C     YYI - INCREMENT IN Y VALUE FOR LABELED TICKS
C           (MUST BE INTEGER MULTIPLE OF YI)
C           = 0 SET TO 2 YI FOR LINEAR PLOT
C           = 0 SET TO 10 YI FOR LOG PLOT
C     YDIZ - TRACE PLOTTING INCREMENT (IN Y VALUE/CM)
C          - LINEAR PLOT (Y/CM VALUE)
C            = 0 SET TO YI
C          - LOG PLOT (TOTAL Y CM FOR ALL CYCLES)
C            < 1 SET TO 6 INCHES
C
      GO TO 700
C
  600 IF(LGF.NE.1)GO TO 800
C
C     LINEAR PLOT FOR Y AXIS
C
      IF(YI.LE.0.0)YI=ABS(YMAX-YF)/10.0
      IF(YYI.EQ.0.0)YYI=2.0*YI
      IF(YDIZ.LE.0.0)YDIZ=YI/(0.60*CMPIN)
      SCALY=1.0/(YDIZ*CMPIN)
      WIN=ABS(YMAX-YF)*SCALY+3.0
      IF(WIN.LT.12.5)WIN=12.5
      GO TO 850
C
C     LOG PLOT FOR X AXIS
```

```
C
  700 IF(XF.LE.0.0)XF=1.0
      XI=XF
      XXI=10.0*XI
      SCALX=XDIZ/CMPIN
      IF(SCALX.LE.1.0)SCALX=6.0
      GO TO 600
C
C     LOG PLOT FOR Y AXIS
C
  800 IF(YF.LE.0.0)YF=1.0
      YI=YF
      YYI=10.0*YI
      SCALY=YDIZ/CMPIN
      IF(SCALY.LE.1.0)SCALY=6.0
      WIN=SCALY+3.0
      IF(WIN.LT.12.5)WIN=12.5
C
  850 WRITE(IWR,900)
      WRITE(IWR2,900)
  900 FORMAT(/,10X,'PLOT PARAMETERS',/)
      WRITE(IWR,1000)XF,XMAX,XI,XXI,XDIZ
      WRITE(IWR2,1000)XF,XMAX,XI,XXI,XDIZ
 1000 FORMAT(/,10X,'XF = ',F8.2,5X,'XMAX = ',F8.2,/,10X,'XI = ',F8.2,
     * 5X,'XXI = ',F9.2,3X,'XDIZ = ',F8.2,/)
      WRITE(IWR,1100)YF,YMAX,YI,YYI,YDIZ
      WRITE(IWR2,1100)YF,YMAX,YI,YYI,YDIZ
 1100 FORMAT(/,10X,'YF = ',F8.2,5X,'YMAX = ',F8.2,/,10X,'YI = ',F8.2,
     * 5X,'YYI = ',F9.2,3X,'YDIZ = ',F8.2,/)
C
      WRITE(IWR2,2000)
 2000 FORMAT(/,'++ INPUT: ','MODEP',3X,'FAC',3X,'SIZ1',3X,'SIZ2',3X,
     * 'SIZ3',/)
      READ(IRE,*,END=9000)MODEP,FAC,SIZ1,SIZ2,SIZ3
C
C     MODEP - MODE FOR OPENING PLOT SYSTEM
C             = 0 SET TO 3
C             = 1 IMMEDIATE MODE
C             = 2 DEFERRED MODE
C             = 3 EDIT MODE
C     FAC - PLOT SCALE FACTOR
C           = 0 SET TO 1.0
C     SIZ1 - NUMBER SIZE FOR X AND Y AXES (IN INCHES)
C            = 0 SET T0 0.14
C     SIZ2 - TITLE SIZE FOR X AND Y AXES (IN INCHES)
C            = 0 SET TO 0.28
C     SIZ3 - SIZE FOR MAIN TITLE (IN INCHES)
C            = 0 SET TO 0.28
C
      IF(MODEP.LE.0)MODEP=3
      IF(FAC.LE.0.0)FAC=1.0
      WRITE(IWR,2200)MODEP,FAC,SIZ1,SIZ2,SIZ3
      WRITE(IWR2,2200)MODEP,FAC,SIZ1,SIZ2,SIZ3
 2200 FORMAT(/,10X,'MODEP = ',I2,5X,'FAC = ',F7.3,//,10X,'SIZ1 = ',
     * F7.3,5X,'SIZ2 = ',F7.3,5X,'SIZ3 = ',F7.3,/)
C
      IF(IPLFL.NE.0)GO TO 2500
C
```

```
      CALL OPNPLT(MODEP,WIN,111)
C
 2500 IF(IHEAD.EQ.0)GO TO 3000
C
      WRITE(IWR2,2700)
 2700 FORMAT(/,'++ INPUT: ','AXTIT',/)
      READ(IRE,2800,END=9000)AXTIT
 2800 FORMAT(A20)
C
      WRITE(IWR2,2900)
 2900 FORMAT(/,'++ INPUT: ','AYTIT',/)
      READ(IRE,2800,END=9000)AYTIT
C
      WRITE(IWR2,2950)
 2950 FORMAT(/,'++ INPUT: ','AMTIT',/)
      READ(IRE,2800,END=9000)AMTIT
C
C     AXTIT - X AXIS TITLE (20 CHARACTERS MAXIMUM)
C     AYTIT - Y AXIS TITLE (20 CHARACTERS MAXIMUM)
C     AMTIT - MAIN PLOT TITLE (20 CHARACTERS MAXIMUM)
C
C     PLOT AXES
C
 3000 CALL XYAX(XF,XMAX,SCALX,XI,XXI,YF,YMAX,SCALY,YI,YYI,IPF1,IPF2,
     * FAC,AXTIT,AYTIT,AMTIT,LGF,SIZ1,SIZ2,SIZ3)
C
C     READ PLOT PARAMETERS
C
      WRITE(IWR2,5000)
 5000 FORMAT(/,'++ INPUT: ','LINE',3X,'MARK',/)
      READ(IRE,*,END=9000)LINE,MARK
C
C     LINE - FLAG FOR CONTINUOUS LINE
C            = 0 NO LINE JOINING POINTS
C            = 1 LINE JOINING POINTS
C     MARK - SYMBOL CODE USED FOR DATA POINTS
C            = 0 DON'T PLOT SYMBOL
C            = 1 PLOT X
C            = 2 PLOT O
C            = 3 PLOT *
C            = 4 PLOT +
C
      IF(MARK.LT.0)MARK=0
      IF(MARK.GT.4)MARK=4
      WRITE(IWR,5200)LINE,MARK
      WRITE(IWR2,5200)LINE,MARK
 5200 FORMAT(/,10X,'LINE = ',I2,5X,'MARK = ',I3,/)
C
      CALL XYDAPL(X,Y,LINE,MARK,NP,XF,XMAX,SCALX,YF,YMAX,SCALY,FAC,
     * LGF)
C
      IPLFL=1
C
C     RETURN CONTROL TO SCREEN
C
      CALL HLDPLT(5.0,1.0,MODE,MSG)
C
      GO TO 8000
```

```
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT PLOTTED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY PLOTTED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE ZER(IFL,IFIR,ILAS,ICHAN)
C
C*************************************************************************
C
C     ZERO ANY SPECIFIED PART OF ANY ARRAY
C
C     IFL - FLAG FOR DATA ARRAY TO ZERO - INPUT
C           = 0 ZERO PART OF DATA ARRAY
C           = 1 ZERO PART OF BX ARRAY
C     IFIR - FIRST POINT TO ZERO - INPUT
C     ILAS - LAST POINT TO ZERO - INPUT
C     ICHAN - CHANNEL TO ZERO - INPUT
C             = 0 ZERO ALL CHANNELS IF DATA ARRAY SELECTED
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 20, 1993  COLORADO
C
C*************************************************************************
C
      COMMON/IN/BX
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION DATA(1),BX(1)
C
      IF(IFL.LE.0)GO TO 200
C
      DO 100 J=IFIR,ILAS
      BX(J)=0.0
  100 CONTINUE
C
      GO TO 9000
C
  200 IF(ICHAN.LE.0)GO TO 500
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 2000
C
  500 NCH1=1
      NCH2=NCHLS
C
 2000 DO 3000 I=NCH1,NCH2
      INDEX=(I-1)*LENG
C
      DO 2500 J=IFIR,ILAS
      IN=INDEX+J
      DATA(IN)=0.0
 2500 CONTINUE
C
 3000 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE ZEROED
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR ZEROING UNUSED PART OF ALL DATA CHANNELS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 28, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
C
      COMPLEX RPOLES,RZEROS
C
      IFL=0
      ILAS=LENG
C
      DO 3000 I=1,NACHLS
      IFIR=JSAMP(I)+1
C
      CALL ZER(IFL,IFIR,ILAS,I)
C
 3000 CONTINUE
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'ARRAYS SUCCESSFULLY ZEROED',/)
C
      RETURN
      END
```

```
      SUBROUTINE ZEROX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR ZEROING ANY SPECIFIED PART OF ANY ARRAY
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 20, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
C
C     READ ZERO PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','IFL',3X,'ICHAN',3X,'IFIR',3X,'ILAS',/)
      READ(IRE,*,END=9000)IFL,ICHAN,IFIR,ILAS
C
C     IFL - FLAG FOR DATA ARRAY TO ZERO
C           = 0 ZERO PART OF DATA ARRAY
C           = 1 ZERO PART OF BX ARRAY
C     ICHAN - CHANNEL TO ZERO
C             = 0 ZERO ALL CHANNELS IF DATA ARRAY SELECTED
C     IFIR - FIRST POINT TO ZERO
C            = 0 SET TO 1
C     ILAS - LAST POINT TO ZERO
C            = 0 SET TO LENG FOR DATA OR IDEM1 FOR BX
C
      IF(IFIR.LE.0)IFIR=1
      IF(IFL.EQ.1)ICHAN=1
      IF(IFL.EQ.1.AND.ILAS.LE.0)ILAS=IDEM1
      IF(IFL.EQ.1.AND.ILAS.GT.IDEM1)ILAS=IDEM1
      IF(IFL.LE.0.AND.ILAS.LE.0)ILAS=LENG
      IF(IFL.LE.0.AND.ILAS.GT.LENG)ILAS=LENG
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'ZERO PARAMETERS',/)
      WRITE(IWR,300)IFL,IFIR,ILAS,ICHAN
      WRITE(IWR2,300)IFL,IFIR,ILAS,ICHAN
  300 FORMAT(/,10X,'IFL = ',I2,5X,'IFIR = ',I5,5X,'ILAS = ',I6,5X,
     * 'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
      CALL ZER(IFL,IFIR,ILAS,ICHAN)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'ARRAY NOT ZEROED',/)
      GO TO 9000
```

```
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'ARRAY SUCCESSFULLY ZEROED',/)
C
 9000 RETURN
      END
```