PSD: Power Spectra by Sine Multitapers

Robert L. Parker

This program calculates the power spectral density of a real time series, based on the theory of Riedel and Sidorenko (*IEEE Trans Sig. Proc.*, 43, 188-195, 1995). The program makes plotfiles for *plotxy* and runs them during execution. The user need not exit the program to experiment with different parameter settings or different data files.

Psd 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 as necessary.

Here is an example of the simplest program; it should produce a very satisfactory spectrum.

```
file mag.dat
output mag.psd
exec
quit
```

Data values are in a single column in the ASCII file mag.dat; they are sampled evenly in time. *Psd* uses adaptive tapering to make an optimal spectral estimate, which is plotted to the terminal as well as recorded in the file mag.psd.

The following is an example of a more elaborate program; remarks after the percent signs are treated as comments by *psd*. The spectrum is computed with the adaptive algorithm to optimize the balance between bias and variance and there is no prewhitening – these are default settings.

```
file both.fin           % Name the ASCII file containing the time series
column 4                % Read from column 4 of the file
interval 0.056          % Give data sampling interval
units sec nT            % Name units of independent variable & data
execute                 % Find the power spectrum

superimpose             % Plot next spectrum on same graph as the last one
column 3                % Now read a series from col 3 (same file)
execute                 % Find the new spectrum

quit                    % Terminate session
```

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. The command may appear alone

or may be followed by one or more spaces, then numbers or text. The command word must begin in the first position; lines starting with a blank are treated as comments and are ignored. In the description below, a bracketed term, like [k], may be omitted.

The general procedure is to enter a collection of commands describing a calculation, then enter the command **execute**, which tells the program to perform the specified task. When the calculation is complete, control is returned to the user via the command line, where further instructions can be issued. All the commands given up to this point remain in force, but they can be replaced in subsequent calculations as required. The program writes to the screen the current list of commands before beginning its work; you can see the list at any time with the **review** command. Upon receiving the **execute** command, the program works its way from the bottom of the command list to the top; if there are duplicate commands, *psd* performs the most recently issued command. The internal list of commands is called the command stack. To turn off a command that doesn't have an argument (like **hold**) use the **clear** command, which erases the named command throughout the stack. Note there are four commands that take effect immediately and do not go into the stack: they are **execute**, **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 saved in memory, compute the spectrum values as requested. Then go on to further commands.

quit: Cease.

data: the same as **file**.

file *filename*: Specify the name of a diskfile containing the time series to be analyzed. The data file must be ASCII and may be in the format of several columns of data or may comprise a continuous string of numbers. To read from a particular column you must use the **column** command. If **column** is not issued, the program reads the first number on every line. Normally the program will read to the end-of-file, but with the **nterms** or **terms** command you can get it to stop reading data early.

file *-continue*: Usually a data file is read from the beginning to the end of file, but sometimes several series are written end-to-end in the file. To input each one, you must know how many points are to be read, and you specify this in **nterms**. When the –continue flag is present, input resumes from the current file, starting immediately after the earlier series. If you omit this flag, the file is rewound and read from the beginning again.

column *k1, [k2]*: The data file may contain several series arranged in columns. To select the third column, for example, use the first form with a single argument, k1=3; and so on. The second form, with two column numbers, is used in conjunction only with the **spline** command, to tell the program which column contains time and which the data values. If **column** is not present, the program takes k1=1.

Psd can also read every number on every line; this special kind of reading is accomplished by setting k1=0. Then there may be different numbers of values on each line.

skip *n*: Before reading data from the file, skip *n* lines.

nterms *n*: Read only n data values from the file. If this command is omitted the program reads to the end-of-file, or until memory is filled. With **decimate** the value of n is the desired number of terms in the *decimated* series.

terms *n1 n2*: Read from term number n1 to n2 in the file. The *-continue* option in **file** is overridden when this command is used: the file is rewound, and the command obeyed as stated. If **terms** and **skip** or **nterms** appear together, any conflict is resolved by giving **terms** priority.

spline *[natural|Akima]*: Take two columns from the data file as defined by the **column** command, where column k1 contains time and k2 data values. Now interpolate with cubic splines onto evenly spaced samples at an interval dt; the value of dt must be set in the **interval** command. The values in the time column must be strictly increasing. The argument specifies the type of spline to be employed; the default without a argument is natural. Experience suggests Akima splines are better for highly irregular series.

validate *filename*: If the data have been interpolated or decimated a record of the series used to generate the psd is written to the named diskfile for comparison with the original; both t and x(t) are written to the file. Such inspection is a wise policy, particularly with **spline** if there are large gaps to be filled by interpolation. If the command is not **clear**ed, all interpolated input series will be written to the file in sequence.

detrend: Fit a straight line by least squares to the series and subtract the line before estimating the spectrum. Detrending is done after spline interpolation. 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 the data are recorded in the data file. If the data are not uniformly sampled in time, you must use the **spline** command to interpolate. Then dt sets the scale for the spectra. If the command is omitted dt will be set to unity.

decimate *nd [nw]*: Decimate the input series by the factor nd before analysis. This feature allows the very-low-frequency spectrum to be estimated for time series too long to fit into memory. An anti-alias convolution filter with 10*nd terms is applied by default; the number of terms can be chosen with nw, but nw must not be less than nd. The sampling interval dt defined in **interval** is the one in the file BEFORE decimation. The desired number terms requested in **nterms** is the number AFTER decimation. Decimation cannot be performed in combination with **spline**.

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 listing.

output [*filename*]: Write the estimated spectrum to this file; if no file is named, use fort.1; if this command is omitted, DO NOT SAVE results. The file contains 5 columns: (1) frequency; (2) estimated PSD; (3) estimated error in the PSD; (4) estimated frequency resolution; (5) number of tapers used in estimate. The estimated error is the one-standard-deviation value; when adaptive tapering has been applied (the default), multiply by square-root of 1.25 to obtain the total error, which includes a bias term. If during the course of a run, the program generates several spectra, they will appear in this file, in order. You can switch files for each new result with **output** if you like, but returning to a previously used file will overwrite the earlier material.

output *-none*: suppresses further output to disk. Same as *clear output* since the default state does not write the spectra to disk.

plot *filename*: Prepare a plotfile in the named file for the program *plotxy*, summarizing the most recent calculation. The program also launches a shell script plotting the spectrum to

the terminal. This terminal graphic need not be dismissed for the program to continue. If the command is omitted, a plot will be prepared anyway and the *plotxy* written to fort.2. The plotfile is self-contained and does not refer to the file named in the **output** command, so that a graphic is displayed even when the output file has been canceled. When several spectra are generated in a single session and only one plotfile has been named, that file is overwritten by each successive plot. The graph can contain a series of spectra through the use of the **superimpose** command.

plot *-none*: inhibits the generation of files for plotting.

superimpose: Normally, when several spectra are generated, they are plotted on separate graphs. To cause successive spectra to be plotted on the same graph, issue this command. Then the spectra are shown in different colors, in the sequence of *plotxy* color numbers: 1, black; 2, red; 3, blue; 4, green; 5, brown; 6, orange; 7, yellow; 8, purple; 9, gray. Particularly when *psd* is running from a script, you may not want to see the successive plots appear on the screen as they are generated. They can be saved up with the **hold** command.

hold: When **superimpose** is issued, the program plots the graphs with one, then two, then three spectra, and so on, each one on top the last. This causes clutter on the screen, particularly if *psd* is running from a script. If **hold** is issued, no plots will be displayed, even though they will be accumulated in the current plotfile. If you wish to see the current set of spectra displayed on the terminal before the end of the run, remove **hold** with the **clear** command; the current accumulation of spectra is always plotted when you quit the program. Thus a typical script for *psd* generating several spectra might look like this:

```
file data1
superimpose
hold
exec
file data2
exec
file data3
exec
file data4
exec
quit
```

The spectra for the four data series would then appear on a single plot at the end of the run.

logfreq: The power spectral graph will be drawn on a logarithmic frequency axis. To cancel use *clear logfreq*.

title *text*: Identify the plot produced at the next **execute** command with this title.

detail *f1 f2*: Plot the spectrum only in the frequency range from f1 to f2. When a particular interval is of special interest this command allows the user to concentrate on it. Naturally, the output file still contains the complete PSD.

replot *f1 f2*: Without performing any new calculations, replot the spectra on the stated frequency interval. This command just edits the current plotfile. You can include the command **logfreq**, but anything requiring new calculations is ignored. **replot** is automatically cleared after execution and is only useful in exploratory spectral analysis running *psd* interactively. Clear the **superimpose** command before using **replot**, since otherwise the benefits of an expanded scale may not be realized. An **execute** command is required; commands requiring further spectral calculations will not be performed until the next **execute** command.

tapers [<] *nt*: Without the symbol < specify the number of tapers to be used in the estimate, over-riding the implied adaptive process; the same number will be used for each frequency, so the relative error and frequency resolution will be uniform across the spectrum. See the endnotes. With < the adaptive process is applied, but the number tapers used is bounded above by nt. This option is useful with very long time series, where time can be consumed needlessly reducing an already acceptable variance.

adapt *ntimes*: Perform adaptive estimation with the sine multitapers in which the number of tapers (and hence the resolution and uncertainty) vary according to spectral shape. In frequency ranges where the spectrum is 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 spectrum, then uses Riedel and Sidorenko's estimate of the MSE (minimum square error) value, which is based on an estimate of the 2nd derivative of the PSD. The process is repeated ntimes. This is the default mode of operation so that omitting the command results in adaptive estimation with 4 iterations. Larger values are occasionally necessary to reach convergence; you can usually detect an unconverged state by a rather jagged appearence of the spectrum. See the endnotes.

prewhiten *nar*: The R&S multitapers do not exhibit the remarkable spectral-leakage suppression properties of the Thomson tapers, so that in spectra with large dynamic range, power bleeds from the strong peaks into neighboring frequency bands of low amplitude – spectral leakage. Prewhitening can ameliorate the problem, at least for red spectra (see Chapter 9, Percival and Walden, *Spectral Analysis for Physical Applications*, 1993). The time series is first filtered in the time domain with a finite-impluse-response filter of nar terms. The filter is found by solving the Yule-Walker equations for which it is assumed the series was generated by an autoregressive process, order nar. Next the multitaper spectral estimates are made on the filtered series. Finally, the effect of the filter is removed by dividing out the spectrum of the autoregressive process. Experience suggests there is little benefit in choosing nar greater than 4.

It is possible to apply the prewhitening of one series to later time series by setting nar=−1. Then the current calculation employs the prior filter, provided of course that there has been an earlier calculation involving prehitening with nar > 1.

save filename: Save the prewhitened series to the named diskfile, if prewhitening has been done. This command is cleared automatically so it must be repeated every time it is needed. If the same file name is re-used, earlier information is overwritten.

smooth *wt*: It is sometimes desirable to control spectral smoothness, even in adaptive estimation: for example, if resonance peaks are suspected, one might want to increase resolution at the expense of statistical reliability. In the adaptive scheme, the loss function is normally the sum of variance and squared bias (which is minimized pointwise in the spectrum); the bias term in the loss function is weighted by the factor wt-squared, thus favoring smoother and less variable spectra when wt>1. It is prudent to stay within the interval (0.2, 5), and it may be necessary to increase the number of iterations with **adapt** when wt<1.

autocorrelation *filename*: Find the autocorrelation function of the series from the cosine transform of the PSD, and save this to the named file. Also display the function to the screen. None of the refinements, like **superimpose**, **replot**, etc, apply to the plot function. The named file is rewritten for each series, so that to keep a record of the autocorrelation functions of several series in a single run, you must give each one a unique filename. If the filename is omitted the information is written to fort.15.

review: Print out the current command list.

clear *command*: Remove all occurrences of the named command from the command list. Used with commands like **hold** or **logfreq** which cannot be modified through their arguments.

Annex

A primitive Prolate Spheroidal multitaper capability is included with *psd*. Then spectral resolution (which varies inversely with X, the time-bandwidth product) is traded against variance, which goes like one over the number of tapers selected. It is unproductive to take more than about 2X tapers, because higher order tapers offer poor leakage protection; setting tapers to X is safe. So the user is faced with specifying two relatively unfamiliar tuning parameters for this technique. Prewhitening is permitted. In my experience these tapers are most useful when used with short series (less than 1000 terms); for long time series, the ability to reduce variance severely and selectively gives the sine-taper approach a clear advantage.

prolate [*filename*]: when present, substitutes prolate spheroidal tapers for the sine tapers. A fixed number of tapers is applied to the record, and so spectral resolution is constant over the whole frequency interval, unlike the variable resolution of sine multitapers. This number can be set with **tapers** or it will default to 1.2 times the time-bandwidth product. See **time-bp**. The commands **smooth** and **adapt** apply only to the sine multitaper mode. If the *filename* is included, the complete set of tapers is written to that file with three numbers per line: j, n, psi(j,n), where n is the order of the taper, and j increasing by 1 per line within each order.

time-bp *X*: specify the time-bandwidth-product, which gives the frequency interval in which the spectrum is smoothed, and outside which energy is suppressed. X gives the number of frequency bins over which the spectrum is averaged, and also, approximately, the number of tapers averaged to make each estimate. Typical values are between 4 and 10. The averaging band can also be specified in frequency units with **bandwidth**. The parameter X must be set (or its equivalent in **bandwidth**) or the program will revert to sine multitaper mode. Large values of X lead to severe numerical instability in the computation of the prolate functions, so that heavy spectral smoothing and strong variance reduction cannot be performed.

bandwidth *B*: specify the frequency band over which the power spectrum will be averaged in the multitaper process. See also **time-bp**. This is an alternative way of specifying X. If both B and X are provided, the value of B will be preferred.

Notes

Like any other spectral estimation program, this one has too many potential knobs to turn. The first thing to do with the adjustable parameters is to ignore them all and run with everything at default. The result should be a fairly reliable estimate with variable resolution and variance as the program adjusts the number of tapers according to the local curvature in the spectrum. If the spectrum is very red (that is, power is concentrated at the low frequency end), prewhitening is worth exploring. If there are low bumps in the default spectrum, which could indicate important peaks, rerun the program with **smooth** set to 0.5 to enhance the resolution.

Unlike the prolate spheroidal multitapers, the sine multitaper 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 estimates the maximum and minimum values of the uncertainty and plots them as a pair of error bars on the graph. But complete information is contained in the output file as the corridor of 1-standard-deviation errors, and in K, the number of tapers used at each frequency. The errors are estimated in the simplest way, from the number of degrees of freedom (two per taper), not by jack-knifing. The frequency resolution is found from K*fN/Nf where fN is the Nyquist frequency and Nf is the number of frequencies estimated.

The adaptive process used is as follows. A quadratic fit to the log PSD within an adaptively determined frequency band is used to find an estimate of the local second derivative of the spectrum. This is used in an equation like R & S's eq (13) for the MSE taper number, with the difference that a parabolic weighting is applied with increasing taper order. Because the FFTs of the tapered series can be found by resampling the FFT of the original time series (doubled in length and padded with zeros) only one FFT is required per series, no matter how many tapers are used. This makes the program *fast*. Compared with the Thomson multitaper programs, this code is not only fast but simple and short. The spectra associated with the sine tapers are weighted before averaging with a parabolically varying weight. The expression for the optimal number of tapers given by R & S must be modified since it gives an unbounded result near points where S" vanishes, which happens at many points in most spectra. This program restricts the rate of growth of the number of tapers so that a neighboring covering interval estimate is never completely contained in the next such interval.

A mixed radix fast Fourier transform is used and so series are not truncated to the nearest power of two in length; rather, the series is reduced in length (if necessary) to the next largest even integer of composed of powers of 2, 3 and 5.

Spectra with sharp cutoffs and very deep valleys can be estimated with the prolate spheroidal tapers when the series is not too long to avoid bias from spectral leakage. Long data series naturally come with high resolution so that leakage is a less serious issue.