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

KEY ANALYSIS SYSTEM

Volume 3 - Subroutines

Robert P. Massé

March 29, 1994

Version 1.0

Open-File Report 94-185

National Earthquake Information Center
Golden, Colorado

1994

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

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

This project is funded by the Nuclear Regulatory Commission.

```
      SUBROUTINE JUL(IYR,IMO,IDY,IDOY,LEAP)
C
C**********************************************************************
C
C     DETERMINE JULIAN DAY
C
C     IYR - YEAR - INPUT
C     IMO - MONTH (NUMBER) - INPUT
C     IDY - DAY - INPUT
C     IDOY - JULIAN DAY - OUTPUT
C     LEAP - LEAP YEAR FLAG - OUTPUT
C            = 0 NOT LEAP YEAR
C            = 1 LEAP YEAR
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 3, 1993  COLORADO
C
C**********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION IB(11)
C
      DATA IB/31,28,31,30,31,30,31,31,30,31,30/
C
      IF(IYR.LE.0)GO TO 7000
      LEAP=0
      IC=100
C
      CALL ITYR(IYR,IC,IFL)
C
      IF(IFL.GT.0)GO TO 2000
      IC=4
C
      CALL ITYR(IYR,IC,IFL)
C
      IF(IFL.EQ.0)GO TO 5000
      GO TO 3000
C
 2000 IC=400
C
      CALL ITYR(IYR,IC,IFL)
C
      IF(IFL.EQ.0)GO TO 5000
 3000 LEAP=1
C
 5000 IF(IMO.LE.0)GO TO 7000
      IF(IMO.GT.12)GO TO 7000
      IF(IDY.LE.0)GO TO 7000
      IF(IDY.GT.366)GO TO 7000
C
      IM=IMO-1
      IDOY=0
C
      K=1
 6000 IF(IM.LE.0)GO TO 6100
      IDOY=IDOY+IB(K)
```

```
      K=K+1
      IM=IM-1
      GO TO 6000
C
 6100 IDOY=IDOY+IDY
      IF(LEAP.EQ.0)GO TO 9000
      IF(IMO.LE.2)GO TO 9000
      IDOY=IDOY+1
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN DATE INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE JULDY
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR DETERMINING THE JULIAN DAY
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 26, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','IYR',3X,'IMO',3X,'IDY',/)
      READ(IRE,*,END=9000)IYR,IMO,IDY
C
C     IYR - YEAR
C     IMO - MONTH (NUMBER)
C     IDY - DAY
C
      WRITE(IWR,300)
      WRITE(IWR2,300)
  300 FORMAT(/,10X,'JULIAN DAY PARAMETERS',/)
      WRITE(IWR,400)IYR,IMO,IDY
      WRITE(IWR2,400)IYR,IMO,IDY
  400 FORMAT(/,10X,'IYR = ',I4,5X,'IMO = ',I2,5X,'IDY = ',I3,/)
C
      CALL JUL(IYR,IMO,IDY,IDOY,LEAP)
C
      WRITE(IWR,500)IYR,IMO,IDY,IDOY
      WRITE(IWR2,500)IYR,IMO,IDY,IDOY
  500 FORMAT(/,10X,'THE DATE:  YEAR ',I4,3X,'MONTH ',I2,3X,'DAY ',I2,
     * //,10X,'CORRESPONDS TO JULIAN DAY ',I3,/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE KEYBRD
C
C*********************************************************************
C
C     READS A DATA TRACE FROM THE KEYBOARD INTO A CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 13, 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
  100 WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','ICHAN',3X,'SAMR',/)
      READ(IRE,*,END=9000)ICHAN,SAMR
C
C     ICHAN - CHANNEL NUMBER TO STORE DATA
C             = 0 SET TO 1
C     SAMR - SAMPLING RATE (IN SAMPLES/S)
C            = 0 SET TO 40.0
C
      IF(ICHAN.LE.0)ICHAN=1
      IF(ICHAN.GT.NACHLS)NACHLS=ICHAN
      IF(SAMR.LE.0.0)SAMR=40.0
      WRITE(IWR,250)
      WRITE(IWR2,250)
  250 FORMAT(/,10X,'DATA PARAMETERS',/)
      WRITE(IWR,300)ICHAN,SAMR
      WRITE(IWR2,300)ICHAN,SAMR
  300 FORMAT(/,10X,'ICHAN = ',I4,5X,'SAMR = ',F7.3,/)
C
      IF(ICHAN.GT.NCHLS)GO TO 7000
C
      INDEX=(ICHAN-1)*LENG+1
      J=0
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','X',/)
  500 READ(IRE,*,END=1000)X
C
C     X - TIME SERIES VALUES
C         = end-of-file NO MORE DATA VALUES
C
      DATA(INDEX)=X
```

```
      INDEX=INDEX+1
      J=J+1
      GO TO 500
C
 1000 JSAMP(ICHAN)=J
      RSAMR(ICHAN)=SAMR
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE LAGRAN(X,TABLX,TABLY,Y)
C
C*********************************************************************
C
C     PERFORMS CUBIC LAGRANGIAN INTERPOLATION
C
C     X - DESIRED X VALUE - INPUT
C     TABLX - TABLE OF X VALUES - INPUT
C     TABLY - TABLE OF Y VALUES CORRESPONDING TO X VALUES - INPUT
C     Y - INTERPOLATED VALUE OF Y CORRESPONDING TO DESIRED
C         X VALUE - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 9, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION TABLX(4),TABLY(4),D(4)
C
      DO 10 I=1,4
      D(I)=X-TABLX(I)
   10 CONTINUE
C
      A=D(2)*D(3)*D(4)*TABLY(1)
      A=A/((D(1)-D(2))*(D(1)-D(3))*(D(1)-D(4)))
C
      B=D(1)*D(3)*D(4)*TABLY(2)
      B=B/((D(2)-D(1))*(D(2)-D(3))*(D(2)-D(4)))
C
      C=D(1)*D(2)*D(4)*TABLY(3)
      C=C/((D(3)-D(1))*(D(3)-D(2))*(D(3)-D(4)))
C
      E=D(1)*D(2)*D(3)*TABLY(4)
      E=E/((D(4)-D(1))*(D(4)-D(2))*(D(4)-D(3)))
C
      Y=A+B+C+E
      Y=-Y
C
      RETURN
      END
```

```
      SUBROUTINE LISTTT(LUN,NPHAS,DI,DF,DINC,DPI,DPF,DPINC)
C
C*****************************************************************
C
C     LIST TRAVEL TIMES FOR A GIVEN PHASE
C
C     LUN - OUTPUT LOGICAL UNIT NUMBER - INPUT
C     NPHAS - DESIRED PHASE NAME - INPUT
C     DI - INITIAL DISTANCE (IN DEGREES) - INPUT
C     DF - FINAL DISTANCE (IN DEGREES) - INPUT
C     DINC - DISTANCE INCREMENT (IN DEGREES) - INPUT
C     DPI - INITIAL HYPOCENTER DEPTH (IN KILOMETERS) - INPUT
C     DPF - FINAL HYPOCENTER DEPTH (IN KILOMETERS) - INPUT
C     DPINC - HYPOCENTER DEPTH INCREMENT (IN KILOMETERS) - INPUT
C             (ONLY 4 DEPTH VALUES AT A TIME)
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 14, 1993  COLORADO
C
C*****************************************************************
C
      DIMENSION DEPTH(4),TT(4),MIN(4),SEC(4)
C
      EELAT=0.0
      SLAT=0.0
      SELEV=0.0
      DEPTH(1)=DPI
      NUM=1
      IF(DPINC.LE.0.0)GO TO 500
C
      DO 100 I=2,4
      NUM=I
      DEPTH(I)=DEPTH(I-1)+DPINC
      IF(DEPTH(I).GT.DPF)GO TO 200
  100 CONTINUE
C
      GO TO 500
C
  200 NUM=NUM-1
C
  500 WRITE(LUN,510)NPHAS
  510 FORMAT(/,10X,'TRAVEL TIMES FOR PHASE ',A4,' AND DEPTHS:',/)
      WRITE(LUN,550)(DEPTH(I),I=1,NUM)
  550 FORMAT(/,21X,F5.1,9X,F5.1,8X,F5.1,8X,F5.1)
      WRITE(LUN,600)
  600 FORMAT(/)
C
      DIST=DI
C
 1000 DO 2000 I=1,NUM
C
      TT(I)=TRVLTM(DIST,DEPTH(I),NPHAS,EELAT,SLAT,SELEV)
C
C     CHANGE SECONDS TO HOURS, MIMUTES, AND SECONDS
C
      CALL ITIMC(TT(I),IHR,MIN(I),SEC(I))
C
```

```
 2000 CONTINUE
C
      WRITE(LUN,3000)DIST,(MIN(J),SEC(J),J=1,NUM)
 3000 FORMAT(10X,F5.1,5X,I2,F7.2,4X,I2,F7.2,4X,I2,F7.2,4X,I2,F7.2)
C
      DIST=DIST+DINC
      IF(DIST.LE.DF)GO TO 1000
C
      RETURN
      END
```

```
      SUBROUTINE LISTTX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR LISTING TRAVEL TIMES
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 16, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DATA LAST/'last'/
C
C     READ TRAVEL TIME PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,200,END=9000)NPHAS
  200 FORMAT(A4)
C
C     NPHAS - PHASE NAME
C             = 'last' NO MORE DATA
C
      IF(NPHAS.EQ.LAST)GO TO 9000
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','DI',3X,'DF',3X,'DINC',/)
       READ(IRE,*,END=9000)DI,DF,DINC
C
C     DI - INITIAL DISTANCE VALUE (IN DEGREES)
C     DF - FINAL DISTANCE VALUE (IN DEGREES)
C     DINC - INCREMENT IN DISTANCE VALUE (IN DEGREES)
C
      WRITE(IWR2,400)
  400 FORMAT(/,'++ INPUT: ','DPI',3X,'DPINC',/)
      READ(IRE,*,END=9000)DPI,DPINC
C
C     DPI - INITIAL DEPTH VALUE (IN KILOMETERS)
C     DPINC - INCREMENT IN DEPTH VALUE (IN KILOMETERS)
C             (4 DEPTH VALUES PRINTED)
C
      WRITE(IWR,500)
      WRITE(IWR2,500)
  500 FORMAT(/,10X,'TRAVEL TIME PARAMETERS',/)
      WRITE(IWR,600)NPHAS
      WRITE(IWR2,600)NPHAS
  600 FORMAT(/,10X,'PHASE ',A4)
      WRITE(IWR,700)DI,DF,DINC
      WRITE(IWR2,700)DI,DF,DINC
  700 FORMAT(/,10X,'DI = ',F8.2,5X,'DF = ',F8.2,5X,'DINC = ',
     * F8.2)
      WRITE(IWR,800)DPI,DPINC
      WRITE(IWR2,800)DPI,DPINC
  800 FORMAT(/,10X,'DPI = ',F7.2,5X,'DPINC = ',F7.2,/)
C
```

```
      DPF=DPI+3.0*DPINC
      IF(DPF.GT.700.0)DPF=700.0
C
      CALL LISTTT(IWR,NPHAS,DI,DF,DINC,DPI,DPF,DPINC)
C
      CALL LISTTT(IWR2,NPHAS,DI,DF,DINC,DPI,DPF,DPINC)

 9000 RETURN
      END
```

```
      SUBROUTINE LOVAIJ(ARGK,WVNO,TTT,MM,KWRITE,MMAX,NTRLS,ILAST,LCT)
C
C*****************************************************************
C
C     FORM THE ELEMENTS OF HASKELL MATRICES FOR LOVE WAVES
C
C     ARGK - S WAVE ARGUMENT - INPUT
C     WVNO - WAVE NUMBER - INPUT
C     TTT - PERIOD (IN SECONDS) - INPUT
C     MM - CURRENT LAYER NUMBER - INPUT
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     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     NTRLS - NUMBER OF TRIALS TO FIND A ROOT - INPUT AND OUTPUT
C             = 1000 ERROR ENCOUNTERED
C     ILAST - LAST ITERATION FLAG - INPUT
C             = 0 NOT LAST ITERATION
C             = 1 LAST ITERATION
C     LCT - NUMBER OF SUBDIVISIONS TO THE CURRENT LAYER - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 14, 1994  COLORADO
C
C*****************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION V(1000,4),A(2,2)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(40001),V(1,1)),(BX(50801),A(1,1))
C
      REAL*8 A,QM,EP,ARGK,TKSQ,TK,FMUR,ETOQM,SINQM
C
      M=MM
      LCT=0
      PFV=0.50
      FMU=V(M,3)*V(M,3)*V(M,4)
      TKSQ=DABS(ARGK)
      TK=DSQRT(TKSQ)
      IF(ARGK.LT.0.0)TKSQ=-TKSQ
   50 FMUR=FMU*TK
C
C     IS THIS HALF-SPACE?
C
      IF(M.LT.MMAX)GO TO 200
C
C     IS HALF-SPACE FLUID?
C
      IF(V(M,3).LE.0.0)GO TO 100
C
C     SOLID HALF-SPACE
C
      A(1,1)=FMUR
```

```
      GO TO 9000
C
C     FLUID HALF-SPACE
C
  100 A(1,1)=0.0
      GO TO 9000
C
  200 QM=WVNO*TK*V(M,1)
      IF(ARGK.LT.0.0)GO TO 300
      A(1,1)=DCOS(QM)
      SINQM=DSIN(QM)
      GO TO 500
C
C     COMPUTE ELEMENT EXPONENT
C
  300 EP=2.0*QM
C
C     IS THIS FIRST LAYER?
C
      IF(M.LE.1)GO TO 400
C
C     CHECK MAGNITUDE OF EXPONENT
C
      IF(EP.GT.10.0)GO TO 700
C
C     COMPUTE MATRIX ELEMENTS FOR NEGATIVE ARGK
C
  400 ETOQM=DEXP(-EP)
      A(1,1)=PFV+PFV*ETOQM
      SINQM=PFV-PFV*ETOQM
C
      IF(ILAST.LE.0)GO TO 500
C
C     IF LAST PASS AFTER ROOT HAS BEEN FOUND, DON'T FACTOR OUT
C     LARGE TERMS
C
      ETOQM=DEXP(QM)
      A(1,1)=A(1,1)*ETOQM
      SINQM=SINQM*ETOQM
C
  500 A(1,2)=-SINQM/FMUR
      A(2,1)=SINQM*FMUR
      IF(ARGK.GE.0.0)GO TO 600
      A(2,1)=-A(2,1)
  600 A(2,2)=A(1,1)
      GO TO 9000
C
C     TRY SUBDIVIDING LAYERS IF EXPONENTS TOO LARGE
C
  700 DO 800 I=2,50
      LCT=I
      FLCT=LCT
C
C     CHECK MAGNITUDE OF EXPONENT AFTER LAYER SUBDIVISION
C
      IF((EP/FLCT).LE.10.0)GO TO 900
  800 CONTINUE
C
```

```
      GO TO 7000
C
C     RECOMPUTE EXPONENT
C
  900 QM=QM/FLCT
      EP=2.0*QM
C
      IF(KWRITE.NE.2)GO TO 400
      WRITE(IWR2,1000)M,LCT,TTT
 1000 FORMAT(10X,'LAYER ',I3,' SUBDIVIDED INTO ',I3,' LAYERS',/,
     * 10X,'FOR PERIOD OF ',F7.3,' SECONDS')
      GO TO 400
C
 7000 WRITE(IWR2,7100)M,TTT,QM
 7100 FORMAT(/,10X,'COULD NOT SUBDIVIDE LAYER ',I3,' INTO LESS THAN ',
     * '50 LAYERS',/,10X,'FOR PERIOD OF 'F7.3,' SECONDS WITH QM = ',
     * D15.5,/)
C
      MM=MMAX
      NTRLS=1000
C
9000  RETURN
      END
```

```
      SUBROUTINE LSRESX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR LISTING RESPONSE INFORMATION
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 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/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 RESPONSE LIST INFORMATION
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL NUMBER TO LIST RESPONSE
C             = 0 LIST RESPONSE FOR 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.LE.0)GO TO 1000
      NCH1=ICHAN
      NCH2=ICHAN
      GO TO 1500
C
 1000 NCH1=1
      NCH2=NACHLS
C
 1500 DO 5000 I=NCH1,NCH2
C
C     LIST HEADER INFORMATION
C
      CALL PRHEAD(I)
C
C     LIST CALIBRATION
C
      NP=JNP(I)
      NZ=JNZ(I)
C
      WRITE(IWR,1700)
```

```
      WRITE(IWR2,1700)
 1700 FORMAT(/,10X,'RESPONSE POLES',/)
      WRITE(IWR,1800)(RPOLES(K,I),K=1,NP)
      WRITE(IWR2,1800)(RPOLES(K,I),K=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(K,I),K=1,NZ)
      WRITE(IWR2,1800)(RZEROS(K,I),K=1,NZ)
      WRITE(IWR,1950)RA0(I)
      WRITE(IWR2,1950)RA0(I)
 1950 FORMAT(/,10X,'A0 = ',E10.3,/)
      WRITE(IWR,2000)JFLAG(I)
      WRITE(IWR2,2000)JFLAG(I)
 2000 FORMAT(/,10X,'JFLAG = ',I3,/)
      IF(I.EQ.NCH2)GO TO 5000
      WRITE(IWR,2500)
      WRITE(IWR2,2500)
 2500 FORMAT(/,10X,62('-'),/)
C
 5000 CONTINUE
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE LSTTX
C
C*****************************************************************
C
C     LIST TRAVEL TIMES FOR CURRENT STATIONS AND EVENT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     MARCH 22, 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/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
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
      DATA ILW/'L   '/,IRW/'R   '/
C
C     READ PHASE NAME
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,500,END=9000)NPHAS
  500 FORMAT(A4)
C
C     NPHAS - NAME OF DESIRED PHASE FOR TT COMPUTATION
C
C     CALCULATE TRAVEL TIMES AND DISTANCES
C
      UVEL=0.0
      PERIOD=0.0
      NPER=0
      IF(NPHAS.NE.ILW.AND.NPHAS.NE.IRW)GO TO 800
C
      WRITE(IWR2,600)
  600 FORMAT(/,'++ INPUT: ','UVEL',/)
      READ(IRE,*,END=9000)UVEL
C
C     UVEL - GROUP VELOCITY FOR RAYLEIGH OR LOVE WAVES (IN KM/S)
C            = 0 SET TO 4.0
C
      IF(UVEL.LE.0.0)UVEL=4.0
```

```
      WRITE(IWR,700)UVEL
      WRITE(IWR2,700)UVEL
  700 FORMAT(/,10X,'UVEL = ',F10.3,/)
C
      NPER=1
C
  800 WRITE(IWR,900)NPHAS
      WRITE(IWR2,900)NPHAS
  900 FORMAT(/,10X,'TRAVEL TIME COMPUTED FOR PHASE ',A4,/)
C
      WRITE(IWR,1000)
      WRITE(IWR2,1000)
 1000 FORMAT(/,10X,'STATION',5X,'DISTANCE',6X,'DISTANCE',8X,'TT',/)
C
      IDF=0
C
      DO 1500 I=1,NACHLS
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
      WRITE(IWR,1200)JCODE(I),DELTA(I),DISKM(I),CTT(I)
      WRITE(IWR2,1200)JCODE(I),DELTA(I),DISKM(I),CTT(I)
 1200 FORMAT(12X,A4,5X,F8.2,6X,F8.2,6X,F8.3)
 1500 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE MATCH(X,ISAMP,C,LF,N2PWR,N2LEN,NH2)
C
C*********************************************************************
C
C     FORM CORRELOGRAM OF TWO TIME SERIES
C
C     X - TIME SERIES - INPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C     C - MATCHED FILTER - INPUT
C       - CORRELOGRAM - OUTPUT
C     LF - NUMBER OF SAMPLES IN MATCHED FILTER - INPUT
C     N2PWR - POWER OF 2 TO TRANSFORM DATA AND FILTER - INPUT
C     N2LEN - NUMBER OF POINTS FOR N2PWR - INPUT
C     NH2 - NUMBER OF COMPONENTS = NELEN/2+1 - INPUT
C
C     NOVEMBER 18, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1),C(1)
C
C     ZERO DATA ARRAY BEYOND ISAMP AND FILTER BEYOND LF
C
      DO 300 I=1,N2LEN
      IF((LF-I).GE.0)GO TO 100
      C(I)=0.0
  100 IF((ISAMP-I).GE.0)GO TO 300
      X(I)=0.0
  300 CONTINUE
C
C     COMPUTE TWO DIRECT TRANSFORMS
C
      CALL CLTWO(X,C,N2PWR,-1)
C
      J=NH2
C
C     FORM CORRELOGRAM IN THE COMPLEX DOMAIN
C
      DO 400 I=1,NH2
      J=J+1
      Z=X(I)*X(J)+C(I)*C(J)
      C(J)=C(I)*X(J)-X(I)*C(J)
      X(J)=Z
  400 CONTINUE
C
C     COMPUTE INVERSE TRANSFORM
C
      CALL CLTWO(X,C,N2PWR,+1)
C
C     CORRELOGRAM IS IN THE C ARRAY
C
C     ZERO STORAGE USED BY CLFORM
C
      C(N2LEN+1)=0.0
      C(N2LEN+2)=0.0
      X(N2LEN+1)=0.0
      X(N2LEN+2)=0.0
C
```

```
C      SCALE ARRAYS
C
       DO 500 I=1,N2LEN
       X(I)=X(I)/N2LEN
       C(I)=4.0*C(I)/(N2LEN*N2LEN)
  500  CONTINUE
C
       RETURN
       END
```

```
      SUBROUTINE MATCHX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR MATCHED FILTERING
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 DATA TIME SERIES
C              = 0 SET TO 1
C     ICHAN2 - CHANNEL NUMBER FOR MATCHED FILTER
C               (CORRELOGRAM PLACED IN THIS CHANNEL AFTER FILTERING)
C
      IF(ICHAN1.LE.0)ICHAN1=1
      WRITE(IWR,600)ICHAN1,ICHAN2
      WRITE(IWR2,600)ICHAN1,ICHAN2
  600 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN2.LE.0)GO TO 7000
      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 TIME SERIES = ',3X,I6,/,10X,
     * 'SAMPLES IN MATCHED FILTER = ',I6,/)
      WRITE(IWR,900)SAMR1,SAMR2
      WRITE(IWR2,900)SAMR1,SAMR2
  900 FORMAT(/,10X,'SAMPLING RATE FOR TIME SERIES = ',F10.3,/,10X,
     * 'SAMPLING RATE FOR MATCHED FILTER = ',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     MATCHED FILTER 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 MATCHED FILTERED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY MATCHED FILTERED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE METR(DATA,ISAMP,IDEG)
C
C*********************************************************************
C
C     REMOVE MEAN AND LINEAR TREND
C
C     DATA - TIME SERIES - INPUT
C     ISAMP - NUMBER OF POINTS IN A TIME SERIES - INPUT
C     IDEG -  FLAG FOR MEAN OR TREND - INPUT
C             = 0 REMOVE MEAN FROM THE TIME SERIES
C             = 1 REMOVE MEAN AND LINEAR TREND FROM THE TIME SERIES
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JULY 16, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/METD/TM1,TM2,IFP,ILP
C
      DIMENSION DATA(1)
C
      TM1=0.0
      TM2=0.0
      SUMY=0.0
C
C     CHECK FOR LEADING ZEROS
C
      IFP=1
C
      DO 50 J=1,ISAMP
      IF(DATA(J).NE.0.0)GO TO 60
      IFP=J+1
   50 CONTINUE
C
   60 IF(IFP.LE.3)IFP=1
C
C     CHECK FOR TRAILING ZEROS
C
      ILP=ISAMP
C
      DO 150 J=1,ISAMP
      K=ISAMP-J+1
      IF(DATA(K).NE.0.0)GO TO 180
      ILP=K-1
  150 CONTINUE
C
  180 FN=ILP-IFP+1
C
      IF(IDEG.GT.0)GO TO 500
C
C     REMOVE THE MEAN
C
      DO 200 J=IFP,ILP
      SUMY=SUMY+DATA(J)
  200 CONTINUE
C
      FMEAN=SUMY/FN
```

```
C
      DO 300 J=IFP,ILP
      DATA(J)=DATA(J)-FMEAN
  300 CONTINUE
C
      TM1=0.0
      TM2=FMEAN
      GO TO 9000
C
C     REMOVE MEAN AND LINEAR TREND
C
  500 SUMX=((FN+1.0)*FN)/2.0
      SUMXX=(FN*(FN+1.0)*(2.0*FN+1.0))/6.0
      CC=SUMXX*FN-SUMX*SUMX
      SUMXY=0.0
C
      DO 600 J=IFP,ILP
      FJ=J
      SUMY=SUMY+DATA(J)
      SUMXY=SUMXY+FJ*DATA(J)
  600 CONTINUE
C
      AA=(FN*SUMXY-SUMX*SUMY)/CC
      FMEAN=(SUMXX*SUMY-SUMXY*SUMX)/CC
C
      DO 700 J=IFP,ILP
      FJ=J
      DATA(J)=DATA(J)-AA*FJ-FMEAN
  700 CONTINUE
C
      TM1=AA
      TM2=FMEAN
C
 9000 RETURN
      END
```

```
      SUBROUTINE METX(IDEG)
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR REMOVING THE MEAN AND LINEAR TREND
C
C     IDEG - FLAG FOR MEAN OR TREND - INPUT
C            = 0 REMOVE MEAN
C            = 1 REMOVE MEAN AND LINEAR TREND
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 19, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/METD/TM1,TM2,IFP,ILP
      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 - DESIRED CHANNEL
C             = 0 REMOVE MEAN OR TREND FROM ALL CHANNELS
C
      WRITE(IWR,200)
      WRITE(IWR2,200)
  200 FORMAT(/,10X,'REMOVE MEAN/TREND PARAMETERS',/)
      WRITE(IWR2,300)ICHAN
  300 FORMAT(/,10X,'CHANNEL SELECTED IS ',I4,/)
C
      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     REMOVE MEAN OR LINEAR TREND
C
 2000 DO 3000 I=NCH1,NCH2
      ISAMP=JSAMP(I)
```

```
      INDEX=(I-1)*LENG+1
C
      CALL METR(DATA(INDEX),ISAMP,IDEG)
C
      IF(IDEG.NE.0)GO TO 2500
C
      WRITE(IWR,2100)TM2,I
      WRITE(IWR2,2100)TM2,I
 2100 FORMAT(/,10X,'MEAN OF ',F10.3,' FOR CHANNEL ',I4,/,
     * 10X,'WAS REMOVED',/)
      GO TO 3000
C
 2500 WRITE(IWR,2200)TM1,TM2,I
      WRITE(IWR2,2200)TM1,TM2,I
 2200 FORMAT(/,10X,'LINEAR TREND OF '2F9.3,' REMOVED FOR CHANNEL ',
     * I4,/)
C
 3000 CONTINUE
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'MEAN/TREND NOT REMOVED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE MNMX(X,ISAMP,FPM,IPM,FPX,IPX,FNM,INM,FNX,INX)
C
C*********************************************************************
C
C     DETERMINE MIMIMUM/MAXIMUM OF DATA SET
C
C     X - DATA SET - INPUT
C     ISAMP - NUMBER OF SAMPLES IN DATA SET - INPUT
C     FPM - POSITIVE VALUE MINIMUM - OUTPUT
C     IPM - INDEX OF POSITIVE VALUE MINIMUM - OUTPUT
C     FPX - POSITIVE VALUE MAXIMUM - OUTPUT
C     IPX - INDEX OF POSITIVE VALUE MAXIMUM - OUTPUT
C     FNM - NEGATIVE VALUE MINIMUM - OUTPUT
C     INM - INDEX OF NEGATIVE VALUE MINIMUM - OUTPUT
C     FNX - NEGATIVE VALUE MAXIMUM - OUTPUT
C     INX - INDEX OF NEGATIVE VALUE MAXIMUM - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 6, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1)
C
      FPM=0.0
      IPM=0
      FPX=0.0
      IPX=0
      FNM=0.0
      INM=0
      FNX=0.0
      INX=0
C
      DO 1000 I=1,ISAMP
      IF(X(I).LT.0.0)GO TO 500
C
      IF(IPM.NE.0)GO TO 100
      FPM=X(I)
      IPM=I
      FPX=X(I)
      IPX=I
      GO TO 1000
  100 IF(X(I).GE.FPM)GO TO 200
      FPM=X(I)
      IPM=I
  200 IF(X(I).LE.FPX)GO TO 1000
      FPX=X(I)
      IPX=I
      GO TO 1000
C
  500 IF(INM.NE.0)GO TO 600
      FNM=X(I)
      INM=I
      FNX=X(I)
      INX=I
      GO TO 1000
  600 IF(X(I).GE.FNM)GO TO 700
```

```
      FNM=X(I)
      INM=I
  700 IF(X(I).LE.FNX)GO TO 1000
      FNX=X(I)
      INX=I
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MNMXEX
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR DETERMINING MINIMUM AND MAXIMUM OF
C     A DATA CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 7, 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 CHANNEL NUMBER
C
  100 WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL TO EXAMINE
C             = 0 SET TO 1
C
      IF(ICHAN.LE.0)ICHAN=1
      WRITE(IWR,300)ICHAN
      WRITE(IWR2,300)ICHAN
  300 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      INDEX=(ICHAN-1)*LENG+1
      ISAMP=JSAMP(ICHAN)
C
      CALL MNMX(DATA(INDEX),ISAMP,FPM,IPM,FPX,IPX,FNM,INM,FNX,INX)
C
      WRITE(IWR,500)ICHAN,ISAMP
      WRITE(IWR2,500)ICHAN,ISAMP
  500 FORMAT(/,10X,'MAXIMUM/MINIMUM VALUES FOR CHANNEL ',I4,/,10X,
     * 'WITH ',I6,' SAMPLES ARE: ',/)
      WRITE(IWR,600)FPM,IPM,FPX,IPX,FNM,INM,FNX,INX
      WRITE(IWR2,600)FPM,IPM,FPX,IPX,FNM,INM,FNX,INX
  600 FORMAT(/,10X,'POSITIVE MINIMUM IS ',E10.3,' AT SAMPLE ',I6,/,
     * 10X,'POSITIVE MAXIMUM IS ',E10.3,' AT SAMPLE ',I6,/,
     * 10X,'NEGATIVE MINIMUM IS ',E10.3,' AT SAMPLE ',I6,/,
     * 10X,'NEGATIVE MAXIMUM IS ',E10.3,' AT SAMPLE ',I6,/)
```

```
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE MOVE(IFL,IB1,IL1,ICHAN1,IB2,ICHAN2,IHF)
C
C*****************************************************************************
C
C     MOVE ANY SPECIFIED PART OF ANY DATA CHANNEL
C
C     IFL - FLAG FOR DATA TRANSFER - INPUT
C           = 0 DATA ARRAY TO DATA ARRAY
C           = 1 DATA ARRAY TO BX ARRAY
C           = 2 BX ARRAY TO DATA ARRAY
C     IB1 - FIRST POINT OF DATA TO MOVE - INPUT
C           (MUST = 1 OR GREATER)
C     IL1 - LAST POINT OF DATA TO MOVE - INPUT
C           (MUST = 1 OR GREATER)
C     ICHAN1 - CHANNEL OF DATA TO BE MOVED - INPUT
C              (MUST = 1 OR GREATER)
C     IB2 - FIRST POINT IN CHANNEL TO STORE DATA - INPUT
C           (MUST = 1 OR GREATER)
C     ICHAN2 - CHANNEL TO STORE DATA - INPUT
C              (MUST = 1 OR GREATER)
C     IHF - MOVE HEADER FLAG - INPUT
C           = 0 MOVE HEADER IF IFL = 0
C           = 1 DO NOT MOVE HEADER IF IFL = 0
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 2, 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

      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
      INDEX1=(ICHAN1-1)*LENG
      INDEX2=(ICHAN2-1)*LENG
C
      IF(IFL.EQ.1)GO TO 1000
      IF(IFL.EQ.2)GO TO 2000
C
C     MOVE DATA ARRAY TO DATA ARRAY
C
      K=0
C
      DO 500 J=IB1,IL1
      IN1=INDEX1+J
      IN2=INDEX2+IB2+K
```

```
      DATA(IN2)=DATA(IN1)
      K=K+1
  500 CONTINUE
C
      JSAMP(ICHAN2)=IL1-IB1+1
      IF(IHF.GT.0)GO TO 9000
C
C     MOVE HEADER INFORMATION
C
      CALL MOVHD(ICHAN1,ICHAN2,1)
C
      GO TO 9000
C
C     MOVE DATA ARRAY TO BX ARRAY
C
 1000 K=0
C
      DO 1500 J=IB1,IL1
      IN1=INDEX1+J
      IN2=IB2+K
      BX(IN2)=DATA(IN1)
      K=K+1
 1500 CONTINUE
C
      GO TO 9000
C
C     MOVE BX ARRAY TO DATA ARRAY
C
 2000 K=0
C
      DO 2500 J=IB1,IL1
      IN1=J
      IN2=INDEX2+IB2+K
      DATA(IN2)=BX(IN1)
      K=K+1
 2500 CONTINUE
C
      JSAMP(ICHAN2)=IL1-IB1+1
C
 9000 RETURN
      END
```

```
      SUBROUTINE MOVEX
C
C*****************************************************************
C
C     EXECUTIVE ROUTINE FOR MOVING A DATA CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 20, 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 MOVE PARAMETERS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','IFL',3X,'IHF',/)
      READ(IRE,*,END=9000)IFL,IHF
C
C     IFL - FLAG FOR DATA TRANSFER
C           = 0 DATA ARRAY TO DATA ARRAY
C           = 1 DATA ARRAY TO BX ARRAY
C           = 2 BX ARRAY TO DATA ARRAY
C     IHF - MOVE HEADER FLAG
C           = 0 MOVE DATA AND MOVE HEADER IF IFL = 0
C           = 1 MOVE DATA BUT DO NOT MOVE HEADER IF IFL = 0
C
      WRITE(IWR2,150)
  150 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'IFIR1',3X,'ILAS1',/)
      READ(IRE,*,END=9000)ICHAN1,IFIR1,ILAS1
      WRITE(IWR2,200)
C
C     ICHAN1 - CHANNEL OF DATA TO BE MOVED
C              = 0 SET TO 1
C     IFIR1 - FIRST POINT OF DATA TO MOVE
C             = 0 SET TO 1
C     ILAS1 - LAST POINT OF DATA TO MOVE
C             = 0 SET TO JSAMP
C
  200 FORMAT(/,'++ INPUT: ','ICHAN2',3X,'IFIR2',/)
      READ(IRE,*,END=9000)ICHAN2,IFIR2
C
C     ICHAN2 - CHANNEL TO STORE DATA
C              = 0 SET TO NACHLS+1
C     IFIR2 - FIRST POINT IN CHANNEL TO STORE DATA
C             = 0 SET TO 1
C
```

```
      IF(IFIR1.LE.0)IFIR1=1
      IF(IFIR2.LE.0)IFIR2=1
      IF(ICHAN1.LE.0)ICHAN1=1
      IF(ICHAN2.LE.0)ICHAN2=NACHLS+1
      IF(ICHAN2.GT.NACHLS)NACHLS=ICHAN2
      IF(IFL.EQ.1)ICHAN2=1
      IF(IFL.EQ.2)ICHAN1=1
      IF(IFL.EQ.0.AND.ILAS1.LE.0)ILAS1=JSAMP(ICHAN1)
      IF(IFL.EQ.1.AND.ILAS1.LE.0)ILAS1=JSAMP(ICHAN2)
      IF(IFL.EQ.2.AND.ILAS1.LE.0)ILAS1=LENG
      IF(ILAS1.GT.LENG)ILAS1=LENG
C
      WRITE(IWR,250)
      WRITE(IWR2,250)
  250 FORMAT(/,10X,'MOVE PARAMETERS',/)
      WRITE(IWR,270)IFL,IHF
      WRITE(IWR2,270)IFL,IHF
  270 FORMAT(/,10X,'IFL = ',I2,5X,'IHF = ',I2,/)
      WRITE(IWR,300)IFIR1,ILAS1,ICHAN1,IFIR2,ICHAN2
      WRITE(IWR2,300)IFIR1,ILAS1,ICHAN1,IFIR2,ICHAN2
  300 FORMAT(/,10X,'IFIR1 = ',I2,5X,'ILAS1 = ',I6,5X,
     * 'ICHAN1 = ',I4,//,10X,'IFIR2 = ',I2,24X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NCHLS)GO TO 7000
C
      CALL MOVE(IFL,IFIR1,ILAS1,ICHAN1,IFIR2,ICHAN2,IHF)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'NO DATA MOVED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY MOVED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE MOVHD(ICHAN1,ICHAN2,IFL)
C
C*****************************************************************************
C
C     MOVE CHANNEL HEADER INFORMATION
C
C     ICHAN1 - CHANNEL TO BE MOVED - INPUT
C     ICHAN2 - CHANNEL TO MOVE TO OR EXCHANGE - INPUT
C     IFL - MOVE HEADER FLAG - INPUT
C           = 0 EXCHANGE HEADERS
C           = 1 ICHAN1 HEADER TO ICHAN2
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 25, 1993  COLORADO
C
C*****************************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
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,ZZ
C
      IF(IFL.EQ.0)GO TO 1000
C
      JCODE(ICHAN2)=JCODE(ICHAN1)
      RLAT(ICHAN2)=RLAT(ICHAN1)
      RLON(ICHAN2)=RLON(ICHAN1)
      RELEV(ICHAN2)=RELEV(ICHAN1)
      JYEAR(ICHAN2)=JYEAR(ICHAN1)
      JDOFY(ICHAN2)=JDOFY(ICHAN1)
      JHOUR(ICHAN2)=JHOUR(ICHAN1)
      JMIN(ICHAN2)=JMIN(ICHAN1)
      RSEC(ICHAN2)=RSEC(ICHAN1)
      RSAMR(ICHAN2)=RSAMR(ICHAN1)
      RA0(ICHAN2)=RA0(ICHAN1)
      JNP(ICHAN2)=JNP(ICHAN1)
      JNZ(ICHAN2)=JNZ(ICHAN1)
      JSAMP(ICHAN2)=JSAMP(ICHAN1)
      JFLAG(ICHAN2)=JFLAG(ICHAN1)
C
      DO 500 I=1,4
      JCHN(I,ICHAN2)=JCHN(I,ICHAN1)
  500 CONTINUE
C
      DO 600 I=1,30
      RPOLES(I,ICHAN2)=RPOLES(I,ICHAN1)
  600 CONTINUE
C
      DO 700 I=1,20
      RZEROS(I,ICHAN2)=RZEROS(I,ICHAN1)
  700 CONTINUE
C
```

```
      GO TO 9000
C
 1000 JJ=JCODE(ICHAN1)
      JCODE(ICHAN1)=JCODE(ICHAN2)
      JCODE(ICHAN2)=JJ
      RR=RLAT(ICHAN1)
      RLAT(ICHAN1)=RLAT(ICHAN2)
      RLAT(ICHAN2)=RR
      RR=RLON(ICHAN1)
      RLON(ICHAN1)=RLON(ICHAN2)
      RLON(ICHAN2)=RR
      RR=RELEV(ICHAN1)
      RELEV(ICHAN1)=RELEV(ICHAN2)
      RELEV(ICHAN2)=RR
      JJ=JYEAR(ICHAN1)
      JYEAR(ICHAN1)=JYEAR(ICHAN2)
      JYEAR(ICHAN2)=JJ
      JJ=JDOFY(ICHAN1)
      JDOFY(ICHAN1)=JDOFY(ICHAN2)
      JDOFY(ICHAN2)=JJ
      JJ=JHOUR(ICHAN1)
      JHOUR(ICHAN1)=JHOUR(ICHAN2)
      JHOUR(ICHAN2)=JJ
      JJ=JMIN(ICHAN1)
      JMIN(ICHAN1)=JMIN(ICHAN2)
      JMIN(ICHAN2)=JJ
      RR=RSEC(ICHAN1)
      RSEC(ICHAN1)=RSEC(ICHAN2)
      RSEC(ICHAN2)=RR
      RR=RSAMR(ICHAN1)
      RSAMR(ICHAN1)=RSAMR(ICHAN2)
      RSAMR(ICHAN2)=RR
      RR=RA0(ICHAN1)
      RA0(ICHAN1)=RA0(ICHAN2)
      RA0(ICHAN2)=RR
      JJ=JNP(ICHAN1)
      JNP(ICHAN1)=JNP(ICHAN2)
      JNP(ICHAN2)=JJ
      JJ=JNZ(ICHAN1)
      JNZ(ICHAN1)=JNZ(ICHAN2)
      JNZ(ICHAN2)=JJ
      JJ=JSAMP(ICHAN1)
      JSAMP(ICHAN1)=JSAMP(ICHAN2)
      JSAMP(ICHAN2)=JJ
      JJ=JFLAG(ICHAN1)
      JFLAG(ICHAN1)=JFLAG(ICHAN2)
      JFLAG(ICHAN2)=JJ
C
      DO 1500 I=1,4
      JJ=JCHN(I,ICHAN1)
      JCHN(I,ICHAN1)=JCHN(I,ICHAN2)
      JCHN(I,ICHAN2)=JJ
 1500 CONTINUE
C
      DO 1600 I=1,30
      ZZ=RPOLES(I,ICHAN1)
      RPOLES(I,ICHAN1)=RPOLES(I,ICHAN2)
      RPOLES(I,ICHAN2)=ZZ
```

```
 1600 CONTINUE
C
      DO 1700 I=1,20
      ZZ=RZEROS(I,ICHAN1)
      RZEROS(I,ICHAN1)=RZEROS(I,ICHAN2)
      RZEROS(I,ICHAN2)=ZZ
 1700 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE NSNFIL
C
C*********************************************************************
C
C     SET UP FILE OF STATION-TIMES FOR RETRIEVING NSN DATA
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 7, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
C
      DIMENSION LCODE(200),SLAT(200),SLON(200),SELEV(200),LYR(200),
     * LDOY(200),LHR(200),LMIN(200),SSEC(200),LCOMP(200,3)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(1),LCODE(1)),(BX(401),SLAT(1)),(BX(801),SLON(1))
      EQUIVALENCE (BX(1201),SELEV(1)),(BX(1601),LYR(1))
      EQUIVALENCE (BX(2001),LDOY(1)),(BX(2401),LHR(1))
      EQUIVALENCE (BX(2801),LMIN(1)),(BX(3201),SSEC(1))
      EQUIVALENCE (BX(4001),LCOMP(1,1))
C
      CHARACTER*40 FILENM,FILENM2
C
      DATA ISTOP/'STOP'/,JSTOP/'stop'/,NY/'y'/
      DATA ISS/'S'/,ILL/'L'/,III/'I'/,IBB/'B'/,IAA/'A'/,IHH/'H'/
      DATA LSS/'s'/,LLL/'l'/,LII/'i'/,LBB/'b'/,LAA/'a'/,LHH/'h'/
      DATA IGG/'G'/,IMM/'M'/,IZZ/'Z'/,INN/'N'/,IEE/'E'/
      DATA LGG/'g'/,LMM/'m'/,LZZ/'z'/,LNN/'n'/,LEE/'e'/
      DATA ILW/'L   '/,IRW/'R   '/
C
      MAX=200
C
  300 WRITE(IWR2,310)
  310 FORMAT(/,'++ INPUT: ','FILE NAME FOR INPUT STATION LIST',/)
      READ(IRE,315,END=9500)FILENM
  315 FORMAT(A40)
C
C     FILENM - FILE NAME FOR INPUT STATION LIST
C
      WRITE(IWR2,320)FILENM
  320 FORMAT(/,10X,'SELECTED FILE NAME IS ',A40,
     * //,'++ INPUT: ','IS IT CORRECT (y/n)?',/)
      READ(IRE,330,END=9500)KY
  330 FORMAT(A1)
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 300
C
      OPEN(LUN3,FILE=FILENM,STATUS='old')
C
      REWIND LUN3
C
```

```
C     READ STATION LIST
C
      K=1
 1000 READ(LUN3,1100,END=9200)LCODE(K),SLAT(K),SLON(K),SELEV(K)
 1100 FORMAT(A4,3F10.4)
C
C     LCODE - STATION CODE
C     SLAT - STATION GEOGRAPHIC LATITUDE (IN DEGREES)
C            - = S
C     SLON - STATION GEOGRAPHIC LONGITUDE (IN DEGREES)
C            - = W
C     SELEV - STATION ELEVATION (IN KILOMETERS)
C
      IF(LCODE(K).EQ.ISTOP)GO TO 1500
      IF(LCODE(K).EQ.JSTOP)GO TO 1500
      K=K+1
      IF(K.LE.MAX)GO TO 1000
C
 1500 NSTA=K-1
C
C     SELECT STATIONS
C
      ICNT=0
      NST=NSTA-1
 1550 WRITE(IWR2,1600)(K,LCODE(K),SLAT(K),SLON(K),SELEV(K),K=1,NSTA)
 1600 FORMAT(10X,I3,5X,A4,5X,F10.3,5X,F10.3,5X,F10.3,)
C
C     DELETE STATIONS
C
 1650 WRITE(IWR2,1700)
 1700 FORMAT(/,'++ INPUT: ','NDEL',3X,'NDEL2',/)
 1730 READ(IRE,*,END=1800)NDEL,NDEL2
C
C     NDEL - NUMBERS OF THE STATIONS TO DELETE
C            (MUST BE IN INCREASING ORDER)
C            < 0 OR end-of-file STOP
C     NDEL2 - DELETE THROUGH THIS NUMBER
C             = 0 JUST DELETE NDEL
C
      WRITE(IWR,1735)NDEL,NDEL2
      WRITE(IWR2,1735)NDEL,NDEL2
 1735 FORMAT(/,10X,'NDEL = 'I5,5X,'NDEL2 = ',I5,/)
C
      NDELX=NDEL
      IF(NDEL.LT.0)GO TO 1800
      IF(NDEL.EQ.0)GO TO 1650
 1740 NDEL=NDELX
      IF(NDEL.GT.NSTA)GO TO 1650
      NDEL=NDEL-ICNT
C
      DO 1750 K=NDEL,NST
      J=K+1
      LCODE(K)=LCODE(J)
      SLAT(K)=SLAT(J)
      SLON(K)=SLON(J)
      SELEV(K)=SELEV(J)
 1750 CONTINUE
C
```

```
      ICNT=ICNT+1
      NST=NST-1
      IF(NDELX.GE.NDEL2)GO TO 1730
      NDELX=NDELX+1
      GO TO 1740
C
 1800 NSTA=NST+1
      IF(ICNT.LE.0)GO TO 1900
      WRITE(IWR2,1850)
 1850 FORMAT(//,10X,'STATIONS SELECTED',/)
      WRITE(IWR2,1600)(K,LCODE(K),SLAT(K),SLON(K),SELEV(K),K=1,NSTA)
C
C     READ EPICENTER COORDINATES
C
 1900 WRITE(IWR2,1950)
 1950 FORMAT(/,'++ INPUT: ','EELAT',5X,'EELONG',5X,'DEPTH',/)
      READ(IRE,*,END=9200)EELAT,EELONG,DEPTH
C
C     EELAT - EPICENTER GEOGRAPHIC LATITUDE (IN DEGREES)
C             - = S
C     EELONG - EPICENTER GEOGRAPHIC LONGITUDE (IN DEGREES)
C              - = W
C     DEPTH - HYPOCENTER DEPTH (IN KILOMETERS)
C
      WRITE(IWR,1960)
      WRITE(IWR2,1960)
 1960 FORMAT(//,10X,'EVENT HYPOCENTER',/)
      WRITE(IWR,1970)EELAT,EELONG,DEPTH
      WRITE(IWR2,1970)EELAT,EELONG,DEPTH
 1970 FORMAT(10X,'EELAT = ',F7.2,5X,'EELONG = ',F7.2,5X,
     * 'DEPTH = ',F7.2,/)
C
C     READ EVENT ORIGIN TIME
C
      WRITE(IWR2,2000)
 2000 FORMAT(/,'++ INPUT: ','IYR',3X,'IMO',3X,'IDY',3X,'IHR',3X,
     * 'IMIN',3X,'SEC',/)
      READ(IRE,*,END=9200)IYR,IMO,IDY,IHR,IMIN,SEC
C
C     IYR - EVENT YEAR
C     IMO - EVENT MONTH (NUMBER)
C     IDY - EVENT DAY
C     IHR - EVENT HOUR
C     IMIN - EVENT MINUTE
C     SEC - EVENT SECONDS
C
      WRITE(IWR,2100)
      WRITE(IWR2,2100)
 2100 FORMAT(//,10X,'EVENT ORIGIN TIME',/)
      WRITE(IWR,2200)IYR,IMO,IDY,IHR,IMIN,SEC
      WRITE(IWR2,2200)IYR,IMO,IDY,IHR,IMIN,SEC
 2200 FORMAT(10X,'IYR = ',I4,3X,'IMO = ',I3,3X,'IDY = ',I2,//,10X,
     * 'IHR = ',I2,5X,'IMIN = ',I2,3X,'SEC = ',F5.2,/)
C
      WRITE(IWR2,2500)
 2500 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,2600,END=9200)NPHAS
 2600 FORMAT(A4)
```

```
C
C     NPHAS - DESIRED PHASE NAME
C
      WRITE(IWR,2700)NPHAS
      WRITE(IWR2,2700)NPHAS
 2700 FORMAT(/,10X,'TRAVEL TIMES COMPUTED FOR PHASE ',A4,/)
C
      IDF=0
      NPER=0
      IF(NPHAS.EQ.ILW.OR.NPHAS.EQ.IRW)GO TO 2710
      GO TO 2760
C
 2710 WRITE(IWR2,2720)
 2720 FORMAT(/,'++ INPUT: ','UVEL',/)
      READ(IRE,*,END=9200)UVEL
C
C     UVEL - GROUP VELOCITY FOR RAYLEIGH OR LOVE WAVES (IN KM/S)
C            = 0 SET TO 4.0
C
      IF(UVEL.LE.0.0)UVEL=4.0
      WRITE(IWR,2740)UVEL
      WRITE(IWR2,2740)UVEL
 2740 FORMAT(/,10X,'UVEL = ',F10.3,/)
C
      NPER=1
C
 2760 DO 3000 K=1,NSTA
C
      CALL TTCOMP(SLAT(K),SLON(K),SELEV(K),EELAT,EELONG,DEPTH,
     * NPHAS,DELTA,DISKM,AZSE,AZES,TT,UVEL,PERIOD,NPER,IDF)
C
      IF(TT.NE.0.0)GO TO 2800
      WRITE(IWR2,2750)LCODE(K)
 2750 FORMAT(/,10X,'TRAVEL TIME IS ZERO FOR STATION ',A4,/)
C
 2800 BT=TT-20.0
C
      CALL ADDTIM(IYR,IMO,IDY,IHR,IMIN,SEC,BT,LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K))
C
 3000 CONTINUE
C
 3200 WRITE(IWR2,3500)
 3500 FORMAT(/,'++ INPUT: ','FILE NAME FOR OUTPUT',/)
      READ(IRE,315,END=9200)FILENM2
C
C     FILENM2 - FILE NAME FOR OUTPUT LIST OF STATION-TIMES
C
      WRITE(IWR2,320)FILENM2
      READ(IRE,330,END=9200)KY
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 3200
C
      OPEN(LUN4,FILE=FILENM2,STATUS='unknown')
C
      REWIND LUN4
```

```
C
      ICF=0
C
      WRITE(IWR2,4000)
 4000 FORMAT(/,'++ INPUT: ','KC1',3X,'KC2',3X,'KC3',/)
      READ(IRE,4100,END=9100)KC1,KC2,KC3
 4100 FORMAT(3A1)
C
C     KC1 - COMPONENT S, L, I, OR B
C           = A IF BHZ, BHN, AND BHE DESIRED
C     KC2 - COMPONENT H, L, G, OR M
C     KC3 - COMPONENT Z, N, OR E
C
      IF(KC1.EQ.LSS)KC1=ISS
      IF(KC1.EQ.LLL)KC1=ILL
      IF(KC1.EQ.LII)KC1=III
      IF(KC1.EQ.LBB)KC1=IBB
      IF(KC1.EQ.LAA)KC1=IAA
      IF(KC2.EQ.LHH)KC2=IHH
      IF(KC2.EQ.LLL)KC2=ILL
      IF(KC2.EQ.LGG)KC2=IGG
      IF(KC2.EQ.LMM)KC2=IMM
      IF(KC3.EQ.LZZ)KC3=IZZ
      IF(KC3.EQ.LNN)KC3=INN
      IF(KC3.EQ.LEE)KC3=IEE
C
      WRITE(IWR2,4200)
 4200 FORMAT(/,'++ INPUT: ','SECLEN',/)
      READ(IRE,*,END=9100)SECLEN
C
C     SECLEN - AMOUNT OF DATA TO ACQUIRE (IN SECONDS)
C              = 0 SET TO 400.0
C
      IF(SECLEN.LE.0.0)SECLEN=400.0
      IF(KC1.NE.IAA)GO TO 4500
      ICF=1
      KC1=IBB
      KC2=IHH
      KC3=IZZ
C
 4500 DO 5000 K=1,NSTA
      LCOMP(K,1)=KC1
      LCOMP(K,2)=KC2
      LCOMP(K,3)=KC3
      WRITE(LUN4,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR2,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
 4700 FORMAT(A4,2X,3A1,2X,I4,1X,I3,1X,I2,1X,I2,1X,F5.2,3X,F7.2)
      IF(ICF.EQ.0)GO TO 5000
      LCOMP(K,3)=INN
      WRITE(LUN4,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR2,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
```

```
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      LCOMP(K,3)=IEE
      WRITE(LUN4,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
      WRITE(IWR2,4700)LCODE(K),(LCOMP(K,J),J=1,3),LYR(K),LDOY(K),
     * LHR(K),LMIN(K),SSEC(K),SECLEN
 5000 CONTINUE
C
 9100 CLOSE(LUN4)
C
 9200 CLOSE(LUN3)
C
 9500 RETURN
      END
```

```
      FUNCTION PASTIM(JUMP,DELTX,DEPHX)
C
C*****************************************************************************
C
C     DETERMINE FIRST ARRIVING P OR S TRAVEL TIME FOR A GIVEN
C     DISTANCE AND DEPTH
C            FOR P - P, PDIFF, OR PKP
C            FOR S - S OR SKS
C            IASPEI 1991 TABLES USED
C
C     JUMP - FLAG FOR P OR S TRAVEL TIME DESIRED - INPUT
C            = 1 P TIME
C            = 2 S TIME
C     DELTX - DISTANCE (IN DEGREES) - INPUT
C             (CAN BE FROM 0.0 TO 180.0 DEGREES)
C     DEPHX - DEPTH (IN KILOMETERS) - INPUT
C             (CAN BE FROM 0.0 TO 700.0 KILOMETERS)
C     PASTIM - TRAVEL TIME (IN SECONDS) - OUTPUT
C
C     WRITTEN BY GEOTECH ON CONTRACT TO U.S. GOVERNMENT
C
C     JANUARY 2, 1968  DALLAS
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     SEPTEMBER 3, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION DEPTH(5),DPTH(40),TPTH(40,5),TSTH(40,5)
      DIMENSION P1(40),P2(40),P3(40),P4(40),P5(40),S1(40),S2(40),
     * S3(40),S4(40),S5(40)
      DIMENSION TABLX(4),TABLY(4)
C
      EQUIVALENCE (TPTH(1,1),P1(1)),(TPTH(1,2),P2(1)),(TPTH(1,3),P3(1))
      EQUIVALENCE (TPTH(1,4),P4(1)),(TPTH(1,5),P5(1))
      EQUIVALENCE (TSTH(1,1),S1(1)),(TSTH(1,2),S2(1)),(TSTH(1,3),S3(1))
      EQUIVALENCE (TSTH(1,4),S4(1)),(TSTH(1,5),S5(1))
C
      DATA DEPTH/0.0,100.0,300.0,600.0,700.0/
C
      DATA DPTH/
     *       0.0 ,       1.0 ,       3.0 ,       6.0 ,       8.0 ,
     *      10.0 ,      12.0 ,      14.0 ,      16.0 ,      18.0 ,
     *      20.0 ,      22.0 ,      24.0 ,      26.0 ,      28.0 ,
     *      30.0 ,      35.0 ,      40.0 ,      45.0 ,      50.0 ,
     *      55.0 ,      60.0 ,      65.0 ,      70.0 ,      75.0 ,
     *      80.0 ,      85.0 ,      90.0 ,      95.0 ,     100.0 ,
     *     105.0 ,     110.0 ,     120.0 ,     130.0 ,     140.0 ,
     *     150.0 ,     160.0 ,     170.0 ,     180.0 ,     181.0 /
C
      DATA P1/
     *       0.00 ,     19.17 ,     48.78 ,     90.01 ,    117.47 ,
     *     144.90 ,    172.27 ,    199.59 ,    226.38 ,    251.58 ,
     *     274.10 ,    295.71 ,    316.31 ,    334.51 ,    352.50 ,
     *     370.27 ,    413.98 ,    456.30 ,    496.97 ,    535.89 ,
     *     572.99 ,    608.29 ,    641.77 ,    673.43 ,    703.26 ,
```

```
     *     731.23 ,    757.28 ,    781.35 ,    804.38 ,    826.77 ,
     *     848.96 ,    871.16 ,    915.54 ,    959.93 ,   1004.32 ,
     *    1186.77 ,   1200.43 ,   1209.15 ,   1212.12 ,   1212.45 /
C
      DATA P2/
     *      13.84 ,     20.39 ,     45.71 ,     86.24 ,    113.42 ,
     *     140.62 ,    167.61 ,    193.84 ,    219.17 ,    242.77 ,
     *     264.56 ,    285.96 ,    305.31 ,    323.45 ,    341.36 ,
     *     359.07 ,    402.62 ,    444.75 ,    485.22 ,    523.93 ,
     *     560.85 ,    595.97 ,    629.29 ,    660.79 ,    690.48 ,
     *     718.31 ,    744.23 ,    768.19 ,    791.18 ,    813.54 ,
     *     835.73 ,    857.93 ,    902.32 ,    946.71 ,    991.10 ,
     *    1173.00 ,   1186.64 ,   1195.32 ,   1198.28 ,   1959.00 /
C
      DATA P3/
     *      37.97 ,     40.34 ,     55.71 ,     89.26 ,    113.52 ,
     *     138.07 ,    162.47 ,    185.98 ,    207.93 ,    229.57 ,
     *     250.81 ,    269.90 ,    288.05 ,    305.99 ,    323.73 ,
     *     341.31 ,    384.43 ,    426.06 ,    466.02 ,    504.25 ,
     *     540.72 ,    575.42 ,    608.34 ,    639.48 ,    668.81 ,
     *     696.31 ,    721.93 ,    745.65 ,    768.57 ,    790.86 ,
     *     813.06 ,    835.25 ,    879.64 ,    924.03 ,    968.42 ,
     *    1149.05 ,   1162.60 ,   1171.21 ,   1174.15 ,   1174.45 /
C
      DATA P4/
     *      70.07 ,     71.14 ,     79.07 ,    101.05 ,    119.23 ,
     *     138.66 ,    158.64 ,    178.78 ,    197.24 ,    215.51 ,
     *     233.63 ,    251.54 ,    269.27 ,    286.87 ,    304.29 ,
     *     321.52 ,    363.61 ,    404.18 ,    443.14 ,    480.41 ,
     *     516.00 ,    549.89 ,    582.07 ,    612.51 ,    641.20 ,
     *     668.09 ,    693.14 ,    716.51 ,    739.26 ,    761.48 ,
     *     783.67 ,    805.87 ,    850.26 ,    894.65 ,    939.04 ,
     *    1117.27 ,   1130.67 ,   1139.16 ,   1142.05 ,   1142.40 /
C
      DATA P5/
     *      79.70 ,     80.58 ,     87.18 ,    106.23 ,    122.48 ,
     *     140.09 ,    158.18 ,    176.34 ,    194.44 ,    212.42 ,
     *     230.24 ,    247.92 ,    265.48 ,    282.87 ,    300.08 ,
     *     317.09 ,    358.66 ,    398.73 ,    437.23 ,    474.10 ,
     *     509.32 ,    542.87 ,    574.74 ,    604.90 ,    633.33 ,
     *     659.98 ,    684.77 ,    708.05 ,    730.74 ,    752.94 ,
     *     775.13 ,    797.33 ,    841.72 ,   1073.00 ,   1092.00 ,
     *    1108.00 ,   1122.00 ,   1131.00 ,   1134.00 ,   1134.30 /
C
      DATA S1/
     *       0.00 ,     33.09 ,     86.47 ,    160.59 ,    209.91 ,
     *     259.10 ,    308.14 ,    356.99 ,    405.62 ,    454.06 ,
     *     500.86 ,    540.49 ,    575.41 ,    607.31 ,    638.86 ,
     *     670.27 ,    747.93 ,    823.79 ,    897.41 ,    968.53 ,
     *    1037.00 ,   1102.74 ,   1165.67 ,   1225.75 ,   1282.90 ,
     *    1337.05 ,   1388.07 ,   1435.78 ,   1450.40 ,   1466.79 ,
     *    1485.00 ,   1503.54 ,   1549.85 ,   1581.40 ,   1601.37 ,
     *    1616.10 ,   1627.22 ,   1634.09 ,   1636.41 ,   1637.00 /
C
      DATA S2/
     *      24.46 ,     36.12 ,     81.46 ,    154.14 ,    202.86 ,
     *     251.52 ,    300.08 ,    348.51 ,    396.81 ,    444.65 ,
     *     484.69 ,    523.46 ,    555.92 ,    587.61 ,    619.11 ,
```

```
     *     650.46 ,    727.92 ,    803.51 ,    876.84 ,    947.66 ,
     *    1015.84 ,   1081.29 ,   1143.95 ,   1203.76 ,   1260.66 ,
     *    1314.56 ,   1365.33 ,   1412.81 ,   1425.00 ,   1442.75 ,
     *    1465.00 ,   1479.39 ,   1525.60 ,   1557.08 ,   1576.96 ,
     *    1591.67 ,   1602.78 ,   1609.64 ,   1611.95 ,   1612.20 /
C
      DATA S3/
     *      68.43 ,     72.73 ,    100.58 ,    161.65 ,    206.06 ,
     *     251.25 ,    296.46 ,    340.89 ,    381.27 ,    420.94 ,
     *     458.95 ,    491.50 ,    523.27 ,    554.79 ,    586.17 ,
     *     617.39 ,    694.31 ,    769.24 ,    841.87 ,    911.98 ,
     *     979.46 ,   1044.24 ,   1106.25 ,   1165.44 ,   1221.74 ,
     *    1275.07 ,   1325.28 ,   1372.21 ,   1385.00 ,   1399.72 ,
     *    1423.00 ,   1436.14 ,   1482.09 ,   1513.43 ,   1539.05 ,
     *    1547.76 ,   1558.84 ,   1565.67 ,   1567.98 ,   1568.20 /
C
      DATA S4/
     *     127.54 ,    129.49 ,    144.00 ,    184.21 ,    217.50 ,
     *     253.00 ,    289.67 ,    325.72 ,    358.75 ,    391.29 ,
     *     423.11 ,    454.66 ,    486.07 ,    517.32 ,    548.35 ,
     *     579.13 ,    654.79 ,    728.31 ,    799.50 ,    868.20 ,
     *     934.33 ,    997.81 ,   1058.60 ,   1116.62 ,   1171.81 ,
     *    1224.06 ,   1273.23 ,   1319.15 ,   1331.00 ,   1342.32 ,
     *    1365.00 ,   1378.32 ,   1423.83 ,   1454.90 ,   1474.17 ,
     *    1488.78 ,   1499.79 ,   1506.58 ,   1508.87 ,   1509.10 /
C
      DATA S5/
     *     145.04 ,    146.65 ,    158.70 ,    193.46 ,    223.11 ,
     *     255.09 ,    287.75 ,    320.33 ,    352.55 ,    384.25 ,
     *     415.77 ,    447.16 ,    478.40 ,    509.42 ,    540.21 ,
     *     570.72 ,    645.66 ,    718.48 ,    788.98 ,    857.03 ,
     *     922.55 ,    985.48 ,   1045.73 ,   1103.26 ,   1157.98 ,
     *    1209.79 ,   1258.53 ,   1304.03 ,   1347.12 ,   1360.00 ,
     *    1360.00 ,   1372.00 ,   1408.00 ,   1439.00 ,   1458.00 ,
     *    1473.00 ,   1484.00 ,   1491.00 ,   1493.00 ,   1493.30 /
C
      PASTIM=0.0
      DELT=DELTX
      DEPH=DEPHX
      IF(DELT.LT.0.0)DELT=0.0
      IF(DELT.GT.180.0)DELT=180.0
      IF(DEPH.LT.0.0)DEPH=0.0
      IF(DEPH.GT.700.0)DEPH=700.0
C
C     FIND INDEX ID OF TABLE DISTANCE LARGER OR EQUAL TO DELT
C
      DO 10 I=3,39
      ID=I
      IF(DELT.LE.DPTH(ID))GO TO 20
   10 CONTINUE
C
C     FIND INDEX IH OF TABLE DEPTH LARGER OR EQUAL TO DEPH
C
   20 DO 30 I=2,5
      IH=I
      IF(DEPH.LE.DEPTH(IH))GO TO 40
   30 CONTINUE
C
```

```
      IH=5
C
   40 TABLX(1)=DPTH(ID-2)
      TABLX(2)=DPTH(ID-1)
      TABLX(3)=DPTH(ID)
      TABLX(4)=DPTH(ID+1)
      XX=DELT
      GO TO (60,70,9000),JUMP
C
C     SET FOR P TABLE FOR TABLE DEPTH ABOVE DESIRED VALUE
C
   60 TABLY(1)=TPTH(ID-2,IH-1)
      TABLY(2)=TPTH(ID-1,IH-1)
      TABLY(3)=TPTH(ID,IH-1)
      TABLY(4)=TPTH(ID+1,IH-1)
      GO TO 80
C
C     SET UP S TABLE FOR TABLE DEPTH ABOVE DESIRED VALUE
C
   70 TABLY(1)=TSTH(ID-2,IH-1)
      TABLY(2)=TSTH(ID-1,IH-1)
      TABLY(3)=TSTH(ID,IH-1)
      TABLY(4)=TSTH(ID+1,IH-1)
C
C     INTERPOLATE TO OBTAIN TRAVEL TIME FOR CORRECT DISTANCE
C     WITH DEPTH ABOVE DESIRED VALUE
C
   80 CALL LAGRAN(XX,TABLX,TABLY,TT1)
C
      GO TO (100,110,9000),JUMP
C
C     SET FOR P TABLE FOR TABLE AT OR BELOW DESIRED VALUE
C
  100 TABLY(1)=TPTH(ID-2,IH)
      TABLY(2)=TPTH(ID-1,IH)
      TABLY(3)=TPTH(ID,IH)
      TABLY(4)=TPTH(ID+1,IH)
      GO TO 120
C
C     SET FOR S TABLE FOR TABLE DEPTH AT OR BELOW DESIRED VALUE
C
  110 TABLY(1)=TSTH(ID-2,IH)
      TABLY(2)=TSTH(ID-1,IH)
      TABLY(3)=TSTH(ID,IH)
      TABLY(4)=TSTH(ID+1,IH)
C
C     INTERPOLATE TO OBTAIN TRAVEL TIME FOR CORRECT DISTANCE
C     WITH DEPTH AT OR BELOW DESIRED VALUE
C
  120 CALL LAGRAN(XX,TABLX,TABLY,TT2)
C
C     PERFORM LINEAR INTERPOLATION TO OBTAIN TIME FOR DESIRED
C     DISTANCE AND DEPTH
C
      PASTIM=(TT2-TT1)*(DEPH-DEPTH(IH-1))/(DEPTH(IH)-DEPTH(IH-1))
      PASTIM=PASTIM+TT1
C
 9000 RETURN
```

```
END
```

```
      FUNCTION PHASOR(B,A)
C
C*************************************************************************
C
C     COMPUTE ANGLE FROM X-AXIS COUNTER-CLOCKWISE
C
C     B - FOURIER SINE COEFFICIENT - INPUT
C     A - FOURIER COSINE COEFFICIENT - INPUT
C     PHASOR - 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
      IF(ABS(A).GT.10.0E-20)GO TO 50
C
      IF(B)30,20,40
   20 ANGLE=0.0
      GO TO 500
   30 ANGLE=270.0
      GO TO 500
   40 ANGLE=90.0
      GO TO 500
C
   50 ANGLE=ATAN(B/A)*RD2DG
C
      IF(A.LT.0.0)GO TO 80
      IF(B)70,20,500
   70 ANGLE=360.0+ANGLE
      GO TO 500
   80 IF(B.NE.0.0)GO TO 100
      ANGLE=180.0
      GO TO 500
  100 ANGLE=180.0+ANGLE
C
  500 PHASOR=ANGLE
C
 9000 RETURN
      END
```

```
      SUBROUTINE PHILTR(X,ISAMP,SAMR,E0,E1,E2,E3,IERR)
C
C*********************************************************************
C
C     DESIGN ZERO PHASE BANDPASS FILTER AND FILTER TIME SERIES
C
C     X - TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C             (MUST BE AT LEAST 256 SAMPLES)
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     E0 - LOW FREQUENCY CUTOFF - INPUT AND OUTPUT
C     E1 - LOW FREQUENCY BEGIN TAPER (E1 > E0) - INPUT
C     E2 - HIGH FREQUENCY BEGIN TAPER (E2 < E3) - INPUT
C     E3 - HIGH FREQUENCY CUTOFF - INPUT
C     IERR - ERROR STATUS FLAG - OUTPUT
C            = 0 NO ERROR
C            = 1 ERROR OCCURRED - TOO FEW SAMPLES
C            = 2 ERROR OCCURRED - E0 TOO SMALL
C
C     NOVEMBER 17, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
      COMMON/IN/BX
C
      DIMENSION BX(1),TRACE(1),WORK(1),TEMP(1),X(1)
C
      EQUIVALENCE (BX(1),WORK(1)),(BX(20001),TEMP(1))
      EQUIVALENCE (BX(40001),TRACE(1))
C
      IERR=0
      DT=1.0/SAMR
      IF(E1.LT.E0)E1=E0/0.8
      IF(E2.GT.E3)E2=E3*0.8
C
C     SET NUMBER OF POINTS IN THE FILTER = NPF
C
      NPF=SAMR/E0
      IF(ISAMP.LT.NPF)NPF=ISAMP/2+1
      IF(NPF.GT.3601)NPF=3601
      IF(ISAMP.LT.65)GO TO 7000
C
C     FD - INCREMENT IN FREQUENCY
C     FR - NYQUIST FREQUENCY
C
      FD=SAMR/NPF
      FR=SAMR/2.0
      IF(FD.GT.(E0+0.0001))GO TO 7300
C
C     COMPUTE NUMBER OF FREQUENCY VALUES = NF
C
      NHAF=NPF/2
      NF=1+NHAF
      FTEST=E0+E1
      F=0.0
C
```

```
C     SET UP THE FILTER IN THE FREQUENCY DOMAIN IN WORK ARRAY
C
      DO 1000 I=1,NF
      IF(FTEST.LE.0.0)GO TO 400
      IF((E0-F).LT.0.0)GO TO 300
  200 WORK(I)=0.0
      GO TO 700
  300 IF((E1-F).LT.0.0)GO TO 400
      WORK(I)=(1.0+COS(PI*(F-E1)/(E0-E1)))/2.0
      GO TO 700
  400 IF((E2-F).LT.0.0)GO TO 500
      WORK(I)=1.0
      GO TO 700
  500 IF((E3-F).LT.0.0)GO TO 200
      WORK(I)=(1.0-COS(PI*(F-E3)/(E2-E3)))/2.0
  700 F=F+FD
 1000 CONTINUE
C
C     FORM SYMMETRICAL FILTER IN TIME DOMAIN IN TEMP ARRAY
C
      LF=(NPF-1)/2
      Y=-DT*FLOAT(LF)
C
      DO 2000 I=1,NPF
      TEMP(I)=WORK(1)
      F=FD
C
      DO 1500 J=2,NF
      TEMP(I)=TEMP(I)+2.0*COS(TWOPI*F*Y)*WORK(J)
      F=F+FD
 1500 CONTINUE
C
      Y=Y+DT
      TEMP(I)=TEMP(I)/NPF
 2000 CONTINUE
C
C     REMOVE MEAN OF TIME SERIES
C
      IDEG=0
C
      CALL METR(X,ISAMP,IDEG)
C
C     ZERO FIRST PART OF TRACE ARRAY (1/2 LENGTH OF FILTER)
C
      DO 3000 I=1,NHAF
      TRACE(I)=0.0
 3000 CONTINUE
C
C     STORE TIME SERIES IN TRACE ARRAY FOLLOWING ZERO SEGMENT
C
      DO 3200 J=1,ISAMP
      TRACE(NHAF+J)=X(J)
 3200 CONTINUE
C
C     ZERO END OF TRACE ARRAY
C
      K=NHAF+ISAMP+1
      L=K+NHAF-1
```

```
C
      DO 3500 I=K,L
      TRACE(I)=0.0
 3500 CONTINUE
C
C     FILTER INPUT TRACE BY CONVOLVING WITH FILTER
C
      ZERO=TEMP(NHAF+1)
C
      DO 4000 J=1,ISAMP
      L=J-1
      K=NPF+J
      SUM=0.0
C
      DO 3800 I=1,NHAF
      SUM=SUM+TEMP(I)*(TRACE(I+L)+TRACE(K-I))
 3800 CONTINUE
C
      WORK(J)=SUM+ZERO*TRACE(NHAF+J)
 4000 CONTINUE
C
      DO 5000 J=1,ISAMP
      X(J)=WORK(J)
 5000 CONTINUE
C
      GO TO 9000
C
 7000 IERR=1
      GO TO 9000
C
 7300 IERR=2
      E0=FD
C
 9000 RETURN
      END
```

```
      SUBROUTINE PLSPEC(C,PH,NH2,N2LEN,FMN,FMX,DF,SAMR,IPOW,IFFT)
C
C*********************************************************************
C
C     PLOT SPECTRAL INFORMATION
C
C     C - AMPLITUDE OR POWER SPECTRA - INPUT
C     PH - PHASE (IN DEGREES) - INPUT
C     NH2 - NUMBER OF REAL OR IMAGINARY VALUES - INPUT
C     N2LEN - NUMBER OF POINTS FOR THE POWER OF 2 - INPUT
C     FMN - LOWEST FREQUENCY VALUE TO LIST - INPUT OR OUTPUT
C           = 0 SET TO SAMR/N2LEN
C     FMX - HIGHEST FREQUENCY VALUE TO LIST - INPUT OR OUTPUT
C           = 0 SET TO SAMR/2
C     DF - INCREMENT IN FREQUENCY - INPUT
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     IPOW - SPECTRA OPTION - INPUT
C            = 2 POWER SPECTRA
C            = 3 AMPLITUDE AND PHASE SPECTRA
C            = 9 FREQUENCY RESPONSE
C     IFFT - FREQUENCY/PERIOD FLAG - INPUT
C            = 0 PLOT FREQUENCY VALUES
C            = 1 PLOT PERIOD VALUES
C
C     WRITTEN BY ROBERT P. MASSE
C
C     DECEMBER 6, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      EQUIVALENCE (BX(30001),FORT(1))
C
      DIMENSION FORT(1),BX(1),C(1),PH(1)
C
      IF(IPOW.EQ.9)GO TO 500
      IF(IPOW.NE.2.AND.IPOW.NE.3)GO TO 9000
C
C     CALCULATE ALL FREQUENCIES/PERIODS
C
      EHZ=SAMR/2.0
      ELZ=SAMR/N2LEN
      IF(FMX.LE.0.0.OR.FMX.GT.EHZ)FMX=EHZ
      IF(FMN.LE.0.0.OR.FMN.LT.ELZ)FMN=ELZ
      DF=ELZ
      F=ELZ
      GO TO 700
C
  500 MIN=1
      NP=NH2
      F=FMN
C
      DO 600 I=1,NP
      FORT(I)=F
      IF(IFFT.EQ.1)FORT(I)=1.0/FORT(I)
      F=F+DF
```

```
  600 CONTINUE
C
      GO TO 2000
C
  700 NP=0
      MIN=0
C
      DO 1000 I=2,NH2
      IF(F.LT.FMN)GO TO 800
      IF(F.GT.FMX)GO TO 800
      IF(MIN.EQ.0)MIN=I
      NP=NP+1
      FORT(I)=F
      IF(IFFT.EQ.1)FORT(I)=1.0/FORT(I)
  800 F=F+DF
 1000 CONTINUE
C
C     DETERMINE MINIMUM AND MAXIMUM OF SPECTRA AMPLITUDE OR POWER
C
 2000 CALL MNMX(C(MIN),NP,FPM,IPM,FPX,IPX,FNM,INM,FNX,INX)
C
      XF=FPM
      XMAX=FPX
C
C     DETERMINE MINIMUM AND MAXIMUM OF FREQUENCY OR PERIOD RANGE
C
      CALL MNMX(FORT(MIN),NP,FPM,IPM,FPX,IPX,FNM,INM,FNX,INX)
C
      YF=FPM
      YMAX=FPX
      NCY=0
C
C     SET XF TO BE A MULTIPLE OF 10
C
      CALL CYCLES(XF,XMAX,NCY,ZFT,XF,ZMAXT,XMAX,ZFCYT)
C
C     SET YF TO BE A MULTIPLE OF 10
C
      CALL CYCLES(YF,YMAX,NCY,ZFT,YF,ZMAXT,YMAX,ZFCYT)
C
C     PLOT POWER OR AMPLITUDE SPECTRA
C
      WRITE(IWR2,3000)
 3000 FORMAT(/,10X,'PLOTTING POWER OR AMPLITUDE SPECTRA',/)
C
      CALL XYPSPC(C(MIN),FORT(MIN),NP,XF,XMAX,YF,YMAX)
C
      IF(IPOW.NE.3.AND.IPOW.NE.9)GO TO 9000
C
C     DETERMINE MINIMUM AND MAXIMUM OF SPECTRA PHASE
C
      CALL MNMX(PH(MIN),NP,FPM,IPM,FPX,IPX,FNM,INM,FNX,INX)
C
      XF=FPM
      XMAX=FPX
C
C     SET XF TO BE A MULTIPLE OF 10
C
```

```
      CALL CYCLES(XF,XMAX,NCY,ZFT,XF,ZMAXT,XMAX,ZFCYT)
C
C     PLOT PHASE
C
      WRITE(IWR2,4000)
 4000 FORMAT(/,10X,'PLOTTING PHASE SPECTRA',/)
C
      CALL XYPSPC(PH(MIN),FORT(MIN),NP,XF,XMAX,YF,YMAX)
C
 9000 RETURN
      END
```

```
      SUBROUTINE POW(C,N2LEN,NH2)
C
C*************************************************************************
C
C     CALCULATE POWER FROM AMPLITUDE SPECTRA
C
C     C - AMPLITUDE - INPUT
C       - POWER - OUTPUT
C     N2LEN - NUMBER OF POINTS FOR THE POWER OF 2 - INPUT
C     NH2 - NUMBER OF REAL OR IMAGINARY VALUES - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 5, 1993  C0LORADO
C
C*************************************************************************
C
      DIMENSION C(1)
C
      DO 100 I=2,NH2
      C(I)=C(I)*C(I)
      C(I)=C(I)/N2LEN
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE POWBEM(X,ISAMP)
C
C*****************************************************************************
C
C     COMPUTE RECTIFIED TIME SERIES
C
C     X - DATA TIME SERIES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN TIME SERIES - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 7, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION X(1)
C
      DO 100 J=1,ISAMP
      X(J)=ABS(X(J))
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE POWER2(ISAMP,LENG,N2PWR,N2LEN,NH2,N2P1,NTOT,NZ)
C
C*****************************************************************************
C
C     COMPUTE PARAMETERS OF POWER OF 2
C
C     ISAMP - NUMBER OF SAMPLES - INPUT
C     LENG - TOTAL NUMBER OF LOCATIONS AVAILABLE - INPUT
C     N2PWR - POWER OF 2 - OUTPUT
C     N2LEN - NUMBER OF POINTS FOR THE POWER OF 2 - OUTPUT
C     NH2 - NUMBER OF REAL COMPONENTS = N2LEN/2+1 - OUTPUT
C     N2P1 - N2LEN+1 - OUTPUT
C     NTOT - TOTAL LENGTH NEEDED FOR TRANSFORM = N2LEN+2 - OUTPUT
C     NZ - NUMBER OF ZEROS TO BE ADDED - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 22, 1993  COLORADO
C
C*****************************************************************************
C
      N2PWR=1
      N2LEN=1
      IF(LENG.LE.0)GO TO 9000
   10 N2LEN=N2LEN*2
      IF(ISAMP.LE.N2LEN)GO TO 40
      N2PWR=N2PWR+1
      GO TO 10
   30 N2PWR=N2PWR-1
      N2LEN=N2LEN/2
   40 IF(LENG.LT.(N2LEN+2))GO TO 30
      NH2=N2LEN/2+1
      N2P1=N2LEN+1
      NTOT=N2LEN+2
      NZ=NTOT-ISAMP
C
 9000 RETURN
      END
```

```
      SUBROUTINE PREVT
C
C*********************************************************************
C
C     LISTS EVENT INFORMATION STORED 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(IWR,200)EXNAME
      WRITE(IWR2,200)EXNAME
  200 FORMAT(/,10X,'EVENT - ',A20,/)
C
      WRITE(IWR,400)
      WRITE(IWR2,400)
  400 FORMAT(/,10X,'EVENT HYPOCENTER',/)
      WRITE(IWR,500)EXLAT,EXLON,EXDEPH
      WRITE(IWR2,500)EXLAT,EXLON,EXDEPH
  500 FORMAT(10X,'LAT = ',F10.3,5X,'LONG = ',F10.3,5X,'DEPTH = ',
     * F8.2,/)
C
      WRITE(IWR,600)
      WRITE(IWR2,600)
  600 FORMAT(//,10X,'EVENT ORIGIN TIME',/)
      WRITE(IWR,700)IEXYR,IEXMO,IEXDY,IEXHR,IEXMN,EXSEC
      WRITE(IWR2,700)IEXYR,IEXMO,IEXDY,IEXHR,IEXMN,EXSEC
  700 FORMAT(10X,5I6,4X,F6.3,/)
C
      WRITE(IWR,800)
      WRITE(IWR2,800)
  800 FORMAT(//,10X,'EVENT MAGNITUDE',/)
      WRITE(IWR,900)EXMAG1,IEXM1,EXMAG2,IEXM2
      WRITE(IWR2,900)EXMAG1,IEXM1,EXMAG2,IEXM2
  900 FORMAT(10X,F10.3,2X,A4,5X,F10.3,2X,A4,/)
C
      WRITE(IWR,1000)EXCOMT
      WRITE(IWR2,1000)EXCOMT
 1000 FORMAT(//,10X,'EVENT COMMENTS - ',A20,/)
C
      WRITE(IWR,1100)
      WRITE(IWR2,1100)
 1100 FORMAT(/,10X,'EVENT FLAGS'/)
      WRITE(IWR,1200)(I,EXFLAG(I),I=1,10)
      WRITE(IWR2,1200)(I,EXFLAG(I),I=1,10)
 1200 FORMAT(10X,I3,3X,F10.3)
      WRITE(IWR,1400)
```

```
      WRITE(IWR2,1400)
 1400 FORMAT(//)
C
 9000 RETURN
      END
```

```
      SUBROUTINE PRHEAD(ICHAN)
C
C*********************************************************************
C
C     LIST CHANNEL HEADER INFORMATION
C
C     ICHAN - CHANNEL NUMBER TO WRITE HEADER - INPUT
C             = 0 LIST INFORMATION FOR ALL CHANNELS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 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/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
      IF(ICHAN.GT.0)GO TO 800
      K=1
      MORE=1
      GO TO 850
C
  800 K=ICHAN
      MORE=0
C
  850 WRITE(IWR,870)K
      WRITE(IWR2,870)K
  870 FORMAT(//,10X,'CHANNEL NUMBER = ',I5,/)
      WRITE(IWR,900)
      WRITE(IWR2,900)
  900 FORMAT(/,10X,'STATION',3X,'CHANNEL',3X,'LATITUDE',4X,
     * 'LONGITUDE',5X,'ELEV',/)
      WRITE(IWR,1000)JCODE(K),(JCHN(I,K),I=1,4),RLAT(K),RLON(K),
     * RELEV(K)
      WRITE(IWR2,1000)JCODE(K),(JCHN(I,K),I=1,4),RLAT(K),RLON(K),
     * RELEV(K)
 1000 FORMAT(12X,A4,6X,4A1,2X,F10.4,2X,F10.4,4X,F7.3,/)
C
      WRITE(IWR,1200)JYEAR(K),JDOFY(K),JHOUR(K),JMIN(K),RSEC(K)
      WRITE(IWR2,1200)JYEAR(K),JDOFY(K),JHOUR(K),JMIN(K),RSEC(K)
 1200 FORMAT(/,10X,'DATA START TIME: ',I5,3X,I4,3X,I3,3X,I3,3X,F6.3,/)
C
      WRITE(IWR,1300)RSAMR(K),JFLAG(K)
      WRITE(IWR2,1300)RSAMR(K),JFLAG(K)
 1300 FORMAT(/,10X,'RSAMR = ',F6.2,5X,'JFLAG = ',I8,/)
C
      IF(MORE.EQ.0)GO TO 9000
      K=K+1
```

```
      IF(K.LE.NACHLS)GO TO 850
C
 9000 RETURN
      END
```

```
      SUBROUTINE PRPLT(A,B,IFIR,ILAS,INC,IOPT)
C
C*********************************************************************
C
C     MAKE PRINTER PLOT OF ONE OR TWO DATA ARRAYS
C
C     A - DATA ARRAY - INPUT
C     B - DATA ARRAY - INPUT
C     IFIR - INDEX OF FIRST DATA POINT TO PLOT - INPUT
C            = 0 SET TO 1
C     ILAS - INDEX OF LAST DATA POINT TO PLOT - INPUT
C     INC - INCREMENT FOR SELECTING DATA POINTS TO PLOT - INPUT
C           = 0 SET TO 1
C     IOPT - OPTION TO PLOT ONE OR TWO DATA ARRAYS - INPUT
C            = 0 PLOT BOTH ARRAYS
C            = 1 PLOT ONLY A ARRAY
C
C     OCTOBER 13, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION A(1),B(1),PLOTR(61)
C
      LOGICAL*1 BLANK,BAR,STAR,DOT,PLOTR
C
      DATA BLANK,BAR,STAR,DOT/' ','I','*','+'/
C
      WRITE(IWR,10)
      WRITE(IWR2,10)
   10 FORMAT(/,10X,'FIRST(*)',5X,'SECOND(+)',//)
      WRITE(IWR,20)
      WRITE(IWR2,20)
   20 FORMAT(9X,'I---------.---------.---------0---------.---------.',
     * '---------I')
C
      IF(IFIR.LE.0)IFIR=1
      IF(INC.LE.0)INC=1
      IF(ILAS.LE.IFIR)GO TO 9000
C
C     DETERMINE MAXIMUM ABSOLUTE VALUE IN A AND B ARRAYS
C
      AMAX=0.0
C
      DO 100 I=IFIR,ILAS,INC
      TEMP=ABS(A(I))
      IF(TEMP.GT.AMAX)AMAX=TEMP
      IF(IOPT.EQ.1)GO TO 100
      TEMP=ABS(B(I))
      IF(TEMP.GT.AMAX)AMAX=TEMP
  100 CONTINUE
C
C     SCALE ALL DATA POINTS TO A MAXIMUM OF 30.0 AND SHIFT BY 31
C     TO MAKE POSITIVE NUMBERS FOR PRINTING
C
      DO 1000 I=IFIR,ILAS,INC
      AI=A(I)*30.0/AMAX
```

```
      IA=AI
      IAS=IA+31
      IF(IOPT.EQ.1)GO TO 200
      BI=B(I)*30.0/AMAX
      IB=BI
      IBS=IB+31
C
C     MAKE PLOTTING ARRAY INITIALLY BLANK
C
  200 DO 300 J=1,61
      PLOTR(J)=BLANK
  300 CONTINUE
C
C     SET SYMBOLS A = *, B = +, CO-LOCATED = +
C
      PLOTR(1)=BAR
      PLOTR(31)=BAR
      PLOTR(61)=BAR
      PLOTR(IAS)=STAR
      IF(IOPT.EQ.1)GO TO 400
      PLOTR(IBS)=DOT
C
  400 WRITE(IWR,500)I,PLOTR
      WRITE(IWR2,500)I,PLOTR
  500 FORMAT(4X,I4,1X,61A1)
 1000 CONTINUE
C
      WRITE(IWR,1200)
      WRITE(IWR2,1200)
 1200 FORMAT(9X,'I---------.---------.---------0---------.',
     * '---------.---------I')
      WRITE(IWR,1300)AMAX
      WRITE(IWR2,1300)AMAX
 1300 FORMAT(/,10X,'MAXIMUM VALUE IS ',F12.3,//)
C
 9000 RETURN
      END
```

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

```
      IF(ILAS.LE.0.AND.ICHAN1.GT.0)ILAS=JSAMP(ICHAN1)
      IF(ILAS.LE.0.AND.ICHAN2.GT.0)ILAS=JSAMP(ICHAN2)
      IF(INC.LE.0)INC=1
      WRITE(IWR,700)IFIR,ILAS,INC
      WRITE(IWR2,700)IFIR,ILAS,INC
  700 FORMAT(/,10X,'IFIR = ',I6,5X,'ILAS = ',I6,5X,'INC = ',I3,/)
C
      IF(ILAS.LE.0)GO TO 7000
C
 1400 INDEX1=(ICHAN1-1)*LENG+IFIR
      IF(ICHAN1.LE.0)GO TO 3000
      IF(ICHAN2.LE.0)GO TO 2000
      INDEX2=(ICHAN2-1)*LENG+IFIR
C
      CALL PRPLT(DATA(INDEX1),DATA(INDEX2),IFIR,ILAS,INC,IOPT)
C
      GO TO 9000
C
 2000 CALL PRPLT(DATA(INDEX1),DATA(INDEX1),IFIR,ILAS,INC,IOPT)
C
      GO TO 9000
C
 3000 IF(ICHAN2.LE.0)GO TO 4000
C
      CALL PRPLT(BX,DATA(INDEX2),IFIR,ILAS,INC,IOPT)
C
      GO TO 9000
C
 4000 CALL PRPLT(BX,BX,IFIR,ILAS,INC,IOPT)
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE PRSPC1(Z,C,PH,NH2,N2LEN,N2P1,FMN,FMX,SAMR,IPOW)
C
C*********************************************************************
C
C     LIST SPECTRAL INFORMATION
C
C     Z - SPECTRAL VALUES - INPUT
C     C - AMPLITUDE OR POWER SPECTRA - INPUT
C     PH - PHASE (IN DEGREES) - INPUT
C     NH2 - NUMBER OF REAL OR IMAGINARY VALUES - INPUT
C     N2LEN - NUMBER OF POINTS FOR THE POWER OF 2 - INPUT
C     N2P1 - N2LEN+1 - INPUT
C     FMN - LOWEST FREQUENCY VALUE TO LIST - INPUT OR OUTPUT
C          = 0 SET TO SAMR/N2LEN
C     FMX - HIGHEST FREQUENCY VALUE TO LIST - INPUT OR OUTPUT
C          = 0 SET TO SAMR/2
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     IPOW - SPECTRA OPTION - INPUT
C            = 0 REAL AND IMAGINARY COMPONENTS
C            = 2 POWER SPECTRA
C            = 3 AMPLITUDE AND PHASE SPECTRA
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 1, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION Z(1),C(1),PH(1)
C
      EHZ=SAMR/2.0
      ELZ=SAMR/N2LEN
      IF(FMX.LE.0.0.OR.FMX.GT.EHZ)FMX=EHZ
      IF(FMN.LE.0.0.OR.FMN.LT.ELZ)FMN=ELZ
      DF=ELZ
      F=ELZ
C
      IF(IPOW.NE.0)GO TO 1000
C
C     LIST REAL AND IMAGINARY COMPONENTS
C
      WRITE(IWR,500)
      WRITE(IWR2,500)
  500 FORMAT(//,10X,'FREQ',4X,'PERIOD',5X,'REAL',8X,'IMAG',/)
      GO TO 6000
C
 1000 IF(IPOW.NE.2)GO TO 2000
C
C     LIST POWER
C
      WRITE(IWR,1100)
      WRITE(IWR2,1100)
 1100 FORMAT(//,10X,'FREQ',10X,'PERIOD',10X,'POWER',/)
      GO TO 4000
C
C     LIST AMPLITUDE AND PHASE
```

```
C
 2000 WRITE(IWR,2100)
      WRITE(IWR2,2100)
 2100 FORMAT(//,10X,'FREQ',4X,'PERIOD',5X,'AMP',8X,'PHASE',/)
C
 4000 DO 5000 I=2,NH2
      T=1.0/F
      IF(F.LT.FMN)GO TO 4800
      IF(F.GT.FMX)GO TO 4800
      IF(IPOW.NE.2)GO TO 4500
      WRITE(IWR,4100)F,T,C(I)
      WRITE(IWR2,4100)F,T,C(I)
 4100 FORMAT(6X,F8.3,7X,F8.2,5X,E12.3)
      GO TO 4800
 4500 WRITE(IWR,4600)F,T,C(I),PH(I)
      WRITE(IWR2,4600)F,T,C(I),PH(I)
 4600 FORMAT(6X,F8.3,1X,F8.2,2F12.3)
 4800 F=F+DF
 5000 CONTINUE
C
      GO TO 6700
C
 6000 DO 6500 I=2,NH2
      T=1.0/F
      K=I+NH2
      IF(F.LT.FMN)GO TO 6300
      IF(F.GT.FMX)GO TO 6300
      WRITE(IWR,6100)F,T,Z(I),Z(K)
      WRITE(IWR2,6100)F,T,Z(I),Z(K)
 6100 FORMAT(6X,F8.3,1X,F8.2,F12.3,F12.3)
 6300 F=F+DF
 6500 CONTINUE
C
 6700 WRITE(IWR,6800)
      WRITE(IWR2,6800)
 6800 FORMAT(//)
C
 9000 RETURN
      END
```

```
      SUBROUTINE PRSPC3(N2LEN,NH2,SAMR,A,B,C,PH)
C
C*********************************************************************
C
C     LIST FOURIER TRANSFORM RESULTS FOR FOURT
C
C     N2LEN - NUMBER OF SAMPLES - INPUT
C     NH2 - NUMBER OF REAL COMPONENTS = N2LEN/2+1 - INPUT
C     SAMR - SAMPLING RATE (IN SAMPLES/S) - INPUT
C     A - FOURIER COSINE COEFFICIENT - INPUT
C     B - FOURIER SINE COEFFICIENT - INPUT
C     C - FOURIER POWER - INPUT
C     PH - PHASE ANGLE (IN DEGREES) - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 5, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION A(1),B(1),C(1),PH(1)
C
      WRITE(IWR,100)
      WRITE(IWR2,100)
  100 FORMAT(//,10X,'FREQ',4X,'PERIOD',5X,'REAL',8X,'IMAG',9X,
     *'AMP',7X,'PHASE',/)
C
      IF(N2LEN.EQ.0)GO TO 9000
      DF=SAMR/N2LEN
      F=0.0
C
      DO 500 I=2,NH2
      F=F+DF
      T=1.0/F
      WRITE(IWR,200)F,T,A(I),B(I),C(I),PH(I)
      WRITE(IWR2,200)F,T,A(I),B(I),C(I),PH(I)
  200 FORMAT(6X,F8.3,1X,F8.2,F12.3,F12.3,F12.3,F10.3)
  500 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE RAVAGE(X,ISAMP,ISM)
C
C*********************************************************************
C
C     SMOOTHING ROUTINE USING A RUNNING AVERAGE OVER ISM POINTS
C
C     X - DATA VALUES - INPUT AND OUTPUT
C     ISAMP - NUMBER OF DATA VALUES - INPUT
C     ISM - NUMBER OF POINTS TO SMOOTH OVER - INPUT
C           EVEN SET TO ISM+1
C           < 5 SET TO 5
C           > 25 SET TO 25
C
C     WRITTEN BY ROBERT SHUMWAY
C
C     MODIFIED BY A. R. HILL AND ROBERT P. MASSE
C
C     OCTOBER 28, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1),Y(14)
C
      KK=ISM/2+1
      FIV=0.5
      IF(KK.GT.13)KK=13
      IF(KK.LT.3)KK=3
      I=1
      NN=KK+1
      N2=2*KK-1
C
      X(1)=X(1)*FIV
      X(ISAMP)=X(ISAMP)*FIV
C
      DO 1000 J=1,ISAMP
      IF(J.GT.KK)GO TO 200
      N1=2*J-1
      SUM=0.0
C
      DO 100 L=1,N1
      SUM=SUM+X(L)
  100 CONTINUE
C
      AVE=SUM/(FLOAT(N1)-FIV)
      GO TO 900
C
  200 IF((J-ISAMP+KK).GT.0)GO TO 400
      SUM=SUM+X(J+KK-1)-X(J-KK)
      AVE=SUM/FLOAT(N2)
C
  250 X(I)=Y(1)
      Y(KK+1)=AVE
      I=I+1
C
      DO 300 L=1,NN
      Y(L)=Y(L+1)
  300 CONTINUE
C
```

```
      GO TO 1000
C
  400 N1=2*(ISAMP-J+1)-1
      SUM=0.0
C
      DO 500 L=1,N1
      SUM=SUM+X(ISAMP-L+1)
  500 CONTINUE
C
      AVE=SUM/(FLOAT(N1)-FIV)
      GO TO 250
C
  900 Y(J)=AVE
 1000 CONTINUE
C
      N1=ISAMP-KK+1
      I=1
C
      DO 1100 L=N1,ISAMP
      X(L)=Y(I)
      I=I+1
 1100 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE RAXEPL(TF,TMAX,SCALT,TI,TTI,FSDIST,DMAX,SCALD,DI,DDI,
     * IPF1,IPF2,FAC,AXTIT,AYTIT,AMTIT)
C
C*********************************************************************
C
C     PLOT AXES AND LABEL FOR TIME SERIES PLOT
C
C     TF - PLOT ORIGIN TIME VALUE - INPUT
C     TMAX - MAXIMUM TIME VALUE - INPUT
C     SCALT - INCHES/SECOND TRACE PLOTTING INCREMENT - INPUT
C     TI - INCREMENT IN TIME VALUE FOR TICKS - INPUT
C     TTI - INCREMENT IN TIME VALUE FOR LABELED TICKS - INPUT
C           (MUST BE INTEGER MULTIPLE OF TI)
C     FSDIST - PLOT ORIGIN DISTANCE VALUE - INPUT
C     DMAX - MAXIMUM DISTANCE VALUE - INPUT
C            (PLOT SIZE MUST BE 1.5 INCHES LARGER THAN THIS)
C     SCALD - INCHES/KILOMETER DISTANCE PLOTTING INCREMENT - INPUT
C     DI - INCREMENT IN DISTANCE VALUE FOR TICKS - INPUT
C     DDI - INCREMENT IN DISTANCE VALUE FOR LABELED TICKS - INPUT
C     IPF1 - PLOT FLAG FOR NUMBERS ON TIME AXIS - INPUT
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     IPF2 - PLOT FLAG FOR NUMBERS ON DISTANCE AXIS - INPUT
C            = 0 PLOT NUMBERS
C            = 1 OMIT NUMBERS
C     FAC - PLOT SCALE FACTOR - INPUT
C     AXTIT - TITLE FOR TIME AXIS (20 CHARACTERS) - INPUT
C     AYTIT - TITLE FOR DISTANCE AXIS (20 CHARACTERS) - INPUT
C     AMTIT - MAIN PLOT TITLE (20 CHARACTERS) - INPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 27, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/PCKF/ICHECK,IPLFL
C
      CHARACTER*1 BCD
      CHARACTER*20 AXTIT,AYTIT,AMTIT
C
      DATA BCD/'-'/
C
      IF(IPLFL.EQ.0)GO TO 50
C
C     CLEAR SCREEN PLOT FRAME AND ADVANCE DEFERRED FRAME
C
      CALL ADVPLT
C
   50 CALL FACTOR(FAC)
C
      CALL PLOT(1.0,0.0,-3)
C
      IEND=0
      IPASS=1
      IPF=IPF1
      XLT=0.50
      XA=XLT
```

```
      YLT=(DMAX-FSDIST)*SCALD+1.50
      YA=YLT
      YYLT=YLT
      FTM=TF
      FCNT=TF+TTI
C
      CALL PLOT(XA,YA,3)
C
C---------------------------------------------------------------------
C
C     PLOT TIME AXIS
C
      IF(IPF.NE.0)GO TO 100
C
C     PLOT TIME NUMBER VALUE
C
      XXA=XLT-0.08
      YYA=YLT+0.70
C
      CALL NUMBER(XXA,YYA,0.14,FTM,270.0,'F4.0')
C
      XA=XLT
      YA=YLT
C
      CALL PLOT(XA,YA,3)
C
C     PLOT TIME TICK MARK
C
  100 XXA=XLT+0.24
      YYA=YYLT-0.08
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,90.0,1)
C
      CALL PLOT(XA,YA,3)
C
  200 FTM=FTM+TI
      XA=XA+TI*SCALT
      IF(FTM.LT.TMAX)GO TO 250
      IF(FTM.EQ.TMAX)GO TO 230
      IEND=1
      XA=XLT+(TMAX-TF)*SCALT
      GO TO 250
  230 IEND=2
C
  250 CALL PLOT(XA,YA,2)
C
      IF(IEND.NE.0)GO TO 350
C
C     PLOT TIME TICK MARK
C
      XXA=XA+0.24
      YYA=YYLT-0.08
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,90.0,1)
C
      CALL PLOT(XA,YA,3)
C
      IF(IPF.NE.0)GO TO 300
```

```
      IF(FTM.NE.FCNT)GO TO 300
C
C     PLOT TIME NUMBER VALUE
C
      FCNT=FCNT+TTI
      XXA=XA-0.08
      YYA=YLT+0.70
C
      CALL NUMBER(XXA,YYA,0.14,FTM,270.0,'F4.0')
C
      CALL PLOT(XA,YA,3)
C
  300 IF(FTM.LT.TMAX)GO TO 200
  350 IF(IPASS.EQ.2)GO TO 400
C
C-----------------------------------------------------------------------
C
C     PLOT TOP DISTANCE AXIS
C
      XTOP=XA
      YA=YLT
C
      CALL PLOT(XA,YA,3)
C
      YYLT=1.50
      YA=YYLT
C
      CALL PLOT(XA,YA,2)
C
C-----------------------------------------------------------------------
C
C     PLOT RIGHT TIME AXIS
C
      XA=XLT
      FTM=TF
      FCNT=TF+TTI
      IEND=0
      IPASS=2
      IPF=1
C
      CALL PLOT(XA,YA,3)
C
      GO TO 100
C
C-----------------------------------------------------------------------
C
C     PLOT BOTTOM DISTANCE AXIS
C
  400 XA=XLT
      YA=YLT
      IEND=0
      FTM=FSDIST
      FCNT=FSDIST+DDI
C
      CALL PLOT(XA,YA,3)
C
C     PLOT DISTANCE TICK MARK
C
```

```
      XXA=XLT-0.08
      YYA=YLT-0.24
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,0.0,1)
C
      CALL PLOT(XA,YA,3)
C
      IF(IPF2.NE.0)GO TO 600
C
C     PLOT DISTANCE NUMBER VALUE
C
      XXA=XA-0.40
      YYA=YA+0.25
C
      CALL NUMBER(XXA,YYA,0.14,FTM,270.0,'F5.0')
C
      CALL PLOT(XA,YA,3)
C
  600 FTM=FTM+DI
      YA=YA-DI*SCALD
      IF(YA.GT.YYLT)GO TO 700
      IF(YA.EQ.YYLT)GO TO 650
      IEND=1
      YA=YYLT
      GO TO 700
  650 IEND=2
C
  700 CALL PLOT(XA,YA,2)
C
      IF(IEND.NE.0)GO TO 2000
C
C     PLOT DISTANCE TICK MARK
C
      XXA=XA-0.08
      YYA=YA-0.24
C
      CALL SYMBOL(XXA,YYA,0.48,BCD,0.0,1)
C
      CALL PLOT(XA,YA,3)
C
      IF(IPF2.NE.0)GO TO 1000
      IF(FTM.NE.FCNT)GO TO 1000
C
C     PLOT DISTANCE NUMBER VALUE
C
      FCNT=FCNT+DDI
      XXA=XA-0.40
      YYA=YA+0.25
C
      CALL NUMBER(XXA,YYA,0.14,FTM,270.0,'F5.0')
C
      CALL PLOT(XA,YA,3)
C
 1000 IF(FTM.LT.DMAX)GO TO 600
C
C-----------------------------------------------------------------------
C
 2000 CALL PLOT(0.0,YA,3)
```

```
C
      CALL PLOT(0.0,0.0,3)
C
C     PLOT TIME TITLE
C
      XXA=XLT+((TMAX-TF)*SCALT)/2.0-2.0
      YYA=YLT+0.80
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,0.28,AXTIT,0.0,20)
C
C     PLOT DISTANCE TITLE
C
      XXA=XLT-0.80
      YYA=YLT-((DMAX-FSDIST)*SCALD)/2.0+2.0
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,0.28,AYTIT,270.0,20)
C
C     PLOT MAIN TITLE
C
      XXA=XTOP+0.40
C
      CALL PLOT(XXA,YYA,3)
C
      CALL SYMBOL(XXA,YYA,0.28,AMTIT,270.0,20)
C
      CALL PLOT(0.0,0.0,3)
C
      CALL FACTOR(1.0)
C
      RETURN
      END
```

```
      SUBROUTINE RAYGIJ(ARGH,ARGK,WVNO,TTT,CSQ,MM,KWRITE,MMAX,NTRLS,LCT)
C
C*****************************************************************************
C
C     FORM THE ELEMENTS OF DUNKIN SUBMATRICES
C
C     ARGH - P WAVE ARGUMENT - INPUT
C     ARGK - S WAVE ARGUMENT - INPUT
C     WVNO - WAVE NUMBER - INPUT
C     TTT - PERIOD (IN SECONDS) - INPUT
C     CSQ - PHASE VELOCITY SQUARED - INPUT
C     MM - CURRENT LAYER NUMBER - INPUT
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     MMAX - NUMBER OF LAYERS IN MODEL - INPUT
C     NRTLS - NUMBER OF TRIALS TO FIND A ROOT - INPUT AND OUTPUT
C             = 1000 ERROR ENCOUNTERED
C     LCT - NUMBER OF SUBDIVISIONS TO THE CURRENT LAYER - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     JANUARY 14, 1994  COLORADO
C
C*****************************************************************************
C
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
C
      DIMENSION V(1000,4),G(6,6)
      DIMENSION BX(1)
C
      EQUIVALENCE (BX(40001),V(1,1)),(BX(50601),G(1,1))
C
      REAL*8 G,GAM,GAMSQ,GAM1,GAM1SQ,THSQ,TKSQ,TH,TK,PM,QM,EP,EQ,EQ1,
     * EQ2,ARGH,ARGK,EEP,SP,CP,SQ,CQ,ETOPM,ETOQM,ETOQM2,EPM,CPCQ,SPSQ,
     * CPSQ,SPCQ,THKSQ,RPD,FLP
C
      M=MM
      PFV=0.50
      GAM=-2.0*V(M,3)*V(M,3)/CSQ
      GAMSQ=GAM*GAM
      GAM1=GAM+1.0
      GAM1SQ=GAM1*GAM1
      THSQ=DABS(ARGH)
      TKSQ=DABS(ARGK)
      TH=DSQRT(THSQ)
      TK=DSQRT(TKSQ)
      RLK=V(M,4)*CSQ
      RLKSQ=RLK*RLK
C
C     IS THIS HALF-SPACE?
C
      IF(M.GE.MMAX)GO TO 2000
C
C     SET UP PARAMETERS
C
```

```
      PM=WVNO*TH*V(M,1)
      QM=WVNO*TK*V(M,1)
      IP=0
      IQ=0
      LCT=1
      EP=2.0*PM
      EQ=2.0*QM
      EQ1=QM-PM
      EQ2=-(QM+PM)
      EPM=1.0
C
C     IS THIS FIRST LAYER?
C
      IF(M.LE.1)GO TO 700
C
C     CHECK SIZE OF EXPONENTS OF COMPUTED LAYER PARAMETERS
C
      IF(ARGH.LE.0.0)GO TO 800
      IF(EP.LE.2.40)GO TO 50
      IP=1
   50 IF(ARGK.LE.0.0)GO TO 100
      IF(EQ.LE.2.40)GO TO 100
      IQ=1
  100 IF(IP.EQ.0)GO TO 200
      EEP=EP
      IF(IQ.EQ.0)GO TO 300
      IF(EP.GE.EQ)GO TO 300
      EEP=EQ
      GO TO 300
  200 IF(IQ.EQ.0)GO TO 700
C
C     SUBDIVIDE LAYER
C
  300 DO 400 I=2,99
      LCT=I
      FLCT=LCT
C
C     CHECK MAGNITUDE OF EXPONENT AFTER LAYER SUBDIVISION
C
      IF((EEP/FLCT).LE.2.40)GO TO 500
  400 CONTINUE
C
      GO TO 7000
C
C     RECOMPUTE LAYER PARAMETERS FOR SUBDIVIDED LAYERS
C
  500 QM=QM/FLCT
      PM=PM/FLCT
      EP=2.0*PM
      EQ1=QM-PM
      EQ2=-(QM+PM)
C
      IF(KWRITE.NE.2)GO TO 700
      WRITE(IWR2,600)M,LCT,TTT
  600 FORMAT(10X,'LAYER ',I3,' SUBDIVIDED INTO ',I3,' LAYERS',/,
     * 10X,'FOR PERIOD OF ',F7.3,' SECONDS')
C
C     COMPUTATION FOR LAYER PARAMETERS
```

```
C
  700 IF(V(M,3).LE.0.0)GO TO 1000
      IF(ARGH.GT.0.0)GO TO 1100
  800 SP=DSIN(PM)
      CP=DCOS(PM)
      THSQ=-THSQ
  900 SQ=DSIN(QM)
      CQ=DCOS(QM)
      TKSQ=-TKSQ
      GO TO 1300
C
 1000 ETOPM=DEXP(PM)
      SP=PFV*(ETOPM-1.0/ETOPM)
      CP=PFV*(ETOPM+1.0/ETOPM)
C
C     CHECK FOR FLUID SURFACE LAYER
C
      IF(V(M,3).LE.0.0)GO TO 1400
      GO TO 900
C
C     SET UP EXPONENTIAL TERMS
C
 1100 IF(ARGK.LE.0.0)GO TO 1000
 1200 ETOPM=DEXP(-EP)
      SP=PFV-PFV*ETOPM
      CP=PFV+PFV*ETOPM
      ETOQM=DEXP(EQ1)
      ETOQM2=DEXP(EQ2)
      SQ=PFV*(ETOQM-ETOQM2)
      CQ=PFV*(ETOQM+ETOQM2)
      EPM=ETOPM
 1300 SP=SP/TH
      SQ=SQ/TK
      CPCQ=CP*CQ
      SPSQ=SP*SQ
      CPSQ=CP*SQ
      SPCQ=SP*CQ
      THKSQ=THSQ*TKSQ
C
C     COMPUTE SUBMATRIX ELEMENTS FOR SOLID LAYERS
C
      G(1,1)=-2.0*GAM*GAM1*EPM+(2.0*GAMSQ+2.0*GAM+1.0)*CPCQ-(GAM1SQ+
     * GAMSQ*THKSQ)*SPSQ
      G(6,6)=G(1,1)
      G(2,1)=-RLK*(GAMSQ*TKSQ*CPSQ-GAM1SQ*SPCQ)
      G(6,5)=G(2,1)
      G(3,1)=-RLK*(GAM*GAM1*(2.0*GAM+1.0)*(EPM-CPCQ)+(GAM1*GAM1SQ+
     * GAM*GAMSQ*THKSQ)*SPSQ)
      G(4,1)=G(3,1)
      G(6,3)=-G(3,1)
      G(6,4)=-G(3,1)
      G(5,1)=-RLK*(GAM1SQ*CPSQ-GAMSQ*THSQ*SPCQ)
      G(6,2)=G(5,1)
      G(6,1)=-RLKSQ*(2.0*GAMSQ*GAM1SQ*(EPM-CPCQ)+(GAM1SQ*GAM1SQ+
     * GAMSQ*GAMSQ*THKSQ)*SPSQ)
      G(1,2)=(-CPSQ+THSQ*SPCQ)/RLK
      G(5,6)=G(1,2)
      G(1,3)=((2.0*GAM+1.0)*(EPM-CPCQ)+(GAM1+GAM*THKSQ)*SPSQ)/RLK
```

```
      G(1,4)=G(1,3)
      G(3,6)=-G(1,3)
      G(4,6)=-G(1,3)
      G(1,5)=(SPCQ-TKSQ*CPSQ)/RLK
      G(2,6)=G(1,5)
      G(1,6)=-(2.0*(EPM-CPCQ)+(1.0+THKSQ)*SPSQ)/RLKSQ
      G(2,2)=CPCQ
      G(5,5)=CPCQ
      G(2,3)=-(GAM1*SPCQ-GAM*TKSQ*CPSQ)
      G(2,4)=G(2,3)
      G(3,5)=-G(2,3)
      G(4,5)=-G(2,3)
      G(2,5)=-TKSQ*SPSQ
      G(3,2)=-(GAM1*CPSQ-GAM*THSQ*SPCQ)
      G(4,2)=G(3,2)
      G(5,3)=-G(3,2)
      G(5,4)=-G(3,2)
      G(3,3)=EPM+2.0*GAM*GAM1*(EPM-CPCQ)+(GAM1SQ+GAMSQ*THKSQ)*SPSQ
      G(4,4)=G(3,3)
      G(3,4)=G(3,3)-EPM
      G(4,3)=G(3,4)
      G(5,2)=-THSQ*SPSQ
      GO TO 9000
C
C     COMPUTE SUBMATRIX ELEMENTS FOR FLUID SURFACE LAYER
C
 1400 DO 1500 I=1,6
      DO 1500 J=1,6
      G(I,J)=0.0
 1500 CONTINUE
C
      DO 1600 I=1,6
      G(1,I)=CP
      G(2,I)=RLK*SP/TH
 1600 CONTINUE
C
      GO TO 9000
C
C     COMPUTATIONS FOR HALF-SPACE
C
 2000 RPD=TH*TK
      ALPSQ=V(M,2)*V(M,2)
      FLP=-4.0*V(M,4)*ALPSQ*GAM
      G(1,2)=1.0/(FLP*TK)
      G(1,5)=-1.0/(FLP*TH)
C
      IF(ARGH.LT.0.0)GO TO 2100
      RPD=-RPD
 2100 G(1,1)=CSQ*(GAM+GAM1SQ/(RPD*GAM))/(4.0*ALPSQ)
      G(1,3)=-(1.0+GAM1/(GAM*RPD))/(4.0*V(M,4)*ALPSQ)
      G(1,4)=G(1,3)
      G(1,6)=-(1.0+1.0/RPD)/(FLP*RLK)
      GO TO 9000
C
 7000 WRITE(IWR2,7100)M,TTT,PM,QM
 7100 FORMAT(/,10X,'COULD NOT SUBDIVIDE LAYER ',I3,' INTO LESS THAN ',
     * '99 LAYERS',/,10X,'FOR PERIOD OF 'F7.3,' SECONDS WITH ',/,
     * 10X,'PM = ',D15.5,' AND  QM = ',D15.5,/)
```

```
C
      MM=MMAX
      NTRLS=1000
C
 9000 RETURN
      END
```

```
      SUBROUTINE RCOOL(X,N,ISIGN)
C
C*********************************************************************
C
C     COMPUTE FOURIER TRANSFORM
C
C     X - REAL ARRAY, DIMENSIONED AT LEAST (2**N)+2 - INPUT AND OUTPUT
C     N - POWER OF 2 REPRESENTING NUMBER OF SAMPLES - INPUT
C     ISIGN - TRANSFORM FLAG - INPUT
C             = 1 FOR DIRECT TRANSFORM
C             =-1 FOR INVERSE TRANSFORM
C
C     AFTER DIRECT TRANSFORM:
C         COSINE TRANSFORMS ARE IN FIRST (2**N)/2+1 LOCATIONS OF X
C         SINE TRANSFORMS ARE IN SECOND (2**N)/2+1 LOCATIONS OF X
C
C     JULY 16, 1993  COLORADO
C
C*********************************************************************
C
      DIMENSION X(1)
C
      ISIG=-ISIGN
      NN=2**N
      M=N-1
      L=NN/2
      LLL=L/2
      LL=LLL+1
      L1=L+1
      L2=L+2
      NN1=NN+1
      NN2=NN+2
      FL=L
      F=3.14159265/FL
      IP2=1
      LENGTH=2
      CM=1.0
      IF(ISIG.GE.0)GO TO 110
   10 CM=0.5
C
C     SEPARATE ODD AND EVEN-NUMBERED SAMPLES
C
   20 DO 80 JANE=1,M
      IF(ISIG.GE.0)GO TO 40
      IF(JANE.LE.1)GO TO 30
      IP2=IP2+IP2
   30 LENGTH=NN/IP2
      GO TO 50
   40 LENGTH=LENGTH+LENGTH
      IP2=NN/LENGTH
   50 LHALF=LENGTH*0.5
      JST=1
C
      DO 70 JACK=1,IP2
      IST=JST+1
      J=JST+LHALF
      IEND=J-1
C
```

```
      DO 60 I=IST,IEND,2
      HOLDX=X(I)
      X(I)=X(J)
      X(J)=HOLDX
      J=J+2
   60 CONTINUE
C
      JST=JST+LENGTH
   70 CONTINUE
C
   80 CONTINUE
C
      IF(ISIG.GE.0)GO TO 9000
      J=NN
C
      DO 100 I=1,L
      X(J+1)=X(J)
      J=J-1
  100 CONTINUE
C
C     TRANSFORM SEPARATED DATA
C
      CALL CLFORM(X(1),X(L2),M,-1)
C
      HOLDX=X(1)
      HOLDY=X(L2)
      X(1)=HOLDX+HOLDY
      X(L2)=0.0
      X(NN2)=0.0
      X(L1)=HOLDX-HOLDY
  110 J=L
      JJ=NN1
      II=L2+1
C
      DO 120 I=2,LL
      FL=I-1
      ARG=FL*F
      HCOS=COS(ARG)*ISIG
      HSIN=-SIN(ARG)
      AR=CM*(X(I)+X(J))
      AI=CM*(X(II)-X(JJ))
      BR=CM*(X(I)-X(J))
      BI=-CM*(X(II)+X(JJ))
      PR=BI*HSIN+BR*HCOS
      PI=BR*HSIN-BI*HCOS
      X(I)=AR-PI
      X(II)=AI+PR
      X(J)=AR+PI
      X(JJ)=-AI+PR
      J=J-1
      II=II+1
      JJ=JJ-1
  120 CONTINUE
C
      IF(ISIG.LT.0)GO TO 9000
      X(L2)=X(1)-X(L1)
      X(1)=X(1)+X(L1)
C
```

```
      DO 150 I=L1,NN
      X(I)=X(I+1)
  150 CONTINUE
C
      CALL CLFORM(X(1),X(L1),M,1)
C
      GO TO 20
C
 9000 RETURN
      END
```

```
      SUBROUTINE RDDATA(LUN,TD)
C
C*************************************************************************
C
C     READS A DATA RECORD IN RETRV FORMAT
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ IN - INPUT
C     TD - READ FLAG - OUTPUT
C          = 0 READ SUCCESSFUL
C          =-1 END OF FILE READ
C
C     WRITTEN BY MADELEINE ZIRBES
C
C     SEPTEMBER 15, 1980  COLORADO
C
C*************************************************************************
C
      COMMON/TRDATA/PTS,BUFF
C
      DIMENSION BUFF(500)
C
      INTEGER*4 PTS,TD
C
      TD=0
      READ(LUN,END=20)PTS,(BUFF(I),I=1,PTS)
C
C     PTS - NUMBER OF POINTS IN PHYSICAL RECORD - UP TO 500
C     BUFF - ACTUAL DATA OF CHANNEL READ
C
      GO TO 9000
C
C     END OF FILE READ
C
   20 TD=-1
C
 9000 RETURN
      END
```

```
      SUBROUTINE RDFIL(LUN,ICHAN,IRFL1)
C
C*********************************************************************
C
C     READ A KEYS FILE INTO A CHANNEL
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     ICHAN - CHANNEL NUMBER TO FILL WITH DATA - INPUT
C     IRFL1 - READ FLAG - OUTPUT
C             = 0 SUCCESSFUL DATA READ
C             = 1 END OF FILE READ
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/EVT/EXNAME,EXLAT,EXLON,EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,
     * IEXMN,EXSEC,EXMAG1,IEXM1,EXMAG2,IEXM2,EXCOMT,EXFLAG
      COMMON/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),EXFLAG(10)
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 EXNAME,EXCOMT
C
      DATA NY/'y'/
C
      IRFL1=0
      K=ICHAN
C
  200 READ(LUN,END=7000)JCODE(K),(JCHN(I,K),I=1,4),RLAT(K),RLON(K),
     * RELEV(K),JYEAR(K),JDOFY(K),JHOUR(K),JMIN(K),RSEC(K),
     * RSAMR(K),RA0(K),JNP(K),(RPOLES(I,K),I=1,30),JNZ(K),
     * (RZEROS(I,K),I=1,20),JSAMP(K),JFLAG(K),EXNAME,EXLAT,EXLON,
     * EXDEPH,IEXYR,IEXMO,IEXDY,IEXHR,IEXMN,EXSEC,EXMAG1,IEXM1,
     * EXMAG2,IEXM2,EXCOMT,EXFLAG
C
      ISAMP=JSAMP(K)
C
      READ(LUN,END=7000)(BX(J),J=1,ISAMP)
C
      WRITE(IWR2,500)JCODE(K),(JCHN(I,K),I=1,4)
  500 FORMAT(/,10X,'STATION-CHANNEL IS ',A4,3X,4A1,/)
C
      WRITE(IWR2,550)
  550 FORMAT(/,'++ INPUT: ','KEEP THIS STATION-CHANNEL (y/n)?',/)
```

```
      READ(IRE,600,END=9000)KY
  600 FORMAT(A1)
C
C       KY - y OR n
C
      IF(NY.NE.KY)GO TO 200
C
 2000 IF(ISAMP.LE.LENG)GO TO 2500
C
      WRITE(IWR2,2050)
 2050 FORMAT(/,10X,'***** WARNING *****',/)
      WRITE(IWR2,2100)ISAMP,LENG,LENG
 2100 FORMAT(/,10X,'NUMBER OF SAMPLES IN FILE = ',I6,/,10X,'LENG = ',
     * I6,/,'ONLY ',I6,' SAMPLES RETAINED',/)
C
      ISAMP=LENG
C
 2500 INDEX=(ICHAN-1)*LENG
C
      DO 3000 J=1,ISAMP
      IN=INDEX+J
      DATA(IN)=BX(J)
 3000 CONTINUE
C
      GO TO 9000
C
 7000 IRFL1=-1
C
 9000 RETURN
      END
```

```
      SUBROUTINE RDHEAD(LUN,TF)
C
C*********************************************************************
C
C     READS A HEADER RECORD FROM LOGICAL UNIT NUMBER LUN FOR
C     A FILE IN RETRV FORMAT
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     TF - READ RESULT FLAG - OUTPUT
C          = 0 HEADER READ
C          = 1 END CODE READ IN HEADER RECORD
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/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
C
      DATA KUE,KUN,KUD,KBL/'E','N','D',' '/
C
      TF=0
C
      READ(LUN,END=20)CODE(1),CODE(2),CODE(3),CODE(4),CHN(1),CHN(2),
     * CHN(3),CHN(4),LAT,LON,ELEV,SYEAR,SDOFY,SHOUR,SMIN,SSECS,RATE,
     * A0,NP,(POLES(I),I=1,NP),NZ,(ZEROES(I),I=1,NZ),NFLAG(1),
     * NFLAG(2),NFLAG(3)
C
C     CODE -  STATION CODE (4 CHARACTER STRING)
C     CHN - INSTRUMENT CODE (4 CHARACTER STRING)
C     LAT - STATION GEOGRAPHIC LATITUDE (IN DEGREES)
C           - = S
C     LON - STATION GEOGRAPHIC LONGITUDE (IN DEGREES)
C           - = W
C     ELEV - STATION ELEVATION (IN FEET)
C     SYEAR - START TIME OF DATA - YEAR
C     SDOFY - START TIME OF DATA - DAY OF YEAR
C     SHOUR - START TIME OF DATA - HOUR
C     SMIN - START TIME OF DATA - MINUTE
C     SSECS - START TIME OF DATA - SECOND
C     RATE - SAMPLING RATE (IN SAMPLES/S)
C     A0 - NORMALIZATION CONSTANT
C          - = REVERSAL
C     NP - NUMBER OF POLES
```

```
C      POLES - COMPLEX POLES(30)
C      NZ - NUMBER OF ZEROES
C      ZEROES - COMPLEX ZEROES(20)
C      FLAGS - INTEGER NFLAG(3)
C
C      CHECK FOR END IN STATION NAME
C
       IF(.NOT.(CODE(1).EQ.KUE.AND.CODE(2).EQ.KUN.AND.CODE(3).EQ.KUD
      * .AND.CODE(4).EQ.KBL))GO TO 9000
C
C      END CODE READ
C
       TF = 1
       GO TO 9000
C
C      END OF FILE READ
C
   20  TF=-1
C
 9000  RETURN
       END
```

```
      SUBROUTINE READG
C
C*********************************************************************
C
C     GENERAL FILE READ ROUTINE FOR KEYS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 11, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/IO/LUN,LUN2,LUN3,LUN4,LUN5,LUN6,LUN7,LUN8
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/OPTION/IOPS
      COMMON/STAT/ISTAT
C
      DIMENSION ISTAT(11),IOPS(1)
C
      INTEGER*4 RETR
      CHARACTER*40 FILENM
C
      DATA LAST/'last'/,RETR/'retr'/,KEYS/'keys'/,NY/'y'/
C
      IF(NCHLS.EQ.0)NCHLS=1
      ICHAN=0
      LLUN=-1
C
      WRITE(IWR,5)NCHLS,LENG,NACHLS
      WRITE(IWR2,5)NCHLS,LENG,NACHLS
    5 FORMAT(/,10X,'CURRENT DATA STORAGE PARAMETERS ARE:',//,10X,
     * 'NCHLS = ',I4,5X,'LENG = ',I6,5X,'NACHLS = ',I4,/)
C
      WRITE(IWR2,10)
   10 FORMAT(/,'++ INPUT: ','NC',3X,'LE',/)
      READ(IRE,*,END=9000)NC,LE
C
C     NC - NEW NUMBER OF CHANNELS
C          = 0 DO NOT CHANGE
C     LE - NEW TOTAL STORAGE RESERVED FOR EACH CHANNEL
C          = 0 DO NOT CHANGE
C
      IF(NC.GT.0)NCHLS=NC
      IF(LE.GT.0)LENG=LE
      IF((NC*LE).GT.IDEM)GO TO 7000
      IF(NC.GT.0.AND.LE.LE.0)LENG=IDEM/NC
      IF(LE.GT.0.AND.NC.LE.0)NCHLS=IDEM/LE
      IF(NC.EQ.0.AND.LE.EQ.0)GO TO 20
C
      WRITE(IWR,5)NCHLS,LENG,NACHLS
      WRITE(IWR2,5)NCHLS,LENG,NACHLS
C
   20 WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','NEXT FILE TYPE (retr, keys, or last)',/)
      READ(IRE,60,END=9000)ITYPE
   60 FORMAT(A4)
```

```
C
C     ITYPE - NEXT FILE TYPE
C             CHOICES: retr, keys, last
C
      WRITE(IWR2,70)ITYPE
   70 FORMAT(/,10X,'TYPE ',A4,' SELECTED')
C
      IF(ITYPE.EQ.RETR)GO TO 200
      IF(ITYPE.EQ.KEYS)GO TO 600
      IF(ITYPE.EQ.LAST)GO TO 2000
      GO TO 20
C
C     READ FILE IN RETRV FORMAT
C
  200 WRITE(IWR2,210)
  210 FORMAT(/,'++ INPUT: ','FILE UNIT NUMBER FOR INPUT',/,10X,
     * '(MUST BE GREATER THAN 14)',/)
      READ(IRE,*,END=9000)LUN
C
C     LUN - FILE UNIT NUMBER FOR INPUT
C           (MUST BE GREATER THAN 14)
C
      IF(LUN.EQ.0)LUN=LLUN
      IF(LUN.LT.0)GO TO 800
      WRITE(IWR2,230)LUN
  230 FORMAT(/,10X,'SELECTED FILE UNIT NUMBER IS ',I3)
C
      IF(LUN.EQ.LLUN)GO TO 350
      LLUN=LUN
C
C     INITIALIZE FILE READ
C
      CALL INITFL
C
  300 WRITE(IWR2,310)
  310 FORMAT(/,'++ INPUT: ','NEXT FILE NAME FOR INPUT ',/)
      READ(IRE,315,END=9000)FILENM
  315 FORMAT(A40)
C
C     FILENM - FILE NAME FOR INPUT
C
      WRITE(IWR2,320)FILENM
  320 FORMAT(/,10X,'SELECTED FILENAME IS ',A40,//,'++ INPUT: ',
     * 'IS CORRECT (y/n)?',/)
      READ(IRE,330,END=9000)KY
  330 FORMAT(A1)
C
C     KY - y OR n (IS CORRECT?)
C
      IF(NY.NE.KY)GO TO 300
C
  350 OPEN(LUN,FILE=FILENM,STATUS='OLD',FORM='UNFORMATTED',
     * ACCESS='SEQUENTIAL')
      REWIND LUN
C
      IOPS(50)=1
C
C     INPUT CHANNEL INFORMATION
```

```
C
  400 WRITE(IWR2,450)
  450 FORMAT(/,'++ INPUT: ','DESIRED CHANNEL NUMBER TO FILL',/)
      READ(IRE,*,END=9000)KCHAN
C
C     KCHAN - DESIRED CHANNEL NUMBER TO FILL
C             = 0 FOR NEXT CHANNEL
C
      IF(KCHAN.LE.0)KCHAN=ICHAN+1
      ICHAN=KCHAN
      IF(ICHAN.GT.NCHLS)GO TO 950
      WRITE(IWR2,480)ICHAN
  480 FORMAT(/,10X,'CHANNEL IS ',I6,/)
C
      CALL RERETR(LUN,NSAMP,IRFL1,IRFL2,IRFL3,ISTAT)
C
      IF(IRFL1.NE.1)GO TO 900
      IF(IRFL2.LT.0)GO TO 900
      IF(NSAMP.LE.2)GO TO 900
C
C     FILL DATA ARRAY CHANNEL
C
      CALL FILLCH(ICHAN,NSAMP)
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','"more" OR "last" DATA IN THIS FILE',/)
      READ(IRE,60,END=9000)IOPT
C
C     IOPT - more OR last OPTION
C
      IF(IOPT.NE.LAST)GO TO 400
C
      CLOSE(LUN)
      GO TO 20
C
C     READ FILE IN KEYS FORMAT
C
C     INITIALIZE FILE READ
C
  600 LUN=10
  650 WRITE(IWR2,310)
      READ(IRE,315,END=9000)FILENM
C
C     FILENM - FILE NAME FOR INPUT
C
      WRITE(IWR2,320)FILENM
      READ(IRE,330,END=9000)KY
C
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 650
C
      IF(IOPS(51).NE.0)GO TO 670
      OPEN(LUN,FILE=FILENM,STATUS='OLD',FORM='UNFORMATTED',
     * ACCESS='SEQUENTIAL')
  670 REWIND LUN
C
C     INPUT CHANNEL INFORMATION
```

```
C
  700 WRITE(IWR2,450)
      READ(IRE,*,END=9000)KCHAN
C
C     KCHAN - DESIRED CHANNEL NUMBER TO FILL
C             = 0 FOR NEXT CHANNEL
C
      IF(KCHAN.LE.0)KCHAN=ICHAN+1
      ICHAN=KCHAN
      IF(ICHAN.GT.NCHLS)GO TO 950
      WRITE(IWR2,480)ICHAN
C
      CALL RDFIL(LUN,ICHAN,IRFL1)
C
      IF(IRFL1.NE.0)GO TO 900
C
      WRITE(IWR2,500)
      READ(IRE,60,END=9000)IOPT
C
C     IOPT - more OR last OPTION
C
      IF(IOPT.NE.LAST)GO TO 700
      CLOSE (LUN)
      GO TO 20
C
C     ERROR IN UNIT NUMBER
C
  800 WRITE(IWR2,810)LUN
  810 FORMAT(/,10X,'***** ERROR IN UNIT NUMBER FOR FILE *****'
     * ,I4,/)
      GO TO 20
C
  900 WRITE(IWR2,910)LUN
  910 FORMAT(/,10X,'***** READ ERROR OCCURRED FOR FILE *****',I4,/)
      ICHAN=ICHAN-1
      GO TO 20
C
  950 WRITE(IWR2,960)ICHAN,NCHLS
  960 FORMAT(/,10X,'CHANNEL NUMBER ',I4,/,' IS GREATER THAN TOTAL',
     * ' NUMBER OF CHANNELS ',I4,/)
      ICHAN=NCHLS
      GO TO 20
C
 2000 WRITE(IWR2,2050)
 2050 FORMAT(/,10X,'OLD DATA STORAGE PARAMETERS',/)
      WRITE(IWR2,2100)ICHAN,NACHLS,NCHLS
 2100 FORMAT(10X,'ICHAN = ',I4,5X,'NACHLS = ',I4,5X,'NCHLS = ',
     * I4,/)
      IF(ICHAN.GT.NACHLS)NACHLS=ICHAN
      WRITE(IWR2,2200)
 2200 FORMAT(/,10X,'NEW DATA STORAGE PARAMETERS',/)
      WRITE(IWR2,2100)ICHAN,NACHLS,NCHLS
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'ERROR IN INPUT PARAMETERS',/)
C
 9000 RETURN
```

```
      SUBROUTINE RECSEC
C
C*********************************************************************
C
C     PLOT RECORD SECTION
C
C     WRITTEN BY ROBERT P. MASSE
C
C     SEPTEMBER 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/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/PCKF/ICHECK,IPLFL
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1),DIST(100),TT(100),NP(100),TZERO(100),
     * RDT(100),KFP(100),ST(100),PF(100),BX(1),EXFLAG(10)
C
      EQUIVALENCE (BX(1),DIST(1)),(BX(101),TT(1)),(BX(201),NP(1))
      EQUIVALENCE (BX(301),TZERO(1)),(BX(401),RDT(1))
      EQUIVALENCE (BX(501),KFP(1)),(BX(601),ST(1)),(BX(701),PF(1))
C
      COMPLEX RPOLES,RZEROS
      CHARACTER*4 IEXM1,IEXM2
      CHARACTER*20 AXTIT,AYTIT,AMTIT
      CHARACTER*20 EXNAME,EXCOMT
      CHARACTER*80 MSG
C
      DATA ILW/'L   '/,IRW/'R   '/
C
      CMPIN=2.54
      MODE=0
      ELAT=EXLAT
      ELON=EXLON
      EDEPTH=EXDEPH
C
C     READ RECORD SECTION PARAMETERS
C
      WRITE(IWR2,200)
  200 FORMAT(/,'++ INPUT: ','NCH',3X,'RVEL',3X,'FNOISE',/)
      READ(IRE,*,END=9000)NCH,RVEL,FNOISE
C
C     NCH - NUMBER OF CHANNELS TO INCLUDE IN RECORD SECTION
C           (INCLUDE FROM CHANNEL 1 TO NCH)
C           = 0 INCLUDE ALL CHANNELS
C     RVEL - REDUCTION VELOCITY (IN KILOMETERS/S)
C            = 0 REDUCED TIME NOT USED
```

```
C     FNOISE - SECONDS OF NOISE TO INCLUDE BEFORE SIGNAL
C              = 0 SET TO 10.0
C
      IF(NCH.LE.0)NCH=NACHLS
      IF(FNOISE.EQ.0.0)FNOISE=10.0
      WRITE(IWR,300)NCH,RVEL,FNOISE
      WRITE(IWR2,300)NCH,RVEL,FNOISE
  300 FORMAT(/,10X,'NCH = ',I4,5X,'RVEL = ',F6.2,5X,'FNOISE = ',
     * F6.2,/)
C
      WRITE(IWR2,350)
  350 FORMAT(/,'++ INPUT: ','ISC',3X,'SF',/)
      READ(IRE,*,END=9000)ISC,SF
C
C      ISC - TRACE SCALING OPTION FLAG
C            = 0 SCALE INDIVIDUALLY
C            = 1 SCALE ALL TOGETHER
C            = 2 DON'T SCALE
C      SF - MAXIMUM TRACE AMPLITUDE (IN CM)
C           = 0 SET TO 1.0
C
      IF(SF.EQ.0.0)SF=1.0
      WRITE(IWR,400)ISC,SF
      WRITE(IWR2,400)ISC,SF
  400 FORMAT(/,10X,'ISC = ',I4,5X,'SF    = ',F6.2,/)
C
C     READ TIME AXIS PLOT PARAMETERS
C
      WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','TDIZ',3X,'TMAX',3X,'TI',3X,'TTI',/)
      READ(IRE,*,END=9000)TDIZ,TMAX,TI,TTI
C
C     TDIZ - SECONDS/CM TRACE PLOTTING INCREMENT
C            = 0 SET TO 5.0
C     TMAX - MAXIMUM RELATIVE TIME VALUE TO BE PLOTTED (IN SECONDS)
C            = 0 SET TO 60.0
C     TI - INCREMENT IN TIME VALUE FOR TICKS (IN SECONDS)
C          = 0 SET TO 10.0
C     TTI - INCREMENT IN TIME VALUE FOR LABELED TICKS (IN SECONDS)
C           (MUST BE INTEGER MULTIPLE OF TI)
C           = 0 SET TO 20
C
      IF(TDIZ.LE.0.0)TDIZ=5.0
      IF(TMAX.EQ.0.0)TMAX=60.0
      IF(TI.LE.0.0)TI=10.0
      IF(TTI.LE.0.0)TTI=20.0
      WRITE(IWR,600)
      WRITE(IWR2,600)
  600 FORMAT(/,10X,'TIME AXIS PLOT PARAMETERS',/)
      WRITE(IWR,700)TDIZ,TMAX,TI,TTI
      WRITE(IWR2,700)TDIZ,TMAX,TI,TTI
  700 FORMAT(10X,'TDIZ    = ',F7.2,3X,'TMAX = ',F6.2,//,10X,
     * 'TI      = ',F7.2,3X,'TTI  = ',F6.2,/)
C
      SCALT=1.0/(TDIZ*CMPIN)
      TOFSET=10.0
C
C     READ DISTANCE AXIS PARAMETERS
```

```
C
      WRITE(IWR2,1000)
 1000 FORMAT(/,'++ INPUT: ','FSDIST',3X,'DDIZ',3X,'DMAX',3X,'DI',
     * 3X,'DDI',/)
      READ(IRE,*,END=9000)FSDIST,DDIZ,DMAX,DI,DDI
C
C     FSDIST - PLOT ORIGIN FOR DISTANCE
C     DDIZ - DISTANCE PLOTTING INCREMENT (IN KILOMETERS/CM)
C            = 0 SET TO 100.0
C     DMAX - MAXIMUM IN DISTANCE VALUE FOR TICKS
C            = 0 SET TO MAXIMUM FOR 9.0 INCHES
C     DI - INCREMENT IN DISTANCE VALUE FOR TICKS
C          = 0 SET TO 100
C     DDI - INCREMENT IN DISTANCE VALUE FOR LABELED TICKS
C           = 0 SET TO 500
C
      IF(DDIZ.LE.0.0)DDIZ=100.0
      IF(DMAX.EQ.0.0)DMAX=DDIZ*CMPIN*9.0+FSDIST
      IF(DI.EQ.0.0)DI=100.0
      IF(DDI.EQ.0.0)DDI=500.0
      WRITE(IWR,1500)
      WRITE(IWR2,1500)
 1500 FORMAT(/,10X,'DISTANCE AXIS PLOT PARAMETERS',/)
      WRITE(IWR,1600)FSDIST,DDIZ,DMAX,DI,DDI
      WRITE(IWR2,1600)FSDIST,DDIZ,DMAX,DI,DDI
 1600 FORMAT(10X,'FSDIST = ',F7.2,3X,'DDIZ = ',F6.2,3X,'DMAX = ',
     * F8.2,//,10X,'DI     = ',F7.2,3X,'DDI  = ',F6.2,/)
C
      SCALD=1.0/(DDIZ*CMPIN)
      IPF1=0
      IPF2=0
C
C     READ PLOT TITLES AND SCALING
C
      WRITE(IWR2,2000)
 2000 FORMAT(/,'++ INPUT: ','MODEP',3X,'FAC',/)
      READ(IRE,*,END=9000)MODEP,FAC
C
C     MODEP - MODE FOR OPENING PLOT SYSTEM
C             = 0 SET TO 3
C             = 1 IMMEDIATE MODE
C             = 2 DEFERRED MODE
C             = 3 EDIT MODE
C     FAC - PLOT SCALE FACTOR
C           = 0 SET TO 1.0
C
      IF(MODEP.LE.0)MODEP=3
      IF(FAC.EQ.0.0)FAC=1.0
      WRITE(IWR,2100)MODEP,FAC
      WRITE(IWR2,2100)MODEP,FAC
 2100 FORMAT(/,10X,'MODEP   = ',I2,3X,'FAC = ',F7.2,/)
C
      WRITE(IWR2,2300)
 2300 FORMAT(/,'++ INPUT: ','AXTIT',/)
      READ(IRE,2400,END=9000)AXTIT
 2400 FORMAT(A20)
C
      WRITE(IWR2,2500)
```

```
 2500 FORMAT(/,'++ INPUT: ','AYTIT',/)
      READ(IRE,2400,END=9000)AYTIT
C
      WRITE(IWR2,3000)
 3000 FORMAT(/,'++ INPUT: ','AMTIT',/)
      READ(IRE,2400,END=9000)AMTIT
C
C     AXTIT - TIME AXIS TITLE (20 CHARACTERS MAXIMUM)
C     AYTIT - DISTANCE AXIS TITLE (20 CHARACTERS MAXIMUM)
C     AMTIT - MAIN PLOT TITLE (20 CHARACTERS MAXIMUM)
C
C     READ PHASE NAME
C
      WRITE(IWR2,3300)
 3300 FORMAT(/,'++ INPUT: ','NPHAS',/)
      READ(IRE,3500,END=9000)NPHAS
 3500 FORMAT(A4)
C
C     NPHAS - NAME OF DESIRED PHASE FOR TT COMPUTATION
C
C     CALCULATE TRAVEL TIMES AND DISTANCES
C
      UVEL=0.0
      PERIOD=0.0
      NPER=0
      IF(NPHAS.NE.ILW.AND.NPHAS.NE.IRW)GO TO 3550
C
      WRITE(IWR2,3530)
 3530 FORMAT(/,'++ INPUT: ','UVEL',/)
      READ(IRE,*,END=9000)UVEL
C
C     UVEL - GROUP VELOCITY FOR RAYLEIGH OR LOVE WAVES (IN KM/S)
C            = 0 SET TO 4.0
C
      IF(UVEL.LE.0.0)UVEL=4.0
      WRITE(IWR,3540)UVEL
      WRITE(IWR2,3540)UVEL
 3540 FORMAT(/,10X,'UVEL = ',F10.3,/)
C
      NPER=1
C
 3550 WRITE(IWR,3600)
      WRITE(IWR2,3600)
 3600 FORMAT(/,10X,'STATION',10X,'DISTANCE',10X,'TT',/)
C
      IDF=0
C
      DO 4000 I=1,NCH
C
      CALL TTCOMP(RLAT(I),RLON(I),RELEV(I),ELAT,ELON,EDEPTH,
     * NPHAS,DELTA,DIST(I),AZ1,AZ2,TT(I),UVEL,PERIOD,NPER,IDF)
C
      WRITE(IWR,3700)JCODE(I),DIST(I),TT(I)
      WRITE(IWR2,3700)JCODE(I),DIST(I),TT(I)
 3700 FORMAT(12X,A4,11X,F8.2,6X,F8.2)
 4000 CONTINUE
C
C     FORM REDUCTION TERM
```

```
C
      DO 4500 I=1,NCH
      IF(RVEL.GT.0.0)GO TO 4200
      RDT(I)=0.0
      GO TO 4500
 4200 RDT(I)=DIST(I)/RVEL
 4500 CONTINUE
C
C     FIND TF AS SMALLEST REDUCED TT
C
      TF=TT(1)-FNOISE-RDT(1)
      TZERO(1)=TF
      IF(NCH.LE.1)GO TO 4900
C
      DO 4800 I=2,NCH
      TTF=TT(I)-FNOISE-RDT(I)
      TZERO(I)=TTF
      IF(TTF.LT.TF)TF=TTF
 4800 CONTINUE
C
 4900 TF=TF-TOFSET
      ITF=TF/TI
      TF=ITF*TI
      TMAX=TMAX+TF+TOFSET
C
C     PLOT RECORD SECTION AXES
C
      ZLT=(DMAX-FSDIST)*SCALD+1.5
      IF(IPLFL.NE.0)GO TO 5000
      WIN=ZLT+2.0
      IF(WIN.LE.12.5)WIN=12.5
C
      CALL OPNPLT(MODEP,WIN,111)
C
 5000 CALL RAXEPL(TF,TMAX,SCALT,TI,TTI,FSDIST,DMAX,SCALD,DI,DDI,
     * IPF1,IPF2,FAC,AXTIT,AYTIT,AMTIT)
C
      IPLFL=1
C
C     CALCULATE TZERO AND CORRESPONDING SAMPLE NUMBER KFP
C
      CALL TIMC(EOT,IEXHR,IEXMN,EXSEC)
C
      DO 6000 I=1,NCH
      DST=EOT+TT(I)-FNOISE
C
      CALL TIMC(ST(I),JHOUR(I),JMIN(I),RSEC(I))
C
      AST=ST(I)
      TDIF=DST-AST
      IF(ABS(TDIF).LT.43200.0)GO TO 5200
C
C     CORRECT FOR DAY ERROR
C
      IF(TDIF.LT.0.0)THEN
         TDIF=TDIF+86400.0
      ELSE
         TDIF=TDIF-86400.0
```

```
      ENDIF
C
 5200 IF(TDIF.GE.0.0)GO TO 5400
      KFP(I)=1
      TZERO(I)=TZERO(I)-TDIF
      GO TO 5500
C
 5400 KFP(I)=TDIF*RSAMR(I)+1.5
 5500 NP(I)=(TMAX-TZERO(I))*RSAMR(I)
      NPX=JSAMP(I)-KFP(I)+1
      IF(NP(I).GT.NPX)NP(I)=NPX
      IF(NP(I).LT.0)NP(I)=0
 6000 CONTINUE
C
C     SCALE TIME SERIES
C
      DO 6100 I=1,NCH
      PF(I)=1.0
      IF(ISC.EQ.2)GO TO 6100
      ISAMP=NP(I)
      IF(ISAMP.LE.0)GO TO 6100
      INDEX=(I-1)*LENG+KFP(I)
C
      CALL SFS(DATA(INDEX),ISAMP,SFX)
C
      PF(I)=SFX
 6100 CONTINUE
C
      IF(ISC.NE.1)GO TO 6250
      IF(NCH.LE.1)GO TO 6250
      PFMAX=PF(1)
C
      DO 6150 I=2,NCH
      IF(PFMAX.LT.PF(I))PFMAX=PF(I)
 6150 CONTINUE
C
      DO 6200 I=1,NCH
      PF(I)=PFMAX
 6200 CONTINUE
C
 6250 DO 6300 I=1,NCH
      PF(I)=-PF(I)*CMPIN*SCALD/SF
 6300 CONTINUE
C
      DO 6500 I=1,NCH
      ISAMP=NP(I)
      INDEX=(I-1)*LENG+KFP(I)
C
      DO 6350 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)/PF(I)
 6350 CONTINUE
C
C     POSITION DATA
C
      DO 6400 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)+DIST(I)
```

```
 6400 CONTINUE
C
 6500 CONTINUE
C
C     PLOT TIME SERIES
C
      DO 6600 I=1,NCH
      IF(NP(I).LE.0)GO TO 6550
      IF((DIST(I)-SF/(CMPIN*SCALD)).LE.FSDIST)GO TO 6600
      IF((DIST(I)+SF/(CMPIN*SCALD)).GE.DMAX)GO TO 6600
      INDEX=(I-1)*LENG+KFP(I)
C
      CALL SPLDAT(TF,TZERO(I),NP(I),SCALT,FSDIST,DMAX,SCALD,
     * RSAMR(I),DATA(INDEX),FAC)
C
      GO TO 6600
C
 6550 WRITE(IWR2,6570)I
 6570 FORMAT(/,10X,'CHANNEL ',I4,' NOT PLOTTED - NO SAMPLES IN FRAME',
     * /)
C
 6600 CONTINUE
C
C     REMOVE POSITIONING TERM
C
      DO 6900 I=1,NCH
      ISAMP=NP(I)
      INDEX=(I-1)*LENG+KFP(I)
C
      DO 6700 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)-DIST(I)
 6700 CONTINUE
C
C     REMOVE SCALING FACTOR
C
      DO 6800 J=1,ISAMP
      IN=INDEX+J-1
      DATA(IN)=DATA(IN)*PF(I)
 6800 CONTINUE
C
 6900 CONTINUE
C
C     RETURN CONTROL TO SCREEN
C
      CALL HLDPLT(5.0,1.0,MODE,MSG)
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'RECORD SECTION SUCCESSFULLY PLOTTED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE RECTIF
C
C*********************************************************************
C
C     FORM A RECTIFIED POWER TIME SERIES
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/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 RECTIFY PARAMETERS
C
      WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL
C             = 0 RECTIFY ALL CHANNELS
C
      WRITE(IWR,300)ICHAN
      WRITE(IWR2,300)ICHAN
  300 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      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     RECTIFY CHANNELS
C
 2000 DO 3000 I=NCH1,NCH2
      ISAMP=JSAMP(I)
      INDEX=(I-1)*LENG+1
C
      CALL POWBEM(DATA(INDEX),ISAMP)
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 RECTIFIED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'DATA SUCCESSFULLY RECTIFIED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE REOR(X,Y,Z,ISAMP,INF)
C
C*****************************************************************************
C
C     REORDER DATA CHANNELS
C
C     X - DATA ARRAY - INPUT AND OUTPUT
C     Y - DATA ARRAY - INPUT AND OUTPUT
C     Z - TEMPORARY SCRATCH ARRAY IF INF = 0 - INPUT AND OUTPUT
C     ISAMP - NUMBER OF SAMPLES IN X - INPUT
C     INF - REORDER FLAG - INPUT
C           = 0 X TO Y AND Y TO X
C           = 1 X TO Y
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*****************************************************************************
C
      DIMENSION X(1),Y(1),Z(1)
C
      IF(INF.EQ.0)GO TO 2000
C
      DO 1000 J=1,ISAMP
      Y(J)=X(J)
 1000 CONTINUE
C
      GO TO 9000
C
 2000 DO 3000 J=1,ISAMP
      Z(J)=X(J)
      X(J)=Y(J)
      Y(J)=Z(J)
 3000 CONTINUE
C
 9000 RETURN
      END
```

```
      SUBROUTINE REORX
C
C*****************************************************************
C
C     EXECUTIVE ROUTINE FOR REORDERING DATA CHANNELS
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 24, 1993  COLORADO
C
C*****************************************************************
C
      COMMON/ARR/JCODE,JCHN,RLAT,RLON,RELEV,JYEAR,JDOFY,JHOUR,JMIN,
     * RSEC,RSAMR,RA0,JNP,RPOLES,JNZ,RZEROS,JSAMP,JFLAG
      COMMON/DEM/IDEM,IDEM1
      COMMON/IN/BX
      COMMON/INOUT/IRE,IWR,IWR2
      COMMON/MUCHO/NCHLS,LENG,NACHLS
      COMMON/XDATA/DATA
C
      DIMENSION JCODE(100),JCHN(4,100),RLAT(100),RLON(100),
     * RELEV(100),JYEAR(100),JDOFY(100),JHOUR(100),JMIN(100),
     * RSEC(100),RSAMR(100),RA0(100),JNP(100),RPOLES(30,100),
     * JNZ(100),RZEROS(20,100),JSAMP(100),JFLAG(100)
      DIMENSION DATA(1),BX(1)
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN1',3X,'ICHAN2',/)
      READ(IRE,*,END=9000)ICHAN1,ICHAN2
C
C     ICHAN1 - DESIRED CHANNEL TO EXCHANGE WITH ICHAN2
C              = 0 NO EXCHANGE, BX TO ICHAN2
C     ICHAN2 - DESIRED CHANNEL TO EXCHANGE WITH ICHAN1
C              = 0 NO EXCHANGE, ICHAN1 TO BX
C
      WRITE(IWR,300)
      WRITE(IWR2,300)
  300 FORMAT(/,10X,'REORDER DATA PARAMETERS',/)
      WRITE(IWR,400)ICHAN1,ICHAN2
      WRITE(IWR2,400)ICHAN1,ICHAN2
  400 FORMAT(/,10X,'ICHAN1 = ',I4,5X,'ICHAN2 = ',I4,/)
C
      IF(ICHAN1.GT.NACHLS)GO TO 7000
      IF(ICHAN2.GT.NACHLS)GO TO 7000
      IF(LENG.GT.IDEM1)GO TO 7000
      IF(ICHAN1.LE.0)GO TO 3000
      IF(ICHAN2.LE.0)GO TO 4000
C
C     EXCHANGE TWO DATA ARRAY CHANNELS
C
      INF=0
      INDEX1=(ICHAN1-1)*LENG+1
      INDEX2=(ICHAN2-1)*LENG+1
C
      CALL REOR(DATA(INDEX1),DATA(INDEX2),BX,LENG,INF)
C
```

```
      CALL MOVHD(ICHAN1,ICHAN2,INF)
C
      GO TO 8000
C
C     BX TO ICHAN2
C
 3000 INF=1
      INDEX2=(ICHAN2-1)*LENG+1
C
      CALL REOR(BX,DATA(INDEX2),BX,LENG,INF)
C
      GO TO 8000
C
C     ICHAN2 TO BX
C
 4000 INF=1
      INDEX1=(ICHAN1-1)*LENG+1
C
      CALL REOR(DATA(INDEX1),BX,BX,LENG,INF)
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'FILES NOT EXCHANGED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'FILES SUCCESSFULLY EXCHANGED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE RERETR(LUN,NSAMP,IRFL1,IRFL2,IRFL3,ISTAT)
C
C*********************************************************************
C
C     EXECUTIVE ROUTINE FOR READING RETRV FILES
C
C     LUN - LOGICAL UNIT NUMBER OF FILE TO READ - INPUT
C     NSAMP - TOTAL NUMBER OF DATA SAMPLES READ - OUTPUT
C     IRFL1 - READ HEADER FLAG - OUTPUT
C             = 0 END CODE ENCOUNTERED
C             = 1 HEADER READ
C             =-1 END OF FILE READ
C     IRFL2 - READ DATA FLAG - OUTPUT
C             = 0 END OF DATA ENCOUNTERED
C             = 1 SUCCESSFUL DATA READ
C             =-1 ERROR OR END OF FILE
C     IRFL3 - DATA STATUS FLAG - OUTPUT
C             = 1 OBTAINED STATUS BITS
C             =-1 NOT ABLE TO OBTAIN STATUS BITS
C     ISTAT - STATUS BITS - OUTPUT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 11, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DEM/IDEM,IDEM1
      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
C
      DIMENSION CODE(4),CHN(4),POLES(30),ZEROES(20),NFLAG(3),BX(1),
     * ISTAT(11)
C
      COMPLEX POLES,ZEROES
      REAL*4 LAT,LON
      INTEGER*4 CODE,CHN,SYEAR,SDOFY,SHOUR,SMIN
C
      DATA NY/'y'/
C
      NSAMP=1
C
C     READ A HEADER RECORD
C
   50 IRFL1=IRDHD(LUN)
C
      IF(IRFL1.NE.1)GO TO 9000
C
      WRITE(IWR2,100)(CODE(I),I=1,4),(CHN(I),I=1,4),SYEAR,SDOFY,
     * SHOUR,SMIN,SSECS
  100 FORMAT(/,10X,4A1,1X,4A1,I6,3I4,F7.3)
      WRITE(IWR2,110)
  110 FORMAT(/,'++ INPUT: ','KEEP THIS STATION-CHANNEL (y/n)?',/)
      READ(IRE,150,END=9000)KY
  150 FORMAT(A1)
C
```

```
C     KY - y OR n
C
      IF(NY.NE.KY)GO TO 50
C
C     FILL DATA ARRAY BX
C
      DO 200 I=1,IDEM1
C
C     READ A DATA SAMPLE
C
      IRFL2=IRDDAT(LUN,VALU)
C
      IF(IRFL2.LT.0)GO TO 9000
      IF(IRFL2.EQ.0)GO TO 500
      BX(NSAMP)=VALU
      NSAMP=NSAMP+1
  200 CONTINUE
C
      NSAMP=NSAMP-1
      GO TO 9000
C
  500 NSAMP=NSAMP-1
C
C     OBTAIN STATUS BITS
C
      IRFL3=ISTBIT(LUN,ISTAT)
C
 9000 RETURN
      END
```

```
      COMPLEX FUNCTION RESP(ICHAN,OMEGA,IFL)
C
C*********************************************************************
C
C     COMPUTE THE COMPLEX INSTRUMENT RESPONSE FROM POLES AND ZEROES
C
C     RETURNS T(S)=A0*(S-Z(1))*...*(S-Z(M))/(S-P(1))*...*(S-P(N)),
C     WHERE P, Z, AND S ARE COMPLEX AND S = CMPLX(0.0,OMEGA).
C
C     ICHAN - CHANNEL TO COMPUTE RESPONSE FOR - INPUT
C     OMEGA - ANGULAR FREQUENCY TO COMPUTE RESPONSE FOR - INPUT
C     IFL - RESPONSE OUTPUT FLAG - INPUT
C           = 1 DISPLACEMENT RESPONSE
C           = 2 VELOCITY RESPONSE
C           = 3 ACCELERATION RESPONSE
C     RESP - COMPLEX RESPONSE FOR FREQUENCY OMEGA - OUTPUT
C
C     WRITTEN BY RAY BULAND
C
C     AUGUST 23, 1979  COLORADO
C
C
C     MODIFIED BY MADELEINE ZIRBES
C
C     JANUARY 2, 1986  COLORADO
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     AUGUST 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
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
      DOUBLE PRECISION WW
      COMPLEX RPOLES,RZEROS
      COMPLEX*16 ZZ,S,ZD
C
      IP=JNP(ICHAN)
      IF(IP.GT.0)GO TO 100
      RESP=CMPLX(0.0,0.0)
      GO TO 9000
C
  100 WW=OMEGA*1D0
      ZD=DCMPLX(1.D0,0.D0)
      S=DCMPLX(0.D0,WW)
C
C     COMPUTE THE CONTRIBUTION OF THE ZEROES
C
      IZ=JNZ(ICHAN)
C
```

```
      DO 200 J=1,IZ
      ZZ=RZEROS(J,ICHAN)
      ZD=ZD*(S-ZZ)
  200 CONTINUE
C
C     COMPUTE THE CONTRIBUTION OF THE POLES
C
      DO 300 J=1,IP
      ZZ=RPOLES(J,ICHAN)
      ZD=ZD/(S-ZZ)
  300 CONTINUE
C
      WW=RA0(ICHAN)*1.D0
C
C     DISPLACEMENT RESPONSE
C
      IF(IFL.NE.1)GO TO 400
      ZD=WW*ZD
      GO TO 1000
C
C     VELOCITY RESPONSE
C
  400 IF(IFL.NE.2)GO TO 500
      ZD=WW*ZD/S
      GO TO 1000
C
C     ACCELERATION RESPONSE
C
  500 IF(IFL.NE.3)GO TO 600
      ZD=WW*ZD/(S*S)
      GO TO 1000
C
  600 ZD=(0.D0,0.D0)
C
 1000 RESP=ZD
      RESP=RESP*10.**JFLAG(ICHAN)
C
 9000 RETURN
      END
```

```
      COMPLEX FUNCTION RESP2(OMEGA,IFL)
C
C*********************************************************************
C
C     COMPUTE THE COMPLEX INSTRUMENT RESPONSE FROM POLES AND ZEROES
C
C     RETURNS T(S)=A0*(S-Z(1))*...*(S-Z(M))/(S-P(1))*...*(S-P(N)),
C     WHERE P, Z, AND S ARE COMPLEX AND S = CMPLX(0.0,OMEGA).
C
C     OMEGA - ANGULAR FREQUENCY TO COMPUTE RESPONSE FOR - INPUT
C     IFL - RESPONSE OUTPUT FLAG - INPUT
C           = 1 DISPLACEMENT RESPONSE
C           = 2 VELOCITY RESPONSE
C           = 3 ACCELERATION RESPONSE
C     RESP2 - COMPLEX RESPONSE FOR FREQUENCY OMEGA - OUTPUT
C
C     WRITTEN BY RAY BULAND
C
C     AUGUST 23, 1979  COLORADO
C
C
C     MODIFIED BY MADELEINE ZIRBES
C
C     JANUARY 2, 1986  COLORADO
C
C
C     MODIFIED BY ROBERT P. MASSE
C
C     DECEMBER 29, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/RESPON/SA0,KNP,SPOLES,KNZ,SZEROS,KFLAG
C
      DIMENSION SPOLES(30),SZEROS(20)
C
      DOUBLE PRECISION WW
      COMPLEX SPOLES,SZEROS
      COMPLEX*16 ZZ,S,ZD
C
      IP=KNP
      IF(IP.GT.0)GO TO 100
      RESP2=CMPLX(0.0,0.0)
      GO TO 9000
C
  100 WW=OMEGA*1D0
      ZD=DCMPLX(1.D0,0.D0)
      S=DCMPLX(0.D0,WW)
C
C     COMPUTE THE CONTRIBUTION OF THE ZEROES
C
      IZ=KNZ
C
      DO 200 J=1,IZ
      ZZ=SZEROS(J)
      ZD=ZD*(S-ZZ)
  200 CONTINUE
C
```

```
C     COMPUTE THE CONTRIBUTION OF THE POLES
C
      DO 300 J=1,IP
      ZZ=SPOLES(J)
      ZD=ZD/(S-ZZ)
  300 CONTINUE
C
      WW=SA0*1.D0
C
C     DISPLACEMENT RESPONSE
C
      IF(IFL.NE.1)GO TO 400
      ZD=WW*ZD
      GO TO 1000
C
C     VELOCITY RESPONSE
C
  400 IF(IFL.NE.2)GO TO 500
      ZD=WW*ZD/S
      GO TO 1000
C
C     ACCELERATION RESPONSE
C
  500 IF(IFL.NE.3)GO TO 600
      ZD=WW*ZD/(S*S)
      GO TO 1000
C
  600 ZD=(0.D0,0.D0)
C
 1000 RESP2=ZD
      RESP2=RESP2*10.**KFLAG
C
 9000 RETURN
      END
```

```
      SUBROUTINE RESPC(ICHAN,FMN,FMX,DF,IFL,FREQ,C,PH,N,IRX)
C
C*********************************************************************
C
C     COMPUTE THE COMPLETE RESPONSE FUNCTION FOR A CHANNEL
C
C     ICHAN - CHANNEL NUMBER - INPUT
C     FMN - MINIMUM FREQUENCY - INPUT
C     FMX - MAXIMUM FREQUENCY - INPUT
C     DF - FREQUENCY INCREMENT - INPUT
C     IFL - RESPONSE OUTPUT FLAG - INPUT
C           = 0 SET TO 1
C           = 1 DISPLACEMENT RESPONSE
C           = 2 VELOCITY RESPONSE
C           = 3 ACCELERATION RESPONSE
C     FREQ - ARRAY OF FREQUENCY VALUES - OUTPUT
C     C - AMPLITUDES - OUTPUT
C     PH - PHASES - OUTPUT
C     N - NUMBER OF RESPONSE VALUES - OUTPUT
C         (LIMITED TO 10000 VALUES)
C     IRX - RESPONSE FLAG - INPUT
C           = 0 USE RESPONSE IN CHANNEL HEADER
C           = 1 USE RESPONSE FROM MASTER FILE
C
C     WRITTEN BY ROBERT P. MASSE
C
C     OCTOBER 20, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/DAZCON/PI,RD2DG,DG2RD,DG2KM,GEOCO1,GEOCO2,TWOPI,KNN,KSS,
     * KEE,KWW
C
      DIMENSION FREQ(1),C(1),PH(1)
C
      COMPLEX RES,RESP,RESP2
C
      I=1
      FREQ(I)=FMN
      GO TO 200
C
  100 FREQ(I)=FREQ(I-1)+DF
      IF(FREQ(I).GT.FMX)GO TO 9000
  200 OMEGA=TWOPI*FREQ(I)
C
C     CALCULATE COMPLEX RESPONSE
C
      IF(IRX.NE.0)GO TO 300
C
      RES=RESP(ICHAN,OMEGA,IFL)
C
      GO TO 500
C
  300 RES=RESP2(OMEGA,IFL)
C
C     CALCULATE AMPLITUDE
C
  500 C(I)=CABS(RES)
```

```
C
C      CALCULATE PHASE
C
       PH(I)=PHASOR(AIMAG(RES),REAL(RES))
C
       N=I
       I=I+1
       IF(N.GT.9999)GO TO 9000
       GO TO 100
C
 9000 RETURN
       END
```

```
      SUBROUTINE RESPX
C
C*****************************************************************************
C
C     EXECUTIVE ROUTINE FOR CALCULATING INSTRUMENT RESPONSE FOR
C     A CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     NOVEMBER 5, 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
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 BX(1),FREQ(1),T(1),AMP(1),PH(1)
C
      EQUIVALENCE (BX(20001),FREQ(1)),(BX(30001),T(1))
      EQUIVALENCE (BX(40001),AMP(1)),(BX(50001),PH(1))
C
      COMPLEX RPOLES,RZEROS
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - CHANNEL NUMBER TO DETERMINE INSTRUMENT RESPONSE
C             = 0 SET TO 1
C
      IF(ICHAN.LE.0)ICHAN=1
      WRITE(IWR,200)ICHAN
      WRITE(IWR2,200)ICHAN
  200 FORMAT(/,10X,'ICHAN = ',I4,/)
C
      IF(ICHAN.GT.NACHLS)GO TO 7000
C
      WRITE(IWR2,300)
  300 FORMAT(/,'++ INPUT: ','IFL',3X,'IRX',/)
      READ(IRE,*,END=9000)IFL,IRX
C
C     IFL - RESPONSE OUTPUT FLAG
C           = 0 SET TO 1
C           = 1 DISPLACEMENT RESPONSE
C           = 2 VELOCITY RESPONSE
C           = 3 ACCELERATION RESPONSE
C     IRX - RESPONSE FLAG
C           = 0 USE INSTRUMENT POLES AND ZEROS IN CHANNEL HEADER
C           = 1 USE INSTRUMENT POLES AND ZEROS FROM MASTER FILE
C
      IF(IFL.LE.0)IFL=1
```

```
      WRITE(IWR,400)IFL,IRX
      WRITE(IWR2,400)IFL,IRX
  400 FORMAT(/,10X,'IFL = ',I2,5X,'IRX = ',I2,/)
C
      IF(IRX.LE.0)GO TO 450
C
C     SET UP DESIRED POLES AND ZEROS FROM MASTER FILE OR KEYBOARD
C
      CALL SETRES
C
  450 WRITE(IWR2,500)
  500 FORMAT(/,'++ INPUT: ','FMN',3X,'FMX',3X,'DF',3X,'NL2',3X,
     * 'SAMR',/)
      READ(IRE,*,END=9000)FMN,FMX,DF,NL2,SAMR
C
C     FMN - MINIMUM FREQUENCY TO COMPUTE RESPONSE FOR
C     FMX - MAXIMUM FREQUENCY TO COMPUTE RESPONSE FOR
C     DF - INCREMENT IN FREQUENCY FOR COMPUTATIONS
C          (NUMBER OF FREQUENCY VALUES LIMITED TO 10000)
C          = 0 SET TO FMN
C     NL2 - LENGTH OF TIME SERIES THAT FMN AND FMX WILL CORRESPOND TO
C           (MUST BE A POWER OF 2)
C           = 0 USE PARAMETERS FMN,FMX,DF
C           > 0 USE NL2 AND SAMR TO COMPUTE FMN,FMX,DF
C     SAMR - SAMPLING RATE (SAMPLES/S)
C            = 0 USE VALUE FROM CHANNEL HEADER
C
      IF(NL2.LE.0)GO TO 600
      IF(SAMR.LE.0.0)SAMR=RSAMR(ICHAN)
      FMN=SAMR/NL2
      FMX=SAMR/2.0
      DF=FMN
C
  600 IF(DF.LE.0.0)DF=FMN
      WRITE(IWR,700)FMN,FMX,DF,NL2,SAMR
      WRITE(IWR2,700)FMN,FMX,DF,NL2,SAMR
  700 FORMAT(/,10X,'FMN = ',F8.3,5X,'FMX  = ',F8.3,5X,'DF = ',F8.4,/,
     * 10X,'NL2 = ',I8,5X,'SAMR = ',F8.3,/)
C
      IF(FMN.LE.0.0.OR.FMX.LE.0.0)GO TO 7000
      IF(FMN.GT.FMX)GO TO 7000
C
C     CALCULATE RESPONSE
C
      CALL RESPC(ICHAN,FMN,FMX,DF,IFL,FREQ,AMP,PH,N,IRX)
C
      IF(IFL.NE.1)GO TO 1500
      WRITE(IWR,1100)
      WRITE(IWR2,1100)
 1100 FORMAT(//,20X,'DISPLACEMENT RESPONSE',/)
      GO TO 2200
C
 1500 IF(IFL.NE.2)GO TO 2000
      WRITE(IWR,1600)
      WRITE(IWR2,1600)
 1600 FORMAT(/,20X,'VELOCITY RESPONSE',/)
      GO TO 2200
C
```

```
 2000 WRITE(IWR,2100)
      WRITE(IWR2,2100)
 2100 FORMAT(/,20X,'ACCELERATION RESPONSE',/)
C
 2200 WRITE(IWR2,2500)
 2500 FORMAT(/,'++ INPUT: ','IPRT',3X,'IPLOT',3X,'IFFT',/)
      READ(IRE,*,END=9000)IPRT,IPLOT,IFFT
C
C     IPRT - LIST RESPONSE FLAG
C            > 0 LIST RESPONSE
C     IPLOT - PLOT RESPONSE FLAG
C             > 0 PLOT RESPONSE FLAG
C     IFFT - FREQUENCY/PERIOD FLAG
C            = 0 PLOT FREQUENCY VALUES
C            = 1 PLOT PERIOD VALUES
C
      WRITE(IWR,2600)IPRT,IPLOT,IFFT
      WRITE(IWR2,2600)IPRT,IPLOT,IFFT
 2600 FORMAT(/,10X,'IPRT = ',I2,5X,'IPLOT = ',I2,5X,'IFFT = ',I2,/)
C
      IF(IPRT.LE.0)GO TO 5000
C
C     LIST RESPONSE
C
 3000 WRITE(IWR,3200)
      WRITE(IWR2,3200)
 3200 FORMAT(/,10X,'FREQ',4X,'PERIOD',8X,'AMP',8X,'PHASE',/)
C
      DO 4000 I=1,N
      T(I)=1.0/FREQ(I)
 4000 CONTINUE
C
      WRITE(IWR,4100)(FREQ(I),T(I),AMP(I),PH(I),I=1,N)
      WRITE(IWR2,4100)(FREQ(I),T(I),AMP(I),PH(I),I=1,N)
 4100 FORMAT(6X,F8.3,2X,F8.2,1X,2F12.3)
C
 5000 IF(IPLOT.LE.0)GO TO 9000
C
C     PLOT RESPONSE
C
      N2X=1
      IPOW=9
      SAMR=1.0
C
      CALL PLSPEC(AMP,PH,N,N2X,FMN,FMX,DF,SAMR,IPOW,IFFT)
C
      GO TO 9000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE REVOLV(RN,TE,ISAMP,GAM,BETA)
C
C*********************************************************************
C
C     ROTATE HORIZONTAL COMPONENTS
C
C     RN - INSTALLED NORTH OR RADIAL COMPONENT - INPUT
C        - ROTATED RADIAL COMPONENT - OUTPUT
C     TE - INSTALLED EAST OR TRANSVERSE COMPONENT - INPUT
C        - ROTATED TRANSVERSE COMPONENT - OUTPUT
C          (POSITIVE VALUE IS FOR GROUND MOTION TO THE RIGHT
C           OF THE DIRECTION OF PROPAGATION)
C     ISAMP - NUMBER OF SAMPLES TO ROTATE - INPUT
C     GAM - STATION-TO-EPICENTER AZIMUTH (IN DEGREES) - INPUT
C     BETA - ANGLE OF INSTALLED RADIAL (NORTH) COMPONENT WITH
C            RESPECT TO TRUE NORTH (IN DEGREES) - INPUT
C            ( 0 FOR INSTALLED N-S AND E-W COMPONENTS)
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 RN(1),TE(1)
C
      ALPHA=(GAM-BETA)*DG2RD
      C=COS(ALPHA)
      S=SIN(ALPHA)
C
      DO 100 I=1,ISAMP
      RTEMP=-RN(I)*C-TE(I)*S
      TE(I)=RN(I)*S-TE(I)*C
      RN(I)=RTEMP
  100 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE REVPOL
C
C*********************************************************************
C
C     REVERSE POLARITY OF A DATA CHANNEL
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 20, 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 POLARITY PARAMETERS
C
      WRITE(IWR2,50)
   50 FORMAT(/,'++ INPUT: ','ICHAN',/)
      READ(IRE,*,END=9000)ICHAN
C
C     ICHAN - DESIRED CHANNEL
C             = 0 REVERSE ALL CHANNELS
C
      WRITE(IWR,100)ICHAN
      WRITE(IWR2,100)ICHAN
  100 FORMAT(/,10X,'REVERSE DATA IN CHANNEL ',I4,/)
C
      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     REVERSE POLARITY
C
 2000 DO 3000 I=NCH1,NCH2
      INDEX=(I-1)*LENG
      ISAMP=JSAMP(I)
C
      DO 3000 J=1,ISAMP
      IN=INDEX+J
      DATA(IN)=-DATA(IN)
```

```
 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 POLARITY NOT REVERSED',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'POLARITY OF DATA SUCCESSFULLY REVERSED',/)
C
 9000 RETURN
      END
```

```
      SUBROUTINE REWD
C
C*********************************************************************
C
C     REWIND LOGICAL UNIT
C
C     WRITTEN BY ROBERT P. MASSE
C
C     AUGUST 23, 1993  COLORADO
C
C*********************************************************************
C
      COMMON/INOUT/IRE,IWR,IWR2
C
      WRITE(IWR2,100)
  100 FORMAT(/,'++ INPUT: ','LUN',/)
      READ(IRE,*,END=9000)LUN
C
C     LUN - LOGICAL UNIT NUMBER
C
      IF(LUN.LE.0)GO TO 7000
      WRITE(IWR,200)LUN
      WRITE(IWR2,200)LUN
  200 FORMAT(/,10X,'REWIND LOGICAL UNIT ',I3,/)
C
      REWIND LUN
C
      GO TO 8000
C
 7000 WRITE(IWR2,7100)
 7100 FORMAT(/,10X,'***** ERROR IN INPUT PARAMETERS *****',/)
      WRITE(IWR2,7200)
 7200 FORMAT(/,10X,'FILE NOT REWOUND',/)
      GO TO 9000
C
 8000 WRITE(IWR,8100)
      WRITE(IWR2,8100)
 8100 FORMAT(/,10X,'FILE SUCCESSFULLY REWOUND',/)
C
 9000 RETURN
      END
```