



AHELP for CIAO 3.4

## interval-projection

Context: [sherpa](#)

*Jump to:* [Description](#) [Examples](#) [CHANGES IN CIAO 3.2](#) [Bugs](#) [See Also](#)

### Synopsis

Plots the fit statistic as a function of parameter value, using the PROJECTION algorithm. The commands INT-PROJ and INTPROJ are abbreviated equivalents.

### Syntax

```
sherpa> INTERVAL-PROJECTION [<dataset range> | ALLSETS] <arg>
```

where <dataset range> = #, or more generally #:#,#:#,..., 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 is to create plots using data from all appropriate datasets.

### Description

The command-line argument may be:

#### INTERVAL-PROJECTION Command Argument

Argument	Description
<sherpa_modelname>.{<paramname>   <#>}	A specified model component parameter (e.g., GAUSS.pos).
<modelname>.{<paramname>   <#>}	A specified model component parameter (e.g., g.pos).

The user may configure INTERVAL-PROJECTION via the Sherpa state object structure intproj. The current values of the fields of this structure may be displayed using the command print(sherpa.intproj), or using the more verbose Sherpa/S-Lang module function list\_intproj().

The structure fields are:

#### intproj Structure Fields

Field	Description
fast	If 1, use a fast optimization algorithm (LEVENBERG-MARQUARDT or SIMPLEX) regardless of the current METHOD. If 0, use the current METHOD.
expfac	A multiplicative factor that expands the grid limits estimated by the COVARIANCE algorithm, if the grid limits are determined automatically (see arange, and below).
arange	If 1, the grid limits are to be determined automatically. If 0, the grid limits are specified (see min and

	max).
min	Specifies the grid minimum. This is always a linear quantity, regardless of the setting of log (see below). The setting is ignored if arange = 1.
max	Specifies the grid maximum. This is always a linear quantity, regardless of the setting of log (see below). The setting is ignored if arange = 1.
log	Specifies whether to use a linear (0) or logarithmic (1) spacing of grid points.
nloop	Specifies the number of grid points.
sigma	Specifies the number of sigma (i.e., the change in statistic) for the plot.

Field values may be set using directly, e.g.,

```
sherpa> sherpa.intproj.arange = 0
```

NOTE: strict checking of value inputs is not done, i.e., the user can errantly change arrays to scalars, etc. To restore the default settings of the structure at any time, use the Sherpa/S-Lang module function `restore_intproj()`.

The plot is created by varying each selected parameter's value on the determined (`arange = 1`) or specified (`arange = 0`) grid, and computing the best-fit statistic at each grid point. `INTERVAL-PROJECTION` differs from `INTERVAL-UNCERTAINTY` in that all other thawed parameters are allowed to float to new best-fit values, instead of being fixed to their best-fit values. This makes a plot created by `INTERVAL-PROJECTION` a more accurate rendering of the projected shape of statistical hypersurface, but but causes the computation to proceed more slowly. For a fuller theoretical description of error estimation, see `PROJECTION`, `UNCERTAINTY`, and `COVARIANCE`.

If `arange = 1`, then the grid limits for the plot are determined automatically using the `PROJECTION` algorithm. The selected parameter's value is varied until the fit statistic is increased by `delta_S`, which is a function of `INTERVAL-PROJECTION.sigma` (e.g., `delta_S = 1` if the statistic is chi-square and `sigma = 1`).

The grid-point values and best-fit statistics at each grid point may be retrieved using the Sherpa/S-Lang module function `get_intproj`. See the examples below.

## Example 1

List the current and default values of the `intproj` structure, and restore the default values:

```
sherpa> sherpa.intproj.arange = 0
sherpa> sherpa.intproj.log = 1
sherpa> sherpa.intproj.sigma = 5
sherpa> list_intproj()
Parameter      Current      Default      Description
-----
fast           1             1      Switch to LM/simplex: 0(n)/1(y)
expfac        3             3      Expansion factor for grid
arange        0             1      Auto-range: 0(n)/1(y)
min           0             0      Minimum value
max           0             0      Maximum value
log           1             0      Log-spacing: 0(n)/1(y)
nloop        20            20      Number of grid points
sigma         5             1      Number of sigma
sherpa> restore_intproj()
sherpa> list_intproj()
Parameter      Current      Default      Description
-----
fast           1             1      Switch to LM/simplex: 0(n)/1(y)
expfac        3             3      Expansion factor for grid
arange        1             1      Auto-range: 0(n)/1(y)
```

min	0	0	Minimum value
max	0	0	Maximum value
log	0	0	Log-spacing: 0(n)/1(y)
nloop	20	20	Number of grid points
sigma	1	1	Number of sigma

## Example 2

Plot chi-square within the 3-sigma confidence interval for a fit:

```
sherpa> READ DATA example1.dat
sherpa> PARAMPROMPT OFF
sherpa> SOURCE = POLYNOM1D[my]
sherpa> THAW my.c1 my.c2
sherpa> my.c0.min = -10
sherpa> FIT
...
sherpa> sherpa.intproj.sigma = 3
sherpa> INTERVAL-PROJECTION my.c0
Interval-Projection: computing grid size with covariance...done.
                    outer grid loop 20% done...
                    outer grid loop 40% done...
                    outer grid loop 60% done...
                    outer grid loop 80% done...
```

## Example 3

Plot chi-square for the same fit within manually set grid limits:

```
sherpa> sherpa.intproj.arange = 0
sherpa> sherpa.intproj.min = -25
sherpa> sherpa.intproj.max = 25
sherpa> INTERVAL-PROJECTION my.c0
Interval-Projection: grid size set by user.
                    outer grid loop 20% done...
                    outer grid loop 40% done...
                    outer grid loop 60% done...
                    outer grid loop 80% done...
```

## Example 4

Save the results of INTERVAL-PROJECTION to an ASCII file:

```
[...run INTERVAL-PROJECTION...]
sherpa> my_var = get_intproj()
sherpa> writeascii("my_output.dat",my_var.x0,my_var.y)
sherpa> quit
Goodbye.
unix> more my_output.dat
-25      34.2524
-22.3684    27.5464
-19.7368    21.5764
...
```

## CHANGES IN CIAO 3.2

Prior to CIAO 3.2 the INTERVAL-PROJECTION command could not be used until the dataset had been fit. This was done to ensure that the parameter values were at their best-fit location, but caused problems when fitting multiple datasets or loading previously-saved analysis sessions. This restriction has now been removed. Please note that the results of INTERVAL-PROJECTION may not be valid unless the parameters are at their best-fit values.

## Bugs

See the [Sherpa bug pages](#) online for an up-to-date listing of known bugs.

## See Also

*sherpa*

[berrors](#), [bsyserrors](#), [compute\\_errors](#), [compute\\_statistic](#), [covariance](#), [errors](#), [ftest](#), [get\\_paramest](#), [get\\_paramestint](#), [get\\_paramestlim](#), [get\\_paramestreg](#), [goodness](#), [interval-uncertainty](#), [list\\_paramest](#), [mlr](#), [projection](#), [region-projection](#), [region-uncertainty](#), [restore\\_paramest](#), [run\\_paramest](#), [run\\_paramestint](#), [run\\_paramestlim](#), [run\\_paramestreg](#), [set\\_errors](#), [set\\_syserrors](#), [staterrors](#), [syserrors](#), [uncertainty](#)

---

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/interval-projection.html>  
Last modified: December 2006