Chapter 15. Frequency analysis

15.1 Introduction

According to frequency analysis, any physical waveform can be decomposed into a number of discrete frequencies, or a spectrum of frequencies over a continuous range. The study of the way general functions may be represented or approximated by sums of simpler trigonometric functions is known in mathematics as Fourier analysis , named after Joseph Fourier (1768-1830), who showed that representing a function as a sum of trigonometric functions greatly simplifies the study of heat transfer.

In sciences and engineering, the process of decomposing a function into oscillatory components (Fourier analysis or frequency analysis or frequency deconvolution)) is often coupled with the operation of rebuilding the function from its components, known as Fourier synthesis. For example, determining what component frequencies are present in a musical note would involve computing the Fourier transform of a sampled musical note. One could then re-synthesize the same sound by including the frequency components as revealed in the Fourier analysis. In mathematics, the term Fourier analysis often refers to the study of both operations.

The time-evolution of a physical process often contains essential information about the nature of its driving force(s). This chapter describes the results obtained by applying the Fourier analysis (or deconvolution) to temperature records, mainly obtained by satellite data, in the last decades. The records may refer to lower troposphere, lower stratosphere, sea surface temperatures and others, and can be relative to selected areas or to the whole earth. Methods and scripts described in here can be applied to a much wider collection of physical processes, provided a complete dataset for them is available as a function of time, space or other variables.

15.2 The Fourier procedure based on a least square refinement

Let us start with an example of deconvolution from a series of data, generated by two sinewave functions with different periods (3, 20 years) in a certain interval, say 0 to 100 years, step 0.5. To the data a random noise is added. The lines are evidenced in orange .

Once the raw data are prepared, the first step in the analysis is to fit the most suitable sinewave function, in a certain range of low frequency, in this case from 10 to 31.6 years
This is done with a *for...endfor* cycle which recursively call the function *best()*. This function contains the powerful *ols* (ordinary least square estimation) function.
Once the best fit is found, the results of the search are displayed in a proper grapic output.
The low frequency deconvolution is then subtracted to the raw data and the resulting data are then analyzed for an higher frequency, in this case from 0.4 to 10 years. Again the results of the search are displayed in a graphic output. Finally, in a third graphic figure, the raw data and the sum of the two deconvoluted sinewaves, are plotted together

```
clear all;clc;format short;format compact;
global x y xd Ud;
function sd = best(p)       % ------ minimizer according to least square summation
  global x y xd Ud;
  Ud(:,1) = sin(2*pi/p*x);    % sin
  Ud(:,2) = cos(2*pi/p*x);    % cos
  [xd,sd,rd] = ols(y,Ud);     % ols = Ordinary Least Square refinement
endfunction                   % ------ end of minimizer

% generate some data in a 100 years period, 2 data available per year
x = linspace(0,100,201);  % a row vector is generated
x = x';                   % usually a column vector is preferred
p1 = 3;p2 = 20;           % these are the 2 periods in year
y = sin(2*pi/p1*x) + 0.5*cos(2*pi/p2*x); % y vector is obtained
g = randn(21); s = g(1:201);s = 0.3*s';  % random noise is added to it
y = y + s;
% find and detract low frequency component (search between 10 and 31.6 years)
k = 0;
for i = logspace(1,1.5,200);  % frequency is spanned by a logarithmic increment
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);lF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
xlabel ('Period , years');ylabel ('R-squared');title ('Low Frequency
Analysis');axis([10,32]);
x2 = best(lF);
y2 = y;
y = y - Ud*xd(:,1);
xd2(:,1) = xd(:,1);

% find high frequency component (search between 0.4 and 10 years)
figure;
k = 0;
for i = logspace(-0.4,1,200);  % frequency is spanned by a logarithmic increment
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);hF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
xlabel ('Period , years');ylabel ('R-squared');title ('High Frequency
Analysis');axis([0,10]);
x2 = best(hF);
figure;

% 3) final graphics
yN = xd2(1)*sin(2*pi/lF*x) + xd2(2)*cos(2*pi/lF*x) + xd(1)*sin(2*pi/hF*x) +
xd(2)*cos(2*pi/hF*x);
plot (x,y2,'b','linestyle',':','linewidth',2,x,yN,'r','linewidth',2);grid on;grid
minor on;axis([0 100 -2 2]);hold on;
```

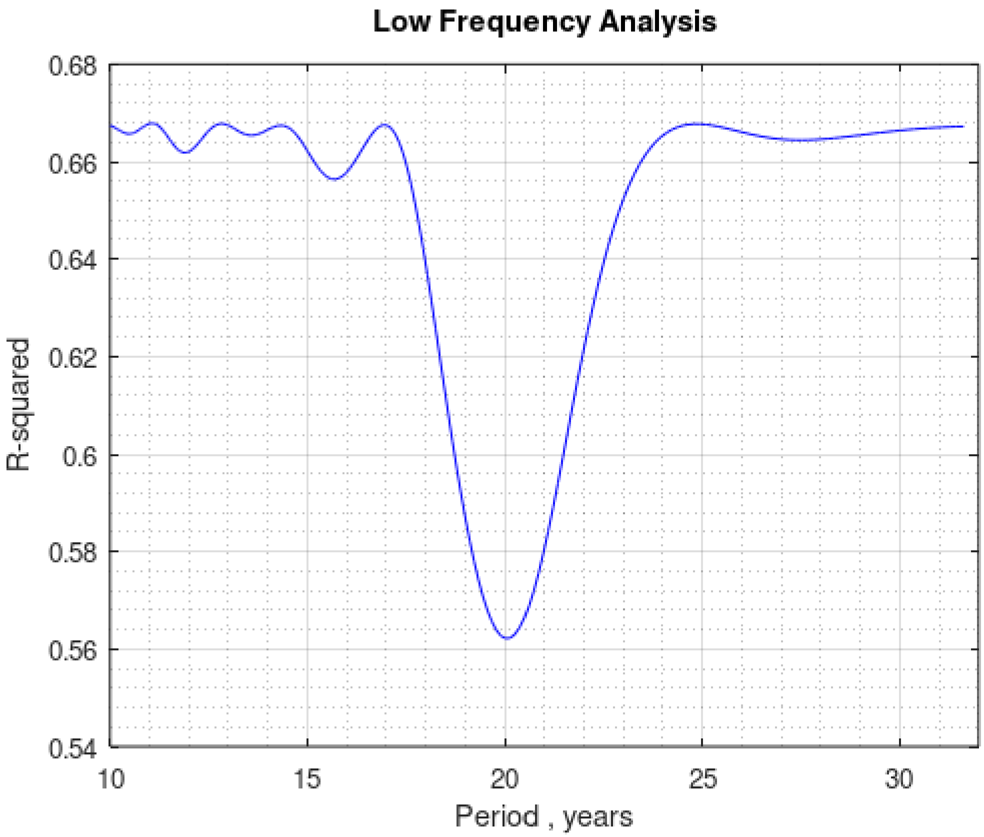


fig. 15.1

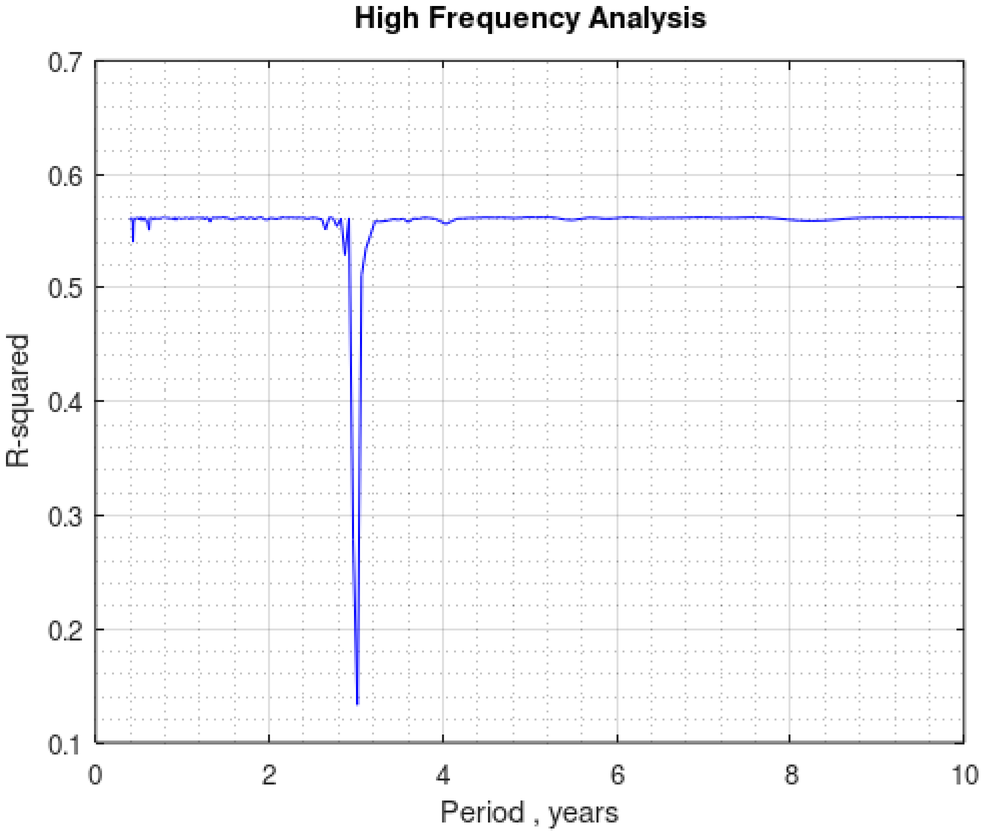


fig. 15.2

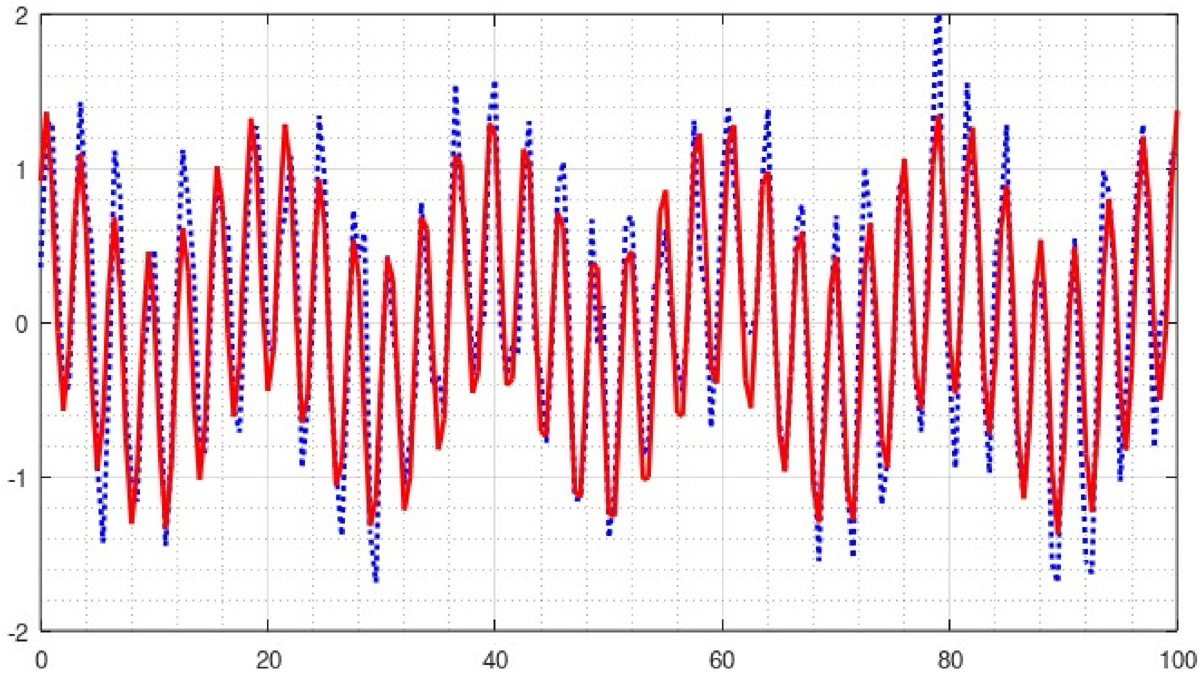


fig. 15.3 Raw data (blue dotted line) and deconvoluted two frequencies summation

15.3 The collection of data from web, a simple example

Octave offers two different function to download data from a web source, namely *urlwrite* and *webread* . Using *urlwrite* two strings are needed, as in the following example. The first is the *url address* of the text file do be downloaded (in this case the lower-troposphere temperature data), the second being the file name to be assigned to that file, when downloaded (if not otherwise stated, in the current directory)

```
f=urlwrite('http://vortex.nsstc.uah.edu/data/msu/v6.0/tlt/
uahncdc_lt_6.0.txt','uahncdc_lt_6.0.txt');
```

Each time the program is executed, the download will be activated , so (unless new data are available) the line can be temporarly suppressed by a comment sign (%) .
As an example for the data download from the web and subsequent analysis according to a low frequency and an high frequency, we examine the UAH (University of Alabama in Huntsville) repository. Global monthly average temperature are reported on one of the most visited climate-site, the University of Alabama at Huntsville (UAH), USA. Other authoritative web-based sources for global temperature records are RSS(Remote Sensing System) and HadCRUT. The data are obtained by the National Oceanographic and Atmospheric Administration (NOAA) TIROS-N satellite, interpreted by Dr. Roy Spencer and Dr. John Christy, both at Global Hydrology and Climate Center of the same University.

Records start from december,1978 and are regularly updated. They refer to four layers of the planet’s atmosphere:

1. lower troposphere
2. middle troposphere
3. tropopause
4. lower stratosphere

For each of the four regions of the atmosphere 24 regions of the globe are examined, namely :

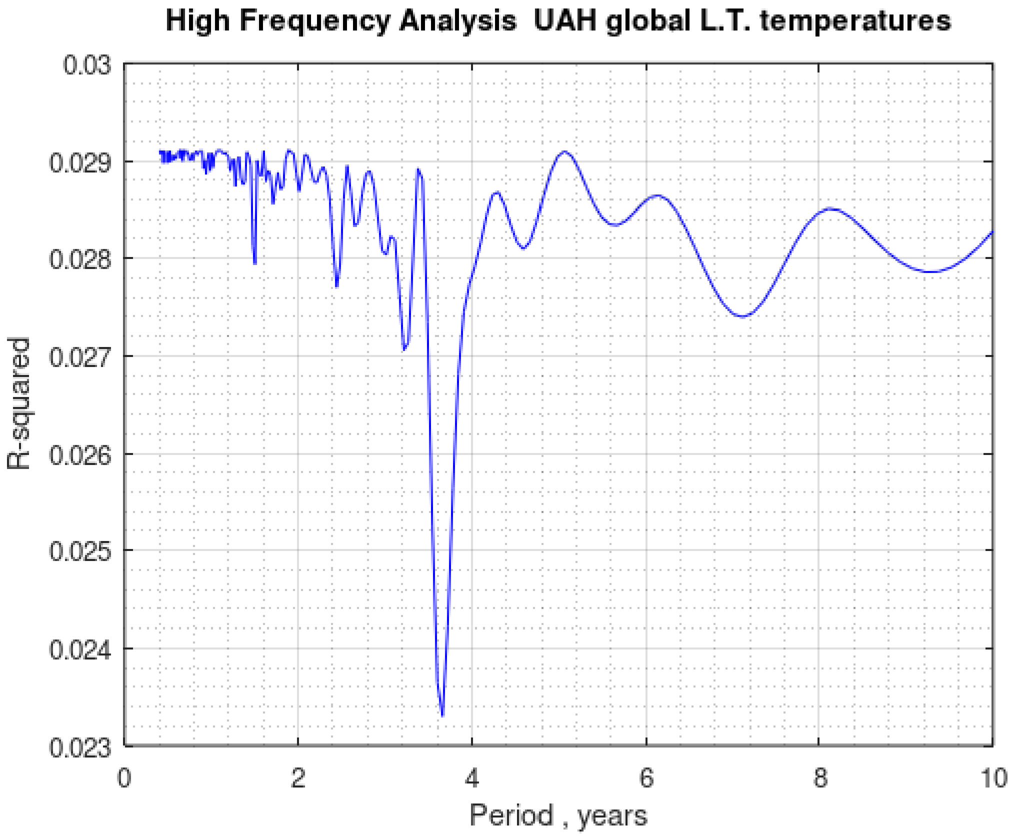
3 = Globe	*4 = Land*	*5 = Ocean*	*whole globe 90S-90N*
6 = Globe	*7 = Land*	*8 = Ocean*	*northern hemisphere 0-90N*
9 = Globe	*10 = Land*	*11 = Ocean*	*southern hemisphere 90S-0*
12 = Globe	*13 = Land*	*14 = Ocean*	*tropics 20S-20N*
15 = Globe	*16 = Land*	*17 = Ocean*	*northern extention 20N-90N*
18 = Globe	*19 = Land*	*20 = Ocean*	*southern extention 20N-90N*
21 = Globe	*22 = Land*	*23 = Ocean*	*north pole 60N-90N*
24 = Globe	*25 = Land*	*26 = Ocean*	*south pole 90S-60N*

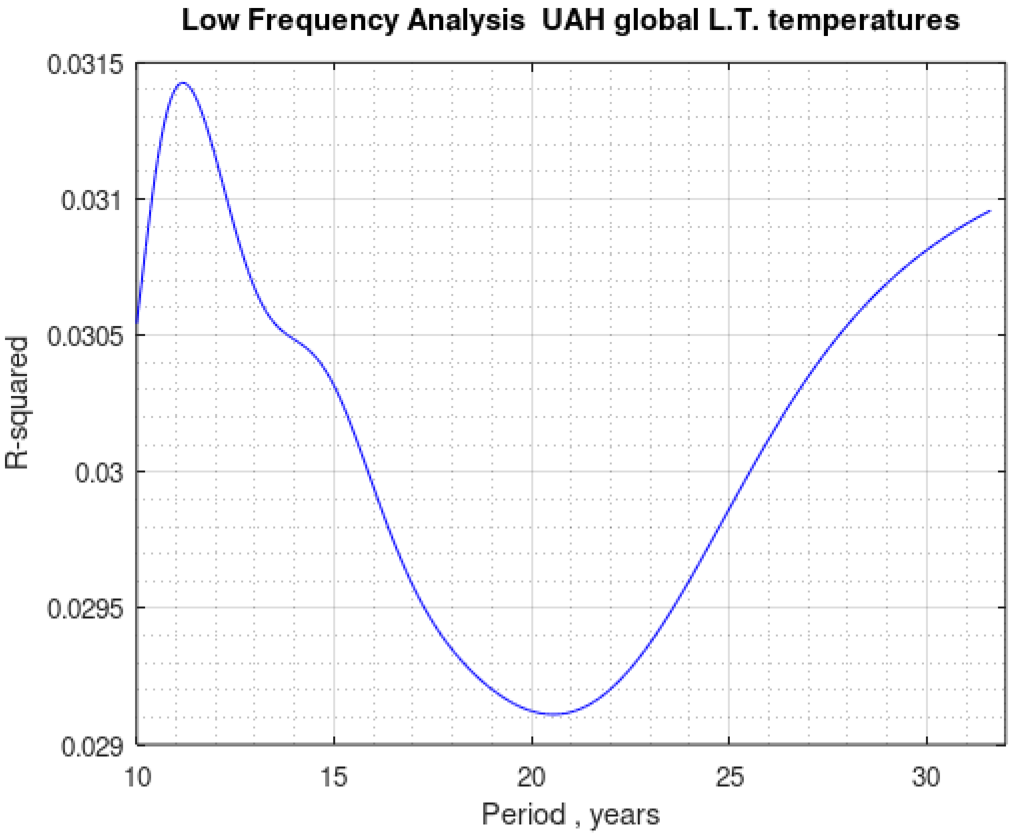
(numbers refer to the columns in the matrix)

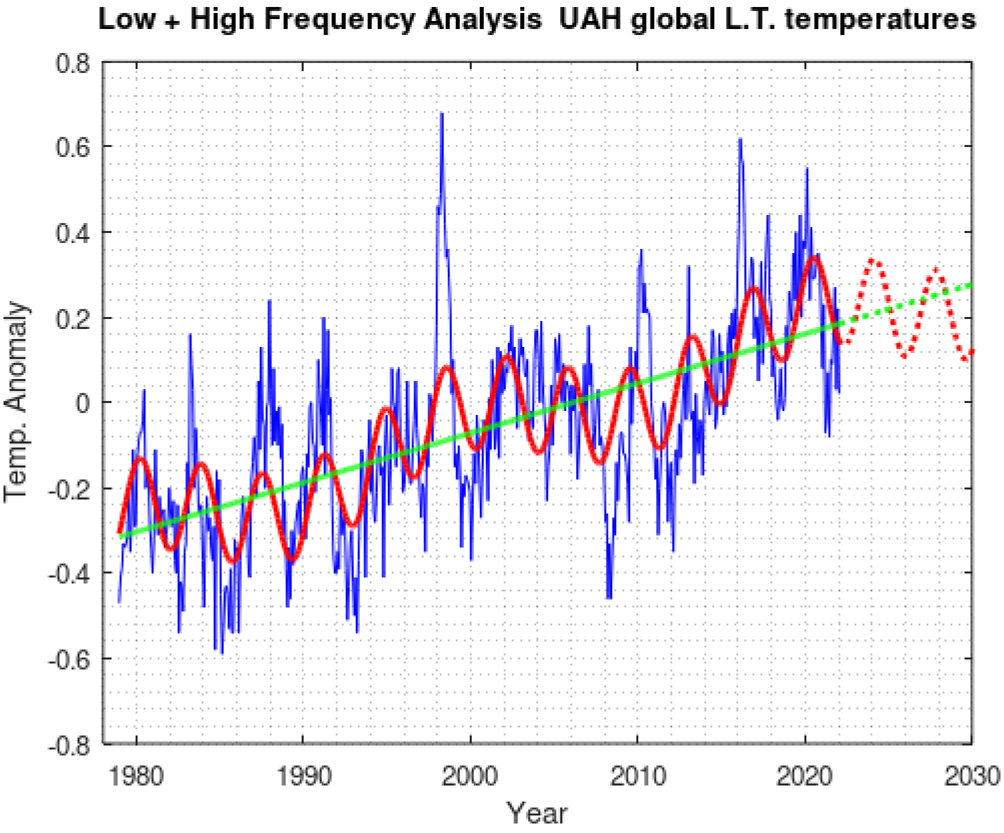
The combination of the 4 layers with the 24 regions gives a total of 24x4 = 96 possible graphs; in the following we use combination 3(globe) + lower troposphere. As a further option the program calculates and subtract a linear trend. Indeed in may happen that for a relatively long time span the temperature data are subjected to a bias or linear trend, which must be accounted for. Still a further option allows to predict an estimate for the future , on the basis of the frequency analyses and linear trend

```
clear all;clc;format short;format compact;
global Ud xd yd2 x sumS;
function sd = best(p)   % ------ frequency minimizer
  global Ud yd2 xd x sumS;
  Ud(:,1) = sin(2*pi/p*x);    % sin
  Ud(:,2) = cos(2*pi/p*x);    % cos
  [xd,sd,rd] = ols(yd2,Ud);   % ols = Ordinary Least Square refinement
endfunction                   % ------ end of minimizer
tit = 'UAH global L.T. temperatures'

%f=urlwrite('http://vortex.nsstc.uah.edu/data/msu/v6.0/tlt/
uahncdc_lt_6.0.txt','uahncdc_lt_6.0.txt');

S = fileread('uahncdc_lt_6.0.txt');
choice = 5;  % 3=Globe  4=Land  5=Ocean ...
startYear = 1978;                                      % <<<< User
endYear = 2022;
a1 = index(S,num2str(startYear)) - 1;
a2 = rindex(S,num2str(endYear)) + 170;
M = S(a1:a2);X = str2num(M);
x = X(':',1) + (X(':',2)-0.5)./12;                     %  0.5 = Januar
yd = X(':',choice);
nTot = length(yd);

% 1) find and detract linear slope
pSlope = polyfit(x,yd,1)
y2 = polyval(pSlope,x);
yd2 = yd - y2;
Ud = zeros(nTot,2);xd = zeros(2,1);
%xNew = linspace(endYear,endYear+10,55);
%ySin = polyval(p2,x); % ySin è un vettore colonna
%yNew = polyval(p2,xNew);yNew = yNew.';

% 2) find and detract low frequency component (between 10 and 31.6 years)
k = 0;
for i = logspace(1,1.5,200);
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);lF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
```

```
xlabel ('Period , years');ylabel ('R-squared');title (['Low Frequency Analysis
',tit]);axis([10,32]);
x2 = best(lF);
yd2 = yd2 - Ud*xd(:,1);
xd2(:,1) = xd(:,1);
figure;

% 3) find and detract high frequency component (between 0.4 and 10 years)
k = 0;
for i = logspace(-0.4,1,200);
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);hF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
xlabel ('Period , years');ylabel ('R-squared');title (['High Frequency Analysis
',tit]);axis([0,10]);
x2 = best(hF);
figure;

% 4) future estimate
yN = y2 + xd2(1)*sin(2*pi/lF*x) + xd2(2)*cos(2*pi/lF*x) + xd(1)*sin(2*pi/hF*x) +
xd(2)*cos(2*pi/hF*x);
plot (x,yd,'b',x,yN,'r','Linewidth',2,x,y2,'g','Linewidth',2);hold on;
xNew = linspace(endYear,endYear+10,55);
yNew = polyval(pSlope,xNew);
yN = yNew + xd2(1)*sin(2*pi/lF*xNew) + xd2(2)*cos(2*pi/lF*xNew) +
xd(1)*sin(2*pi/hF*xNew) + xd(2)*cos(2*pi/hF*xNew);
plot(xNew,yN,'r','Linewidth',2,'Linestyle',":",xNew,yNew,'g','Linewidth',2,'Linest
yle',":");
grid minor on;axis([1978,2030]);
xlabel('Year');ylabel('Temp. Anomaly');title (['Low + High Frequency Analysis
',tit]);
```







Different other application of the above algorithms to Sea Surface Temperatures are to be found here.