```
      programme to calculate least square method
      parameter (npt=30000)
      real*8 i
      real ain,bsl,sumt,sumx,sumtx,meanx,sumqt,pi,sumx1,sumtx1



      dimension x(npt),x1(npt),x2(npt),x3(npt),x4(npt)
       sumt=0
       sumx=0
       sumtx=0
       sumqt=0
       sumx1=0.0
       sumtx1=0.0
       pi=3.1416

      open(unit=10,file='tstomls.dat',status='old')
      open(unit=11,file='tstomlscs.out')


      do i=1.0D00,npt
       read(10,*) isec,x(i)
       sumt=sumt+i
       sumx=sumx+x(i)
       sumtx=sumtx+(i*x(i))
       sumqt=sumqt+(i*i)
      end do
       meanx=sumx/npt

        do i=1.0D00,npt
         x1(i)=(x(i)-meanx)
         sumx1=sumx1+x1(i)
         sumtx1=sumtx1+(i*x1(i))
        end do
        do i=1.0D00,npt
         ain=((sumx1*sumqt)-(sumt*sumtx1))/((npt*sumqt)-(sumt*sumt))
         bsl=((npt*sumtx1)-(sumx1*sumt))/((npt*sumqt)-(sumt*sumt))
         x2(i)=ain+(bsl*i)
         x3(i)=x1(i)-x2(i)
         x4(i)=x3(i)*(1+cos(pi*(i/npt)*1.0))*0.5

      write (11,20) i,x(i),x1(i),x2(i),x3(i),x4(i)
      end do

0     format (6f12.6)

      close(10)

    stop
    end
```

```
% Program P1_5
% Signal smoothing by Averaging
clf;
R=51;
d=0.8*(rand(R,1)-0.5);% Generate random noise
m= 0:R-1;
s=2*m.*(0.9.^m);% generate uncorrupted signal
x= s+d';% Generate noise corrupted signal
subplot(2,1,1);
plot(m,d','r-',m,s,'g--',m,x,'b-.');
xlabel('Time index n');ylabel('Amplitude');
legend('d[n]','s[n]','x[n]');
x1=[0 0 x]; x2=[0 x 0]; x3=[x 0 0];
y= (x1 + x2 + x3)/3;
subplot(2,1,2);
plot(m,y(2:R+1),'r-',m,x,'g--');
legend('y[n]','x[n]');
xlabel('Time index n');ylabel('Amplitude');
```

```
% Program P2_1
% Simulation of an M-point Moving Average Filter
% Generate the input signal
clf;
n=0:100;
s1=cos(2*pi*0.05*n);% A low- frequency sinusoid
s2=cos(2*pi*0.47*n);% A high frequency sinusoid
x= s1+s2;
% Implementation of the moving average filter
M= input('Desired length of the filter=');
num= ones(1,M);
y= filter(num,1,x)/M;
% Display the input and output signals
subplot(2,2,1);
plot(n, s1);
axis([0, 100, -2, 2]);
xlabel('Time index n');ylabel('Amplitude');
title('Signal # 1');
subplot(2,2,2);
plot(n, s2);
axis([0,100, -2, 2]);
xlabel('Time index n');ylabel('Amplitude');
title('Signal # 2');
subplot(2,2,3);
plot(n,x);
axis([0, 100, -2, 2]);
xlabel('Time index n');ylabel('Amplitude');
title('Input signal');
subplot(2,2,4);
plot(n,y);
axis([0, 100, -2, 2]);
xlabel('Time index n');ylabel('Amplitude');
title('Output Signal');
axis;
```

to remove meantend

```
programme ~~read and write file~~
parameter (npt=30000)
real sumx,sumy,sumz,meanx,meany,meanz

dimension x(npt),y(npt),z(npt)
 sumx=0
 sumy=0
 sumz=0

open(unit=10,file='tst.dat',status='old')
open(unit=11,file='tstmr.dat')

 write(11,5)
 format('Sec',10x,'X',8x,'Y',9x,'Z')

to calculate the mean

 do i=1,npt
  read(10,*) isec,x(i),y(i),z(i)
  sumx=sumx+x(i)
  sumy=sumy+y(i)
  sumz=sumz+z(i)
  end do

  meanx=sumx/npt
  meany=sumy/npt
  meanz=sumz/npt
  print*,'mean of x',meanx
print*,'mean of y',meany
 print*,'mean of z',meanz

calculate the mean trend

 do i=1,30000
 write (11,20) i,(x(i)-meanx),(y(i)-meany),(z(i)-meanz)
 end do

 format (i5,3f10.2)
 close(10)

stop
end
```

Scanned with OKEN Scanner

```
  programme for trend removal
parameter (npt=30000)
   real*8 i

 dimension x(npt),y(npt),z(npt)
 dimension x1(npt),y1(npt),z1(npt)


    x1(0)=0.0
    y1(0)=0.0
    z1(0)=0.0


  open(unit=10,file='tstmr.dat',status='old')
  open(unit=11,file='tstmrtr.dat')


     do i=1.0D00,npt
     read(10,*) isec,x(i),y(i),z(i)
     end do




     do i=1.0D00,npt

           i * (x(i) - x(1)) / npT     x(i)-
     x1(i)=(x(i)-((i/npt)*(x(npt)-x(1))))
     y1(i)=(y(i)-((i/npt)*(y(npt)-y(1))))
     z1(i)=(z(i)-((i/npt)*(z(npt)-z(1))))

      print *,x1(i)

    write (11,20) i,x1(i),y1(i),z1(i)

  format (4f10.4)

  end do
  close(10)


  stop
  end
```

```
programme to apply cosine bell to data

parameter (npt=30000)
real*8 i
real pi
dimension x(npt),y(npt),z(npt),sec(npt)
dimension x1(npt),y1(npt),z1(npt)


 pi=3.1416


open(unit=10,file='tstmrtr.dat',status='old')
open(unit=11,file='tstmrtrcos.dat')


   do i=1.0D00,npt
   read(10,*) sec(i),x(i),y(i),z(i)
   end do


 do i=1.0D00,npt
   x1(i)=x(i)*(0.5*(1+cos(pi*(i/(npt)))))
   y1(i)=y(i)*(0.5*(1+cos(pi*(i/(npt)))))
   z1(i)=z(i)*(0.5*(1+cos(pi*(i/(npt)))))

 write (11,20) i,x1(i),y1(i),z1(i)
 end do


 format (4f10.4)


 close(10)

 stop
 end
```

(1+ cos (pi x

```
c       programme to normalize the data

        parameter (npt=30000)
        real a
        dimension x(npt),x1(npt)

        open(unit=10,file='tst.dat',status='old')
        open(unit=11,file='tst.out')

        do i=1,npt

        read (10,*) isec,x(i)

        end do

        a=x(npt)-x(1)

        do i=1,npt

        x1(i)=(x(i)-(a*(i-1)))/((npt-1)*1.0)

        write (11,20) i,x1(i)
        end do

 20        format (i5,1f10.4)

        close(10)

        stop
        end
```

```
format long
s=load('sine.dat');
t=s(:,1);
x=s(:,2);
%y=s(:,3);
%z=s(:,4);
%x1=detrend(x);
%y1=detrend(y);
%z1=detrend(z);
f1=fft(x,1024);
%f2=fft(y,30000);
%f3=fft(z,30000);
amp1=abs(f1);
%amp2=abs(f2);
%amp3=abs(f3);
f1 = 1*(0:512)/1024;
%f2 = 1*(0:15000)/30000;
%f3 = 1*(0:15000)/30000;
plot(f1,amp1(1:513))
%,f2,amp2(1:15001),f3,amp3(1:15001));
grid on
title('Frequency content of x')
%,y,z')
xlabel('Frequency(Hz)')
ylabel('Amplitude')
```

```
format long
s=load('gsh01xyz.out');
t=s(:,1);
x=s(:,2);
y=s(:,3);
z=s(:,4);
x1=detrend(x);
y1=detrend(y);
z1=detrend(z);
f=fft(x,30000);
f1=fft(x1,30000);
amp = abs(f);
amp1=abs(f1)
f = 1*(0:15000)/30000;
f1 = 1*(0:15000)/30000;
plot(f,amp(1:15001),f1,amp1(1:15001))
grid on
title('Frequency content of x')
xlabel('frequency (Hz)')
ylabel('Amplitude')
```

```
c      programme to apply tukeywin window to a data

      parameter (npt=30000)
      real*8 i
      real pi
        dimension x(npt),x1(npt),x2(npt),x3(npt),x4(npt)

       pi=3.1416


      open(unit=10,file='tst1.dat',status='old')
      open(unit=11,file='tst5.out')


        do i=1.0D00,npt
        read(10,*) isec,x(i)
        end do


      do i=1.0D00,npt
        x1(i)=x(i)*(1+cos(pi*(i/npt)*1.0))*0.5
        x2(i)=x(i)*(0.5*(1+cos(4*pi*((i-1)/(npt-1))-pi)))
        x3(i)=x(i)*(0.5*(1+cos(2*pi*((i-1)/(npt-1))-pi)))
        x4(i)=x(i)*(0.5*(1-cos(2*pi*(i/(npt-1)))))

      write (11,20) i,x(i),x1(i),x2(i),x3(i),x4(i)
      end do


20    format (6f10.4)


      close(10)

      stop
      end
```

```
c       programme to applz sliding window and linear trend

      parameter (npt=30000)
      parameter (nt=4900)


      real*8i
      real pi

      dimension z(npt),z1(npt),z2(npt),z3(npt),z4(npt),z5(npt),z6(npt),
     *x(npt),y(npt)



      pi=3.1416

      open(unit=10,file='sastr.dat',status='old')
      open(unit=11,file='hn1-49.dat')
      open(unit=12,file='hn50-99.dat')
      open(unit=13,file='hn101-149.dat')
      open(unit=14,file='hn151-199.dat')
      open(unit=15,file='hn201-249.dat')
      open(unit=16,file='hn251-299.dat')

      do i=1.0D00,npt
      read(10,*) isec,x(i),y(i),z(i)
      end do

      do i=1.0D00,4900

      z1(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))
      write (11,20) i,z(i),z1(i)
      end do

      do i=5001.0D00,9900

      z2(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))
      write (12,20) i,z(i),z2(i)
      end do

      do i=10001.0D00,14900

      z3(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))
      write (13,20) i,z(i),z3(i)
      end do

      do i=15001.0D00,19900

      z4(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))
      write (14,20) i,z(i),z4(i)
      end do

      do i=20001.0D00,24900

      z5(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))
      write (15,20) i,z(i),z5(i)
      end do

      do i=25001.0D00,29900

      z6(i)=z(i)*(0.5*(1-cos(2*pi*(i/(nt-1)))))

      write (16,20) i,z(i),z6(i)
      end do

20    format (3f10.4)


      close(10)

      stop
      end
```

```
% Program P10_9
% Design of Lth Band FIR Filter using the
% window fourier series approach
clf;
k=11;
n=-k:k;
%generate truncated impuls response of
% the tidal lowpass filter
b = sinc(n/2)/2;
%generate the window sequence
win=hamming(23)
% generate coefficient of the window filter
fil = b.*win';
c = fil/sum(fil);
% plot the gain response of the window filter
[h,w] = freqz(c,1,256);
g = 20*log10(abs(h));
%plot(fil,n)
plot(w/pi,g);axis([0 1 -90 10]); grid
xlabel('\omega/\pi');ylabel('gain,dB');
```

```
c       programe to write low pass windowed- sinc filter

        parameter(npt=30000)
        real*8 i
        real pi,fc,m,sum

        dimension x(npt),x1(npt),h(500)

        sum=0

        pi=3.1416,fc=0.14,m=500

        open(unit=10,file='tst.dat',status='old')
        open(unit=11,file='tsthan.dat')


        do i=1.0D00,npt
        read(10,*) sec(i),x(i)
        end do

        do i=1,500

        if (i-m/2.eq.0.0) then
        h(i)=2*pi*fc

        if (i-m/2.lt.0.0) then
        h(i)= sin(2*pi*fc*(i-m/2))/(i-m/2)
        h(i)= h(i)*(0.5-0.5*cos(2*pi*i)/m)

        if (i-m/2.gt.0.0) then
        h(i)= sin(2*pi*fc*(i-m/2))/(i-m/2)
        h(i)= h(i)*(0.5-0.5*cos(2*pi*i)/m)

        end if

        end do
```

```
%programe of p7_1
%Design of a Butterworth Bandstop Digital filter
Ws=[0.4 0.6]; Wp=[0.3 0.7]; Rp=0.4; Rs=50;
%Estimate the Filter Order
[N1,Wn1]=buttord(Wp,Ws,Rp,Rs);
% Design the filter
[num,den]=butter(N1,Wn1,'stop');
%Display the transfer function
disp('Numerator coefficient are');disp(num);
disp('Denominator coefficient are');disp(den);
%Compute the gain resopnse
[g,w]= gain(num,den);
%plot the gain response
plot(w/pi,g);grid
axis([0 1 -60 5]);
xlabel('\omega/\pi');
ylabel('gain,db');
title('Gain Response of a Butterworth Bandstop Filter');
```

```
% Program P10_9
% Design of Lth Band FIR Filter using the
% window fourier series approach
clf;
k=11;
n=-k:k;
%generate truncated impuls response of
% the tidal lowpass filter
b = sinc(n/2)/2;
%generate the window sequence
wvtool(hamming(n))
% generate coefficient of the window filter
fil = b.*win';
c = fil/sum(fil);
% plot the gain response of the window filter
[h,w] = freqz(c,1,256);
g = 20*log10(abs(h));
%plot(fil,n)
plot(w/pi,g);axis([0 1 -90 10]); grid
xlabel('\omega/\pi');ylabel('gain,dB');
```