

Cross: Cross Spectra by Sine Multitapers

Robert L. Parker

This program calculates the power spectral densities, cross spectra, coherences and phases of a pair of time series, using the theory of Riedel and Sidorenko (*IEEE Trans Sig. Proc.*, 43, 188-195, 1995). The program makes plotfiles for *plotxy* and runs then during execution. The user need not exit the program to experiment with different parameter settings or different data files.

Cross runs as a command interpreter. The user enters options from the catalog below. Options not required may be omitted and defaults will be chosen by the program if necessary.

The following is an example of a short program; remarks after the percent signs are treated as comments by *cross*.

```
file easter.dat      % Names the file containing the series
adapt 2             % Employ adaptive process with 2 iterations
nterms 3000         % Read 3000 terms from the file
columns 4 6         % Read values from these columns
units km nT        % Units of indep variable and data
interval 0.35       % Values are spaced every 0.35 km
detail 0 0.2        % Plot spectra in (0, 0.2) /km
unwrap             % Unwrap phase jumps
exec               % Perform instructions entered so far

colu 5 6           % Another calculation – now use these columns
exec               % Do the calculations

quit               % Enough
```

Below is an explanation of all the commands.

Introduction

Every command takes the following form: a command word chosen from the catalog below, which may always be abbreviated to its first four letters and which may appear alone or may be followed by one or more spaces, then numbers or text. Numbers must be separated by spaces. The command word must begin in the first column; lines starting with a blank are treated as comments and are ignored.

Following **exec**, after the calculations from one set of commands are over, control is returned to the command line for further tasks. All the commands issued up to this point remain in force, but may be replaced in subsequent calculations as required. The program prints out the current stack of commands before beginning its work; you can list a stack at any time with the **review** command. In every execution the

program works its way from the bottom of the command stack to the top, executing the one it finds first. To turn off a command that doesn't have an argument (like **unwrap**) use the **clear** command, which erases the named command from the list. Note that four commands take effect immediately and do not go into the stack: they are **exec**, **quit**, **review** and **clear**.

For concreteness in the documentation the independent variable will be called time and the data to be analyzed a time series.

Command Catalog

?: Print a brief version of this catalog.

execute: With the specifications entered up to this point, compute the field values as requested. Then wait for further commands.

quit: Cease.

data filename [file-2]: Specify the name (or names) of the diskfile(s) containing the time series to be analyzed. The data must be in ASCII. If there is one filename, both series are read from that file, otherwise separate files hold the two series. With one filename the data must be in columns (at least two of them). To read from a particular column you must use the **column** command. If that command is not issued, the program reads from column 1 and column 2. Normally the program will read to the end-of-file, but with the **nterms** command you may stop reading data early. With two file names you can still use the **column** command if you need to, with the first column read from file-1, the second from file-2. Without the **column** command, the program reads from the first column of each named file. There is no **spline** interpolation with data in two files. Very long series can be decimated to fit into memory.

file: The same as **data**.

column k1, k2 [k3]: If one data filename is given the file must contain the two series arranged in columns. But there may be more than two such columns in the file. To select values from the third column and fifth column, for example, use $k1 = 3$, $k2=5$; and so on. The third column specifier, $k3$, is used with **spline**: then $k1$ corresponds to time; $k2$ to series 1; $k3$ to series 2.

spline [natural/Akima]: When the series are not evenly sampled in time they can be transformed to an equally spaced pair with spline interpolation. Now the observations must be provided as three columns in a single diskfile, with one column giving the times of observations. The two-file option is not permitted with **spline**. The values of the time variable must increase. The desired spacing for the series must be provided in the **interval** command. The default ordering of the information in each line of the file is assumed to be: time, $x1$, $x2$, but this can be changed with **column**. Two types of spline interpolation are provided; Akima splines generate less overshoot with very irregular data series. You can get an idea of how well the interpolation works with the **validate** command.

validate filename: Write a diskfile with the provided name containing the interpolated series. The file is written with three values per line: t , $x1(t)$, $x2(t)$.

skip n1 [n2]: Before reading any data, skip $n1$ lines. If there are two filenames in the **data** command, $n1$ refers to the first file and $n2$ to the second. If only $n1$ is provided for two data files, the same number of lines is skipped in both.

decimate *nf*: Subsample both input series by the factor *nf*. The original series are smoothed with a simple anti-aliasing filter before sampling, thus allowing the user to obtain the low-frequency portion of the spectra of very long data sets without loss of accuracy. The command **decimate** must not be issued with **spline** or used with data on two input files. Note that the parameters to other commands, such as **interval** and **nterms**, refer to the series **before** decimation.

nterms *n*: Read only *n* data values from the file.

detrend: Fit straight lines by least squares to the series and subtract the lines before estimating the spectra. Note that the mean level is always removed from the series whether or not the trend is taken out.

interval *dt*: The number *dt* is the sampling interval in time at which that data are recorded. If the data are not uniformly sampled in time, you can interpolate them with **spline** onto even sampling with spacing *dt*.

units *time data*: Name the units used for time and the data series; these names enable the program to label the graph axes properly and to get the spectral units right in the print out.

output *filename*: Write the estimated spectra to this file. The file contains 9 columns: (1) frequency; (2) PSD of series 1; (3) PSD of series 2; (4) co-spectrum between series; (5) quadrature spectrum; (6) gain of transfer function from series 1 to 2; (7) squared coherency spectrum; (8) phase spectrum; (9) number of tapers *K*, used to make the estimates. The value of *K* in the last column will enable you to compute uncertainties, confidence levels, etc. The same number of tapers is applied to all four spectra at each frequency, but will vary with frequency in the adaptive scheme. If unnamed, the program uses the file *fort.1*. If the program generates several spectra they will appear in this file in the order they were generated. You can switch files for each new result if you like by renaming the file, but returning to a previously used file will overwrite the earlier spectrum.

output *-none*: Inhibit output of results to disk.

plot *filename*: The program launches a shell script that plots the spectrum to the terminal with the program *plotxy*. This terminal graphic runs independently in parallel with *cross*, and if several series are analyzed, each will generate its own plot. The *plotxy* plotfile is named in this command, or will default to *fort.2* if the command is omitted. The plotfile is self-contained and does not refer to the file named in the **output** command. When several spectra are generated in a single session and only one plotfile has been named, that file is overwritten by each successive new result so that at the end of calculations only one plotfile survives.

plot *-none*: Inhibits the generation of a plotfile.

detail *f1 f2*: Plot the spectrum only in the frequency range from *f1* to *f2*. If a particular interval is more interesting this command allows the user to concentrate on it. Naturally the output file still contains the complete solution.

replot *f1 f2*: As in **detail** but without recalculating anything. Thus **replot** redisplay results from the previous calculation and is most useful during an interactive, exploratory phase of data analysis. The command is cleared after execution. If you enter other commands describing further calculations along with **replot**, they will be ignored until the next **execute** command.

title *heading*: Plot the text heading over the current plot. If this command is not issued the name of the current data file(s) will be used for a heading.

logfreq: By default the frequency axis is plotted linearly; this command causes the frequency axis to be logarithmic. Use **clear logfreq** to undo.

hold: When present in the command list, prevents plotting to the screen, but unlike **plot -none**, does not suppress the output of a plotfile for later plotting. This is useful when *cross* is run inside a script.

tapers *q*: Specify the exact number of tapers to be used in the estimate. The same number will be used for each frequency, so the relative error and frequency resolution will be uniform across the spectrum. The command **adapt** overrides this one, so you should clear **adapt** to cancel it if it has appeared prior to **tapers**. See endnotes.

adapt *ntimes*: Perform adaptive estimation in which the number of tapers (and hence the resolution and uncertainty) vary according to spectral shape. This is the default mode with *ntimes*=3. In frequency ranges where the spectra are relatively flat, more tapers are taken and so a higher accuracy is attained at the expense of lower frequency resolution. The program makes a pilot estimate of the spectra, then uses Riedel and Sidorenko's estimate of the MSE (minimum square error) value, which is based on a knowledge of the 2nd derivative of the PSD. The process is repeated *ntimes*; if the numeral is omitted, *ntimes* is taken to be 3, which is usually sufficient. If *ntimes* is set to zero, adaptive tapering is not carried out – the same number of tapers is applied at all frequencies.

review: Print out the current command list.

clear *command*: Remove all earlier occurrences of the named command from the command list. Used with commands like **unwrap** that cannot be canceled through their arguments.

unwrap: Conventionally phase is plotted between -180 and 180 degrees, but if the phase is hovering around 180 degrees, or if there is displacement in time between the two series, a multitude of 360 degree phase jumps will clutter the graphs. This command unwraps the phase jumps, usually at the expense of giving a smaller plot scale in this variable.

confidence *p*: When the true coherence is small, the coherence estimator is biased upward. So, to decide if there is significant coherence, *cross* finds and plots in gray the level below which the estimates will fall with probability *p*, when the two series are actually incoherent. If the command is not issued *p* = 0.95. If a value of *p* is set outside the interval (0, 1) the confidence line is not plotted.

difference *j*: Near zero frequency standard methods of phase estimation are biased towards 0 or 180 degrees. If the true phase is near 90 degrees we can get a better estimate by first differencing one of the series, performing cross-spectral analysis then compensating in the frequency domain. That is what this command does; *j* is 1 or 2, indicating the 1st or 2nd series is differenced. The estimates of the PSDs may not be very good.

prewhiten *nar*: Before estimating the spectra, convolve both series with an AR filter designed to flatten the spectrum of one of the series using the Yule-Walker equations (this is prewhitening). When *nar* is positive the AR filter is based on the first series and *nar* terms are used; with negative *nar*, the second series is selected, and of course the order of the filter is the magnitude of *nar*. To cancel this option set *nar* to zero.

smooth *fw*: It is sometimes desirable to control the amount of smoothing, even in adaptive estimation. In the adaptive scheme, the spectra are estimated by minimizing a loss function that is the a weighted sum of variance and squared bias. A weight of

wt-squared is applied to variance, thus emphasizing smoothness as the expense of local bias. So to increase smoothing, raise wt, and conversely. It is prudent to stay within the interval (0.2, 5).

Notes

The adaptive process introduces a variable resolution and error in the frequency domain. These are not plotted during the run as they tend to clutter up the graph intolerably. Instead the program outputs K (the number of tapers) in the diskfile of results and the user can calculate statistical quantities from it. For example, the frequency resolution is given by $K \cdot f_N / N_f$ where f_N is the Nyquist frequency and N_f is the number of frequencies estimated. Although there is nothing in theory that says the four different spectra and cross spectra should not each be computed with different number of tapers, doing so opens the possibility to coherence estimates that exceed unity, a highly undesirable property. Thus the program examines the shapes of the two PSDs and selects the more conservative (that is the smaller) number for all the calculations.

The adaptive process used is as follows. For each series the PSD is estimated by a sine multitaper weighted sum of K tapers. The number K is decided by choosing the smaller of the two optimum tapers estimated for the individual PSDs, based on a local, variable length, second derivative estimator. This is used in R & S's eq (13) for the MSE taper number. Compared with the Thomson multitaper programs, this one is fairly simple, short and fast.

The sines are not computed and multiplied into the series; instead only one FFT is done per series, and the indices juggled to extract the transforms of the tapered time series.

The sine multitapers do not share the splendid spectral-leakage rejection properties of Thomson's prolate spheroidal tapers. Therefore be warned that if your spectra have strong peaks or are highly red, you are in danger of obtaining biased results due to spectral leakage. Since prolate multitapers are not available in *cross* one option is to **prewhiten**.

Coherence estimates will be biased downward if there are large phase shifts. If a simple translation in time can bring the two series into alignment, and thereby remove a linear ramp in phase, performing that shift will improve the coherence estimates. This can be done without editing the data files by using the *skip* command with two values and two file names; notice the names may be identical if the original numbers are held in one file. To help determine the required shift, the program prints an estimate of the number of time steps series 2 lags behind series 1.