




---

 AHELP for CIAO 3.4

## csmooth

Context: [tools](#)

*Jump to:* [Description](#) [Examples](#) [Parameters](#) [Bugs](#) [See Also](#)

---

## Synopsis

Adaptively smooth a 2–D image

## Syntax

```
csmooth infile sclmap outfile outsigfile outscfile conmeth
conkerneltype sigmin sigmax sclmin sclmax sclmode [stepzero] [bkgmode]
[bkgmap] [bkgerr] [clobber] [verbose] [kernel]
```

## Description

`csmooth' adaptively smooths a two–dimensional image of integer counts since Poisson statistics are assumed with a circular Gaussian or Tophat kernel using the algorithm of Ebeling, White and Rangarajan (private communication); see the [Transformation, Convolution, and Deconvolution manual](#) for details.

Each pixel in the input image is smoothed on its "natural" scale, in the sense that the smoothing scale is increased until the total number of counts under the kernel exceeds a value that is determined from a preset significance and the expected number of background counts in the kernel area. The input image must contain raw, integer counts. Alternatively a map of smoothing scales can be supplied.

This tool smooths a two–dimensional image with a circular Gaussian or Tophat kernel of varying size. By default `csmooth' determines the smallest possible smoothing scale by convolving the raw image with a kernel whose size grows from a small initial value to the maximal allowable value (corresponding to the size of the raw image itself). For each kernel size (= smoothing scale), a significance is computed at each pixel location by comparing the total counts under the kernel to the expected background in the same area. `csmooth' computes the background expectation locally from the counts in an area surrounding the kernel. Alternatively an external background can be supplied. All pixels for which the significance exceeds the threshold value are smoothed at the current scale, and are excluded from subsequent smoothing steps at larger scales. Hence no pixel is smoothed twice which is a crucial requirement if the total counts in the image are to be preserved.

If the default background estimation procedure is used, the features in the resulting adaptively smoothed image (as apparent in, e.g., a contour plot) are locally significant at the level given in the output significance map accompanying the smoothed image. Alternatively, a map of smoothing scales to be used at each pixel location can be supplied by the user. In this case, features in the output image are in general not uniformly significant at any

level.

The convolution is performed using either a sliding cell or FFT algorithm. The sliding cell method is slow, but the kernel is normalized such that no assumption is needed about data beyond the field edges. FFT convolution is much faster and should be used for images larger than 128x128 pixel size. The FFT runs fastest on images with sizes  $2n$  (power of 2). However, the FFT creates artifacts if there are sources close to the field edges. FFT artifacts can be minimized by selecting the area in the input image such that no sources are close to the edges.

Unfortunately, there is no a priori way to know the maximum image size that csmooth can work on. As a benchmark, consider that for a 512x512 image, it typically needs more than about 100Mb, which will scale as the area of the region. This could include both physical memory or disk cache; the latter is of course much slower to use. The total size depends on the smoothing scales that are allowed (if one sets a limit or lets it run until the scale is as big as the input image). It is also somewhat machine dependent since some machines are eager to reclaim freed memory and re-allocate memory, while some are not.

'csmooth' is adapted from the original ASMOOTH routine written in IDL. The ASMOOTH algorithm was conceived and developed by Harald Ebeling, David White and Vijay Rangarajan. A detailed description of the algorithm and applications to X-ray imaging data can be found in: Ebeling H., White D.A., Rangarajan F.V.N. "ASMOOTH: A simple and efficient algorithm for adaptive kernel smoothing of two-dimensional imaging data", 2006, MNRAS, accepted ([astro-ph/0601306](#)).

The current version of 'csmooth' considers only positive deviations from the background level, i.e. emission features. Negative deviations (absorption features) are always smoothed at the largest possible scale.

For smoothing Chandra images see the [csmooth thread](#).

## Example 1

```
csmooth image.fits outfile=smoothed_img.fits
outsigfile=smoothed_sig.fits outscfile=smoothed_scl.fits sigmin=3
```

The data in image.fits are smoothed with a minimum S/N of 3; the adaptively smoothed output image is written to smoothed\_img.fits. There are two other output files created: smoothed\_fill\_sig.fits (outsigfile), a map of the significance of the signal in each pixel of the smoothed image, and smoothed\_fill\_scl.fits (outscfile), an image of the smoothing scales (kernel sizes) used at each location of the smoothed image.

## Example 2

```
csmooth test_data.fits outfile=outfile.fits outsigfile=outsigfile.fits
outscfile=outscfile.fits conmeth=slide sigmin=3 sigmax=5
conkerneltype=gauss verbose=1
```

The data in test\_data.fits are adaptively smoothed using a Gaussian kernel. Features in the output image outfile.fits that lie within the region marked by the sigma=3 contour in the outsigfile.fits image are significant at the 3 to 5 sigma confidence level. Pixels at greater than 5 sigma confidence level in the raw data remain unsmoothed. The significance is evaluated relative to the local background on the smallest scale that allows the significance threshold to be reached. These scales (the kernel sizes used in the smoothing process) are returned in outscfile.fits. Information about the progress of the smoothing process is printed to the screen.

## Parameters

name	type	ftype	def	min	max	units	reqd	autoname
<u>infile</u>	file	input					yes	
<u>sclmap</u>	file	input						
<u>outfile</u>	file	output					yes	yes
<u>outsigfile</u>	file	output	.					yes
<u>outsclfile</u>	file	output	.					yes
<u>conmeth</u>	string		fft				yes	
<u>conkerneltype</u>	string		gauss				yes	
<u>sigmin</u>	real		4				yes	
<u>sigmax</u>	real		5				yes	
<u>sclmin</u>	real		INDEF			pixels	yes	
<u>sclmax</u>	real		INDEF			pixels	yes	
<u>stepzero</u>	real		0.01			pixels	no	
<u>sclmode</u>	string		compute				yes	
<u>bkgmode</u>	string		local				no	
<u>bkgmap</u>	file	input					no	
<u>bkgerr</u>	file	input					no	
<u>clobber</u>	boolean		no				no	
<u>verbose</u>	integer		0	0	5		no	
<u>kernel</u>	string		default				no	

## Detailed Parameter Descriptions

**Parameter=infile (file required filetype=input)**

*Input file name: raw image*

A two-dimensional image of one of the following data types: "short" (BITPIX=16), "long" (BITPIX=32), "float" (BITPIX=-32), or "double" (BITPIX=-64). Except for the image dimensions, all header information is ignored.

If a scale map (sclmap parameter) is not supplied, the input image must contain integer counts (regardless of data type); i.e. no corrections such as background subtraction or corrections for exposure time variations have been applied.

The tool is best run on limited size arrays (less than about 512x512), otherwise it requires large amounts of memory and CPU times can become excessive.

Note that for ASCA and ROSAT data it is necessary to add the block name to the file specification, e.g. infile="ad57027000g2west.fits[image]" .

**Parameter=sclmap (file filetype=input default=)**

*File name of input image of smoothing scales to be used at each pixel location. Required if sclmode is set to "user".*

The image must be of the same dimensions as the data image.

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

*Output file name: adaptively smoothed image.*

The output file is an image of the same dimensions as the input image. The output data type is always "float" (BITPIX=-32). The output format is controlled by the kernel parameter. The autoname filename suffix is "\_asm".

**Parameter=outsigfile (file filetype=output default=. autoname=yes)**

*Output sigmap file name; sigmap is a map of the significance of the signal in each pixel of the smoothed image. Autoname suffix is "\_sig".*

The significance is computed using Gaussian statistics and taking into account the expectation value of the background in the kernel area. If not supplied by the user, the background expectation is computed from a local estimate obtained from the counts in an annulus surrounding the kernel.

See also information on infile parameter.

**Parameter=outscfile (file filetype=output default=. autoname=yes)**

*Output file name: image of the smoothing scales (kernel sizes) used at each location of the smoothed image. Autoname suffix is "\_scl".*

The smoothing scales are the smallest that allow the significance threshold to be reached. The scales are the sizes of the smoothing kernel used at any location of the image: standard deviation in the case of a Gaussian kernel, radius in the case of a tophat kernel.

See also information on infile parameter.

**Parameter=conmeth (string required default=fft)**

*Convolution method: (slide/fft [default]).*

Two convolution methods are implemented.

Sliding cell convolution (conmeth=slide) is convolution from first principles. The boundary conditions used with conmeth=slide renormalize the kernel whenever it extends beyond the boundaries of the image. While sliding cell convolution is slow it returns optimal results in the sense that no artifacts are introduced by possibly erroneous assumptions about data beyond the field edges.

conmeth=fft will cause a fast fourier transforms (FFT) of the data and the kernel to be computed. For large kernels, FFT convolution is much faster than sliding cell convolution. However, it requires much more memory and, internally, wraps the input image in both dimensions. Hence, signal at the field edges will spill over to the opposite edge in the output image.

**Parameter=conkerneltype (string required default=gauss)**

*Convolution kernel type. (Gauss/Tophat)*

If a Gaussian kernel is chosen, the term `smoothing scale' refers to the standard deviation of the Gaussian and the kernel array has dimensions of six standard deviations on the side (i.e., the Gaussian is computed out to three standard deviations in radius). For a tophat kernel, the smoothing scale is the radius of the tophat.

**Parameter=sigmin (real required default=4)**

*Minimal significance (S/N ratio) of the signal under the kernel.*

The minimal significance is translated into a minimal number of counts under the kernel via

$$\text{sigmin} = (\text{countsm} - \text{back}) / \sqrt{(\text{countsm\_err}^2 + \text{back\_err}^2)}$$

where back is the expected number of background events under the kernel. Unless supplied by the user, the background expectation is computed from a local estimate obtained from the counts in an annulus surrounding the kernel.

**Parameter=sigma (real required default=5)**

*Maximal significance (S/N ratio) of the signal under the kernel. Defaults to sigmin+1.*

The maximal significance is used to prevent oversmoothing of the data on larger scales.

**Parameter=sclmin (real required default=INDEF units=pixels)**

*Initial, minimal value of the smoothing scale, use INDEF for default (size of kernel array matches pixel size).*

**Parameter=sclmax (real required default=INDEF units=pixels)**

*Maximal value of the smoothing scale, use INDEF for default (size of kernel array matches image size).*

**Parameter=stepzero (real not required default=0.01 units=pixels)**

*Initial stepsize by which smoothing scale is increased.*

By default, CSMOOTH initially increases the initial smoothing scale by the value of stepzero (default = 0.01). As the smoothing proceeds, larger stepsizes may be chosen to minimize processing time.

**Parameter=sclmode (string required default=compute)**

*Controls whether CSMOOTH computes all smoothing scales internally based on the specified significance threshold (the default), or whether the smoothing scales are to be taken from a user-supplied map.*

If a map of predefined, smoothing scales is supplied by the user (through the sclmap parameter), the values of the parameters sigmin, sigma, sclmin, sclmax, and stepzero are ignored, and any features in the smoothed output image will, in general, not be significant at any uniform level.

**Parameter=bkgmode (string not required default=local)**

*Controls whether the background is computed locally from the data ("local") or whether a user-supplied background map is used ("user").*

By default, CSMOOTH computes the background from the input data in a region surrounding the kernel. Alternatively, an external background (and its associated error) can be specified by the user through the bkgmap and bkgerr parameters.

**Parameter=bkgmap (file not required filetype=input default=)**

*File name of background image. Required if bkgmode is set to "user".*

The image must be of the same dimensions as the data image. The units are assumed to be the same as in the data image.

**Parameter=bkgerr (file not required filetype=input default=)**

*File name of background error image. Required if bkgmode is set to "user".*

The image must be of the same dimensions as the data image. The units are assumed to be the same as in the data image.

**Parameter=clobber (boolean not required default=no)**

*Clobber existing output files?*

**Parameter=verbose (integer not required default=0 min=0 max=5)**

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

**Parameter=kernel (string not required default=default)**

*Kernel controls the output format.*

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").

## 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](#), [apowerspectrum](#), [arestore](#)

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/csmooth.html>  
Last modified: December 2006

Ahelp: csmooth – CIAO 3.4