Chandra X-ray Center

AHELP for CIAO 3.4   **kernel**   Context: sherpa

Jump to: Description Example Bugs See Also

Synopsis

Defines an expression to be used as a kernel in 1D source model convolution.

Syntax

```
sherpa> KERNEL [<dataset range> | ALLSETS [ID]] = <modelExpr>

<dataset range> = # (or more generally #:#,#:#, etc.) such that #
specifies a dataset number and #:# represents an inclusive range of
datasets; one may specify multiple inclusive ranges by separating them
with commas. The default dataset is dataset 1.
```

Description

The model expression, <modelExpr>, is an algebraic combination of one or more of the following elements:

```
{<sherpa_modelname> | <sherpa_modelname>[modelname] |
 <modelname> | <model_stack> | <nested_model>}
```

along with numerical values. The following operators are recognized: + − * / ( ) { }. See the CREATE command for further information.

Note that:

- The documentation on Sherpa Models contains a summary list, and descriptions, of the models that are available within Sherpa, which include models from XSPEC, v. 11.3.
- By default, if the model expression includes a model component that has not previously been established, Sherpa will prompt for the initial parameter values for that model component. This prompting can be turned off using the PARAMPROMPT OFF command.

To reset the kernel model stack, issue the command:

```
sherpa> KERNEL [<dataset range> | ALLSETS] =
```

The KERNEL model stack is used to convolve source model expressions. As such, it has many uses, of which two are the following:

- It can be used instead of an INSTRUMENT model stack if the instrumental response is representable

by analytic functions (e.g., by a Gaussian).

- It can be used to model circumstellar smearing of photon spectra; these spectra may then be further smeared by the observing instrument.

Note that the parameter values of a KERNEL model stack are frozen in fits; the kernel shape cannot be optimized.

The KERNEL model stack is evaluated on the same data grid as the SOURCE model stack. In general, the data grid will not be the correct grid for convolution; in the case of Gaussian smoothing, the convolved spectrum may exhibit, e.g., spectral lines that are both dilated and translated. To prevent translation, the user must set an origin for the convolution kernel using the command:

```
sherpa> KERNEL.ORIGIN [<dataset range> | ALLSETS] = <coordinate>
```

The example below will help illustrate this point.

Example

Define SOURCE and KERNEL model stacks; define the origin of the convolution kernel; and fit:

```
sherpa> SOURCE = GAUSS[modela]
modela.fwhm parameter value [19.3451]
modela.pos parameter value [804]
modela.ampl parameter value [126]
sherpa> KERNEL = NGAUSS[modelb]
modelb.fwhm parameter value [19.3451] 20
modelb.pos parameter value [804]
modelb.ampl parameter value [2594.62] 1:1:1
sherpa> KERNEL.ORIGIN = 804
sherpa> FIT
 LVMQT: V2.0
 LVMQT: initial statistic value = 578.346
 LVMQT: final statistic value = 64.8424 at iteration 6
       modela.fwhm  25.3426
       modela.pos  807.882
       modela.ampl  150.302
sherpa> KERNEL =
sherpa> FIT
 LVMQT: V2.0
 LVMQT: initial statistic value = 294.76
 LVMQT: final statistic value = 64.3803 at iteration 7
       modela.fwhm  32.3536
       modela.pos  807.863
       modela.ampl  117.826
```

The convolution kernel is a normalized Gaussian, with integrated area set to 1 and full–width half–max of 20. (For a normalized Gaussian, the integrated area equals the amplitude.) The kernel origin is set to be the centroid of the normalized Gaussian (modelb.pos). Note the differences in the values of modela.fwhm and modela.ampl between each fit.

Bugs

See the Sherpa bug pages online for an up–to–date listing of known bugs.

See Also

sherpa

autoest, background, create, create_model, createparamset, fit, freeze, get_defined_models,
get_model_params, get_models, get_num_par, get_par, get_stackexpr, getx, gety, guess, instrument,
integrate, is_paramset, jointmode, lineid, linkparam, mdl, modelexpr, modelstack, nestedmodel, noise,
paramprompt, paramset, pileup, rename, run_fit, set_par, set_paramset, set_stackexpr, source, thaw,
truncate, unlink

See Also