INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
I.C.T.P., P.O. BOX 586, 34100 TRIESTE, ITALY, CABLE: CENTRATOM TRIESTE

UNITED NATIONS INDUSTRIAL DEVELOPMENT ORGANIZATION
INTERNATIONAL CENTRE FOR SCIENCE AND HIGH TECHNOLOGY
INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS 34100 TRIESTE (ITALY) VIA GRIGNANO, 9 (ADRIATICO PALACE) P.O. BOX 586 TELEPHONE 040-224572 TELEFAX 040-224575 TELEX 460449 APH I

SMR/548-6

Course on Oceanography of Semi-Enclosed Seas
15 April - 3 May 1991

"SEASPECT: A Software Package for Oceanographic
Time Series Analysis"

A. LASCARATOS
University of Athens
Department of Applied Physics
Greece

Please note: These notes are intended for internal distribution only

SEASPECT : A software package for Oceanographic Time Series Analysis

[Software package for oceanographic]
[illegible]

by

A. Lascaratos, J. Daskalakis, L. Perivoliotis and N. Vlastos

University of Athens
December 1990

Introduction.

SEASPECT is a software package specially designed for oceanographic time series analysis. It contains five main programs (PROJ, SPECT, CROSS, ROT, SERIES), which perform data handling and manipulation, univariate, bivariate and rotary spectral analysis. The package has initially been developed by, and installed at, the Department of Applied Physics (Laboratory of Meteorology and Oceanography) of the University of Athens, Greece. The package was implemented using the Fortran 77 programming language. The initial version was written for the Hewlett-Packard Series 9000/300 supermicro computers in HP-Fortran under the HP-UX (Unix) operating system.

At the request of IOC/UNESCO, a PC version was engineered using Microsoft Fortran V5.0 under MS-DOS for the IBM PC/XT/AT compatible computers. These two versions are practically identical, exception made for the implementation of screen and plotter graphics which follow a different philosophy on the Hewlett-Packard supermicro and on the PC's. The present user's manual refers to the PC version of SEASPECT.

All calculations in the package are performed in single precision i.e. one real number = 4 bytes. The program is able to recognize automatically and handle adequately all kinds of graphics adapters and displays (CGA, Hercules, EGA, VGA, combined with colour or monochrome displays). If no graphics adapter is installed the drawing capabilities of the program are not available, but the other functions are unaffected. The graphics screen can be printed on almost any dot matrix printer. The print screen feature is certified on all Epson models (FX,MX,EX,LQ), Star Micronics models (SD, SR, NL, ND, LC) and IBM Proprinters (XL). In the present version (V3.2) pen plotters are not supported. In order to run on the PC the minimum main memory (RAM) requirement is 512 K and when the program is initiated at least 300 K must be available to the user. The executable code (program size) is about 100 K for each of the five main programs of the package, and uses about 126 K for data and scratch space.

Note: 1.- The package is delivered with a file "COURB.FON". If you are working in another directory than the one containing the "SEASPECT" package, make sure to copy the "COURB.FON" file in your working directory. Otherwise graphic capabilities will not be available.

2.- The package is delivered with a program named "MSHERC". If you are going to work on a Hercules type screen, run this program before using the SEASPECT package. This operation will have to be repeated only once each time you "switch on" your computer.

To run the package, type SEASPECT and press ↵. Next enter your SEASPECT "login", which is the serial number appearing on your two original program diskettes. (This procedure should be followed each time you enter the package shell). A menu containing the five main programs of the package, will appear. You can run any of the programs of the package one or more times, then switch to another, without exiting the SEASPECT shell.

SEASPECT PACKAGE

TIME SERIES PROCESSING PROGRAM **PROJ**

TABLE OF CONTENTS

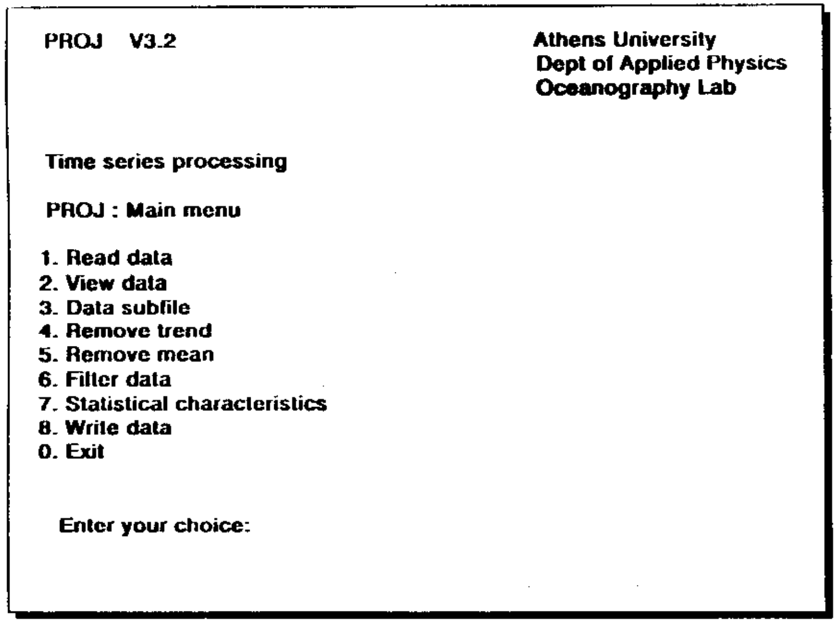
1. PROGRAM CHARACTERISTICS

With PROJ you can read a file containing a time series, view it on the screen, compute it's statistical characteristics, subsample your file and apply a digital filter to obtain the low and high-pass parts of your signal. The input consists of a set of N real numbers X(1),X(2),...,X(N) assumed to be samples of a physical signal equally sampled at intervals T_s

Program limitations:
- Times series length ≤ 10000 points

2. USING *PROJ*

To run the program select **1** from the SEASPECT menu and press ↵. Information about the program, appears. By pressing ↵, you enter the main menu:

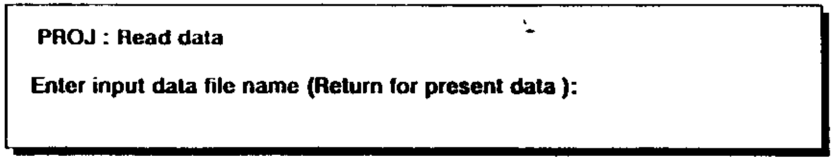


```
PROJ   V3.2                                   Athens University
                                              Dept of Applied Physics
                                              Oceanography Lab


Time series processing

PROJ : Main menu

1. Read data
2. View data
3. Data subfile
4. Remove trend
5. Remove mean
6. Filter data
7. Statistical characteristics
8. Write data
0. Exit


  Enter your choice:
```

The cursor now is next to 'Enter your choice'. To begin using the program type 1 and ↵.

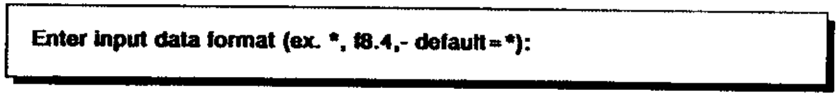
Submenu 1. Read data

You have entered the first menu of the program. The screen now shows:



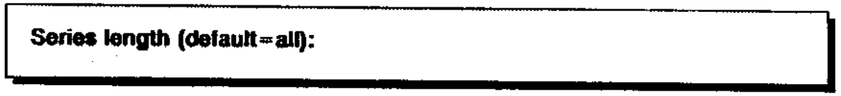
```
PROJ : Read data

Enter input data file name (Return for present data ):
```

Enter the name of the data file and press ↵. If a file has already been read and you do not wish to load another one, simply press ↵ to continue. Note, that all the files must contain a four line title, which is read and displayed by the program. If the specified file does not exist or a wrong path name has been given, then warning message 'File does not exist' will appear. Try again, giving the correct path and file name.



```
Enter input data format (ex. *, f8.4,- default=*):
```

Specify the format in which the data will be read and press ↵. If you wish to read your data in free format, enter * or simply press ↵, but note that if your file contains more than one column, only the first one will be read.



```
Series length (default=all):
```

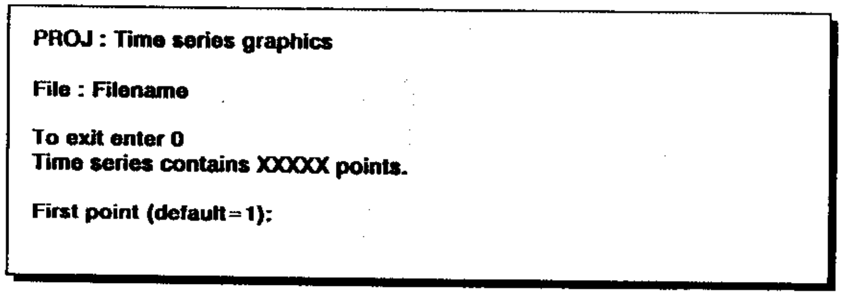
Enter the number of points to be read, or press ↵ if you want to read all the data points contained in the file.

After answering the last question, you return to the main menu.

Note, that none of the other options of the main menu will be active, if you have not specified the data.

Submenu 2. View data.

The program allows the user to visualize on the screen the time series that has been read. Select the option 2 of the main menu and press ↵. The screen now shows:



```
PROJ : Time series graphics

File : Filename

To exit enter 0
Time series contains XXXXX points.

First point (default=1):
```

Enter the first point of the series, you wish to be plotted (default option is the first point).

```
Last point (default=end):
```

Enter the last point of the series, you wish to be plotted (default option is the last point).

```
Enter y-min:
```

```
Enter y-max:
```

Specify the minimum and maximum value of your y-axis and press ↵. Appropriate values can be chosen by referring to submenu 7 -'Statistical characteristics' (see below).

```
Enter major ticy value:
Enter minor ticy value:
```

Specify the major and minor tics for y-axis

```
X-scale factor (default=1):
```

By default the X-axis will represent the total number of samples which are included between the first and last point you have already selected. The program allows the user,to alter the X-axis units by setting a different factor than 1.

Example: If you have a total of 2400 measurements with a sampling interval of 1 hour, then (by default) the X-axis will be scaled from 0 to 2400. If you wish the scaling to represent days, set the above factor equal to 24, and the X-axis will be scaled from 0 to 100.

```
X-axis shift (positive, default=0):
```

By default the numbering of the x-axis, at it's intersection with the y-axis, starts from 0. If you specify a positive number other than zero, then the whole x-axis numbering will be shifted by an amount equal to the specified number. Suppose that the plot will consist of 20 daily values, starting on the 3rd of a specific month. By default your x-axis will be numbered 0 to 20. If you shift your axis by the positive number 3, then the numbers on the x-axis will correspond to actual dates.

```
X-minimum (default=0):
X-maximum (default=last point):
```

Specify the minimum and maximum values for your x-axis. The default options are 0 and last point of the time series you selected. Note, that if you have set a scale factor different than 1 then default values will be divided by this factor and if you have shifted x-axis by a number, then this number will be added to the default values.

```
Enter major ticx value:
Enter minor ticx value:
```

Specify the major and minor tics for x-axis

```
Curve line style (default=0):
```

Various line types can be chosen for the graph, using different numbers. The default option is 0 (solid line) and alternatives styles are:

1 (dashed line), 2 (dotted line), 3 (dot-dashed line).

```
Graph title (1st line,60 chars):
Graph title (2nd line,60 chars):
```

Enter the main title of the graph (2 lines are allowed, with 60 characters maximum per line). By simply pressing ↵ you can specify blank titles.

```
X-axis title (20 chars):
Y-axis title (20 chars):
```

Specify the axes title (maximum 20 characters) and press ↵.

The graph now appears on the screen.

At the bottom of the screen you see the following message:

New graph(1), Plot(2), Exit(0)

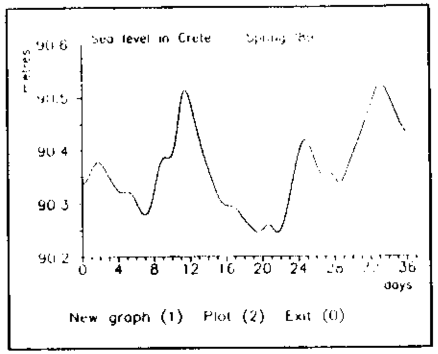


Figure 1 : PROJ graphics

1 sends you back to entry menu 2 (View data), **2** sends the graph to the printer and **3** returns you to the main menu.

Submenu 3. Data subfile

In order to subsample a part of your time series, select the option 3 of main menu and press ↵. The screen now shows:

```
PROJ : Data subfile.

File : Filename

First point (default=1):
```

Enter the first point of the subfile you wish to create (default option is the first point of the original time series as read from the disk).

```
Last point (default=end):
```

Enter the last point of your subfile (default option is the last point of the original time series). Note, that if you create a subfile, only that part of your original time series will be active in the program. To save your subfile as a disk file, go through entry menu 8 - 'Write data' (see below).

4. Remove trend.

Option 4 removes the linear trend (estimated by least square fitting) from your original time series or from the subfile you have created.

5. Remove mean.

Option 5 removes the mean value from your original time series or from the subfile you have created.

Submenu 6. Filter data.

With this option a digital filter can be applied on the data. Select the option 6 of the main menu and press ↵. The screen shows:

```
PROJ : Filter data

File : Filename

Enter filter filename (Enter to exit):
```

Enter the name of the disk file which contains the digital filter you wish to apply on the data and press ↵. Note, that the filter files must contain only the weights (one weight per line), without any title. In order to save the filtered time series on your disk, go through submenu 8 - 'Write data' (see below).

Submenu 7. Statistical characteristics.

This option gives you the statistical characteristics of your time series. Select 7, in the main menu and press ↵.

The following will appear on the screen:

Xmin: minimum value of the time series

Xmax: maximum value of the time series

Xmean: mean value of the time series

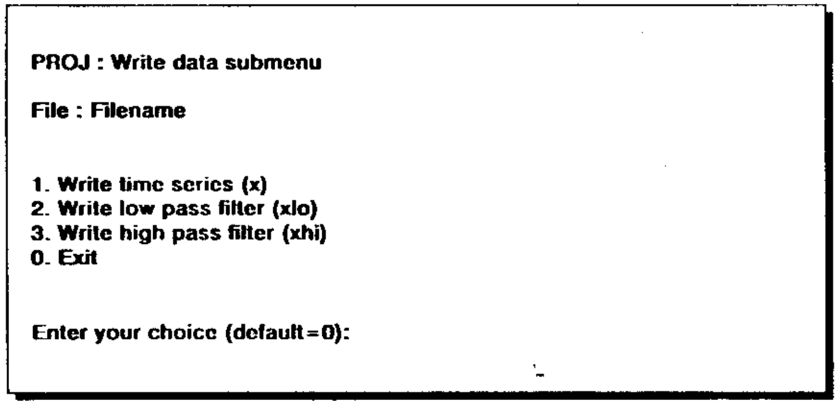
Xsd: standard deviation defined as $\sqrt{[\Sigma x^2 - (\Sigma x)^2/n]/(n-1)}$

Energy: defined as Σx^2
Emean: mean energy defined as $\Sigma x^2/n$
Slope: defined as
Length: the length of time series that has been read.

When you read the data, the statistical characteristics are based on the original length of time series as read from the disk file. But, if you have created a subfile (through entry menu 3) then the statistical characteristics of the subfile will be displayed. Press ↵, to return to the main menu.

Submenu 8. Write data.

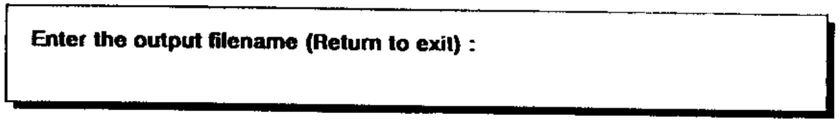
To save the data previously selected as a subfile or resulting from the time series filtering (submenu 3 and 6 respectively) on a disk file, select option 8 of the main menu and press ↵. You will see on screen:



```
PROJ : Write data submenu

File : Filename


1. Write time series (x)
2. Write low pass filter (xlo)
3. Write high pass filter (xhi)
0. Exit


Enter your choice (default=0):
```

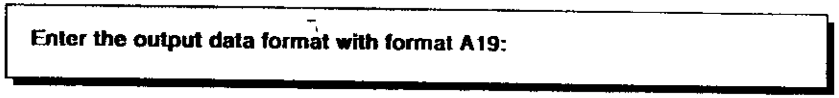
Select 1 ↵ to save the contents of your subfile on a disk file, 2 ↵ to save the low pass part of your filtered series and 3 ↵ to save it's high pass part.

Important Note: By default, the program assumes that a low-pass filter has been applied, since this is the most frequent case in Oceanography. If another kind of filter is used (High-pass, Band-pass or Band-stop) then the time series produced by the filter will be erroneously named "low" and its difference from the original will be named "high". The program will correctly apply any digital filter but this point should be kept in mind when using a non low-pass filter.



```
Enter the output filename (Return to exit) :
```

Enter the name of the file you will create.



```
Enter the output data format with format A19:
```

Specify the format in which the data will be written (19 characters allowed for format description).

Type 0 ↵ to return to the main menu.

To exit PROJ, type 0 ↵.

SEASPECT PACKAGE

UNIVARIATE SPECTRAL ANALYSIS PROGRAM **SPECT**

TABLE OF CONTENTS

1. PROGRAM CHARACTERISTICS

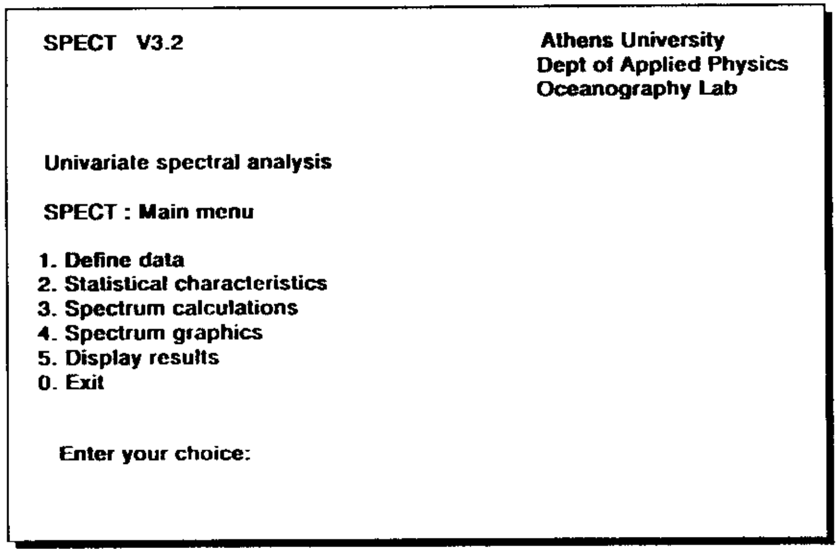
Program SPECT calculates the spectral characteristics of a discrete signal. The input consists of a set of N real numbers X(1), X(2),...X(N) assumed to be samples of a physical signal with sampling period $T_s = 1$. The output is the power spectrum of the time series (for more details concerning theory and methodology, see the theoretical notes in the first part of the manual).

Program limitations:

- Times series length ≤ 8192 points
- Power spectrum ≤ 8192 points

2. USING *SPECT*

To run the program select **2** from SEASPECT menu and press ↵. Information about the program appears. By pressing ↵, you enter the main menu:

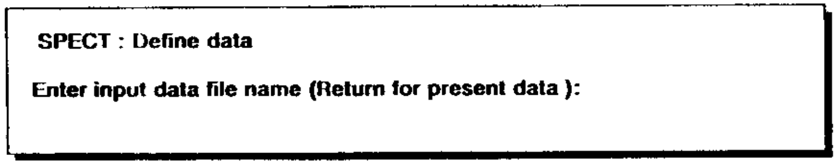


```
SPECT   V3.2                              Athens University
                                          Dept of Applied Physics
                                          Oceanography Lab


Univariate spectral analysis

SPECT : Main menu

1. Define data
2. Statistical characteristics
3. Spectrum calculations
4. Spectrum graphics
5. Display results
0. Exit

  Enter your choice:
```

The cursor is now next to 'Enter your choice'. To begin using the program, type **1** ↵

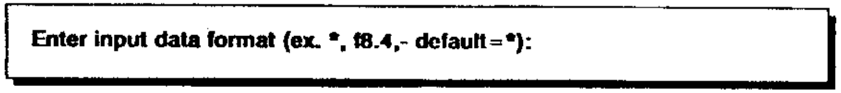
Submenu 1. Define Data

You have entered the first menu of the program. The screen now shows:



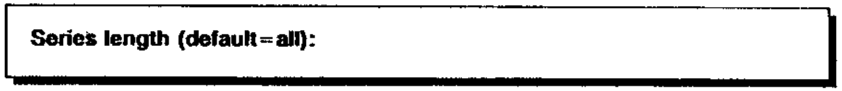
```
SPECT : Define data

Enter input data file name (Return for present data ):
```

Enter the name of your data file and press ↵. If a file has already been read and you don't want to load another one, simply press ↵ to continue. Note, that all the files must contain a four line title, which is read and displayed by the program. If the specified file does not exist or a wrong pathname has been given, the warning message 'File does not exist' will appear. Try again, giving the correct path and file name.



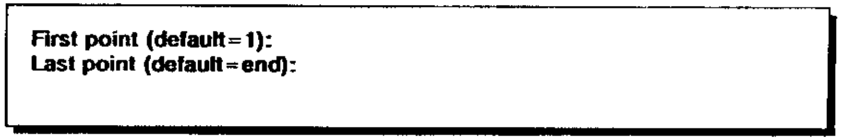
```
Enter input data format (ex. *, f8.4,- default=*):
```

Specify the format in which the data will be read and press ↵. If you wish to read your data in free format, enter * or just press ↵, but note that if your file contains more than one column, only the first one will be read.



```
Series length (default=all):
```

Enter the number of data points to be read, or press ↵ if you want to read all the data points contained in the file.



```
First point (default=1):
Last point (default=end):
```

Specify the first and last point of the series read, to be used by the program. This defines the part of the time series on which spectrum calculations will be performed (NMAX). Default options are (pressing ↵), the first and last point of the series.

Next, the program returns to the main menu.

Note, that none of the other options of the main menu will be active, if you have not specified the data.

Submenu 2. Statistical characteristics.

Select this option by typing 2 ↵, in the main menu.

The following statistical characteristics of the series chosen and read, will appear as follows:

Xmin, Xmax, Xmean: the min., max. and mean value of the time series

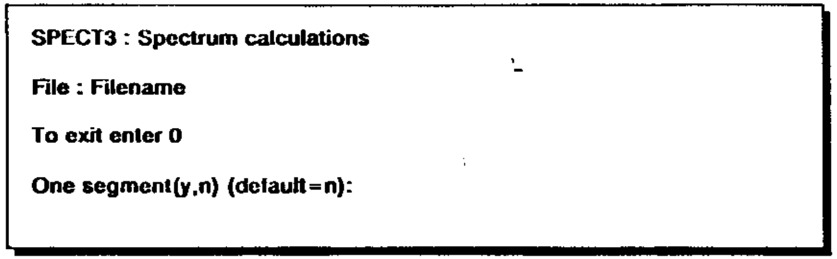
Xsd: standard deviation defined as $\sqrt{[\Sigma x^2-(\Sigma x)^2/n]/(n-1)}$

Energy: defined as Σx^2
Emean: mean energy defined as $\Sigma x^2/n$
Slope: defined by least square estimation
Length: the length on which the previous calculations are based

When the spectrum has not been estimated, the statistical characteristics are based on the original length of the time series as read from the data file. But, when the spectrum is calculated (option 3 of main menu), another reference length may be defined as we will see below. This length is the length on which the spectrum estimation is based, excluding of course, any zero padding that has been possibly used. In this case, the statistical characteristics will be based on this newly defined length. By pressing ↵, return to the main menu

Submenu 3. Spectrum calculations

To begin the spectrum calculations, select option 3 of the main menu and press ↵. The screen now shows:



SPECT3 : Spectrum calculations

File : Filename

To exit enter 0

One segment(y,n) (default=n):

If one segment is chosen (answer 'y" to the above question), then the series is padded with zeros until its length reaches the nearest integral power of two.

Example-1: if length of series=1000, 24 zeros are added at the end of the series to reach 1024 points

Example-2: if length=1026, 1022 zeros are added to sum up to 2048 points.

The spectrum calculation will now be based on this new length and the spectrum estimation will proceed with a one pass FFT on the whole length.

It is evident from the theory (see 1.4), that in the case of one segment the spectrum estimation will have a statistical error of 100%.

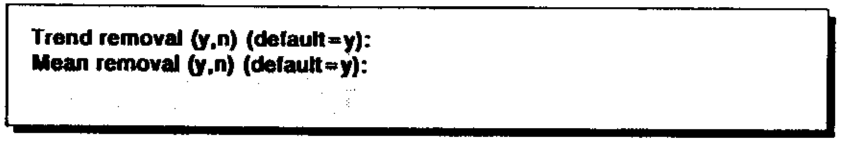
If you answer "n", the program will ask to define the length of the segment to be used.

Segment size:

The segment length NT must be an integral power of two and ≤ 8192. The operational length of the time series is defined by the maximum number of 50% overlapping segments that the series can contain.

Consider a time series with 1000 data points. If a 512 segment size is chosen, the operational length will be 768 points and will contain two 50% overlapping segments. If the segment size is 256, the operational length will be 856 points and will contain six segments.

Note that the estimation error of your spectrum will decrease as the number of segments increases, but at the same time its resolution will decrease.



Trend removal (y,n) (default=y):
Mean removal (y,n) (default=y):

Here, you have to answer if the linear trend contained in the time series and detected by least squares estimation or the mean value of the time series, will be subtracted or not. If you have selected 'one segment' for your spectrum calculations and answer "y" in these questions, then the trend and mean will be removed from all the values except the tail of zeros. If you have specified a window size, this process will be applied to the individual segments, i.e. the trend and mean will be removed from each segment.

```
Window (0=No,1=Parzen,2=Hamming) (default=2):
```

Application of a window consists of multiplying the time series or the individual segments by an appropriate bell shaped function. Two kinds of windowing functions are available. The first (Parzen) is a piecewise linear (triangular) window and the second (Hamming) consists of a half period of a cosine function. The default option here is 2 (Hamming).

By answering this last question, you return to the main menu.

Submenu 4. Spectrum graphics

The program allows the user to visualize on the screen the power spectrum that has been estimated, through the entry menu 3.

Select the option 4 of the main menu and press ↵. The following will appear on the screen:

```
SPECT3 : Spectrum graphics

File : Filename

To exit enter 0

Linear(1),Log spectrum(2) (default=1):
```

The energy values calculated by SPECT, are normalized. If you choose the option 'Linear spectrum' (answer 1) then the values of y-axis will be between 0 and 1. 'Log spectrum'(answer 2) calculates the logarithm of normalized values multiplied by 10 and plots them using linear scale. Obviously, in this case the maximum value of y-axis will be 0, while the minimum will extend toward negative numbers. In order to specify the minimum value to be plotted (threshold), you will be asked by the program:

```
Threshold value (<0):
```

The answer to this question, will be the minimum of your y-axis (a negative value).

```
On X-axis: Channel (1) or Frequency (2) (default =1):
```

Here, you have to choose, if your x-axis will represent spectrum channels or spectrum frequencies. The option for channels is 1, and for frequency is 2. In order to calculate the frequency of a given channel, use of the following equation is necessary:

$F=(CH/NT)*(1/T_s)$ where F stands for frequency, CH for spectrum channel, NT for segment length (window size), and T_s for the sampling interval. So, if 'frequency' is selected (option 2), it is necessary to reply to the following:

```
Multiplying factor for x-axis (default=1):
```

The above factor refers to T_s, with the default value set to 1 and so the frequency units are cycles/sampling period. The program allows the user to convert the frequency units, using a different factor.

Example-1: If the sampling interval of your measurements is 1 hour, then the default frequency units are cycles/hour. To convert these units to cycles/day, you have to set the multiplying factor equal to 24.

Example-2: If sampling period=1 day, the default units are cycles/day, and to convert them to cycles/hour, the multiplying factor should be set equal to 1/24≈0.042.

After answering the above questions and pressing ↵, the program informs you about the total number of channels the spectrum contains, which is equal to NT/2+1 (from 0 to NT/2).

```
First channel (default=0):
```

Enter the first channel of the spectrum, that is to be plotted (default option is channel 0).

```
Last channel (default=NT/2):
```

Enter the last channel of spectrum, that is to be plotted (default option is the last channel).

```
Enter x-min (default=first channel or frequency):
```

Specify the minimum value for your x-axis (default value is the first channel selected or the corresponding frequency the if the 'frequency' option is has been chosen)

```
Enter x-max (default=last channel or frequency):
```

Enter the maximum value of x-axis (default option is the last channel you selected or the corresponding frequency)

```
Enter major ticx value:
Enter minor ticx value:
```

Specify the major and minor tics for x-axis.

```
Enter major ticy value:
```

Enter the major tic value for y-axis. Note, that if 'Linear spectrum' has been chosen then y-axis will be extended from 0 to 1, but in the case that 'Log spectrum' has been selected, then the minimum value for y-axis will be the threshold value you gave and the maximum will be 0.

```
Enter minor ticy value:
```

Specify the minor tic for y-axis.

```
Curve line style (default=0):
```

Various line types can be chosen for the graph, using different numbers. The default option is 0 (solid line) and alternative styles are:
1 (dashed line), 2 (dotted line), 3 (dot-dashed line).

```
Graph title (First line,60 chars):
Graph title (Second line,60 chars):
```

Enter the main title of the graph (2 lines are allowed, with 60 characters maximum per line). By pressing ↵ you can specify blank titles.

```
X-axis title (20 chars):
Y-axis title (20 chars):
```

Specify the axes titles (maximum 20 characters).

The spectrum graph will now be displayed on screen

At the bottom of the screen the following appears:

New graph (1), Plot (2), Exit (0)

By pressing 1 (new graph), return to entry menu 4 (spectrum graphics)

By pressing 2 (plot), the graph will be sent to printer.

By pressing 0 (exit), return to main menu.

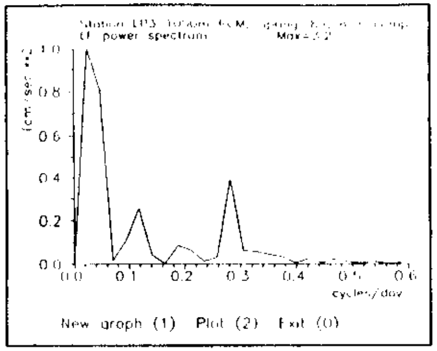


Figure 1 : SPECT graphics

Submenu 5. Display results

Select the option 5 of main menu and press ↵ . The screen now shows:

SPECT : Display results

File : Filename

To exit enter 0

Screen(1) or File(2) (default=1):

Spectrum results may be sent to screen or to a disk file. The default option sends the results to screen. If 'File' option (2) is chosen, then the program requests the file name where the results will be written.

Enter output file name:

Linear (1) Log (2) spectrum (default=1):

Same as submenu 4 (spectrum graphics)

Full output (y,n) (default=n)

If the reply to this question is "y", then NT/2+1 channels (from 0 to NT/2) will be sent to screen or file. The default option here is "n" and SPECT requests the first and last channel that is to be reported:

First channel:
Last channel:

Job title:

Specify the job title and press ↵. This title will appear on the output (screen or file).

The spectrum results as viewed on the screen or sent to a disk file consist of:

a. Four lines of comments that identify the run case and general signal characteristics which are:

First line: - Job title

Second line: - Number of points retained for calculations (NMAX)
- Number of points used for spectrum calculations (N ≤NMAX)
- Number of segments (NSP)
- Segment length (NT)
- B_{min} and B_{max} minimum and maximum bounds of the 95% confidence level.

Third line: - Mean energy of time series
- Mean of time series

- SNF = spectrum normalizing factor.
- ET = total energy
- EF = Energy factor

Fourth line: - Information on whether trend and mean have been removed.
- The number of channels contained in the output file.
- The format of the output file.

b. A basic pattern is repeated four times across each line of the output file.

- If the spectrum calculation has been performed on one segment only, then the pattern consists of three items: channel number, power spectrum estimation and phase estimation
- If the spectrum calculation has been based on many segments, the pattern consists of two items: the channel and the power spectrum estimation

When the above results are viewed on screen, or are written in a file, the program returns to main menu

To exit SPECT, type 0 ↵.

SEASPECT PACKAGE

BIVARIATE SPECTRAL ANALYSIS PROGRAM **CROSS**

TABLE OF CONTENTS

1. PROGRAM CHARACTERISTICS

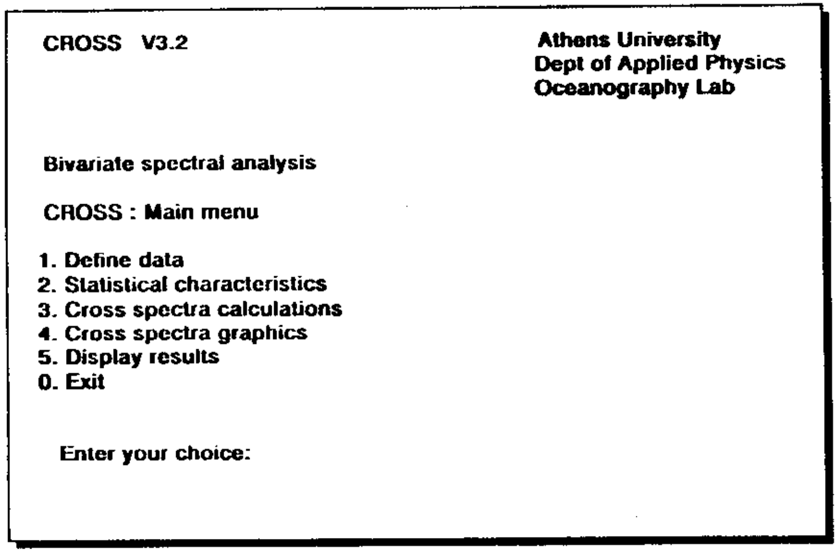
Program CROSS calculates the cross (combined) spectral characteristics of a pair of discrete signals. Two sets of real numbers X(1),.. X(N) and Y(1),..Y(N) are provided as input to the program. These values are assumed to be synchronous samples of two physical signals with sampling period $T_s=1$. The program output consists of an estimation of the cross power spectrum based on specified length. This cross spectrum will be complex and we will refer to its real and imaginary parts or its modulus and phase.

Program limitations:

- Time series length ≤ 8192 points
- Power spectrum ≤ 8192 points

2. USING *CROSS*

To run the program select **3** from the SEASPECT menu and press ↵. Information about the program, appears. By pressing ↵, you enter the main menu:

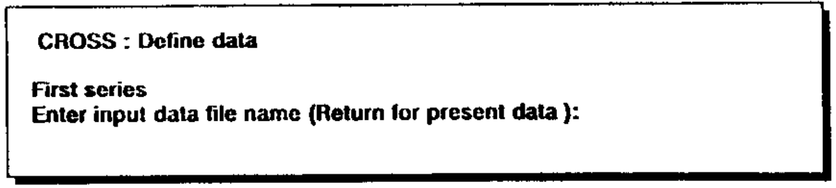


```
CROSS   V3.2                                 Athens University
                                             Dept of Applied Physics
                                             Oceanography Lab


Bivariate spectral analysis

CROSS : Main menu

1. Define data
2. Statistical characteristics
3. Cross spectra calculations
4. Cross spectra graphics
5. Display results
0. Exit


  Enter your choice:
```

The cursor is now next to 'Enter your choice'. To begin using the program, type **1** ↵.

Submenu 1. Define Data

You have entered the first menu of the program. The screen now shows :



```
CROSS : Define data

First series
Enter input data file name (Return for present data ):
```

Enter the name of the first data file and press ↵. If a file has already been read and you don't want to load another one, simply press ↵ to continue. Note, that all the files must contain a four line title which is read and displayed by the program. If the specified file does not exist or a wrong path name has been given, then the warning message 'File does not exist' will appear. Try again, giving the correct path and file name.

Enter input data format (ex. *, f8.4,- default=*):
Series length (default=all):
First point (default=1):
Last point (default=end):

(for an explanation of the above questions, see SPECT user's guide)

The above procedure is repeated for the second series. The two series may reside in the same or different files and may be read in with different lengths. However, the minimum of the two lengths is considered as the common reference length and all subsequent calculations will be based on this length.

After answering the last question about the second series, you return to the main menu.

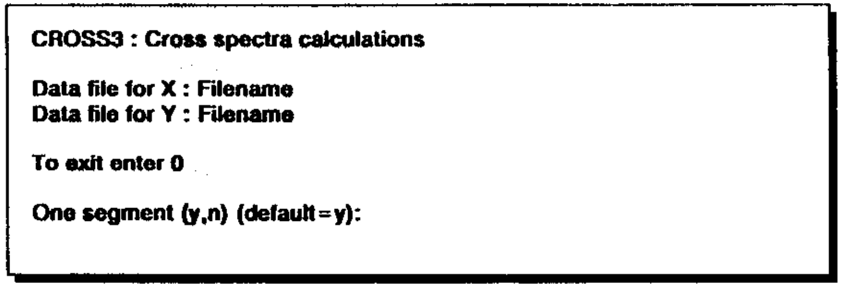
Note, that none of the other options of the main menu will be active, if you have not specified the data.

Submenu 2. Statistical characteristics.

(see SPECT user's guide)

Submenu 3. Cross spectra calculations

To begin the spectrum calculations, select the option 3 of the main menu and press ↵. The screen now shows:



```
CROSS3 : Cross spectra calculations

Data file for X : Filename
Data file for Y : Filename

To exit enter 0

One segment (y,n) (default=y):
```

Segment size:

Trend removal (y,n) (default=y):

Mean removal (y,n) (default=y):

Window (0=No,1=Parzen,2=Hamming) (default=2):

(see SPECT user's guide)

Submenu 4. Cross spectra graphics

The program produces graphic outputs of the coherence, phase and gain of the cross spectrum.

Select option 4 of the main menu and press ↵. The following will appear on the screen:

```
CROSS3 : Cross spectra screen graphics

Data file for X : Filename
Data file for Y : Filename

To exit enter 0 to any of the first two questions

Coherence (1), Phase (2), Gain (3) (default=1):
```

Select 1, 2 or 3 and press ↵. The program will ask you the following: (see SPECT manual)

Channel (1) or Frequency (2) (default =1):

Multiplying factor for x-axis (default=1):

First channel (default=0):

Last channel (default=NT/2):

Enter x-min (default=first channel or frequency):

Enter x-max (default=last channel or frequency):

Enter major ticx value:

Enter minor ticx value:

If you have chosen 'Phase' or 'Gain' graphs,you will have to specify the minimum and maximum values for y-axis. If you have chosen the 'Coherence' option, the minimum and maximum values of y-axis are automatically set to 0 and 1 respectively.

Enter y-min:

Enter y-max:

Note, that the phase values calculated by CROSS extended from -180 degrees to 180.

Enter major ticy value:

Enter minor ticy value:

Curve line style (default=0):

Graph title (First line,30 chars):

Graph title (Second line,30 chars):

X-axis title (20 chars):

Y-axis title (20 chars):

(see SPECT manual)

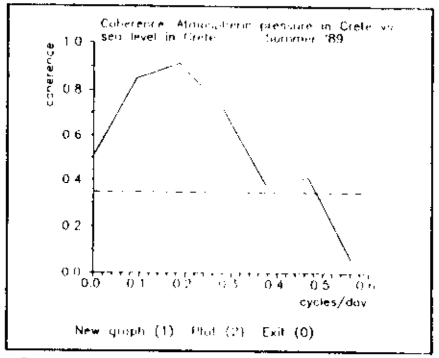

Coherence Atmospheric pressure in Crete vs
sea level in Crete Summer '89
coherence
1 0
0 8
0 6
0 4
0 2
0 0
0.0
0 1
0 2
0 3
0 4
0 5
0 6
cycles/day
New graph (1) Plot (2) Exit (0)

Figure 1: CROSS graphics - Coherence

The coherence, phase or gain graphs will now be displayed on the screen.

The dashed line that appears in the coherence graph, represents the 95% confidence level.

At the bottom of the screen the following appears:

New graph (1), Plot (2), Exit (0)

By pressing 1 (new graph), you return to entry menu 4 (spectrum graphics)

By pressing 2 (plot), the graph will be sent to printer.

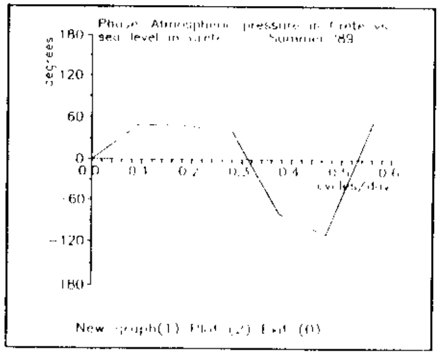
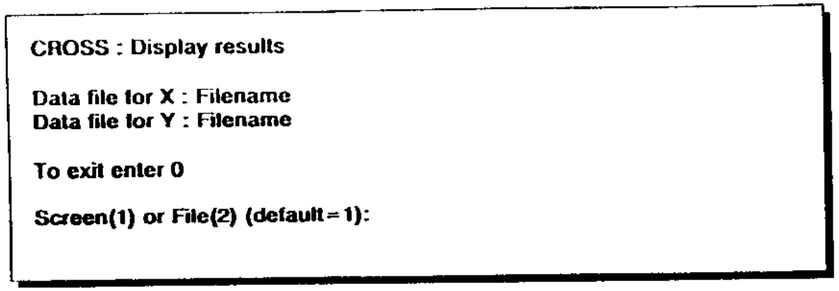

Phase Atmospheric pressure in Crete vs
sea level in Crete Summer '89
degrees
180
120
60
0
-60
-120
180
0 0
0 1
0 2
0 3
0 4
0 5
0 6
cycles/day
New graph(1) Plot (2) Exit (0)

Figure 2 : CROSS graphics - Phase

By pressing 0 (exit), you return to the main menu.

Submenu 5. Display results

Select the option 5 of main menu and press ↵. The screen now shows:



```
CROSS : Display results

Data file for X : Filename
Data file for Y : Filename

To exit enter 0

Screen(1) or File(2) (default=1):
```

Cross spectrum results may be sent to screen or to a disk file. The default option sends the results to screen. If 'File' option (2) is chosen, then the program requests the file name where the results will be written.

Enter output file name:
Full output (y,n) (default=n)
First channel:
Last channel:
Job title:
(see SPECT user's guide)

The cross spectrum results as viewed on the screen or sent to a disk file consist of:

a. Four lines of comments that identify the run case and general signal characteristics.

First line: - Job title

Second line: - Number of points retained for calculations (NMAX)
- Number of points used for spectrum calculations ($N \leq NMAX$)
- Number of segments (NSP)
- Segment length (NT)
- B_{95} and B_{80} coherence confidence levels

Third line: - Mean energy of x-time series
- Mean of x-time series
- Mean energy of y-time series
- Mean of y-time series

Fourth line: - Information on whether trend and mean have been removed
- The number of channels contained in the output file
- The format of the nine-column output body:

b. The output file contains:
Channel (k), Frequency (f_k), Power spectrum of x series $G_{xx}(f_k)$, Power spectrum of y series $G_{yy}(f_k)$, Gain = $|G_{xy}|/G_{xx}$, Δgain = Standard error of the gain estimation, Phase = $\angle G_{xy}$, Δphase = Standard error of the phase estimation and Coherence = γ^2 = $|G_{xy}|^2/(G_{xx}G_{yy})$

When the above results are viewed on screen, or are written in a file, the program returns to the main menu

To exit CROSS, type 0 ↵.

SEASPECT PACKAGE

ROTARY SPECTRAL ANALYSIS PROGRAM ROT

TABLE OF CONTENTS

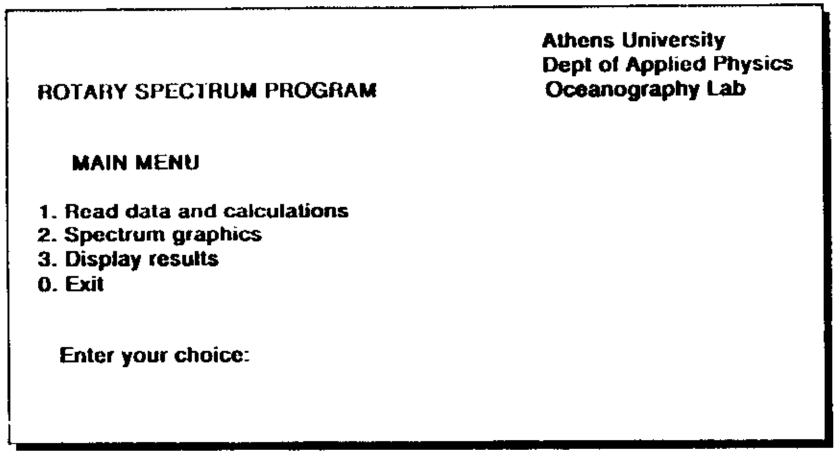
1. PROGRAM CHARACTERISTICS

Program ROT computes the rotary spectrum of two-dimensional vector time series. Before running ROT, the cross-spectrum between the two components (x and y of a right-handed coordinate system) of the vector time series has to be computed through program CROSS and its output saved on a file (see CROSS user's guide). This file will be used as the input for program ROT. The program outputs are the rotary power spectrum components.

Program limitations:
- CROSS results file ≤ 2049 channels

2. USING *ROT*

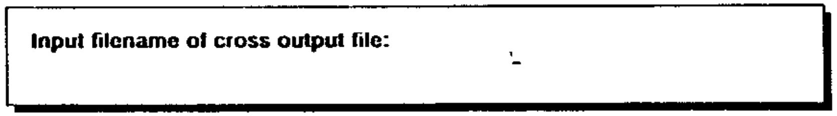
To run the program select **4** from the SEASPECT menu and press ↵. Information about the program, appears. By pressing ↵, you enter the main menu:



```
                                          Athens University
                                          Dept of Applied Physics
ROTARY SPECTRUM PROGRAM                   Oceanography Lab

   MAIN MENU

1. Read data and calculations
2. Spectrum graphics
3. Display results
0. Exit

  Enter your choice:
```

The cursor now is next to 'Enter your choice'. To begin using the program, type 1 and ↵.

Submenu 1. Read data and calculations



```
Input filename of cross output file:        _
```

Enter the name of the file to be used and press ↵. (Remember this file must be an output file of program CROSS !)

Note, that CROSS files intended to be used as entries to ROT should start from channel 0, otherwise an error message will be displayed.

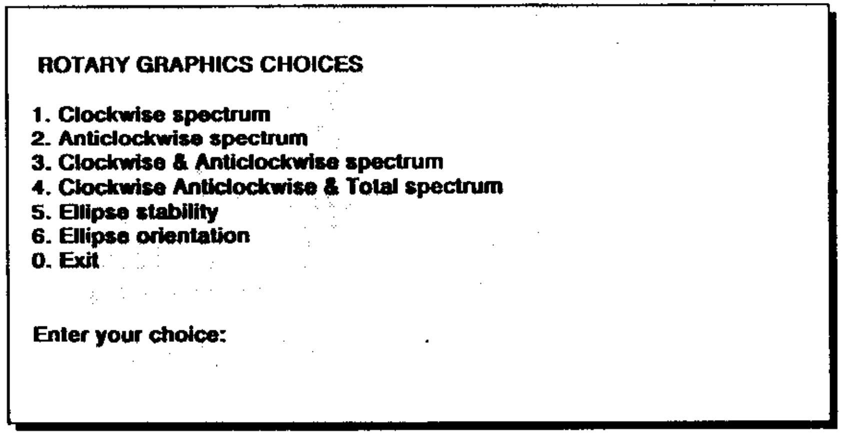
If the specified file does not exist or a wrong path name has been given, then the warning message 'File does not exist' will appear. Try again, giving the correct path and file name. The file will be read in free format, and then ROT informs you about the total points which have been read. By pressing ↵, the program calculates the components of rotary spectrum and returns you to main menu.

Note, that none of the other options of the main menu will be active, if you have not specified the data.

Submenu 2. Spectrum graphics

The program produces graphic outputs of the main components of the rotary spectrum that has been estimated.

Select option 2 of the main menu and press ↵. The screen now shows:



```
ROTARY GRAPHICS CHOICES

1. Clockwise spectrum
2. Anticlockwise spectrum
3. Clockwise & Anticlockwise spectrum
4. Clockwise Anticlockwise & Total spectrum
5. Ellipse stability
6. Ellipse orientation
0. Exit

Enter your choice:
```

Select a number between 1 and 6 and press ↵.

Following that, a number of questions such as "Linear or Logarithmic spectrum", "X-axis: Channel(1) or Frequency(2)", etc. will appear on the screen. Refer to SPECT user's guide for detailed description of those questions. If options 3 or 4 have been selected it is advised to use different pen numbers for each of the parameters plotted in order to distinguish them on the graph (see example in Figure 1).

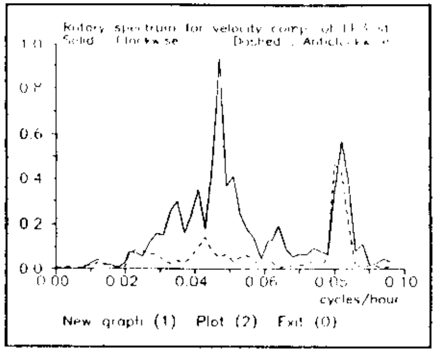


Figure 1 : ROT graphics

Submenu 3. Display results

Select option 3 of main menu and press ↵. The screen now shows:

DISPLAY RESULTS

To exit answer 0 to any of the first three questions

Screen (1) or File (2) (default=1):

ROT spectrum results may be sent to screen or to a disk file. A series of questions such as the following will appear:

Linear (1) Log (2) spectrum (default=1):
Full output (y,n) (default=n)
First channel:
Last channel:
Input file name for results:
Job title:

(see SPECT user's guide for details)

The rotary spectrum results as viewed on the screen or sent to a disk file consist of:

a. Four lines of comments that identify the run case and general signal characteristics.

First line: - Job title for 'File option'.

Second line: - Number of points initially read (NMAX) for the cross spectrum calculations (through CROSS program)
- Number of points on which the cross spectrum calculation has been based (N ≤ NMAX) (through CROSS program)
- Number of segments of the cross spectrum calculation(NSP)
- Segment length (window size - NT) of the cross spectrum calculation

Third line: - The total number of channels the output contains
- The format of the seven - column output
- The maximum total energy of the rotary spectrum

Fourth line: - The reference titles for each column

b. Next follows the seven column output which contains the Channel number, Clockwise spectrum, Anticlockwise spectrum, Total spectrum, Rotary coefficient, Ellipse orientation and Ellipse stability.

After displaying the results on the screen or saving them on a file, the program returns to the main menu

To exit ROT, type 0 and press ↵.

SEASPECT PACKAGE

TIME SERIES MANIPULATION PROGRAM **SERIES**

TABLE OF CONTENTS

1. PROGRAM CHARACTERISTICS

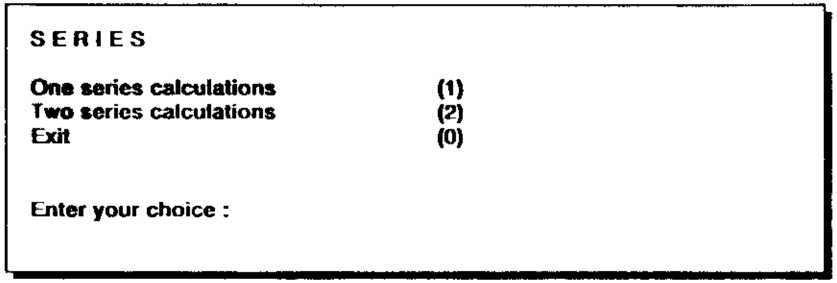
Program SERIES is a program used to perform a certain number of operations on one or two time series such as adding a constant on each element of a time series or summing up two time series etc. No graphic utilities are available on this program. Refer to program PROJ for graphic display of time series.

Program limitations.

- Time series length ≤ 10000 points.

2.USING *SERIES*

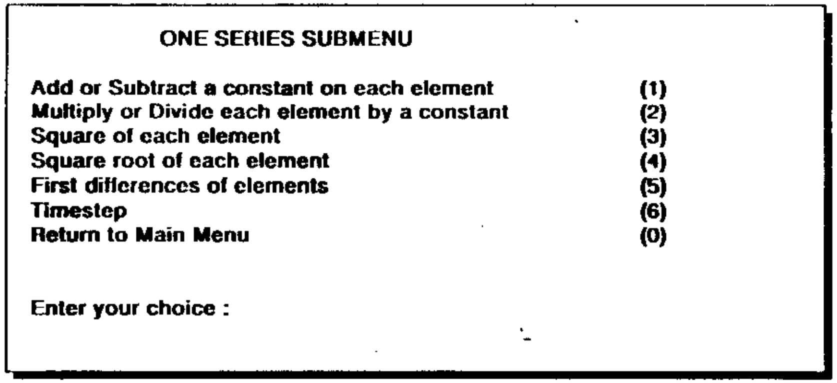
To run the program select "5" from the SEASPECT menu and press ↵. Information about the program, appears. By pressing ↵, you enter the main menu:



```
SERIES

One series calculations          (1)
Two series calculations          (2)
Exit                             (0)


Enter your choice :
```

The cursor is now next to 'Enter your choice'. To begin using the program, select "1" or "2" and press ↵.

If you have selected "1" the following will appear on your screen:



```
              ONE SERIES SUBMENU

Add or Subtract a constant on each element       (1)
Multiply or Divide each element by a constant    (2)
Square of each element                           (3)
Square root of each element                      (4)
First differences of elements                    (5)
Timestep                                         (6)
Return to Main Menu                              (0)


Enter your choice :
```

Option 1 : With this option a positive/negative number will be added algebraically to each element of the time series.

Option 2 : Each element of the time series read will be multiplied by the specified number.

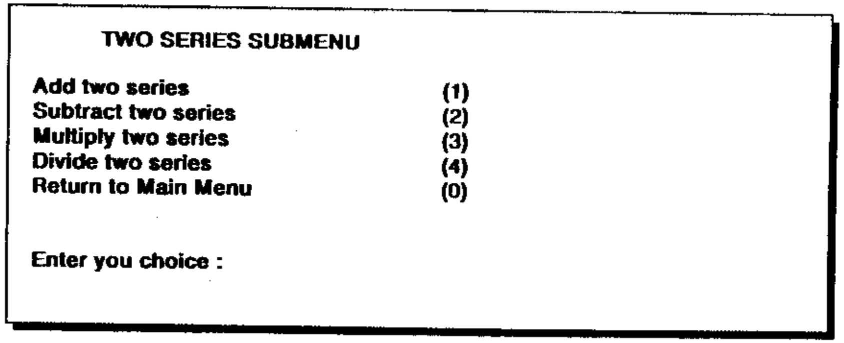
Option 3 : Each element of the time series will be multiplied by itself.

Option 4 : The square root of each element will be computed and written on a specified file. If the time series contains negative numbers a warning message will inform the user that the program cannot proceed further with the desired operation.

Option 5 : Each element is replaced by its difference from its previous element in the time series. This procedure is called "pre-whitening" and is often used on time series with a strongly "red" spectrum. The spectrum of the pre-whitened time series contains stronger energy peaks at higher frequencies which can therefore be better put in evidence.

Option 6 : This option is used to subsample a file by selecting elements "n" points apart. You have to specify the first point to be selected and the step "n". This option is useful, for example, in transforming a time series containing data sampled at 10 minutes interval to hourly values. Attention should be given, at this point, in eventual aliasing of the spectrum. It is recommended to first filter adequately the time series and then proceed to its subsampling.

If you have selected "2" , the following will appear on your screen :



```
        TWO SERIES SUBMENU

Add two series                 (1)
Subtract two series            (2)
Multiply two series            (3)
Divide two series              (4)
Return to Main Menu            (0)


Enter you choice :
```

Options (1) and (2) are used to add or subtract, respectively, the elements of two time series. The first element of the second file will be added (subtracted) to (from) the first element of the first file, etc. Suppose a file A contains sea-pressure data, while a file B contains the corresponding atmospheric pressure data. This option may be used to compute the sea level by subtracting file B from A.

Options (3) and (4) are used to multiply or divide, respectively, the elements of two time series. In option (4) if the second file contains zeroes a warning message notifying the user that the program cannot proceed further will appear.

In all options of program SERIES reading and writing procedures are the same as in program PROJ.

To exit SERIES, type 0 and press ↵.

SPECTRAL ANALYSIS OF DISCRETE SIGNALS

by John E. Daskalakis
and A. Lascaratos

February 1991

TABLE OF CONTENTS

1. Discrete Fourier Transform (DFT)

1.1. Definitions

Let us consider a discrete signal consisting of N real or complex values x_i, $i = 0, 1, \dots, N-1$. The discrete Fourier transform (DFT) is by definition the complex discrete signal X_k defined by the relation

$$X_k = \sum_{i=0}^{N-1} x_i \exp(-j2\pi ik/N) \qquad \text{for } k \geq 0 \tag{1}$$

We often write $W_N = \exp(-j2\pi/N)$ and the DFT as

$$X_k = \sum_{i=0}^{N-1} x_i W_N^{ik} \qquad \text{for } k \geq 0$$

This relation may be inverted and the x_i can be expressed as functions of the X_k by the following relations

$$x_i = \frac{1}{N} \sum_{k=0}^{N-1} X_k \exp(j2\pi ik/N) \qquad \text{for } i \geq 0 \tag{2}$$

The above relation defines also the inverse discrete Fourier transform. The DFT defines generally a sequence of complex numbers with a real and an imaginary part or a magnitude $|X_k|$and a phase ϕ_k.

One can easily prove the following elementary properties of the DFT

1. The DFT is a linear operation

2. Periodicity property
   $X_{N+k} = X_k$ for $k \geq 0$
   so that values of X_k are useful only for $0 \leq k \leq N-1$
   (as for the initial signal). The initial signal is thus regarded as an implicitly periodic signal of period N.

3. Symmetry properties for the DFT of a real signal
   $X_{N-k} = X_k^*$ and $|X_{N-k}| = |X_k|$ for $k \geq 0$
   or written around the midpoint of the sequence
   $X_{N/2+k} = X_{N/2-k}^*$ and $|X_{N/2+k}| = |X_{N/2-k}|$ for $k \geq 0$

   meaning that the real part of X_k is an even function and the imaginary part of X_k is an odd function of the index k with its central value at $N/2$ (or two central values if N is odd).

4. Parseval Theorem relating x_i and X_k

$$\sum_{i=0}^{N-1} |x_i|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X_k|^2$$

<u>Example 1</u>

A very simple result can be obtained when the series has only two points x_0 and x_1. We find easily applying formula (1) and taking into account that
$W_2 = -1$
$X_0 = x_0 + x_1$
$X_1 = x_0 - x_1$

<u>Example 2</u>

If $N = 3$ the four value series x_i transforms into the following DFT

$$X_0 = x_0 + x_1 + x_2$$

$$X_1 = x_0 + x_1W_3^1 + x_2W_3^2$$

$$X_2 = x_0 + x_1W_3^2 + x_2W_3^4$$

where $W_3 = -\sqrt{3}/2 - j(1/2)$

<u>Example 3</u>

$x_i = e^{-ai}$ with constant a and $i = 0,1,2,\dots,N-1$.
We can analytically calculate X_k using directly relation (1) and a geometric progression formula
$X_k = (1-e^{-aN}))/(1-e^{-a}e^{-2j\pi k/N})$ $\qquad k = 0,1,2,\dots,N-1$

Example 4

$x_i = \sin(a \cdot i)$ with constant a and $i = 0,1,2,\dots,N-1$.

We can analytically calculate X_k using directly relation (1) and a geometric progression formula

$X_k = (1-e^{jaN})/2/j/(1-e^{j(a-2j\pi k/N}) - (1-e^{-jaN})/2/j/(1-e^{-j(a+2j\pi k/N})$

$k = 0,1,2,\dots,N-1$

1.2. Fast Fourier Transform (FFT)

Using equation (1) to calculate the DFT X_k we observe that a considerable number of calculations of sine and cosine functions is involved. Even if we interpret the formula in terms of integer powers of the basic function $W_N=e^{-2\pi j/N}$ the number of multiplications needed is of the order of N^2. We thus realize that even on a powerful computer a straightforward computation of the DFT becomes impractical as soon as the time series grows over 1000 points.

The Fast Fourier Transform is an algorithmic method to improve the speed of computation of the DFT. One of the most popular FFT algorithms is the Cooley-Tukey radix 2 decimation-in-time algorithm. The key to the development of this FFT algorithm is the fact that the sequence X_k can be obtained by a suitable combination of the results of two transforms each of length $N/2$, and the combination requires only N multiplications. Assuming the series length N is a power of 2 the decomposition process can be repeated a number L of times, $L = \log_2 N$. Basically, we could first form $N/2$ transforms of length 2, combine them to get $N/4$ transforms of length 4, combine these to get $N/8$ transforms of length 8, and so on until after L steps one transform of length N is obtained. Only the first stage (transforms of length 2) requires the calculation of the basic functions W_k, but as $W_2 = e^{-j\pi/2} = -1$ only addition and subtraction are needed. In the process of combining half length transforms to obtain full length transforms only N multiplications are required. As a result the total number of multiplications is $N \cdot L = N \cdot \log_2 N$.

Furthermore the calculation can be done in place, i.e. no extra storage is required if the input sequence can be erased and its DFT written on it instead.

Comparing this algorithm to the direct method we obtain for a series of length $N = 1024$ approximately a 100 fold reduction in computation time. This method is obviously applicable only to time series with lengths equal to integer powers of 2 (2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384,....).

The continuing improvements in digital technology have made it possible to construct efficient and cost effective FFT "boxes". These are hardware dedicated to the computation of FFT either as standalone units or as attachments to general purpose computers. Also many of the newer oscilloscopes have incorporated the capabilities of digital processing of the signals and contain hardware to perform the FFT algorithm.

In many applications the design of dedicated FFT hardware is optimized to permit very high throughput rates to meet the high processing speed required in these applications.

In the FFT algorithm one can also detect various levels of inherent computational parallelism that can be exploited to obtain a wider range of performance. We note thus the existence of pipeline FFT.

2. Signals and Fourier transforms

2.1. Classification of signals

The signals that can be found in the physical world or can be constructed in a modeling mathematical environment can be classified into two broad classes: deterministic and random.

Deterministic signals can be characterized as periodic or non-periodic and the definitions here are obvious.

The random signals can belong into the stationary or non-stationary family and we will describe briefly the stationarity property by giving some definitions pertaining to random signals.

A single time history representing a random phenomenon is called a sample function or sample record. The collection of all possible sample functions which the random phenomenon might have produced is called a random process or stochastic process. Hence a sample record of data for a random physical phenomenon may be thought of as one physical realization of the random process.

For a random process, given by all its physical realizations $x_k(t)$, we can calculate at least two interesting values

$$\mu(t_1) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} x_k(t_1)$$

$$R(t_1, t_1+\tau) = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} x_k(t_1)\; x_k(t_1+\tau)$$

If these two characteristics do not vary as t_1 varies, the random process is said to be weakly stationary.

An infinite collection of moments and joint moments could be calculated in a similar way. If these moments are time invariant we say that the process is strongly stationary.

The stationary processes can further be classified into ergodic and non ergodic processes where the ergodicity is a notion applicable only to stationary processes.

To define the property of ergodicity we introduce the moment and joint moment calculation by considering only one physical realization of the signal and performing time averaging.

$$\mu_x(k) = \lim_{T\to\infty} \frac{1}{T}\int_0^T x_k(t)\; dt$$

$$R_x(\tau,k) = \lim_{T\to\infty} \frac{1}{T}\int_0^T x_k(t)\; x_k(t+\tau)\; dt$$

For an ergodic process these time averages in fact do not depend on the realization k and yield exactly the same values as the ensemble (statistical) averages calculated above.

Ergodic processes are clearly an important and nice class of random processes since all their properties can be deduced from a single time history performing time averaging.

Frequently, and fortunately, in practice our random data are or can be considered as both stationary and ergodic and consequently we can work on a single realization of the signal, viz. the only measured realization of the signal.

2.2. *Fourier expansions*

2.2.1. Periodic data can be expanded in Fourier series

If we consider a signal $x(t)$, periodic with period T and fundamental frequency $f_1 = 1/T$ we can write

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_1 t) + b_n \sin(2\pi n f_1 t)$$

where the coefficients a_n and b_n are given by the expressions

$$a_n = \frac{2}{T}\int_0^T x(t)\cos(2\pi n f_1 t)\,dt \qquad n = 0,1,2,..$$

$$b_n = \frac{2}{T}\int_0^T x(t)\sin(2\pi n f_1 t)\,dt \qquad n = 1,2,..$$

The complex spectrum of the periodic signal is

$X_n = a_n + j\,b_n$

and obviously is discrete. Usually the magnitude of X_n is plotted against the harmonic or channel number n.

2.2.2 Transient nonperiodic data can be represented by their Fourier transform.

The well known Fourier transform of a function $x(t)$ is given by

$$X(f) = \int_{-\infty}^{+\infty} x(t)\, e^{-j2\pi ft}\, dt$$

and the original signal can be written as

$$x(t) = \int_{-\infty}^{+\infty} X(f)\, e^{j2\pi ft}\, df$$

The meaning of this decomposition is well known as it consists in representing the arbitrary signal $x(t)$ as an infinite sum of sinusoidal (harmonic) components. The spectrum is here a continuous function of frequency f.

2.2.3. Finite Fourier Transform

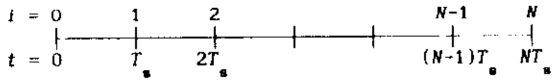
In practice we do not have the possibility to observe and integrate the signal on an infinite time span and we are thus restricted to a more physical approach involving a time span T, say from t=0 to t=T.

Consequently we define the finite Fourier transform of the analog signal $x(t)$ by

$$X(f,T) = \int_0^T x(t)\exp(-j2\pi ft)\,dt \qquad (3)$$

where T is the observation time of the signal, i.e. the time during which the signal has been observed and its values recorded.

Let us now sample the analog signal $x(t)$ with a sampling period T_s obtaining the discrete signal $x_i = x(t_i) = x(iT_s)$ for $i \geq 0$, on a total observation time of $T = N \cdot T_s$. ,



We will assume also that the observation time T is sufficiently large that the signal repeats itself for $t \geq T$ (i.e. all the information is contained in the laps of time T). For example its value at $t = N \cdot T_s$ is supposed to be the same as its value at t = 0.

If we approximate the integral of relation (3) with a finite sum we get

$$X(f,T) = \int_0^T x(t)\, e^{-j2\pi ft}\, dt \propto \sum_{i=0}^{N-1} x_i e^{-j2\pi f T_s i}\, T_s$$

For the frequencies $f = f_k = \frac{k}{NT_s} = \frac{k}{T} \qquad k = 0,\ 1,\ \ldots,\ N-1$

we have $X(f_k,T) \simeq T_s \sum_{i=0}^{N-1} x_i \exp(-j2\pi ik/N) = T_s X_k$

where X_k is the DFT of signal x_i.

Consequently $X_k = \frac{1}{T_s} X(f_k,T) \qquad k = 0,1,\ldots,N-1 \qquad (4)$

This last expression (4) determines the relation between DFT and finite Fourier transform.

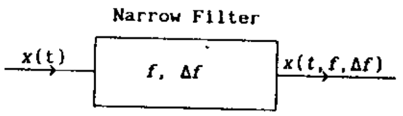
In what follows we will exhibit the relation between finite Fourier transform and the power spectrum of a signal.

3. Power spectrum and DFT

3.1. Definitions

We have already defined the DFT and the finite Fourier transform. We still do not know what the power spectrum of a signal is. Let us clarify first that the terms power spectrum, power spectral density and autospectrum are used interchangeably to designate the same concept.

By definition the power spectral density of signal $x(t)$ is the distribution in frequency of its mean square value and we identify the mean square value of a signal as its energy. If we imagine the signal $x(t)$ going through a narrow band filter with frequencies f and $f+\Delta f$



and designate by Ψ the energy of the output signal we can define the power spectral density by the following relations

$$\Psi = \lim_{T\to\infty} \frac{1}{T}\int_0^T x(t,f,\Delta f)^2\, dt \approx G_{xx}(f)\, \Delta f$$

or $G_{xx}(f) = \lim_{\Delta f\to 0} \frac{\Psi}{\Delta f}$

<u>Examples</u> <u>of</u> <u>signal</u> <u>PSDs</u>

Constant	$x(t) = c$	$G_{xx}(f) = c^2\, \delta(f)$
Harmonic signal	$x(t) = c\, \sin(2\pi f_0 t+\theta)$	$G_{xx}(f) = c^2/2\ \delta(f-f_0)$
White Noise	$x(t) = ?$	$G_{xx}(f) = c$

Looking for the relation between the power spectral density as defined above and the finite Fourier transform we can prove, and this is not obvious at all, (cf. Bendat and Piersol) that

$$G_{xx}(f) \approx \lim_{T\to\infty} \frac{1}{T}\int_0^T x(t,f,\Delta f)^2\, dt \,/\, \Delta f = 2 \lim_{T\to\infty} \frac{1}{T}\, E\, [|X(f,T)|^2] \qquad (5)$$

where $X(f,T)$ is the finite Fourier transform of signal $x(t)$ and E the expected value (statistical) operator for a random variable.

We have till now defined the power spectral density (PSD) and we have related it to the finite Fourier transform. We have one more step toward the DFT and this is the following.

When we dispose of only one observation $x(t)$ of finite length T, the only possible approximation for $G_{xx}(f)$ is

$$\hat{G}_{xx}(f) = \frac{2}{T}\,|X(f,T)|^2 \qquad (6)$$

where the expected value has been replaced by the unique measurement available on time span [0,T].

For $f = f_k = k/(NT_s)$ we will have $\hat{G}_{xx}(f) = \frac{2}{T}\,|X(f_k,T)|^2$

$$\hat{G}_{xx}(f) = \frac{2}{T}\,T_s^2\,|X_k|^2 = \frac{2T_s}{N}\,|X_k|^2 \qquad (7)$$

Relation (7) gives us the capability of estimation of the power spectrum of $x(t)$ using the DFT of $x(t)$. We note that we can evaluate the PSD only at frequencies equal to discrete f_k. This is the price paid for the loss of information consequent to signal sampling.

3.2. *Signal energy and power spectrum*

For the discrete signal x_i, i = 0,1,..N-1 we define the mean energy as equal to its mean square value

$$E_m = \frac{1}{N}\sum_{i=0}^{N-1} x_i^2$$

If the mean value of the signal is subtracted from all its values the previous definition identifies the mean energy as equal to the variance of the signal

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N-1} (x_i-\bar{x})^2$$

Note

We draw the attention of the reader to the following point. The above estimation of the variance of the time series is based on a given sample of the time series and is a biased estimate. We mean by biased estimate that if the experiment could be repeated a number of times and we could thus calculate the expected value of the variance $E[\sigma^2]$ we would not find the true value but instead (N-1)/N times the true value. A better, unbiased, estimate of the true value of the variance of the time series is thus

$$\sigma^2 = \frac{1}{N-1}\sum_{i=0}^{N-1} (x_i-\bar{x})^2$$

This is also the definition used in the SEASPECT package.

Say X_k is the DFT of the signal x_i. From Parseval's theorem we note that

$$\sum_{i=0}^{N-1} x_i^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X_k|^2$$

So, for the mean energy we get

$$E_m = \frac{1}{N}\sum_{i=0}^{N-1} x_i^2 = \frac{1}{N^2}\sum_{k=0}^{N-1} |X_k|^2$$

Let $\hat{G}_{xx}(f_k) = (2T_s/N)\;|X_k|^2$ be an estimation of the power spectrum of the signal $x(t)$.

Integrating the power spectral density $\hat{G}_{xx}(f)$ we get the total energy of the signal

$$E = \int_0^{T/2} G_{xx}(f)df = \frac{1}{2}\int_0^{T} G_{xx}(f)df \quad \text{(due to symmetry)}$$

Replacing the spectral density $G_{xx}(f)$ with its estimation $\hat{G}_{xx}(f_k)$ and the integral with a sum we get

$$E_m = \frac{1}{2}\sum_{k=0}^{N-1}\hat{G}_{xx}(f_k)\Delta f_k = \frac{1}{2}\sum_{k=0}^{N-1}\frac{2}{T_s N}|X_k|^2\frac{1}{T_s N} = \frac{1}{N^2}\sum_{k=0}^{N-1}|X_k|^2$$

Finally

$$E_m = \frac{1}{N}\sum_{i=0}^{N-1}x_i^2 = \frac{1}{N^2}\sum_{k=0}^{N-1}|X_k|^2 = \frac{1}{2}\sum_{k=0}^{N-1}\hat{G}_{xx}(f_k)\Delta f \quad \text{with } \Delta f = 1/(N\cdot T_s) \qquad (8)$$

Expression (8) relates the PSD of the signal to its mean energy.

4. Power spectrum estimation

For the power spectrum estimation we will make use of formula (7) and of the DFT of signal x_i.

Using the Fast Fourier Transform, as it is almost always the case in order to calculate the DFT X_k, we may face a problem as the series length is arbitrary and not necessarily an integral power of 2. Hence the time series must either be truncated or have zeros added.

In these cases we resort to what is called a zero padding, i.e. we extend the time series with zero values up to the next integral power of 2. For example if for x_i n=900, we form a new series x'_i with 1024 points by adding 124 zeros at the tail of the initial series. We may then use the FFT on the new series x'_i to estimate its power spectrum X'_k. It can be proven (cf. Bendat) that the two DFT's X_k and X'_k under most circumstances do not differ notably. We can spot the main difference in the frequency resolution: as the spectrum frequencies are $f_k = k/(N\cdot T_s)$, in the case of zero padding the resolution increases artificially due to a higher N.

Let us write for instance the finite Fourier transform on which square the PSD is based as

$$X(f,T) = \int_0^T x(t)\, e^{-j2\pi ft}\, dt = \int_0^T y(t)\, u_T(t) e^{-j2\pi ft}\, dt$$

where $y(t)$ is the real (physical) signal from which $x(t)$ resulted by observation between times 0 and T, and $u_T(t)$ is a boxcar function equal to 1 from 0 to T and null elsewhere.

We repeat here that our main goal is to estimate the power spectral density of the random process $y(t)$ from which the realization $x(t)$ or times series x_i originated. Thus the bias introduced by the abrupt truncation of the original random process between times 0 and T must be compensated.

The boxcar weighting causes "leakage" by spreading the main lobe of the PSD function and by adding an infinite number of smaller side lobes. A smoother than the boxcar tapering of the function is thus necessary to provide for less distortion of the real signal function.

Hence we usually taper the realization $x(t)$ by multiplying its values by a function which is equal to 1 to the middle of the time series and decays smoothly to 0 toward the two ends of the time interval. This tapering is called windowing and the tapering function is called a window.

On the other hand it is obvious that for a better estimation of the expected value of the power spectral density of the random process one must calculate a mean value of $|X(f,T)|^2$ for the largest possible number of realizations of the signal $x(t)$. But when there is available only one observation of duration T, the only way to proceed is to break down this unique observation into T/N_T smaller observations (called segments) of length N_T and take for each one of these segments one approximation of the power spectrum. The final estimation of $G_{xx}(f)$ will consist of the arithmetic mean value of these estimations. Also each segment may be windowed for the reasons explained above. This method of averaging segment FFTs is due to Welch and is frequently called the method of modified periodograms.

The segment tapering mentioned above may be done with various windowing functions as the only requirements for them are to be symmetric with a maximum value of approximately 1 at the middle of the segment and smooth decaying toward the ends of the interval. We use mainly a triangular window (usually called Parzen window) with values given by

$$w_j = 1 - |j - \frac{n+1}{2}| / \frac{n+1}{2} \qquad 1 \le j \le n \qquad (9)$$

or a sinusoidal bell-shaped window (named Hamming window) of the following form

$$w_j = 0.54 - 0.46 \cos(2\pi\frac{j-1}{n-1}) \qquad 1 \le j \le n \qquad (10)$$

In expressions (9) and (10) n is the segment length (power of 2).

As we window each segment by the Parzen or Hamming window an adjustment is necessary in order to have an optimum PSD estimator. This adjustment consists in dividing the PSD values of each segment by the energy of the window, i.e. its sum of squares.

Using this segment averaging procedure to estimate the power spectrum, and in order to assure the continuity of successive estimations stemming from individual segments, we must "link" in some way the adjacent segments. It can be proven that if adjacent segments have an overlap of 50% a minimal distortion of the final power spectrum estimation is obtained. Thus if N_T is the segment length (power of 2) and N_s the number of segments we arrive at the equation

$$N = N_T + (N_s-1)\ (N_T/2) \le N_{max} \qquad (11)$$

where N is the length of the series on which the power spectrum estimation will be based and N_{max} the available total series length.

Example: Consider a time series of 800 points and choose a segment of 256 points. From above equation we find N_T = 256 and N_s = 5 and N = 768.

Frequency smoothing

Comparable to the segment averaging just explained, we note that a frequency smoothing could also be done in order to estimate the PSD. In that case we loose some frequency resolution because we combine adjacent frequencies. In the SEASPECT package there is no frequency smoothing option available.

Finally let us note that when we estimate the PSD from many segments we do not pad with zeros the time series. If we added zeros we should pad each segment with a proportionate number of zeros in order to maintain the same number of degrees of freedom for the resulting estimation of the PSD (see below paragraph 5). Instead we truncate the time series to the number $N \le N_{max}$ given by formula (11) above.

Computational note

We conclude this section with a brief discussion of a method of performing FFT on real sequences. In most practical cases the data sequence x_i is real and doing a N point DFT on it will require $N \cdot \log_2 N$ complex multiplications, i.e. the same number of operations as if it was a complex sequence. We will show now that it is possible to do a N point DFT to transform simultaneously two N point real sequences.

Consider the real sequences x_n and y_n. We form a complex sequence

$$a_n = x_n + j\, y_n$$

and take its DFT X_k.

Calculate now

$$A_k = X_k + j\, Y_k$$
$$A_{N-k} = X_{N-k} + j\, Y_{N-k} = X_k^* + j\, Y_k^*$$

Combining these two expressions we get the DFT's of x and y as follows

$$X_k = (A_k + A_{N-k}^*)/2 \qquad (12)$$

$$Y_k = (A_k - A_{N-k}^*)/2j \qquad (13)$$

Therefore to obtain the DFT of the two N point real sequences x_n and y_n we can do a N point DFT on the complex sequence a_n and then use the above relations to get back the individual transforms X_k and Y_k.

PSD estimation recipe

Let us sum up now how we estimate the PSD of a real signal in the SEASPECT package.

1. Define the segment length, the kind of window (Parzen or Hamming) that will be used and calculate the operational series length N given by formula (11).

2. Load two adjacent segments into a running (dummy) complex sequence a_n. Multiply this sequence by the window.

3. Take the FFT of this complex sequence.

4. Extract the FFTs of the two real sequences and divide by the energy of the window to compensate for its effects.

5. Add the squared values to a running sum PSD and repeat the process until no more segments.

6. When the process is over divide the PSD by the number of segments to get the mean peiododgram.

5. Power spectrum confidence interval

Let us try to estimate the true value of $G_{xx}(f)$ by an estimate $\hat{G}_{xx}(f)$ based on only observation of the signal. We have using relation (5)

$$G_{xx}(f) = 2 \lim_{T \to \infty} \frac{1}{T} E \{|X(f,T)|^2\}$$

and

$$\hat{G}_{xx}(f) = 2 \; |X(f,T)|^2$$

In fact as

$$|X(f,T)|^2 = X_R(f,T)^2 + X_I(f,T)^2$$

and if we consider X_R and X_I as gaussian random variables it appears that the estimation

$$\hat{G}(f,T) = 2 \frac{1}{T} \; |X(f,T)|^2$$

follows the statistical estimation

$$\frac{\hat{G}(f,T)}{G(f,T)} = \frac{\chi_2^2}{2} \qquad (14)$$

where $G(f,T)$ is the true value of the PSD at frequency f and χ_2^2 is the χ^2 statistical distribution with 2 degrees of freedom (d.o.f.).

We divert here and note, in relation to above expression (14), that a random variable follows the χ_n^2 distribution with n degrees of freedom when it is defined as the sum of squares of n independent gaussian random variables each with zero mean and unit variance. Also, its mean is n and its variance is $2 \cdot n$.

Taking the variance of relation (14)

$$\frac{\sigma[\hat{G}(f,T)]}{G(f,T)} = \frac{\sqrt{2 \cdot n}}{2} = \frac{\sqrt{2 \cdot 2}}{2} = 1$$

Hence we realize that the estimation of the expected value from only one observation of signal $x(t)$ has a standard error equal to one (100% error) or otherwise that the real value of $G_{xx}(f)$ lies between 0 and $2 \cdot \hat{G}_{xx}(f)$.

$0 \le G_{xx}(f) \le 2 \cdot \hat{G}_{xx}(f)$

Referring to the power spectrum estimation procedure outlined in paragraph 4 one can show that the averaging of q adjacent overlapping segments

$$G_{xx}(f) = \frac{\hat{G}^{(1)}_{xx}(f) + \hat{G}^{(2)}_{xx}(f) + \ldots . + \hat{G}^{(q)}_{xx}(f)}{q} \qquad (q = n_s)$$

leads to an estimation of the power spectrum which follows a χ^2 statistical distribution with d degrees of freedom where d is approximately equal to the double of the number of segments, $d = 2 \cdot n_s$, with roughly each segment contributing 2 d.o.f., one from its real part squared and the other from its imaginary part squared.

The theoretical value of the spectral density at frequency f will lie then in the confidence interval

$$\hat{G}_{xx}(f)\frac{d}{\chi^2_{d,\alpha/2}} \le G_{xx}(f) \le \hat{G}_{xx}(f)\frac{d}{\chi^2_{d,1-\alpha/2}} \qquad (15)$$

with probability α, where $\chi^2_{d,\alpha/2}$ and $\chi^2_{d,1-\alpha/2}$ are the values of χ^2 distribution with d degrees of freedom and probability threshold $\alpha/2$ and $1-\alpha/2$.

We will designate these error bounds by B_{min} and B_{max}

$$B_{min} = \frac{d}{\chi^2_{d,\alpha/2}} \quad \text{and} \quad B_{max} = \frac{d}{\chi^2_{d,1-\alpha/2}}$$

and we note that they are usually calculated for α=0.05, or at the 95% confidence level.

Example

Say $n_s = 10$ so that $d = 2 \cdot n_s = 20$ and assume $\alpha = 0.05$.
From the χ^2 distribution tables
$\chi^2_{20,0.025} = 34.17$ and $\chi^2_{20,0.975} = 9.59$
so that $B_{min} = 0.585$ and $B_{max} = 2.086$

6. Spectral analysis of two signals

With the previously exposed univariate spectral analysis we can study the harmonic components present in two independent random processes. The bivariate or cross spectral analysis of these two signals will strive to answer questions about the relation between the harmonic components of the two signals at the same frequency. If we may represent the studied signals as input and output to a physical process the cross spectral analysis will parallel the study of the transfer function of the process.

Consider first two sequences of discrete data x_i and y_i. The correlation coefficient r between these two series is defined as

$$r = \frac{\sum_i (x_i-\bar{x})\ (y_i-\bar{y})}{\sigma_x \sigma_y\ (n-1)}$$

where σ_x and σ_y are the standard deviations of the two series of data and n is the number of joint observations.

Let us consider now two random processes $x(t)$ and $y(t)$. We can correlate them at two different times t_1 for the first and t_2 for the second signal using the usual statistical tool of correlation coefficient. The cross-correlation function results as the correlation coefficient when times t_1 and t_2 are τ time apart ($\tau = t_2 - t_1$).

Imitating the above definition of the correlation coefficient we can define an estimate for the cross-correlation function of values of $x(t)$ at time t and $y(t)$ at time $t+\tau$ by taking the average product of the two sets of values over the observation time T (assuming that the two random processes have zero mean or that the mean has been subtracted from the values). The resulting average product will approach the exact cross-correlation function as T approaches infinity.

Hence we define

$$R_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_0^T x(t)\ y(t+\tau)\ dt$$

When $y(t) \equiv x(t)$ we speak about the autocorrelation function of signal $x(t)$.

Elementary properties of the cross-correlation and autocorrelation functions are

$$R_{xy}(-\tau) = R_{yx}(\tau)$$

$$|R_{xy}(\tau)|^2 \leq R_{xx}(0)\ R_{yy}(0) \qquad \text{(Schwartz inequality)}$$

$$|R_{xy}(\tau)| \leq \frac{1}{2}\ [R_{xx}(0) + R_{yy}(0)]$$

When $R_{xy}(\tau) = 0$ for all times τ, signals $x(t)$ and $y(t)$ are statistically independent (assumed having zero mean values).

The analog cross-correlation function has its discrete counterpart defined as

$$R_{xy}(k) = \frac{1}{N-k} \sum_{i=1}^{N-k} x_i\ y_{i+k}$$

for the cross-correlation function and as

$$r_{xy}(k) = \frac{R_{xy}(k)}{\sqrt{R_{xx}(0)\ R_{yy}(0)}}$$

for the cross-correlation coefficients with

$-1 \leq r_{xy}(k) \leq 1$

Definition of the cross PSD

The cross power spectral density $G_{xy}(f)$ of two signals is now defined as the Fourier transform of the cross-correlation function. As such the cross PSD gives the distribution in frequency of the correlation of the two processes.

We can also prove another important result closely related to the above definition : the PSD of one signal is the Fourier transform of the autocorrelation function.

An important inequality is produced by transforming Schwartz inequality in the frequency domain

$$|G_{xy}(f)|^2 \le G_{xx}(f)\ G_{yy}(f)$$

or alternatively

$$\gamma^2_{xy} = \frac{|G_{xy}(f)|}{G_{xx}(f)\ G_{yy}(f)} \le 1 \qquad (16)$$

defining thus the coherence function $\gamma_{xy}(f)$ of two signals.

Fundamentals of input/output relationships for physical systems

Consider two analog signals $x(t)$ and $y(t)$ one as the input and the other as the output of a physical system. If the system is linear with constant parameters (CPLS) we know how the input and output are related by the impulse response function h(τ)and the convolution integral

$$y(t) = \int_0^{+\infty} h(\tau)\ x(t-\tau)\ d\tau$$

(Note that the integral does not extend into negative time values as h(τ) is considered as the impulse response of a physically realizable linear system, $h(\tau)=0$ for $\tau<0$).

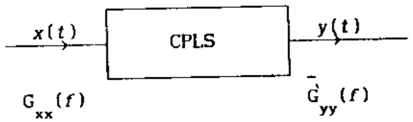
By transcribing the above relation into the frequency domain by Fourier transform we define the complex transfer function $H(f)$ of the system by the relation

$$Y(f) = H(f)\ X(f)$$

where

$$H(f) = \int_0^{+\infty} h(\tau)\ e^{-j2\pi f\tau}\ d\tau$$

Let us examine how is the PSD of the input signal transformed when the signal passes through the CPLS.



We can prove the following relations

$$G_{yy}(f) = |H(f)|^2\ G_{xx}(f)$$

$$G_{xy}(f) = H(f)\ G_{xx}(f)$$

Coherence and CPLS

The coherence function value at frequency f, $\gamma^2_{xy}(f)$, can be interpreted as the fractional portion of the mean square value of the output $y(t)$ which is contributed by the input $x(t)$ at frequency f. Conversely the quantity $1-\gamma^2_{xy}(f)$ is a measure of the mean square value of $y(t)$ not accounted for by $x(t)$ at frequency f.

For two signals that are linearly dependent being input and output of a CPLS we can easily prove that the coherence function is unity at all frequencies

$$\gamma^2_{xy}(f) = 1$$

On the opposite side if $x(t)$ and $y(t)$ are completely unrelated the coherence function will be zero.

When the coherence is less than 1 and greater than 0, one or more of three possible situations exist

(a) Noise is present in the measurements

(b) The system relating $x(t)$ and $y(t)$ is not linear

(c) $y(t)$ is an output due to $x(t)$ as well as to other inputs.

Estimation of the cross PSD

Following the same procedure with the estimation of the power spectrum of one signal, we can estimate the cross spectrum (cross PSD) of two stationary random processes $x(t)$ and $y(t)$ as

$$G_{xy}(f) = 2 \lim_{T\to\infty} \frac{1}{T} E\,[|X^{*}(f,T)\;Y(f,T)|^{2}]$$

The cross power spectrum is obviously a complex number and writing

$$G_{xy}(f) = C_{xy}(f) + j\cdot Q_{xy}(f)$$

we define $C_{xy}(f)$ as the cospectrum (or coincident spectrum) and $Q_{xy}(f)$ as the quad spectrum (or quadrature spectrum) of the two signals. We may also write

$$G_{xy}(f) = |G_{xy}(f)|\;\exp(j\phi_{xy}(f))$$

where $|G_{xy}(f)|$ is the magnitude and $\phi_{xy}(f)$ the phase of the cross power spectrum.

From one observation of the two signals $x(t)$ and $y(t)$ we can have the following estimation using the DFTs of $x(t)$ and $y(t)$

$$\hat{G}_{xy}(f) = \frac{2}{T}\,X^{*}(f,T)\;Y(f,T)$$

and for frequencies $f = f_k = k/T$ we will have

$$\hat{G}_{xy}(f) = \frac{2}{T}\,T_s X^{*}_k\;T_s Y_k = (2T_s/N)\;X^{*}_k\;Y_k = \hat{C}_{xy}(f) + j\cdot\hat{Q}_{xy}(f)$$

The coherence coefficient of the two signals is defined by the relation

$$\gamma_{xy}(f) = \frac{|G_{xy}(f)|}{\sqrt{G_{xx}(f)\;G_{yy}(f)}}$$

Estimating the coherence function from a single observation of signals $x(t)$ and $y(t)$ yields

$$\hat{\gamma}_{xy}(f) = \frac{|(2T_s/N)\;X^{*}_k\;Y_k|}{\sqrt{(2T_s/N)|X_k|^{2}\;(2T_s/N)|Y_k|^{2}}} = 1$$

at all frequencies because when there are only two observations available (one for each signal) there is always a trivial linear relation between $x(t)$ and $y(t)$ at all frequencies.

On the contrary, when the coherence function is estimated from many observations (usually segments)

$$0 \le \hat{\gamma}_{xy}(f) \le 1$$

When the coherence values are near 1 in a given frequency domain, it is possible to have between signals $x(t)$ and $y(t)$ (or more precisely between their harmonic components) a linear dependence with constant coefficients for this frequency domain. The physical process may be, for this particular frequency domain, represented as a linear system with constant coefficients.

Note also that the phase ϕ_{xy} of the cross spectrum is interpreted as the phase difference between the harmonic components of $x(t)$ and $y(t)$ at frequency f.

Computational recipe for the cross PSD

Let us sum up now how we can estimate the cross PSD of two real signals.

1. Define the segment length, the kind of window (Parzen or Hamming) that will be used and calculate the operational series length n given by $n = n_T + (n_s - 1)\,(n_T/2) \le n_{max}$ and valid for both signals.

2. Load two segments, each form one series, into a dummy complex sequence and multiply this sequence by the window.

3. Take the FFT of this complex sequence.

4. Extract the FFTs of the two real sequences and divide by the energy of the window to compensate for its effects.

5. Sum the result $X_k^* Y_k$ into $G_{xy}(f)$ (complex number) and the squares into G_{xx} and G_{yy} and repeat the process until no more segments.

6. Divide the total by the number of segments to obtain $\hat{G}_{xx}$, $\hat{G}_{yy}$ and $\hat{G}_{xy}$.

Estimation of the gain of a physical system

As we already remarked it is sometimes convenient to represent $x(t)$ as an input signal and $y(t)$ as an output signal to a virtual physical system. With this representation in mind we define the gain as the magnitude of the transfer function of the system

$$T(f) = |H(f)|$$

From previous relations linking the PSDs of the input and output of a CPLS we note that we dispose of two ways to estimate the gain $T(f)$ of a physical system

$$T(f)_1 = \sqrt{\frac{G_{yy}(f)}{G_{xx}(f)}}$$

$$T(f)_2 = \frac{|G_{xy}(f)|}{G_{xx}(f)} \qquad (17)$$

We can show that the first estimate of the gain is a biased estimate for all cases except when the coherence function equals 1. The second estimate will be biased if noise is present at the input but will be an unbiased estimate if noise is present at the output. (cf. Bendat and Piersol). And in fact the second estimate provides an unbiased estimate of the frequency response of the system in multiple input problems when the inputs are uncorrelated.

In the SEASPECT package the second definition (expression (17)) is used for the gain estimation.

7. Confidence intervals for cross spectral characteristics

7.1. *Coherence confidence interval*

As all the statistically estimated quantities the coherence has a threshold of statistical significance. This means that the coherence value at a given frequency will be considered as significantly different from 0 if $\gamma \geq \gamma_0$.

The zero significance level γ_0 is calculated at a probability level 1-α by the formula

$$\gamma_0 = \sqrt{1 - \alpha^{1/(d-1)}} \qquad (18)$$

where d is the number of segments on which the estimation has been based.

If the coherence value is greater than this minimum threshold the coherence at this frequency can be thought of as statistically significant at the 1-α level. Usually α=0.05 and 1-α=0.95 or 95% or α=0.20 and 1-α=0.80 or 80% and we note B_{95} and B_{80} the corresponding significance thresholds.

Example

Number of segments = 5, α = 0.05

$d = 5$ and $\gamma_0 = \sqrt{1 - 0.05^{1/4}} = 0.726$

Consequently only values of γ greater than 0.726 can be thought of as significantly different from 0 with a probability of 95%. We note that with small segment numbers the requirements on γ are rather stringent.

7.2. *Phase confidence interval*

The phase, as well as the gain, of the cross spectral analysis are calculated by averaging on segments. Consequently a statistical error is also attached to their estimation.

For the phase φ at a specified frequency the confidence interval in which the theoretical phase lies with probability 1-α is φ ± Δφ with

$$\Delta\phi = \sin^{-1}[\ t(d,1-\alpha)\cdot\sqrt{(1-\gamma^2)/d/\gamma^2}\ ] \qquad (19)$$

where $t(d,1-\alpha)$ is the Student t distribution value at the probability level 1-α (usually 95%) and for d degrees of freedom, γ^2 is the coherence estimation and d represents the degrees of freedom of the statistical test equal to 2*number of segments - 2.

If at a given frequency the coherence value is not statistically significant the phase confidence interval is not calculated as it is meaningless.

Example

Number of segments = 7, γ = 0.45, α = 0.05

$d = 2\text{x}7-2=12$, $t = 2.179$

$\Delta\phi = \sin^{-1}[\ 2.179\ \sqrt{(1-0.45^2)/12/0.45^2}\ ] = 45.6$ degrees.

We remind here that the Student's t distribution is followed by a random variable of the form

$$t = \frac{z}{\sqrt{y/n}}$$

where z is a random variable following the normal distribution with zero mean and unit variance and y is a random variable obeying the χ^2 law with n d.o.f. When $n \longrightarrow \infty$ the t distribution tends to behave like the normal (gaussian) distribution.

7.3. *Gain confidence interval*

In a similar way we can calculate a confidence interval ΔT for the gain by the formula

$$\Delta T = \sqrt{F(2,d,1-\alpha)(1-\gamma^2)G_{yy}/G_{xx}/(d-1)} \qquad (20)$$

where $F(2,d,1-\alpha)$ is the Fisher F distribution value at a given probability level 1-α (usually 95%) and for 2 and d degrees of freedom, G_{yy} and G_{xx} are the autospectra of x and y and d represents the degrees of freedom of the statistical test equal to 2*number of segments-2. When the estimated coherence is less than the zero significance level the gain confidence interval is not calculated.

Note on the F distribution.

A random variable follows the Fisher distribution when it can be expressed as

$$F = \frac{y_1/n_1}{y_2/n_2}$$

where y_1 is a χ^2 random variable with n_1 d.o.f. and y_2 is a χ^2 random variable with n_2 d.o.f. It should be also noted that the t_n^2 distribution reduces to the F distribution with n_1=1 and n_2=n d.o.f.

Example

Number of segments = 8, γ = 0.45, α = 0.90, G_{xx} = 10, G_{yy} = 90

d=2x3-2=14, F(2,14,0.10)=2.73 $\Delta T = \sqrt{2.73(1-0.45^2)90/10/13} = 1.228$

8. Signal preprocessing

8.1. Mean removal

It is often desirable to transform the data so that the time series has a zero mean value. In this case we define a new time history by

$\hat{x}(t) = x(t) - \bar{x}$

The effect of mean removal on the PSD estimation is mainly to have considerable less power on channel 0 (frequency = 0).

8.2. Trend removal

A trend present in the data is any frequency component whose period is longer than the record length.

The trend introduces low frequencies that cannot be removed by high pass digital filters that will be discussed next. Hence some special trend removal technique must be applied.

Least squares procedures can be employed for the removal of a linear as well as a polynomial trend. Consider the time series u_n and the polynomial fit $\hat{u}_n$ of order K and coefficients b_k given by the expression

$$\hat{u}_n = \sum_{k=0}^{K} b_k \,(n\,h)^k$$

where h is the sampling period of the times series.

In the least squares fit the set of coefficients b_k is chosen so as to minimize the expression

$$Q = \sum_{n=1}^{N} (u_n - \hat{u}_n)^2 = \sum_{n=1}^{N} \left[\, u_n - \sum_{k=0}^{K} b_k (nh)^k \,\right]^2$$

which is always non negative for any choice of b.

The desired coefficients can be obtained by taking partial derivatives of this equation with respect to b_l and setting them equal to 0.

$$\frac{\partial Q}{\partial b_l} = 0$$

This yields K + 1 equations of the form

$$\sum_{k=0}^{K} b_k \left[\sum_{n=1}^{N} (n\ h)^{k+l} \right] - \sum_{n=1}^{N} u_n (n\ h)^l \qquad l = 0, \ldots, K$$

The simplest LSQ polynomial is that of degree 0 and we find

$$b_o = \frac{1}{N} \sum_{n-1}^{N} u_n$$

For a straight line

$$\bar{u}_n = b_o + b_1\ (n\ h) \qquad n = 1,\ldots,N$$

we can prove the following formulas

$$b_o = \frac{2(2N+1)\sum u_n - 6 \sum (n\ u_n)}{N\ (N-1)} \tag{21}$$

$$b_1 = \frac{12 \sum (n\ u_n) - 6\ (N+1) \sum u_n}{h\ N\ (N-1)\ (N+1)} \tag{22}$$

These formulas are implemented in the trend removal option of the SEASPECT package.

Noting that the trend removal can transform an apparently non-stationary signal into an (also apparently) weakly stationary signal, the trend removal is an important intermediate step in the digital processing of random data and it should be given its due consideration. If trends are not eliminated in data, large distortions can occur in the later processing of spectral quantities. In particular trends in data can completely nullify the estimation of low frequency spectral content. Caution is nevertheless advised here, however, in that trend removal should be performed only if trends are physically expected or clearly apparent in the data.

8.3. *Digital Filtering*

Data filtering may be desired for two purposes

- data smoothing
- separation of frequency components

those two points being interrelated.

Analog filters

x(t)
ANALOG FILTER
y(t)

$$y(t) = \int_0^{+\infty} h(\tau)\ x(t-\tau)\ d\tau \quad \text{with } h(t) = 0 \text{ for } t < 0$$

$$H(f) = \int_0^{+\infty} h(\tau)\ e^{-j2\pi f\tau}\ d\tau$$

In designing a digital filter, unlike an analog filter, it is not necessary for the filter to be physically realizable. That is, it is not required that $h(t)$ be zero for negative t, since all the data can be stored in a computer and then run backwards to filter the data in reverse order.

With the suitable selection of a filter we can achieve the cut-out of a specific region of the spectrum of the x_i and the maintaining of other regions.

If we want to cut out the power of high frequencies of the spectrum we use low-pass filters. In the opposite case we use high-pass filters. An ideal low-pass filter must multiply the spectrum of x_i by the function

$$\chi(f) = \begin{cases} 1 & 0 \le f \le f_c \\ 0 & f \ge f_c \end{cases}$$

where f_c is the cutoff frequency, the basic characteristic of the filter.

If we filter the input signal x_i by a low-pass filter and consider the signal $z_i = x_i - y_i$ it is obvious that the power spectrum of this last signal

will be confined in the high frequencies. In fact $Z_k=X_k-Y_k$ and for small k, $Y \simeq X$ so that

will be confined in the high frequencies. In fact $Z_k=X_k-Y_k$ and for small k, $Y_k \simeq X_k$ so that $Z_k \simeq 0$. We may thus assume that the signal z_i stems from the initial signal x_i via a high-pass filter.

We note nevertheless that the sum of the power spectra of y_i and z_i is not equal to the spectrum of the original signal x_i and that the cutoff frequencies of the filters $x_i \longrightarrow y_i$ and $x_i \longrightarrow z_i$ are not the same.

Non-recursive digital filters

(Moving average or Finite Impulse Response Filters)

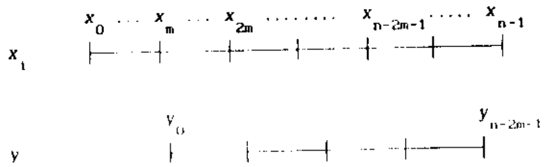
We confine our attention only to symmetric filters and by the term filter we designate here a finite linear numerical processing of the values of the discrete signal x_i.

If we call x_i the initial (unfiltered) signal and y_i the filtered signal by definition we will have

$$y_i = \sum_{k=1}^{m} h_k \,(x_{i+k} - x_{i-k}) \tag{23}$$

The coefficients h_k, $k = 1,..,m$ are called filter weights and their values define entirely the filter. Index i in the previous relation takes all the values for which the left hand side sum makes sense.

It is obvious that the filtered signal y_i will contain less values than the original signal x_i. In fact the very first value of the filtered signal that can be calculated is y_0 and uses the values $x_0, x_1, \dots, x_m, x_{m+1}, \dots, x_{2m-1}$



because calculation of y_n involves only values of the input signal. The transfer function of this filter is

$$H(f) = 2\sum_{k=1}^{m} h_k \cos(2\pi f k\Delta t) \tag{24}$$

and is related to the DFT's of input and output signals x_i and y_i by the following relation where $|H(f_k)|$ is termed filter gain.

$$|Y_k| = |H(f_k)|\;|X_k| \qquad f_k=k/(N\cdot T_s) \qquad k \geq 0$$

We remark that the transfer function has real values and that it does not introduce a phase shift between the input and the output time series. This is one of the advantages of the symmetric filters and for this reason they are called zero-phase filters.

Inverting above relation (24)

$$h_k = \int_{-\infty}^{\infty} H(f)\cos(2\pi f k\Delta t)\,df$$

Hence we can calculate the weights of a filter given its transfer function $H(f)$ and simulate thus the ideal transfer functions discussed above.

Assume for instance that we want to simulate the following transfer function

$$H(f) = \begin{cases} 1 & -f_0 \leq f \leq f_0 \\ 0 & \text{otherwise} \end{cases}$$

The corresponding coefficients are given by

$$h_k = \int_{-f_0}^{f_0} \cos(2\pi f k\Delta t)\,df = \frac{\sin(2\pi f_0 k\Delta t)}{\pi k\Delta t}$$

The coefficients h_k exhibit a slow decay rate $h_k \propto 1/k$ so that large values of k are required before these weights become negligible. In practice this type of nonrecursive filter requires usually so many weights (100 or more) that it is not considered a very efficient method of filtering. Also the abrupt cutoff at frequency f_0 will produce undesirable frequencies and generally smoother transfer functions are simulated.

Smoothing the values of a time series can be achieved very easily with a simple filter of the form

$$y_i = w_{-1}x_{i-1} + w_0 x_i + w_1 h_{i+1}$$

involving only three values of the original series and usually symmetric ($w_{-1} = w_1$). Smoothing the time series values alters obviously the series power spectrum by cutting off some high frequencies.

Recursive digital filters
(Autoregressive or Infinite Impulse Response Filters)

The recursive digital filters are filters where the output results not only from a finite sum of input terms but also by using previous outputs as inputs. Stated in engineering terms this procedure is called feedback.

A simple standard type of recursive filter is given by

$$y_n = c\ x_n + \sum_{k=1}^{m} h_k\ y_{n-k} \qquad (25)$$

which uses m previous outputs and only one input.

The Fourier transform of this equation yields the result

$$Y(f) = c\ X(f) + Y(f) \sum_{k=1}^{m} h_k\ e^{-j2\pi fk\Delta t}$$

Replacing the exponential by the letter z leads to procedures for analyzing digital filters in terms of what is called z-transform theory

$$H(f) = \frac{Y(f)}{X(f)} = \frac{c}{1 - \sum_{k=1}^{m} h_k\ e^{-j2\pi fk\Delta t}} = \frac{c}{1 - \sum_{k=1}^{m} h_k\ z^k}$$

Studies of the properties of $H(f)$ are thus reduced to determining the location and nature of the poles in the denominator of this last result.

An important application of lowpass filters can be found in data decimation. In order to decrease the amount of data for later analysis we subsample the times series every r^{th} point. Suppose, as usual, that a rrecord is sampled at Δt apart and an r^{th} order decimation is performed. Then the new sampling rate is actually $\Delta t' = r\Delta t$ and the new maximum detectable frequency $1/(2r\Delta t)$. Hence all the information above $1/(2r\Delta t)$ will be folded back into the interval $[0, 1/(2r\Delta t)]$. To avoid this folding the original data should be filtered to remove the information in the frequency range above $1/(2r\Delta t)$ by means of a suitable lowpass filter such as the above.

REFERENCES

A. Peled and B. Liu
Digital Signal Processing (theory, design and implementation)
Wiley, New York, 1976

G. Jenkins and D. Watts
Spectral Analysis (and its applications)
Holden-Day, San Francisco, 1968

J.S.Bendat and A.G. Piersol
Random Data: Analysis and Measurement Procedures
Wiley-Interscience, New York, 1971

C. Chatfield
Time Analysis of Time Series: Theory and Practice
Chapman and Hall, 1975

ROTARY SPECTRAL ANALYSIS

by Miroslav Gacic

Introduction

Periodic changes in the vector field can be identified by the rotation of vectors. During the time lapse of one period the vector rotates for 360^{0}. Certainly, during one period vector not only changes direction but also a magnitude. The rotation could be clockwise or anti-clockwise i.e. according to the convention negative or positive. Therefore, to periodic or any other type of time-dependent variations in the vector field positive or negative sign can be attributed. In fact any time-dependent variations in vector field can be represented as a sum of positive and negative rotation of different amplitudes. Relationship of amplitudes of positive and negative rotation determines the sense of the resultant rotation and the degree of the polarization. If amplitudes of positive and negative rotation at a given frequency or period are equal then the variations are bi-directional with vector tip moving along the straight line. On the other hand, if amplitude of one of the senses of rotation is zero then the vector tip describes the circle in a clockwise or anticlockwise sense depending on which of the two amplitudes is non-zero. Rotary spectral analysis is a statistical method developed to define some statistical coefficients which will enable us to describe time-dependent variations in a vector field. This method was developed and introduced first by Gonella (1972) for auto-spectral analysis of two-dimensional vector time-series and later it was generalized to cross-spectral analysis and in 3D coordinate system by Mooers (1973). However, most of the applications of this method have come from the paper by Gonella i.e. mostly applied to a single vector time-series analysis.

Here, basic ideas and formulae for the rotary spectral analysis following Gonella (1972) paper will be presented. Also, an application of the method to current vector time-series will be presented.

Rotary spectral analysis

Let us take a two-dimensional vector "U" with scalar components "u" and "v" along horizontal rectangular axis. As we have already said the frequency of temporal variations (σ) in current field is always positive by definition, however the angular velocity (ω) may be positive or negative depending on the sense of the vector rotation; $\omega=+\sigma$ for anti-clockwise or

$\omega=-\sigma$ for clockwise rotation.

The Fourier transform for the angular velocity ω is given by:

$$u_\omega = (1/d) \int_0^d u(t)\, e^{-i\omega t}\, dt = |u|_\omega e^{i\Theta}$$

This represents the amplitude and the phase with an angular velocity ω. There is certainly a complex conjugate which generally differs from it. Symbol "d" represents the length of the record and ω is a negative or positive multiple of $2\pi/d$ which determines the frequency resolution of our record.

Along each of the coordinate axis the component at the angular frequency σ can be expressed by following relations:

$$u_\sigma = a_{1\sigma} \cos \sigma t + b_{1\sigma} \sin \sigma t$$

$$v_\sigma = a_{2\sigma} \cos \sigma t + b_{2\sigma} \sin \sigma t$$

The tip of the vector rotating with the angular frequency σ describes the ellipse represented by the equation:

$$u_\sigma + iv_\sigma = u_{+\sigma} e^{+i\sigma t} + u_{-\sigma} e^{-i\sigma t}$$

From there components rotating in the anti-clockwise and clockwise sense are given by the following expression:

$$u_+ = (1/2)\{(a_1+b_2)+i(a_2-b_1)\}$$

$$u_- = (1/2)\{(a_1-b_2)+i(a_2+b_1)\}$$

The mean kinetic energy of the motion is then given by:

$$S_t = S_- + S_+$$

Clockwise and anti-clockwise spectra are expressed by the following relations:

$$S_+ = \langle u_+^* u_+ \rangle; \quad S_- = \langle u_-^* u_- \rangle$$

where "*" denotes complex conjugate variables. The symbol "<>" means the average over number of records. The procedure of averaging over number of records is suggested for any analysis in order to obtain more reliable spectral estimates.

The difference between the clockwise and anti-closkwise spectra is proportional to the average of the area of surfaces limited by ellipses and the sign gives the rotation sense. The ratio between this difference and the total spectrum is called "rotary coefficient" and gives the partition of energy between the two rotation senses. It is zero for the unidirectional motion and + or -1 for pure clockwise or anti-clockwise motion.

Since we assumed that at the each frequency the resultant vector rotation can be presented as a sum of positive and negative one, the orientation of the major ellipse axis is:

$$\Theta = (\Theta_+ + \Theta_-)/2 + k\Pi$$

Therefore the orientation of the major axis depends on the phase of both positive and negative rotation component; i.e. major axis is oriented in the direction where both positively and negatively rotated vectors are at the same straight line. Minor axis lie on the straight line where the two vectors have the opposite orientation i.e. $\Pi/2+(\Theta_+ + \Theta_-)/2$.

All parameters concerning the rotary spectral analysis can be related to autospectral and cross spectral estimates. These estimates can be represented in terms of Fourier coefficients a_1, b_1, $a_2$and b_2:

-autospectra $\quad P_{uu} = \langle a_1^2 + b_1^2 \rangle$ and $P_{vv} = \langle a_2^2 + b_2^2 \rangle$,

-cross spectra $\quad P_{uv} = \langle a_1 a_2 + b_1 b_2 \rangle$,

-quadrature spectra $\quad Q_{uv} = \langle a_1 b_2 - a_2 b_1 \rangle$.

Then, the clockwise spectrum is:

$$S_- = (1/2)\langle u_- u_-^* \rangle = (1/8)\{P_{uu} + P_{vv} - 2Q_{uv}\},$$

the anti-clockwise spectrum is:

$$S_+ = (1/2)\langle u_+ u_+^* \rangle = (1/8)\{P_{uu} + P_{vv} + 2Q_{uv}\}.$$

From there the total spectrum can be calculated from the formula:

$$S_t = S_- + S_+ = (1/4)\{P_{uu} + P_{vv}\},$$

and the rotary coefficient is:

$$C_R = \frac{S_- + S_+}{S_t} = \frac{-2Q_{uv}}{P_{uu} + P_{vv}} .$$

The mean orientation of the major ellipse axis is then expressed by the equation:

$$\Phi = \frac{2P_{uv}}{P_{uu} - P_{vv}} .$$

In these notes no mention will be given to the analysis for pair of vector series. In Gonella (1972) the formulae for calculation of the coherence in complex form for each angular velocity are also presented.

References:

Gonella, J., 1972: A rotary-component method for analysing meteorological and oceanographic time series. Deep-Sea Research, 19, 12, 833-846.

Mooers, C. N. K., 1973: A technique for the cross-spectrum analysis of pairs of complex-valued time series, with emphasis on properties of polarized components and rotational invariants. Deep-Sea Research, 20, 1129-1241.