



AHELP for CIAO 3.4

## simspec

Context: [sherpa](#)

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

## Synopsis

Create and fit a simulated PHA spectrum.

## Syntax

```
simspec simmodel rmffile arffile bkgfile exposure backscale grouptype
grouptypeval normtype normval normparam normmin normmax fitmodel fitmin
fitmax paramest outfile mdlfile psplotfile plottype [simparams]
[fitparams] [usefile] [clobber] [verbose] [mode]
```

## Description

simspec is a command-line script that uses the Sherpa S-Lang module ('ahelp sherpa-module') to generate and optionally fit a simulated PHA spectrum. The script employs Sherpa's FAKEIT command ('ahelp fakeit') and automates the procedure described in the ["Simulating 1-D Data" thread](#). It is intended to simplify the process of running simulations (particularly for Chandra observation-proposal planning) and allows for script-driven generation of numerous spectra.

## INPUT AND OUTPUT

Input to simspec falls into three categories:

- simulation parameters (source model, RMF, ARF, background dataset, exposure time, background scale factor),
- normalization parameters (method, value, source model parameter, lower/upper bound for calculation), and
- fit parameters (source model, lower/upper bound of data filter, confidence-interval estimation method)

When run, the script prints fit results (best-fit parameters, goodness-of-fit statistics, confidence intervals) to the screen and produces three output files:

- a PHA file containing the simulated spectrum;
- a MDL file ('ahelp mdl') containing information on source/background datasets, source and instrument models, and source model parameters; and
- a postscript file containing a plot of the simulated data and/or fit (if requested)

For details on input and output parameters, see the PARAMETERS section below.

## SETTING SOURCE MODEL PARAMETERS

When running simspec, the user will most likely need to set the source–model parameters used in generating and fitting the simulated spectrum. By default, the script will prompt the user for all parameter values. However, it is also possible to set parameter values on the command line or use a text file that contains parameter settings. For details on the various methods available, see the description of the 'simparams' parameter in the section "DETAILED PARAMETER DESCRIPTIONS" below.

## SETTING SHERPA PARAMETERS

Sherpa settings such as optimization methods, statistics and plotting can be changed through the "usefile" parameter. This is a simple text file with Sherpa commands specified at each line. If "usefile" parameter is not set to NONE then simspec evaluates each line of this file (first as a Sherpa statement and then, if not recognized, as a S–lang statement). This evaluation happens before the models are set up. This can be used to do "anything" but is probably mostly–useful for commands like:

```
method powell
statistic cash
set_log
require("sherpa_plotfns");
```

## GROUPING SIMULATED DATA

simspec has the ability to group the simulated dataset before it is written out or fit. There are 2 parameters: "grouptype" and "grouptypeval". At present the only methods are NUM\_CTS, ADAPTIVE, SNR, and ADAPTIVE\_SNR.

## Example 1

### INTERACTIVE USAGE

If simspec is run with no command–line arguments, the script will prompt the user for all necessary input parameters (including source–model parameter values). This should make running the script a fairly intuitive process for first–time users.

The following example demonstrates how to respond to the input prompts in order to create and fit a basic spectrum. (For details on each of the input parameters, see the PARAMETERS section below.) The source model is a simple 1–D power law ("powlaw1d[p1]"). The model parameters will be set so that p1.gamma has a value of 2; p1.ref has all its default settings; and p1.ampl has its default value but with minimum and maximum changed to 1e–9 and 10, respectively. The simulation will use a RMF, an ARF, and a background dataset. The generated dataset will have an exposure time of 33483.2 sec and a background scale factor of 0.0441895. The data will be normalized so that the energy flux, integrated over the range 2 to 10 keV, is 1.0e–13 ergs/cm<sup>2</sup>/s. The fit will be performed using the same source model and parameter settings employed in the simulation, and parameter confidence intervals will be estimated using the COVARIANCE command. Finally, a plot showing data, best fit model, and sigma residuals will be created:

```
unix% simspec
Abundances set to Anders & Grevesse
Model for simulating data (): powlaw1d[p1]
```

## Ahelp: simspec – CIAO 3.4

```

Input RMF file (): rmf.fits
Input ARF file (): arf.fits
Input background data file (NONE): bkg.pha
Exposure time [s] (0:) (1): 33483.2
Background scale factor (0:) (1): 0.0441895
Normalization method (NONE|EFLUX|PFLUX|COUNTS|COUNT_RATE) (NONE): EFLUX
Normalization value (0:) (): 1.0e-13
Model parameter to use for normalization (): p1.ampl
Lower bound for normalization [keV] (0:) (): 2
Upper bound for normalization [keV] (0:) (): 10
Model for fitting data (NONE|SIM|<model_expr>) (): SIM
Lower bound for fit [keV] (0:) normmin -> 2):
Upper bound for fit [keV] (0:) normmax -> 10):
Confidence-interval estimation method (NONE|UNC|COV|PROJ) (NONE): COV
Output PHA file (): pha.fits
Output MDL file (): mdl.fits
Output postscript plot (NONE): plot.ps
Output plot type (DATA|FIT|FIT_DELCHI|FIT_RATIO) (FIT_DELCHI):

Enter simulation source model parameters:
p1.gamma parameter value [1] 2
p1.ref parameter value [1]
p1.ampl parameter value [1] :1e-9:10
Write X-Axis: Bin Y-Axis: Counts

Enter fit source model parameters:
p1.gamma parameter value [2]
p1.ref parameter value [1]
p1.ampl parameter value [3.87804e-05]

-----
FIT RESULTS
-----

LVMQT: V2.0
LVMQT: initial statistic value = 33.9046
LVMQT: final statistic value = 32.1615 at iteration 5
      p1.gamma  5.71548
      p1.ampl  0.000742059

Goodness: computed with Chi-Squared Gehrels

DataSet 1: 280 data points -- 278 degrees of freedom.
Statistic value      = 32.1615
Probability [Q-value] = 1
Reduced statistic    = 0.115689

Computed for sherpa.cov.sigma = 1

-----
Parameter Name      Best-Fit Lower Bound      Upper Bound
-----
      p1.gamma      5.71548 -0.994742      +0.994742
      p1.ampl      0.000742059 -0.00132777      +0.00132777

```

Once simspec has finished, the output MDL file can be read into Sherpa as follows:

```
sherpa> read mdl "mdl.fits"
```

This will load the simulated spectrum and background dataset, create the source and instrument models used in the fit, and set the source model parameters to their best-fit values.

## Example 2

### NON-INTERACTIVE USAGE

The above example can also be run non-actively by setting all input parameters on the command line (or, equivalently, setting them with the 'pset' command). In order to prevent simspec from prompting for model parameters, parameter values for the simulation and fit must be specified using the 'simparams' and 'fitparams' parameters, respectively:

```

unix% simspec simmodel="powlaw1d[p1]" rmffile=rmf.fits arffile=arf.fits \
? bkgfile=bkg.pha exposure=33483.2 backscale=0.0441895 normtype=EFLUX \
? normval=1e-13 normparam=p1.ampl normmin=2 normmax=10 fitmodel=SIM \
? fitmin=2 fitmax=10 paramest=COV outfile=pha.fits mdlfile=mdl.fits \
? psplotfile=plot.ps plottype=FIT_DELCHI \
? simparams="p1.gamma=2,p1.ampl=:1e-9:10" fitparams=SIM
    
```

For information on the syntax used for model setting parameters at the command line, see the descriptions of the 'simparams' and 'fitparams' parameters in the section "DETAILED PARAMETER DESCRIPTIONS" below.

## Parameters

name	type	ftype	def	min	max	units	reqd	stacks
<u>simmodel</u>	string						yes	
<u>rmffile</u>	file	input					yes	
<u>arffile</u>	file	input					yes	
<u>bkgfile</u>	file	input	NONE				yes	
<u>exposure</u>	real		1.0	0.0		s	yes	
<u>backscale</u>	real		1.0	0.0			yes	
<u>grouptype</u>	string		NONE				yes	
<u>grouptypeval</u>	real		0				yes	
<u>normtype</u>	string		NONE				yes	
<u>normval</u>	real			0.0			yes	
<u>normparam</u>	string						yes	
<u>normmin</u>	real			0.0		keV	yes	
<u>normmax</u>	real			0.0		keV	yes	
<u>fitmodel</u>	string						yes	
<u>fitmin</u>	real		)normmin	0.0		keV	yes	
<u>fitmax</u>	real		)normmax	0.0		keV	yes	
<u>paramest</u>	string		NONE				yes	
<u>outfile</u>	file	output					yes	
<u>mdlfile</u>	file	output					yes	
<u>psplotfile</u>	file	output	NONE				yes	
<u>plottype</u>	string		FIT_DELCHI				yes	
<u>simparams</u>	string						no	yes
<u>fitparams</u>	string						no	yes

<u>usefile</u>	string		none				no	
<u>clobber</u>	boolean		no				no	
<u>verbose</u>	integer		0	0	5		no	
<u>mode</u>	string		ql				no	

## Detailed Parameter Descriptions

### Parameter=simmodel (string required)

*Model for simulating data*

The source model expression that is used to create the simulated dataset. Any expression that is valid on the Sherpa command line is also valid here.

For information on setting model parameters, see the 'simparams' parameter.

### Parameter=rmffile (file required filetype=input)

*Input RMF file*

The RMF used when generating and fitting the simulated dataset. Note that a RMF is required; the script will fail unless one is provided.

### Parameter=arffile (file required filetype=input)

*Input ARF file*

The ARF used when generating and fitting the simulated dataset. The ARF is optional. If this parameter is empty or set to "NONE", then no ARF will be used.

### Parameter=bkgfile (file required filetype=input default=NONE)

*Input background data file*

Background data used when generating and fitting the simulated dataset. If a background file is provided, the background data are added to the simulated source data by the FAKEIT command. If this parameter is empty or set to "NONE", then no background data will be used.

### Parameter=exposure (real required default=1.0 min=0.0 units=s)

*Exposure time*

Exposure time for the simulated dataset. This value is recorded in the EXPOSURE header keyword of the output PHA file.

### Parameter=backscale (real required default=1.0 min=0.0)

*Background scale factor*

Background scale factor for the simulated dataset. This value is recorded in the BACKSCAL header keyword of the output PHA file.

**Parameter=grouptype (string required default=NONE)**

*Grouping type*

Type of grouping: one of NONE, NUM\_CTS, ADAPTIVE, SNR, ADAPTIVE\_SNR.

**Parameter=grouptypeval (real required default=0)**

*Grouping value*

The numerical value used for the grouping types NUM\_CTS, ADAPTIVE, SNR, ADAPTIVE\_SNR.

**Parameter=normtype (string required default=NONE)**

*Normalization method*

Method used to normalize the simulated data. The following options are available:

Method	Description
NONE	Do not normalize data
EFLUX	Normalize by energy flux (ergs/cm <sup>2</sup> /sec)
PFLUX	Normalize by photon flux (photons/cm <sup>2</sup> /sec)
COUNTS	Normalize by source (i.e. background-subtracted) counts (counts)
COUNT_RATE	Normalize by source (i.e. background-subtracted) count rate (counts/sec)

Note that if the COUNTS or COUNT\_RATE method is used with a background dataset, then the total counts or count rate in the output PHA file (over the range given by 'normmin' and 'normmax') will be the value specified by the 'normval' parameter plus the contribution from the background.

**Parameter=normval (real required min=0.0)**

*Normalization value*

Value to which the simulated data should be normalized. For example, if normtype=EFLUX and normval=1e-12, then the simulated data will be normalized so that the energy flux is 1e-12 ergs/cm<sup>2</sup>/sec over the range specified by 'normmin' and 'normmax'.

**Parameter=normparam (string required)**

*Model parameter to use for normalization*

Name of model parameter that represents the normalization of the simulation source model. For example, if simmodel="xsphabs[abs]\*powlaw1d[p1]", then 'normparam' should be set to "p1.ampl".

**Parameter=normmin (real required min=0.0 units=keV)**

*Lower bound for normalization*

The simulated dataset is normalized over the energy range given by 'normmin' and 'normmax'. For example, if normtype=EFLUX, normval=1e-12, normmin=2, and normmax=10, then the simulated data will be normalized so that the energy flux is 1e-12 ergs/cm<sup>2</sup>/sec over the range 2 to 10 keV.

If 'normmin' is less than the lowest energy in the dataspace of the simulated data, then the lowest energy in the

dataspace will be used as the lower bound for normalization.

**Parameter=normmax (real required min=0.0 units=keV)**

*Upper bound for normalization*

The simulated dataset is normalized over the energy range given by 'normmin' and 'normmax'. For example, if normtype=EFLUX, normval=1e-12, normmin=2, and normmax=10, then the simulated data will be normalized so that the energy flux is 1e-12 ergs/cm^2/sec over the range 2 to 10 keV.

If 'normmax' is greater than the highest energy in the dataspace of the simulated data, then the highest energy in the dataspace will be used as the upper bound for normalization. Setting 'normmax' to 0 will also cause the upper bound of the dataspace to be used. (Hence, to use the full energy range of the dataset, set both 'normmin' and 'normmax' to 0.)

**Parameter=fitmodel (string required)**

*Model for fitting data*

The source model expression that is used to fit the simulated dataset. Any expression that is valid on the Sherpa command line is also valid here. If 'fitmodel' is empty or set to "NONE", then no fit will be performed. If 'fitmodel' is set to "SIM", then the fit will be performed using the model that was used to simulate the data.

For information on setting model parameters, see the 'fitparams' parameter.

**Parameter=fitmin (real required default=)normmin min=0.0 units=keV)**

*Lower bound for fit*

Lower bound of energy filter used when fitting the simulated data. By default, 'fitmin' will have the same value as 'normmin'. If 'fitmin' is set to 0, then the filter will have no lower bound.

**Parameter=fitmax (real required default=)normmax min=0.0 units=keV)**

*Upper bound for fit*

Upper bound of energy filter used when fitting the simulated data. By default, 'fitmax' will have the same value as 'normmax'. If 'fitmax' is set to 0, then the filter will have no upper bound.

**Parameter=paramest (string required default=NONE)**

*Confidence–interval estimation method*

Method used to estimate confidence intervals for thawed source model parameters after fitting. The following options are available:

Method	Description
NONE	Do not estimate confidence intervals
UNC	Use UNCERTAINTY
COV	Use COVARIANCE
PROJ	Use PROJECTION

Note that confidence–interval estimates are displayed on screen but are not written to any output file.

**Parameter=outfile (file required filetype=output)***Output PHA file*

Output FITS file for the simulated spectrum. The header of the file will contain keywords indicating the background dataset, RMF, ARF, exposure time, and background scale factor used in simulating the data. The file will also contain parameter history for simspec, which can be extracted with dmhistory. (See 'ahelp dmhistory'.)

**Parameter=mdlfile (file required filetype=output)***Output MDL file*

Output MDL file for the simulation and fit. For a description of the MDL format, see 'ahelp mdl'. If a fit was performed, this file will contain the best-fit parameters. It will also contain parameter history for simspec, which can be extracted with dmhistory. (See 'ahelp dmhistory'.)

**Parameter=psplotfile (file required filetype=output default=NONE)***Output postscript plot*

Output postscript file containing a plot of the simulated data and/or fit. If 'psplotfile' is empty or set to "NONE", then no plot will be created.

**Parameter=plottype (string required default=FIT\_DELCHI)***Output plot type*

Type of output plot to create. The options are "DATA" (data only), "FIT" (data and best-fit model), "FIT\_DELCHI" (data and best-fit model in top pane, sigma residuals of fit in bottom pane), and "FIT\_RATIO" (data and best-fit model in top pane, data/model ratio in bottom pane). For more details on plot types, see 'ahelp lplot'.

Note that if 'fitmodel' is empty or set to "NONE", then only the data will be plotted, regardless of the setting of 'plottype'.

**Parameter=simparams (string not required stacks=yes)***Parameters for simulation model*

The method used to set simulation-model parameters (i.e. parameters for the source model specified by 'simmodel') depends on the value of 'simparams'. Four methods are available, each of which is described below.

**1. Using the default parameter values**

To use the default values for all source model parameters, set 'simparams' to "DEF".

**2. Entering parameter values interactively**

If 'simparams' is left empty, the script will prompt for parameter values. When prompted for a value, the user can accept the default (shown in square brackets); enter a new value; change the minimum, maximum, or initial step size (delta) for the parameter; and/or freeze or thaw the parameter. The syntax for entering these values is as follows ([ ]'s indicate an optional item, | indicates a choice between two items):



```
[<value>][:[:<min>][:[:<max>][:[:<delta>]]]][:,-1|1]
```

The fields <value>, <min>, <max>, and <delta> indicate the parameter value, minimum, maximum, and step size, respectively. A trailing ",-1" freezes the parameter, while ",1" thaws it. To use the default settings for all parameter properties, enter nothing (i.e. just hit <RETURN> at the prompt).

For example, if 'simmodel' is set to "xsphabs[abs]\*powlaw1d[p1]", then the following shows how one would respond to the parameter prompts in order to set abs.nh to 0.07 and freeze the parameter; set the value of p1.gamma to 2; keep the default settings for p1.ref; and change the minimum to 1e-9, maximum to 10, and step size to -2 for p1.ampl:

```
Enter simulation source model parameters:
abs.nh parameter value [0.1] 0.07,-1
p1.gamma parameter value [1] 2
p1.ref parameter value [1]
p1.ampl parameter value [1] :1e-9:10:-2
```

### 3. Specifying model parameters on the command line

In order to run the script non-interactively, one can specify parameter values on the command line. If 'simparams' is not empty or set to "DEF", its value will be interpreted as a stack of parameter expressions. For example, the following command-line fragment will set the same parameter values used in (2) above:

```
simspec simparams="abs.nh=(0.07,-1),p1.gamma=2,p1.ampl=:1e-9:10:-2" ...
```

Alternatively, one could use pset to store this value in the parameter file:

```
pset simspec simparams="abs.nh=(0.07,-1),p1.gamma=2,p1.ampl=:1e-9:10:-2"
```

Note that since no settings are given for p1.ref, the default values will be used. Also, since the specification for abs.nh contains a comma, the entire expression must be enclosed in parentheses. The reason for this is that without the parentheses, the comma between 0.07 and -1 will be interpreted as a stack separator, and -1 will be identified as an invalid model parameter expression. (For more information on stack syntax, see 'ahelp stack'.)

### 4. Specifying model parameters in an external file

Since the value of 'simparams' is treated as a stack expression, it is also possible to store the parameter settings in a file and use the stack library's "@<filename>" syntax. For example, if the file "simparams.stk" contains these lines:

```
abs.nh=0.07,-1
p1.gamma=2
p1.ampl=:1e-9:10:-2
```

then the following will set the model parameters as in (2) and (3) above:

```
simspec simparams="@simparams.stk" ...
```

Note that since each stack item is listed on a separate line, there is no need to wrap the abs.nh expression in parentheses.

## Parameter=fitparams (string not required stacks=yes)

### Parameters for fit model

The 'fitparams' parameter performs the same function for 'fitmodel' as 'simparams' performs for 'simmodel'. Usage of the parameter is identical to that of 'simmodel', with one additional option: If both 'fitmodel' and 'fitparams' are set to "SIM", then the script will use the parameter settings employed in creating the simulated dataset as the initial parameter values for fitting. (If only 'fitparams' is set to "SIM", then the script will prompt

for parameter values.)

Note that if 'fitmodel' is set to "SIM", then setting 'fitparams' to "DEF" is equivalent to setting it to "SIM" (i.e. the simulation source–model parameter values will be used).

**Parameter=usefile (string not required default=none)**

*File of Sherpa commands to run*

The 'usefile' parameter allows to input a list of Sherpa commands to simspec. If this is not set to NONE then the script evaluates each line of this file (first as a sherpa statement and then, if not recognized, as a S–lang statement). This evaluation happens before the models are set up. This parameter can be used to set for example Sherpa optimization methods, statistics, background subtraction or plotting option.

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

*Clobber existing output files?*

If 'clobber' is set to "yes" and the output files specified by 'outfile', 'mdlfile', and/or 'psplotfile' already exist, they will be overwritten with the new output. If 'clobber' is set to "no", the tool will quit with an error message if a file already exists.

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

*Verbosity level*

Amount of debugging output to display. Each successive verbosity level includes all the messages displayed at lower levels, plus the information listed in the following table:

Level	Additional Output
0	Fit results (best–fit parameters, goodness–of–fit statistics, and confidence intervals), plus a few insuppressible messages from the sherpa module (see NOTES below); Sherpa verbosity is set to 0 (no output)
1	Input parameters
2	Current model–parameter settings (interactive input only)
3	General information on the steps the script is performing
4	Detailed information on the steps the script is performing; updated model–parameter settings (interactive input only); Sherpa verbosity is set to 1 (normal output)
5	Highly–detailed output, including specifics of normalization calculations; Sherpa verbosity is set to 2 (maximum output)

**Parameter=mode (string not required default=q1)**

*Parameter–query mode*

If 'mode' is set to "h", the script will not prompt for any input parameters. (However, it will still prompt for source model parameters if 'simparams' or 'fitparams' is empty.) For more information, see the section "The mode parameter" in 'ahelp parameter'.

**NOTES**

## Part of CIAO Scripts Distribution

This script is not an official part of the CIAO release. Rather, it is contributed software that is distributed in the hope that it will be useful to a large number of users. For more information, see the [CIAO scripts page](#).

## Extra Output

The following two output lines are insuppressible and always printed, regardless of the verbosity level:

```
Abundances set to Anders & Grevesse
Write X-Axis: Bin Y-Axis: Counts
```

## Functions in sherpa\_simspec.sl

Most of the work of simspec is performed by S–Lang functions defined in the file "sherpa\_simspec.sl" (which is installed in the directory \$ASCDS\_INSTALL/contrib/bin/interpreted/). These functions can be made available in Sherpa or a script that has imported the sherpa module via the statement

```
() = evalfile("sherpa_simspec.sl");
```

For information on the available functions, see the comments in "sherpa\_simspec.sl".

## Bugs

## See Also

*chandra*

[guide](#)

*sherpa*

[bye](#), [calc\\_kcorr](#), [dataspace](#), [dcounts](#), [dollarsign](#), [echo](#), [eflux](#), [eqwidth](#), [erase](#), [flux](#), [get](#), [get\\_dcounts\\_sum](#), [get\\_dir](#), [get\\_eflux](#), [get\\_eqwidth](#), [get\\_filename](#), [get\\_flux2d](#), [get\\_flux\\_str](#), [get\\_lfactorial](#), [get\\_mcounts\\_sum](#), [get\\_pflux](#), [get\\_source\\_components](#), [get\\_verbose](#), [groupbycounts](#), [guess](#), [is](#), [journal](#), [list](#), [list\\_par](#), [mcounts](#), [numbersign](#), [paramest](#), [plot\\_eprof](#), [plot\\_rprof](#), [prompt](#), [reset](#), [run](#), [set](#), [set\\_analysis](#), [set\\_axes](#), [set\\_coord](#), [set\\_dataspace](#), [set\\_dir](#), [set\\_verbose](#), [setplot](#), [sherpa-module](#), [sherpa\\_plotfns](#), [sherpa\\_utils](#), [show](#), [use](#), [version](#)

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/simspec.html>  
Last modified: December 2006

