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

KEY ANALYSIS SYSTEM

Volume 2 - Subroutines

Robert P. Massé

March 29, 1994

Version 1.0

Open-File Report 94-184

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature. Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

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.

```
      PROGRAM KEY
C
C*****************************************************************************
C
C     KEY ANALYSIS SYSTEM
C
C     PROCESSING SYSTEM FOR BROADBAND SEISMIC NETWORK DATA.
C     SYSTEM PERFORMS DATA READING, QUALITY CONTROL, INSTRUMENT
C     RESPONSE CORRECTION, FILTERING, MATCHED FILTERING, MOVING
C     WINDOW ANALYSIS, BEAMFORMING, SPECTRAL COMPUTATIONS, DISTANCE-
C     AZIMUTH COMPUTATIONS, TRAVEL TIME COMPUTATIONS, THEORETICAL
C     MODELING, PLOTTING, AND DATA STORAGE FUNCTIONS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 1, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      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
      COMMON/PCKF/ICHECK,IPLFL
      COMMON/XDATA/DATA
C
      DIMENSION DATA(200000),IOPS(100),BX(60005),EXFLAG(10)
C
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
C     DEFINE DATA DIMENSION
C
      IDEM=200000
      IDEM1=60000
      NACHLS=0
      NCHLS=12
      LENG=IDEM/NCHLS
C
C     SET UP VARIOUS CONSTANTS
C
      CALL DZCON
C
      IPLFL=0
      IEXYR=0
C
C     SET READ AND PRINT UNITS
C
      IRE=5
      IWR=7
      IWR2=6
      LUN=15
      LUN2=10
      LUN3=8
```

```
      LUN4=9
C
      OPEN(IWR,FILE='keyout',STATUS='UNKNOWN')
C
C     READ AND EXECUTE ALL OPTIONS
C
      CALL AASEL
C
C     IF PLOTTING WAS DONE, TERMINATE PLOT
C
      CALL ENDPF
C
C     CLOSE ALL UNITS USED
C
 9000 CLOSE(IWR)
      IF(IOPS(50).NE.0)CLOSE(LUN)
      IF(IOPS(51).NE.0)CLOSE(LUN2)
C
      END
```

```
      SUBROUTINE AASEL
C
C*********************************************************************
C
C     DECIPHER AND EXECUTE PROGRAMS OPTIONS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 4, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/OPTION/IOPS
      COMMON/PCKF/ICHECK,IPLFL
C
      DIMENSION IOPS(1)
C
      INTEGER*4 BEAM,DESP,FILT,MOVE,PLOX,LAST,TAPE,SPEC,POLA
      INTEGER*4 MEAN,TREN,DECI,DIST,RDST,RDHD,MATC,VESP,INST
      INTEGER*4 GROU,SWTT,SCAL,ROTA,SHIF,CROS,RECO,RECT,SIGN
      INTEGER*4 RWDS,LWDS,RDDP,RESP,PRPL,XYPL,PRPT,ZERO,FIRS
      INTEGER*4 NEXT,READ,TEST,REWI,LSHD,REOR,WRFL,LSDA,LSDP
      INTEGER*4 LSSA,CLOS,JDAY,ENDP,BWTT,NSNR,RDDA,RDEV,LSEV
      INTEGER*4 DAAD,DASU,LSDS,APVL,MNMX,RDSA,DELE,LSRP,ZEND
      INTEGER*4 COPY,STRP,LSTT
C
      DATA BEAM/'beam'/,DESP/'desp'/,FILT/'filt'/,MOVE/'move'/
      DATA PLOX/'plot'/,LAST/'last'/,TAPE/'tape'/,SPEC/'spec'/
      DATA POLA/'pola'/,MEAN/'mean'/,TREN/'tren'/,DECI/'deci'/
      DATA DIST/'dist'/,RDST/'rdst'/,RDHD/'rdhd'/,MATC/'matc'/
      DATA VESP/'vesp'/,INST/'inst'/,GROU/'grou'/,SWTT/'swtt'/
      DATA SCAL/'scal'/,ROTA/'rota'/,SHIF/'shif'/,CROS/'cros'/
      DATA RECO/'reco'/,RECT/'rect'/,SIGN/'sign'/,RWDS/'rwds'/
      DATA LWDS/'lwds'/,RDDP/'rddp'/,RESP/'resp'/,PRPL/'prpl'/
      DATA XYPL/'xypl'/,PRPT/'prpt'/,ZERO/'zero'/,FIRS/'firs'/
      DATA NEXT/'next'/,READ/'read'/,TEST/'test'/,REWI/'rewi'/
      DATA LSHD/'lshd'/,REOR/'reor'/,WRFL/'wrfl'/,LSDA/'lsda'/
      DATA LSDP/'lsdp'/,LSSA/'lssa'/,CLOS/'clos'/,JDAY/'jday'/
      DATA ENDP/'endp'/,BWTT/'bwtt'/,NSNR/'nsnr'/,RDDA/'rdda'/
      DATA RDEV/'rdev'/,LSEV/'lsev'/,DAAD/'daad'/,DASU/'dasu'/
      DATA LSDS/'lsds'/,APVL/'apvl'/,MNMX/'mnmx'/,RDSA/'rdsa'/
      DATA DELE/'dele'/,LSRP/'lsrp'/,ZEND/'zend'/,COPY/'copy'/
      DATA STRP/'strp'/,LSTT/'lstt'/
C
   10 WRITE(IWR,15)
   15 FORMAT(////)
      WRITE(IWR2,20)
   20 FORMAT(50(/))
      WRITE(IWR,30)
      WRITE(IWR2,30)
   30 FORMAT(///,22X,'******** KEY PROGRAM SYSTEM ********',//)
C
C     INITIALIZE ALL OPTION FLAGS
C
      DO 100 I=1,100
```

```
      IOPS(I)=0
  100 CONTINUE
C
 1500 ICHECK=0
C
C     READ NEXT OPTION
C
 1510 WRITE(IWR,1520)
      WRITE(IWR2,1520)
 1520 FORMAT(/,10X,62('*'),/)
      WRITE(IWR2,1530)
 1530 FORMAT(/,'++ INPUT: ','NEXT DESIRED OPTION',/)
      READ(IRE,1550,END=3800)IOPT
C
C     IOPT - PROGRAM OPTION SELECTED
C
 1550 FORMAT(A4)
      WRITE(IWR,1570)IOPT
      WRITE(IWR2,1570)IOPT
 1570 FORMAT(/,10X,'PROGRAM OPTION SELECTED - ',A4,/)
C
      IF(IOPT.EQ.FIRS)GO TO 10
      IF(IOPT.EQ.NEXT)GO TO 1510
      IF(IOPT.EQ.READ)GO TO 1700
      IF(IOPT.EQ.DESP)GO TO 1800
      IF(IOPT.EQ.APVL)GO TO 1900
      IF(IOPT.EQ.XYPL)GO TO 2000
      IF(IOPT.EQ.BEAM)GO TO 2100
      IF(IOPT.EQ.RDDP)GO TO 2200
      IF(IOPT.EQ.MNMX)GO TO 2300
      IF(IOPT.EQ.GROU)GO TO 2400
      IF(IOPT.EQ.LSDS)GO TO 2500
      IF(IOPT.EQ.SHIF)GO TO 2600
      IF(IOPT.EQ.DASU)GO TO 2700
      IF(IOPT.EQ.DAAD)GO TO 2800
      IF(IOPT.EQ.RECT)GO TO 2900
      IF(IOPT.EQ.MATC)GO TO 3100
      IF(IOPT.EQ.FILT)GO TO 3200
      IF(IOPT.EQ.PRPT)GO TO 3300
      IF(IOPT.EQ.PRPL)GO TO 3400
      IF(IOPT.EQ.INST)GO TO 3500
      IF(IOPT.EQ.RESP)GO TO 3600
      IF(IOPT.EQ.SPEC)GO TO 3700
      IF(IOPT.EQ.LAST)GO TO 3800
      IF(IOPT.EQ.SIGN)GO TO 3900
      IF(IOPT.EQ.LSDA)GO TO 4000
      IF(IOPT.EQ.TAPE)GO TO 4100
      IF(IOPT.EQ.TEST)GO TO 4200
      IF(IOPT.EQ.ZERO)GO TO 4300
      IF(IOPT.EQ.LSSA)GO TO 4400
      IF(IOPT.EQ.POLA)GO TO 4500
      IF(IOPT.EQ.CLOS)GO TO 4600
      IF(IOPT.EQ.REWI)GO TO 4700
      IF(IOPT.EQ.MEAN)GO TO 4800
      IF(IOPT.EQ.TREN)GO TO 4900
      IF(IOPT.EQ.DECI)GO TO 5000
      IF(IOPT.EQ.DIST)GO TO 5100
      IF(IOPT.EQ.JDAY)GO TO 5200
```

```
      IF(IOPT.EQ.ENDP)GO TO 5300
      IF(IOPT.EQ.RDST)GO TO 5400
      IF(IOPT.EQ.PLOX)GO TO 5500
      IF(IOPT.EQ.BWTT)GO TO 5600
      IF(IOPT.EQ.NSNR)GO TO 5700
      IF(IOPT.EQ.RDDA)GO TO 5800
      IF(IOPT.EQ.RDEV)GO TO 5900
      IF(IOPT.EQ.LSEV)GO TO 6000
      IF(IOPT.EQ.ROTA)GO TO 6100
      IF(IOPT.EQ.SCAL)GO TO 6200
      IF(IOPT.EQ.WRFL)GO TO 6300
      IF(IOPT.EQ.CROS)GO TO 6400
      IF(IOPT.EQ.RDSA)GO TO 6500
      IF(IOPT.EQ.DELE)GO TO 6600
      IF(IOPT.EQ.LSDP)GO TO 6700
      IF(IOPT.EQ.LSRP)GO TO 6800
      IF(IOPT.EQ.ZEND)GO TO 6900
      IF(IOPT.EQ.COPY)GO TO 7000
      IF(IOPT.EQ.STRP)GO TO 7100
      IF(IOPT.EQ.RWDS)GO TO 7200
      IF(IOPT.EQ.LWDS)GO TO 7300
      IF(IOPT.EQ.VESP)GO TO 7400
      IF(IOPT.EQ.LSTT)GO TO 7500
      IF(IOPT.EQ.MOVE)GO TO 8100
      IF(IOPT.EQ.LSHD)GO TO 8200
      IF(IOPT.EQ.REOR)GO TO 8300
      IF(IOPT.EQ.RDHD)GO TO 8400
      IF(IOPT.EQ.RECO)GO TO 8600
C
      WRITE(IWR2,1600)
 1600 FORMAT(/,10X,'***** PROGRAM OPTION NOT RECOGNIZED *****',/)
      GO TO 3000
C
C     READ A FILE AND PLACE IN DATA ARRAY CHANNELS OPTION
C
 1700 CONTINUE
      CALL READG
      GO TO 1500
C
C     DESPIKE OPTION
C
 1800 CONTINUE
      CALL DESPIK
      GO TO 1500
C
C     CALCULATE APPARENT VELOCITY OPTION
C
 1900 CONTINUE
      CALL APVELX
      GO TO 1500
C
C     X-Y PLOT OPTION
C
 2000 CONTINUE
      CALL XYPLT
      GO TO 1500
C
C     BEAMFORMING OPTION
```

```
C
 2100 CONTINUE
      IVESP=0
      CALL BEAMX(IVESP)
      GO TO 1500
C
C     SET DATA PARAMETERS FOR STORAGE OPTION
C
 2200 CONTINUE
      CALL DPARA
      GO TO 1500
C
C     DETERMINE MINIMUM/MAXIMUM OF DATA OPTION
C
 2300 CONTINUE
      CALL MNMXEX
      GO TO 1500
C
C     DETERMINE SIGNAL GROUP VELOCITY OPTION
C
 2400 CONTINUE
      CALL UDISPX
      GO TO 1500
C
C     LIST DISTANCE-AZIMUTH OPTION
C
 2500 CONTINUE
      IFL=1
      NCH1=1
      CALL ADAZE(IFL,NCH1,NACHLS)
      GO TO 1500
C
C     SHIFT TIME SERIES OPTION
C
 2600 CONTINUE
      CALL SHIFTX
      GO TO 1500
C
C     DATA SUBTRACT OPTION
C
 2700 CONTINUE
      IFL=1
      CALL DAAS(IFL)
      GO TO 1500
C
C     DATA ADD OPTION
C
 2800 CONTINUE
      IFL=0
      CALL DAAS(IFL)
      GO TO 1500
C
C     RECTIFY TIME SERIES OPTION
C
 2900 CONTINUE
      CALL RECTIF
      GO TO 1500
C
```

```
C     SEARCH FOR FIRST, NEXT, OR LAST OPTION
C
 3000 CONTINUE
      WRITE(IWR2,3030)
 3030 FORMAT(/,'++ INPUT: ','SEARCHING FOR firs, next, OR last',
     * ' OPTION',/)
 3050 READ(IRE,1550,END=3800)IOPT
      IF(IOPT.EQ.FIRS)GO TO 10
      IF(IOPT.EQ.NEXT)GO TO 1510
      IF(IOPT.EQ.LAST)GO TO 3800
      GO TO 3000
C
C     MATCHED FILTER OPTION
C
 3100 CONTINUE
      CALL MATCHX
      GO TO 1500
C
C     FILTER OPTION
C
 3200 CONTINUE
      CALL FILTX
      GO TO 1500
C
C     PRINTER PLOT OF TIME SERIES OPTION
C
 3300 CONTINUE
      CALL PRPLTX
      GO TO 1500
C
C     PRINTER PLOT OPTION
C
 3400 CONTINUE
      CALL SPLOTX
      GO TO 1500
C
C     REMOVE OR ADD INSTRUMENT RESPONSE OPTION
C
 3500 CONTINUE
      CALL INSTX
      GO TO 1500
C
C     CALCULATE AND LIST INSTRUMENT RESPONSE OPTION
C
 3600 CONTINUE
      CALL RESPX
      GO TO 1500
C
C     FOURIER SPECTRA OPTION
C
 3700 CONTINUE
      CALL SPECTR
      GO TO 1500
C
C     LAST OPTION
C
 3800 WRITE(IWR,3810)
      WRITE(IWR2,3810)
```

```
 3810 FORMAT(///,10X,'PROGRAM RUN HAS BEEN COMPLETED',/)
      GO TO 9000
C
C     CALCULATE SYNTHETIC TEST SIGNAL OPTION
C
 3900 CONTINUE
      CALL SIGNAL
      GO TO 1500
C
C     LIST DATA POINTS OPTION
C
 4000 CONTINUE
      CALL DATAL
      GO TO 1500
C
C     TAPER OPTION
C
 4100 CONTINUE
      CALL TAPEX
      GO TO 1500
C
C     TEST OPTION
C
 4200 CONTINUE
      CALL TESTF
      GO TO 1500
C
C     ZERO DATA ARRAY OPTION
C
 4300 CONTINUE
      CALL ZEROX
      GO TO 1500
C
C     LIST NUMBER OF SAMPLES OPTION
C
 4400 CONTINUE
      CALL XSAMP
      GO TO 1500
C
C     REVERSE POLARITY OPTION
C
 4500 CONTINUE
      CALL REVPOL
      GO TO 1500
C
C     CLOSE LOGICAL UNIT OPTION
C
 4600 CONTINUE
      CALL CLOSEX
      GO TO 1500
C
C     REWIND FILE OPTION
C
 4700 CONTINUE
      CALL REWD
      GO TO 1500
C
C     REMOVE MEAN OPTION
```

```
C
 4800 CONTINUE
      IDEG=0
      CALL METX(IDEG)
      GO TO 1500
C
C     REMOVE TREND OPTION
C
 4900 CONTINUE
      IDEG=1
      CALL METX(IDEG)
      GO TO 1500
C
C     DECIMATE DATA OPTION
C
 5000 CONTINUE
      CALL DECIX
      GO TO 1500
C
C     CALCULATE DISTANCE-AZIMUTH OPTION
C
 5100 CONTINUE
      IOPS(2)=1
      CALL DAZEX
      GO TO 1500
C
C     CALCULATE JULIAN DAY OPTION
C
 5200 CONTINUE
      CALL JULDY
      GO TO 1500
C
C     END PLOT OPTION
C
 5300 CONTINUE
      CALL ENDPF
      GO TO 1500
C
C     READ START TIME FROM KEYBOARD OPTION
C
 5400 CONTINUE
      CALL SETST
      GO TO 1500
C
C     PLOT TIME SERIES OPTION
C
 5500 CONTINUE
      CALL SEIPLT
      GO TO 1500
C
C     LIST TRAVEL TIMES OPTION
C
 5600 CONTINUE
      CALL LISTTX
      GO TO 1500
C
C     SET UP FILE FOR REQUESTING NSN DATA OPTION
C
```

```
 5700 CONTINUE
      CALL NSNFIL
      GO TO 1500
C
C     READ DATA POINTS FROM KEYBOARD OPTION
C
 5800 CONTINUE
      CALL KEYBRD
      GO TO 1500
C
C     READ EVENT INFORMATION FROM KEYBOARD OPTION
C
 5900 CONTINUE
      CALL EVTRD
      GO TO 1500
C
C     LIST EVENT INFORMATION OPTION
C
 6000 CONTINUE
      CALL PREVT
      GO TO 1500
C
C     ROTATE HORIZONTAL COMPONENTS OPTION
C
 6100 CONTINUE
      CALL CALROT
      GO TO 1500
C
C     SCALE DATA OPTION
C
 6200 CONTINUE
      CALL CALEX
      GO TO 1500
C
C     WRITE DATA TO FILE OPTION
C
 6300 CONTINUE
      CALL WRDATX
      GO TO 1500
C
C     CROSSCORRELATION OPTION
C
 6400 CONTINUE
      CALL CROSSX
      GO TO 1500
C
C     READ NUMBER OF SAMPLES FROM KEYBOARD OPTION
C
 6500 CONTINUE
      CALL XSAMP2
      GO TO 1500
C
C     DELETE A CHANNEL OPTION
C
 6600 CONTINUE
      CALL DELEX
      GO TO 1500
C
```

```
C     LIST STORAGE PARAMETERS OPTION
C
 6700 CONTINUE
      CALL STORLT
      GO TO 1500
C
C     LIST RESPONSE POLES AND ZEROES OPTION
C
 6800 CONTINUE
      CALL LSRESX
      GO TO 1500
C
C     ZERO UNUSED PART OF DATA CHANNELS OPTION
C
 6900 CONTINUE
      CALL ZEROED
      GO TO 1500
C
C     COPY CHANNEL TO SEVERAL CHANNELS OPTION
C
 7000 CONTINUE
      CALL COPYF
      GO TO 1500
C
C     SET RESPONSE FROM MASTER FILE
C
 7100 CONTINUE
      CALL STRPX
      GO TO 1500
C
C     RAYLEIGH WAVE DISPERSION OPTION
C
 7200 CONTINUE
      LVRY=0
      CALL SDISP(LVRY)
      GO TO 1500
C
C     LOVE WAVE DISPERSION OPTION
C
 7300 CONTINUE
      LVRY=1
      CALL SDISP(LVRY)
      GO TO 1500
C
C     VESPAGRAM OPTION
C
 7400 CONTINUE
      GO TO 1500
C
C     LIST TRAVEL TIMES FOR CURRENT STATIONS OPTION
C
 7500 CONTINUE
      CALL LSTTX
      GO TO 1500
C
C     MOVE DATA IN CORE OPTION
C
 8100 CONTINUE
```

```
      CALL MOVEX
      GO TO 1500
C
C     LIST CHANNEL HEADER OPTION
C
 8200 CONTINUE
      ICHAN=0
      CALL PRHEAD(ICHAN)
      GO TO 1500
C
C     REORDER CHANNELS OPTION
C
 8300 CONTINUE
      CALL REORX
      GO TO 1500
C
C     READ DATA CHANNEL HEADER INFORMATION FROM KEYBOARD OPTION
C
 8400 CONTINUE
      CALL SETHD
      GO TO 1500
C
C     PLOT RECORD SECTION OPTION
C
 8600 CONTINUE
      CALL RECSEC
      GO TO 1500
C
 9000 RETURN
      END
```

```
      SUBROUTINE ADAZE(IFL,NCH1,NCH2)
C
C*********************************************************************
C
C     COMPUTE AND LIST DISTANCE AND AZIMUTH FOR DATA CHANNELS
C     (STORE RESULTS IN COMMOM/DTAZ/)
C
C     IFL - LIST DISTANCE-AZIMUTH FLAG - INPUT
C           = 0 NO LIST
C           = 1 LIST
C     NCH1 - FIRST CHANNEL TO COMPUTE DISTANCE-AZIMUTH - INPUT
C     NCH2 - LAST CHANNEL TO COMPUTE DISTANCE-AZIMUTH - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 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/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/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 DATA(1)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      IF(IFL.EQ.0)GO TO 500
C
C     LIST EPICENTER DATA
C
      CALL PREVT
C
C     CONVERT EPICENTER COORDINATES
C
  500 CALL CONVRT(EXLAT,EXLON,ELAT,ELON)
C
C     PROCESS ALL STATION DATA
C
C     IF(NCH1.LE.0)GO TO 7000
      IF(NCH1.GT.NCH2)GO TO 7000
      IF(NCH2.GT.NACHLS)GO TO 7000
      IF(IFL.EQ.0)GO TO 1000
C
```

```
      WRITE(IWR,700)
      WRITE(IWR2,700)
  700 FORMAT(10X,'STATION',8X,'DISTANCE',4X,'DISTANCE',5X,'S-TO-E',
     * 5X,'E-TO-S',/,25X,'DEGREES',4X,'KILOMETERS',3X,'AZIMUTH',4X,
     * 'AZIMUTH',/)
C
 1000 DO 2000 I=NCH1,NCH2
C
C     CONVERT STATION COORDINATES
C
      CALL CONVRT(RLAT(I),RLON(I),SLAT,SLON)
C
C     COMPUTE DISTANCE-AZIMUTH
C
      CALL DIAZ(ELAT,ELON,SLAT,SLON,DELTA(I),DISKM(I),AZSE(I),AZES(I))
C
      IF(IFL.EQ.0)GO TO 2000
C
C     LIST DISTANCE-AZIMUTH DATA
C
      WRITE(IWR,1300)JCODE(I),DELTA(I),DISKM(I),AZSE(I),AZES(I)
      WRITE(IWR2,1300)JCODE(I),DELTA(I),DISKM(I),AZSE(I),AZES(I)
 1300 FORMAT(12X,A4,7X,F9.3,4X,F9.3,4X,F7.3,4X,F7.3)
C
 2000 CONTINUE
C
      IF(IFL.EQ.0)GO TO 9000
C
      WRITE(IWR,1400)
      WRITE(IWR2,1400)
 1400 FORMAT(//)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN CHANNEL NUMBER IN ADAZE *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE ADDTIM(IYR,IMO,IDY,IHR,IMIN,SEC,ASEC,KYR,JULDY,
     * KHR,KMIN,BSEC)
C
C*****************************************************************
C
C     ADD SECONDS AND CONVERT TO JULIAN DAY
C
C     IYR - YEAR - INPUT
C     IMO - MONTH (NUMBER) - INPUT
C     IDY - DAY - INPUT
C     IHR - HOUR - INPUT
C     IMIN - MINUTE - INPUT
C     SEC - SECOND - INPUT
C     ASEC - SECONDS TO ADD TO TIME - INPUT
C     KYR - YEAR - OUTPUT
C     JULDY - JULIAN DAY - OUTPUT
C     KHR - HOUR - OUTPUT
C     KMIN - MINUTE - OUTPUT
C     BSEC - SECOND - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 7, 1993  COLORADO
C
C*****************************************************************
C
C     CONVERT ORIGINAL TIME TO JULIAN DAY FORM
C
      CALL JUL(IYR,IMO,IDY,JULDY,LEAP)
C
C     CONVERT ORIGINAL TIME TO SECONDS
C
      CALL TIMC(ESEC,IHR,IMIN,SEC)
C
C     ADD TIMES IN SECONDS
C
      BSEC=ESEC+ASEC
C
C     CONVERT BACK FROM SECONDS
C
      CALL ITIMC(BSEC,KHR,KMIN,BSEC)
C
      KYR=IYR
  300 IF(KHR.LE.23)GO TO 500
      KHR=KHR-1
      JULDY=JULDY+1
      GO TO 300
C
  500 IF(JULDY.LE.365)GO TO 9000
      IF(JULDY.EQ.366.AND.LEAP.NE.0)GO TO 9000
      KYR=KYR+1
      JULDY=JULDY-1
C
 9000 RETURN
      END
```

```
      SUBROUTINE AMPPH1(Z,C,PH,NH2)
C
C*************************************************************************
C
C     CALCULATE AMPLITUDE AND PHASE
C
C     Z - REAL AND IMAGINARY COMPONENTS - INPUT
C     C - AMPLITUDE - OUTPUT
C     PH - PHASE ANGLE (IN DEGREES) - OUTPUT
C     NH2 - NUMBER OF REAL COMPONENTS = N2LEN/2+1 - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 5, 1993  COLORADO
C
C*************************************************************************
C
      DIMENSION Z(1),C(1),PH(1)
C
      I=1
C
      DO 100 J=2,NH2
      I=I+1
      K=J+NH2
C
C     CALCULATE AMPLITUDE
C
      C(I)=Z(J)*Z(J)+Z(K)*Z(K)
      C(I)=SQRT(C(I))
C
C     CALCULATE PHASE
C
      PH(I)=PHASOR(Z(K),Z(J))
C
  100 CONTINUE
C
C     SET DC VALUE
C
      C(1)=0.0
      PH(1)=0.0
C
      RETURN
      END
```

```
      SUBROUTINE APVEL(NPHAS,VEL,IFL,NCH)
C
C*********************************************************************
C
C     DETERMINE APPARENT VELOCITY ACROSS AN ARRAY FROM CALCULATED
C     TRAVEL TIMES FOR CURRENT EVENT AND CHANNEL DATA
C
C     NPHAS - PHASE NAME - INPUT
C     VEL - APPARENT VELOCITY (IN KILOMETERS/S) - OUTPUT
C           IF VEL = -9.999, THEN ERROR OCCURRED
C     IFL - COMPUTE FLAG - INPUT
C           = 0 COMPUTE TRAVEL TIMES AND DISTANCES
C           = 1 TRAVEL TIMES AND DISTANCES AVAILABLE IN COMMON/DTAZ/
C     NCH - NUMBER OF CHANNELS TO USE (USE FROM 1 TO NCH) - INPUT
C           = 0 SET TO NACHLS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 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/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/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)
      DIMENSION DELTA(100),DISKM(100),AZSE(100),AZES(100),CTT(100),
     * ATT(100),DTT(100)
      DIMENSION EXFLAG(10)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      NPER=0
      IDF=0
      IF(NCH.LE.0)NCH=NACHLS
      IF(NCHLS.GT.1)GO TO 100
   50 VEL=-9.999
      GO TO 9000
C
  100 DO 500 I=1,NCH
      IF(IFL.NE.0)GO TO 150
      IF(IEXYR.LE.0)GO TO 50
C
      CALL TTCOMP(RLAT(I),RLON(I),RELEV(I),EXLAT,EXLON,EXDEPH,NPHAS,
     * DELTA(I),DISKM(I),AZSE(I),AZES(I),CTT(I),UVEL,PERIOD,NPER,IDF)
C
  150 IF(CTT(I).LE.0.0)GO TO 50
      IF(I.NE.1)GO TO 200
      TMAX=CTT(I)
```

```
      TMIN=CTT(I)
      DMAX=DISKM(I)
      DMIN=DISKM(I)
      GO TO 500
C
  200 IF(DISKM(I).LE.DMAX)GO TO 300
      DMAX=DISKM(I)
      TMAX=CTT(I)
      GO TO 500
C
  300 IF(DISKM(I).GE.DMIN)GO TO 500
      DMIN=DISKM(I)
      TMIN=CTT(I)
C
  500 CONTINUE
C
C     COMPUTE APPARENT VELOCITY
C
      IF(TMAX.EQ.TMIN)GO TO 50
      VEL=(DMAX-DMIN)/(TMAX-TMIN)
C
 9000 RETURN
      END
```

```
      SUBROUTINE APVELX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR COMPUTING THE APPARENT VELOCITY OF
C     A PHASE ACROSS THE ARRAY OF CHANNELS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 29, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      ERROR=-9.999
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,200,END=9000)NPHAS
  200 FORMAT(A4)
C
C     NPHAS - PHASE NAME TO COMPUTE APPARENT VELOCITY
C
      WRITE(IWR,300)NPHAS
      WRITE(IWR2,300)NPHAS
  300 FORMAT(/,10X,'COMPUTE APPARENT VELOCITY FOR PHASE ',A4,/)
C
      IFL=0
      NCH=0
C
      CALL APVEL(NPHAS,VEL,IFL,NCH)
C
      IF(VEL.EQ.ERROR)GO TO 7000
C
      WRITE(IWR,500)VEL
      WRITE(IWR2,500)VEL
  500 FORMAT(/,10X,'APPARENT VELOCITY IS ',F10.3,/)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS OR EVENT ',
     * 'INFORMATION *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE BEAM(IDELT,NCH,ICHAN,KSAMP,NBEAM)
C
C*********************************************************************
C
C     FORM BEAM
C
C     IDELT - ARRAY OF TIME DELAYS (IN SAMPLES) - INPUT
C             > 0 ADD CHANNEL TO BX(IDELT+1)
C             < 0 DON'T ADD CHANNEL TO BX
C     NCH - NUMBER OF CHANNELS TO BEAM - INPUT
C           (BEAM FIRST NCH CHANNELS - MAXIMUM NUMBER IS 100)
C     ICHAN - CHANNEL NUMBER TO STORE FINAL BEAM - INPUT
C             = 0 LEAVE IN BX
C     KSAMP - NUMBER OF SAMPLES IN BEAM TO STORE - INPUT
C             = 0 COMPUTE FROM CHANNELS BEAMED
C     NBEAM - NUMBER OF CHANNELS ACTUALLY USED IN FORMING
C             BEAM - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 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/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 IDELT(1),BX(1),DATA(1)
C
      COMPLEX RPOLES,RZEROS
C
      LSAMP=LENG
      NBEAM=0
C
C     ZERO BX ARRAY
C
      DO 100 I=1,IDEM1
      BX(I)=0.0
  100 CONTINUE
C
C     FORM BEAM
C
      DO 2000 I=1,NCH
      IF(IDELT(I).LT.0)GO TO 2000
C
      NBEAM=NBEAM+1
      INDEX=(I-1)*LENG+IDELT(I)
      ISAMP=JSAMP(I)-IDELT(I)
      IF(LSAMP.GT.ISAMP)LSAMP=ISAMP
```

```
C
      DO 1000 J=1,ISAMP
      IN=INDEX+J
      BX(J)=BX(J)+DATA(IN)
 1000 CONTINUE
C
 2000 CONTINUE
C
      IF(NBEAM.EQ.0)GO TO 9000
      IF(KSAMP.LE.0)KSAMP=LSAMP
      IF(KSAMP.GT.LSAMP)KSAMP=LSAMP
C
C     REMOVE MEAN OF BEAM
C
      IDEG=0
C
      CALL METR(BX,KSAMP,IDEG)
C
C     SCALE BEAM
C
      DO 2500 J=1,KSAMP
      BX(J)=BX(J)/NBEAM
 2500 CONTINUE
C
      IF(ICHAN.LE.0)GO TO 9000
      IF(ICHAN.GT.NACHLS)NACHLS=ICHAN
      JSAMP(ICHAN)=KSAMP
      RSAMR(ICHAN)=RSAMR(1)
      INDEX=(ICHAN-1)*LENG
C
C     STORE BEAM IN DATA CHANNEL
C
      DO 3000 J=1,KSAMP
      IN=INDEX+J
      DATA(IN)=BX(J)
 3000 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE BEAMX(IVESP)
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR BEAMFORMING
C
C     IVESP - VESPAGRAM OPTION FLAG - INPUT
C             = 0 NO VESPAGRAM
C             = 1 VESPAGRAM BEAMS FORMED (FIRST PASS)
C             = 2 VESPAGRAM BEAMS FORMED (NOT FIRST PASS)
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 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/DTAZ/DELTA,DISKM,AZSE,AZES,CTT,ATT,DTT
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
      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 DATA(1),BX(1),ANOM(100),IDELT(100)
C
      EQUIVALENCE (BX(59001),ANOM(1)),(ATT(1),IDELT(1))
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      IF(IVESP.GT.1)GO TO 2100
C
C     READ BEAMING INFORMATION
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','NCH',3X,'ICHAN',3X,'KSAMP',/)
      READ(IRE,*,END=9000)NCH,ICHAN,KSAMP
C
C     NCH - NUMBER OF CHANNELS TO INCLUDE IN THE BEAM
C           (BEAM CHANNELS 1 THROUGH NCH)
C           = 0 SET TO NACHLS
C     ICHAN - CHANNEL NUMBER TO STORE FINAL BEAM
C             (MAY BE LARGER THAN NACHLS)
C             = 0 LEAVE IN BX
```

```
C      KSAMP - NUMBER OF SAMPLES TO INCLUDE IN BEAM
C              = 0 USE SMALLEST NUMBER OF SAMPLES OF ALL CHANNELS
C                  IN BEAM
C
       WRITE(IWR,200)NCH,ICHAN,KSAMP
       WRITE(IWR2,200)NCH,ICHAN,KSAMP
  200  FORMAT(/,10X,'NCH = ',I4,5X,'ICHAN = ',I4,5X,'KSAMP = ',I6,/)
C
       IF(NCH.LE.0)NCH=NACHLS
       IF(NCH.GT.NACHLS)GO TO 7000
       IF(ICHAN.GT.NCHLS)GO TO 7000
       IF(KSAMP.GT.LENG)KSAMP=LENG
       SAMR=RSAMR(1)
C
       DO 300 I=2,NCH
       IF(RSAMR(I).NE.SAMR)GO TO 7300
  300  CONTINUE
C
       WRITE(IWR2,400)
  400  FORMAT(/,'++ INPUT: ','IDL',3X,'IDIR',3X,'FNOISE',/)
       READ(IRE,*,END=9000)IDL,IDIR,FNOISE
C
C      IDL - TIME DELAY FLAG
C            = 0 NO TIME ANOMALIES
C            = 1 READ IN ADDITIONAL TIME ANOMALIES
C      IDIR - DIRECTION TO PHASE ARRAY
C             = 0 TOWARDS SHORTEST DISTANCE TO EVENT
C             = 1 TOWARDS LONGEST DISTANCE TO EVENT
C      FNOISE - NOISE TO INCLUDE BEFORE SIGNAL IN BEAM (IN SECONDS)
C               = 0 SET TO 10.0
C
       IF(FNOISE.LE.0.0)FNOISE=10.0
       WRITE(IWR,450)IDL,IDIR,FNOISE
       WRITE(IWR2,450)IDL,IDIR,FNOISE
  450  FORMAT(/,10X,'IDL = ',I2,5X,'IDIR = ',I2,5X,'FNOISE = ',F8.2,/)
C
       IF(IDL.LE.0)GO TO 1000
C
       WRITE(IWR2,470)
  470  FORMAT(/,'++ INPUT: ','ANOM',/)
       READ(IRE,*,END=9000)(ANOM(I),I=1,NCH)
C
C      ANOM - TIME ANOMALIES FOR EACH CHANNEL (IN SECONDS)
C
C      CALCULATE DISTANCE-AZIMUTH FOR ALL CHANNELS
C
 1000  IFL=1
       NCH1=1
C
       CALL ADAZE(IFL,NCH1,NCH)
C
C      CHECK DIRECTION OF APPROACH OF SIGNAL
C
       IF(IDIR.LE.0)GO TO 1400
C
       DO 1300 I=1,NCH
       DELTA(I)=360.0-DELTA(I)
       DISKM(I)=DELTA(I)*DG2KM
```

```
      AZSE(I)=AZSE(I)+180.0
      AZES(I)=AZES(I)+180.0
 1300 CONTINUE
C
 1400 IF(IVESP.GT.0)GO TO 1700
C
      WRITE(IWR2,1500)
 1500 FORMAT(/,'++ INPUT: ','VEL',3X,'IVP',/)
      READ(IRE,*,END=9000)VEL,IVP
C
C     VEL - BEAMING PHASE VELOCITY (IN KILOMETERS/S)
C         - BEAMING RAY PARAMETER (IN S/DEGREE)
C           = 0 USE APPARENT VELOCITY BASED UPON PREDICTED ARRIVAL
C               TIMES FOR SPECIFIED PHASE
C     IVP - VELOCITY FLAG
C           = 0 VEL IS PHASE VELOCITY
C           = 1 VEL IS RAY PARAMETER
C
      WRITE(IWR,1600)VEL,IVP
      WRITE(IWR2,1600)VEL,IVP
 1600 FORMAT(/,10X,'VEL = ',,F9.3,5X,'IVP = ',I2,/)
C
      GO TO 1900
C
 1700 WRITE(IWR,1750)
 1750 FORMAT(/,'++ INPUT: ','VEL',3X,'DV',3X,'IVP',/)
      READ(IRE,*,END=9000)VEL,DV,IVP
C
C     VEL - BEAMING PHASE VELOCITY (IN KILOMETERS/S)
C         - BEAMING RAY PARAMETER (IN S/DEGREE)
C     DV - INCREMENT IN PHASE VELOCITY OR RAY PARAMETER
C          = 0 SET TO 0.02
C     IVP - VELOCITY FLAG
C           = 0 VEL IS PHASE VELOCITY
C           = 1 VEL IS RAY PARAMETER
C
      IF(DV.LE.0.0)DV=0.02
      IF(VEL.LE.0.0)GO TO 7000
      WRITE(IWR,1800)VEL,DV,IVP
      WRITE(IWR2,1800)VEL,DV,IVP
 1800 FORMAT(/,10X,'VEL = ',F10.3,5X,'DV = ',F9.4,5X,'IVP = ',I2,/)
C
      VELX=VEL
      GO TO 2200
C
 1900 IF(VEL.NE.0.0)GO TO 2200
C
      WRITE(IWR2,1950)
 1950 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,1960,END=9000)NPHAS
 1960 FORMAT(A4)
C
C     NPHAS - PHASE NAME
C
      WRITE(IWR,1970)NPHAS
      WRITE(IWR2,1970)NPHAS
 1970 FORMAT(/,10X,'PHASE = ',A4,/)
C
```

```
C     CALCULATE TRAVEL TIME FOR EACH CHANNEL FOR PHASE NPHASE
C
      UVEL=0.0
      PERIOD=0.0
      NPER=0
      IDF=1
C
      DO 2000 I=1,NCH
C
      CALL TTCOMP(RLAT(I),RLON(I),RELEV(I),EXLAT,EXLON,EXDEPH,
     * NPHAS,DELTA(I),DISKM(I),AZSE(I),AZES(I),CTT(I),UVEL,PERIOD,
     * NPER,IDF)
C
      IF(CTT(I).LE.0.0)GO TO 7500
 2000 CONTINUE
C
      GO TO 3000
C
 2100 VELX=VELX+DV
      VEL=VELX
C
C     CALCULATE TRAVEL TIME FOR EACH CHANNEL FROM THE VELOCITY VEL
C
 2200 IF(IVP.EQ.1.AND.VEL.NE.0.0)VEL=DG2KM/VEL
C
      DO 2500 I=1,NCH
      CTT(I)=DISKM(I)/VEL
 2500 CONTINUE
C
      IF(IVESP.GT.1)GO TO 3400
C
C     LIST DISTANCES AND TRAVEL TIMES
C
 3000 WRITE(IWR,3100)
      WRITE(IWR2,3100)
 3100 FORMAT(/,10X,'STATION',5X,'DISTANCE',4X,'DISTANCE',4X,
     * 'AZIMUTH',4X,'TRAVEL TIME')
      WRITE(IWR,3200)
      WRITE(IWR2,3200)
 3200 FORMAT(25X,'DEG',9X,'KM',7X,'S TO E',9X,'SEC',/)
      WRITE(IWR,3300)(JCODE(I),DELTA(I),DISKM(I),AZSE(I),CTT(I),
     * I=1,NCH)
      WRITE(IWR2,3300)(JCODE(I),DELTA(I),DISKM(I),AZSE(I),CTT(I),
     * I=1,NCH)
 3300 FORMAT(12X,A4,5X,F8.3,5X,F8.3,4X,F7.3,4X,F9.3)
C
C     CALCULATE ARRIVAL TIMES FOR EACH CHANNEL
C
 3400 CALL TIMC(ET,IEXHR,IEXMN,EXSEC)
C
      DO 3500 I=1,NCH
      CTT(I)=CTT(I)+ET
 3500 CONTINUE
C
C     CALCULATE TIME DELAYS
C
      CALL DELTT(CTT,DTT,IDELT,NCH,FNOISE)
C
```

```
      IF(IVESP.GT.1)GO TO 3750
C
C     LIST TIME DELAYS
C
      WRITE(IWR,3600)
      WRITE(IWR2,3600)
 3600 FORMAT(///,10X,'STATION',6X,'TIME DELAY',5X,'TIME DELAY',5X,
     * 'ANOMALY',/,25X,'SECONDS',7X,'SAMPLES',/)
      WRITE(IWR,3700)(JCODE(I),DTT(I),IDELT(I),ANOM(I),I=1,NCH)
      WRITE(IWR2,3700)(JCODE(I),DTT(I),IDELT(I),ANOM(I),I=1,NCH)
 3700 FORMAT(12X,A4,5X,F10.3,6X,I8,7X,F7.3)
C
 3750 IF(IDL.EQ.0)GO TO 4000
C
C     APPLY TIME ANOMALIES
C
      DO 3800 I=1,NCH
      IDELT(I)=IDELT(I)+ANOM(I)*SAMR
 3800 CONTINUE
C
C     FORM BEAM
C
 4000 CALL BEAM(IDELT,NCH,ICHAN,KSAMP,NBEAM)
C
      IF(IDIR.LE.0)GO TO 5000
C
      DO 4500 I=1,NCH
      DELTA(I)=360.0-DELTA(I)
      DISKM(I)=DELTA(I)*DG2KM
      AZSE(I)=AZSE(I)+180.0
      AZES(I)=AZES(I)+180.0
 4500 CONTINUE
C
 5000 IF(NBEAM.NE.0)GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR - NO CHANNELS USED IN BEAM *****',/)
 7150 WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'NO BEAMS FORMED',/)
      GO TO 9000
C
 7300 WRITE(IWR2,7400)
 7400 FORMAT(/,10X,'***** SAMPLING RATE NOT SAME FOR ALL CHANNELS',/,
     * 10X,'CANNOT BEAM CHANNELS *****',/)
      GO TO 7150
C
 7500 WRITE(IWR2,7600)
 7600 FORMAT(/,10X,'***** CALCULATED TRAVEL TIME IS ZERO *****',/)
      GO TO 7150
C
 8000 IF(IVESP.GT.1)GO TO 9000
      WRITE(IWR,8100)NBEAM
      WRITE(IWR2,8100)NBEAM
 8100 FORMAT(/,10X,'BEAM SUCCESSFULLY FORMED WITH ',I4,' CHANNELS',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE BNDPAS(FLOW,FHIGH,SAMR,D,G)
C
C*********************************************************************
C
C     DESIGN A RECURSIVE BUTTERWORTH BANDPASS FILTER
C
C     FILTER WILL HAVE 8 POLES IN THE S PLANE AND IS APPLIED IN THE
C     FORWARD AND REVERSE DIRECTIONS SO AS TO HAVE ZERO PHASE DELAY
C
C     FREQUENCY ROLLOFF IS 96 DB PER OCTAVE; A BILINEAR Z TRANSFORM
C     IS USED IN DESIGNING THE FILTER TO PREVENT ALIASING PROBLEMS
C
C     FLOW - LOW FREQUENCY CUTOFF (6 DB DOWN) - INPUT
C     FHIGH - HIGH FREQUENCY CUTOFF (6 DB DOWN) - INPUT
C     SAMR - SAMPLING RATE (SAMPLES/S) - INPUT
C     D - EIGHT Z DOMAIN COEFFICIENTS OF RECURSIVE FILTER - OUTPUT
C     G - GAIN OF FILTER - OUTPUT
C
C     NOVEMBER 16, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION D(1),P(4),S(8)
C
      COMPLEX P,S,Z1,Z2
C
      DT=1.0/SAMR
      TDT=2.0/DT
      FDT=4.0/DT
C
      P(1)=CMPLX(-0.3826834,0.9238795)
      P(2)=CMPLX(-0.3826834,-0.9238795)
      P(3)=CMPLX(-0.9238795,0.3826834)
      P(4)=CMPLX(-0.9238795,-0.3826834)
      W1=TWOPI*FLOW
      W2=TWOPI*FHIGH
      W1=TDT*TAN(W1/TDT)
      W2=TDT*TAN(W2/TDT)
      HWID=(W2-W1)/2.0
      WW=W1*W2
C
      DO 100 I=1,4
      Z1=P(I)*HWID
      Z2=Z1*Z1-WW
      Z2=CSQRT(Z2)
      S(I)=Z1+Z2
      S(I+4)=Z1-Z2
  100 CONTINUE
C
      G=0.5/HWID
      G=G*G*G*G
C
      DO 200 I=1,7,2
      B=-2.0*REAL(S(I))
      Z1=S(I)*S(I+1)
```

```
      C=REAL(Z1)
      A=TDT+B+C/TDT
      G=G*A
      D(I)=(C*DT-FDT)/A
      D(I+1)=(A-2.0*B)/A
  200 CONTINUE
C
      G=G*G
C
      RETURN
      END
```

```
      SUBROUTINE CALB(DATA,ISAMP,XXCAL)
C
C*************************************************************************
C
C     CALIBRATE DATA CHANNEL
C
C     DATA - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF POINTS IN TIME SERIES - INPUT
C     XXCAL - SCALING FACTOR - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 19, 1993  COLORADO
C
C*************************************************************************
C
      DIMENSION DATA(1)
C
      DO 100 J=1,ISAMP
      DATA(J)=DATA(J)*XXCAL
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CALEX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR DATA SCALING
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 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/CAL/XXCAL
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/OPTION/IOPS(1)
      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 SCALING PARAMETERS
C
      WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','ICHAN',3X,'XXCAL',/)
      READ(IRE,*,END=9000)ICHAN,XXCAL
C
C     ICHAN - DESIRED CHANNEL
C             = 0 SCALE ALL DATA CHANNELS
C     XXCAL - SCALING FACTOR
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'SCALING PARAMETERS',/)
      WRITE(IWR,300)ICHAN,XXCAL
      WRITE(IWR2,300)ICHAN,XXCAL
  300 FORMAT(/,10X,'ICHAN = ',I4,5X,'XXCAL =',E12.3,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
      IF(XXCAL.EQ.0.0)GO TO 7000
      IOPS(23)=1
C
      IF(ICHAN.LE.0)GO TO 500
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 2000
C
  500 NCH1=1
      NCH2=NACHLS
C
C     APPLY SCALE FACTOR
```

```
C
 2000 DO 3000 I=NCH1,NCH2
      ISAMP=JSAMP(I)
      INDEX=(I-1)*LENG+1
C
      CALL CALB(DATA(INDEX),ISAMP,XXCAL)
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 SCALED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY SCALED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE CALROT
C
C*********************************************************************
C
C     ROTATE HORIZONTAL COMPONENTS OF MOTION
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 14, 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/DTAZ/DELTA,DISKM,AZSE,AZES,CTT,ATT,DTT
      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 DATA(1)
C
      COMPLEX RPOLES,RZEROS
C
C     READ CHANNEL NUMBERS TO BE ROTATED
C
  100 WRITE(IWR2,150)
  150 FORMAT(/,'++ INPUT: ','IR',3X,'IT',/)
      READ(IRE,*,END=9000)IR,IT
C
C     IR - CHANNEL NUMBER OF RADIAL (N-S) COMPONENT
C     IT - CHANNEL NUMBER OF TRANSVERSE (E-W) COMPONENT
C
      WRITE(IWR,200)IR,IT
      WRITE(IWR2,200)IR,IT
  200 FORMAT(/,10X,'IR = ',I4,5X,'IT = ',I4,/)
C
      IF(IR.LE.0)GO TO 7000
      IF(IT.LE.0)GO TO 7000
      IF(IR.GT.NACHLS)GO TO 7000
      IF(IT.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','GAM',3X,'BETA',/)
      READ(IRE,*,END=9000)GAM,BETA
C
C     GAM - STATION-TO-EPICENTER AZIMUTH (IN DEGREES)
C           < 0 USE COMPUTED BACK AZIMUTH IN COMMON/DTAZ/
C     BETA - ANGLE OF INSTALLED RADIAL (NORTH) COMPONENT WITH
C            RESPECT TO TRUE NORTH (IN DEGREES)
C            = 0 FOR INSTALLED N-S AND E-W COMPONENTS
C
      IF(GAM.LT.0.0)GAM=AZSE(IR)
```

```
C
      WRITE(IWR,400)
      WRITE(IWR2,400)
  400 FORMAT(/,10X,'ROTATION PARAMETERS',/)
      WRITE(IWR,500)GAM,BETA
      WRITE(IWR2,500)GAM,BETA
  500 FORMAT(/,10X,'GAM = ',F7.3,5X,'BETA = ',F7.3,/)
C
      INDEXR=(IR-1)*LENG+1
      INDEXT=(IT-1)*LENG+1
C
      IF(JSAMP(IR).NE.JSAMP(IT))GO TO 7000
      IF(RSAMR(IR).NE.RSAMR(IT))GO TO 7000
      ISAMP=JSAMP(IR)
C
C     ROTATE TWO CHANNELS
C
      CALL REVOLV(DATA(INDEXR),DATA(INDEXT),ISAMP,GAM,BETA)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,'DATA NOT ROTATED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY ROTATED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE CLFORM(A,G,M,ISIGN)
C
C*********************************************************************
C
C     COMPUTE FOURIER TRANSFORM
C
C     A - COSINE TRANSFORMS - INPUT AND OUTPUT
C     G - SINE TRANSFORMS - INPUT AND OUTPUT
C     M - POWER OF 2 OF TIME SERIES - INPUT
C     ISIGN - TRANSFORM FLAG - INPUT
C             =-1 FOR DIRECT TRANSFORM
C             =+1 FOR INVERSE TRANSFORM
C
C     JULY 28, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION INT(16),H(2),A(1),G(1)
C
C     INITIALIZE ALL CONSTANTS
C
      N=2**M
      INT(M)=1
C
      DO 10 I=2,M
      L=M-I+1
      INT(L)=INT(L+1)*2
   10 CONTINUE
C
      ANG=(3.14159265*ISIGN)/INT(1)
C
C     COMPUTE FOR M LAYERS
C
      DO 50 L=1,M
      K=M-L+1
C
C     SET NB, LB, LH, AND NW FOR THE NEW LAYER
C
      NB=INT(K)
      LB=INT(L)*2
      LH=INT(L)
      NW=0
C
C     LOOP OVER PAIRS OF COMPLEX NUMBERS IN EACH LAYER
C
      DO 45 IB=1,NB
      LS=LB*(IB-1)
      ARG=NW*ANG
      H(1)=COS(ARG)
      H(2)=-SIN(ARG)
C
C     COMPUTE ELEMENTS FOR BOTH HALVES OF EACH PAIR
C
      DO 20 I=1,LH
      J=I+LS
      K=J+LH
      Q=A(K)*H(1)-G(K)*H(2)
      R=A(K)*H(2)+G(K)*H(1)
```

```
      A(K)=A(J)-Q
      G(K)=G(J)-R
      A(J)=A(J)+Q
      G(J)=G(J)+R
   20 CONTINUE
C
C     COMPUTE THE NEXT INDEX NUMBER - NW
C
      DO 30 I=2,M
      II=I
      IF(INT(I).GT.NW)GO TO 40
      NW=NW-INT(I)
   30 CONTINUE
C
   40 NW=NW+INT(II)
   45 CONTINUE
C
   50 CONTINUE
C
C     REARRANGE THE RESULTS
C
      NW=0
C
      DO 100 K=1,N
      NW1=NW+1
      IF(NW1-K)70,70,60
   60 HOLD1=A(NW1)
      HOLD2=G(NW1)
      A(NW1)=A(K)
      G(NW1)=G(K)
      A(K)=HOLD1
      G(K)=HOLD2
   70 CONTINUE
C
C     COMPUTE THE NEXT INDEX NUMBER - NW
C
      DO 80 I=1,M
      II=I
      IF(INT(I).GT.NW)GO TO 90
      NW=NW-INT(I)
   80 CONTINUE
C
   90 NW=NW+INT(II)
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CLOSEX
C
C*********************************************************************
C
C     CLOSE LOGICAL UNIT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 26, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/OPTION/IOPS
C
      DIMENSION IOPS(1)
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','LUNX',/)
      READ(IRE,*,END=9000)LUNX
C
C     LUNX - LOGICAL UNIT NUMBER TO CLOSE
C
      WRITE(IWR,200)LUNX
      WRITE(IWR2,200)LUNX
  200 FORMAT(/,10X,'CLOSE LOGICAL UNIT ',I3,/)
C
      IF(LUNX.LE.0)GO TO 7000
C
      CLOSE(LUNX)
C
      IF(LUNX.EQ.LUN)IOPS(50)=0
      IF(LUNX.EQ.LUN2)IOPS(51)=0
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'FILE NOT CLOSED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'FILE SUCCESSFULLY CLOSED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE CLTWO(X,Y,M,ISIGN)
C
C*********************************************************************
C
C     TRANSFORMS TWO ARRAYS AT THE SAME TIME USING CLFORM
C
C     TIME TO FREQUENCY
C
C     X - TIME SERIES - INPUT
C       - FOURIER TRANSFORM - OUTPUT
C     Y - TIME SERIES - INPUT
C       - FOURIER TRANSFORM - OUTPUT
C     M - POWER OF 2 OF NUMBER OF SAMPLES IN TIME DOMAIN - INPUT
C     ISIGN - TRANSFORM FLAG - INPUT
C             =-1 FOR DIRECT (TIME TO FREQUENCY)
C             =+1 FOR INDIRECT (FREQUENCY TO TIME)
C
C     X - FIRST 2**(M-1) HAS X COSINE TRANSFORMS
C         NEXT 2**(M-1) HAS Y COSINE TRANSFORMS
C     Y - FIRST 2**(M-1) HAS X SINE TRANSFORMS
C         NEXT 2**(M-1) HAS Y SINE TRANSFORMS
C
C     FREQUENCY TO TIME, REVERSE INPUT AND OUTPUT ABOVE
C
C     X AND Y MUST BE DIMENSIONED AT LEAST 2**M+2
C     TRANSFORMS ARE COMPUTED ONLY TO THE FOLDING FREQUENCY
C
C     AUGUST 19, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1),Y(1)
C
      KT=2**M
      KT2=KT+2
      IK=KT2
      IP22=IK/2
      NN=(IK-IP22)/2+IP22
      N=IP22+1
      IF(ISIGN.GE.0)GO TO 30
C
      CALL CLFORM(X,Y,M,ISIGN)
C
      X(KT2)=Y(1)
      Y(1)=0.0
      Y(KT2)=0.0
C
      DO 20 I=2,IP22
      KT2=KT2-1
      X(KT2)=0.5*(Y(I)+Y(KT))
      Y(KT2)=-0.5*(X(I)-X(KT))
      X(I)=0.5*(X(I)+X(KT))
      Y(I)=0.5*(Y(I)-Y(KT))
      KT=KT-1
   20 CONTINUE
C
   30 DO 40 I=N,NN
      HOLD=X(I)
```

```
      X(I)=X(IK)
      X(IK)=HOLD
      HOLD=Y(I)
      Y(I)=Y(IK)
      Y(IK)=HOLD
      IK=IK-1
   40 CONTINUE
C
      IF(ISIGN.LT.0)GO TO 9000
C
      Y(I)=X(KT2)
      Y(IP22)=X(N)
C
      DO 60 I=N,KT
      X(I)=X(I+1)
      Y(I)=Y(I+1)
   60 CONTINUE
C
      JP22=IP22-1
C
      DO 70 I=2,JP22
      HOLD=X(KT)
      X(KT)=X(I)+Y(KT)
      X(I)=X(I)-Y(KT)
      Y(KT)=-Y(I)+HOLD
      Y(I)=Y(I)+HOLD
      KT=KT-1
   70 CONTINUE
C
      CALL CLFORM(X,Y,M,ISIGN)
C
 9000 RETURN
      END
```

```
      SUBROUTINE COMPAS(SLA,XNS,SLO,XEW)
C
C*****************************************************************************
C
C     SET N OR S AND E OR W DIRECTIONS (MUST CALL DZCON FIRST)
C
C     SLA - GEOGRAPHIC LATITUDE (IN DEGREES) - INPUT AND OUTPUT
C           - = S FOR INPUT
C           + = N AND S FOR OUTPUT
C     XNS - N OR S - OUTPUT
C           S = -SLA
C     SLO - GEOGRAPHIC LONGITUDE (IN DEGREES) - INPUT AND OUTPUT
C           - = W FOR INPUT
C           + = E AND W FOR OUTPUT
C     XEW - E OR W - OUTPUT
C           W = -SLO
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 6, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      INTEGER*4 XNS,XEW
C
      IF(SLA.LT.0.0)THEN
         SLA=-SLA
         XNS=KSS
      ELSE
         XNS=KNN
      ENDIF
C
      IF(SLO.LT.0.0)THEN
         SLO=-SLO
         XEW=KWW
      ELSE
         XEW=KEE
      ENDIF
C
      RETURN
      END
```

```
      SUBROUTINE CONVRT(XLAT,XLON,ZLAT,ZLON)
C
C*********************************************************************
C
C     CONVERT GEOGRAPHIC LATITUDE AND LONGITUDE TO GEOCENTRIC
C     CO-LATITUDE AND EAST LONGITUDE
C
C     XLAT - GEOGRAPHIC LATITUDE (IN DEGREES) - INPUT
C            - = S
C     XLON - GEOGRAPHIC LONGITUDE (IN DEGREES) - INPUT
C            - = W
C     ZLAT - GEOCENTRIC CO-LATITUDE (IN DEGREES) - OUTPUT
C     ZLON - EAST LONGITUDE (IN DEGREES) - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 29, 1993  COLORADO
C
C*********************************************************************
C
      ZLAT=XLAT
      ZLON=XLON
C
C     CONVERT TO COMPASS DIRECTIONS
C
      CALL COMPAS(ZLAT,INS,ZLON,IEW)
C
C     CONVERT TO GEOCENTRIC CO-LATITUDE AND EAST LONGITUDE
C
      CALL COORD1(ZLAT,INS,ZLON,IEW)
C
      RETURN
      END
```

```
      SUBROUTINE COORD1(XLAT,KNS,XLONG,KEW)
C
C*********************************************************************
C
C     CONVERTS GEOGRAPHIC LATITUDE AND LONGITUDE INTO GEOCENTRIC
C     LATITUDE AND EAST LONGITUDE
C
C     XLAT - GEOGRAPHIC LATITUDE (IN DEGREES) - INPUT
C          - GEOCENTRIC CO-LATITUDE (IN RADIANS) - OUTPUT
C     KNS - N OR S FOR NORTH OR SOUTH  - INPUT
C           BLANK = N
C     XLONG - GEOGRAPHIC LONGITUDE (IN DEGREES) - INPUT
C           - EAST LONGITUDE (IN RADIANS) - OUTPUT
C     KEW - E OR W FOR EAST OR WEST - INPUT
C           BLANK = E
C
C     JULY 29, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      XLAT=XLAT+90.0
C
      IF(KNS.NE.KSS)XLAT=180.0-XLAT
      IF(KEW.EQ.KWW)XLONG=360.0-XLONG
C
      XLONG=XLONG*DG2RD
      ARG=2.0*XLAT*DG2RD
      X=GEOCO1*SIN(ARG)
      X=GEOCO2*SIN(ARG+ARG)+X
      XLAT=(XLAT+X)*DG2RD
C
      RETURN
      END
```

```
      SUBROUTINE COPYF
C
C*****************************************************************************
C
C     MAKE MULTIPLE COPIES OF A DATA FILE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 27, 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
C     READ COPY PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,10X,'++ INPUT: ','ICHAN1',3X,'ICHAN2',3X,'ICHAN3',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2,ICHAN3
C
C     ICHAN1 - CHANNEL TO MAKE COPIES OF
C              = 0 SET TO 1
C     ICHAN2 - FIRST CHANNEL TO STORE COPY OF ICHAN1
C              = 0 SET TO NACHLS+1
C     ICHAN3 - LAST CHANNEL TO STORE COPY OF ICHAN1
C              (IF ICHAN2 = 0 SET ICHAN3 = NACHLS+ICHAN3)
C
      IF(ICHAN1.LE.0)ICHAN1=1
      IF(ICHAN2.LE.0)ICHAN3=NACHLS+ICHAN3
      IF(ICHAN2.LE.0)ICHAN2=NACHLS+1
      WRITE(IWR,500)ICHAN1,ICHAN2,ICHAN3
  500 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,5X,'ICHAN3 = ',
     * I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN3.LT.ICHAN2)GO TO 7000
      IF(ICHAN2.GT.NCHLS)GO TO 7000
      IF(ICHAN3.GT.NCHLS)GO TO 7000
C
      IF(ICHAN3.GT.NACHLS)NACHLS=ICHAN3
C
      IFIR1=1
      ILAS1=JSAMP(ICHAN1)
      IFIR2=1
      IFL=0
      IHF=0
C
C     COPY CHANNEL
```

```
C
      DO 1000 I=ICHAN2,ICHAN3
C
      CALL MOVE(IFL,IFIR1,ILAS1,ICHAN1,IFIR2,I,IHF)
C
 1000 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 COPIED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY COPIED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE CROSSX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR CROSSCORRELATION
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 18, 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
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'ICHAN2',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2
C
C     ICHAN1 - CHANNEL NUMBER FOR FIRST TIME SERIES
C              = 0 SET TO 1
C     ICHAN2 - CHANNEL NUMBER FOR SECOND TIME SERIES
C              (CORRELOGRAM PLACED IN THIS CHANNEL AFTER
C               CROSSCORRELATION)
C
      IF(ICHAN1.LE.0)ICHAN1=1
      WRITE(IWR,550)
      WRITE(IWR2,550)
  550 FORMAT(/,10X,'CROSSCORRELATION PARAMETERS',/)
      WRITE(IWR,600)ICHAN1,ICHAN2
      WRITE(IWR2,600)ICHAN1,ICHAN2
  600 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NACHLS)GO TO 7000
C
      ISAMP=JSAMP(ICHAN1)
      LF=JSAMP(ICHAN2)
      SAMR1=RSAMR(ICHAN1)
      SAMR2=RSAMR(ICHAN2)
C
      WRITE(IWR,800)ISAMP,LF
      WRITE(IWR2,800)ISAMP,LF
  800 FORMAT(/,10X,'SAMPLES IN FIRST TIME SERIES = ',1X,I6,/,10X,
     * 'SAMPLES IN SECOND TIME SERIES = ',I6,/)
      WRITE(IWR,900)SAMR1,SAMR2
      WRITE(IWR2,900)SAMR1,SAMR2
```

```
  900 FORMAT(/,10X,'SAMPLING RATE FOR FIRST TIME SERIES = ',F8.3,/,
     * 10X,'SAMPLING RATE FOR SECOND TIME SERIES = ',F7.3,/)
C
      IF(SAMR1.NE.SAMR2)GO TO 7000
C
C     DETERMINE POWER OF 2 PARAMETERS
C
      CALL POWER2(ISAMP,LENG,N2PWR,N2LEN,NH2,N2P1,NTOT,NZ)
C
      INDEX1=(ICHAN1-1)*LENG+1
      INDEX2=(ICHAN2-1)*LENG+1
C
C     CROSSCORRELATE DATA
C
      CALL MATCH(DATA(INDEX1),ISAMP,DATA(INDEX2),LF,N2PWR,N2LEN,NH2)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT CROSSCORRELATED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY CROSSCORRELATED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE CYCLES(RF,RMAX,NCY,ZFT,ZFL,ZMAXT,ZMAXU,ZFCYT)
C
C*********************************************************************
C
C     SET AXES PARAMETERS FOR LOG PLOT
C
C     RF - AXIS ORIGIN PLOT VALUE - INPUT
C     RMAX - AXIS MAXIMUM PLOT VALUE - INPUT
C     NCY - NUMBER OF CYCLES FROM RMAX DESIRED TO PLOT - INPUT
C     ZFT - RF ROUNDED TO NEAREST LOWER POWER OF TEN - OUTPUT
C     ZFL - RF ROUNDED TO NEAREST LOWER TICK MARK - OUTPUT
C     ZMAXT - RMAX ROUNDED TO NEAREST HIGHER POWER OF TEN - OUTPUT
C     ZMAXU - RMAX ROUNDED TO NEAREST HIGHER TICK MARK - OUTPUT
C     ZFCYT - NCY CYCLES LESS THAN ZMAXT - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 9, 1993  COLORADO
C
C*********************************************************************
C
C     DETERMINE ZFT
C
      ZFT=ALOG10(RF)
      ICYC=ZFT
      RCYC=ICYC
      IF((ZFT.LT.0.0).AND.(RCYC.NE.ZFT))ICYC=ICYC-1
      ZFT=10.0**ICYC
C
C     DETERMINE ZFL
C
      IFAC=RF/ZFT
      ZFL=ZFT*IFAC
C
C     DETERMINE ZMAXT
C
      ZMAXT=ALOG10(RMAX)
      ICYC=ZMAXT
      RCYC=ICYC
      IF((ZMAXT.LT.0.0).AND.(RCYC.NE.ZMAXT))ICYC=ICYC-1
      ZMAXT=10.0**ICYC
C
C     DETERMINE ZMAXU
C
      IFAC=RMAX/ZMAXT
      ZMAXU=ZMAXT*(IFAC+1)
C
C     DETERMINE ZFCYT
C
      ICYC=ICYC-NCY+1
      ZFCYT=10.0**ICYC
C
      RETURN
      END
```

```
      SUBROUTINE DAAS(IFL)
C
C*************************************************************************
C
C     ADD OR SUBTRACT DATA CHANNELS
C
C     IFL - ADD/SUBTRACT FLAG - INPUT
C           = 0 ADD
C           = 1 SUBTRACT
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 PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'ICHAN2',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2
C
C     ICHAN1 - DATA CHANNEL TO BE MODIFIED
C     ICHAN2 - DATA CHANNEL TO BE ADDED TO OR SUBTRACTED FROM ICHAN1
C
      WRITE(IWR,200)ICHAN1,ICHAN2
      WRITE(IWR2,200)ICHAN1,ICHAN2
  200 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN1.LE.0)GO TO 7000
      IF(ICHAN2.LE.0)GO TO 7000
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NACHLS)GO TO 7000
C
      ISAMP=JSAMP(ICHAN2)
      INDEX1=(ICHAN1-1)*LENG
      INDEX2=(ICHAN2-1)*LENG
      FX=1.0
      IF(IFL.EQ.1)FX=-FX
C
      DO 1000 J=1,ISAMP
      IND1=INDEX1+J
      IND2=INDEX2+J
      DATA(IND1)=DATA(IND1)+FX*DATA(IND2)
```

```
 1000 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 MODIFIED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY MODIFIED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE DATAL
C
C*****************************************************************
C
C     LIST 200 DATA POINTS
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/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: ','ICHAN',3X,'IPT',/)
      READ(IRE,*,END=9000)ICHAN,IPT
C
C     ICHAN - DESIRED CHANNEL NUMBER TO LIST DATA
C             = 0 LIST BX
C     IPT - FIRST POINT TO LIST
C           = 0 SET TO 1
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'LIST DATA POINTS PARAMETERS',/)
      IF(IPT.LE.0)IPT=1
      WRITE(IWR,300)ICHAN,IPT
      WRITE(IWR2,300)ICHAN,IPT
  300 FORMAT(/,10X,'ICHAN = ',I4,5X,'IPT = ',I6,//)
C
      IF(ICHAN.LE.0)GO TO 500
      ISAMP=JSAMP(ICHAN)
      IF(ICHAN.GT.NACHLS)GO TO 7000
      IF(IPT.GT.ISAMP)GO TO 7000
C
C     LIST DATA
C
  500 IPTL=IPT+199
      IF(ICHAN.LE.0)GO TO 2000
      IF(IPTL.GT.LENG)IPTL=LENG
C
      INDEX=(ICHAN-1)*LENG
      WRITE(IWR2,1100)(DATA(INDEX+J),J=IPT,IPTL)
 1100 FORMAT(10X,5F12.3)
```

```
      GO TO 9000
C
 2000 WRITE(IWR2,1100)(BX(J),J=IPT,IPTL)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT LISTED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE DAZEX
C
C*****************************************************************
C
C     COMPUTE DISTANCE AND AZIMUTH FOR INPUT POINTS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 2, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION SLAT(100),SLON(100),SNS(100),SEW(100),ELAT(100),
     * ELON(100),ENS(100),EEW(100),ISTA(100),IEPI(100),BX(1)
C
      EQUIVALENCE (BX(1),SLAT(1)),(BX(101),SLON(1)),(BX(201),ELAT(1)),
     * (BX(301),ELON(1)),(BX(401),SNS(1)),(BX(501),SEW(1)),
     * (BX(601),ENS(1)),(BX(701),EEW(1)),(BX(801),ISTA(1)),
     * (BX(901),IEPI(1))
C
      INTEGER*4 SNS,SEW,ENS,EEW
C
      DATA LAST/'last'/
C
      WRITE(IWR2,20)
   20 FORMAT(/,'++ INPUT: ','IFME',3X,'IFMS',3X,'IES',/)
      READ(IRE,*,END=9000)IFME,IFMS,IES
C
C     IFME - FORMAT FLAG FOR EPICENTER INPUT
C            = 0 COORDINATES IN DECIMAL
C            = 1 COORDINATES IN DEG, MIN, SEC
C     IFMS - FORMAT FLAG FOR STATION INPUT
C            = 0 COORDINATES IN DECIMAL
C            = 1 COORDINATES IN DEG, MIN, SEC
C     IES - DATA INPUT SEQUENCE FLAG
C           = 0 ALL STATION DATA
C               'LAST'
C               EPICENTER DATA - COMPUTE FOR ALL STATIONS
C               'LAST'
C           = 1 ALL EPICENTER DATA
C               'LAST'
C               STATION DATA - COMPUTE FOR ALL EPICENTERS
C               'LAST'
C
      WRITE(IWR,30)
      WRITE(IWR2,30)
   30 FORMAT(/,10X,'DISTANCE-AZIMUTH PARAMETERS',/)
      WRITE(IWR,50)IFME,IFMS,IES
      WRITE(IWR2,50)IFME,IFMS,IES
   50 FORMAT(/,10X,'IFME =',I2,10X,'IFMS =',I2,10X,'IES =',I2,/)
C
C     SET ALL DISTANCE-AZIMUTH CONSTANTS
C
      CALL DZCON
C
```

```
      IF(IES.GT.0)GO TO 5000
C
C     READ STATION DATA FIRST
C
   80 I=1
  100 IF(IFMS.GT.0)GO TO 200
C
  105 WRITE(IWR2,110)
  110 FORMAT(/,'++ INPUT: ','ISTA',/)
  120 READ(IRE,130,END=9000)ISTA(I)
  130 FORMAT(A4)
      IF(ISTA(I).EQ.LAST)GO TO 500
C
      WRITE(IWR2,135)
  135 FORMAT(/,'++ INPUT: ','SLAT',3X,'SLON',/)
      READ(IRE,*,END=9000)SLAT(I),SLON(I)
C
C     ISTA - STATION NAME
C     SLAT - STATION GEOGRAPHIC LATITUDE (IN DEGREES)
C            - = S
C     SLON - STATION GEOGRAPHIC LONGITUDE (IN DEGREES)
C            - = W
C
      CALL COMPAS(SLAT(I),SNS(I),SLON(I),SEW(I))
C
      I=I+1
      IF(I.LE.100)GO TO 105
      GO TO 400
C
  200 WRITE(IWR2,210)
  210 FORMAT(/,'++ INPUT: ','ISTA',/)
      READ(IRE,130,END=9000)ISTA(I)
      IF(ISTA(I).EQ.LAST)GO TO 500
C
      WRITE(IWR2,215)
  215 FORMAT(/,'++ INPUT: ','SDEG',1X,'SDEM',1X,'SDES',
     * 3X,'SLOD',1X,'SLOM',1X,'SLOS',/)
  250 READ(IRE,*,END=9000)SDEG,SDEM,SDES,SLOD,SLOM,SLOS
C
C     ISTA - STATION NAME
C     SDEG - STATION GEOGRAPHIC LATITUDE (IN DEGREES)
C            - = S
C     SDEM - STATION GEOGRAPHIC LATITUDE (IN MINUTES)
C     SDES - STATION GEOGRAPHIC LATITUDE (IN SECONDS)
C     SLOD - STATION GEOGRAPHIC LONGITUDE (IN DEGREES)
C            - = W
C     SLOM - STATION GEOGRAPHIC LONGITUDE (IN MINUTES)
C     SLOS - STATION GEOGRAPHIC LONGITUDE (IN SECONDS)
C
      CALL COMPAS(SDEG,SNS(I),SLOD,SEW(I))
C
      SLAT(I)=SDEG+SDEM/60.0+SDES/3600.0
      SLON(I)=SLOD+SLOM/60.0+SLOS/3600.0
C
      I=I+1
      IF(I.LE.100)GO TO 200
C
  400 WRITE(IWR,420)
```

```
      WRITE(IWR2,420)
  420 FORMAT(//,10X,'TOO MUCH STATION DATA - ONLY 100 USED',/)
      WRITE(IWR2,425)
  425 FORMAT(/,10X,'SEARCHING FOR "last"',/)
  430 READ(IRE,130,END=9000)II
      IF(II.NE.LAST)GO TO 430
C
  500 NSTA=I-1
      IF(NSTA.LE.0)GO TO 9000
C
      WRITE(IWR,550)
      WRITE(IWR2,550)
  550 FORMAT(////)
      WRITE(IWR,600)
      WRITE(IWR2,600)
  600 FORMAT(//,30X,'STATION COORDINATES',//)
      WRITE(IWR,620)(ISTA(I),SLAT(I),SNS(I),SLON(I),SEW(I),I=1,NSTA)
      WRITE(IWR2,620)(ISTA(I),SLAT(I),SNS(I),SLON(I),SEW(I),I=1,NSTA)
  620 FORMAT(20X,A4,5X,F9.3,1X,A1,5X,F9.3,1X,A1)
C
C     CONVERT TO GEOCENTRIC LATITUDE AND EAST LONGITUDE
C
      DO 800 I=1,NSTA
C
      CALL COORD1(SLAT(I),SNS(I),SLON(I),SEW(I))
C
  800 CONTINUE
C
C     READ THE EPICENTER DATA
C
 1000 IF(IFME.GT.0)GO TO 1300
C
      WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','IEPI',/)
 1200 READ(IRE,130,END=9000)IEPI(1)
      IF(IEPI(1).EQ.LAST)GO TO 80
C
      WRITE(IWR2,1210)
 1210 FORMAT(/,'++ INPUT: ','ELAT',3X,'ELON',/)
      READ(IRE,*)ELAT(1),ELON(1)
C
C     IEPT - EPICENTER ID
C     ELAT - EVENT GEOGRAPHIC LATITUDE (IN DEGREES)
C            - = S
C     ELON - EVENT GEOGRAPHIC LONGITUDE (IN DEGREES)
C            - = W
C
      CALL COMPAS(ELAT(1),ENS(1),ELON(1),EEW(1))
C
      GO TO 1400
C
 1300 WRITE(IWR2,1350)
 1350 FORMAT(/,'++ INPUT: ','IEPI',/)
      READ(IRE,130,END=9000)IEPI(1)
      IF(IEPI(1).EQ.LAST)GO TO 80
C
      WRITE(IWR2,1360)
 1360 FORMAT(/,'++ INPUT: ','EDEG',1X,'EDEM',1X,'EDES',
```

```
     * 3X,'ELOD',1X,'ELOM',1X,'ELOS',/)
       READ(IRE,*,END=9000)EDEG,EDEM,EDES,ELOD,ELOM,ELOS
C
C      IEPI - EPICENTER ID
C      EDEG - EVENT GEOGRAPHIC LATITUDE (IN DEGREES)
C             - = S
C      EDEM - EVENT GEOGRAPHIC LATITUDE (IN MINUTES)
C      EDES - EVENT GEOGRAPHIC LATITUDE (IN SECONDS)
C      ELOD - EVENT GEOGRAPHIC LONGITUDE (IN DEGREES)
C             - = W
C      ELOM - EVENT GEOGRAPHIC LONGITUDE (IN MINUTES)
C      ELOS - EVENT GEOGRAPHIC LONGITUDE (IN SECONDS)
C
       CALL COMPAS(EDEG,ENS(1),ELOD,EEW(1))
C
       ELAT(1)=EDEG+EDEM/60.0+EDES/3600.0
       ELON(1)=ELOD+ELOM/60.0+ELOS/3600.0
C
 1400  WRITE(IWR,1410)
       WRITE(IWR2,1410)
 1410  FORMAT(////,30X,'EPICENTER DATA',/)
       WRITE(IWR,1420)IEPI(1),ELAT(1),ENS(1),ELON(1),EEW(1)
       WRITE(IWR2,1420)IEPI(1),ELAT(1),ENS(1),ELON(1),EEW(1)
 1420  FORMAT(/,20X,A4,F9.3,1X,A1,10X,F9.3,1X,A1,///)
C
C      CONVERT TO GEOCENTRIC LATITUDE AND EAST LONGITUDE
C
       CALL COORD1(ELAT(1),ENS(1),ELON(1),EEW(1))
C
       WRITE(IWR,1500)
       WRITE(IWR2,1500)
 1500  FORMAT(10X,'STATION',8X,'DISTANCE',4X,'DISTANCE',5X,'E-TO-S',
     * 5X,'S-TO-E',/25X,'DEGREES',4X,'KILOMETERS',3X,'AZIMUTH',4X,
     *  'AZIMUTH'/)
C
C      COMPUTE DISTANCE-AZIMUTH
C
       DO 2000 I=1,NSTA
C
       CALL DIAZ(SLAT(I),SLON(I),ELAT(1),ELON(1),DELTA,DISKM,AZ1,AZ2)
C
       WRITE(IWR,1900)ISTA(I),DELTA,DISKM,AZ1,AZ2
       WRITE(IWR2,1900)ISTA(I),DELTA,DISKM,AZ1,AZ2
 1900  FORMAT(10X,A4,9X,F9.3,4X,F9.3,3X,F7.3,5X,F7.3)
 2000  CONTINUE
C
       GO TO 1000
C
C-----------------------------------------------------------------------
C
C      READ EPICENTER DATA
C
 5000  I=1
 5100  IF(IFME.GT.0)GO TO 5300
 5120  WRITE(IWR2,5150)
 5150  FORMAT(/,'++ INPUT:',3X,'IEPI',/)
 5200  READ(IRE,130,END=9000)IEPI(I)
       IF(IEPI(I).EQ.LAST)GO TO 5500
```

```
C
      WRITE(IWR2,5250)
 5250 FORMAT(/,'++ INPUT: ','ELAT',3X,'ELON',/)
      READ(IRE,*,END=9000)ELAT(I),ELON(I)
C
      CALL COMPAS(ELAT(I),ENS(I),ELON(I),EEW(I))
C
      I=I+1
      IF(I.LE.100)GO TO 5120
      GO TO 5400
C
 5300 WRITE(IWR2,5320)
 5320 FORMAT(/,'++ INPUT: ','IEPI',/)
      READ(IRE,130,END=9000)IEPI(I)
      IF(IEPI(I).EQ.LAST)GO TO 5500
C
      WRITE(IWR2,5330)
 5330 FORMAT(/,'++ INPUT: ','EDEG',1X,'EDEM',1X,'EDES',
     * ,3X,'ELOD',1X,'ELOM',1X,'ELOS')
 5360 READ(IRE,*,END=9000)EDEG,EDEM,EDES,ELOD,ELOM,ELOS
C
      CALL COMPAS(EDEG,ENS(I),ELOD,EEW(I))
C
      ELAT(I)=EDEG+EDEM/60.0+EDES/3600.0
      ELON(I)=ELOD+ELOM/60.0+ELOS/3600.0
C
      I=I+1
      IF(I.LE.100)GO TO 5300
C
 5400 WRITE(IWR,5420)
      WRITE(IWR2,5420)
 5420 FORMAT(//,10X,'***** TOO MUCH EPICENTER DATA - ONLY 100 ',
     * 'USED *****',/)
      WRITE(IWR2,5425)
 5425 FORMAT(/,10X,'SEARCHING FOR "last"',/)
 5430 READ(IRE,130,END=9000)II
      IF(II.NE.LAST)GO TO 5430
C
 5500 NEPI=I-1
      IF(NEPI.LE.0)GO TO 9000
C
      WRITE(IWR,550)
      WRITE(IWR2,550)
      WRITE(IWR,5600)
      WRITE(IWR2,5600)
 5600 FORMAT(//,30X,'EPICENTER COORDINATES',//)
      WRITE(IWR,5620)(IEPI(I),ELAT(I),ENS(I),ELON(I),EEW(I),
     * I=1,NEPI)
      WRITE(IWR2,5620)(IEPI(I),ELAT(I),ENS(I),ELON(I),EEW(I),
     * I=1,NEPI)
 5620 FORMAT(20X,A4,5X,F9.3,1X,A1,15X,F9.3,1X,A1)
C
C     CONVERT TO GEOCENTRIC LATITUDE AND EAST LONGITUDE
C
      DO 5800 I=1,NEPI
C
      CALL COORD1(ELAT(I),ENS(I),ELON(I),EEW(I))
C
```

```
 5800 CONTINUE
C
C     READ STATION DATA
C
 6000 IF(IFMS.GT.0)GO TO 6200
C
      WRITE(IWR2,6010)
 6010 FORMAT(/,'++ INPUT: ','ISTA',/)
      READ(IRE,130,END=9000)ISTA(1)
      IF(ISTA(1).EQ.LAST)GO TO 5000
C
      WRITE(IWR2,6020)
 6020 FORMAT(/,'++ INPUT: ','SLAT',3X,'SLON',/)
      READ(IRE,*,END=9000)SLAT(1),SLON(1)
C
      CALL COMPAS(SLAT(1),SNS(1),SLON(1),SEW(1))
C
      GO TO 6400
C
 6200 WRITE(IWR2,6250)
 6250 FORMAT(/,'++ INPUT: ','ISTA',/)
      READ(IRE,130,END=9000)ISTA(1)
      IF(ISTA(1).EQ.LAST)GO TO 5000
C
      WRITE(IWR2,6260)
 6260 FORMAT(/,'++ INPUT: ','SDEG',1X,'SDEM',1X,'SDES',
     * 3X,'SLOD',1X,'SLOM',1X,'SLOS',/)
      READ(IRE,*,END=9000)SDEG,SDEM,SDES,SLOD,SLOM,SLOS
C
      CALL COMPAS(SDEG,SNS(1),SLOD,SEW(1))
C
      SLAT(1)=SDEG+SDEM/60.0+SDES/3600.0
      SLON(1)=SLOD+SLOM/60.0+SLOS/3600.0
C
 6400 WRITE(IWR,6410)
      WRITE(IWR2,6410)
 6410 FORMAT(////,30X,'STATION DATA'/)
      WRITE(IWR,6450)ISTA(1),SLAT(1),SNS(1),SLON(1),SEW(1)
      WRITE(IWR2,6450)ISTA(1),SLAT(1),SNS(1),SLON(1),SEW(1)
 6450 FORMAT(/,20X,A4,F9.3,1X,A1,10X,F9.3,1X,A1,///)
C
C     CONVERT TO GEOCENTRIC LATITUDE AND EAST LONGITUDE
C
      CALL COORD1(SLAT(1),SNS(1),SLON(1),SEW(1))
C
      WRITE(IWR,6500)
      WRITE(IWR2,6500)
 6500 FORMAT(12X,'EVENT',8X,'DISTANCE',4X,'DISTANCE',5X,'E-TO-S',
     * 5X,'S-TO-E',/,25X,'DEGREES',4X,'KILOMETERS',3X,'AZIMUTH',4X,
     * 'AZIMUTH',/)
C
C     COMPUTE DISTANCE-AZIMUTH
C
      DO 8000 I=1,NEPI
C
      CALL DIAZ(SLAT(1),SLON(1),ELAT(I),ELON(I),DELTA,DISKM,AZ1,AZ2)
C
      WRITE(IWR,7500)IEPI(I),DELTA,DISKM,AZ1,AZ2
```

```
      WRITE(IWR2,7500)IEPI(I),DELTA,DISKM,AZ1,AZ2
 7500 FORMAT(12X,A4,7X,F9.3,4X,F9.3,4X,F7.3,4X,F7.3)
 8000 CONTINUE
C
      GO TO 6000
C
 9000 WRITE(IWR,550)
      WRITE(IWR2,550)
C
      RETURN
      END
```

```
      SUBROUTINE DECI(DATA,ISAMP,IFACT)
C
C*****************************************************************************
C
C     DECIMATE DATA
C
C     DATA - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF POINTS IN TIME SERIES - INPUT AND OUTPUT
C     IFACT - DECIMATE FACTOR - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION DATA(1)
C
      K=0
C
      DO 100 J=1,ISAMP,IFACT
      K=K+1
      DATA(K)=DATA(J)
  100 CONTINUE
C
      ISAMP=K
C
      RETURN
      END
```

```
      SUBROUTINE DECIX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR DECIMATING DATA
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/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
      IFL=0
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',3X,'IFACT',3X,'SAMR',/)
      READ(IRE,*,END=9000)ICHAN,IFACT,SAMR
C
C     ICHAN - DESIRED CHANNEL
C             = 0 DECIMATE ALL CHANNELS
C     IFACT - DECIMATION FACTOR
C     SAMR - SAMPLING RATE (IN SAMPLES/S)
C            > 0 DO NOT DECIMATE THOSE CHANNELS HAVING
C                THIS SAMPLING RATE
C            = 0 DECIMATE ALL CHANNELS DESIGNATED BY ICHAN
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'DECIMATION PARAMETERS',/)
      WRITE(IWR,300)ICHAN,IFACT,SAMR
      WRITE(IWR2,300)ICHAN,IFACT,SAMR
  300 FORMAT(/,10X,'ICHAN = ',I4,5X,'IFACT = ',I3,5X,'SAMR = ',F6.2,/)
C
      IF(IFACT.LE.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
```

```
C     DECIMATE DATA
C
 2000 DO 3000 I=NCH1,NCH2
      ISAMP=JSAMP(I)
      INDEX=(I-1)*LENG+1
      IF(RSAMR(I).EQ.SAMR)GO TO 3000
C
      CALL DECI(DATA(INDEX),ISAMP,IFACT)
C
      IFL=1
      WRITE(IWR2,2500)JSAMP(I),ISAMP,I
 2500 FORMAT(10X,'OLD SAMPLES = ',I6,3X,'NEW SAMPLES = ',I6,
     * ' FOR CHANNEL ',I4,/)
      JSAMP(I)=ISAMP
      RSAMR(I)=RSAMR(I)/IFACT
 3000 CONTINUE
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
 7200 WRITE(IWR2,7300)
 7300 FORMAT(/,10X,'DATA NOT DECIMATED',/)
      GO TO 9000
C
 8000 IF(IFL.EQ.0)GO TO 7200
      WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY DECIMATED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE DELEX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR DELETING A DATA CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 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/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
      DATA NY/'y'/
C
C     READ DELETE PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL TO BE DELETED
C
      IF(ICHAN.LE.0.OR.ICHAN.GT.NACHLS)GO TO 7000
      WRITE(IWR,200)ICHAN
      WRITE(IWR2,200)ICHAN
  200 FORMAT(/,10X,'DELETE CHANNEL ',I4,' ?   y/n',/)
      READ(IRE,300,END=9000)KY
  300 FORMAT(A1)
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 9000
C
      IFL=0
      IB1=1
      IL1=LENG
      IB2=1
      NAC=NACHLS-1
      IHF=0
C
      DO 2000 I=ICHAN,NAC
      NCH1=I+1
      NCH2=I
C
      CALL MOVE(IFL,IB1,IL1,NCH1,IB2,NCH2,IHF)
C
```

```
 2000 CONTINUE
C
      NACHLS=NACHLS-1
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)ICHAN
 7200 FORMAT(/,10X,'DATA CHANNEL ',I6,' NOT DELETED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)ICHAN
      WRITE(IWR2,8100)ICHAN
 8100 FORMAT(/,10X,'DATA CHANNEL ',I6,' SUCCESSFULLY DELETED',/)
      WRITE(IWR,8200)
      WRITE(IWR2,8200)
 8200 FORMAT(/,10X,'NEW CHANNELS ARE:',//)
      WRITE(IWR,8300)(I,JCODE(I),I=1,NACHLS)
      WRITE(IWR2,8300)(I,JCODE(I),I=1,NACHLS)
 8300 FORMAT(10X,I6,5X,A4)
      WRITE(IWR,8400)
      WRITE(IWR2,8400)
 8400 FORMAT(//)
C
 9000 RETURN
      END
```

```
      SUBROUTINE DELTT(CTT,DTT,IDELT,NCH,FNOISE)
C
C*********************************************************************
C
C     CALCULATE TIME DELAYS FOR BEAMFORMING
C
C     CTT - ARRIVAL TIME OF SIGNAL (IN SECONDS) - INPUT
C     DTT - TIME DELAY BEFORE SIGNAL OF CHANNEL (IN SECONDS) - OUTPUT
C     IDELT - ARRAY OF TIME DELAYS (IN SAMPLES) - OUTPUT
C     NCH - NUMBER OF CHANNELS TO BEAM - INPUT
C           (BEAM CHANNELS 1 THROUGH NCH)
C     FNOISE - SECONDS OF NOISE TO INCLUDE BEFORE SIGNAL - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 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
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 CTT(1),DTT(1),IDELT(1)
C
      COMPLEX RPOLES,RZEROS
C
      DO 500 I=1,NCH
C
C     COMPUTE START TIME OF CHANNEL (IN SECONDS)
C
      CALL TIMC(ST,JHOUR(I),JMIN(I),RSEC(I))
C
C     COMPUTE TIME DELAYS
C
      DTT(I)=CTT(I)-ST-FNOISE
      IF(DTT(I).GE.86400.0)DTT(I)=DTT(I)-86400.0
      IDELT(I)=DTT(I)*RSAMR(I)
  500 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DESP(X,ISAMP,ISPIKE)
C
C*********************************************************************
C
C     DESPIKE ONE DATA CHANNEL
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C     ISPIKE - NUMBER OF SPIKES FOUND AND ELIMINATED - OUTPUT
C
C     OCTOBER 15, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1)
C
      ISPIKE=0
      RATIO=10.0
C
      DO 1000 J=3,ISAMP
      DENOM=ABS(X(J)-X(J-2))
      IF(DENOM.LT.1.0)DENOM=1.0
      IF((ABS(X(J-1)-X(J-2))/DENOM).LT.RATIO)GO TO 1000
      X(J-1)=(X(J)+X(J-2))/2.0
      ISPIKE=ISPIKE+1
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE DESPIK
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR DESPIKING ONE OR MORE DATA CHANNELS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 15, 1993
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/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
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL NUMBER TO DESPIKE
C             = 0 DESPIKE ALL CHANNELS
C
      WRITE(IWR,200)ICHAN
      WRITE(IWR2,200)ICHAN
  200 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
      IF(ICHAN.LT.0)GO TO 7000
C
      IF(ICHAN.EQ.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 DESP(DATA(INDEX),ISAMP,ISPIKE)
C
      WRITE(IWR2,2500)ISPIKE,I
 2500 FORMAT(/,10X,'NUMBER OF SPIKES FOUND = ',I6,' FOR CHANNEL ',
     * I4,/)
```

```
 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 DESPIKED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY DESPIKED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE DIAZ(ELAT1,ELON1,ELAT2,ELON2,DELTA,DISKM,AZ1,AZ2)
C
C*****************************************************************
C
C     COMPUTE DISTANCE AND AZIMUTH BETWEEN TWO POINTS
C
C     ELAT1 - CO-LATITUDE OF ONE COORDINATE (IN RADIANS) - INPUT
C     ELON1 - EAST LONGITUDE OF ONE COORDINATE (IN RADIANS) - INPUT
C     ELAT2 - CO-LATITUDE OF OTHER COORDINATE (IN RADIANS) - INPUT
C     ELON2 - EAST LONGITUDE OF OTHER COORDINATE (IN RADIANS) - INPUT
C     DELTA - DISTANCE (IN DEGREES) - OUTPUT
C     DISKM - DISTANCE (IN KILOMETERS) - OUTPUT
C     AZ1 - AZIMUTH FROM ELAT2 ELON2 TO ELAT1 ELON1
C           (IN DEGREES) - OUTPUT
C     AZ2 - AZIMUTH FROM ELAT1 ELON1 TO ELAT2 ELON2
C           (IN DEGREES) - OUTPUT
C
C     AUGUST 4, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      A=SIN(ELAT1)
      B=COS(ELAT1)
      C=SIN(ELAT2)
      D=COS(ELAT2)
      ARG=ABS(ELON1-ELON2)
      VE=COS(ARG)
      DELTA=B*D+A*C*VE
C
      CALL SARCOS(DELTA)
C
      SIND=SIN(DELTA)
      IF(SIND.LT.0.0)SIND=0.0
      IF(SIND.NE.0.0)GO TO 60
C
      AZ1=0.0
      AZ2=TWOPI
      GO TO 100
C
   60 IF(SIND.LT.0.0001)SIND=0.0001
      C=B*C
      D=A*D
      AZ1=(C-D*VE)/SIND
      AZ2=(D-C*VE)/SIND
C
      CALL SARCOS(AZ1)
      CALL SARCOS(AZ2)
C
C     RESOLVE AZIMUTH AMBIGUITIES
C
      IF(ARG.LE.PI)GO TO 70
      IF(ELON2-ELON1)80,100,90
   70 IF(ELON2.LE.ELON1)GO TO 90
   80 AZ1=TWOPI-AZ1
      GO TO 100
```

```
   90 AZ2=TWOPI-AZ2
C
C     CONVERT TO DEGREES
C
  100 AZ1=AZ1*RD2DG
      AZ2=AZ2*RD2DG
      DELTA=DELTA*RD2DG
      DISKM=DELTA*DG2KM
C
      RETURN
      END
```

```
      SUBROUTINE DPARA
C
C*****************************************************************************
C
C     SET DATA PARAMETERS NCHLS, LENG, AND NACHLS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 30, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/PCKF/ICHECK,IPLFL
C
      WRITE(IWR2,20)
   20 FORMAT(/,'++ INPUT: ','NC',3X,'LE',3X,'NAC',/)
      READ(IRE,*,END=9000)NC,LE,NAC
C
C     NC - NEW NUMBER OF CHANNELS
C          = 0 DO NOT CHANGE NCHLS
C     LE - NEW TOTAL STORAGE RESERVED FOR EACH CHANNEL
C          = 0 DO NOT CHANGE LENG
C     NAC - NEW NUMBER OF ACTUAL CHANNELS
C           = 0 DO NOT CHANGE NACHLS
C
      IF(NC.GT.0)NCHLS=NC
      IF(LE.GT.0)LENG=LE
      IF(NAC.GT.0)NACHLS=NAC
C
      WRITE(IWR,50)
      WRITE(IWR2,50)
   50 FORMAT(/,10X,'NEW DATA STORAGE PARAMETERS',/)
      WRITE(IWR,100)NCHLS,LENG,NACHLS
      WRITE(IWR2,100)NCHLS,LENG,NACHLS
  100 FORMAT(/,10X,'NCHLS = ',I4,5X,'LENG = ',I6,5X,'NACHLS = ',I4,/)
C
      IC=NCHLS*LENG
      IF(IC.LE.IDEM)GO TO 9000
C
 1000 WRITE(IWR,1100)
      WRITE(IWR2,1100)
 1100 FORMAT(/,10X,'***** WARNING - DATA PARAMETERS EXCEED ',
     * 'IDEM *****',/)
      ICHECK=1
C
 9000 RETURN
      END
```

```
      SUBROUTINE DZCON
C
C*****************************************************************************
C
C     SET ALL DISTANCE-AZIMUTH COMPUTATION CONSTANTS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 14, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DATA JNN/'n'/,JSS/'s'/,JEE/'e'/,JWW/'w'/
C
      PI=3.14159265
      RD2DG=57.2957795
      DG2RD=0.017453292
      DG2KM=111.195
      GEOCO1=0.192433
      GEOCO2=0.000325
      TWOPI=6.28318530
      KNN=JNN
      KSS=JSS
      KEE=JEE
      KWW=JWW
C
      RETURN
      END
```

```
      SUBROUTINE EETIME(START,ORGT,TDIF,ICHAN,IFIR)
C
C*********************************************************************
C
C     DETERMINE CHANNEL AND EVENT TIMES IN SECONDS
C
C     START - START TIME OF DATA CHANNEL (IN SECONDS) - OUTPUT
C     ORGT - ORIGIN TIME OF EVENT (IN SECONDS) - OUTPUT
C     TDIF - (START-ORGT) TIME DIFFERENCE (IN SECONDS) - OUTPUT
C     ICHAN - DESIRED CHANNEL - INPUT
C     IFIR - FIRST SAMPLE DESIRED - INPUT
C            = 0 SET TO 1
C
C     WRITTEN BY ROBERT P. MASSE
C
C     MARCH 1, 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/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      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)
      DIMENSION EXFLAG(10)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      IF(ICHAN.LE.0.OR.ICHAN.GT.NACHLS)GO TO 9000
      IF(IFIR.LE.0)IFIR=1
C
      CALL TIMC(START,JHOUR(ICHAN),JMIN(ICHAN),RSEC(ICHAN))
C
      CALL TIMC(ORGT,IEXHR,IEXMN,EXSEC)
C
      CALL JUL(IEXYR,IEXMO,IEXDY,IDOY,LEAP)
C
      ICK=JDOFY(ICHAN)-IDOY
      IF(IABS(ICK).LT.300)GO TO 500
C
      IF(ICK.GT.0)GO TO 200
C
      JDOFY(ICHAN)=JDOFY(ICHAN)+365
      IF(LEAP.NE.0)JDOFY(ICHAN)=JDOFY(ICHAN)+1
      GO TO 500
C
  200 IMO=1
      IDY=1
C
      CALL JUL(JYEAR(ICHAN),IMO,IDY,IDD,LEAP)
C
```

```
      IDOY=IDOY+365
      IF(LEAP.NE.0)IDOY=IDOY+1
C
  500 SAMR=RSAMR(ICHAN)
      START=START+JDOFY(ICHAN)*86400.0+(IFIR-1)/SAMR
      ORGT=ORGT+IDOY*86400.0
C
      TDIF=START-ORGT
C
 9000 RETURN
      END
```

```
      SUBROUTINE ENDPF
C
C*********************************************************************
C
C     TERMINATE PLOTTING FUNCTIONS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 27, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/PCKF/ICHECK,IPLFL
C
      IF(IPLFL.EQ.0)GO TO 9000
C
      IPLFL=0
      ISW=-1
C
      CALL HLDSET(ISW)
C
      CALL ENDPLT
C
 9000 RETURN
      END
```

```
      SUBROUTINE ENGL(TTT,CCC,R11,R21,MODENO,MMAX,WVNO,WVNOSQ,
     * ARGK,IPARMO,IPDER,FI0L,FI1L,FI2L,MM,LCT,KWRITE)
C
C*****************************************************************
C
C     COMPUTE THE ENERGY INTEGRALS AND THE PARTIAL DERIVATIVES
C
C     TTT - PERIOD OF RAYLEIGH WAVE (IN SECONDS) - INPUT
C     CCC - TRIAL PHASE VELOCITY VALUE (IN KM/S) - INPUT
C     R11 - HASKELL MATRIX TERM (1,1) - INPUT
C     R21 - HASKELL MATRIX TERM (2,1) - 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     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     WVNO - WAVE NUMBER - INPUT
C     WVNOSQ - WAVE NUMBER SQUARED - INPUT
C     ARGK - S WAVE ARGUMENT - INPUT
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     MM - LAYER NUMBER - INPUT AND OUTPUT
C     LCT - NUMBER OF SUBDIVISIONS TO THE CURRENT LAYER - INPUT AND
C           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
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 19, 1994  COLORADO
C
C*****************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION V(1000,4),A(2,2),FIPD1(1500),FIPD2(1500),FIPD3(1500)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(40001),V(1,1)),(BX(50801),A(1,1))
      EQUIVALENCE (BX(51001),FIPD1(1)),(BX(53001),FIPD2(1))
      EQUIVALENCE (BX(55001),FIPD3(1))
C
      REAL*8 A,R11,R21,SR11,SR21,ARGK,TKSQ,TK,FC1,FC2
C
      M=MM
```

```
      FLCT=LCT
      IF(LCT.LE.0)FLCT=1.0
      TKSQ=DABS(ARGK)
      TK=DSQRT(TKSQ)
      IF(ARGK.LT.0.0)TKSQ=-TKSQ
      FMU=V(M,3)*V(M,3)*V(M,4)
C
C     IS FIRST LAYER?
C
      IF(M.GT.1)GO TO 2000
C
C     INITIALIZE PARAMETERS FOR FIRST LAYER
C
      IFIN=0
      WLGTH=TTT*CCC
      CSQ=CCC*CCC
      SR11=R11
      SR21=R21
      MODE=MODENO
      IFGDV=1
      DEPTH=V(1,1)
C
      DO 300 I=1,1500
      FIPD1(I)=0.0
      FIPD2(I)=0.0
      FIPD3(I)=0.0
  300 CONTINUE
C
C     COMPUTE INTEGRALS FOR LAYER 1
C
      FC1=WVNO*FMU*TKSQ*V(1,1)
      FC2=WVNO*TKSQ*FMU/V(1,4)
      FI0L=0.5*(FC1+A(2,1)*A(1,1))
      FI0L=FI0L/FC2
      FI1L=FMU*FI0L/V(1,4)
      FI2L=0.5*WVNO*(FC1-A(2,1)*A(1,1))
C
C     CHECK SIGN OF PARTICLE MOTION
C
      IF(R11.GT.0.0)GO TO 500
      MODE=MODE-1
C
C     CHECK IF LISTING OF PARTICLE MOTION AS A FUNCTION OF DEPTH
C     IS DESIRED
C
  500 IF(IPARMO.LE.0)GO TO 1000
C
C     LIST PARTICLE MOTION
C
      WRITE(IWR,600)TTT,CCC,WLGTH
      WRITE(IWR2,600)TTT,CCC,WLGTH
  600 FORMAT(/,10X,'PERIOD = ',F8.3,5X,'PHASE VEL = ',F9.5,/,10X,
     * 'WAVE LENGTH = ',F10.5,/)
      WRITE(IWR,700)
      WRITE(IWR2,700)
  700 FORMAT(20X,'DEPTH',10X,'VM/V0',/)
      WRITE(IWR,800)DEPTH,A(1,1)
      WRITE(IWR2,800)DEPTH,A(1,1)
```

```
  800 FORMAT(14X,F10.5,2X,D15.5)
C
C     IS COMPUTATION OF PARTIAL DERIVATIVES DESIRED?
C
 1000 IF(IPDER-1)9000,1200,1400
C
C     COMPUTE DERIVATIVES FOR LAYER 1
C
 1200 FIPD1(M)=A(1,1)*A(1,1)
      FIPD2(M)=FMU*FIPD1(M)
      FIPD1(M)=V(1,4)*FIPD1(M)
      FIPD3(M)=WVNOSQ*A(2,1)*A(2,1)/FMU
      GO TO 9000
C
 1400 IF(IPDER-3)9000,1500,1600
 1500 FIPD2(M)=2.0*FI1L/V(1,3)
      FIPD3(M)=2.0*FI2L/V(1,3)
      GO TO 9000
C
 1600 IF(IPDER-4)9000,1700,9000
 1700 FIPD1(M)=FI0L/V(1,4)
      FIPD2(M)=FI1L/V(1,4)
      FIPD3(M)=FI2L/V(1,4)
      GO TO 9000
C
C     IS THIS HALF-SPACE?
C
 2000 IF(M.GE.MMAX)GO TO 5000
      DEPTH=DEPTH+V(M,1)/FLCT
C
C     COMPUTE FOR LAYER M
C
C     CHECK IF ENERGY INTEGRAL PRECISION ALLOWS CONTINUATION TO
C     GREATER DEPTHS
C
      IF(IFIN.NE.0)GO TO 9000
C
C     CHECK SIGN
C
      IF((R11*SR11).GT.0.0)GO TO 2300
      MODE=MODE-1
      IF(MODE.GE.0)GO TO 2300
 2200 IFIN=1
      IFGDV=M-2
      LCT=0
      MM=MMAX-1
      GO TO 9000
C
 2300 IF(DABS(R11).GE.10000.0)GO TO 2200
C
C     COMPUTE INTEGRALS FOR LAYER M
C
 2500 FC1=WVNO*FMU*TKSQ*V(M,1)/FLCT
      FC2=WVNO*TKSQ*FMU/V(M,4)
      FC3=WVNO*V(M,1)/(FMU*FLCT)
      FINT1=0.5*SR11*SR11*(FC1+A(2,1)*A(1,1))
      FINT2=SR11*SR21*(A(1,1)*A(1,1)-1.0)
      FINT3=0.5*SR21*SR21*(FC3+A(1,2)*A(1,1))
```

```
      FINT=FINT1+FINT2+FINT3
      FINT=FINT/FC2
      SFI0L=FI0L+FINT
      SFI1L=FI1L+FMU*FINT/V(M,4)
      FINTP=FINT
      FINT1=0.5*WVNO*SR11*SR11*(FC1-A(2,1)*A(1,1))
      FINT2=WVNO*SR11*SR21*(1.0-A(1,1)*A(1,1))
      FINT3=0.5*WVNO*SR21*SR21*(FC3-A(1,2)*A(1,1))
      FINT=FINT1+FINT2+FINT3
      SFI2L=FI2L+FINT
C
C     FORM POTENTIAL AND KINETIC ENERGY INTEGRALS (PARTIAL SUM)
C
      SEP=(WVNOSQ*SFI1L+SFI2L)*0.5
      SEK=WVNOSQ*CSQ*SFI0L*0.5
C
      IF(KWRITE.EQ.0)GO TO 2600
      WRITE(IWR2,2550)M,SEK,SEP
 2550 FORMAT(10X,'M = ',I3,3X,'SEK = ',E16.9,3X,'SEP = ',E16.9)
C
C     CHECK EQUALITY OF KINETIC AND POTENTIAL ENERGY
C
 2600 IFGDV=M
C
C     ARE PREVIOUS VALUES OF ENERGY AVAILABLE?
C
      IF(M.GT.2)GO TO 3000
C
C     IFGDV - INDICATES GREATEST DEPTH FOR WHICH INTEGRATION IS VALID
C
 2800 IF((ABS((SEP-SEK)/SEP)).GE.0.00001)GO TO 3200
C
C     COMPARE NEW ENERGY INTEGRAL DIFFERENCE WITH ENERGY INTEGRAL
C     DIFFERENCE FOR LAYERS ABOVE
C
 3000 IF(ABS(SEK-SEP).LE.ABS(EK-EP))GO TO 3200
      IF(DEPTH.GT.((MODENO+2)*WLGTH))GO TO 2200
C
C     SET ENERGY CONSTANTS FOR NEXT LAYER COMPUTATION
C
 3200 EK=SEK
      EP=SEP
      FI0L=SFI0L
      FI1L=SFI1L
      FI2L=SFI2L
C
C     IS LISTING OF PARTICLE MOTION WITH DEPTH DESIRED?
C
      IF(IPARMO.LE.0)GO TO 3500
C
C     LIST PARTICLE MOTION
C
      WRITE(IWR,800)DEPTH,R11
      WRITE(IWR2,800)DEPTH,R11
C
C     COMPUTE DERIVATIVE TERMS FOR LAYER M
C
 3500 IF(IPDER-1)4800,3600,4000
```

```
 3600 FIP=SR11*SR11*A(1,1)*A(1,1)+2.0*SR11*SR21*A(1,2)*A(1,1)
     * +SR21*SR21*A(1,2)*A(1,2)
      FIPD1(M)=V(M,4)*FIP
      FIPD2(M)=FMU*FIP
      FIP=SR11*SR11*A(2,1)*A(2,1)+2.0*SR11*SR21*A(2,1)*A(1,1)
     * +SR21*SR21*A(1,1)*A(1,1)
      FIPD3(M)=WVNOSQ*FIP/FMU
      GO TO 4400
C
 4000 IF(IPDER-3)4800,4100,4200
 4100 FIPD2(M)=2.0*V(M,3)*FINTP
      FIPD3(M)=2.0*FINT/V(M,3)
      GO TO 4800
C
 4200 IF(IPDER-4)4800,4300,4800
 4300 FIPD1(M)=FINTP/V(M,4)
      FIPD2(M)=FINTP*FMU/(V(M,4)*V(M,4))
      FIPD3(M)=FINT/V(M,4)
      GO TO 4800
C
 4400 FIP=SR11*SR11*A(1,1)*A(1,1)+2.0*SR11*SR21*A(1,2)*A(1,1)
     * +SR21*SR21*A(1,2)*A(1,2)
C
      MMM=M-1
      FIPD1(MMM)=-V(M,4)*FIP+FIPD1(MMM)
      FIPD2(MMM)=-FMU*FIP+FIPD2(MMM)
      FIP=SR11*SR11*A(2,1)*A(2,1)+2.0*SR11*SR21*A(2,1)*A(1,1)
     * +SR21*SR21*A(1,1)*A(1,1)
      FIPD3(MMM)=-WVNOSQ*FIP/FMU+FIPD3(MMM)
C
 4800 SR11=R11
      SR21=R21
      GO TO 9000
C
C     COMPUTE INTEGRALS FOR HALF-SPACE
C
 5000 R11=SR11
      R21=SR21
C
C     IS INTEGRAL COMPUTATION ALREADY TERMINATED?
C
      IF(IFIN.NE.0)GO TO 5300
C
C     CHECK FOR LIQUID HALF-SPACE
C
      IF(V(M,3).LE.0.0)GO TO 5300
C
      FINT=0.5*V(M,4)*R11*R11/(WVNO*TK)
      FI0L=FI0L+FINT
      FI1L=FI1L+FMU*FINT/V(M,4)
      FI2L=FI2L+WVNO*R21*R21*0.5/(FMU*TK)
C
      IF(IPDER.LT.1)GO TO 9000
C
C     FORM AND LIST ALL PARTIAL DERIVATIVES
C
      WRITE(IWR,600)TTT,CCC,WLGTH
      IF(IPARA.GT.1)GO TO 5300
```

```
      MMM=MMAX-1
      FIPD1(MMM)=FIPD1(MMM)-V(M,4)*R11*R11
      FIPD2(MMM)=FIPD2(MMM)-FMU*R11*R11
      FIPD3(MMM)=FIPD3(MMM)-WVNOSQ*R21*R21/FMU
C
 5300 IF(IFGDV.GE.0)GO TO 5500
      WRITE(IWR,5400)
      WRITE(IWR2,5400)
 5400 FORMAT(/,10X,'SOLUTION NOT ACCURATE ENOUGH FOR FORMATION',/,
     * 10X,'OF DESIRED MEDIUM DERIVATIVES',/)
      GO TO 9000
C
 5500 IF(IPDER-1)9000,5600,6000
 5600 WRITE(IWR,5700)
      WRITE(IWR2,5700)
 5700 FORMAT(/,10X,'LAYER NO.',10X,'(DC/DH)',10X,'(DU/DH)')
      WRITE(IWR,5800)
      WRITE(IWR2,5800)
 5800 FORMAT(31X,'T,RHO,BETA',26X,'T,RHO,BETA',/)
      GO TO 6750
C
 6000 IF(IPDER.LE.2)GO TO 9000
      IF(IPDER-3)9000,6100,6500
 6100 WRITE(IWR,6200)
      WRITE(IWR2,6200)
 6200 FORMAT(/,10X,'LAYER NO.',10X,'(DC/DBETA)',10X,'(DU/DBETA)')
      WRITE(IWR,6300)
      WRITE(IWR2,6300)
 6300 FORMAT(30X,'T,H,RHO',32X,'T,H,RHO',/)
      GO TO 6750
C
 6500 IF(IPDER-4)9000,6600,6750
 6600 WRITE(IWR,6650)
      WRITE(IWR2,6650)
 6650 FORMAT(/,10X,'LAYER NO.',10X,'(DC/DRHO)',10X,'(DU/DRHO)')
      WRITE(IWR,6700)
      WRITE(IWR2,6700)
 6700 FORMAT(30X,'T,H,RHO',31X,'T,H,RHO',/)
C
 6750 AAX=0.5*CCC/FI1L
      BBX=0.5/(CCC*FI0L)
C
      DO 6850 I=1,IFGDV
      CDER=AAX*(-CSQ*FIPD1(I)+FIPD2(I)+(FIPD3(I)/WVNOSQ))
      UDER=0.0
      WRITE(IWR,6800)I,CDER,UDER
      WRITE(IWR2,6800)I,CDER,UDER
 6800 FORMAT(10X,I4,10X,E15.8,10X,E15.8)
 6850 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE EVTRD
C
C*********************************************************************
C
C     READS EVENT INFORMATION AND STORES IN COMMON
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 13, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION EXFLAG(10)
C
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','EXNAME',/)
      READ(IRE,300,END=9000)EXNAME
  300 FORMAT(A20)
C
C     EXNAME - EVENT NAME (20 CHARACTERS MAXIMUM)
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','EXLAT',3X,'EXLON',3X,'EXDEPH',/)
      READ(IRE,*,END=9000)EXLAT,EXLON,EXDEPH
C
C     EXLAT - EVENT GEOGRAPHIC LATITUDE (IN DEGREES)
C             - = S
C     EXLON - EVENT GEOGRAPHIC LONGITUDE (IN DEGREES)
C             - = W
C     EXDEPH - EVENT HYPOCENTER DEPTH (IN KILOMETERS)
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','IEXYR',3X,'IEXMO',3X,'IEXDY',3X,
     * 'IEXHR',3X,'IEXMN',3X,'EXSEC',/)
      READ(IRE,*,END=9000)IEXYR,IEXMO,IEXDY,IEXHR,IEXMN,EXSEC
C
C     IEXYR - EVENT ORIGIN TIME - YEAR
C     IEXMO - EVENT ORIGIN TIME - MONTH
C     IEXDY - EVENT ORIGIN TIME - DAY
C     IEXHR - EVENT ORIGIN TIME - HOUR
C     IEXMN - EVENT ORIGIN TIME - MINUTE
C     EXSEC - EVENT ORIGIN TIME - SECONDS
C
      WRITE(IWR2,800)
  800 FORMAT(/,'++ INPUT: ','EXMAG1',3X,'EXMAG2',/)
      READ(IRE,*,END=9000)EXMAG1,EXMAG2
C
C     EXMAG1 - MAGNITUDE FOR EVENT
C     EXMAG2 - MAGNITUDE FOR EVENT
C
      WRITE(IWR2,900)
```

```
  900 FORMAT(/,'++ INPUT: ','IEXM1',3X,'IEXM2',/)
      READ(IRE,950,END=9000)IEXM1,IEXM2
  950 FORMAT(2A4)
C
C     IEXM1 - MAGNITUDE EXMAG1 NAME
C     IEXM2 - MAGNITUDE EXMAG2 NAME
C
      WRITE(IWR2,1000)
 1000 FORMAT(/,'++ INPUT: ','EXCOMT',/)
      READ(IRE,300,END=9000)EXCOMT
C
C     EXCOMT - COMMENTS (20 CHARACTERS MAXIMUM)
C
      WRITE(IWR2,1100)
 1100 FORMAT(/,'++ INPUT: ','IFL',/)
      READ(IRE,*,END=9000)IFL
C
C     IFL - READ EXFLAG FLAG
C           = 0 DO NOT READ 10 EVENT FLAGS
C
      IF(IFL.LE.0)GO TO 2000
      WRITE(IWR2,1200)
 1200 FORMAT(/,'++ INPUT: ','EXFLAG',/)
      READ(IRE,*,END=9000)EXFLAG
C
C     EXFLAG - 10 EVENT FLAGS
C
 2000 CALL PREVT
C
 9000 RETURN
      END
```

```
      SUBROUTINE EXPWF(X,ISAMP,SAMR,ALPHA)
C
C*********************************************************************
C
C     EXPONENTIALLY WEIGHT A TIME SERIES
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES TO PROCESS - INPUT
C     SAMR - SAMPLING RATE OF TIME SERIES - INPUT
C     ALPHA - WEIGHTING FACTOR exp(-alpha * t) - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 24, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1)
C
      ARG=-ALPHA/SAMR
C
      DO 500 J=1,ISAMP
      FJ=J-1
      X(J)=X(J)*EXP(ARG*FJ)
  500 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE FELIP(DIST,F)
C
C*********************************************************************
C
C     DETERMINE ELIPTICITY DISTANCE FUNCTION F
C
C     DIST - DISTANCE (IN DEGREES) - INPUT
C     F - ELLIPICITY DISTANCE FUNCTION F - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 30, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION D(19),FD(19)
C
      DATA D/0.0,10.0,20.0,30.0,40.0,50.0,60.0,70.0,80.0,90.0,100.0,
     * 105.0,110.0,142.5,145.0,150.0,160.0,170.0,180.0/
      DATA FD/0.0,0.007,0.023,0.029,0.034,0.040,0.043,0.052,0.062,0.068,
     * 0.069,0.070,0.100,0.100,0.092,0.093,0.094,0.095,0.095/
C
      DO 10 I=2,19
      IF(DIST.LE.D(I))GO TO 20
   10 CONTINUE
C
C     LINEAR INTERPOLATION TO DETERMINE F
C
   20 F=(DIST-D(I-1))*(FD(I)-FD(I-1))/(D(I)-D(I-1))+FD(I-1)
C
      RETURN
      END
```

```
      SUBROUTINE FILLCH(ICHAN,NSAMP)
C
C*********************************************************************
C
C     FILL IN DESIRED CHANNEL WITH DATA IN BX ARRAY
C
C     ICHAN - DESIRED CHANNEL TO FILL - INPUT
C     NSAMP - NUMBER OF ACTUAL SAMPLES IN BX - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 12, 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/HEAD/CODE,CHN,LAT,LON,ELEV,SYEAR,SDOFY,SHOUR,SMIN,
     * SSECS,RATE,A0,NP,POLES,NZ,ZEROES,NFLAG
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/STAT/ISTAT
      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 CODE(4),CHN(4),POLES(30),ZEROES(20),NFLAG(3),BX(1),
     * ISTAT(11),JJ(11)
      DIMENSION DATA(1)
C
      COMPLEX POLES,ZEROES
      COMPLEX RPOLES,RZEROS
      REAL*4 LAT,LON
      INTEGER*4 CODE,CHN,SYEAR,SDOFY,SHOUR,SMIN
C
      DATA NY/'y'/
C
      ENCODE(4,'(4A1)',KCODE)CODE(1),CODE(2),CODE(3),CODE(4)
      JCODE(ICHAN)=KCODE
C
      DO 100 I=1,4
      JCHN(I,ICHAN)=CHN(I)
  100 CONTINUE
C
      RLAT(ICHAN)=LAT
      RLON(ICHAN)=LON
      RELEV(ICHAN)=ELEV*0.0003048
      JYEAR(ICHAN)=SYEAR
      JDOFY(ICHAN)=SDOFY
      JHOUR(ICHAN)=SHOUR
      JMIN(ICHAN)=SMIN
      RSEC(ICHAN)=SSECS
      RSAMR(ICHAN)=RATE
      RA0(ICHAN)=A0
      JNP(ICHAN)=NP
```

```
      JNZ(ICHAN)=NZ
      JFLAG(ICHAN)=NFLAG(3)
C
      DO 200 I=1,NP
      RPOLES(I,ICHAN)=POLES(I)
  200 CONTINUE
C
      DO 300 I=1,NZ
      RZEROS(I,ICHAN)=ZEROES(I)
  300 CONTINUE
C
C     LIST HEADER INFORMATION
C
      CALL PRHEAD(ICHAN)
C
      DO 500 I=1,11
      JJ(I)=I-1
  500 CONTINUE
C
C     WRITE STATUS INFORMATION
C
      WRITE(IWR,550)
      WRITE(IWR2,550)
  550 FORMAT(/,10X,'FILE STATUS',5X,'BIT',3X,'STATUS',/)
      WRITE(IWR,600)(JJ(I),ISTAT(I),I=1,11)
      WRITE(IWR2,600)(JJ(I),ISTAT(I),I=1,11)
  600 FORMAT(26X,I2,5X,I2)
C
C     ZERO DESIRED DATA CHANNEL
C
  700 INDEX=(ICHAN-1)*LENG
C
      DO 750 J=1,LENG
      IN=INDEX+J
      DATA(IN)=0.0
  750 CONTINUE
C
C     READ DESIRED START TIME
C
      WRITE(IWR2,800)
  800 FORMAT(/,'++ INPUT: ','DESIRED CHANNEL START TIME',/,9X,'DAY',
     * 3X,'HOUR',3X,'MINUTE',3X,'SECONDS',/)
      READ(IRE,*,END=9000)IBDY,IBHR,IBMIN,BSEC
C
C     IBDY - DESIRED START TIME - DAY
C            = 0 ASSUME SAME BEGIN TIME AS FILE
C     IBHR - DESIRED START TIME - HOUR
C     IBMIN - DESIRED START TIME - MINUTE
C     BSEC - DESIRED START TIME - SECOND
C
      IBF=1
      IF(IBDY.NE.0)GO TO 850
  820 IBF=0
      IBDY=JDOFY(ICHAN)
      IBHR=JHOUR(ICHAN)
      IBMIN=JMIN(ICHAN)
      BSEC=RSEC(ICHAN)
      GO TO 950
```

```
C
  850 IBH=IBHR
      IF(IBDY.GT.SDOFY)IBH=IBH+24
      IF(IBDY.LT.SDOFY)GO TO 820
C
      CALL TIMC(DBT,IBH,IBMIN,BSEC)
C
  950 CALL TIMC(ABT,JHOUR(ICHAN),JMIN(ICHAN),RSEC(ICHAN))
C
      IF(IBF.NE.0)GO TO 1200
 1000 INIT=0
      JSAMP(ICHAN)=NSAMP
      IF(NSAMP.GT.LENG)JSAMP(ICHAN)=LENG
      IFP=1
      ILP=JSAMP(ICHAN)
      GO TO 1500
C
 1200 TDIFF=DBT-ABT
      IF(TDIFF.LT.0.0)GO TO 820
C
      JDOFY(ICHAN)=IBDY
      JHOUR(ICHAN)=IBHR
      JMIN(ICHAN)=IBMIN
      RSEC(ICHAN)=BSEC
C
      IF(TDIFF.EQ.0.0)GO TO 1000
C
C     ACTUAL BEGIN TIME BEFORE DESIRED BEGIN TIME
C
      INIT=TDIFF*RSAMR(ICHAN)+0.5
      IFP=INIT+1
      IF(IFP.GT.NSAMP)GO TO 7000
      ILP=NSAMP
      JSAMP(ICHAN)=NSAMP-INIT
      IF(JSAMP(ICHAN).LE.LENG)GO TO 1500
      ILP=IFP+LENG-1
      JSAMP(ICHAN)=LENG
C
C     DETERMINE END TIME
C
 1500 TOTS=JSAMP(ICHAN)-1
      TOTS=TOTS/RSAMR(ICHAN)
      IF(IBF.EQ.0)DBT=ABT
      ESEC=DBT+TOTS
C
      CALL ITIMC(ESEC,IEH,IEMIN,ESEC)
C
      IEDY=IBDY
      IF(IEH.LT.24)GO TO 1550
      IEH=IEH-24
      IEDY=IEDY+1
C
C     MOVE DATA IN DATA ARRAY
C
 1550 INDEX=(ICHAN-1)*LENG
C
      DO 1600 J=IFP,ILP
      IN=INDEX+J
```

```
      DATA(IN)=BX(J)
 1600 CONTINUE
C
      WRITE(IWR,1630)ICHAN
      WRITE(IWR2,1630)ICHAN
 1630 FORMAT(/,10X,'DATA BEGIN TIME FOR CHANNEL ',I4,' IS')
      WRITE(IWR,1640)IBDY,IBHR,IBMIN,BSEC
      WRITE(IWR2,1640)IBDY,IBHR,IBMIN,BSEC
 1640 FORMAT(/,10X,'DAY',I4,3X,'HOUR',I3,3X,'MINUTE',I3,3X,'SECONDS',
     * F6.2,/)
C
      WRITE(IWR,1650)ICHAN
      WRITE(IWR2,1650)ICHAN
 1650 FORMAT(/,10X,'DATA END TIME FOR CHANNEL ',I4,' IS')
      WRITE(IWR,1640)IEDY,IEH,IEMIN,ESEC
      WRITE(IWR2,1640)IEDY,IEH,IEMIN,ESEC
C
      WRITE(IWR,1660)JSAMP(ICHAN),TOTS
      WRITE(IWR2,1660)JSAMP(ICHAN),TOTS
 1660 FORMAT(/,10X,'NUMBER OF DATA SAMPLES IS ',I6,/,
     * 10X,'NUMBER OF SECONDS OF DATA IS ',F8.2,/)
C
C     WRITE CALIBRATION
C
      WRITE(IWR,1700)
      WRITE(IWR2,1700)
 1700 FORMAT(/,10X,'RESPONSE POLES',/)
      WRITE(IWR,1800)(RPOLES(I,ICHAN),I=1,NP)
      WRITE(IWR2,1800)(RPOLES(I,ICHAN),I=1,NP)
 1800 FORMAT(10X,E10.3,3X,E10.3,6X,E10.3,3X,E10.3)
      WRITE(IWR,1900)
      WRITE(IWR2,1900)
 1900 FORMAT(/,10X,'RESPONSE ZEROS',/)
      WRITE(IWR,1800)(RZEROS(I,ICHAN),I=1,NZ)
      WRITE(IWR2,1800)(RZEROS(I,ICHAN),I=1,NZ)
      WRITE(IWR,1950)A0
      WRITE(IWR2,1950)A0
 1950 FORMAT(/,10X,'A0 = ',E10.3,/)
C
      IF(A0.GE.0.0)GO TO 9000
      WRITE(IWR2,2000)
 2000 FORMAT(/,10X,'A0 IS NEGATIVE, REVERSE DATA?',//,
     * '++ INPUT: ','(y/n)',/)
      READ(IRE,2100,END=9000)KY
C
C     KY - y OR n
C
 2100 FORMAT(A1)
      IF(NY.NE.KY)GO TO 9000
C
C     REVERSE POLARITY OF DATA
C
      INDEX=(ICHAN-1)*LENG
C
      DO 2500 J=1,JSAMP(ICHAN)
      IN=INDEX+J
      DATA(IN)=-DATA(IN)
 2500 CONTINUE
```

```
C
      A0=-A0
      RA0(ICHAN)=-RA0(ICHAN)
C
      WRITE(IWR,3000)
      WRITE(IWR2,3000)
 3000 FORMAT(/,10X,'DATA POLARITY HAS BEEN SUCCESSFULLY REVERSED',/)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** DESIRED BEGIN TIME AFTER ALL DATA *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT READ',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE FILTT(X,ISAMP,D,G)
C
C*********************************************************************
C
C     APPLY BUTTERWORTH FILTER COEFFICIENTS TO BANDPASS FILTER
C     A TIME SERIES
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN THE TIME SERIES - INPUT
C     D - RECURSIVE FILTER COEFFICIENTS - INPUT
C     G - GAIN OF FILTER - INPUT
C
C     NOVEMBER 17, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1),D(1),XC(3),XD(3),XE(3)
C
C     APPLY FILTER IN THE FORWARD DIRECTION
C
      N=ISAMP
      XM2=X(1)
      XM1=X(2)
      XM=X(3)
      XC(1)=XM2
      XC(2)=XM1-D(1)*XC(1)
      XC(3)=XM-XM2-D(1)*XC(2)-D(2)*XC(1)
      XD(1)=XC(1)
      XD(2)=XC(2)-D(3)*XD(1)
      XD(3)=XC(3)-XC(1)-D(3)*XD(2)-D(4)*XD(1)
      XE(1)=XD(1)
      XE(2)=XD(2)-D(5)*XE(1)
      XE(3)=XD(3)-XD(1)-D(5)*XE(2)-D(6)*XE(1)
      X(1)=XE(1)
      X(2)=XE(2)-D(7)*X(1)
      X(3)=XE(3)-XE(1)-D(7)*X(2)-D(8)*X(1)
C
      DO 1000 I=4,N
      XM2=XM1
      XM1=XM
      XM=X(I)
      K=I-((I-1)/3)*3
      GO TO (100,200,300),K
C
  100 M=1
      M1=3
      M2=2
      GO TO 400
C
  200 M=2
      M1=1
      M2=3
      GO TO 400
C
  300 M=3
      M1=2
      M2=1
C
```

```
  400 XC(M)=XM-XM2-D(1)*XC(M1)-D(2)*XC(M2)
      XD(M)=XC(M)-XC(M2)-D(3)*XD(M1)-D(4)*XD(M2)
      XE(M)=XD(M)-XD(M2)-D(5)*XE(M1)-D(6)*XE(M2)
      X(I)=XE(M)-XE(M2)-D(7)*X(I-1)-D(8)*X(I-2)
C
 1000 CONTINUE
C
C     APPLY FILTER IN THE REVERSE DIRECTION
C
      XM2=X(N)
      XM1=X(N-1)
      XM=X(N-2)
      XC(1)=XM2
      XC(2)=XM1-D(1)*XC(1)
      XC(3)=XM-XM2-D(1)*XC(2)-D(2)*XC(1)
      XD(1)=XC(1)
      XD(2)=XC(2)-D(3)*XD(1)
      XD(3)=XC(3)-XC(1)-D(3)*XD(2)-D(4)*XD(1)
      XE(1)=XD(1)
      XE(2)=XD(2)-D(5)*XE(1)
      XE(3)=XD(3)-XD(1)-D(5)*XE(2)-D(6)*XE(1)
      X(N)=XE(1)
      X(N-1)=XE(2)-D(7)*X(1)
      X(N-2)=XE(3)-XE(1)-D(7)*X(2)-D(8)*X(1)
C
      DO 2000 I=4,N
      XM2=XM1
      XM1=XM
      J=N-I+1
      XM=X(J)
      K=I-((I-1)/3)*3
      GO TO (1100,1200,1300),K
C
 1100 M=1
      M1=3
      M2=2
      GO TO 1400
C
 1200 M=2
      M1=1
      M2=3
      GO TO 1400
C
 1300 M=3
      M1=2
      M2=1
C
 1400 XC(M)=XM-XM2-D(1)*XC(M1)-D(2)*XC(M2)
      XD(M)=XC(M)-XC(M2)-D(3)*XD(M1)-D(4)*XD(M2)
      XE(M)=XD(M)-XD(M2)-D(5)*XE(M1)-D(6)*XE(M2)
      X(J)=XE(M)-XE(M2)-D(7)*X(J+1)-D(8)*X(J+2)
C
 2000 CONTINUE
C
      DO 2500 I=1,N
      X(I)=X(I)/G
 2500 CONTINUE
C
```

```
RETURN
END
```

```
      SUBROUTINE FILTX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR FILTERING DATA
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 16, 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),D(8)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','IFL',/)
      READ(IRE,*,END=9000)IFL
C
C     IFL - FILTER FLAG
C           = 0 USE BUTTERWORTH BANDPASS FILTER IN TIME DOMAIN
C           = 1 DESIGN BANDPASS FILTER IN FREQUENCY DOMAIN
C
      WRITE(IWR,200)IFL
      WRITE(IWR2,200)IFL
  200 FORMAT(/,10X,'IFL = ',I2,/)
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL TO FILTER
C             = 0 FILTER ALL CHANNELS
C
      WRITE(IWR,400)ICHAN
      WRITE(IWR2,400)ICHAN
  400 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','FLOW',3X,'FHIGH',/)
      READ(IRE,*,END=9000)FLOW,FHIGH
C
C     FLOW - LOW FREQUENCY CUTOFF  (6 DB DOWN)
C     FHIGH - HIGH FREQUENCY CUTOFF  (6 DB DOWN)
```

```
C              (MUST BE LESS THAN SAMR/2)
C              > OR = SAMR/2 THEN 0.9*SAMR/2 USED
C
       WRITE(IWR,600)FLOW,FHIGH
       WRITE(IWR2,600)FLOW,FHIGH
   600 FORMAT(/,10X,'FLOW = ',F10.4,5X,'FHIGH = ',F10.5,/)
       IF(FLOW.GE.FHIGH)GO TO 7000
C
       IF(IFL.LE.0)GO TO 700
C
       WRITE(IWR2,650)
   650 FORMAT(/,'++ INPUT: ','FLBGIN',3X,'FHBGIN',/)
       READ(IRE,*,END=9000)FLBGIN,FHBGIN
C
C      FLBGIN - LOW FREQUENCY BEGIN TAPER (> FLOW)
C      FHBGIN - HIGH FREQUENCY BEGIN TAPER (< FHIGH)
C
       WRITE(IWR,670)FLBGIN,FHBGIN
       WRITE(IWR2,670)FLBGIN,FHBGIN
   670 FORMAT(/,10X,'FLBGIN = ',F10.4,5X,'FHBGIN = ',F10.5,/)
C
   700 IF(ICHAN.LE.0)GO TO 750
       NCH1=ICHAN
       NCH2=ICHAN
       GO TO 800
C
   750 NCH1=1
       NCH2=NACHLS
C
   800 DO 2000 I=NCH1,NCH2
       INDEX=(I-1)*LENG+1
       SAMR=RSAMR(I)
       ISAMP=JSAMP(I)
       FHIX=FHIGH
       IF(FHIX.GE.(SAMR/2.0))FHIX=0.9*SAMR/2.0
       IF(IFL.LE.0)GO TO 900
       IF(FHIX.LE.FHBGIN)FHBGIN=0.8*FHIX
       GO TO 1500
C
C      OBTAIN BUTTERWORTH FILTER COEFFICIENTS AND GAIN FACTOR
C      FOR A RECURSIVE BANDPASS FILTER
C
   900 CALL BNDPAS(FLOW,FHIX,SAMR,D,G)
C
C      APPLY BUTTERWORTH FILTER IN TIME DOMAIN
C
  1000 CALL FILTT(DATA(INDEX),ISAMP,D,G)
C
       GO TO 2000
C
C      DESIGN FREQUENCY DOMAIN FILTER AND CONVOLVE WITH DATA
C      IN TIME DOMAIN
C
  1500 IF(ISAMP.GT.16400)GO TO 7300
C
       CALL PHILTR(DATA(INDEX),ISAMP,SAMR,FLOW,FLBGIN,FHBGIN,FHIX,
      * IERR)
C
```

```
      IF(IERR.EQ.1)GO TO 7500
      IF(IERR.EQ.2)GO TO 7400
 2000 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 FILTERED',/)
      GO TO 9000
C
 7300 WRITE(IWR2,7350)
 7350 FORMAT(/,10X,'TOO MANY SAMPLES TO FILTER',/)
      GO TO 7000
C
 7400 WRITE(IWR2,7450)FLOW
 7450 FORMAT(/,10X,'FLOW TOO SMALL FOR FILTER ROUTINE;',/,10X,
     * 'SMALLEST POSSIBLE VALUE IS ',E12.4,/)
      GO TO 7000
C
 7500 WRITE(IWR2,7350)
 7550 FORMAT(/,10X,'TOO FEW SAMPLES TO FILTER',/)
      GO TO 7000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY FILTERED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE FOURT(X,N,SAMR,A,B,C,PH)
C
C*********************************************************************
C
C     SLOW FFT ROUTINE USED FOR CERTAIN TESTS
C
C     X - TIME SERIES - INPUT
C     N - NUMBER OF SAMPLES - INPUT
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - 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
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 13, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION A(1),B(1),C(1),PH(1),X(1)
C
      FN=N
      NN=N/2+1
      T=FN/SAMR
C
C     INITIALIZE FOURIER PARAMETERS
C
      DO 100 I=1,NN
      A(I)=0.0
      B(I)=0.0
      C(I)=0.0
      PH(I)=0.0
  100 CONTINUE
C
      DO 300 I=1,NN
      FI=I-1
      ARG=TWOPI*FI/(SAMR*T)
      XARG=0.0
C
      DO 200 J=1,N
      CC=X(J)*COS(XARG)
      SS=X(J)*SIN(XARG)
      A(I)=A(I)+CC
      B(I)=B(I)+SS
      XARG=XARG+ARG
  200 CONTINUE
C
C     SCALE COEFFICIENTS
C
      A(I)=A(I)/FN
      IF(I.EQ.1)GO TO 300
      A(I)=2.0*A(I)
      B(I)=2.0*B(I)/FN
C
```

```
C       COMPUTE AMPLITUDE
C
        C(I)=A(I)*A(I)+B(I)*B(I)
        C(I)=SQRT(C(I))
C
C       COMPUTE PHASE ANGLE
C
        PH(I)=PHASOR(B(I),A(I))
C
  300 CONTINUE
C
        C(1)=0.0
C
        RETURN
        END
```

```
      FUNCTION FUNC(I,X,Z)
C
C*************************************************************************
C
C     EVALUATE ONE OF SEVEN SPECIAL FUNCTIONS FOR TRAVEL TIMES
C
C     I - INDEX TO SPECIFY FUNCTION - INPUT
C     X - ARGUMENT FOR FUNCTION - INPUT
C     Z - ARGUMENT FOR FUNCTION - INPUT
C     FUNC - FUNCTION VALUE - OUTPUT
C
C     WRITTEN BY GEOTECH ON CONTRACT TO U.S. GOVERNMENT
C
C     JANUARY 3, 1968  DALLAS
C
C*************************************************************************
C
      FUNC=0.0
C
      GO TO (10,20,30,40,50,60,70),I
C
   10 FUNC=1.0
      GO TO 9000
C
   20 FUNC=X
      GO TO 9000
C
   30 FUNC=Z
      GO TO 9000
C
   40 FUNC=X*X
      GO TO 9000
C
   50 FUNC=Z*X
      GO TO 9000
C
   60 FUNC=Z*Z
      GO TO 9000
C
   70 FUNC=X*X*X
C
 9000 RETURN
      END
```

```
      SUBROUTINE GPVEL(KLAST,T,C,U,DT)
C
C*********************************************************************
C
C     CALCULATE GROUP VELOCITY BY NUMERICAL DIFFERENTIATION
C
C     KLAST - NUMBER OF ROOTS FOUND - INPUT
C     T - PERIOD (IN SECONDS) - INPUT
C     C - PHASE VELOCITY (IN KM/S) - INPUT
C     U - GROUP VELOCITY (IN KM/S) - OUTPUT
C     DT - INCREMENT IN T VALUE (IN SECONDS) - INPUT
C
C     FEBRUARY 3, 1994  COLORADO
C
C*********************************************************************
C
      DIMENSION T(1),C(1),U(1)
C
      IF(KLAST.GT.2)GO TO 200
C
      U(1)=0.0
      U(2)=0.0
      GO TO 9000
C
  200 U(1)=0.0
      U(KLAST)=0.0
      M=KLAST-1
C
      DO 500 K=2,M
      DCDT=0.5*(C(K+1)-C(K-1))/DT
      U(K)=C(K)*(1.0/(1.0+T(K)*DCDT/C(K)))
  500 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE HELIP(FLAT,H)
C
C*********************************************************************
C
C     CALCULATES PERPENDICULAR DISTANCE ABOVE A REFERENCE SPHERE
C     FOR AN ELLIPICITY CORRECTION TO THE TRAVEL TIME
C
C     FLAT - GEOCENTRIC CO-LATITUDE OF POINT (IN DEGREES) - INPUT
C     H - PERPENDICULAR DISTANCE ABOVE REFERENCE SPHERE TO POINT
C         (IN KILOMETERS) - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 30, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      H=-10.663133*COS(2.0*FLAT*DG2RD)-3.554378
C
      RETURN
      END
```

```
      SUBROUTINE INITFL
C
C*************************************************************************
C
C     INITIALIZE FLAGS USED BY IRDHD AND IRDDAT. INITFL MUST
C     BE CALLED BEFORE ANY REFERENCE TO IRDHD OR IRDDAT
C
C     WRITTEN BY MADELEINE ZIRBES
C
C     SEPTEMBER 15, 1980  COLORADO
C
C*************************************************************************
C
      COMMON/FLAGS/HFLG,DFLG,NPTS
      COMMON/TRDATA/PTS,BUFF
C
      DIMENSION BUFF(500)
C
      INTEGER*4 HFLG,DFLG,PTS
C
      HFLG=1
      DFLG=0
      NPTS=1
      PTS=0
C
      RETURN
      END
```

```
      SUBROUTINE INSTR(A,FMN,FMX,DF,NH2,IFLG,ICHAN,IRX)
C
C*********************************************************************
C
C     REMOVE (OR ADD) INSTRUMENT RESPONSE FROM (OR TO) TIME SERIES
C
C     A - FOURIER TRANSFORM OF TIME SERIES - INPUT AND OUTPUT
C     FMN - MINIMUM FREQUENCY = SAMR/N2LEN - INPUT
C     FMX - MAXIMUM FREQUENCY = SAMR/2 - INPUT
C     DF - FREQUENCY INCREMENT = SAMR/N2LEN - INPUT
C     NH2 - NUMBER OF REAL COMPONENTS = N2LEN/2+1 - INPUT
C     IFLG - RESPONSE OUTPUT FLAG - INPUT
C            = 0 SET TO -1
C            =-1 REMOVE DISPLACEMENT RESPONSE
C            =-2 REMOVE VELOCITY RESPONSE
C            =-3 REMOVE ACCELERATION RESPONSE
C            = 1 ADD DISPLACEMENT RESPONSE
C            = 2 ADD VELOCITY RESPONSE
C            = 3 ADD ACCELERATION RESPONSE
C     ICHAN - CHANNEL TO CORRECT FOR INSTRUMENT RESPONSE - INPUT
C     IRX - RESPONSE FLAG - INPUT
C           = 0 USE RESPONSE IN CHANNEL HEADER
C           = 1 USE RESPONSE FROM MASTER FILE
C
C     OCTOBER 18, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION A(1)
C
      COMPLEX BB,B,RESP,RESP2
C
      WMN=TWOPI*FMN
      WMX=TWOPI*FMX
C
      IF(IFLG.EQ.0)IFLG=-1
      IFL=IFLG
      IF(IFL.LT.0)IFL=-IFL
C
      DW=TWOPI*DF
      W=WMN-DW
C
      DO 1000 I=2,NH2
      W=W+DW
      K=I+NH2
      B=CMPLX(A(I),A(K))
C
      IF(IRX.GT.0)GO TO 200
C
      BB=RESP(ICHAN,W,IFL)
C
      GO TO 300
C
  200 BB=RESP2(W,IFL)
C
```

```
  300 IF(IFLG.LT.0)GO TO 400
      B=B*BB
      GO TO 800
  400 B=B/BB
C
  800 A(I)=REAL(B)
      A(K)=AIMAG(B)
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE INSTX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR REMOVING OR ADDING INSTRUMENT RESPONSE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 8, 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
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL TO CORRECT FOR INSTRUMENT RESPONSE
C             = 0 CORRECT ALL DATA CHANNELS
C
      WRITE(IWR,400)ICHAN
      WRITE(IWR2,400)ICHAN
  400 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','IFL',3X,'IRX',3X,'IMT',/)
      READ(IRE,*,END=9000)IFL,IRX,IMT
C
C     IFL - RESPONSE OUTPUT FLAG
C           = 0 SET TO -1
C           =-1 REMOVE DISPLACEMENT RESPONSE
C           =-2 REMOVE VELOCITY RESPONSE
C           =-3 REMOVE ACCELERATION RESPONSE
C           = 1 ADD DISPLACEMENT RESPONSE
C           = 2 ADD VELOCITY RESPONSE
C           = 3 ADD ACCELERATION RESPONSE
C     IRX - RESPONSE FLAG
C           = 0 USE RESPONSE IN CHANNEL HEADER
C           = 1 USE RESPONSE FROM MASTER FILE
C     IMT - MEAN/TREND FLAG
C           = 0 DO NOT REMOVE MEAN AND TAPER DATA
C           = 1 REMOVE MEAN AND TAPER DATA
C
```

```
      IF(IFL.EQ.0)IFL=-1
      WRITE(IWR,1000)IFL,IRX,IMT
      WRITE(IWR2,1000)IFL,IRX,IMT
 1000 FORMAT(/,10X,'IFL = ',I3,5X,'IRX = ',I2,5X,'IMT = ',I2,/)
C
      IF(IRX.LE.0)GO TO 1300
C
C     SET UP DESIRED RESPONSE IN MASTER FILE
C
      CALL SETRES
C
 1300 IF(ICHAN.LE.0)GO TO 1500
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 1700
C
 1500 NCH1=1
      NCH2=NACHLS
C
 1700 DO 5000 I=NCH1,NCH2
      II=I
      INDEX=(I-1)*LENG+1
      ISAMP=JSAMP(I)
      SAMR=RSAMR(I)
C
      IF(ISAMP.LT.4)GO TO 7000
      IF(SAMR.LE.0.0)GO TO 7000
C
      IF(IMT.LE.0)GO TO 1800
C
C     REMOVE MEAN
C
      IDEG=0
C
      CALL METR(DATA(INDEX),ISAMP,IDEG)
C
C     TAPER SIGNAL
C
      PERCNT=2.0
C
      CALL TAP(DATA(INDEX),ISAMP,PERCNT)
C
 1800 ISIGN=1
C
C     TRANSFORM TIME SERIES TO FREQUENCY DOMAIN
C
      CALL TTOF1(DATA(INDEX),ISAMP,LENG,ISIGN,N2PWR,N2LEN,NH2,N2P1,NZ)
C
      FMN=SAMR/N2LEN
      FMX=SAMR/2.0
      DF=FMN
C
C     MAKE INSTRUMENT CORRECTION IN THE FREQUENCY DOMAIN
C
      CALL INSTR(DATA(INDEX),FMN,FMX,DF,NH2,IFL,II,IRX)
C
      ISIGN=-1
C
```

```
C     TRANSFORM FREQUENCY DOMAIN CORRECTED SIGNAL BACK TO THE
C     TIME DOMAIN
C
      CALL TTOF1(DATA(INDEX),ISAMP,LENG,ISIGN,N2PWR,N2LEN,NH2,N2P1,NZ)
C
      JSAMP(I)=N2LEN
      IF(IABS(IFL).NE.1)GO TO 2500
      IF(IFL.LT.0)GO TO 2200
      WRITE(IWR,2100)I
      WRITE(IWR2,2100)I
 2100 FORMAT(/,10X,'DISPLACEMENT RESPONSE ADDED TO CHANNEL ',I4,/)
      GO TO 5000
C
 2200 WRITE(IWR,2300)I
      WRITE(IWR2,2300)I
 2300 FORMAT(/,10X,'DISPLACEMENT RESPONSE REMOVED FROM CHANNEL ',I4,/)
      GO TO 5000
C
 2500 IF(IABS(IFL).NE.2)GO TO 3000
      IF(IFL.LT.0)GO TO 2700
      WRITE(IWR,2600)I
      WRITE(IWR2,2600)I
 2600 FORMAT(/,10X,'VELOCITY RESPONSE ADDED TO CHANNEL ',I4,/)
      GO TO 5000
C
 2700 WRITE(IWR,2800)I
      WRITE(IWR2,2800)I
 2800 FORMAT(/,10X,'VELOCITY RESPONSE REMOVED FROM CHANNEL ',I4,/)
      GO TO 5000
C
 3000 IF(IFL.LT.0)GO TO 3200
      WRITE(IWR,3100)I
      WRITE(IWR2,3100)I
 3100 FORMAT(/,10X,'ACCELERATION RESPONSE ADDED TO CHANNEL ',I4,/)
      GO TO 5000
C
 3200 WRITE(IWR,3300)I
      WRITE(IWR2,3300)I
 3300 FORMAT(/,10X,'ACCELERATION RESPONSE REMOVED FROM CHANNEL ',I4,/)
C
 5000 CONTINUE
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'DATA NOT CORRECTED FOR INSTRUMENT RESPONSE',/)
C
 9000 RETURN
      END
```

```
      INTEGER FUNCTION IRDDAT(LUN,VALU)
C
C**********************************************************************
C
C     SEQUENTIALLY READ THE NEXT DATA POINT OF A DATA RECORD ON
C     EACH CALL, RETURNING IT IN VALU.
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     NPTS - INDEX NUMBER OF NEXT POINT TO READ - INPUT AND OUTPUT
C     VALU - DATA SAMPLE OBTAINED - OUTPUT
C     IRDDAT - READ FLAG - OUTPUT
C              = 0 END OF DATA
C              = 1 SUCCESSFUL DATA READ
C              =-1 ERROR OR END OF FILE
C
C     WRITTEN BY MADELEINE ZIRBES
C
C     SEPTEMBER 15, 1980  COLORADO
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     AUGUST 9, 1993  COLORADO
C
C**********************************************************************
C
      COMMON/FLAGS/HFLG,DFLG,NPTS
      COMMON/TRDATA/PTS,BUFF
C
      DIMENSION BUFF(500)
C
      INTEGER*4 HFLG,DFLG,PTS,TD
C
      VALU=0.0
C
C     CHECK IF DESIRED DATA ALREADY READ AND IS IN BUFF ARRAY
C
      IF(NPTS.LE.PTS)GO TO 40
C
C     CHECK IF NEXT RECORD IS A DATA RECORD
C
      IF(DFLG.NE.1)GO TO 20
C
      CALL RDDATA(LUN,TD)
C
C     CHECK IF END OF FILE READ
C
      IF(TD.LT.0)GO TO 25
C
C     CHECK IF END OF DATA READ
C
      IF(.NOT.((PTS.EQ.1).AND.(BUFF(1).GT.1000000000.0)))GO TO 10
C
C     END OF DATA
C
      HFLG=1
      DFLG=0
      NPTS=1
```

```
      PTS=0
      IRDDAT=0
      GO TO 9000
C
C     SUCCESSFUL READ
C
   10 VALU=BUFF(1)
      NPTS=2
      IRDDAT=1
      GO TO 9000
C
   20 IF(HFLG.NE.0)GO TO 30
C
C     END OF FILE
C
   25 IRDDAT=-1
      GO TO 9000
C
C     NEXT RECORD IS NOT A DATA RECORD - ERROR
C
   30 IRDDAT=-1
      GO TO 9000
C
C     DESIRED DATA ALREADY READ INTO BUFF ARRAY
C     SUCCESSFUL READ
C
   40 VALU=BUFF(NPTS)
      NPTS=NPTS+1
      IRDDAT=1
C
 9000 RETURN
      END
```

```
      INTEGER FUNCTION IRDHD(LUN)
C
C**********************************************************************
C
C     READS THE NEXT HEADER RECORD FROM LOGICAL UNIT LUN.
C     IT WILL ALWAYS TRY TO RETURN THE NEXT HEADER RECORD,
C     READING OVER INTERMEDIATE DATA RECORDS IF NECESSARY
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     IRDHD - READ STATUS - OUTPUT
C             = 0 END ENCOUNTERED
C             = 1 SUCCESSFUL READ
C             =-1 END OF FILE READ
C
C     WRITTEN BY MADELEINE ZIRBES
C
C     SEPTEMBER 15, 1980  COLORADO
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     AUGUST 9, 1993  COLORADO
C
C**********************************************************************
C
      COMMON/FLAGS/HFLG,DFLG,NPTS
      COMMON/HEAD/CODE,CHN,LAT,LON,ELEV,SYEAR,SDOFY,SHOUR,SMIN,
     * SSECS,RATE,A0,NP,POLES,NZ,ZEROES,NFLAG
C
      DIMENSION CODE(4),CHN(4),POLES(30),ZEROES(20),NFLAG(3)
C
      COMPLEX POLES,ZEROES
      REAL*4 LAT,LON
      INTEGER*4 CODE,CHN,SYEAR,SDOFY,SHOUR,SMIN,TF,HFLG,DFLG
C
C     CHECK IF NEXT RECORD IS A DATA RECORD
C
      IF(DFLG.NE.1)GO TO 30
C
C     READ RECORD UNTIL END OF DATA IS FOUND
C
   10 IDUM=IRDDAT(LUN,VALU)
      IF(IDUM.EQ.1)GO TO 10
      IF(IDUM.LT.0)GO TO 50
C
C     READ A HEADER RECORD
C
   30 CALL RDHEAD(LUN,TF)
C
C     CHECK FOR END OR END OF FILE
C
      IF(TF.EQ.0)GO TO 40
      IF(TF.LT.0)GO TO 50
C
C     END ENCOUNTERED
C
      HFLG=0
      DFLG=0
```

```
      IRDHD=0
      GO TO 9000
C
C     SUCCESSFULLY READ A HEADER RECORD
C
   40 HFLG=0
      DFLG=1
      IRDHD=1
      GO TO 9000
C
C     END OF FILE READ
C
   50 HFLG=0
      DFLG=0
      IRDHD=-1
C
 9000 RETURN
      END
```

```
      INTEGER FUNCTION ISTBIT(LUN,ISTAT)
C
C*********************************************************************
C
C     RETURNS THE STATUS BITS ASSOCIATED WITH THE DATA. IT WILL
C     AUTOMATICALLY SKIP TO THE END OF THE DATA RECORDS, IF NOT
C     POSITIONED THERE ALREADY.
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     ISTAT - ARRAY OF STATUS BITS - OUTPUT
C     ISTBIT - READ FLAG - OUTPUT
C              = 1 OBTAINED STATUS BITS
C              =-1 NOT ABLE TO OBTAIN STATUS BITS
C
C      WRITTEN BY MADELEINE ZIRBES
C
C      MAY 23, 1983  COLORADO
C
C*********************************************************************
C
      COMMON/FLAGS/HFLG,DFLG,NPTS
      COMMON/TRDATA/PTS,BUFF
C
      DIMENSION BUFF(500),ISTAT(11)
C
      INTEGER*4 HFLG,DFLG,PTS
C
C     CHECK IF AT END OF DATA
C
      IF(.NOT.(HFLG.EQ.1.AND.DFLG.EQ.0))GO TO 20
C
   10 CALL STABIT(BUFF(1),ISTAT)
C
      ISTBIT=1
      GO TO 9000
C
C     CHECK IF THIS IS A DATA RECORD
C
   20 IF(DFLG.NE.1)GO TO 50
C
C     READ RECORDS UNTIL END-OF-DATA
C
   30 IDUM=IRDDAT(LUN,VALU)
C
      IF(IDUM.LT.0)GO TO 50
      IF(IDUM.EQ.1)GO TO 30
      GO TO 10
C
C     NOT ABLE TO OBTAIN STATUS BITS
C
   50 ISTBIT=-1
C
 9000 RETURN
      END
```

```
      SUBROUTINE ITIMC(TSEC,IHR,MIN,SEC)
C
C*********************************************************************
C
C     DECODE TOTAL TIME FROM SECONDS
C
C     TSEC - TOTAL TIME IN SECONDS - INPUT
C     IHR - HOURS - OUTPUT
C     MIN - MINUTES - OUTPUT
C     SEC - SECONDS - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 19, 1993  COLORADO
C
C*********************************************************************
C
      DATA IT1/3600/,IT2/60/
C
      IT=TSEC
      IHR=IT/IT1
      IT=IT-IHR*IT1
      MIN=IT/IT2
      SEC=TSEC-IHR*IT1-MIN*IT2
C
      RETURN
      END
```

```
      SUBROUTINE ITYR(IYR,IC,IFL)
C
C*****************************************************************************
C
C     TEST YEAR DIVISIBILITY
C
C     IYR - YEAR - INPUT
C     IC - TEST DIVISOR - INPUT
C     IFL - YEAR FLAG - OUTPUT
C           = 0 NOT EXACT DIVISOR
C           = 1 EXACT DIVISOR
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 24, 1993  COLORADO
C
C*****************************************************************************
C
      IFL=0
      IT=IYR/IC
      ITEST=IYR-IT*IC
      IF(ITEST.EQ.0)IFL=1
C
      RETURN
      END
```