



AHELP for CIAO 3.4

apowerspectrum

Context: [tools](#)

Jump to: [Description](#) [Examples](#) [Parameters](#) [CHANGES IN CIAO 3.4](#) [CHANGES IN CIAO 3.0](#) [Bugs](#) [See Also](#)

Synopsis

Compute the power spectrum of an N-dimensional input array, or from two columns (independent/dependent variable) in an input file

Syntax

```
apowerspectrum infilereal infileimag outfile [pad] [center] [scale]
[crop] [clobber] [verbose] [kernel]
```

Description

The power spectrum is computed as the magnitude squared of the FFT of the input array or an ordered series input as a table array.

Example 1

```
apowerspectrum my_data.fits none my_out.fits crop=yes
```

Computes the powerspectrum of the data in the file my_data.fits and saves in my_out.fits. Since the input is real, the power spectrum is symmetrical about the 0 frequency point so the output is cropped at the Nyquist frequency.

Example 2

```
apowerspectrum my_r_data.fits my_i_data.fits my_out.fits pad=yes
```

Compute powerspectrum from complex input. Pad the data to the next power of 2 in length.

Example 3

```
apowerspectrum my_r_data.fits my_i_data.fits /tmp/. pad=yes
```

Similar to the previous example; now auto-naming is used to create the output file name, which will be

/tmp/my_r_pow.fits

Example 4

```
apowerspectrum my_r_data.fits my_i_data.fits "/tmp/[my_out]" pad=yes
kernel=iraf
```

Similar to above, but output files in /tmp/my_out.imh and /tmp/my_out.pix in IRAF format.

Example 5

```
apowerspectrum "my_ltc.fits[cols time,rate]" none my_out.fits
```

Computes the power spectrum of the rate column from the table array my_ltc.fits, and outputs the results to the fits file my_out.fits. This file contains two columns, INV_[COLUMN 1] (in this case, INV_TIME) giving the Fourier frequencies, and DATA, giving the power spectrum.

Example 6

```
apowerspectrum "my_events.fits[cols expno,counts]" none my_out.fits
```

The same as the above example; however, rather than calculating the PSD of rate as a function of time, counts as a function of expno (presuming the file has previously been created, for example, with dmextract) forms the basis of the PSD. If my_events.fits is properly created, the evenly spaced integer values of the expno column will avoid error messages concerning 'unevenly spaced bins' that can happen with the (relatively small) roundoff errors that often occur in the time column of real lightcurves. Furthermore, binning by exposure number (and using counts) is more likely to lead toward well-behaved PSD without aliasing of the Chandra frame times with the bin time of the lightcurve. As above, the output is placed in the file my_ltc.fits, and contains two columns, INV_EXPNO, which can be easily converted into frequency, and DATA, giving the power spectrum.

Parameters

name	type	ftype	def	min	max	reqd	autoname
<u>infile</u> real	file	input				yes	
<u>infile</u> imag	file	input				yes	
<u>outfile</u>	file	output				yes	yes
<u>pad</u>	boolean		no				
<u>center</u>	boolean		no				
<u>scale</u>	string		linear				
<u>crop</u>	boolean		no				
<u>clobber</u>	boolean		no				
<u>verbose</u>	integer		0	0	5		
<u>kernel</u>	string		default				

Detailed Parameter Descriptions

Parameter=infilereal (file required filetype=input)

Real component of the input.

The input is an image. The input image can have any number of dimensions. The input can have the following data type: "byte" (BITPIX=8), "short" (BITPIX=16), "long" (BITPIX=32), "float" (BITPIX=-32), and "double" (BITPIX=-64).

Currently, there MUST be a real part to the data, though the imaginary part can be "none".

The image can be a virtual image as defined by the datamodel, e.g. "my_file.fits[EVENTS][bin x=1:100:1,y=1:100:1]".

A table array must contain at least two columns. The first column, used to determine the Fourier frequencies, must be evenly spaced, and cannot contain any data gaps. In practice, roundoff errors in generating time bins may make this problematic, and the time column may have to be replaced with a uniformly spaced grid. This can be achieved, for example, via dmtcalc using the following command:

```
dmtcalc infile=my_ltc.fits outfile=my_fixed_ltc.fits expression="INT_TIME((long)#row)"
```

The new column, INT_TIME, now contains the integer value of the row number from the lightcurve, which is guaranteed to be evenly spaced (although information about data gaps is not retained). One can then create the psd via:

```
apowerspectrum infilereal="my_fixed_ltc.fits[cols in_time,total_counts]" outfile=my_psd.fits
```

The output file my_psd.fits contains the frequency column INV_INT_TIME, and the psd is found in the DATA column.

Parameter=infileimag (file required filetype=input)

The input file for the imaginary part of the data. This may be specified as "none" or "NONE" if there is no imaginary part. The infileimag parameter is not required for table array inputs.

Parameter=outfile (file required filetype=output autoname=yes)

Output file name.

The output data file is a "float" image with the power spectrum computed. The output is scaled according to the scale parameter.

The output format is controlled by the kernel parameter. The output will either be a FITS image (kernel="fits"), an IRAF .imh/.pix file (kernel="iraf"), or will default to whatever format the input is in (kernel="default").

The autoname filename suffix is "_pow". To completely specify the file name one should specify the file and extension name like: myfile.fits[foo] -- for fits or ./myfile.imh -- for iraf.

For the case of a table array input, the output file contains two columns, INV_[COLUMN1] giving the Fourier frequencies, and DATA, giving the psd. The latter is calculated as

$$\text{psd} = 1/N * (\text{SUM}_{\{x=0\}^{N-1}} f(x) \exp[-j 2 \pi u x/N])^2$$

where $f(x)$ is the discrete function of the column 1 independent variable, x , u is the Fourier frequency, N is the number of data points, and j is the imaginary unit. Thus, comparing to IDL v.5, for example, the psd here is equivalent to the square of the IDL forward transform times N , or the square of the IDL reverse transform divided by N .

Parameter=pad (boolean default=no)

If pad is set to yes, the data are padded in size to the next power of 2 in all dimensions.

Parameter=center (boolean default=no)

center controls whether the zero frequency point is at pixel=1 or pixel=N/2.

The center parameter controls whether the zero frequency point is at pixel=1 or pixel=N/2. Note: both center and crop cannot be set to "yes".

Parameter=scale (string default=linear)

Scaling of the data: (linear/log/db)

The scale parameter can take the following values.

- linear : linear scaling: output = [absolute value of FFT(a)]**2
- log : log (base 10) scaling: output = log(abs.[FFT(a)]**2)
- db : 10 * log() scaling: output = 10*log(abs.[FFT(a)]**2)

Note: For scale=log and scale=db, IEEE NaN's may be generated in the output files, log(0) = NaN. NB: NaN means "not a number".

Parameter=crop (boolean default=no)

crop controls whether entire power spectrum is output or just up to Nyquist frequency.

The crop parameter controls whether the entire power spectrum is output or just up to the Nyquist frequency (N/2). Note: both center and crop cannot be set to "yes".

Parameter=clobber (boolean default=no)

Clobber existing output files?

Parameter=verbose (integer default=0 min=0 max=5)

Verbosity level: 0 – no output, 5 – max display.

Parameter=kernel (string default=default)

kernel controls the output format.

The output format is controlled by the kernel parameter. The output will either be a FITS image (kernel="fits"), an IRAF .imh/.pix file (kernel="iraf"), or will default to whatever format the input is in (kernel="default").

CHANGES IN CIAO 3.4

IRAF QPOE Files

Support for IRAF QPOE files has been removed from CIAO.

CHANGES IN CIAO 3.0

As regards computing transforms of 2-D images, there are no changes in the functionality of 'apowerspectrum' in CIAO 3.0. The new change in CIAO 3.0 is the ability to calculate the PSD from two columns of an input table, as in the examples above. Again, users are cautioned that steps may have to be taken to ensure strict even spacing of the input "time" column.

Bugs

See the [bugs page for this tool](#) on the CIAO website for an up-to-date listing of known bugs.

See Also

sherpa

[tpsf](#), [tpsf1d](#)

tools

[aconvolve](#), [acrosscorr](#), [arestore](#), [csmooth](#)

The Chandra X-Ray Center (CXC) is operated for NASA by the
Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
<http://cxc.harvard.edu/ciao3.4/apowerspectrum.html>
Last modified: December 2006

