`convexinv` – how to use it

version 0, June 2006

The program (written in Fortran by Mikko Kaasalainen and converted to C by Josef Ďurech) computes the shape+spin+scattering model that gives the best fit to the input lightcurves (calibrated, uncalibrated, or sparse). Scattering law+shape representation are simple and robustly converging. The shape representation this procedure obtains is the Gaussian image of a convex polyhedron, i.e., the areas of the facets (with fixed outward normals). The vertices are directly solved by the Minkowski procedure `minkowski`.

syntax (Unix):

```
cat lcs | convexinv [-v] [-s] [-o out_areas] [-p out_par] input_par out_lcs
```

`-v` verbose mode
`-s` puts output areas to standard output; do not 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 λ parameter of the Levenberg-Marquardt algorithm. The relative chi-square is defined as

$$\chi^2_{\rm rel} = \sum_i \left\| \frac{\boldsymbol{L}^{(i)}_{\rm obs}}{\bar{L}^{(i)}_{\rm obs}} - \frac{\boldsymbol{L}^{(i)}}{\bar{L}^{(i)}} \right\|^2 ,$$

where $\boldsymbol{L}^{(i)}_{\rm obs}$ and $\boldsymbol{L}^{(i)}$ are observed and modelled lightcurves and they are renormalized through the average brightness $\bar{L}^{(i)}_{\rm obs}$ and $\bar{L}^{(i)}$. At the end, the solution of the pole, period and scattering parameters is written, together with the dark facet area (see below).

Input lightcurves (`lcs`)

The input file contains lightcurve data and the corresponding geometry, it is read from the standard input. The first line gives the total number of lightcurves, then the individual lightcurves follow in 'blocks'. Each lightcurve starts with the number of points and 0/1 code for a relative (0) or calibrated (1) lightcurve. Then there are lines with the epoch in JD (*light-time corrected!*), the brightness in *intensity* units (reduced to unit distances from the Earth and the Sun when calibrated), the ecliptic *astrocentric* cartesian coordinates x, y, z of the Sun and of the Earth in AU.

For a slowly moving main belt asteroid, the coordinate vectors can be approximated as to be constant for a single-night lightcurve.

Input parameters (`input_par`)

Initial spin – asteroid's initial ecliptic pole coordinates λ, β (deg), and the rotation period P (hours). The initial values are followed by 0 or 1 depending on whether those are fixed (0) or free (1).

Zero time t_0 (JD), *initial rotation angle* ϕ_0 (deg) – they are needed for the transformation between vectors $\boldsymbol{r}_{\rm ast}$ in the asteroid co-rotating coordinate frame and vectors $\boldsymbol{r}_{\rm ecl}$ in the ecliptic coordinate frame. The transformation is given by the equation

$$\boldsymbol{r}_{\rm ecl} = \mathsf{R}_z(\lambda)\,\mathsf{R}_y(90^\circ - \beta)\,\mathsf{R}_z\left(\phi_0 + \frac{2\pi}{P}(t - t_0)\right)\boldsymbol{r}_{\rm ast},$$

where $\mathsf{R}_i(\theta)$ is the rotation matrix corresponding to the rotation of the angle θ along the i-axis in the anticlockwise direction, t is the time, and P is the rotation period.

If $t_0 \leq 0$ in the input file, then it is set to the lowest JD epoch of the data set. So the basic choice is $t_0 = 0$ and $\phi_0 = 0$.

Convexity regularization weight – this is often needed to keep the shape formally convex. A typical value is 0.1, but it may often have to be orders of magnitude larger or smaller. 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. If you cannot get a good fit with a small dark area and if the fit significantly improves when the dark facet is large, it means that there is likely to be an albedo variation over the surface.

Laplace series expansion – degree l, and order m. It affects the number of shape parameters – if $l = m$, the number of shape parameters is $(l+1)^2$ (and $(l+1)^2 - 1$ for relative lightcurves). A good choice is $l = m = 6$.

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

Light scattering parameters – amplitude a, width d, slope k, and Lambert's coefficient c. Initial values are followed by 0/1 code for fixed (0) or free (1) parameters. The exponential-linear formulation of the phase function in the form

$$f(\alpha) = a \exp\left(-\frac{\alpha}{d}\right) + k\alpha + 1$$

is used in the code. The complete scattering model is

$$S(\mu, \mu_0, \alpha) = f(\alpha)[S_{\rm LS}(\mu, \mu_0) + c\, S_{\rm L}(\mu, \mu_0)],$$

where α is the solar phase angle, μ_0, μ are cosines of angles of incidence and reflection respectively, $S_{\rm L}$ is Lambert's law

$$S_{\rm L} = \mu\mu_0$$

and $S_{\rm LS}$ is the Lommel-Seeliger law

$$S_{\rm LS} = S_{\rm L}/(\mu + \mu_0).$$

Start with, e.g., $a = 0.5$, $d = 0.1$, $k = -0.5$. If you use only relative lightcurves, no solar phase function is needed (i.e., relative fit with almost constant solar phase for each lightcurve), then use fixed values of phase function parameters with zero amplitude and slope. Fitting the phase parameters using relative data leads to divergence. The parameter c has usually only little effect on the solution so you can fix it at, e.g., $c = 0.1$.

Iteration stop condition – if it is an integral number higher than one, then it is the number of iteration steps in the Levenberg-Marquardt loop. If it is lower than one, then it is the smallest difference in rms deviation between two subsequent steps – when the steps have smaller difference, the iteration loop is stopped.

Output lightcurves (`out_lcs`)

In this file the brightness (in intensity units) of the model is stored. The file contains a list of brightness values in the same order as in the input lightcurves file. One can easily check the fit visually by plotting the observed lightcurves together with the modelled brightness.

Output areas and vertices (`out_areas`)

The first line gives the number of facets, then follow facet areas and outward unit normal x, y, z coordinates. This is used as an input for `minkowski` procedure that creates a 3D model (see below). Note that the program writes the size and normal coordinates of a dark facet at the end, so the number of facets is $8n^2 + 1$. The dark facet should be very small (control with convexity regularization parameter). It is needed to make the collection convex, but it does not contribute photometrically.

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.

Output parameters (`out_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

If some of the parameters are fixed (t_0 and ϕ_0 are always fixed), they have the same values as in the `input_par` file.

Creating a 3D shape (`minkowski`)

The shape model is fully described by its Gaussian image (`out_areas`) that is used as an input for the `minkowski` procedure that creates a convex polyhedron. It reads the standard input and puts the result to the standard output. The format of the output polyhedral model is as follows: the first line gives the number of vertices and facets, then follow the vertex x, y, z coordinates, then for each facet the number of vertices and the order numbers of facet vertices (anticlockwise seen from outside the body). The Minkowski problem is solved iteratively, so it takes some time (minutes).

Triangulation (`standardtri`)

For some purposes it is useful to convert the polyhedron from `minkowski` to a polyhedron that has all facets triangular. The `standardtri` reads the standard input (the output of `minkowski`), creates a polyhedron with triangular facets and puts it to the standard output. The format is the same as for `minkowski`, but the number of vertices for each facet is omitted – it is equal to three for all facets. You can use all three programs at once:

```
cat lcs | convexinv -s input_par out_lcs | minkowski | standardtri > model
```

Finding the best period (`period_scan`)

The most important step on your way to get a model is to find the *correct period*. The smallest separation ΔP of local minima in the period parameter space is roughly given by

$$\Delta P \approx 0.5 \frac{P^2}{\Delta t},$$

where Δt is the full epoch range of the dataset. This derives from the fact that the maxima and minima of a double-sinusoidal lightcurve for periods P and $P \pm \Delta P$ are at the same epochs after Δt time.

The `convexinv` program starts from initial values, finds the local minimum of χ^2 and gives the corresponding shape and spin solution. To be sure that you didn't miss the global minimum in period, it is *always absolutely necessary* to scan the whole interval of possible periods (there is usually some guess based on the synodic period and its error) before running `convexinv`. To do this, run `period_scan`

```
cat lcs | period_scan [-v] input_period_scan out_periods
```

The first line of the `input_period_scan` gives the initial period, the final period, and the coefficient p of the period step. The interval is scanned with the step $p\,\Delta P$ (always set $p < 1$, a recommended value is $p = 0.8$). Other input parameters have the same meaning as in the `input_par` file. The last line gives the minimum number of iterations if the iteration stop condition is smaller than one.

The procedure starts with six initial poles for each trial period, selects the period that gives the lowest χ^2 and puts it to the `out_periods` file. This file contains the period solution, rms deviation, χ^2, the number of iteration steps, and the dark facet area (%). The number $N_{\rm per}$ of trial periods within the period range $[P_0, P_1]$ is

$$N_{\rm per} = \frac{2\Delta t(P_1 - P_0)}{P_0 P_1} \frac{1}{p},$$

so it takes some time (up to several hours for a long interval) to scan over the whole interval $[P_0, P_1]$.

If there is a clear minimum in χ^2 when plotted as a function of period, use this best period as the input value for `convexinv`. Run `convexinv` with different initial poles (five to ten is usually enough) randomly distributed over the sphere. If there is only one pole solution that gives significantly lower χ^2 than all others, you have a unique solution. There are always two possible poles with the same β and $\lambda \pm 180^\circ$ for asteroids orbiting near the plane of ecliptic.

It may happen that there is no clear minimum in the χ^2 vs. period plot or that there are many poles giving the same residual – it means that there is not enough data for a unique period+pole+shape model.

Test

To run the test type:

```
cat test_lcs_rel | convexinv -v -o out_areas -p out_par input_convexinv out_lcs
```

and

```
cat test_lcs_rel | period_scan -v input_period_scan out_periods
```

There are lightcurves of asteroid (43) Ariadne in `test_lcs_rel` (all lightcurves are relative) and `test_lcs_abs` (some are calibrated) files. The best solution giving rms residual 0.016 is $\lambda = 253°$, $\beta = -15°$, and $P = 5.761985$ hours. The best solution for calibrated data gives slightly different spin and rms residual 0.019 (after more then 100 iteration steps).

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`.

Good luck!