`conjgradinv` – how to use it

version 0, July 2007

This program is an analogy of `convexinv`. It also computes a model that gives the best fit to the input lightcurves. The main difference between `convexinv` and `conjgradinv` is that with `conjgradinv`, rotation and scattering parameters are fixed and only the shape is optimized. The idea is that once you have a good model from `convexinv`, you can use the rotation parameters and derive a 'nicer' shape model using `conjgradinv`. The reason why `conjgradinv` usually produces models that look better and give slightly better fits than those produced by `convexinv` is that `conjgradinv` uses directly facet areas as parameters and not spherical harmonics representation as `convexinv`. It serves as a final 'polishing' tool in the inversion process.

It was written by Josef Ďurech, it uses routines from Numerical Recipes (conjugate gradient method) and algorithms developed in Fortran by Mikko Kaasalainen.

Do not use `conjgradinv` unless you are familiar with `convexinv`!

syntax (Unix):

```
cat lcs | conjgradinv [-v] [-s] [-o out_areas] input_par input_rot_par out_lcs
```

`-v` verbose mode
`-s` puts output areas to the standard output; do no use it together with `-v`

When run with the `-v` option, the program writes down values of $\chi^2_{\rm rel}$, rms deviation $\sqrt{\chi^2_{\rm rel}/N}$ (where N is the number of data), and the area of the dark facet.

Input lightcurves (`lcs`)

The input lightcurve file has to have the same format as for `convexinv`.

Input parameters (`input_par`)

Resolution – The number n of triangulation rows per octant (typically 8–10). The number of surface areas of the Gaussian image is $8n^2$.

Number of iteration steps – The iteration loop ends after a given number of steps (a typical value is ~ 100).

Convexity regularization weight – There is a dark facet area that makes the whole set of facets convex. Always try to put it below 1% of the total area by increasing the convexity regularization parameter.

Output lightcurves (`out_lcs`)

The brightness (in intensity units) of the model is stored in this file. The file contains a list of brightness values in the same format as in `convexinv`.

Output areas and normals (`out_areas`)

The same format as in `convexinv`. If you use the `-s` option, the list of facets and normals is put to the standard output – it is useful when using a pipeline. Do not use `-s` and `-v` at the same time.

Input rotation parameters (`input_rot_par`)

This file contains the solution for the spin vector direction, period, and scattering parameters in the format:

$\lambda\ \beta\ P$
$t_0\ \phi_0$
$a\ d\ k$
c

Use the `out_par` file from `convexinv`.

Creating a 3D shape (`minkowski`) and triangulation (`standardtri`)

Follow instructions in the `convexinv` manual.

```
cat lcs | conjgradinv -s input_par input_rot_par out_lcs | minkowski | \
   standardtri > model
```

Test

To run the test type:

```
cat test_lcs_rel | conjgradinv -v input_conjgradinv out_par out_lcs_cg
```

where `out_par` is the parameter file obtained from `convexinv`.

Further information

Look at `http://www.rni.helsinki.fi/~mjk/asteroids.html`, read Kaasalainen and Torppa (2001), Kaasalainen et al. (2001), and FAQ.

Updated versions may appear at `http://astro.troja.mff.cuni.cz/projects/asteroids3D`.