```
      program ftrans
c program for calculation of the discrete Fourier spectrum.
c programmed 2002-03-10 by C.C.Tscherning. Last change: 2006-07-28.
c the program calculates the Fourier coefficients a(j) and b(j) and
c the amplitudes a(j)**2+b(j)**2 from a time-series f(x(i))
c with n entries (i=1,.. n) spaced equidistantly. The calculation is done
c by multiplication by cos(m x(i)) and sin(m x(i)) and summation.
c x = 2 pi/n * (i-0.5).
c The program may be run in different modes:
C mode = 0, a test-mode, where cos(m x) and sin(m x) are analyzed,
c           in which case ltest is true.
c mode = 1, one column in a data file is analyzed, and the
c           data where the largest coefficient has been removed is
c           output to a file ftransf.dat
c mode = 2, data on GRAVSOFT format are analyzed, and the file is
c           reproduced in ftransf.dat with a column with the data
c           where the largest coefficient has been removed is output.
c mode = 3, but a specified  wave-number is removed.
c mode = 4, the contribution from all  wave-numbers above a certain
C           threshold are removed.
c mode = 5, the function is split into two parts, one with the wave-numbers
c           above a certain limit and the other one below the limit.
c
c input:
C mode
c if mode > 0:  name of data file and output files for
c coefficients and filtered values.
c number of first record to be used (istart)
c number of data - n
c if mode=0, value of m. Otherwise
c input of the number of the column holding the data.
c if mode = 3, the wave-number
c if mode = 4, the threshold.
c if mode = 5, the limit (integer) .
c
c output:
c input variables are eccoed.
c if mode=0 the values of cos(m*i x) and sin(m*i x)
c are output to a file with the name ftrans.tda.
c list of  wave-number, coeff. value for cos, sin and amplitude**2.
c this list is also output to a file with the name ftrans.dat
c The numerically maximal coefficients are output.
c The data, minus the maximal ocillation are output to the file
c ftranf.dat for mode=1, 2 or 3.
C For mode=4, the difference with respect to all frequencies above
c the threshold is removed.
c for mode=5, the original value, the sum of the values below the
c input wave-number and the difference is output.
c the auto-covariance is output to acov.dat.
c
      implicit none
      integer maxdat,maxcol
      parameter (maxdat=5000000,maxcol=10)
      integer i,j,n,m,n2,npn,ncc,icmax,ismax,icsmax,
     *istart,mode,nlast,nfreq,id(maxdat),ntres,ilimit,imod,imod1
      real*8 d0,d1,d2,aj,bj,cj,dj,abj,tj,ti,rimod,rimod1,rnpn,
     *data(maxdat,maxcol),c(maxdat),s(maxdat),s0,s1,ss,r,ri,
     *pi,pi2,cmax,smax,csmax,ps,pc,pp,power,phase,phmax,
     *pfreq,phfre,thresh,ct(0:maxdat),st(0:maxdat),slim,sslim,
     *ddcos(0:2*maxdat),ddsin(0:2*maxdat),acov
      logical ltest,lall,lint,lf,lt
      character*144 ifile,cofffi,filtfi,fileco
      character*72 udate
c
      write(*,*)' DFT, ver. 2006-07-25 '
      call fdate(udate)
      write(*,*)udate
      write(*,*)' Computation of 1D Discrete Fourier Transform '
      write(*,*)' input mode (=0 test, 1 data column analyzed, '
```



```
     *' 2 GRAVSOFT file data analyzed,'
      write(*,*)' 3 specific freq. removed '
      write(*,*)' 4 contr. from more removed'
      write(*,*)
     *' 5 function split in two parts,'
      write(*,*)' above and below given wave-number '
      read(*,*)mode
      write(*,*)' mode = ', mode
      ltest=mode.eq.0
      if (.not.ltest) then
       write(*,*)' input name of data file '
       read(*,('a'))ifile
       write(*,*)ifile
       open(10,file=ifile)
       write(*,*)' Data input from ',ifile
       write(*,*)' Input name of file for Fourier coefficients '
       read(*,('a'))cofffi
       write(*,*)cofffi
c file for Fourier-coefficients.
       open(12,file=cofffi)
c
       write(*,*)' input name of file to hold filtered values '
       read(*,('a'))filtfi
       write(*,*)filtfi
c file for filtered data on gravsoft format.
       open(13,file=filtfi)
c file for auto-covariance function.
       read(*,('a'))fileco
       write(*,*)fileco
       open(14,file=fileco)
c
       if (mode.eq.5) then
        write(*,*)' input wave-number limit '
        read(*,*),ilimit
        write(*,*)ilimit
       end if
      else
c file for test-data output.
       open(11,file='ftrans.tda')
       open(12,file='ftrans.dat')
      end if
      lt=.true.
      lf=.false.
      lall=lf
c
      ntres=0
      d0=0.0d0
      d1=1.0d0
      d2=2.0d0
      pi=4.0d0*atan(d1)
      cmax=d0
      smax=d0
      csmax=d0
      s0=d0
      ss=d0
      slim=d0
      sslim=d0
      pi2=pi*d2
      write(*,*)' input number of first data record '
      read(*,*)istart
      write(*,*)' first record is ',istart
      write(*,*)' input number of data '
      read(*,*)n
      write(*,*)' number of data input ',n
      n2=n/2
      npn=n*2
      rnpn=npn*d1
      if (n.gt.maxdat) then
       n=maxdat
```

```
       write(*,*)' WARNING: maximum number of data is ',maxdat
      end if
      if (ltest) then
       write(*,*)' input wave-number m '
       read(*,*)m
      else
       write(*,*)' input column number '
       read(*,*)ncc
c
       if (mode.ge.2) then
        write(*,*)' input number of columns (incl. lat., lon., h) '
        read(*,*)nlast
        write(*,*)' is first column in record an integer ? (t/f) '
        read(*,*)lint
c
        lall=.true.
c       write(*,*)' file ftransf.dat include  only input column '
        thresh=d0
        if (mode.eq.3) then
         write(*,*)' input wave-number '
         read(*,*)nfreq
         write(*,*)nfreq
        else
         if (mode.eq.4) then
          write(*,*)' input threshold '
          read(*,*)thresh
          write(*,*)thresh
          do i=1,n
           ct(i)=d0
           st(i)=d0
          end do
         end if
        end if
       else
        nlast=ncc
       end if
c
       write(*,*)' column used',ncc,', last column ',nlast
       if (ncc.gt.maxcol) then
        write(*,*)' ncc too large, ncc:= ',ncc
        stop
       end if
c
      end if
c
c creating a table of cosines and sines.
      if (ltest) write(*,*)' table of cos and sin '
      do i=0,npn
       ddcos(i)=cos(i*0.5d0*pi2/n)
       ddsin(i)=sin(i*0.5d0*pi2/n)
       if (ltest) then
        write(*,*)i,ddcos(i),ddsin(i)
       end if
      end do
c
      if (.not.ltest.and.istart.gt.1) then
      do i=1,istart-1
       if (lint) then
        read(10,*)id(i),(data(i,j),j=2,nlast)
       else
        read(10,*)(data(i,j),j=1,nlast)
       end if
      end do
      end if
c
      do i=1,n
       if (ltest) then
        imod=mod((2*i-1)*m,npn)
        tj=cos(m*pi2/n*(i+istart-0.5d0))
```



```
        ti=sin(m*pi2/n*(i+istart-0.5d0))
        if (abs(tj-ddcos(imod).gt.1.0d-15.or.abs(ti-ddsin(imod))
     *  .gt.1.0d-15)) then
         write(*,*)' test ',m,i,tj,ti,ddcos(imod),ddsin(imod),imod
        end if
        c(i)=tj
        s(i)=ti
        write(11,110)c(i),s(i)
  110   format(2f12.8)
       else
        if (lint) then
         read(10,*)id(i),(data(i,j),j=2,nlast)
        else
         read(10,*)(data(i,j),j=1,nlast)
        end if
        c(i)=data(i,ncc)
        s0=s0+c(i)
        ss=ss+c(i)**2
       end if
      end do
c
      if (.not.ltest) then
       s1=sqrt((ss-s0**2/n)/(n-1))
       write(*,105)s0/n,s1,ss/n
  105  format(/,' Mean,standard dev. variance: ',/,3d16.6)
      end if
c
      power=d0
      if (.not.ltest) then
      write(*,*)' First 1000 values: '
      write(*,*)' Frequency j, a(j), b(j) and a(j)**2+b(j)**2'
      else
       write(*,*)
     *' wave-number, coeff from cos and from sin and phase from cos '
      end if
      if (mode.eq.5) n2=ilimit
c for mode=5 we only need the first coefficients.
      do j=0,n2
       aj=d0
       bj=d0
       cj=d0
       dj=d0
       do i=1,n
c use of tabulated cos and sin introduced 2006-07-17.
c       imod1=mod((2*i-1)*j,npn)
        rimod=(d2*i-d1)*j
        imod=rimod/npn
        rimod1=imod*rnpn
        if (abs(rimod).lt.1.0d-9) then
         imod=0
        else
         imod=rimod-rimod1
        end if
c       if (imod.ne.imod1) write(*,*)' imod ',imod,imod1,rimod
        if (imod.lt.0.or.imod.gt.npn) then
         write(*,*)' imod ',imod,i,j,(2*i-1)*j
         stop
        end if
        aj=aj+c(i)*ddcos(imod)*d2/n
        bj=bj+c(i)*ddsin(imod)*d2/n
c       aj=aj+c(i)*cos(j*pi2/n*(i-0.5d0))*d2/n
c       bj=bj+c(i)*sin(j*pi2/n*(i-0.5d0))*d2/n
        if (ltest) then
         cj=cj+s(i)*cos(j*pi2/n*(i-0.5d0))*d2/n
         dj=dj+s(i)*sin(j*pi2/n*(i-0.5d0))*d2/n
        end if
       end do
       if (j.eq.0) aj=aj/d2
        phase=atan2(-bj,aj)*180.0/pi
```

```
      if (ltest) then
       write(*,100)j,aj,bj,cj,dj,phase
       write(12,100)j,aj,bj,cj,dj,phase
 100   format(i5,4f9.5,f9.3)
 101   format(i5,4d16.5,f9.3)
      else
       abj=aj**2+bj**2
       if (abs(aj).lt.1.0d3.and.abs(bj).lt.1.0d3.and.
     *  abj.lt.1.0d3) then
        if (j.le.1000) write(*,102)j,aj,bj,abj,phase,ntres
 102    format(i5,3f9.4,f9.3,i8)
 103    format(i5,3d16.5,f9.3,i8)
       else
        if (j.le.1000) write(*,103)j,aj,bj,abj,phase,ntres
       end if
       power=power+abj
       write(12,103)j,aj,bj,aj**2+bj**2
       if (mode.ge.4) then
        if (abs(aj).gt.thresh) then
         ct(j)=aj
         if (mode.eq.4) ntres=ntres+1
        end if
        if (abs(bj).gt.thresh) then
         st(j)=bj
         if (mode.eq.4) ntres=ntres+1
        end if
       end if
       if (abs(aj).gt.cmax) then
        cmax=abs(aj)
        icmax=j
       end if
       if (abs(bj).gt.smax) then
        smax=abs(bj)
        ismax=j
       end if
       if ((aj**2+bj**2).gt.csmax) then
        phmax=phase
        csmax=aj**2+bj**2
        icsmax=j
       end if
       if (mode.eq.3.and.j.eq.nfreq) then
        pfreq=sqrt(abj)
        phfre=phase
       end if
      end if
     end do
     if (mode.eq.4) write(*,*)' ntres = ',ntres
c
     if (.not.ltest)  then
      close(10)
      if (icmax.gt.0) then
       pc=n/icmax
      else
       pc=n
      end if
      if (ismax.gt.0) then
       ps=n/ismax
      else
       ps=n
      end if
      if (icsmax.gt.0) then
       pp=n/icsmax
      else
       pp=n
      end if
      write(*,120)icmax,ismax,icsmax,cmax,smax,csmax,
     * pc,ps,pp,phmax,power,sqrt(power)
 120  format(/,' Maximum coefficents found at ',/
     * '        cos          sin       amplitude',/
```



```
     * ' Wave-number',3(4x,i10),/,
     * 9x,3f14.4,/,' Periods:',3f14.4,' Phase ',f10.3,
     * /,' Total Power = ',f12.4,', sqrt(Power)= ',f12.4)
       if (mode.eq.3) then
        write(*,132)nfreq,pfreq,phfre
 132    format(' Wave-number = ',i4,', amplitude and phase = ',2d16.5)
        write(*,*)
     *  ' Output of data with largest wave removed to ftransf.dat '
        icsmax=nfreq
        csmax=pfreq
        phmax=phfre
       else
        if (mode.eq.4) then
         write(*,*)' all wave-numbers above threshold removed '
        else
         if (mode.ne.5) then
         write(*,*)
     *  ' Output of data with largest coeff. removed to ftransf.dat '
         else
          write(*,*)' output  with contribution from first coeff. '
         end if
        end if
       end if
c
       write(*,*)
     * ' Format: number, original values, correction, filtered value.'
       s1=d0
       ss=d0
       csmax=sqrt(csmax)
       write(*,117)csmax
 117   format(' sqrt of max. amplitude ',d16.5)
       if (mode.eq.5) then
        n2=ilimit
        write(*,*)' Output-sequence: #,lat,lon,h,input,below,above '
       end if
       do i=1,n
        acov=d0
        if (mode.ge.4) then
c summation of series with coefficients larger than threshold.
c or for mode=5, contribution up to nlimit.
         ri=d0
         do j=0,n2
c         imod=mod((2*i-1)*j,npn)
          rimod=(d2*i-d1)*j
          imod=rimod/npn
          rimod1=imod*rnpn
          if (abs(rimod).lt.1.0d-9) then
           imod=0
          else
           imod=rimod-rimod1
          end if
c         pp=pi2*(i-0.5)*j/n
c         ri=ri+ct(j)*cos(pp)+st(j)*sin(pp)
          ri=ri+ct(j)*ddcos(imod)+st(j)*ddsin(imod)
c         if (abs(ddcos(imod)-cos(pp)).gt.1.0d-12.or.abs(ddsin(imod)
c    *    -sin(pp)).gt.1.0d-12.or.lf) then
c          write(*,*)' error ',i,j,imod,ddcos(imod),cos(pp)
c         end if
          acov=acov+(ct(j)**2+st(j)**2)*ddcos(i)
         end do
         if (mod(i,50000).eq.0) write(*,129)i,ri,c(i),c(i)-ri
 129     format(I9,3f12.5)
        else
         pp=d1*(i-0.5d0)*icsmax/n
         ri=csmax*cos(pp*pi2+phmax*pi/180.0)
        end if
        r=c(i)-ri
        if (lall) then
c output as id., latitude, longitude, height, data...
```

```
         if (lint) then
          if (mode.lt.5) then
           write(13,128)id(i),(data(i,j),j=2,4),data(i,ncc),r
 128       format(i10,f10.5,f11.5,f10.2,10d16.8)
          else
            write(13,128)id(i),(data(i,j),j=2,4),data(i,ncc),r,ri
          end if
         else
          if (mode.lt.5) then
           write(13,127)(data(i,j),j=1,4),data(i,ncc),r
 127       format(f9.1,f10.5,f11.5,f10.2,10d16.8)
          else
           write(13,127)(data(i,j),j=1,4),data(i,ncc),r,ri
          end if
         end if
        else
         write(13,121)i,c(i),ri,r,pp
 121     format(i7,4d16.8)
        end if
        write(14,158)i,acov
 158    format(i8,d16.6)
        s1=s1+r
        ss=ss+r**2
       end do
       pc=sqrt((ss-s1**2/n)/(n-1))
       write(*,116)s1/n,ss/n,pc
 116   format(' Mean, var. and stdv after low-pass filtering ',/
     * 3d16.5)
      end if
c
      close(12)
      close(11)
      close(13)
      close(14)
      call fdate(udate)
      write(*,*)udate
      stop
      end
```