

# Chapter 7

## Summary

This report describes the multivariate data analysis techniques for the identification of simple patterns within a data set and between data sets, for studying the characteristics of moving patterns in space-time data sets and for analysis of inter-relationships between data sets.

The softwares developed are for,

- Empirical Orthogonal Function (EOF/ SVD) analysis
- Low Pass Filter/ Band Pass Filter/ Hilbert Transform - all in frequency domain
- Complex Empirical Orthogonal Function (CEOF) Analysis in time domain,
- Cross Spectrum & Rotary Spectrum analysis, and
- Canonical Correlation Analysis (CCA).

The above softwares are useful in climate research.

The CEOF analysis software can be used for analyzing geophysical data sets with standing waves as well as propagating waves embedded in them. It may be mentioned that EOF analysis can identify only standing waves.

The software for CVA analysis can be used for, analyzing inter-relationships between data sets. CVA is also used in statistical prediction of climate [Barnett and Preisendorfer, 1987].

The listings of the above mentioned softwares alongwith the associated specification files are given in Appendix A. In this appendix the test data for CCA as well as the results obtained on running the computer program are also given.

The test data sets considered and the results of Hilbert Transform and CEOF Analysis are discussed in Chapter 5. Results of EOF & CEOF analysis of TOPEX/Poseidon derived sea level height anomaly data in the Indian and Pacific Oceans at interannual time scales are discussed in Chapter 6.

# Appendix A

## Software

This Appendix contains the FORTRAN source code constructed on the “Black Box” principle, along with the associated specification file for each computer program - for performing the following analyses which are useful in climate research studies. Interaction with the computer programs is through the Specification files, which includes provision for names of input file, output file, Number of spatial locations, Number of elements in time series, sampling interval in time, format of data file, etc.

- *EOF Analysis* including Varimax rotation. The theory is discussed in Chapter 2. We have included this computer program for the sake of completeness. Originally we did not wish to include this program because as mentioned by Kantha and Clayson [2000], EOF analysis can be performed on MATLAB by giving a single instruction, viz.,  $[U, S, V] = SVD(F)$ , where  $F = U \cdot S \cdot V$  gives the Singular Value Decomposition of the matrix  $F$ . Also we had published the technical reports Fernandes and Mahadevan [1982] and Fernandes and Gupta [1994], which discussed the theory of EOF analysis and listed the computer program for the same.
- *Hilbert Transform* - this program can also perform Low Pass filtering or Band Pass filtering by choosing one of the letters (L/B/H) in the appropriate place in the Specification file. The filtering in this computer program is performed in frequency domain. Originally this program was used for Low Pass and Band Pass filtering only and limited to a single time series. It was only recently, when we ventured into the field of Complex EOF analysis that the computer program was extended to performing the Hilbert transform as well as treatment of multiple time series, i.e., at a number of locations. Sometimes for the purpose of performing Complex EOF analysis, like White [2000a] and White [2000b] one has to apply a Low Pass or Band Pass filter respectively before obtaining the Hilbert Transform - such an operation will necessitate running this program twice, the first time for obtaining

the filtered data (Low or Band Passed as desired) and the second time to obtain the Hilbert Transform of the filtered data. If the original time series input file is say, *sla-pacr.###*; the computer program automatically creates an output file *sla-pacr.lat* or *sla-pacr.bat* or *sla-pacr.hat*, when a low pass or band pass filter or a Hilbert Transform respectively is demanded. This allows the output file of this program *sla-pacr.hat*, which contains both the data as well as its Hilbert Transform, to be the input file of the Complex EOF analysis program mentioned below. There is a provision in the specification file for deleting one tenth of the time series at the beginning and end of each time series in order to take care of “end-effects” associated with frequency domain filters [Horel, 1984].

- *Complex EOF analysis in time domain* including Varimax rotation. The theory is discussed in Chapter 3.
- *Cross Spectrum Analysis* - this program can also perform Rotary Spectrum Analysis by the method of Gonella [1972] when the x- and y- series are the u- and v- components of the wind or the ocean current.

**The computer program in FORTRAN for performing Canonical Correlation Analysis discussed in Chapter 4, along with test data from Pielou [1977] and results obtained on running the program are also given in this Appendix.**

## A.1 EOF Analysis

### A.1.1 Computer Program: eof-vec.f

```

PROGRAM FFT
parameter (MR=1024,MC=10000,MF=20)
DIMENSION X(MR,MC),R(MR,MR),U(MR,MF),V(MC,MF),VV(MC,MF)
DIMENSION amean(MC)
dimension utu(MF,MF),vtv(MF,MF)
dimension T(MF,MF),FAC(MF),E(MF),vg(MC,MF)
dimension percv(MF),cpercv(MF)
DIMENSION com(MC),zzz(MR,MC)
dimension hsq(MC),diff(MF),xdiff(MF)
real lat(MC),long(MC)
REAL LABEL(80),IBIG(19)
CHARACTER*25 INFIL,OUTFIL,ltlfil,ltlfmt,xfmt
CHARACTER*1 ANS

character*75 dummy

```

```

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      Read Specification file

      OPEN (9,FILE='eof-vec.spc',ACCESS='SEQUENTIAL',blank='zero')
      READ(9,'(50x,A25)')OUTFIL
      READ(9,'(50x,A25)')INFIL

      READ(9,'(50x,A25)')xfmt
      READ(9,'(50x,i5)')nc
      READ(9,'(50x,i5)')nr
      READ(9,'(50x,i5)')nf
      READ(9,'(50x,A25)')ltilfil
      READ(9,'(50x,A25)')ltilfmt
      READ(9,'(50x,i5)')iprintall
      OPEN (3,FILE=OUTFIL,ACCESS='SEQUENTIAL',status='unknown')
      WRITE(3,1800)OUTFIL,INFIL,xfmt,nc,nr,nf,ltilfil,ltilfmt,iprintall
      WRITE(*,1800)OUTFIL,INFIL,xfmt,nc,nr,nf,ltilfil,ltilfmt,iprintall

1800   format(1x,'Output file : ',a25/
#      1x,'Input file : ',a25/
#      1x,'format for input file : ',a25/
#      1x,'Number of stations : ',i5/
#      1x,'Number of time steps : ',i5/
#      1x,'Number of time factors: ',I5/
#      1x,'Lat Long file : ',a25/
#      1x,'Format for input file : ',a25/
#      1x,'IPRINTALL : ',I5)

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c      Read Lat-long file

      OPEN(68,FILE=LTLFIL,blank='zero',
#      access='sequential')

      read(68,'(1a75)')dummy
      write(3,'(1a75)')dummy
      do j=1,nc
      read(68,'(1i5,2f7.1)')jc,lat(j),long(j)
      write(3,'(1i5,2f7.1)')jc,lat(j),long(j)

```

```

c          if (jc.ne.j)stop 'lat-long data reading error'
          end do
cccc     c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c          Read Data file

          call datae(nr,nc,x,label,infil,xfmt)
cccc     c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

          write(*,*)' NR, NC = ',nr,nc
c          Data preparation: Removal of temporal mean

          do j=1,nc
          amean(j)=0.
          do i=1,nr
          amean(j)=amean(j)+x(i,j)
          end do
          amean(j)=amean(j)/nr
          end do

          do j=1,nc
          do i=1,nr
          x(i,j)=-amean(j)+x(i,j)
          end do
          end do
          write(*,*)' okay a'

c

c

cccc     c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c          Rendering data matrix with unit variance

          summmm=0.
          do i=1,nr
          do j=1,nc
          summmm=summmm+X(i,j)**2
          end do
          end do

c          write(*,*)' Okay 1'

```

```

c
c
      write(*,*) ' summmm = ',summmm
      write(3,*) ' summmm = ',summmm
      do i=1,nr
      do j=1,nc
      x(i,j)=x(i,j)/sqrt(summmm)
      end do
      end do
c
      if(iprintall.eq.1)then
      WRITE(*,1)
      WRITE(3,1)
1      FORMAT(///1X,'M A T R I X X, A F T E R D A T A P R E P A R A
+      T I O N')
      DO I=1,NR
      WRITE(3,203)I,(X(I,J),J=1,NC)
203     FORMAT(5X,I3,2X,10E12.3/(10X,10E12.3))
      end do
      end if
      NVAL=NF
      NVEC=NVAL

      DO 21 I=1,NR
      DO 21 J=i,NR
      SS=0.
      DO 10 K=1,NC
10     SS=SS+X(I,K)*X(J,K)
      R(I,J)=SS
21     R(J,I)=R(I,J)

      if(iprintall.eq.1)then
      WRITE(3,211)
      DO 200 I=1,NR
200     WRITE(3,203)I,(R(I,J),J=1,NR)
      end if
      CALL EIGAN(R,U,E,NR,NVAL,NVEC,SETA)
      write(*,*) ' seta = ',seta

```

```

write(3,*)' seta = ',seta

open(41,file='matrix-u',access='sequential',status='unknown')
write(41,*)'matrix-u'
write(3,*)'matrix-u'

DO I=1,NR
WRITE(41,201)I,(U(I,J),J=1,NF)
WRITE(3,201)I,(U(I,J),J=1,NF)
end do

cccc  c  cutu matrix nfac2 ccccccccccccccccccccccccccccccccc
do i=1,nf
do j=i,nf
ss=0.
do k=1,nr
ss=ss+U(k,i)*U(k,j)
end do
utu(i,j)=ss
utu(j,i)=ss
end do
end do

write(3,*)' utu matrix (nf)'
do i=1,nf
write(3,201)i,(utu(i,j),j=1,nf)
end do
write(*,*)' okay323'
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccc

do k=1,nf
do i=1,nc
vg(i,k)=0.
do j=1,nr
vg(i,k)=vg(i,k)+x(j,i)*u(j,k)
end do
end do
end do
write(*,*)' okay32x'
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

do i=1,nf
do j=i,nf
ss=0.
do k=1,nc
ss=ss+vg(k,i)*vg(k,j)
end do
vtv(i,j)=ss
vtv(j,i)=ss
end do
end do
write(*,*)' okay32y'
write(3,*)' vtv matrix %'
do i=1,nf
write(3,201)i,(vtv(i,j),j=1,nf)
end do

write(3,282)(vtv(j,j),j=1,nf)

do j=1,nf
percv(j)=vtv(j,j)*100.
end do
cpercv(1)=percv(1)
do j=2,nf
cpercv(j)=percv(j)+cpercv(j-1)
end do
open(51,file='eof-perc-var-greek-new',access='sequential',
# status='unknown')
write(51,284)
284 format(//' ORIGINAL, i.e., UNROTATED MODES'/
# '#####')
WRITE(51,285)
285 FORMAT('MODE',t8,'Percentage variance',t35,
# 'Cumulative percentage'/
# 'No.',t8,'explained',t35,
# 'variance explained'//)
286 format(i4,t8,f10.5,t35,f10.5)
do j=1,nf
write(51,286)j,percv(j),cpercv(j)
end do

```



```

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc      computation of Greek "sigma squared" used for determining whether
cc      varimax rotation is useful or not!
cc      Part I of two parts, part II is lower down
        write(*,*)' i hsq(i)'
        write(3,*)' i hsq(i)'

        do i=1,nc
        hsq(i)=0.
        do j=1,nf
        hsq(i)=hsq(i)+vg(i,j)*vg(i,j)
        end do
        write(*,3715)i,hsq(i)
        write(3,3715)i,hsq(i)
        end do
3715    format(i5,f7.4)
        asumhsq=0.
        do i=1,nc
        asumhsq=asumhsq+hsq(i)
        end do
        write(*,*)' asumhsq = ',asumhsq
        write(3,*)' asumhsq = ',asumhsq

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        write(*,*)' okay32z'

        do i=1,nc
        do k=1,nf
        v(i,k)=vg(i,k)/sqrt(e(k))
        end do
        end do

        if(iprintall.eq.1)then

```

```

write(3,*)'matrix-v'
DO I=1,NC
WRITE(3,201)I,(V(I,J),J=1,NF)
end do
end if
cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
do i=1,nf
do j=i,nf
ss=0.
do k=1,nc
ss=ss+v(k,i)*v(k,j)
end do
vtv(i,j)=ss
vtv(j,i)=ss
end do
end do

write(*,*)' okay32p'

write(3,*)' vtv matrix'
do i=1,nf
write(3,201)i,(vtv(i,j),j=1,nf)
end do
cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
do j=1,nf
do i=1,nc
vv(i,j)=vg(i,j)*sqrt(summmm/nr)
end do
end do

c In UNIT=45, it is really matrix vv which is actually printed
open(45,file='matrix-v',access='sequential',status='unknown')
write(45,*)'matrix-v'
write(3,*)'matrix-vv'

do i=1,nc
write(45,201)i,(vv(i,k),k=1,nf),lat(i),long(i)
write(3,201)i,(vv(i,k),k=1,nf),lat(i),long(i)
end do

```

```

write(*,*)' okay32q'
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

do i=1,nr
do j=1,nc
zzz(i,j)=0.
do k=1,nf
zzz(i,j)=zzz(i,j)+sqrt(e(k))*u(i,k)*v(j,k)
end do
zzz(i,j)=zzz(i,j)*sqrt(summmm)
end do
end do
open(63,file='recomputed-matrix',access='sequential',
# status='unknown')
WRITE(63,*)' recomputed matrix'
write(63,202)
DO I=1,NR
WRITE(63,'(10000e16.7)')(zzz(I,J),J=1,nc)
end do
write(*,*)' okay32r'
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

CALL VARMAX(vg,COM,NF,nc,T)
write(*,*)' okay32s'

WRITE(3,253)
DO I=1,NF
WRITE(3,201)I,(T(I,J),J=1,NF)
end do

open(42,file='matrix-u-rotated',access='sequential',
# status='unknown')
write(42,*)' matrix-u-rotated'
write(3,*)' matrix-u-rotated'

DO J=1,nr
DO I=1,NF
FAC(I)=0.
DO L=1,NF

```

```

FAC(I)=FAC(I)+u(J,L)*T(L,I)
end do
end do

DO I=1,NF
u(J,I)=FAC(I)
end do
WRITE(42,201)J,(u(J,I),I=1,NF)
WRITE(3,201)J,(u(J,I),I=1,NF)
end do

```

```

do j=1,nf
do i=1,nc
vv(i,j)=vg(i,j)*sqrt(summmm/nr)
end do
end do

```

c In UNIT=46, it is really matrix vv (rotated!) which is actually printed

```

open(46,file='matrix-v-rotated',access='sequential',
# status='unknown')
write(46,*)'matrix-v-rotated'
write(3,*)'matrix-vg-rotated'
do i=1,nc
write(46,201)i,(vv(i,k),k=1,nf),lat(i),long(i)
write(3,201)i,(vv(i,k),k=1,nf),lat(i),long(i)
end do

```

```

cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccc
do i=1,nf
do j=i,nf
ss=0.
do k=1,nc
ss=ss+vg(k,i)*vg(k,j)
end do
vtv(i,j)=ss
vtv(j,i)=ss
end do
end do

```

```

write(3,*)' vtv rotated matrix %'
do i=1,nf
write(3,201)i,(vtv(i,j),j=1,nf)
end do
write(3,282)(vtv(j,j),j=1,nf)

do j=1,nf
percv(j)=vtv(j,j)*100.
end do
cpercv(1)=percv(1)
do j=2,nf
cpercv(j)=percv(j)+cpercv(j-1)
end do
write(51,283)
283 format(////' VARIMAX ROTATED MODES'/
# '#####')
WRITE(51,285)
do j=1,nf
write(51,286)j,percv(j),cpercv(j)
end do

cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc computation of Greek "sigma squared" used for determining whether
cc varimax rotation is useful or not!
cc Part II

do l=1,nf
diff(l)=0.
do i=1,nc
diff(l)=diff(l)+vg(i,l)*vg(i,l)/hsq(i)
end do
diff(l)=diff(l)/nc
end do

do l=1,nf
xdiff(l)=0.
do i=1,nc
xdiff(l)=xdiff(l)+(vg(i,l)*vg(i,l)/
# hsq(i)-diff(l))**2

```

```

end do
xdiff(1)=xdiff(1)/nc
end do

sigmasq=0.
do l=1,nf
sigmasq=sigmasq+xdiff(l)
end do

write(*,*)' sigmasq = ',sigmasq
write(3,*)' sigmasq = ',sigmasq

anew=100.*sigmasq*nf/(nf-1)

write(*,*)' anew = ',anew
write(3,*)' anew = ',anew
write(51,290)anew
290 format('#####' /
# //'Greek new = ',f9.2)

c#### # #####

cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

do i=1,nr
do j=1,nc
zzz(i,j)=0.
do k=1,nf
zzz(i,j)=zzz(i,j)+u(i,k)*vg(j,k)
end do
zzz(i,j)=zzz(i,j)*sqrt(summmm)
end do
end do
open(64,file='recomputed-matrix-rotated',access='sequential',
# status='unknown')

```

```

WRITE(64,*)' recomputed matrix - varimax rotated'
write(64,202)
DO I=1,NR
WRITE(64,'(10000e16.7)')(zzz(I,J),J=1,nc)
end do

202      format(//)
201      FORMAT(5X,I5,10000F11.5)
211      FORMAT(//1X,23HMATRIX X(TRANPOSE) X /)
282      FORMAT(//5X,'PERCENTAGE VARIANCE EXPLAINED BY EACH FACTOR ARE GIVE
1 N BELOW'/10X,100(2X,F9.5))
253      FORMAT(// 5X,23HVARIMAX ROTATION MATRIX /)

CLOSE(3)
WRITE(*,12345)INFIL,OUTFIL
WRITE(51,12345)INFIL,OUTFIL
12345   FORMAT(//7X,'FILE DIARY'//2X,'INPUT FILE = ',A25/
# 2X,'OUTPUT FILE = ',A25/
# 2x,'matrix-u'/
# 2x,'matrix-v'/
# 2x,'matrix-u-rotated'/
# 2x,'matrix-v-rotated'/
# 2x,'eof-perc-var-greek-new')
STOP
END

subroutine datae(nr,nc,x,label,infil,xfmt)
parameter (MR=1024,MC=10000)
dimension x(MR,MC)
real label(80)
character*25 xfmt,infil

OPEN (2,FILE=INFIL,ACCESS='SEQUENTIAL',blank='zero')

READ(2,432)(LABEL(I),I=1,80)

```

```

432      FORMAT(20A4)
        WRITE(3,432)(LABEL(I),I=1,80)
        WRITE(*,432)(LABEL(I),I=1,80)

        WRITE(*,6588)
6588      format(' IS THIS THE LABEL INFORMATION (4 LINES) OF YOUR INPUT DAT
#        A ? (y/n): ', $)
        CALL ENTRY(IWW,2,'y','n','3','4','5')
        IF(IWW.EQ.2)STOP' Check your input data file !!'

        WRITE(*,1700)
1700      format(' Is the matrix read rowwise or columnwise? (r/c): ', $)
        CALL ENTRY(IWW,2,'r','c','3','4','5')
        IF(iww.eq.2) GO TO 100
        DO 1 I=1,NR
1         READ(2,XFMT)(X(I,J),J=1,NC)
          go to 3
100        CONTINUE
        DO 2 J=1,NC
2         READ(2,XFMT)(X(I,J),I=1,NR)
3         continue
        WRITE(*,456)
        WRITE(3,456)
456        FORMAT(1X,'D A T A M A T R I X X ROWWISE')
        DO I=1,2
        WRITE(*,201)I,(X(I,J),J=1,NC)
        END DO
        DO 200 I=1,NR
200        WRITE(3,203)I,(X(I,J),J=1,NC)
        WRITE(*,1709)
1709      format(' ARE THESE THE FIRST TWO LINES OF YOUR INPUT DATA MATRIX R
#        OW-WISE? (y/n): ', $)
        CALL ENTRY(IWW,2,'y','n','3','4','5')
        IF(IWW.EQ.2)STOP' Check your input data file !!'
201        FORMAT(5X,I3,2X,5E12.3/(10X,5E12.3))
203        FORMAT(5X,I3,2X,10E12.3/(10X,10E12.3))
        return
        end

        SUBROUTINE PREPAR(X,NR,NC)

```



```
parameter (MR=1024,MC=10000)
DIMENSION X(MR,MC)
return
END

SUBROUTINE EIGAN(A,V,E,N,NVAL,NVEC,SETA)

C
C THIS ROUTINE USES THE GIVENS-HOUSEHOLDER METHOD
C DESCRIBED IN Ortega [1967]
C

parameter (MR=1024,MF=20)
DIMENSION A(MR,MR),V(MR,MF),B(MR),C(MR)
dimension P(MR),Q(MR),R(MR),
1 W(MR),Y(MR+2),IN(MR),E(MF)
EPA=1.E-06
EPB=1.E-20
EPC=1.E-30
EPD=1.E-10
BIG=1.E+20
NTIME=2
NITR=20
SET=0.
DO 17 J=1,N
17 SET=SET+A(J,J)
SETA=SET
WRITE(3,203)SET
NMA=N-1
NMB=N-2
DO 18 I=1,NMB
IPA=I+1
SS=0.
DO 10 J=IPA,N
10 SS=SS+A(J,I)**2
S=SQRT(SS)
IF(A(IPA,I).GT.0.)S=-S
C(I)=A(I,I)
B(I)=S

C
C IF S IS ZERO THEN ALPHA MUST BE ZERO
C
ALPHA=0.
```

```
TOL=1.E-20
IF(SS.LT.TOL) GO TO 18
ALPHA=1./(SS-A(IPA,I)*S)
T=A(IPA,I)-S
A(IPA,I)=T
W(I+1)=T
IPB=I+2
DO 11 J=IPB,N
11 W(J)=A(J,I)
SET=0.
DO 13 J=IPA,N
T=0
DO 12 K=IPA,N
12 T=T+A(J,K)*W(K)
P(J)=T*ALPHA
13 SET=SET+W(J)*P(J)
SET=SET*ALPHA*0.5
DO 14 K=IPA,N
14 Q(K)=P(K)-SET*W(K)
DO 15 J=IPA,N
DO 15 K=J,N
A(J,K)=A(J,K)-Q(J)*W(K)-Q(K)*W(J)
15 A(K,J)=A(J,K)
18 A(I,I)=ALPHA
C(N-1)=A(N-1,N-1)
C(N)=A(N,N)
B(N-1)=A(N-1,N)
B(N)=0.
C
C COMPUTE THE EIGENVALUES
C
TOP=ABS(C(1))+ABS(B(1))
DO 20 I=2,N
T=ABS(C(I))+ABS(B(I))+ABS(B(I-1))
IF(T.GT.TOP) TOP=T
20 CONTINUE
DO 21 I=1,N
21 W(I)=B(I)**2
DO 22 I=1,NVAL
22 E(I)=-TOP
SUM=0.
```

```
DO 38 K=1,NVAL
  BOT=E(K)
  DO 40 ITER=1,NITR
24  PL=0.5*(TOP+BOT)
    ERR=ABS(TOP-BOT)
    IF(ERR.LT.EPA)GO TO 41
    NAG=0
    I=1
26  S=C(I)-PL
28  IF(S.GE.0.)NAG=NAG+1
    IF(S.EQ.0.) GO TO 30
    I=I+1
    IF(I.GT.N) GO TO 32
    S=C(I)-PL-W(I-1)/S
    GO TO 28
30  I=I+2
    IF(I.LE.N)GO TO 26
32  IF(NAG.GE.K) GO TO 34
    TOP=PL
    GO TO 40
34  BOT=PL
40  CONTINUE
41  E(K)=PL
    SUM=SUM+E(K)
    SS=SUM/SETA
    WRITE(3,202)K,E(K),ERR,ITER,SUM,SS
38  CONTINUE
    WRITE(3,204)NVAL,SUM
    IF(NVEC.NE.0) GO TO 42
    GO TO 80
42  DO 70 I=1,NVEC
    DO 44 J=1,N
    P(J)=0.
    Q(J)=B(J)
    R(J)=C(J)-E(I)
44  Y(J)=1.
    Y(N+1)=0.
    Y(N+2)=0.

C
C   BEGIN THE GAUSSIAN ELIMINATION
C
```

```
IF(R(1).EQ.0.)R(1)=EPC
DO 48 J=1,NMA
IF(ABS(R(J)).LT.ABS(B(J)))GO TO 45
FAC=B(J)/R(J)
IN(J)=0
GO TO 46
45 FAC=R(J)/B(J)
IN(J)=1
R(J)=B(J)
T=R(J+1)
R(J+1)=Q(J)
Q(J)=T
P(J)=Q(J+1)
Q(J+1)=0.
46 W(J)=FAC
Q(J+1)=Q(J+1)-FAC*P(J)
R(J+1)=R(J+1)-FAC*Q(J)
IF(R(J+1).EQ.0.)R(J+1)=EPC
48 CONTINUE
C
C BEGIN THE BAC- SUBSTITUTION
C
DO 60 ITIME=1,NTIME
DO 53 JJ=1,N
K=N+1-JJ
T=Y(K)
51 Y(K)=(T-Y(K+1)*Q(K)-Y(K+2)*P(K))/R(K)
IF(ABS(Y(K)).LT.BIG)GO TO 53
DO 52 J=1,N
52 Y(J)=Y(J)*EPD
T=T*EPD
GO TO 51
53 CONTINUE
C
C THIS COMPLETES BAC- SUBSTITUTION
C DO THE REDUCTION IN NEW R.H.S.
IF(ITIME.EQ.NTIME)GO TO 60
DO 57 J=1,NMA
IF(IN(J).EQ.1) GO TO 56
Y(J+1)=Y(J+1)-W(J)*Y(J)
GO TO 57
```

```

56      T=Y(J)
        Y(J)=Y(J+1)
        Y(J+1)=T-W(J)*Y(J+1)
57      CONTINUE
60      CONTINUE
        DO 64 J=1,NMB
          T=0.
          K=N-J-1
          M=K+1
          DO 62 KK=M,N
62          T=T+A(KK,K)*Y(KK)
            T=A(K,K)*T
            DO 64 KK=M,N
64          Y(KK)=Y(KK)-T*A(KK,K)
            SS=0.
            DO 67 J=1,N
67          SS=SS+Y(J)**2
            T=1./SQRT(SS)
            DO 66 J=1,N
66          V(J,I)=Y(J)*T
70          CONTINUE
            DO 71 I=1,NVEEC
              DO 71 J=I,NVEEC
                A(I,J)=0.
                DO 71 M=1,N
                  A(I,J)=A(I,J)+V(M,I)*V(M,J)
71          A(J,I)=A(I,J)
              WRITE(3,206)
              DO 73 J=1,NVEEC
67          WRITE(3,212)J,(V(I,J),I=1,N)
              WRITE(3,207)
              DO 74 J=1,NVEEC
74          WRITE(3,200)J,(A(I,J),I=1,NVEEC)
80          CONTINUE
200         FORMAT(I5,10E13.4/(5X,10E13.4))
202         FORMAT(2X,15HEIGEN VALUE NO. ,I4,E13.4,2X,17HWITH ERROR LIMITS,E13
1          .4,2X,5HAFTER,I4,2X,10HITERATIONS ,2E13.4/)
203         FORMAT(/5X,42HSUM OF THE DIAGONAL ELEMENTS OF MATRIX A= ,E15.4//
1          /T21,1HI,T30,4HE(I),T94,8HSUM E(I),T106,15HCLOSENESS RATIO/)
204         FORMAT(5X,6HSUM OF ,I3,2X,13HEIGEN VALUES= ,E15.4/)
206         FORMAT(/, 5X,13HEIGEN VECTORS )

```

```
207     FORMAT(/,5X,24HMATRIX- V(TRANPOSE) X V )
212     FORMAT(I5,10E13.3/(5X,10E13.3))
      RETURN
      END
      SUBROUTINE VARMAX(F,COM,NF,NN,T)
C
C     THIS ROUTINE IS FROM Klovan and Imbrie [1971]
C
      parameter (MC=10000,MF=20)
      DIMENSION F(MC,MF),COM(MC),T(MF,MF)
      EPS=0.06993
      DO 11 I=1,NF
      DO 10 J=1,NF
10      T(I,J)=0.
11      T(I,I)=1.
      DO 13 I=1,NN
      COM(I)=0.
      DO 12 J=1,NF
12      COM(I)=COM(I)+F(I,J)**2
      COM(I)=SQRT(COM(I))
      DO 13 J=1,NF
13      F(I,J)=F(I,J)/COM(I)
      NFA=NF-1
14      NROT=0
      DO 40 M=1,NFA
      K=M+1
      DO 40 L=K,NF
      A=0.
      B=0.
      C=0.
      D=0.
      DO 15 I=1,NN
      U=F(I,M)**2-F(I,L)**2
      V=F(I,M)*F(I,L)*2
      A=A+U
      B=B+V
      C=C+U**2-V**2
15      D=D+U*V*2
      QNUM=D-2.*A*B/NN
      QDEN=C-(A**2-B**2)/NN
      IF(ABS(QNUM)+ABS(QDEN).EQ.0.)GO TO 30
```

```
IF(ABS(QNUM)-ABS(QDEN))17,24,21
17 R=ABS(QNUM/QDEN)
   IF(R.LT.EPS)GO TO 19
   CS4TH=COS(ATAN(R))
   SN4TH=SIN(ATAN(R))
   GO TO 25
19 IF(QDEN.GE.0.)GO TO 30
   SNPHI=0.70710678
   CSPHI=SNPHI
   GO TO 31
21 R=ABS(QDEN/QNUM)
   IF(R.LT.EPS)GO TO 23
   SN4TH=1./SQRT(1.+R**2)
   CS4TH=SN4TH*R
   GO TO 25
23 CS4TH=0.
   SN4TH=1.
   GO TO 25
24 CS4TH=0.70710678
   SN4TH=CS4TH
25 R=SQRT((1.+CS4TH)*.5)
   CSTH=SQRT((1.+R)*0.5)
   SNTH=SN4TH/(4.*CSTH*R)
   IF(QDEN.GE.0.)GO TO 27
   CSPHI=.70710678*(CSTH+SNTH)
   SNPHI=.70710678*(CSTH-SNTH)
   GO TO 28
27 CSPHI=CSTH
   SNPHI=SNTH
28 IF(QNUM.GE.0.)GO TO 31
   SNPHI=-SNPHI
   GO TO 31
30 NROT=NROT+1
   GO TO 40
31 DO 32 J=1,NN
   R=F(J,M)*CSPHI+F(J,L)*SNPHI
   F(J,L)=F(J,L)*CSPHI-F(J,M)*SNPHI
   F(J,M)=R
   IF(J.GT.NF)GO TO 32
   TP=T(J,M)
   T(J,M)=TP*CSPHI+T(J,L)*SNPHI
```

```

T(J,L)=-TP*SNPHI+T(J,L)*CSPHI
32 CONTINUE
40 CONTINUE
IF(NROT.LT.NF*NFA/2)GO TO 14
DO 42 I=1,NN
DO 41 J=1,NF
41 F(I,J)=F(I,J)*COM(I)
42 COM(I)=COM(I)**2
RETURN
END
SUBROUTINE TIME
c CALL GETTIM(IHR,IMIN,ISEC,I100TH)
c CALL GETDAT(IYR,IMON,IDAY)
c WRITE(*,1)IDAY,IMON,IYR
c WRITE(*,2)IHR,IMIN
c WRITE(3,1)IDAY,IMON,IYR
c WRITE(3,2)IHR,IMIN
c1 FORMAT(T10,'DATE',2X,I2.2,'/',I2.2,'/',I4)
c2 FORMAT(T10,'TIME ',I3,':',I2.2,3X,'HOURS'////)
RETURN
END
SUBROUTINE DATAL(X,NR,NC)
parameter (MR=1024,MC=10000)
DIMENSION X(MR,MC)
CHARACTER*1 ANS
WRITE(*,1700)
1700 format(' Is the matrix read rowwise or columnwise? (r/c): ', $)
CALL ENTRY(IWX,2,'r','c','3','4','5')
IF(iwx.eq.2) GO TO 100
DO 1 I=1,NR
1 READ(2,*)(X(I,J),J=1,NC)
RETURN
100 CONTINUE
DO 2 J=1,NC
2 READ(2,*)(X(I,J),I=1,NR)
RETURN
END
SUBROUTINE ENTRY(IWT,NP,X1,X2,X3,X4,X5)
CHARACTER*1 ICHR(5),X0,X1,X2,X3,X4,X5
IWT=0
ICHR(1)=X1

```



```

ICHR(2)=X2
ICHR(3)=X3
ICHR(4)=X4
ICHR(5)=X5
READ(*,'(A1)')X0
DO I=1,NP
IF(X0.EQ.ICHR(I))IWT=I
END DO
IF(IWT.EQ.0)STOP ' MISKEY !'
RETURN
END

```

### A.1.2 Specification File: eof-vec.spc

Output file	(A25): see
Input file	(A25): sla-pacr.dat
Format for input file	(A25): (8785e16.7)
Number of stations	(I5) : 08785
Number of time steps	(I5) : 00132
Number of factors	(I5) : 00010
Lat Long file	(A25): sla-pac-lat-long.dat
Format for Lat Long file header	(A25): (1a75)
IPRINTALL (0/1)	(I5) : 00000

: ↑ input is from column number 51

Choose IPRINTALL=1 for extensive printout

Choose IPRINTALL=0 for normal printout

Latitude Longitude file is not used for computations  
but just to help later in plotting the spatial EOFs, viz.,  
the output files (spatial): matrix-v, matrix-v-rotated  
Output files (temporal) are : matrix-u, matrix-u-rotated

## A.2 Hilbert Transform

### A.2.1 Computer Program: hilbert.f

```

PROGRAM ALPHA
PARAMETER(IM1=4096)
DIMENSION BXR(IM1),BXI(IM1),XX(IM1),YY(IM1),YF(IM1),AAA(IM1,10000)
DIMENSION BBB(IM1,10000)
DIMENSION LABEL(80)
CHARACTER*12 INFIL,FLTFIL,OUTFIL,CPCFIL
CHARACTER*40 AFMT
CHARACTER*25 FILTYPE(3)
DATA FILTYPE/'LOW PASS','BAND PASS','HILBERT TRANSFORM'/
OPEN(15,FILE='hilbert.spc',ACCESS='SEQUENTIAL',BLANK='ZERO')
CALL SPECR(OUTFIL,NT,NLEAVE,NOUT,NX,INFIL,AFMT,SAMIN,
# PER0,PER1,IFIL)

c      WRITE(*,678)OUTFIL,NT,NLEAVE,NOUT,NX,INFIL,AFMT,SAMIN,IFIL
678    FORMAT( 5X,'OUTPUT FILE ... ',A12/
#      5X,'NUMBER OF POINTS IN SERIES... ',I5/
#      5X,'NUMBER OF POINTS TO LEAVE... ',I5/
#      5X,'NUMBER OF POINTS TO PRINT... ',I5/
#      5X,'NUMBER OF STATIONS ... ',I5/
#      5X,'NAME OF INPUT DATA FILE ... ',A25/
#      5X,'FORMAT FOR INPUT FILE ... ',A25/
#      5X,'SAMPLING INTERVAL ... ',f10.4/
#      5X,'IFIL ... ',I5)

C      NT=300
C      NX=11

C      NT=72
C      NX=221

nstart=nleave+1
nend=nleave+nout
LIST=0
NROW=NT
NCOL=NX

CALL DATAM(AAA,NROW,NCOL,AFMT,INFIL,LABEL)

```

```

C      CALL DATA(YY,NP,INFIL,AFMT,LABEL)

      DO 6767 J=1,NX
      DO I=1,NT
      YY(I)=AAA(I,J)
C      WRITE(*,*) ' I= ',I, ' YY(I) = ',YY(I)
      END DO
      NP=NT

      IF(J.EQ.1)THEN
      CALL FILT_FIL(INFIL,FLTFIL)
      CALL CILT_FIL(INFIL,CPCFIL,ifil)
      WRITE(*,255)FLTFIL,CPCFIL
255  FORMAT(5X,'Filtered file (.FDF) : ',a12/
# 5X,'FILTERED FILE (*.at) : ',a12)
      OPEN(3,FILE=FLTFIL,ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      OPEN(37,FILE=CPCFIL,ACCESS='SEQUENTIAL',STATUS='UNKNOWN')
      WRITE(*,5656)FILTYPE(IFIL)
      WRITE(10,5656)FILTYPE(IFIL)
      END IF

5656  FORMAT(5X,A25,'FILTER USED')
      DO I=1,NP
      BXR(I)=YY(I)
      BXI(I)=0.
      END DO

10    FORMAT(5(I4,2X,F7.2,2X,F7.2))

      SGN=-1.
      call newfft(bxr,bxi,sgn,list,np)
C      WRITE(3,10)(I,BXR(I),BXI(I),I=1,NP)
c      IF(IFIL.EQ.1)CALL LPFIL1(NP,BXR,BXI,SAMIN,PER0,PER1)
      IF(IFIL.EQ.1)CALL LPFILWHITE(NP,BXR,BXI,SAMIN,PER0,PER1)
      IF(IFIL.EQ.2)CALL BPFIL1(NP,BXR,BXI,SAMIN,PER0,PER1)
      IF(IFIL.EQ.3)CALL HILTRA(NP,BXR,BXI)
      SGN=1.
      call newfft(bxr,bxi,sgn,list,np)

```

```

        DO I=1,NP
        YF(I)=BXR(I)/NP
        END DO

CC      WRITE(3,310)(BXR(I),BXI(I),I=1,NP)
C       WRITE(3,10)(BXR(I),I=1,NP)
C       WRITE(3,311)(LABEL(I),I=1,80)
CC      WRITE(3,312)(BXR(I),I=1,NP,6)
CC      WRITE(3,312)(BXR(I),I=1,NP,36)
C       WRITE(3,312)(YF(I),I=1,NP,NOUT)
311     FORMAT(20A4)
312     FORMAT(24F7.2)
        WRITE(3,123)(LABEL(I),I=1,80)
        WRITE(10,123)(LABEL(I),I=1,80)
c       WRITE(3,7159)J
c       WRITE(10,7159)J
7159    FORMAT(2X,'##### SPATIAL ELEMENT, J = ',I4)
        DO I=1,NP
        XX(I)=(I-1)*SAMIN
        END DO
c       WRITE(3,110)(I,XX(I),YY(I),YF(I),I=1,NP)
c       WRITE(10,110)(I,XX(I),YY(I),YF(I),I=1,NP)
110     FORMAT(I5,5X,F10.4,5X,F7.2,5X,F8.3)
C       WRITE(*,987)
C987    FORMAT(5X,'Enter number of points for plot :',)$)
C       READ(*,*)NAB
310     FORMAT(12F11.3)
123     FORMAT(20A4)
        DO I=1,NT
        BBB(I,J)=YF(I)
        END DO
6767    CONTINUE
        WRITE(37,123)(LABEL(I),I=1,80)
c       DO I=1,NT
        DO I=NSTART,NEND
        WRITE(37,5738)(AAA(I,J),J=1,NX),(BBB(I,J),J=1,NX)
        END DO
c5738   FORMAT(22F8.4)
5738    FORMAT(20000e16.7)

```

```
        CLOSE(15)
        CLOSE(14)
        CLOSE(10)
        CLOSE(11)
        CLOSE(3)
        WRITE(*,254)INFIL,OUTFIL,FLTFILE,CPCFIL
254      FORMAT(5X,' FILE DIARY '//
#       5X,'#####'//
#       5X,'Input file : ',a12/
#       5X,'Output file : ',a12//
#       5X,'Filtered file : ',a12/
#       5X,'Filtered file : ',a12)

C       IF(NOUT.EQ.1)THEN

        STOP
        END
        SUBROUTINE LPFIL1(NP,BXR,BXI,SAMIN,PER0,PER1)
        PARAMETER(IM1=4096)
        DIMENSION BXR(IM1),CORR(IM1),BXI(IM1)
        F0=1./PER0
        F1=1./PER1
        NBY2=NP/2

        DO I=1,NBY2
        CORR(I)=0.
        END DO
        CORR(NBY2+1)=0.

        I=0
        FREQ=0.
        DO WHILE (FREQ.LT.F1)
        IF (FREQ.GT.F0) THEN
        CORR(I+1)=(F1-FREQ)/(F1-F0)
        ELSE
        CORR(I+1)=1.
        END IF
        I=I+1
C       FREQ=(I-1)/SAMIN/NP
        FREQ=I/SAMIN/NP
```

```

END DO
JH=I+10
WRITE(10,10)(J,CORR(J),J=1,JH)
10  FORMAT(5X,'FILTER COEFFICIENTS'/(4(I3,2X,F8.6,5X)))
DO ILKJ=2,NBY2
BXR(ILKJ)=BXR(ILKJ)*CORR(ILKJ)
BXI(ILKJ)=BXI(ILKJ)*CORR(ILKJ)
JJJL=NP-ILKJ+2
BXR(JJJL)=BXR(JJJL)*CORR(ILKJ)
BXI(JJJL)=BXI(JJJL)*CORR(ILKJ)
END DO

RETURN
END

SUBROUTINE LPFILWHITE(NP,BXR,BXI,SAMIN,PER0,PER1)
PARAMETER(IM1=4096)
DIMENSION BXR(IM1),CORR(IM1),BXI(IM1)
dimension fff(6),ttt(6),resf(6)
c  data ttt/1.e07,1000,30,24,20,15/
data ttt/1.e07,72,30,24,20,15/
data resf/1.,1.,0.7,0.5,0.175,0./

NBY2=NP/2

DO I=1,NBY2
CORR(I)=0.
END DO
CORR(NBY2+1)=0.

nold=6
fff(1)=0
do i=2,nold
fff(i)=1./ttt(i)
end do

do i=1,nby2

```

```

        if(i.gt.(np*samin*fff(6))) go to 1
        end do
1       nnew=i
c       write(*,*)' nnew= ',nnew
        write(10,*)' nnew= ',nnew

        do 200 j=1,nnew
        freq=(j-1)/(np*samin)
        do 300 k=1,nold
        if(freq-fff(k))600,450,300
450     corr(j)=resf(k)
        go to 200
600     continue
        corr(j)=resf(k)-(fff(k)-freq)*(resf(k)-resf(k-1))/
1       (fff(k)-fff(k-1))
        go to 200
300     continue
200     continue

        JH=nnew+10
        WRITE(10,10)(J,CORR(J),J=1,JH)
c       WRITE(*,10)(J,CORR(J),J=1,JH)
10      FORMAT(5X,'FILTER COEFFICIENTS'/(4(I3,2X,F8.6,5X)))
        DO ILKJ=2,NBY2
        BXR(ILKJ)=BXR(ILKJ)*CORR(ILKJ)
        BXI(ILKJ)=BXI(ILKJ)*CORR(ILKJ)
        JJJL=NP-ILKJ+2
        BXR(JJJL)=BXR(JJJL)*CORR(ILKJ)
        BXI(JJJL)=BXI(JJJL)*CORR(ILKJ)
        END DO

        RETURN
        END

SUBROUTINE BPFIL1(NP,BXR,BXI,SAMIN,PER0,PER1)
PARAMETER(IM1=4096)
DIMENSION BXR(IM1),CORR(IM1),BXI(IM1)
F0=1./PER0

```

```

F1=1./PER1
NBY2=NP/2

DO I=1,NBY2
CORR(I)=0.
END DO
CORR(NBY2+1)=0.

I=0
FREQ=0.
DO WHILE (FREQ.LE.F1)
IF (FREQ.GE.F0) THEN
CORR(I+1)=1.0
ELSE
CORR(I+1)=0.
END IF
I=I+1
C   FREQ=(I-1)/SAMIN/NP
   FREQ=I/SAMIN/NP
END DO
JH=I+10
WRITE(10,10)(J,CORR(J),J=1,JH)
10  FORMAT(5X,'FILTER COEFFICIENTS'/(4(I3,2X,F8.6,5X)))
DO ILKJ=2,NBY2
BXR(ILKJ)=BXR(ILKJ)*CORR(ILKJ)
BXI(ILKJ)=BXI(ILKJ)*CORR(ILKJ)
JJJL=NP-ILKJ+2
BXR(JJJL)=BXR(JJJL)*CORR(ILKJ)
BXI(JJJL)=BXI(JJJL)*CORR(ILKJ)
END DO

RETURN
END

SUBROUTINE SPECR(OUTFIL,NP,NLEAVE,NOUT,NX,INFIL,AFMT,SAMIN,
#  PER0,PER1,IFIL)

CHARACTER*4 DUMMY
CHARACTER*12 INFIL,OUTFIL
CHARACTER*40 AFMT

```



```

CHARACTER*25 FILTYPE(3)
CHARACTER*1 AMY
DATA FILTYPE/'LOW PASS','BAND PASS','HILBERT TRANSFORM'/
READ(15,'(50X,1A1)')AMY
READ(15,'(50X,1A12)')OUTFIL
write(*,'(50X,1A12)')OUTFIL

READ(15,'(50X,I5)')NP
READ(15,'(50X,I5)')NLEAVE
READ(15,'(50X,I5)')NOUT
READ(15,'(50X,I5)')NX
READ(15,'(50X,1A12)')INFIL
READ(15,'(50X,1A40)')AFMT
OPEN(10,FILE=OUTFIL,ACCESS='SEQUENTIAL',status='unknown')
546 WRITE(10,678)OUTFIL,NP,NLEAVE,NOUT,NX,INFIL,AFMT
WRITE(*,678)OUTFIL,NP,NLEAVE,NOUT,NX,INFIL,AFMT
678 FORMAT( 5X,'OUTPUT FILE ... ',A12/
# 5X,'NUMBER OF POINTS IN SERIES ... ',I5/
# 5X,'NUMBER OF POINTS TO LEAVE ... ',I5/
# 5X,'NUMBER OF POINTS TO PRINT ... ',I5/
# 5X,'NUMBER OF STATIONS ... ',I5/
# 5X,'NAME OF INPUT DATA FILE ... ',A25/
# 5X,'FORMAT FOR INPUT FILE ... ',A25)

read(15,'(50x,1f7.6)')samin
READ(15,'(50X,A1)')AMY
READ(15,'(50X,1F7.6)')PER0
READ(15,'(50X,1F7.6)')PER1

IF(AMY.EQ.'L')THEN
IFIL=1
ELSE IF (AMY.EQ.'B') THEN
IFIL=2
ELSE IF (AMY.EQ.'H') THEN
IFIL=3
ELSE
STOP 'Type of filter should be Low or Band pass'
END IF

5463 WRITE(10,6783)SAMIN,FILTYPE(IFIL),
# PER0,PER1

```

```

6783      FORMAT( 5X,'SAMPLING INTERVAL ... ',F10.3/
#         5X,'FILTER TYPE ... ',A25/
#         5X,'PERIOD ZERO ... ',F10.3/
#         5X,'PERIOD ONE ... ',F10.3)

      RETURN
      END

      SUBROUTINE DATA(BXR,NP,INFIL,AFMT,LABEL)
      PARAMETER(IM1=4096)
      DIMENSION BXR(IM1)
      CHARACTER*40 AFMT
      CHARACTER*12 INFIL
      DIMENSION LABEL(80)
      OPEN(14,FILE=INFIL,ACCESS='SEQUENTIAL',BLANK='ZERO')
      READ(14,10)(LABEL(I),I=1,80)
      WRITE(*,10)(LABEL(I),I=1,80)
      READ(14,AFMT)(BXR(I),I=1,NP)
      WRITE(*,345)(BXR(I),I=1,48)
345      FORMAT(12F6.1)
      WRITE(*,*)'///'
      WRITE(*,1234)
1234     FORMAT(5X,'ARE THESE THE FIRST 4+2 LINES OF X SERIES : ', $)

      CALL ENTRY(IWW,2,'Y','N','3','4','5')

      IF(IWW.EQ.2)STOP' CHECK YOUR INPUT SERIES !!'
      WRITE(10,10)(LABEL(I),I=1,80)
      WRITE(10,AFMT)(BXR(I),I=1,NP)
10      FORMAT(20A4)
      RETURN
      END

      SUBROUTINE DATAM(AAA,NROW,NCOL,AFMT,INFIL,LABEL)
      PARAMETER(IM1=4096)
      DIMENSION AAA(IM1,10000)
      CHARACTER*40 AFMT
      CHARACTER*12 INFIL
      DIMENSION LABEL(80)
      OPEN(14,FILE=INFIL,ACCESS='SEQUENTIAL',BLANK='ZERO')
      READ(14,10)(LABEL(I),I=1,80)

```

```

WRITE(*,10)(LABEL(I),I=1,80)
10  FORMAT(20A4)
WRITE(*,1234)
1234  FORMAT(5X,'IS THIS THE 4 LINE LABEL OF YOUR INPUT DATA? (Y/N) ', $)
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP 'CHECK YOUR INPUT DATA!'

WRITE(*,1235)
1235  FORMAT(5X,'IS THE INPUT MATRIX READ ROWWISE OR COLUMNWISE? (R/C) '
#  $)
CALL ENTRY(IWW,2,'R','C','3','4','5')
c  IF(IWW.EQ.2) STOP 'CHECK YOUR INPUT DATA!'

IF(IWW.EQ.1)then
DO I=1,NROW
READ(14,AFMT)(AAA(I,J),J=1,NCOL)
IF(I.EQ.1)THEN
WRITE(*,203)I,(AAA(I,J),J=1,NCOL)
203  FORMAT(2X,I3,2X,5E12.4/(7X,5E12.4))
WRITE(*,1236)
1236  FORMAT(5X,'IS THIS THE FIRST ROW OF YOUR INPUT MATRIX? (Y/N) ', $)
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP 'CHECK YOUR INPUT DATA!'
END IF
c  WRITE(10,203)I,(AAA(I,J),J=1,NCOL)
END DO
RETURN

else if(iww.eq.2)then
DO J=1,NCOL
READ(14,AFMT)(AAA(I,J),I=1,NROW)
IF(J.EQ.1)THEN
WRITE(*,203)J,(AAA(I,J),I=1,NROW)
WRITE(*,1237)
1237  FORMAT(5X,'IS THIS THE FIRST COLUMN OF YOUR INPUT MATRIX? (Y/N) ',
#  $)
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP 'CHECK YOUR INPUT DATA!'
END IF
c  WRITE(10,203)J,(AAA(I,J),I=1,NROW)
END DO

```

```
RETURN
end if
```

```
END
```

```
SUBROUTINE ENTRY(IWT,NP,X1,X2,X3,X4,X5)
CHARACTER*1 ICHR(5),X0,X1,X2,X3,X4,X5
IWT=0
ICHR(1)=X1
ICHR(2)=X2
ICHR(3)=X3
ICHR(4)=X4
ICHR(5)=X5
```

```
READ(*,'(A1)')X0
DO I=1,NP
IF(X0.EQ.ICHR(I))IWT=I
END DO
IF(IWT.EQ.0)STOP ' MISKEY !'
RETURN
END
```

```
SUBROUTINE FILT_FIL(INFIL,FLTFIL)
CHARACTER*12 INFIL,FLTFIL
CHARACTER*1 DOT,BLANK
DATA DOT/'./',BLANK/' '/
DO J=1,12
FLTFIL(J:J)=' '
END DO
DO I=1,9
IF(INFIL(I:I).NE.DOT.AND.INFIL(I:I).NE.BLANK)THEN
FLTFIL(I:I)=INFIL(I:I)
ELSE
FLTFIL(I:I)=DOT
FLTFIL(I+1:I+3)='FDF'
RETURN
END IF
END DO
return
end
```

```
SUBROUTINE CILT_FIL(INFIL,CPCFIL,ifil)
CHARACTER*12 INFIL,CPCFIL
CHARACTER*1 DOT,BLANK
DATA DOT/'./,BLANK/' /
DO J=1,12
CPCFIL(J:J)=' '
END DO
DO I=1,9
IF(INFIL(I:I).NE.DOT.AND.INFIL(I:I).NE.BLANK)THEN
CPCFIL(I:I)=INFIL(I:I)
ELSE
CPCFIL(I:I)=DOT
if(ifil.eq.1)CPCFIL(I+1:I+3)='lat'
if(ifil.eq.2)CPCFIL(I+1:I+3)='bat'
if(ifil.eq.3)CPCFIL(I+1:I+3)='hat'

RETURN
END IF
END DO
RETURN
END
```

```
SUBROUTINE HILTRA(NP,BXR,BXI)
PARAMETER(IM1=4096)
DIMENSION BXR(IM1),BXI(IM1)
NBY2=NP/2
DO 8436 ILKJ=2,NBY2
FFREA=BXR(ILKJ)
FFIMA=BXI(ILKJ)
```

8436

```
BXR(ILKJ)=FFIMA
BXI(ILKJ)=-FFREA
JJJL=NP-ILKJ+2
BXR(JJJL)=BXR(ILKJ)
BXI(JJJL)=-BXI(ILKJ)
CONTINUE
```

```
RETURN
END
```

```
subroutine newfft(xr,xi,sgn,list,np)
parameter (im1=4096,im1by21=im1/2+1)
dimension xr(im1),xi(im1),wc(im1,im1),ws(im1,im1)
dimension yr(im1),yi(im1),gx(im1by21)

if(list.gt.0) go to 1
nby2=np/2
nby21=nby2+1
write(*,*)np,nby2,nby21

twopin=atan(1.)*4.*2./np

do k=1,np
do ni=1,np
phi=twopin*(k-1)*(ni-1)
wc(k,ni)=cos(phi)
ws(k,ni)=sin(phi)
end do

end do
list=1
1 continue

if(sgn.lt.0.)then
do k=1,np
sumc=0.
sums=0.
do ni=1,np
sumc=sumc+xr(ni)*wc(k,ni)+xi(ni)*ws(k,ni)
sums=sums+xi(ni)*wc(k,ni)-xr(ni)*ws(k,ni)
end do
yr(k)=sumc
yi(k)=sums
end do

amul=4./np/np
do i=1,nby21
gx(i)=amul*(yr(i)**2+yi(i)**2)
end do
write(10,*)' GX'
write(10,20)(gx(i),i=1,nby21)
```

```
        do i=1,np
            xr(i)=yr(i)
            xi(i)=yi(i)
        end do

c        write(*,*) ' XR'
20       format(4e14.4)
c        write(*,20)(xr(i),i=1,np)
c        write(*,*) ' XI'
c        write(*,20)(xi(i),i=1,np)

        else

            do k=1,np
                sumc=0.
                sums=0.
                do ni=1,np
                    sumc=sumc+xr(ni)*wc(k,ni)-xi(ni)*ws(k,ni)
                    sums=sums+xi(ni)*wc(k,ni)+xr(ni)*ws(k,ni)
                end do
                yr(k)=sumc
                yi(k)=sums
            end do

            do i=1,np
                xr(i)=yr(i)
                xi(i)=yi(i)
            end do

        end if

        return
    end
```

### A.2.2 Specification file: `hilbert.spc`

C TAB 51

Output file	(A12)	:	seeh
Time series length	(I5) NP=	:	00300
	(I5) NLEAVE=	:	00030
	(I5) NOUT=	:	00240
Number of stations	(I5)	:	00011
Input series	(A12) INFILX	:	ceof-ex3.dat
Format	(A40)	:	(11f8.4)
Sampling interval	(F7.6)	:	1.0
Type of filt (L/B/H)	(A1)	:	H
Period (=1)	(F7.6)	:	40.
Period (=0)	(F7.6)	:	30.

: ↑ input is from column number 51

If type of filter chosen is “L” and *subroutine lpfil1*

is called with sampling interval in hours and Period (=1)

and Period (=0) taking values as given above, the output will be low pass

filtered with a linear taper discussed by Walters and Heston [1982] for removing tidal periodicities

If type of filter chosen is “L” and *subroutine lpfilwhite*

is called with sampling interval = 1, in months, the

output will be low pass filtered data with filter characteristics discussed by White [2000a]

Note how the provision for deleting one tenth of the

time series (=30 points) at the beginning and end of the

time series is made by proper choice of NP, NLEAVE & NOUT



## A.3 Complex EOF Analysis

### A.3.1 Computer Program: ceof-vec.f

```

C      PROGRAM COMPLEX E-O-F
c      PARAMETER (MAXR=300,MAXC=11,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
c      PARAMETER (MAXR=512,MAXC=500,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
c      PARAMETER (MAXR=4320,MAXC=500,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
      PARAMETER (MAXR=1024,MAXC=10000,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
      DIMENSION U(MAXWR,MAXF), V(MAXWC,MAXF),WW(MAXWR,MAXWR),E(MAXF)
      dimension fac(maxc)
      dimension utu(maxf,maxf),vtv(maxf,maxf)
      DIMENSION DR(MAXR,MAXC),DI(MAXR,MAXC)
      dimension drmean(maxc),dimean(maxc)
      dimension ds(maxr,maxc)
      dimension AMPV(MAXWC,MAXF),PHASEV(MAXWC,MAXF)
      dimension AMPU(MAXWR,MAXF),PHASEU(MAXWR,MAXF)
      dimension label(80)
      dimension hsq(maxc),diff(maxf),xdiff(maxf)
      dimension comc(maxc),t(maxf,maxf)
      dimension ttt(maxf,maxf)
      dimension percv(maxf),cperc(v(maxf))
      real lat(maxc),long(maxc)
      complex VV(MAXC,MAXF),vvtvv(maxf,maxf),vvsum
      character*28 recomm,xrecomm(maxf)
      character*36 recommv,xrecommv(maxf)
      character dummy*75
      CHARACTER*60 AFMT,BFMT,OUTFIL,INFIL,ltnfile
      OPEN(15,file='ceof-cpc.spc',access='sequential',blank='zero')
      CALL SPECR(OUTFIL,INFIL,AFMT,NX,NT,NFAC,LTLNFILE,BFMT)
      open(3,file=OUTFIL,access='sequential',status='unknown')
      WRITE(3,5834)OUTFIL,INFIL,AFMT,NX,NT,NFAC,LTLNFILE,BFMT
      WRITE(*,5834)OUTFIL,INFIL,AFMT,NX,NT,NFAC,LTLNFILE,BFMT
5834  FORMAT(5X,'===== COMPLEX EOF =====')
      # 5X,'OUTPUT FILE ',1A60/
      # 5X,'INPUT FILE ',1A60/
      # 5X,'INPUT FILE FORMAT ',1A60/
      # 5X,'NUMBER OF STATIONS ',I5/
      # 5X,'NUMBER OF TIME STEPS ',I5/
      # 5X,'NUMBER OF FACTORS ',I5/
      # 5X,'LAT LONG FILE ',1A60/
      # 5X,'FORMAT FOR LAT LONG ',1A60)

```

```
OPEN(68,FILE=LTLNFILE,blank='zero',
#  access='sequential')

read(68,'(1a75)')dummy
write(3,'(1a75)')dummy
do j=1,nx
read(68,*)jc,lat(j),long(j)
write(3,*)jc,lat(j),long(j)
if (jc.ne.j)stop 'lat-long data reading error'
end do

102  FORMAT(20000E16.7)

NROW=NT
NCOL=NX
call DATAM(dr,di,NROW,NCOL,AFMT,INFIL,LABEL)

write(*,*)' Data reading is over!'

iogb=1
iogb=0
if(iogb.eq.1)go to 7295
do j=1,ncol
drmean(j)=0.
dimean(j)=0.
do i=1,nrow
drmean(j)=drmean(j)+dr(i,j)
dimean(j)=dimean(j)+di(i,j)
end do
drmean(j)=drmean(j)/nrow
dimean(j)=dimean(j)/nrow
do i=1,nrow
dr(i,j)=-drmean(j)+dr(i,j)
```

```
di(i,j)=-dimean(j)+di(i,j)
end do
end do

write(*,*)' Zero-meaning is over!'

7295 continue

nx2=2*nx
nt2=2*nt

DO 12 I=1,nt
DO 12 J=1,nt
SUMR = 0.
SUMI = 0.
DO 10 K=1,nx
SUMR = SUMR + DR(I,K)*DR(J,K) + DI(I,K)*DI(J,K)
SUMI = SUMI - DR(I,K)*DI(J,K) + DI(I,K)*DR(J,K)
10 CONTINUE
WW(I,J)=SUMR
WW(I+nt,J+nt)=SUMR
int=i+nt
WW(int,J)=sumi
jnt=j+nt
WW(I,JNT)=-sumi

12 CONTINUE

sum=0.
do i=1,nt
sum=sum+ww(i,i)
end do

sumdby2=sum/2.
write(3,4709)sum
write(*,4709)sum
4709 format(' sum = ',1e14.5)

do i=1,nt2
do j=1,nt2
```

```

ww(i,j)=ww(i,j)/sum
end do
end do

c      write(3,106)
c 106  format(3x,'THE CORRELATION MATRIX WW')
c      DO I=1,NT2
c      WRITE(3,109)I,(WW(I,J),J=1,NT2)
c 109  FORMAT(I4,5E13.4/(4X,5E13.4))
c      END DO

C      GENERATE MATRIX WW(I,J)

C      CALCULATE EIGEN VALUES AND EIGEN VECTORS OF 'WW'
c      NFAC=8
c      NVAL=NFAC
c      NVEC=NFAC

      CALL EIGAN(WW,U,E,NT2,NVAL,NVEC,SETA)

      write(*,*)' Subroutine eigan call is over!'

      open(9,file='uv-mat-plot-1-2',status='unknown',
#      access='sequential')
c      WRITE(3,1649)
c      WRITE(9,1649)
c 164  9  FORMAT(2X,'MATRIX V IN PLOTTABLE FORMAT')
c      DO J=1,NT
c      WRITE(3,1650)J,(U(J,IVEC),IVEC=1,NVEC),(U(J+NT,IVEC),IVEC=1,NVEC)
c      WRITE(9,1650)J,(U(J,IVEC),IVEC=1,NVEC),(U(J+NT,IVEC),IVEC=1,NVEC)
c      END DO
1650  FORMAT(I5,8F8.4)

cccc  c  c utu matrix nfac ccccccccccccccccccccccccccccccccccc
      do i=1,nfac
      do j=i,nfac
      ss=0.
      do k=1,nt2

```

```

        ss=ss+u(k,i)*u(k,j)
        end do
        utu(i,j)=ss
        utu(j,i)=ss
        end do
    end do

c        write(3,*)' utu matrix (nfac)'
c        do i=1,nfac
c        write(3,1650)i,(utu(i,j),j=1,nfac)
c        end do

cccccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        write(*,*)' okay2233'
        nfac2=nfac/2

        do i=1,nt2
        jj=0
        do j=1,nfac,2
        jj=jj+1
        U(i,jj)=U(i,j)
        end do
        end do

        jj=0
        do j=1,nfac,2
        jj=jj+1
        e(jj)=e(j)
        end do

c        DO J=1,NT
c        WRITE(3,1650)J,(U(J,IVEC),IVEC=1,NFAC2),(U(J+NT,IVEC),
c        # IVEC=1,NFAC2)
c        WRITE(9,1650)J,(U(J,IVEC),IVEC=1,NFAC2),(U(J+NT,IVEC),
c        # IVEC=1,NFAC2)
c        END DO

cccccc  c  cutu matrix nfac2 ccccccccccccccccccccccccccccccccccccccccc
        do i=1,nfac2
        do j=i,nfac2

```

```

      ss=0.
      do k=1,nt2
      ss=ss+U(k,i)*U(k,j)
      end do
      utu(i,j)=ss
      utu(j,i)=ss
      end do
      end do

      write(3,*)' utu matrix (nfac2)'
      do i=1,nfac2
      write(3,1650)i,(utu(i,j),j=1,nfac2)
      end do
      write(*,*)' okay323'
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      DO 19 I=1,NX
      DO 14 IVAL=1,NFAC2
      V(i,ival)=0.
      V(i+NX,ival)=0.

      DO 14 J=1,NT
      V(I,IVAL)=V(I,IVAL)+DR(J,I)*U(J,IVAL)+DI(J,I)*U(J+NT,IVAL)
14     V(I+NX,IVAL)=V(I+NX,IVAL)+DR(J,I)*U(J+NT,IVAL)-DI(J,I)*U(J,IVAL)
19     continue
C      PRINT OUTPUT

cccc  c  c vtv matrix ccccccccccccccccccccccccccccccccccccccccccc

      do i=1,nx2
      do ival=1,nfac2
      v(i,ival)=v(i,ival)/sqrt(e(ival)*sum)
      end do
      end do

```

```

do i=1,nfac2
do j=i,nfac2
ss=0.
do k=1,nx2
ss=ss+v(k,i)*v(k,j)
end do
vtv(i,j)=ss
vtv(j,i)=ss
end do
end do

write(3,*)' vtv matrix'
do i=1,nfac2
write(3,8650)i,(vtv(i,j),j=1,nfac2)
end do
8650 format(i5,10e14.5)
cccccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
do llk=1,nfac2
recomm='comeof-recomputed-matrix.###'
write(recomm(26:28),6789)llk
6789 format(i3.3)
write(*,'(1A30)')recomm
xrecomm(llk)=recomm
open(65,file=recomm,access='sequential',status='unknown')
do i=1,nt
do j=1,nx
ds(i,j)=0.
do k=llk,llk
ds(i,j)=ds(i,j)+(u(i,k)*v(j,k)+u(i+nt,k)*v(j+nx,k))*
# sqrt(sum*e(k))
end do
end do
end do

write(65,507)recomm
507 format(1a28//)
do i=1,nt

```





```

      PI=4.*ATAN(1.)
      CONS=180./PI
1849  format(3x,'amplitude-phase-v')
1949  format(3x,'amplitude-phase-u')
1850  format(i5,30(1e15.5,1f10.2))

c#### # #####
      DO IVAL=1,NFAC2
      DO J=1,NX
      UCOM=V(J,IVAL)
      VCOM=V(J+NX,IVAL)
      pxqrx=CONS*ATAN2(VCOM,UCOM)
      PHASEV(J,IVAL)=XRANGE(pxqrx)
      AMPV(J,IVAL)=SQRT(UCOM**2+VCOM**2)*sqrt(e(ival)*sum/nt)
      END DO
      END DO

      open(19,file='amp-phase-v',status='unknown',
#      access='sequential')
c      write(3,1849)
      write(19,1849)
      do j=1,nx
c      write(3,1850)j,(ampv(j,ivec),phasev(j,ivec),ivec=1,nfac2),
c      # lat(j),long(j)
      write(19,1850)j,(ampv(j,ivec),phasev(j,ivec),ivec=1,nfac2),
#      lat(j),long(j)
      end do

      DO IVAL=1,NFAC2
      DO J=1,NT
      UCOM=U(J,IVAL)
      VCOM=U(J+NT,IVAL)
      pxqrx=CONS*ATAN2(VCOM,UCOM)
      PHASEU(J,IVAL)=xrange(pxqrx)
      AMPU(J,IVAL)=SQRT(UCOM**2+VCOM**2)
      END DO
      END DO

      open(29,file='amp-phase-u',status='unknown',

```

```

      # access='sequential')
c      write(3,1949)
      write(29,1949)
      do j=1,nt
c      write(3,1850)j,(ampu(j,ivec),phaseu(j,ivec),ivec=1,nfac2)
      write(29,1850)j,(ampu(j,ivec),phaseu(j,ivec),ivec=1,nfac2)
      end do
c#### # #####

4651      FORMAT(I5,E13.4)
cccccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      do j=1,nfac2

      asqr=sqrt(e(j))
      do i=1,nx
      aarr=asqr*v(i,j)
      aaii=asqr*v(i+nx,j)
      vv(i,j)=cplx(aarr,aaii)

      end do

      end do

      write(*,*)' nfac, nx =',nfac,nx

      do i=1,nfac2
      do j=1,nfac2
      vvsum=cplx(0.,0.)
      do k=1,nx
      vvsum=vvsum+vv(k,i)*conjg(vv(k,j))
      end do
      vttvv(i,j)=vvsum
      end do
      end do

      write(3,*)' vttvv matrix (amplitude in % )'
      do i=1,nfac2

```

```

write(3,8650)i,(vvtvv(i,j),j=1,nfac2)
end do

do j=1,nfac2
percv(j)=real(vvtvv(j,j))*100.
end do
cpercv(1)=percv(1)
do j=2,nfac2
cpercv(j)=percv(j)+cpercv(j-1)
end do

open(51,file='comeof-perc-var-greek-new',access='sequential',
# status='unknown')
write(51,284)
284 format(///'ORIGINAL, i.e., UNROTATED MODES'/
# '#####')
WRITE(51,285)
285 FORMAT('MODE',t8,'Percentage variance',t35,
# 'Cumulative percentage'/
# 'No.',t8,'explained',t35,
# 'variance explained'//)
286 format(i4,t8,f10.5,t35,f10.5)
do j=1,nfac2
write(51,286)j,percv(j),cpercv(j)
end do

cccc c ccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc computation of Greek "sigma squared" used for determining whether
cc varimax rotation is useful or not!
cc Part I of two parts, part II is lower down
write(*,*)' i hsq(i)'
write(3,*)' i hsq(i)'

do i=1,nx
hsq(i)=0.
do j=1,nfac2
hsq(i)=hsq(i)+vv(i,j)*conjg(vv(i,j))
end do
write(*,3715)i,hsq(i)
write(3,3715)i,hsq(i)

```

```

end do
3715 format(i5,f7.4)
      asumhsq=0.
      do i=1,nx
      asumhsq=asumhsq+hsq(i)
      end do
      write(*,*)' asumhsq = ',asumhsq
      write(3,*)' asumhsq = ',asumhsq

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      CALL VARMAXC(VV,COMC,NFAC2,NX,T)

      write(*,*)' Subroutine varmaxc call is over!'

c      write(3,253)
c      write(9,253)
c 253 format(//2x,'varimax rotation matrix')
c      do i=1,nfac2
c      write(3,1650)i,(t(i,j),j=1,nfac2)
c      end do

c      MATRIX TTT = T (transpose) T

      do i=1,nfac2
      do j=1,nfac2
      ss=0.
      do k=1,nfac2
      ss=ss+t(k,i)*t(k,j)
      end do
      ttt(i,j)=ss
      end do
      end do

      write(3,2253)
      write(9,2253)
2253 format(//2x,'MATRIX TTT = T (TRANSPOSE) T')
      do i=1,nfac2

```

```

        write(3,1650)i,(ttt(i,j),j=1,nfac2)
        end do

c        WRITE(3,4949)
c        WRITE(9,4949)
c 494    9  FORMAT(///2X,'ROTATED MATRIX V IN PLOTTABLE FORMAT one')
c        DO J=1,NX
c        WRITE(3,1650)J,(VV(J,IVEC),IVEC=1,NFAC2)
c        WRITE(9,1650)J,(VV(J,IVEC),IVEC=1,NFAC2)
c        END DO

        do i=1,nfac2
        do j=1,nfac2
        vvsum=cmplx(0.,0.)
        do k=1,nx
        vvsum=vvsum+vv(k,i)*conjg(vv(k,j))
        end do
        vttvv(i,j)=vvsum
        end do
        end do

        write(3,*)' vttvv matrix rotated by t'
        do i=1,nfac2
        write(3,8650)i,(vttvv(i,j),j=1,nfac2)
        end do

        do j=1,nfac2
        percv(j)=real(vttvv(j,j))* 100.
        end do
        cpercv(1)=percv(1)
        do j=2,nfac2
        cpercv(j)=percv(j)+cpercv(j-1)
        end do
        write(51,283)
283    format(///// 'VARIMAX ROTATED MODES' /
#    '#####')
        WRITE(51,285)
        do j=1,nfac2
        write(51,286)j,percv(j),cpercv(j)

```

```

end do

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
cc      computation of Greek "sigma squared" used for determining whether
cc      varimax rotation is useful or not!
cc      Part II

do l=1,nfac2
diff(l)=0.
do i=1,nx
diff(l)=diff(l)+vv(i,l)*conjg(vv(i,l))/hsq(i)
end do
diff(l)=diff(l)/nx
end do

do l=1,nfac2
xdiff(l)=0.
do i=1,nx
xdiff(l)=xdiff(l)+(vv(i,l)*conjg(vv(i,l))/
#  hsq(i)-diff(l))*2
end do
xdiff(l)=xdiff(l)/nx
end do

sigmasq=0.
do l=1,nfac2
sigmasq=sigmasq+xdiff(l)
end do

write(*,*)' sigmasq = ',sigmasq
write(3,*)' sigmasq = ',sigmasq

anew=100.*sigmasq*nfac2/(nfac2-1)

write(*,*)' anew = ',anew
write(3,*)' anew = ',anew
write(51,290)anew

c####  #  #####

```

```

cccc  c  cvtv matrix ccccccccccccccccccccccccccccccccccc

      do i=1,nx
      do j=1,nfac2
      v(i,j)=real(vv(i,j))
      v(i+nx,j)=aimag(vv(i,j))
      end do
      end do

      do i=1,nfac2
      do j=i,nfac2
      ss=0.
      do k=1,nx2
      ss=ss+v(k,i)*v(k,j)
      end do
      vtv(i,j)=ss
      vtv(j,i)=ss
      end do
      end do

      write(3,*)' vtv rotated matrix'
      do i=1,nfac2
      write(3,1650)i,(vtv(i,j),j=1,nfac2)
      end do
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccc
c      WRITE(3,6649)
c      WRITE(9,6649)
c 664  9  FORMAT(2X,'ROTATED MATRIX V IN PROPER PLOTTABLE FORMAT')
c      DO J=1,NX
c      WRITE(3,1650)J,(V(J,IVEC),IVEC=1,NFAC2),
c      # (V(J+NX,IVEC),IVEC=1,NFAC2)
c      WRITE(9,1650)J,(V(J,IVEC),IVEC=1,NFAC2),
c      # (V(J+NX,IVEC),IVEC=1,NFAC2)
c      END DO

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccc

```

```

cccc  c  c utu rotated matrix ccccccccccccccccccccccccccccccc
cccc  r  otated u from T

      do j=1,nt2
      do i=1,nfac2
      fac(i)=0.
      do l=1,nfac2
      fac(i)=fac(i)+u(j,l)*t(l,i)
      end do
      end do
      do i=1,nfac2
      u(j,i)=fac(i)
      end do
      end do

cccc  c  c utu rotated matrix ccccccccccccccccccccccccccccccc
      do i=1,nfac2
      do j=i,nfac2
      ss=0.
      do k=1,nt2
      ss=ss+u(k,i)*u(k,j)
      end do
      utu(i,j)=ss
      utu(j,i)=ss
      end do
      end do

c      write(3,*)' utu rotated matrix'
c      do i=1,nfac2
c      write(3,1650)i,(utu(i,j),j=1,nfac2)
c      end do
cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccc

cccc  c  ccccccccccccccccccccccccccccccccccccccccccccccccccc
      do llk=1,nfac2
      recommv='comeof-recomputed-matrix-varimax.###'
      write(recommv(34:36),6789)llk
      write(*,'(1A38)')recommv
      xrecommv(llk)=recommv
      open(65,file=recommv,access='sequential',status='unknown')

```



```

        do i=1,nt
        do j=1,nx
        ds(i,j)=0.
        do k=llkk,llkk
        ds(i,j)=ds(i,j)+(u(i,k)*v(j,k)+u(i+nt,k)*v(j+nx,k))*
#      sqrt(sum)
        end do
        end do
        end do

607      write(65,607)recommv
        format(1a36///)
        do i=1,nt
        write(65,102)(ds(i,j),j=1,nx)
        end do
        close(65)
        end do

recommv='comeof-recomputed-matrix-varimax.all'
write(*,'(1A38)')recommv
xrecommv(nfac2+1)=recommv
open(65,file=recommv,access='sequential',status='unknown')
do i=1,nt
do j=1,nx
ds(i,j)=0.
do k=1,nfac2
ds(i,j)=ds(i,j)+(u(i,k)*v(j,k)+u(i+nt,k)*v(j+nx,k))*
#      sqrt(sum)
end do
end do
end do

write(65,607)recommv
do i=1,nt
write(65,102)(ds(i,j),j=1,nx)
end do
close(65)

c#### # #####

```

```

DO IVAL=1,NFAC2
DO J=1,NX
UCOM=V(J,IVAL)
VCOM=V(J+NX,IVAL)
pxqrxr=CONS*ATAN2(VCOM,UCOM)
PHASEV(J,IVAL)=XRANGE(pxqrxr)
AMPV(J,IVAL)=SQRT(UCOM**2+VCOM**2)*sqrt(e(ival)*sum/nt)
END DO
END DO
3849 format(3x,'amplitude-phase-rotated-v')
3949 format(3x,'amplitude-phase-rotated-u')

open(39,file='amp-phase-rotated-v',status='unknown',
# access='sequential')
c write(3,3849)
write(39,3849)
do j=1,nx
c write(3,1850)j,(ampv(j,ivec),phasev(j,ivec),ivec=1,nfac2),
c # lat(j),long(j)
write(39,1850)j,(ampv(j,ivec),phasev(j,ivec),ivec=1,nfac2),
# lat(j),long(j)
end do

DO IVAL=1,NFAC2
DO J=1,NT
UCOM=U(J,IVAL)
VCOM=U(J+NT,IVAL)
pxqrxr=CONS*ATAN2(VCOM,UCOM)
PHASEU(J,IVAL)=XRANGE(pxqrxr)
AMPU(J,IVAL)=SQRT(UCOM**2+VCOM**2)
END DO
END DO

open(49,file='amp-phase-rotated-u',status='unknown',
# access='sequential')
c write(3,3949)
write(49,3949)

```

```

      do j=1,nt
c      write(3,1850)j,(ampu(j,ivec),phaseu(j,ivec),ivec=1,nfac2)
      write(49,1850)j,(ampu(j,ivec),phaseu(j,ivec),ivec=1,nfac2)
      end do
c#### # #####
      write(*,4128)outfil
      write(3,4128)outfil
      write(51,4128)outfil
4128   format(///7x,'OUTPUT FILE DIARY'/
#      7x,'===== '//
#      2X,1A40/
#      2X,'amp-phase-v'/
#      2x,'amp-phase-u'/
#      2X,'amp-phase-rotated-v'/
#      2x,'amp-phase-rotated-u'/)
      do i=1,nfac2+1
      write(*,(2x,1a28)')xrecomm(i)
      write(3,(2x,1a28)')xrecomm(i)
      write(51,(2x,1a28)')xrecomm(i)
      end do
      do i=1,nfac2+1
      write(*,(2x,1a36)')xrecommv(i)
      write(3,(2x,1a36)')xrecommv(i)
      write(51,(2x,1a36)')xrecommv(i)
      end do

      STOP
      END
      subroutine hilbert(u,v,n,m)
      return
      end

C      FUNCTION RANGE(XX)
C      YY=ABS(XX)
C      II=YY/360
C      IF(XX.GT.0.1E-05) THEN
C      RANGE=XX-360.*II
C      ELSE
C      RANGE=XX+360.*(II+1)
C      END IF

```

```
C      RETURN
C      END
```

```
      FUNCTION XRANGE(XX)
      YY=ABS(XX)
      II=YY/360
      IF(XX.GT.0.1E-105) THEN
      XRANGE=XX-360.*II
      ELSE
      XRANGE=XX+360.*(II+1)
      END IF
      RETURN
      END
```

```
      SUBROUTINE EIGAN(A,V,E,N,NVAL,NVEC,SETA)
```

```
C
C      THIS ROUTINE IS FROM Ortega [1967]
C
```

```
      PARAMETER (MAXR=1024,MAXC=10000,MAXF=10,MAXRC=2*MAXR)
      parameter (MAXRCP2=MAXRC+2)
      DIMENSION V(MAXRC,MAXF),A(MAXRC,MAXRC)
```

```
      DIMENSION B(MAXRC),C(MAXRC),P(MAXRC),Q(MAXRC),R(MAXRC),W(MAXRC),
1  Y(MAXRCP2),IN(MAXRC),E(MAXF)
```

```
      EPA=1.E-06
      EPB=1.E-20
      EPC=1.E-30
      EPD=1.E-10
      BIG=1.E+20
      NTIME=2
      NITR=20
      SET=0.
      DO 17 J=1,N
```

```
17      SET=SET+A(J,J)
        SETA=SET
        WRITE(3,203)SET
        NMA=N-1
        NMB=N-2
        DO 18 I=1,NMB
          IPA=I+1
          SS=0.
          DO 10 J=IPA,N
10      SS=SS+A(J,I)**2
          S=SQRT(SS)
          IF(A(IPA,I).GT.0.)S=-S
          C(I)=A(I,I)
          B(I)=S
C
C      IF S IS ZERO THEN ALPHA MUST BE ZERO
C
        ALPHA=0.
        TOL=1.E-20
        IF(SS.LT.TOL) GO TO 18
        ALPHA=1./(SS-A(IPA,I)*S)
        T=A(IPA,I)-S
        A(IPA,I)=T
        W(I+1)=T
        IPB=I+2
        DO 11 J=IPB,N
11      W(J)=A(J,I)
        SET=0.
        DO 13 J=IPA,N
          T=0
          DO 12 K=IPA,N
12      T=T+A(J,K)*W(K)
          P(J)=T*ALPHA
13      SET=SET+W(J)*P(J)
          SET=SET*ALPHA*0.5
          DO 14 K=IPA,N
14      Q(K)=P(K)-SET*W(K)
          DO 15 J=IPA,N
            DO 15 K=J,N
              A(J,K)=A(J,K)-Q(J)*W(K)-Q(K)*W(J)
15      A(K,J)=A(J,K)
```

```
18      A(I,I)=ALPHA
        C(N-1)=A(N-1,N-1)
        C(N)=A(N,N)
        B(N-1)=A(N-1,N)
        B(N)=0.

C
C      COMPUTE THE EIGENVALUES
C
        TOP=ABS(C(1))+ABS(B(1))
        DO 20 I=2,N
        T=ABS(C(I))+ABS(B(I))+ABS(B(I-1))
        IF(T.GT.TOP) TOP=T
20      CONTINUE
        DO 21 I=1,N
21      W(I)=B(I)**2
        DO 22 I=1,NVAL
22      E(I)=-TOP
        SUM=0.
        DO 38 K=1,NVAL
        BOT=E(K)
        DO 40 ITER=1,NITR
24      PL=0.5*(TOP+BOT)
        ERR=ABS(TOP-BOT)
        IF(ERR.LT.EPA)GO TO 41
        NAG=0
        I=1
26      S=C(I)-PL
28      IF(S.GE.0.)NAG=NAG+1
        IF(S.EQ.0.) GO TO 30
        I=I+1
        IF(I.GT.N) GO TO 32
        S=C(I)-PL-W(I-1)/S
        GO TO 28
30      I=I+2
        IF(I.LE.N)GO TO 26
32      IF(NAG.GE.K) GO TO 34
        TOP=PL
        GO TO 40
34      BOT=PL
40      CONTINUE
41      E(K)=PL
```

```
SUM=SUM+E(K)
SS=SUM/SETA
WRITE(3,202)K,E(K),ERR,ITER,SUM,SS
38 CONTINUE
WRITE(3,204)NVAL,SUM
IF(NVEC.NE.0) GO TO 42
GO TO 80
42 DO 70 I=1,NVEC
DO 44 J=1,N
P(J)=0.
Q(J)=B(J)
R(J)=C(J)-E(I)
44 Y(J)=1.
Y(N+1)=0.
Y(N+2)=0.
C
C BEGIN THE GAUSSIAN ELIMINATION
C
IF(R(1).EQ.0.)R(1)=EPC
DO 48 J=1,NMA
IF(ABS(R(J)).LT.ABS(B(J)))GO TO 45
FAC=B(J)/R(J)
IN(J)=0
GO TO 46
45 FAC=R(J)/B(J)
IN(J)=1
R(J)=B(J)
T=R(J+1)
R(J+1)=Q(J)
Q(J)=T
P(J)=Q(J+1)
Q(J+1)=0.
46 W(J)=FAC
Q(J+1)=Q(J+1)-FAC*P(J)
R(J+1)=R(J+1)-FAC*Q(J)
IF(R(J+1).EQ.0.)R(J+1)=EPC
48 CONTINUE
C
C BEGIN THE BAC- SUBSTITUTION
C
DO 60 ITIME=1,NTIME
```

```
DO 53 JJ=1,N
K=N+1-JJ
T=Y(K)
51 Y(K)=(T-Y(K+1)*Q(K)-Y(K+2)*P(K))/R(K)
IF(ABS(Y(K)).LT.BIG)GO TO 53
DO 52 J=1,N
52 Y(J)=Y(J)*EPD
T=T*EPD
GO TO 51
53 CONTINUE
C
C THIS COMPLETES BAC- SUBSTITUTION
C DO THE REDUCTION IN NEW R.H.S.
IF(ITIME.EQ.NTIME)GO TO 60
DO 57 J=1,NMA
IF(IN(J).EQ.1) GO TO 56
Y(J+1)=Y(J+1)-W(J)*Y(J)
GO TO 57
56 T=Y(J)
Y(J)=Y(J+1)
Y(J+1)=T-W(J)*Y(J+1)
57 CONTINUE
60 CONTINUE
DO 64 J=1,NMB
T=0.
K=N-J-1
M=K+1
DO 62 KK=M,N
62 T=T+A(KK,K)*Y(KK)
T=A(K,K)*T
DO 64 KK=M,N
64 Y(KK)=Y(KK)-T*A(KK,K)
SS=0.
DO 67 J=1,N
67 SS=SS+Y(J)**2
T=1./SQRT(SS)
DO 66 J=1,N
66 V(J,I)=Y(J)*T
70 CONTINUE
DO 71 I=1,NVEC
DO 71 J=I,NVEC
```



```

      A(I,J)=0.
      DO 71 M=1,N
      A(I,J)=A(I,J)+V(M,I)*V(M,J)
71    A(J,I)=A(I,J)
      c    WRITE(3,206)
      c    DO 73 J=1,NVEC
      c 7    3    WRITE(3,212)J,(V(I,J),I=1,N)
      c    WRITE(3,207)
      c    DO 74 J=1,NVEC
      c 7    4    WRITE(3,200)J,(A(I,J),I=1,NVEC)
80    CONTINUE
200    FORMAT(I5,(10E13.4)/)
202    FORMAT(2X,'EIGEN VALUE NO. ',I4,E13.4,2X,'WITH ERROR LIMITS',E13
1     .4,2X,'AFTER',I4,2X,'ITERATIONS',2E13.4/)
203    FORMAT(//5X,'SUM OF THE DIAGONAL ELEMENTS OF MATRIX A= ',E15.4//
1     T21,'I',T30,'E(I)',T94,'SUM E(I)',T106,'CLOSENESS RATIO'/)
204    FORMAT(5X,'SUM OF ',I3,2X,'EIGEN VALUE= ',E15.4/)
206    FORMAT(//,5X,'EIGEN VECTORS ')
207    FORMAT(//,5X,'MATRIX- V(TRANSPOSE) X V ')
212    FORMAT(I5,11f8.4/(5x,11f8.4))
      RETURN
      END

      SUBROUTINE VARMAXC(F,COM,NF,NN,T)
C
C     THIS ROUTINE FOR VARIMAX ROTATION OF COMPLEX VALUED
C     PRINCIPAL COMPONENTS IS BASED ON THE VARIMAX ROUTINE OF Klován and Imbrie [1971]
C     FOR ROTATION OF REAL VALUED PRINCIPAL COMPONENTS.
C     THE MODIFICATIONS FOLLOW Horel [1984] AND Kaiser [1958]
C
      PARAMETER (MAXR=1024,MAXC=10000,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
      c    DIMENSION F(MAXWR,MAXF)
      DIMENSION COM(MAXC),T(MAXF,MAXF)
      complex F(MAXC,MAXF),RC
      write(*,*) 'nf, nn = ',nf,nn
      EPS=0.06993
      DO 11 I=1,NF
      DO 10 J=1,NF
10     T(I,J)=0.
11     T(I,I)=1.
      DO 13 I=1,NN

```

```

      COM(I)=0.
      DO 12 J=1,NF
c 1      2  COM(I)=COM(I)+F(I,J)**2
12      COM(I)=COM(I)+F(I,J)*conjg(f(i,j))
      COM(I)=SQRT(COM(I))
      DO 13 J=1,NF
13      F(I,J)=F(I,J)/COM(I)
      NFA=NF-1
14      NROT=0
      DO 40 M=1,NFA
      K=M+1
      DO 40 L=K,NF
      A=0.
      B=0.
      C=0.
      D=0.
      DO 15 I=1,NN
c      U=F(I,M)**2-F(I,L)**2
c      V=F(I,M)*F(I,L)*2
      u=f(i,m)*conjg(f(i,m))-f(i,l)*conjg(f(i,l))
      v=2.*(real(f(i,m))*real(f(i,l))+aimag(f(i,m))*aimag(f(i,l)))
      A=A+U
      B=B+V
      C=C+U**2-V**2
15      D=D+U*V*2
      QNUM=D-2.*A*B/NN
      QDEN=C-(A**2-B**2)/NN
      IF(ABS(QNUM)+ABS(QDEN).EQ.0.)GO TO 30
      IF(ABS(QNUM)-ABS(QDEN))17,24,21
17      R=ABS(QNUM/QDEN)
      IF(R.LT.EPS)GO TO 19
      CS4TH=COS(ATAN(R))
      SN4TH=SIN(ATAN(R))
      GO TO 25
19      IF(QDEN.GE.0.)GO TO 30
      SNPHI=0.70710678
      CSPHI=SNPHI
      GO TO 31
21      R=ABS(QDEN/QNUM)
      IF(R.LT.EPS)GO TO 23
      SN4TH=1./SQRT(1.+R**2)

```

```
      CS4TH=SN4TH*R
      GO TO 25
23     CS4TH=0.
      SN4TH=1.
      GO TO 25
24     CS4TH=0.70710678
      SN4TH=CS4TH
25     R=SQRT((1.+CS4TH)*.5)
      CSTH=SQRT((1.+R)*0.5)
      SNTH=SN4TH/(4.*CSTH*R)
      IF(QDEN.GE.0.)GO TO 27
      CSPHI=.70710678*(CSTH+SNTH)
      SNPHI=.70710678*(CSTH-SNTH)
      GO TO 28
27     CSPHI=CSTH
      SNPHI=SNTH
28     IF(QNUM.GE.0.)GO TO 31
      SNPHI=-SNPHI
      GO TO 31
30     NROT=NROT+1
      GO TO 40
31     DO 32 J=1,NN
      RC=F(J,M)*CSPHI+F(J,L)*SNPHI
      F(J,L)=F(J,L)*CSPHI-F(J,M)*SNPHI
      F(J,M)=RC
      IF(J.GT.NF)GO TO 32
      TP=T(J,M)
      T(J,M)=TP*CSPHI+T(J,L)*SNPHI
      T(J,L)=-TP*SNPHI+T(J,L)*CSPHI
32     CONTINUE
40     CONTINUE
      IF(NROT.LT.NF*NFA/2)GO TO 14
      DO 42 I=1,NN
      DO 41 J=1,NF
41     F(I,J)=F(I,J)*COM(I)
42     COM(I)=COM(I)**2
c     write(*,*)(com(i),i=1,nn)
      RETURN
      END
      SUBROUTINE SPECR(OUTFIL,INFIL,AFMT,NX,NT,NFAC,LTLNFILE,BFMT)
      CHARACTER*60 INFIL,OUTFIL,AFMT,BFMT,LTLNFILE
```

```

CHARACTER*1 AMY
READ(15,'(50X,1A1)')AMY
READ(15,'(50X,1A60)')OUTFIL
write(*,'(50X,1A60)')OUTFIL
READ(15,'(50X,1A60)')INFIL
READ(15,'(50X,1A60)')AFMT
READ(15,'(50X,I5)')NX
READ(15,'(50X,I5)')NT
READ(15,'(50X,I5)')NFAC
READ(15,'(50X,1A60)')LTLNFILE
READ(15,'(50X,1A60)')BFMT
RETURN
END

```

```

SUBROUTINE DATAM(dr,di,NROW,NCOL,AFMT,INFIL,LABEL)
PARAMETER (MAXR=1024,MAXC=10000,MAXF=10,MAXWC=2*MAXC,MAXWR=2*MAXR)
DIMENSION DR(MAXR,MAXC),DI(MAXR,MAXC)
DIMENSION LABEL(80)
CHARACTER*60 AFMT
CHARACTER*60 INFIL

```

```

OPEN(25,FILE=INFIL,ACCESS='SEQUENTIAL',BLANK='ZERO')
READ(25,10)(LABEL(I),I=1,80)
WRITE(*,10)(LABEL(I),I=1,80)
10  FORMAT(20A4)
WRITE(*,1234)
1234 format(5x,'IS THIS THE 4 LINE LABEL OF YOUR INPUT DATA? (Y/N) ', $)
call entry(iww,5,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP 'CHECK YOUR INPUT DATA!'
WRITE(*,1235)
1235  FORMAT(5X,'IS THE INPUT MATRIX READ ROWWISE OR COLUMNWISE? (R/C) '
#  , $)
CALL ENTRY(IWW,2,'R','C','3','4','5')
c  IF(IWW.EQ.2) STOP 'CHECK YOUR INPUT DATA!'

IF(IWW.EQ.1)then
DO I=1,NROW
c  READ(25,AFMT)(AAA(I,J),J=1,NCOL)
READ(25,AFMT)(dr(I,J),J=1,NCOL),(di(i,j),j=1,ncol)
IF(I.EQ.1)THEN

```

```

WRITE(*,203)I,(dr(I,J),J=1,NCOL),(di(i,j),j=1,ncol)
203  FORMAT(2X,I3,2X,5E12.4/(7X,5E12.4))
WRITE(*,1236)
1236  FORMAT(5X,'IS THIS THE FIRST ROW OF YOUR INPUT MATRIX? (Y/N) ', $)
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP' CHECK YOUR INPUT DATA!'
END IF
c    WRITE(3,203)I,(AAA(I,J),J=1,NCOL)
write(3,203)I,(dr(I,J),J=1,NCOL),(di(i,j),j=1,ncol)
END DO
RETURN

else if(iww.eq.2)then
DO J=1,NCOL
c    READ(25,AFMT)(AAA(I,J),I=1,NROW)
READ(25,AFMT)(dr(I,J),i=1,nrow),(di(i,j),i=1,nrow)
IF(J.EQ.1)THEN
WRITE(*,203)j,(dr(I,J),i=1,nrow),(di(i,j),i=1,nrow)
WRITE(*,1237)
1237  FORMAT(5X,'IS THIS THE FIRST COLUMN OF YOUR INPUT MATRIX? (Y/N) ',
#    $)
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP' CHECK YOUR INPUT DATA!'
END IF

c    WRITE(3,203)J,(AAA(I,J),I=1,NROW)
write(3,203)j,(dr(I,J),i=1,nrow),(di(i,j),i=1,nrow)
END DO
RETURN
end if

END

SUBROUTINE ENTRY(IWT,NP,X1,X2,X3,X4,X5)
CHARACTER*1 ICHR(5),X0,X1,X2,X3,X4,X5
IWT=0
ICHR(1)=X1
ICHR(2)=X2
ICHR(3)=X3
ICHR(4)=X4
ICHR(5)=X5

```

```
READ(*,'(A1)')X0
DO I=1,NP
IF(X0.EQ.ICHR(I))IWT=I
END DO
IF(IWT.EQ.0)STOP ' MISKEY !'
RETURN
END

SUBROUTINE CILT_FIL(INFIL,CPCFIL)
CHARACTER*60 INFIL,CPCFIL
CHARACTER*1 DOT,BLANK
DATA DOT './,BLANK/' /
DO J=1,60
CPCFIL(J:J)=' '
END DO
DO I=1,60
IF(INFIL(I:I).NE.DOT.AND.INFIL(I:I).NE.BLANK)THEN
CPCFIL(I:I)=INFIL(I:I)
ELSE
CPCFIL(I:I)=DOT
CPCFIL(I+1:I+3)=' cat'
RETURN
END IF
END DO
RETURN
END
```

### A.3.2 Specification File: ceof-cpc.spc

CTAB 51

Output file (A60): seec  
Input file (A60): ceof-ex3.hat  
Format for input file (A60): (22e16.7)  
Number of stations (I5) : 00011  
Number of time steps (I5) : 00240  
Number of time factors (I5) : 00004  
Lat Long file (A60): sla-78s-lat-long.dat  
Format for input file (A60): (1a75)

: ↑ input is from column number 51

Latitude Longitude file is not used for computations

but just to help later in plotting the spatial CEOFs, viz.,

the output files (spatial): amp-phase-v, amp-phase-rotated-v

Output files (temporal) are : amp-phase-u, amp-phase-rotated-u

Output files for recomputed matrix at each mode are also given

## A.4 Cross Spectrum Analysis

### A.4.1 Computer Program: cross-spec.f

```

PROGRAM ALPHA
parameter (im1=8192)
PARAMETER(IM11=IM1+1,IM5=IM1/2+1)
COMPLEX Z,XX,YY,ZZ,ZC,RIM,XY(IM5),ZE
COMPLEX UPLUS,UMINUS,UPLMIN(IM5)
COMPLEX SXSTY(IM5),SUPM(IM5)
CHARACTER*11 WIN(5),TREND(2),UNITS(2)*30
REAL NSE,MEANX,MEANY
DIMENSION Z(IM11),RSIG(IM5),THEOR(IM5)
DIMENSION XN(IM1),YN(IM1),GX(IM5),GY(IM5),F(IM5),T(IM5),GXY(IM5)
DIMENSION BXR(IM1),BXI(IM1)
DIMENSION RET(IM5),PHASE(IM5),COH(IM5),RES(IM5),STAB(IM5)
DIMENSION SGX(IM5),SGY(IM5)
DIMENSION B(IM1)
DIMENSION SPLUS(IM5),SMINUS(IM5),STOTAL(IM5),CRS(IM5)
DIMENSION SSPLUS(IM5),SSMINUS(IM5)
DIMENSION IBIG(19)
CHARACTER*25 OUTFIL,INFILX,INFILY,AXBIG,AYBIG,PLOTFIL
DATA WIN /'RECTANGULAR','HANNING','HAMMING','BLACKMAN','COSINE'/
DATA TREND/'REMOVED','NOT REMOVED'/
DATA UNITS/'HARRIS (1974). PROC. OWM,VOL 2,PG 107','BENDAT & PIERS
O OL (1971).'/

LIST=0
sgn=-1.
WRITE(*,7000)IM1
CALL PAR(OUTFIL,IBIG,NUN,NP,NIJ,DELTT,FACTORX,FACTORY,
# ITREND,IWT,IPRINT,IROT,INFILX,INFILY,AXBIG,NUMBX,AYBIG,NUMBY,
# NLABX,NLABY)
OPEN(3,FILE=OUTFIL,access='sequential',STATUS='unknown')
CALL BIGLET (IBIG,10,3)
c CALL TIME(IBIG)
CALL TIMEZ(IBIG)
WRITE(3,321)OUTFIL,IBIG,UNITS(NUN),NP,NIJ,DELTT
WRITE(3,322)FACTORX,FACTORY,TREND(ITREND),WIN(IWT),IPRINT,IROT
WRITE(3,323)INFILX,INFILY,AXBIG,NUMBX,AYBIG,NUMBY
321 FORMAT(5X,'Output file name :',a25/
# 5X,'Your name :',19a1/

```



```

# 5X,'Density units :',A30/
# 5X,'Number of points in ensemble, NP :',I5/
# 5X,'Number of ensembles :',I2/
# 5X,'Sampling Interval :',F8.3)
322   FORMAT(5X,'Scale factor for X series :',F8.2/
# 5X,'Scale factor for Y series :',F8.2/
# 5X,'Trend :',a11/
# 5X,'Window :',A11/
# 5X,'Print ensemble results (Yes=1, No=0) :',I5/
# 5X,'Compute rotary spectra (Yes=1, No=0) :',I5)
323   FORMAT(5X,'Input file X series :',A25/
# 5X,'Input file Y series :',A25/
# 5X,'Format X series :',A25/
# 5X,'Values per line X series :',I2/
# 5X,'Format Y series :',A25/
# 5X,'Values per line Y series :',I2////)
SMALL=1.E-45
C     SMALL=1.E-30
CALL WINDOW(NP,IWT,SCALE,W,B)
NBY2=NP/2
IF(IWT.EQ.1)GOTO 30
DO I=1,NBY2
I1=NBY2+I
I2=NBY2-I+1
B(I1)=B(I2)
END DO
30   CONTINUE
50   CONTINUE
PAUSE ' Press ENTER key to continue'
RIM=CMPLX(0.,1.)
ZE=CMPLX(0.,0.)
PI=3.14159
NBY21=NBY2+1
THMUL=180./PI
DELNP=NP*DELTT
IF(NUN.EQ.1)AMUL=1./SCALE
IF(NUN.EQ.2)AMUL=.5*DELNP/SCALE
FR=W/DELNP
DF=2.
NSE=1.
F(1)=0.

```

```
T(1)=99999.99
DO 70 I=2,NBY21
F(I)=(I-1.)/DELNP
70 T(I)=1./F(I)
DELTA F=F(2)
DO 430 IJK=1,NIJ
REWIND(39)
READ(39,'(A25,5X,A25)')AXBIG,AYBIG
CC WRITE(3,5001)AXBIG,AYBIG
CC WRITE(*,5001)AXBIG,AYBIG
5001 FORMAT(5X,'AXBIG=',A25,5X,'AYBIG=',A25)
CALL DATA22(XN,YN,NP,IJK,ITREND,AXBIG,AYBIG,NUMBX,NUMBY,
# INFILX,INFILY,NLABX,NLABY)
90 CONTINUE
DO 110 I=1,NP
110 Z(I)=CMPLX(XN(I)*FACTORX,YN(I)*FACTORY)
C WRITE(*,10)(Z(I),I=1,NP)
ZZ=ZE
DO 130 I=1,NP
130 ZZ=ZZ+Z(I)
ZZ=ZZ/NP
MEANX=REAL(ZZ)
MEANY=AIMAG(ZZ)
SMEANX=SMEANX+MEANX
SMEANY=SMEANY+MEANY
VARX=0.
VARY=0.
DO 150 I=1,NP
Z(I)=Z(I)-ZZ
VARX=VARX+REAL(Z(I))**2
150 VARY=VARY+AIMAG(Z(I))**2
VARX=VARX/NP
VARY=VARY/NP
WRITE(3,2000)MEANX,MEANY,VARX,VARY
SVARX=SVARX+VARX
SVARY=SVARY+VARY
IF(IWT.EQ.1)GOTO 190
DO 170 I=1,NP
170 Z(I)=B(I)*Z(I)
190 CONTINUE
```

```
DO 210 I=1,NP
  BXR(I)=REAL(Z(I))
  BXI(I)=AIMAG(Z(I))
210

call newfft(bxr,bxi,sgn,list,np)

DO I=1,NP
  Z(I)=CMPLX(BXR(I),BXI(I))
  Z(I)=Z(I)/NBY2*SQRT(AMUL)
END DO
C  WRITE(*,10)(Z(I),I=1,NBY21)
C  WRITE(3,10)(Z(I),I=1,NBY21)
  Z(NP+1)=Z(1)
  DO 390 I=1,NBY21
    ZZ=Z(I)
    NPI2=NP-I+2
    ZC=CONJG(Z(NPI2))
    XX=(ZZ+ZC)/2.
    YY=(ZZ-ZC)/2./RIM
    XABS=CABS(XX)
    YABS=CABS(YY)
    PXI=0.
    PYI=0.
    IF(XABS.GT.SMALL)PXI=ATAN2(REAL(XX),-AIMAG(XX))
    IF(YABS.GT.SMALL)PYI=ATAN2(REAL(YY),-AIMAG(YY))
    A1=XABS*SIN(PXI)
    B1=XABS*COS(PXI)
    A2=YABS*SIN(PYI)
    B2=YABS*COS(PYI)
    A1P2=A1+B2
    A1M2=A1-B2
    A2P1=A2+B1
    A2M1=A2-B1
    UPLUS=0.5*CMPLX(A1P2,A2M1)
    UMINUS=0.5*CMPLX(A1M2,A2P1)
    SPLUS(I)=UPLUS*CONJG(UPLUS)
    SMINUS(I)=UMINUS*CONJG(UMINUS)
    STOTAL(I)=SPLUS(I)+SMINUS(I)
    CRS(I)=0.
    IF (STOTAL(I).GT.SMALL)CRS(I)=(SMINUS(I)-SPLUS(I))/STOTAL(I)
```

```
UPLMIN(I)=UPLUS*UMINUS
RSIG(I)=CABS(UPLMIN(I))
STAB(I)=1.
THEOR(I)=0.

IF(RSIG(I).GT.SMALL) THEN
THEOR(I)=THMUL*ATAN2(AIMAG(UPLMIN(I)),REAL(UPLMIN(I)))
END IF
XY(I)=CONJG(XX)*YY
COSI=REAL(XY(I))
QOSI=AIMAG(XY(I))
GXY(I)=CABS(XY(I))

IF(CABS(XY(I)).GT.SMALL)THEN
PHASE(I)=THMUL*ATAN2(QOSI,COSI)
ELSE
PHASE(I)=0.
END IF
GX(I)=REAL(XX)**2+AIMAG(XX)**2
GY(I)=REAL(YY)**2+AIMAG(YY)**2
COH(I)=1.
IF(GX(I).GT.SMALL)GO TO 370
RES(I)=0.
RET(I)=0.
GOTO 390
370 RET(I)=SQRT(GY(I)/GX(I))
RES(I)=GXY(I)/GX(I)
390 CONTINUE
IF((IPRINT.EQ.1).OR.(NIJ.EQ.1)) THEN
WRITE(3,4000)IJK
CALL ENERGY(GX,GY,SX,SY,NBY21,FR,DF,NSE,NUN,DELTA F)
CALL OUT(T,GXY,GX,GY,RES,RET,COH,PHASE,F,NBY21)
IF(IROT.EQ.1)CALL OUT1(T,SPLUS,SMINUS,STOTAL,CRS,RSIG,
# STAB,THEOR,F,NBY21)
END IF
DO 410 I=1,NBY21
SXSTY(I)=XY(I)+SXSTY(I)
SUPM(I)=UPLMIN(I)+SUPM(I)
SGX(I)=SGX(I)+GX(I)
SGY(I)=SGY(I)+GY(I)
SSPLUS(I)=SSPLUS(I)+SPLUS(I)
```

```
SSMINUS(I)=SSMINUS(I)+SMINUS(I)
410 CONTINUE
430 CONTINUE
IF(NIJ.EQ.1)GOTO 450
DO 440 I=1,NBY21
SGX(I)=SGX(I)/NIJ
SGY(I)=SGY(I)/NIJ
SSPLUS(I)=SSPLUS(I)/NIJ
SSMINUS(I)=SSMINUS(I)/NIJ
SXSTY(I)=SXSTY(I)/NIJ
SUPM(I)=SUPM(I)/NIJ

SQ0SI=AIMAG(SXSTY(I))
SC0SI=REAL(SXSTY(I))
C0SI=REAL(SUPM(I))
Q0SI=AIMAG(SUPM(I))

STOTAL(I)=SSPLUS(I)+SSMINUS(I)
GXY(I)=SQRT(SC0SI**2+SQ0SI**2)
RSIG(I)=SQRT(C0SI**2+Q0SI**2)
PHASE(I)=0.
IF(GXY(I).GT.SMALL) THEN
PHASE(I)=THMUL*ATAN2(SQ0SI,SC0SI)
END IF
THEOR(I)=0.
IF(RSIG(I).GT.SMALL) THEN
THEOR(I)=THMUL*ATAN2(Q0SI,C0SI)
END IF
RES(I)=0.
RET(I)=0.
STAB(I)=0.
COH(I)=0.
CRS(I)=0.
ADENX=SQRT(SGX(I)*SGY(I))
IF(ADENX.GT.SMALL)COH(I)=GXY(I)/ADENX
IF(STOTAL(I).GT.SMALL)CRS(I)=(SSMINUS(I)-SSPLUS(I))/STOTAL(I)
IF(SGX(I).GT.SMALL)THEN
RES(I)=GXY(I)/SGX(I)
RET(I)=SQRT(SGY(I)/SGX(I))
C IF(SGY(I).GT.SMALL)COH(I)=GXY(I)/SGX(I)*GXY(I)/SGY(I)
ENDIF
```

```

ADENX=SQRT(SSPLUS(I)*SSMINUS(I))
IF(ADENX.GT.SMALL)STAB(I)=RSIG(I)/ADENX
440 CONTINUE

FR=W/(NP*DELTT)
DF=2.*NIJ
NSE=1./SQRT(DF/2)
MEANX=SMEANX/NIJ
MEANY=SMEANY/NIJ
VARX=SVARX/NIJ
VARY=SVARY/NIJ
WRITE(3,3000)
WRITE(3,2000)MEANX,MEANY,VARX,VARY
CALL ENERGY(SGX,SGY,SX,SY,NBY21,FR,DF,NSE,NUN,DELTA F)
CALL OUT(T,GXY,SGX,SGY,RES,RET,COH,PHASE,F,NBY21)
IF(IROT.EQ.1)CALL OUT1(T,SSPLUS,SSMINUS,STOTAL,CRS,RSIG,
# STAB,THEOR,F,NBY21)
WRITE(*,6587)
6587 format(' Enter name of plot file: ', $)
READ(*,'(A25)')PLOTFIL
OPEN(65,FILE=PLOTFIL)
WRITE(65,1002)
DO I=1,NBY21
WRITE(65,2002)I,T(I),SXSTY(I),SGX(I),SGY(I),RES(I),RET(I),
# COH(I),PHASE(I),SUPM(I),SSPLUS(I),SSMINUS(I),STOTAL(I),
# STAB(I),THEOR(I),F(I),I
END DO
1002 FORMAT(/T12,'No.',T20,'T',T39,'SXSTY',T59,'GX',T72,'GY',
# T85,'RES',T98,'RET',T111,'COH',T124,'PHASE',T143,'SUPM',
# T163,'SPLUS',T176,'SMINUS',T189,'STOTAL',T202,'STAB',
# T215,'THEOR',T228,'FREQ',T246,'No.')
```

```

2002 FORMAT(i13,17e13.5,i13)
450 CONTINUE

WRITE(*,6000)OUTFIL
STOP
2000 FORMAT(/5X,'MEAN X =',E11.3,10X,'MEAN Y =',E11.3/
5 5X,'VAR X =',E11.3,10X,'VAR Y =',E11.3//)
3000 FORMAT(1H1,7X,'ENSEMBLE AVERAGED SPECTRAL ESTIMATES')
4000 FORMAT(/7X,'SPECTRAL ESTIMATES OF ENSEMBLE ',I2)
6000 FORMAT(/5X,'Your output file is : ',A25)

```

```

7000      FORMAT(' This package is capable of handling series lengths upto',
F      1X,15//)
      END
      SUBROUTINE TREND(X,N)
      DIMENSION X(N)
      Z=0.
      ZN=0.
      DO 1 I=1,N
      Z=Z+X(I)
1      ZN=ZN+I*X(I)
      DN=N
      DN2=DN*DN
      B0=(4.*N+2.)*Z-6.*ZN
      B0=B0/(DN2-DN)
      B1=12.*ZN-(6.*N+6.)*Z
      B1=B1/(DN2*DN-DN)
      DO 2 I=1,N
2      X(I)=X(I)-B0-B1*I
      RETURN
      END
      SUBROUTINE ENTRY(IWT,NP,X1,X2,X3,X4,X5)
      CHARACTER*1 ICHR(5),X0,X1,X2,X3,X4,X5
      IWT=0
      ICHR(1)=X1
      ICHR(2)=X2
      ICHR(3)=X3
      ICHR(4)=X4
      ICHR(5)=X5

      READ(*,'(A1)')X0
      DO I=1,NP
      IF(X0.EQ.ICHR(I))IWT=I
      END DO
      IF(IWT.EQ.0)STOP ' MISKEY !'
      RETURN
      END

      SUBROUTINE TRENDM(X,N)
      DIMENSION X(N)
C      REMOVE LINEAR TREND
      TX=0.

```

```

      SX=0.
      DO 21 I=1,N
      TX=TX+(I-1)*X(I)
21    SX=SX+X(I)
      DN=N
      P=(TX-(N-1)*SX*.5)*12./(DN**3-DN)
      Q=SX/N-P*(N-1)*.5
      DO 22 I=1,N
22    X(I)=X(I)-P*(I-1)-Q
      RETURN
      END
      SUBROUTINE BIGLET (IBIG,NCHR,IOUT)
C — Subroutine obtained under NORAD programme
C —
C —
C — SUBROUTINE BIGLET WRITES OUT A STRING OF CHARACTERS WITH BIG
C — LETTERS.
C —
C — THE CHARACTERS SHOULD BE STORED IN AN INTEGER ARRAY, ONE
C — CHARACTER PER WORD, LEFT JUSTIFIED, BLANK FILLED (A1 FORMAT).
C — THE LINE MAY CONTAIN UP TO 19 LETTERS. ALL LETTERS ARE MADE UP
C — FROM A 5*7 POINT MATRIX AND WRITTEN USING DOLLAR SIGNS ($).
C — ONLY CHARACTERS FROM THE STANDARD FORTRAN CHARACTER SET MAY
C — BE USED. THIS SET COMPRISES UPPER CASE LETTERS (A-Z), DECIMAL
C — DIGITS (0-9), FORTRAN ARITHMETIC OPERATORS (+-*/=), LEFT AND
C — RIGHT PARENTHESES (()), COMMA (,), DECIMAL POINT (.), DOLLAR
C — SIGN ($)AND BLANK (.). ANY OTHER CHARACTER WILL BE CONVERTED
C — TO A BLANK (.). THIS IS AN INTENTIONAL RESTRICTION TO PROMOTE
C — THE USE OF STANDARD FORTRAN.
C —
C — PARAMETERS (* MEANS THE PARAMETER MAY CHANGE VALUE)
C —
C — IBIG - THE INTEGER ARRAY CONTAINING LINE TO BE WRITTEN
C — NCHR - THE NUMBER OF CHARACTERS IN THE LINE
C — IOUT - LOGICAL UNIT NUMBER FOR OUTPUT
C —
C — THIS ROUTINE IS NOT COMPLETELY MACHINE INDEPENDENT IN THAT IT
C — ASSUMES THAT AN INTEGER CAN HOLD AT LEAST FIVE CHARACTERS. IF
C — THIS IS NOT THE CASE THE ROUTINE MUST BE REVISED.
C —
C —

```



C—

```

CHARACTER*5 IPPT,IBLN
DIMENSION IBIG(NCHR), JBIG(19), IALW(47), IPAT(7,47), IPPT(32)
DIMENSION IBLN(19)

```

C—

C— ARRAY OF ALLOWABLE CHARACTERS

C—

```

DATA IALW / 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J',
- 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V',
- 'W', 'X', 'Y', 'Z', '0', '1', '2', '3', '4', '5', '6', '7',
- '8', '9', '+', '-', '*', '/', '=', '(', ')', ',', '.', '$',
- ' ' /

```

C—

C— ARRAY GIVING DIFFERENT POINT PATTERNS

C—

```

DATA IPPT / ' ', '$', '$', '$$', '$ ', '$$',
- '$$', '$$$', '$ ', '$$', '$$', '$$$',
- '$$', '$$$', '$$$', '$$$$', '$ ', '$$',
- '$$', '$$$', '$$', '$$$', '$$$', '$$$$',
- '$$', '$$$', '$$$', '$$$$', '$$$', '$$$$',
- '$$$$', '$$$$$' /

```

C—

C— ARRAY OF CHARACTER POINT PATTERN

C—

```

DATA IPAT / 14, 17, 17, 17, 31, 17, 17,
- 30, 17, 17, 30, 17, 17, 30,
- 15, 16, 16, 16, 16, 16, 15,
- 30, 17, 17, 17, 17, 17, 30,
- 31, 16, 16, 30, 16, 16, 31,
- 31, 16, 16, 30, 16, 16, 16,
- 15, 16, 16, 16, 23, 17, 14,
- 17, 17, 17, 31, 17, 17, 17,
- 14, 04, 04, 04, 04, 04, 14,
- 01, 01, 01, 01, 17, 17, 14,
- 17, 17, 18, 28, 18, 17, 17,
- 16, 16, 16, 16, 16, 16, 31,
- 17, 27, 21, 17, 17, 17, 17,
- 17, 17, 25, 21, 19, 17, 17,
- 14, 17, 17, 17, 17, 17, 14,
- 30, 17, 17, 30, 16, 16, 16,
- 14, 17, 17, 17, 21, 18, 13,

```

- 30, 17, 17, 30, 20, 18, 17,  
 - 15, 16, 16, 14, 01, 01, 30,  
 - 31, 04, 04, 04, 04, 04, 04,  
 - 17, 17, 17, 17, 17, 17, 31,  
 - 17, 17, 17, 17, 17, 10, 04,  
 - 17, 17, 17, 17, 21, 27, 17,  
 - 17, 17, 10, 04, 10, 17, 17,  
 - 17, 17, 10, 04, 04, 04, 04,  
 - 31, 01, 02, 04, 08, 16, 31,  
 - 14, 17, 19, 21, 25, 17, 14,  
 - 04, 12, 04, 04, 04, 04, 14,  
 - 14, 17, 01, 02, 04, 08, 31,  
 - 14, 17, 01, 02, 01, 17, 14,  
 - 18, 18, 18, 31, 02, 02, 02,  
 - 31, 16, 30, 01, 01, 17, 14,  
 - 14, 16, 16, 30, 17, 17, 14,  
 - 31, 01, 01, 02, 04, 08, 16,  
 - 14, 17, 17, 14, 17, 17, 14,  
 - 14, 17, 17, 15, 01, 01, 14,  
 - 00, 04, 04, 31, 04, 04, 00,  
 - 00, 00, 00, 31, 00, 00, 00,  
 - 00, 21, 14, 27, 14, 21, 00,  
 - 00, 00, 01, 02, 04, 08, 16,  
 - 00, 00, 31, 00, 31, 00, 00,  
 - 04, 08, 16, 16, 16, 08, 04,  
 - 04, 02, 01, 01, 01, 02, 04,  
 - 00, 00, 00, 00, 00, 06, 02,  
 - 00, 00, 00, 00, 00, 06, 06,  
 - 04, 31, 20, 31, 05, 31, 04,  
 - 00, 00, 00, 00, 00, 00, 00/

C —

C — GUARD AGAINST MORE THAN 19 CHARACTERS

C —

MCHR = MIN0(NCHR,19)

C —

C — CONVERT CHARACTERS TO TABLE INDEX AND SUBSTITUTE

C — BLANKS FOR ILLEGAL CHARACTERS

C —

DO 120 L120 = 1, MCHR

DO 110 L110 = 1, 46

110 IF (IBIG(L120).EQ. IALW(L110))GOTO 120

```

      L110 = 47
120    JBIG(L120)= L110
C —
C —    WRITE TWO EXTRA BLANK LINES
C —
      WRITE (IOUT,900)
900    FORMAT (//)
C —
C —    SEVEN LINES OF POINTS FOR EACH CHARACTER LINE
C —
      DO 140 L140 = 1, 7
C —
C —    GET CORRECT POINT PATTERNS FOR THIS ROW
C —
      DO 130 L130 = 1, MCHR
      ILET = JBIG(L130)
      MASK = IPAT(L140,ILET)
130    IBLN(L130)= IPPT(MASK+1)
C —
C —    WRITE OUT ONE ROW OF POINT PATTERNS
C —
      WRITE (IOUT,910)(IBLN(LA),LA=1,MCHR)
140    CONTINUE
910    FORMAT (1H ,18(A5,2X),A5)
      RETURN
      END

      SUBROUTINE TIMEZ(IBIG)
      DIMENSION IBIG(19)
      character*9 adate
      character*3 mmm
      CHARACTER*8 TIME1
      CALL DATE(adate)
c      decode(9,645,adate)iday,mmm,iyr
      read(adate,645)iday,mmm,iyr
645    format(i2,1x,a3,1x,i2)
      iyr=iyr+2000
      WRITE(6,11)IDAY,mmm,IYR
      CALL TIME(TIME1)
c      DECODE(8,456,TIME1)IHR,IMIN
      read(TIME1,456)IHR,IMIN
```

```
456     FORMAT(I2,1x,i2)
        WRITE(6,22)IHR,IMIN
        WRITE(3,11)IDAY,mmm,IYR
        WRITE(3,22)IHR,IMIN
11     FORMAT(///T10,'DATE',2X,I2.2,' ',a3,' ',I4)
22     FORMAT(T10,'TIME ',I2.2,':',I2.2,3X,'HOURS'////)
        WRITE(6,*)
        WRITE(3,47)
        WRITE(3,47)
        WRITE(6,*)
        WRITE(6,*)
        WRITE(3,47)
        WRITE(3,47)
47     FORMAT(/)
        IF(IHR.LT.0.OR.IHR.GT.24)GOTO 4
        IF(IHR.LT.12)GOTO 1
        IF(IHR.EQ.12.AND.IMIN.LT.15)GOTO 3
        IF(IHR.LT.16)GOTO 2
        IF(IHR.LT.20)GOTO 5
        WRITE(6,167)IBIG
167    FORMAT(' Good night ',19A1)
        RETURN
5     WRITE(6,111)IBIG
111    FORMAT(' Good evening ',19A1)
        RETURN
1     WRITE(6,115)IBIG
115    FORMAT(' Good morning ',19A1)
        RETURN
2     WRITE(6,112)IBIG
112    FORMAT(' Good afternoon ',19A1)
        RETURN
3     WRITE(6,114)IBIG
114    FORMAT(' Good day ',19A1)
        RETURN
4     WRITE(6,113)IBIG
113    FORMAT(' Hi ! ',19A1)
        RETURN
        END

SUBROUTINE ENERGY(GX,GY,SX,SY,NBY21,FR,DF,NSE,NUN,DELTA F)
DIMENSION GX(NBY21),GY(NBY21)
```

```

REAL NSE
SX=0.
SY=0.
DO I=1,NBY21
SX=SX+GX(I)
SY=SY+GY(I)
END DO
IF(NUN.EQ.1)GOTO 1
SX=SX*DELTA F
SY=SY*DELTA F
1 WRITE(3,1500)FR,DF,NSE,SX,SY
1500 FORMAT(/1X,'FREQ RESOLUTION=',F7.4,5X,'DEG OF FREEDOM=',F4.0
1 ,5X,'NSE=',F7.4/1X,'ENERGY X SERIES =',F10.3,5X,'ENERGY Y SERIES =
',F10.3)
RETURN
END
CS SUBROUTINE ATTEN(CORR,FREQ,N)
C
C THE SUBROUTINE GENERATES THE CORRECTIONS TO BE APPLIED TO THE RESU
C OF HEIGHT ANALYSIS OF SHIPBORNE WAVERECORDER DATA
C
CS DIMENSION CORR(N),FREQ(N)
CS X1=0.09498**2
CS X2=X1*(1.916**2)
CS X3=0.1065**2
CS X4=X3*(1.241**2)
CS DO 1 J=1,N
CS IF(FREQ(J).GT.0.05)GOTO 2
CS 1 CORR(J)=1.
CS 2 DO 3 I=J,N
CS IF(FREQ(I).GE.0.2)GOTO 4
CS W2=(6.283*FREQ(I))**2
CS W4=W2**2
CS ALPHA=(X1-W2)**2+X2*W2
CS BETA=(X3-W2)**2+X4*W2
CS GAMA=ALPHA*BETA
CS OMEGA=SQRT(GAMA)
CS CORR(I)=W4*EXP(0.535168*W2)/OMEGA
CS 3 CONTINUE
CS 4 DO 5 K=I,N
CS 5 CORR(K)=2.23

```

```

CS      RETURN
CS      END
        SUBROUTINE OUT1(T,SPLUS,SMINUS,STOTAL,CRS,RSIG,STAB,THEOR,F,NBY21)
        DIMENSION T(NBY21),SPLUS(NBY21),SMINUS(NBY21),STOTAL(NBY21)
        DIMENSION CRS(NBY21),RSIG(NBY21),THEOR(NBY21),F(NBY21),STAB(NBY21)
        WRITE(3,1000)
        DO I=1,NBY21
        WRITE(3,2000)I,T(I),SPLUS(I),SMINUS(I),STOTAL(I),CRS(I),
1  RSIG(I),STAB(I),THEOR(I),F(I)
        END DO
        RETURN
1000    FORMAT(///T3,'NO.',
1  T8,'PERIOD',
2  T17,'SPLUS',
3  T24,'SMINUS',
5  T32,'STOTAL',
1  T43,'CRS',
3  T50,'RSIG',
4  T58,'STAB',
3  T65,'THEOR',
6  T76,'FREQ'/)

c 200   0  FORMAT(1X,I4,8F10.2,F10.4)
2000    FORMAT(1X,I4,8F8.2,F10.4)
        END

        SUBROUTINE OUT(T,GXY,GX,GY,RES,RET,COH,PHASE,F,NBY21)
        DIMENSION T(NBY21),GXY(NBY21),GX(NBY21),GY(NBY21),RES(NBY21)
        DIMENSION RET(NBY21),COH(NBY21),PHASE(NBY21),F(NBY21)
        WRITE(3,1000)
        DO I=1,NBY21
        WRITE(3,2000)I,T(I),GXY(I),GX(I),GY(I),RES(I),RET(I),COH(I),
1  PHASE(I),F(I)
        END DO
        RETURN
1000    FORMAT(///T3,'NO.',
1  T7,'PERIOD',
2  T18,'GXY',
3  T27,'GX',
4  T35,'GY',
5  T38,'RESXY/X',

```

```

1  T47,'RESY/X',
2  T56,'COHER',
3  T64,'PHASE Y-X',
6  T75,'FREQ')
2000 FORMAT(I4,8F8.2,F10.4)
      END

```

```

SUBROUTINE MAXIMUM(NPL,XMY,GXY,GX,GY,N)
DIMENSION GXY(N),GX(N),GY(N)
XMY=0.
DO I=1,N
IF(GXY(I).GT.XMY)XMY=GXY(I)
IF(GX(I).GT.XMY)XMY=GX(I)
END DO
IF(NPL.EQ.2)RETURN
DO I=1,N
IF(GY(I).GT.XMY)XMY=GY(I)
END DO
RETURN
END

```

```

SUBROUTINE WINDOW(N,IWT,FACT,WEIGHT,B)
DIMENSION A1(5),A2(5),A3(5),SCALE(5),W(5),B(N)
DATA A1,A2,A3/1.0,0.5,0.54,0.42,0.5,0.0,0.5,0.46,0.5,0.5,3*0.0,0.0
1  8,0.0/,W/0.88,1.44,1.33,1.64,1.03/,SCALE/1.0,0.3750,0.3974,0.303,0
2  .875/
IF(IWT.LT.1.OR.IWT.GT.5)IWT=1
WEIGHT=W(IWT)
FACT=SCALE(IWT)
NBY2=(N-1)/2+1
GOTO (120,30,30,20,10),IWT
10  PHIN=10*3.14159/N
ALPHA=PHIN*(1-(N+1)/2.)
GOTO 40
20  PHIN=2*3.14159/N
ALPHA=PHIN*(1-(N+1)/2.)
C2=COS(2*ALPHA)*A3(IWT)
C4=COS(2*PHIN)

```

```
S2=SIN(2*ALPHA)*A3(IWT)
S4=SIN(2*PHIN)
GOTO 40
30 PHIN=2*3.14159/N
ALPHA=PHIN*(1-(N+1)/2.)
40 C1=COS(ALPHA)*A2(IWT)
C3=COS(PHIN)
S1=SIN(ALPHA)*A2(IWT)
S3=SIN(PHIN)
GOTO (120,50,50,100,70),IWT
50 DO 60 I=1,NBY2
B(I)=A1(IWT)+C1
SH=S1*C3+S3*C1
C1=C1*C3-S1*S3
60 S1=SH
GOTO 120
70 N25=N/10
DO 80 I=1,N25
B(I)=A1(IWT)+C1
SH=S1*C3+S3*C1
C1=C1*C3-S1*S3
80 S1=SH
N251=N25+1
DO 90 I=N251,NBY2
90 B(I)=1.
GOTO 120
100 DO 110 I=1,NBY2
B(I)=A1(IWT)+C1+C2
SH=S1*C3+S3*C1
C1=C1*C3-S1*S3
S1=SH
SH=S2*C4+S4*C2
C2=C2*C4-S2*S4
110 S2=SH
120 RETURN
END
SUBROUTINE PAR(OUTFIL,IBIG,NUN,NP,NIJ,DELTT,
# FACTORX,FACTORY,ITREND,IWT,IPRINT,IROT,INFILX,INFILY,
# AXBIG,NUMBX,AYBIG,NUMBY,NLABX,NLABY)
DIMENSION IBIG(19)
CHARACTER*1 DUMMY
```



```

CHARACTER*25 OUTFIL,INFILX,INFILY,AXBIG,AYBIG
OPEN(15,FILE='cross-spec.spc',access='sequential',BLANK='ZERO')
READ(15,'(50X,A25)')DUMMY
READ(15,'(50X,A25)')OUTFIL
READ(15,'(50X,19A1)')IBIG
READ(15,'(50X,I1)')NUN
READ(15,'(50X,I5)')NP
READ(15,'(50X,I2)')NIJ
READ(15,'(50X,F7.3)')DELTT
READ(15,'(50X,F7.2)')FACTORX
READ(15,'(50X,F7.2)')FACTORY
READ(15,'(50X,I1)')ITREND
IF(ITREND.EQ.0)ITREND=2
READ(15,'(50X,I1)')IWT
READ(15,'(50X,I1)')IPRINT
READ(15,'(50X,I1)')IROT
READ(15,'(50X,A25)')INFILX
READ(15,'(50X,A25)')INFILY
READ(15,'(50X,A25)')AXBIG
READ(15,'(50X,I2)')NUMBX
READ(15,'(50X,A25)')AYBIG
READ(15,'(50X,I2)')NUMBY
READ(15,'(50X,I2)')NLABX
READ(15,'(50X,I2)')NLABY
OPEN(39,FILE='BEE')
WRITE(39,1)AXBIG,AYBIG
1  FORMAT(A25,5X,A25)
RETURN
END

```

```

subroutine newfft(xr,xi,sgn,list,np)
parameter (im1=8192,im1by21=im1/2+1)
dimension xr(im1),xi(im1),wc(im1,im1),ws(im1,im1)
dimension yr(im1),yi(im1),gx(im1by21)

if(list.gt.0) go to 1
nby2=np/2
nby21=nby2+1
write(*,*)np,nby2,nby21

twopin=atan(1.)*4.*2./np

```

```
do k=1,np
do ni=1,np
phi=twopin*(k-1)*(ni-1)
wc(k,ni)=cos(phi)
ws(k,ni)=sin(phi)
end do

end do
list=1
1 continue

if(sgn.lt.0.)then
do k=1,np
sumc=0.
sums=0.
do ni=1,np
sumc=sumc+xr(ni)*wc(k,ni)+xi(ni)*ws(k,ni)
sums=sums+xi(ni)*wc(k,ni)-xr(ni)*ws(k,ni)
end do
yr(k)=sumc
yi(k)=sums
end do

amul=4./np/np
do i=1,nby21
gx(i)=amul*(yr(i)**2+yi(i)**2)
end do

do i=1,np
xr(i)=yr(i)
xi(i)=yi(i)
end do

20 format(4e14.4)

else

do k=1,np
sumc=0.
```

```

sums=0.
do ni=1,np
sumc=sumc+xr(ni)*wc(k,ni)-xi(ni)*ws(k,ni)
sums=sums+xi(ni)*wc(k,ni)+xr(ni)*ws(k,ni)
end do
yr(k)=sumc
yi(k)=sums
end do

do i=1,np
xr(i)=yr(i)
xi(i)=yi(i)
end do

end if

return
end

SUBROUTINE DATA22(XN,YN,NP,IJK,ITREND,AXBIG,AYBIG,NUMBX,NUMBY,
# INFILX,INFILY,NLABX,NLABY)
PARAMETER(IM1=8192)
DIMENSION XN(IM1),YN(IM1)
CHARACTER*25 AXBIG,AYBIG
CHARACTER*25 INFILX,INFILY
CHARACTER*80 ALAB(99),BLAB(99)
IF(IJK.GT.1)GOTO 5
NUM2X=NUMBX*2
IF(NP.LT.NUM2X) NUM2X=NP
NUM2Y=NUMBY*2
IF(NP.LT.NUM2Y) NUM2Y=NP

OPEN(10,FILE=INFILX,BLANK='ZERO',ACCESS='SEQUENTIAL')
OPEN(11,FILE=INFILY,BLANK='ZERO',ACCESS='SEQUENTIAL')
1135 FORMAT(2A1)
1235 FORMAT(8E10.3)
112 FORMAT(A80)
IF(NLABX.GT.0)THEN
READ(10,112)(ALAB(I),I=1,NLABX)
WRITE(3,113)(ALAB(I),I=1,NLABX)
END IF

```

```

IF(NLABY.GT.0)THEN
READ(11,112)(BLAB(I),I=1,NLABY)
WRITE(3,113)(BLAB(I),I=1,NLABY)
END IF
5 CONTINUE
READ(10,AXBIG)(XN(I),I=1,np)
READ(11,AYBIG)(YN(I),I=1,np)

53 IF(IJK.GT.1)GOTO 68
IF(NLABX.GT.0)WRITE(*,113)(ALAB(I),I=1,NLABX)
WRITE(*,1235)(XN(I),I=1,NUM2X)
WRITE(*,6588)
6588 format(' Are these the first 2 lines of X series? (Y/N) :',)$
CALL ENTRY(IWW,2,'Y','N','3','4','5')
IF(IWW.EQ.2)STOP' Check your input series X !!'
IF(NLABY.GT.0)WRITE(*,113)(BLAB(I),I=1,NLABY)
113 FORMAT(1X,A80)
WRITE(*,1235)(YN(I),I=1,NUM2Y)
WRITE(*,6589)
6589 format(' Are these the first 2 lines of Y series? (Y/N) :',)$
CALL ENTRY(IWY,2,'Y','N','3','4','5')
IF(IWY.EQ.2)STOP' Check your input series Y !!'
68 WRITE(3,125)IJK
WRITE(3,1235)(XN(I),I=1,NP)
WRITE(3,126)IJK
WRITE(3,1235)(YN(I),I=1,NP)
126 FORMAT(///' ENSEMBLE NUMBER =',I2,5X,'.....SERIES Y'/)
125 FORMAT(///' ENSEMBLE NUMBER =',I2,5X,'.....SERIES X' /)
226 FORMAT(///' ENSEMBLE NUMBER =',I2,5X,'.....SERIES Y WITH TREND
1 REMOVED'/)
225 FORMAT(///' ENSEMBLE NUMBER =',I2,5X,'.....SERIES X WITH TREND
1 REMOVED'/)
IF(ITREND.EQ.2)RETURN
CALL TREND(XN,NP)
CALL TREND(YN,NP)
WRITE(3,225)IJK
WRITE(3,1235)(XN(I),I=1,NP)
WRITE(3,226)IJK
WRITE(3,1235)(YN(I),I=1,NP)
RETURN
END

```



```

-      '$$$$ ', '$$$$ ' /
C ---

```

## A.5 Canonical Correlation Annalysis

### A.5.1 Computer Program: canonical.f

```

PROGRAM CVA
C THIS PROGRAM CALLS SUBROUTINE DATA TO READ THE DATA
C MATRIX A;
C THEN CALLS SUBROUTINE UNSYM TO COMPUTE EIGEN VALUES AND
C EIGEN VECTORS OF MATRIX SD
C FINALLY CONSTRUCTS CANONICAL VARIATES.
C
C INPUT VARIABLE FOR PROGRAM CVA
C NPTB-WHEN NPTB=0, THE COVARIANCE MATRICES ARE NOT PRINTED.
C
C INPUT VARIABLE FOR SUBROUTINE DATA
C N=NO. OF SAMPLE UNITS.
C NP=NO. OF VARITES IN THE FIRST SET.
C NQ=NO. OF VARITIES IN THE SECOND SET.
C NPTA-WHEN NPTA=0, PRINT OUT OF DATA IS SUPPRESSED.
C
C THE DIMENSION STATEMENTS OF THE ARRAYS IN PROGRAM CVA DEPEND ON N,
C NP AND NQ AS FOLLOW:
C PROGRAM CVA
C DIMENSION A(N,NP+NQ),ZI(N,NP),ET(N,NP),AM(NP+NQ),SA(NP,NP),
C SB(NP,NQ),SC(NQ,NP),SD(NP,NP+NQ),VA(NP,NP),B(NQ,NP+NQ),
C RR(NP),RI(NP)
C SUBROUTINE DATA
C DIMENSION A(N,NP+NQ)
C SUBROUTINE UNSYM
C DIMENSION A(NQ,NP+NQ),S(NQ,NP+NQ),VEC(NP,NP),RR(1),RI(1)
C SUBROUTINE HSBG
C DIMENSION A(1)
C SUBROUTINE ATEIG
C DIMENSION A(1)
C SUBROUTINE MATINV
C DIMENSION C(NQ,NQ)
C DIMENSION A(32,30),ZI(32,10),ET(32,10),AM(30)

```

```
        DIMENSION SA(10,10),SB(10,20),SC(20,10),SD(30,30)
        DIMENSION RR(10),RI(10)
        DIMENSION VA(10,10),B(30,30)
201     FORMAT (5X,4F10.4)
100     FORMAT(5I3)
        open(7,file='pielou-page-357.dat',access='sequential',
# blank='zero')
        open(8,file='see-cca.res',access='sequential',status='unknown')
        READ(7,100)NPTB
        CALL DATA(A,N,NP,NQ,NPQ)
        DO 82 J=1,NPQ
        S=0.
        DO 81 K=1,N
81      S=S+A(K,J)
82      AM(J)=S/N
        DO 12 I=1,NP
        DO 12 J=I,NP
        S=0.0
        DO 10 K=1,N
10     S=S+A(K,I)*A(K,J)
        S=S/N-AM(J)*AM(I)
        SA(I,J)=S
12     SA(J,I)=S
        DO 14 I=1,NP
        DO 14 J=1,NQ
        S=0.0
        JNP=J+NP
        DO 13 K=1,N
13     S=S+A(K,I)*A(K,JNP)
        S=S/N-AM(I)*AM(JNP)
        SB(I,J)=S
14     SC(J,I)=S
        DO 16 I=1,NQ
        INP=I+NP
        DO 16 J=I,NQ
        JNP=J+NP
        S=0.0
        DO 15 K=1,N
15     S=S+A(K,INP)*A(K,JNP)
        S=S/N-AM(INP)*AM(JNP)
        SD(I,J)=S
```

```
16      SD(J,I)=S
        IF(NPTB.EQ.0)GO TO 518
        WRITE(8,302)
        DO 512 I=1,NP
512     WRITE(8,206)J,(SA(I,J),J=1,NP)
        WRITE(8,303)
        DO 514 J=1,NQ
514     WRITE(8,206)J,(SC(J,I),I=1,NP)
        WRITE(8,304)
        DO 516 I=1,NQ
516     WRITE(8,206)I,(SD(I,J),J=1,NQ)
518     CONTINUE
        DO 18 I=1,NP
        DO 18 J=1,NP
18      B(I,J)=SA(I,J)
        DO 20 I=1,NP
        DO 20 J=1,NQ
        JNP=J+NP
20      B(I,JNP)=SB(I,J)
        EP=1.E-30
        CALL MATINV(-1, B,NP,NQ,DET,EP)
        DO 22 I=1,NP
        DO 22 J=1,NQ
        JNP=J+NP
22      SB(I,J)= B(I,JNP)
        DO 26 I=1,NQ
        DO 26 J=1,NP
        JNQ=J+NQ
26      SD(I,JNQ)=SC(I,J)
        CALL MATINV(-1,SD,NQ,NP,DET,EP)
        DO 28 I=1,NQ
        DO 28 J=1,NP
        JNQ=J+NQ
28      SC(I,J)=SD(I,JNQ)
        SS=0
        DO 31 I=1,NP
        DO 30 J=1,NP
        SD(I,J)=0
        DO 30 K=1,NQ
30      SD(I,J)=SD(I,J)+SB(I,K)*SC(K,J)
31      SS=SS+SD(I,I)
```



```
WRITE(8,308)SS
MAX=30
CALL UNSYM(SD,MAX,NP, B,RR,RI,VA,NPR)
WRITE(8,305)
DO 520 J=1,NPR
520 WRITE(8,206)J,(VA(K,J),K=1,NP)
DO 32 I=1,N
DO 32 J=1,NPR
ZI(I,J)=0.0
DO 32 K=1,NP
32 ZI(I,J)=ZI(I,J)+A(I,K)*VA(K,J)
DO 35 J=1,NPR
DO 35 K=1,NQ
SS=0.
DO 34 L=1,NP
34 SS=SS+SC(K,L)*VA(L,J)
IF(K.EQ.1)SX=SS
35 SD(K,J)=SS/SX
WRITE(8,306)
DO 522 J=1,NPR
522 WRITE(8,206)J,(SD(K,J),K=1,NQ)
DO 36 I=1,N
DO 36 J=1,NPR
ET(I,J)=0.0
DO 36 K=1,NQ
KP=K+NP
36 ET(I,J)=ET(I,J)+A(I,KP)*SD(K,J)
WRITE(8,307)
DO 524 J=1,N
524 WRITE(8,206)J,(ZI(J,K),ET(J,K),K=1,NPR)
DO 40 J=1,NPR
JNR=NPR+J
ZIS=0.
ETS=0.
SSA=0.
SSB=0.
DO 38K=1,N
SSA=SSA+ZI(K,J)
ZIS=ZIS+ZI(K,J)**2
ETS=ETS+ET(K,J)**2
38 SSB=SSB+ET(K,J)
```

```

      AM(J)=SSA/N
      AM(JNR)=SSB/N
      RR(J)=SQRT(ZIS/N-AM(J)**2)
      RI(J)=SQRT(ETS/N-AM(JNR)**2)
40    CONTINUE
      DO 44 J=1,NPR
      DO 44 K=1,NPR
      KNR=K+NPR
      S=0.
      DO 42 L=1,N
42    S=S+ZI(L,J)*ET(L,K)
44    SA(J,K)=(S/N-AM(J)*AM(KNR))/RR(J)/RI(J)
      WRITE(8,309)
      DO 50 J=1,NPR
50    WRITE(8,206)J,(SA(J,K),K=1,NPR)
206   FORMAT(I5,8E12.4)
302   FORMAT(/5X,46HWITHIN-SET COVARIANCE MATRIX FOR THE FIRST SET)
303   FORMAT(/5X,29HBETWEEN-SET COVARIANCE MATRIX)
304   FORMAT(/5X,47HWITHIN-SET COVARIANCE MATRIX FOR THE SECOND SET)
305   FORMAT(/5X,33HTRANSFORMATION MATRIX TO GENERATE/5X,36HTHE FIRST S
1    I ET OF CANONICAL VARIATES.)
306   FORMAT(/5X,33HTRANSFORMATION MATRIX TO GENERATE /5X,37HTHE SECOND
1    I SET OF CANONICAL VARIATES.)
307   FORMAT(/5X,19HCANONICAL VARIATES./5X,37HZI(J) AND ETA(J) IN ALTER
1    I NATE COLUMNS)
308   FORMAT(/5X,20HTRACE OF THE MATRIX=,E12.4/)
309   FORMAT(/5X,22HCANONICAL CORRELATIONS)
      STOP
      END
      SUBROUTINE DATA(A,N,NP,NQ,NPQ)
C S  SUBROUTINE DATA MAY BE SUITABLY WRITTEN BY THE USER.
C I  IF DATA PREPARATION IS NEEDED, IT MAY BE INCLUDED IN THIS
C R  OUTINE.
      DIMENSION A(32,30)
      READ(7,100)N,NP,NQ,NPTA
      NPQ=NP+NQ
      NPA=NP+1
      DO 20 J=1,NPQ
20    READ(7,101)(A(I,J),I=1,N)
101   FORMAT(10F4.2)
      IF(NPTA.EQ.0)RETURN

```

```

WRITE(8,300)
DO 50 J=1,N
50 WRITE(8,200)J,(A(J,K),K=1,NP)
WRITE(8,301)
DO 52 J=1,N
52 WRITE(8,200)J,(A(J,K),K=NPA,NPQ)
100 FORMAT(5I3)
200 FORMAT(I5,10F10.4/)
300 FORMAT(/5X,32HMATRIX OF FIRST SET OF VARIATES.)
301 FORMAT(/5X,33HMATRIX OF SECOND SET OF VARIATES.)
RETURN
END
SUBROUTINE UNSYM(A,MAX,N,S,RR,RI,VEC,NR)
C T HIS ROUTINE CALLS SUBROUTINES HSBG & ATEIG TO COMPUTE THE
C E I GEN VALUES OF THE MATRIX A
C T HEN CONSTRUCTS THE EIGEN VECTORS.
DIMENSION A(30,30),S(30,30),VEC(10,10)
DIMENSION RR(1),RI(1)
DO 10 J=1,N
DO 10 K=1,N
10 S(J,K)=A(J,K)
CALL HSBG(N,S,MAX)
CALL ATEIG(N,S,RR,RI,MAX)
WRITE(8,200)(RR(J),RI(J),J=1,N)
c WRITE(8,200)((RR(J),RI(J)),J=1,N)
SR=0.
SI=0.
DO 12 J=1,N
SR=SR+RR(J)
12 SI=SI+RI(J)
WRITE(8,204)SR,SI
NTIM=4
EP=1.E-30
NR=0
DO 38 KK=1,N
IF(RI(KK).NE.0) GO TO 38
IF(RR(KK).LE.0.) GO TO 38
NR=NR+1
E=RR(KK)
RR(NR)=RR(KK)
DO 29 J=1,N

```

```
29      S(J,N+1)=1.
        DO 36 JJ=1,NTIM
        DO 31 J=1,N
        DO 30 K=1,N
30      S(J,K)=A(J,K)
31      S(J,J)=S(J,J)-E*1.01
        CALL MATINV(-1,S,N,1,DET,EP)
        SS=S(1,N+1)
        DO 34 J=1,N
34      S(J,N+1)=S(J,N+1)/SS
36      CONTINUE
        DO 37 J=1,N
37      VEC(J,NR)=S(J,N+1)
38      CONTINUE
200     FORMAT(/5X,11HEIGENVALUES //10X,4HREAL,6X,
# 9HIMAGINARY/(5X,2E12.4))
202     FORMAT(/,5X,I5)
203     FORMAT(1H+,10X,8E12.4/)
204     FORMAT(/5X,24HSUM OF THE EIGEN VALUES= ,2E12.4/)
        RETURN
        END
        SUBROUTINE HSBG(N,A,IA)
C      SUBROUTINE HSBG
C      PURPOSE
C      TO REDUCE A REAL MATRIX INTO UPPER ALMOST TRIANGULAR FORM
C      USAGE
C      CALL HSBG(N,A,IA)
C      DECIPTION OF THE PARAMETERS
C      N ORDER OF THE MATRIX
C      A THE IMPUT MATRIX,N BY N
C      IA=SIZE OF THE FIRST DIMESNION ASSIGNED TO THE ARRAY A
C      IN THE CALLING PROGRAM WHEN THE MATRIX IS IN
C      DOUBLE SUBSCRIBTED DATA STORAGE MODE. IA=N WHEN
C      THE MATRIX IS IN SSP VECTOR STORAGE MODE.
C      REMARKS
C      THE HESSENBERG FORM REPLACES THE ORIGINAL MATRIX IN THE
C      ARRAY A.
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      NONE
C      METHOD
C      SIMILARITY TRANSFORMATIONS USING ELEMENTARY TRANSFORMATION
```

```
C      MATRICES, WITH PARTIAL PIVOTING.
C      REFERENCES
C      J.H. WILKINSON-THE ALGEBRAIC EIGENVALUE PROBLEM-
C      CLARENDON PRESS, OXFORD,1965.
      DIMENSION A(1)
      DOUBLE PRECISION S
      L=N
      NIA=L*IA
      LIA=NIA-IA
C      L IS THE ROW INDEX OF THE ELIMINATION
20     IF(L-3) 360,40,40
40     LIA=LIA-IA
      L1=L-1
      L2=L1-1
C      SEARCH FOR THE PIVOTAL ELEMENT IN THE LTH ROW
      ISUB=LIA+L
      IPIV=ISUB-IA
      PIV=ABS(A(IPIV))
      IF(L-3)90,90,50
50     M=IPIV-IA
      DO 80 I=L,M,IA
      T=ABS(A(I))
      IF(T-PIV) 80,80,60
60     IPIV=I
      PIV=T
80     CONTINUE
90     IF(PIV) 100,320,100
100    IF(PIV-ABS(A(ISUB))) 180,180,120
C      INTERCHANGE THE COLUMNS
120    M=IPIV-L
      DO 140 I=1,L
      J=M+I
      T=A(J)
      K=LIA+I
      A(J)=A(K)
140    A(K)=T
C      INTERCHANGE THE ROWS
      M=L2-M/IA
      DO 160 I=L1,NIA,IA
      T=A(I)
      J=I-M
```

```

      A(I)=A(J)
160   A(J)=T
C     TERMS OF THE ELEMENTARY TRANSFORMATION
180   DO 200 I=L,LIA,IA
200   A(I)=A(I)/A(ISUB)
C     RIGHT TRANSFORMATION
      J=-IA
      DO 240 I=1,L2
      J=J+IA
      LJ=L+J
      DO 220 K=1,L1
      KJ=K+J
      KL=K+LIA
220   A(KJ)=A(KJ)-A(LJ)*A(KL)
240   CONTINUE
C     LEFT TRANSFORMATION
      K=-IA
      DO 300 I=1,N
      K=K+IA
      LK=K+L1
      S=A(LK)
      LJ=L-IA
      DO 280 J=1,L2
      JK=K+J
      LJ=LJ+IA
280   S=S+A(LJ)*A(JK)*1.0D0
300   A(LK)=S
C     SET THE LOWER PART OF THE MATRIX TO ZERO
      DO 310 I=L,LIA,IA
310   A(I)=0.0
320   L=L1
      GO TO 20
360   RETURN
      END
      SUBROUTINE ATEIG(M,A,RR,RI,IA)
C     SUBROUTINE ATEIG
C     PURPOSE
C     COMPUTE THE EIGENVALUES OF A REAL ALMOST TRIANGULAR MATRIX
C     USAGE
C     CALL ATEIG(M,A,RR,RI,IANA,IA)
C     DESCRIPTION OF THE PARAMETERS
```

```
C      M ORDER OF THE MATRIX
C      A THE INPUT MATRIX, M BY M
C      RR VECTOR CONTAINING THE REAL PARTS OF THE EIGENVALUES
C      ON RETURN
C      RI VECTOR CONTAINING THE IMAGINARY PARTS OF THE EIGEN-
C      VALUES ON RETURN
C      IA SIZE OF THE FIRST DIMENSION ASSIGNED TO THE ARRAY A
C      IN THE CALLING PROGRAM WHEN THE MATRIX IS IN DOUBLE
C      SUBSCRIPTED DATA STORAGE MODE.
C      IA=M WHEN THE MATRIX IS IN SSP VECTOR STORAGE MODE.
C      THE ORIGINAL MATRIX IS DESTROYED
C      REMARKS
C      THE DIMENSION OF RR AND RI MUST BE GREATER OR EQUAL TO M
C      SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C      NONE
C      METHOD
C      QR DOUBLE ITERATION
C      REFENCES
C      J.G.F.FRANCIS-THE QR TRANSFORMATION — THE COMPUTER
C      JOURNAL,VOL. 4,NO. 3, OCTOBER 1961,VOL. 4,NO. 4, JANUARY
C      1962. J H. WILKINSON - THE ALGEBRIC EIGENVALUE PROBLEM -
C      CLARENDON PRESS, OXFORD, 1965.
      DIMENSION A(1),RR(1),RI(1),PRR(2),PRI(2)
      INTEGER P,P1,Q
      E7=1.0E-8
      E6=1.0E-6
      E10=1.0E-10
      DELTA=0.5
      MAXIT=30
C      INITIALIZATION
      N=M
20     N1=N-1
      IN=N1*IA
      NN=IN+N
      IF(N1)30,1300,30
30     NP=N+1
C      ITERATION COUNTER
      IT=0
C      ROOTS OF THE 2ND ORDER MAIN SUBMATRIX AT THE PREVIOUS
C      ITERATION
      DO 40 I=1,2
```

```
      PRR(I)=0.0
40     PRI(I)=0.0
C      LAST TWO SUBDIAGONAL ELEMENTS AT THE PREVIOUS ITERATION
      PAN=0.0
      PAN1=0.0
C      ORIGIN SHIFT
      R=0.0
      S=0.0
C      ROOTS OF THE LOWER MAIN 2 BY 2 SUBMATRIX
      N2=N1-1
      IN1=IN-IA
      NN1=IN1+N
      N1N=IN+N1
      N1N1=IN1+N1
60     T=A(N1N1)-A(NN)
      U=T*T
      V=4.0*A(N1N)*A(NN1)
      IF(ABS(V)-U*E7)100,100,65
65     T=U+V
      UMAX=U
      IF (U.LT.ABS(V)) UMAX=ABS(V)
      IF (ABS(T)-UMAX*E6)67,67,68
67     T=0.0
68     U=(A(N1N1)+A(NN))/2.
      V=SQRT(ABS(T))/2.0
      IF(T)140,70,70
70     IF(U)80,75,75
75     RR(N1)=U+V
      RR(N)=U-V
      GO TO 130
80     RR(N1)=U-V
      RR(N)=U+V
      GO TO 130
100    IF(T)120,110,110
110    RR(N1)=A(N1N1)
      RR(N)=A(NN)
      GO TO 130
120    RR(N1)=A(NN)
      RR(N)=A(N1N1)
130    RI(N)=0.0
      RI(N1)=0.0
```



```
GO TO 160
140 RR(N1)=U
RR(N)=U
RI(N1)=V
RI(N)=-V
160 IF(N2)1280,1280,180
C TESTS OF CONVERGENCE
180 N1N2=N1N1-IA
RMOD=RR(N1)*RR(N1)+RI(N1)*RI(N1)
EPS=E10*SQRT(RMOD)
IF(ABS(A(N1N2))-EPS)1280,1280,240
240 IF(ABS(A(NN1))-E10*ABS(A(NN))) 1300,1300,250
250 IF(ABS(PAN1-A(N1N2))-ABS(A(N1N2))*E6) 1240,1240,260
260 IF(ABS(PAN-A(NN1))-ABS(A(NN1))*E6)1240,1240,300
300 IF(IT-MAXIT)320,1240,1240
C COMPUTE THE SHIFT
320 J=1
DO 360 I=1,2
K=NP-I
IF(ABS(RR(K)-PRR(I))+ABS(RI(K)-PRI(I))-DELTA*(ABS(RR(K))
1 +ABS(RI(K)))) 340,360,360
340 J=J+I
360 CONTINUE
GO TO (440,460,460,480),J
440 R=0.0
S=0.0
GO TO 500
460 J=N+2-J
R=RR(J)*RR(J)
S=RR(J)+RR(J)
GO TO 500
480 R=RR(N)*RR(N1)-RI(N)*RI(N1)
S=RR(N)+RR(N1)
C SAVE THE LAST TWO SUBDIAGONAL TERMS AND THE ROOTS OF THE
C SUBMATRIX BEFORE ITERATION
500 PAN=A(NN1)
PAN1=A(N1N2)
DO 520 I=1,2
K=NP-I
PRR(I)=RR(K)
520 PRI(I)=RI(K)
```

```
C      SEARCH FOR A PARTITION OF THE MATRIX, DEFINED BY P AND Q
      P=N2
      IF(N-3)600,600,525
525    IPI=N1N2
      DO 580 J=2,N2
      IPI=IPI-IA-1
      IF(ABS(A(IPI))-EPS)600,600,530
530    IPIP=IPI+IA
      IPIP2=IPIP+IA
      D=A(IPIP)*(A(IPIP)-S)+A(IPIP2)*A(IPIP+1)+R
      IF(D)540,260,540
540    IF(ABS(A(IPI)*A(IPIP+1))*(ABS(A(IPIP)+A(IPIP2+1)-S)+ABS(A(IPIP2+2)
1  ))-ABS(D)*EPS)620,620,560
560    P=N1-J
580    CONTINUE
600    Q=P
      GO TO 680
620    P1=P-1
      Q=P1
      IF(P1-1)680,680,650
650    DO 660 I=2,P1
      IPI=IPI-IA-1
      IF(ABS(A(IPI))-EPS)680,680,660
C      QR DOUBLE ITERATION
660    Q=Q-1
680    II=(P-1)*IA+P
      DO 1220 I=P,N1
      II1=II-IA
      IIP=II+IA
      IF(I-P)720,700,720
700    IPI=II+1
      IPIP=IIP+1
C      INITIALIZATION OF THE TRANSFORMATION
      G1=A(II)*(A(II)-S)+A(IIP)*A(IPI)+R
      G2=A(IPI)*(A(IPIP)+A(II)-S)
      G3=A(IPI)*A(IPIP+1)
      A(IPI+1)=0.0
      GO TO 780
720    G1=A(II1)
      G2=A(II1+1)
      IF(I-N2)740,740,760
```

```
740      G3=A(II1+2)
        GO TO 780
760      G3=0.0
780      CAP=SQRT(G1*G1+G2*G2+G3*G3)
        IF(CAP)800,860,800
800      IF(G1)820,840,840
820      CAP=-CAP
840      T=G1+CAP
        PSI1=G2/T
        PSI2=G3/T
        ALPHA=2.0/(1.0+PSI1*PSI1+PSI2*PSI2)
        GO TO 880
860      ALPHA=2.0
        PSI1=0.0
        PSI2=0.0
880      IF(I-Q)900,960,900
900      IF(I-P)920,940,920
920      A(II1)=-CAP
        GO TO 960
940      A(II1)=-A(II1)
C      ROW OPERATION
960      IJ=II
        DO 1040 J=I,N
        T=PSI1*A(IJ+1)
        IF(I-N1)980,1000,1000
980      IP2J=IJ+2
        T=T+PSI2*A(IP2J)
1000     ETA=ALPHA*(T+A(IJ))
        A(IJ)=A(IJ)-ETA
        A(IJ+1)=A(IJ+1)-PSI1*ETA
        IF(I-N1)1020,1040,1040
1020     A(IP2J)=A(IP2J)-PSI2*ETA
1040     IJ=IJ+IA
C      COLUMN OPERATION
        IF(I-N1)1080,1060,1060
1060     K=N
        GO TO 1100
1080     K=I+2
1100     IP=IIP-I
        DO 1180 J=Q,K
        JIP=IP+J
```

```
      JI=JIP-IA
      T=PSI1*A(JIP)
      IF(I-N1)1120,1140,1140
1120    JIP2=JIP+IA
      T=T+PSI2*A(JIP2)
1140    ETA=ALPHA*(T+A(JI))
      A(JI)=A(JI)-ETA
      A(JIP)=A(JIP)-ETA*PSI1
      IF(I-N1)1160,1180,1180
1160    A(JIP2)=A(JIP2)-ETA*PSI2
1180    CONTINUE
      IF(I-N2)1200,1220,1220
1200    JI=II+3
      JIP=JI+IA
      JIP2=JIP+IA
      ETA=ALPHA*PSI2*A(JIP2)
      A(JI)=-ETA
      A(JIP)=-ETA*PSI1
      A(JIP2)=A(JIP2)-ETA*PSI2
1220    II=IIP+1
      IT=IT+1
      GO TO 60
C      END OF ITERATION
1240    IF(ABS(A(NN1))-ABS(A(N1N2))) 1300,1280,1280
C      TWO EIGENVALUES HAVE BEEN FOUND
1280    CONTINUE
      N=N2
      IF(N2)1400,1400,20
C      ONE EIGENVALUE HAS BEEN FOUND
1300    RR(N)=A(NN)
      RI(N)=0.0
      IF(N1)1400,1400,1320
1320    N=N1
      GO TO 20
1400    RETURN
      END
      SUBROUTINE MATINV(IJOB, C,N,M,DET,EP)
C      THIS ROUTINE COMPUTES THE SOLUTION OF A SYSTEM OF EQUATIONS
C      CX=Y, WHERE C IS THE MATRIX OF COEFFICIENTS, X IS THE
C      SOLUTION VECTOR AND Y IS THE RIGHT HAND SIDE VECTOR.
C      THIS ROUTINE CAN ALSO BE USED TO FIND THE DETERMINANT
```

```
C      AND/OR INVERSE OF A MATRIX.
C      USAGE
C      CALL MATINV(IJOB,C,N,M,DET,EP)
C      DESSCRIPTION OF PARAMETERS
C      C-IS THE COEFFICIENT MATRIX AUGMENTED WITH VECTOR Y
C      APPENDED TO IT
C      IJOB- WHEN IJOB=-1, THE SYSTEM OF SIMULTANEOUS EQUATIONS
C      CX=Y IS SOLVED
C      IJOB=1,INVERSE OF THE MATRIX IS COMPUTED.
C      M - IS THE NUMBER OF COLUMN VECTORS APPENDED TO MATRIX C
C      N - NUMBER OF EQUATIONS .EQ. NUMBER OF UNKNOWNNS
C      C - IS THE COEFFICIENT MATRIX AUGMENTED WITH THE VECTOR
C      Y APPENDED TO IT.
C      METHOD
C      GAUSSIAN ELIMINATION
C      DIMENSION C(30,30)
30     FORMAT(5X,'THE DETERMINANT OF THIS SYSTEM EQUALS ZERO.
1     THE PROGRAM CANNOT HANDLE THIS CASE')
      DET=1.0
      NP1=N+1
      NPM=N+M
      NM1=N-1
      IF(IJOB)2,1,2
1     DO 3 I=1,N
      DO 20 J=NP1,NPM
20    C(I,J)=0.
      IPN=I+N
3     C(I,IPN)=1.
2     DO 66 J=1,NM1
      E=ABS(C(J,J))
      JP1=J+1
      DO 105 I=JP1,N
      D=ABS(C(I,J))
      IF(E-D)6,105,105
6     DET=-DET
      DO 7 K=J,NPM
      F=C(I,K)
      C(I,K)=C(J,K)
7     C(J,K)=F
      E=D
105    CONTINUE
```

```
IF(ABS(C(J,J))-EP)14,15,15
14  DET=0.0
    IF(IJOB)16,16,17
16  WRITE(8,30)
17  RETURN
15  DO 4 I=JP1,N
    CONST=C(I,J)/C(J,J)
    DO 4 K=JP1,NPM
    C(I,K)=C(I,K)-CONST*C(J,K)
4   CONTINUE
66  CONTINUE
IF(ABS(C(N,N))-EP)41,18,18
41  DET=0.0
    IF(IJOB)61,61,71
61  WRITE(8,30)
71  RETURN
18  DO 11 I=1,N
11  DET=DET*C(I,I)
    IF(IJOB)10,10,171
10  DO 12 I=1,N
    K=N-I+1
    KP1=K+1
    DO 12 L=NP1,NPM
    S=0.0
    DO 13 J=KP1,N
    IF(J.GT.N)GO TO 12
13  S=S+C(K,J)*C(J,L)
12  C(K,L)=(C(K,L)-S)/C(K,K)
171 RETURN
END
```

**A.5.2 Data: pielou-page-357.dat**

```
1
10 2 2 1
3. 4. 8. 12. 14. 16. 21. 24. 28. 30.
2. 4. 3. 6. 6. 7. 8. 8. 9. 11.
17. 14. 18. 10. 12. 10. 7. 9. 3. 3.
```

**A.5.3 Results: see-cca.res**

MATRIX OF FIRST SET OF VARIATES.

1	3.0000	2.0000
2	4.0000	4.0000
3	8.0000	3.0000
4	12.0000	6.0000
5	14.0000	6.0000
6	16.0000	7.0000
7	21.0000	8.0000
8	24.0000	8.0000
9	28.0000	9.0000
10	30.0000	11.0000

MATRIX OF SECOND SET OF VARIATES.

1	17.0000	19.0000
2	14.0000	18.0000
3	18.0000	16.0000
4	10.0000	18.0000
5	12.0000	13.0000
6	10.0000	12.0000
7	7.0000	11.0000
8	9.0000	8.0000
9	3.0000	5.0000
10	3.0000	2.0000

WITHIN-SET COVARIANCE MATRIX FOR THE FIRST SET

3	0.8260E+02	0.2320E+02
3	0.2320E+02	0.7040E+01

BETWEEN-SET COVARIANCE MATRIX

1	-0.4120E+02	-0.1252E+02
2	-0.4820E+02	-0.1338E+02

WITHIN-SET COVARIANCE MATRIX FOR THE SECOND SET

1	0.2401E+02	0.2324E+02
---	------------	------------



2 0.2324E+02 0.3036E+02

TRACE OF THE MATRIX= 0.1315E+01

EIGENVALUES

REAL	IMAGINARY
0.9714E+00	0.0000E+00
0.3434E+00	0.0000E+00

SUM OF THE EIGEN VALUES= 0.1315E+01 0.0000E+00

TRANSFORMATION MATRIX TO GENERATE  
THE FIRST SET OF CANONICAL VARIATES.

1	0.1000E+01	0.2615E+01
2	0.1000E+01	-0.3443E+01

TRANSFORMATION MATRIX TO GENERATE  
THE SECOND SET OF CANONICAL VARIATES.

1	0.1000E+01	0.8957E+00
2	0.1000E+01	-0.8888E+00

CANONICAL VARIATES.

ZI(J) AND ETA(J) IN ALTERNATE COLUMNS

1	0.8229E+01	0.3402E+02	-0.3886E+01	0.1125E+00
2	0.1446E+02	0.3012E+02	-0.9773E+01	-0.1999E+01
3	0.1584E+02	0.3233E+02	-0.2329E+01	0.3779E+01
4	0.2769E+02	0.2612E+02	-0.8659E+01	-0.5999E+01
5	0.2969E+02	0.2364E+02	-0.6659E+01	0.4454E+00
6	0.3430E+02	0.2075E+02	-0.8102E+01	-0.6658E+00
7	0.4192E+02	0.1685E+02	-0.6545E+01	-0.2777E+01
8	0.4492E+02	0.1617E+02	-0.3545E+01	0.1889E+01
9	0.5153E+02	0.7478E+01	-0.2988E+01	-0.1444E+01
10	0.5876E+02	0.4791E+01	-0.7875E+01	0.1222E+01

## CANONICAL CORRELATIONS

1	-0.9856E+00	0.3095E-07
2	-0.3353E-05	0.5860E+00

# Bibliography

- T. P. Barnett. Statistical prediction of north american air temperatures from Pacific predictors. *Monthly Weather Review*, 109(5):1021–1041, 1981.
- T. P. Barnett. Interaction of the monsoon and Pacific trade wind system at interannual time scales. part i: The equatorial zone. *Monthly Weather Review*, 111(4):756–773, 1983.
- T. P. Barnett. Interaction of the monsoon and Pacific trade wind system at interannual time scales. part ii: The tropical band. *Monthly Weather Review*, 112(4):756–773, 1984.
- T. P. Barnett and R. Preisendorfer. Origins and levels of monthly and seasonal forecast skill for United States surface air temperatures determined by canonical correlation analysis. *Monthly Weather Review*, 115(9):1825–1850, 1987.
- S. Basu, S. D. Meyers, and J. J. O'Brien. Annual and interannual sea level variations in the indian ocean from topex/poseidon observations and ocean model simulations. *Journal of Geophysical Research*, 105(C1):975–994, 2000.
- J. S. Bendat and A. G. Piersol. *Random data: Analysis and Measurement procedures*. John Wiley and Sons, New York, 1971.
- C. S. Bretherton, C. Smith, and J. M. Wallace. An intercomparison of methods for finding coupled patterns in climate data. *Journal of Climate*, 5(6):541–560, 1992.
- D. P. Chambers, B. D. Tapley, and R. H. Stewart. Anomalous warming in the indian ocean coincident with el nino. *Journal of Geophysical Research*, 104(C2):3035–3047, 1999.
- R. E. Davis. Predictability of sea surface temperature and sea level pressure anomalies over the north Pacific Ocean. *Journal of Physical Oceanography*, 6(3):249–266, 1976.
- A. Eigenheer and D. Quadfasel. Seasonal variability of the Bay of Bengal circulation inferred from TOPEX/Poseidon altimetry. *Journal of Geophysical Research*, 105(C2):3243–3252, 2000.

- A. A. Fernandes. *Study of the directional spectrum of ocean waves using array, buoy and radar measurements*. PhD thesis, Goa University, Taleigao Plateau, Goa, 1999.
- A. A. Fernandes, A. M. Almeida, R. Vaithiyanathan, and P. Vethamony. Determination of wave direction using an orbital following buoy. In S. K. Agadi, V. K. Das, and B. N. Dessai, editors, *Ocean technology perspectives*, pages 378–388. Publication and Information Directorate, New Delhi, 1994.
- A. A. Fernandes, P. Chandramohan, and B. U. Nayak. Observed currents at Bombay High during winter. *Mahasagar*, 26(2):95–104, 1993.
- A. A. Fernandes, V. K. Das, and N. Bahulayan. Harmonic tidal analysis at a few stations using the least squares method. *Mahasagar*, 24(1):1–12, 1991.
- A. A. Fernandes, V. K. Das, and R. Mahadevan. Storm surges along the east coast of India. Technical Report No. 4/83, National Institute of Oceanography, Dona Paula, Goa, 1983.
- A. A. Fernandes, A. D. Gouveia, and R. Nagarajan. Determination of wave direction from linear and polygonal arrays. *Ocean Engineering*, 15:345–357, 1988.
- A. A. Fernandes, A. D. Gouveia, P. V. Sathe, and R. Nagarajan. Wave observations in the Mahanadi Basin (Bay of Bengal) during september 1980. *Mahasagar – Bulletin of the National Institute of Oceanography, Dona Paula, Goa*, 14(4):239–249, 1981.
- A. A. Fernandes and S. M. Gupta. Statistical prediction of late miocene climate. Technical Report No. NIO/TR–5/94, National Institute of Oceanography, Dona Paula, Goa, 1994.
- A. A. Fernandes and R. Mahadevan. Factor analysis of multivariate data. Technical Report No. 2/82, National Institute of Oceanography, Dona Paula, Goa, 1982.
- A. A. Fernandes, H. B. Menon, Y. V. B. Sarma, P. D. Jog, and A. M. Almeida. Wavelength of ocean waves and surf beat at Duck from array measurements. In B. L. Edge and J. M. Hemsley, editors, *Ocean wave measurement and analysis*, volume 2, pages 1809–1818. American Society of Civil Engineers, 2002.
- A. A. Fernandes and S. Pednekar. Intercomparison between Aanderaa and Potok current meters. Technical Report No. NIO/TR–1/97, National Institute of Oceanography, Dona Paula, Goa, 1997.
- A. A. Fernandes, Y. V. B. Sarma, and H. B. Menon. Directional spectrum of ocean waves from array measurements using phase/time/path difference methods. *Ocean Engineering*, 27(4):345–363, 2000a.

- A. A. Fernandes, Y. V. B. Sarma, and H. B. Menon. Wave directional spectrum from SAR imagery. In *Proceedings Fifth Pacific Ocean Remote Sensing Conference (PORSEC)*, volume 1, pages 336–341, National Institute of Oceanography, Goa, India, 2000b.
- M. H. Glantz. *Currents of change: Impacts of El Nino and La Nina on climate and society*. Cambridge University Press, Cambridge, U.K., 2001.
- J. Gonella. A rotary component method for analysing meteorological and oceanographic vector time series. *Deep-Sea Research*, 19:833–846, 1972.
- A. Gouveia and R. Mahadevan. Spectral analysis of time series. Technical Report No. 5/83, National Institute of Oceanography, Dona Paula, Goa, 1983.
- S. M. Gupta and A. A. Fernandes. Supra-generic radiolarian transfer function based late Miocene paleomonsoonal changes in the northeastern tropical Indian Ocean. *Bulletin of the Indian Geologists' Association*, 28(1):29–51, 1995.
- S. M. Gupta and A. A. Fernandes. Quaternary radiolarian faunal changes in the tropical Indian Ocean: Inferences to paleomonsoonal oscillation of the 10°s hydrographic front. *Current Science*, 72(12):965–972, 1997.
- S. M. Gupta, A. A. Fernandes, and R. Mohan. Tropical sea surface temperatures and the earth's orbital eccentricity cycles. *Geophysical Research Letters*, 23(22):3159–3162, 1996.
- L. Guttman. Some necessary conditions for common-factor analysis. *Psychometrika*, 19:149–161, 1954.
- R. A. Hakstian, W. T. Rogers, and R. B. Cattell. The behaviour of number of factor rules with simulated data. *Multivariate Behaviour Research*, 17:193–219, 1982.
- J. D. Horel. Complex principal component analysis: Theory and examples. *Journal of Climate and Applied Meteorology*, 23(12):1660–1673, 1984.
- H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23:187–200, 1958.
- L. H. Kantha and C. A. Clayson. *Numerical models of oceans and oceanic processes*, volume 66 of *International Geophysical Series*. Academic Press, 2000.
- C. M. Kishtawal, S. Basu, and S. Karthikeyan. Retrieval of vertical wind profiles during monsoon from satellite observed winds over the Indian Ocean using complex EOF analysis. *Proc. Indian Academy of Sciences (Earth and Planetary Sciences)*, 110(1):77–86, 2001.

- J. E. Klován and J. Imbrie. An algorithm and FORTRAN–IV program for large scale Q–mode factor analysis and calculation of factor scores. *Mathematical Geology*, 3:61–67, 1971.
- M. R. R. Kumar and A. A. Fernandes. Variability of surface meteorological parameters over the Arabian Sea. *Indian Journal of Marine Sciences*, 23:8–13, 1994.
- P. K. Kundu and J. S. Allen. Some three-dimensional characteristics of low-frequency current fluctuations near the Oregon coast. *Journal of Physical Oceanography*, 6(2):181–199, 1976.
- J. E. Kutzbach. Empirical eigenvectors of sea–level pressure, surface temperature and precipitation complexes over North America. *Journal of Applied Meteorology*, 6(5):791–802, 1967.
- E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. Technical Report Scientific Report No. 1, Department of Meteorology, MIT, 1956.
- R. Mahadevan and A. A. Fernandes. Empirical orthogonal function description of beach profiles. In *Proceedings First National Conference on Dock and Harbour Engineering*, volume 2, pages E–107 to E–111. IIT, Bombay, 1985.
- R. Mahadevan, A. A. Fernandes, and S. W. A. Naqvi. Data assimilation techniques in modeling ocean processes. Technical Report No. NIO/TR–10/2001, National Institute of Oceanography, Dona Paula, Goa, 2001.
- A. Menezes and R. Mahadevan. Generalized inverse method of gravity interpretation. Technical Report No. 2/84, National Institute of Oceanography, Dona Paula, Goa, 1984.
- M. A. Merrifield and R. T. Guza. Detecting propagating signals with complex empirical orthogonal functions: A cautionary note. *Journal of Physical Oceanography*, 20:1628–1633, 1990.
- J. Michaelson. A statistical study of large-scale, long-period variability in North Pacific sea surface temperature anomalies. *Journal of Physical Oceanography*, 12:694–703, 1982.
- T. V. R. Murty and R. Mahadevan. The stochastic inverse method for ocean acoustic tomography studies. *Acoustic Letters*, 1995.
- J. Ortega. The givens-householder method for symmetric matrices. In A. Ralston and H. S. Wilf, editors, *Mathematical methods for digital computers*, volume II, pages 94–115. John Wiley and Sons, 1967.
- J. E. Overland and R. W. Preisendorfer. A significance test for principal components applied to a cyclone climatology. *Monthly Weather Review*, 110(1):1–4, 1982.

- C. Perigaud and P. Delecluse. Annual sea level variations in the southern tropical Indian Ocean from Geosat and shallow-water simulations. *Journal of Geophysical Research*, 97(C12):20169–20178, 1992.
- C. Perigaud and P. Delecluse. Interannual sea level variations in the tropical Indian Ocean from Geosat and shallow water simulations. *Journal of Physical Oceanography*, 23(9):1916–1934, 1993.
- J. Picaut, E. Hackert, A. J. Busalacchi, R. Murtugudde, and G. S. Lagerloef. Mechanisms of the 1997-1998 el nino-la nina, as inferred from space based observations. *Journal of Geophysical Research*, 107(C5), 2002.
- E. C. Pielou. *Mathematical Ecology*. John Wiley & Sons, 1977.
- J. T. Potemra and R. Lukas. Seasonal to interannual modes of sea level variability in the western Pacific and eastern Indian Ocean. *Geophysical Research Letters*, 26(3):365–368, 1999.
- R. W. Preisendorfer. *Principal component analysis in meteorology and oceanography*. Elsevier Science, 1988.
- R. W. Preisendorfer and T. P. Barnett. Significance tests for empirical orthogonal functions. In *Reprints Fifth Conference Probability and Statistics in Atmospheric Sciences*, pages 169–72, Las Vegas, 1977. American Meteorological Society.
- R. W. Preisendorfer, F. W. Zwiers, and T. P. Barnett. Foundations of principal component selection rules. Technical Report SIO Rep. 81-7, Scripps Institution of Oceanography, May 1981. 200 pp.
- S. A. Rao, S. K. Behera, Y. Masumoto, and T. Yamagata. Interannual subsurface variability in the tropical Indian Ocean with a special emphasis on the Indian Ocean dipole. *Deep-Sea Research, Part B*, 49:1549–1572, 2002.
- D. Reynolds. Impact of mount pinatubo aerosol on satellite-derived sea surface temperature. *Journal of Climate*, 6:768–774, 1993.
- D. Reynolds and T. Smith. Improved global sea surface temperature analyses using optimum interpolation. *Journal of Climate*, 7:929–948, 1994.
- N. H. Saji, B. N. Goswami, P. N. Vinayachandran, and T. Yamagata. A dipole in the tropical Indian Ocean. *Nature*, 401:360–363, 1999.
- D. Sengupta and R. Mahadevan. Bispectra of sea surface waves. Technical Report No. 1/83, National Institute of Oceanography, Dona Paula, Goa, 1983.

- S. A. Venegas, L. A. Mysak, and D. N. Straub. Atmosphere-ocean coupled variability in the South Atlantic. Technical report, Centre for Climate and Global Change Research, Montreal, Canada, 1996. Report 96-13.
- J. M. Wallace. Spectral studies of tropospheric wave disturbances in the tropical western Pacific. *Review of Geophysics and Space Physics*, 9(3):557–612, 1971.
- J. M. Wallace. Empirical orthogonal representation of time series in the frequency domain. part ii: Application to the study of tropical wave disturbances. *Journal of Applied Meteorology*, 11(6):893–900, 1972.
- J. M. Wallace and R. E. Dickinson. Empirical orthogonal representation of time series in the frequency domain. part i: Theoretical considerations. *Journal of Applied Meteorology*, 11(6):887–892, 1972.
- R. A. Walters and C. Heston. Removing tidal-period variations from time-series data using low-pass digital filters. *Journal of Physical Oceanography*, 12(1):112–115, 1982.
- D.-P. Wang and C. N. Mooers. Long coastal trapped waves off the west coast of the United States, Summer 1973. *Journal of Physical Oceanography*, 7:856–864, 1977.
- B. C. Weare and J. S. Nasstrom. Examples of extended empirical orthogonal function analyses. *Monthly Weather Review*, 110(6):481–485, 1982.
- P. J. Webster, A. M. Moore, J. P. Loschnigg, and R. R. Leben. Coupled ocean-atmosphere dynamics in the Indian Ocean during 1997-98. *Nature*, 401:356–360, 1999.
- W. B. White. Coupled Rossby waves in the Indian Ocean on interannual time scales. *Journal of Physical Oceanography*, 30(11):1245–1264, 2000a.
- W. B. White. Tropical coupled Rossby waves in the Pacific ocean-atmosphere system. *Journal of Physical Oceanography*, 30(6):1245–1264, 2000b.