

Circumbinary disk analyser - **CirBinDis** (?)

Paul Magnus Sørensen-Clark
Jerome Bouvier

March 15, 2015

1 Introduction

A small piece of software for receiving an artificial light curve from a simulated density map of a gas disk around a binary star.

2 Installing

The source code is available at this Bitbucket repository:

<https://bitbucket.org/paulmag/circumbinarydisk>

Provided that Git is installed on your computer you can easily get all the files by running the following command inside the folder where you want the repository:

```
git clone https://paulmag@bitbucket.org/paulmag/circumbinarydisk.git
```

This link will be updated if the location of the repository or the installation process in any other way changes (note that at this moment the repository is private). Should it not work then contact the authors via email.

We recommend to make the alias `cirbindis` for the command

```
python ~/path_to_repository_folder/circumbinarydisk/src/main.py.
```

F. ex. place this in your `.bashrc` or `.bash_aliases`:

```
alias cirbindis="python ~/GitHub/circumbinarydisk/src/main.py"
```

This alias will be assumed for the rest of this manual.

3 Preparing your data

The format of the input data must be an ASCII/CSV-file with three columns where each line represents a datapoint in space. The two first columns of each line represent the position of a datapoint. (x, y) if using cartesian coordinates and (r, θ) if using polar coordinates. The last column represents the density in this position.

Any units can be used for the input data. How to specify units are covered in the section **Configuring and running CirBinDis**.

(It is currently possible to have four instead of three. Then the three first columns represent (x, y, z) or (r, θ, z) . The use of 3D-datapoints is currently an obsolete feature that will probably be completely removed.)

4 Processing algorithm

`CirBinDis` produces artificial lightcurves by analysing the provided dataset according to given configurations. In this section the process for extracting the lightcurve from the dataset is explained.

4.1 Loading data

TODO

4.2 Cropping

The space covered by the dataset may represent a larger area than the disk you want to analyse. The dataset is cropped to an inner and outer radius such that the shape of the remaining datapoints resembles a donut. The outer radius represents the size of the disk and makes sure that the disk is circular. The inner radius is necessary to avoid treating the stars themselves as dust, and the density of the dust is very low close to the stars anyway.

4.3 Rotating

The coordinates of all datapoints are rotated stepwise with the rotation matrix R_z for $\theta = [0, 360)^\circ$.

$$R_z = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This rotation simulates the physical orbital rotation of the dircumbinary disk. The reason we get a variation in the lightcurve is because when the disk rotates we see the stars through different areas of the disk with different densities.

4.4 Cylinder

A section of the datapoints are cropped out, which represents only the cylinder of gas that is between an observer on Earth and the star. These are the datapoints that fall within the cylinder whose base area is defined by the stellar surface and which extends from the stellar surface and infinitely along the x -axis in positive direction (the de facto limit is the outer radius of the disk). In other words, the observer's position is assumed to be $(\infty, 0, 0)$. A cylinder like this is made once for *each* azimuthal rotation of all the points. Thus, each cylinder will be a little different from the previous one (if $d\theta$ is small). If there are two (or even more) stars a cylinder will be created for the line of sight of each star, so there can be two (or even more) cylinders at the same time.

4.5 Binning

TODO

4.6 Integrating

TODO

5 Configuring and running CirBinDis

How to make necessary configurations and then run `CirBinDis` to perform an analysis.

This is the most practical and maybe the most important section, as it explains how to actually use the software.

5.1 Input parameters

The input parameters for each run of `CirBinDis` is configured in an XML file. Inside the repository you will find `/xml/input.xml`. Copy this file to the path where you will run `CirBinDis` from and change the value of the fields as required (it must still be called `input.xml`). Specifically, this is where you provide the filename of the dataset to analyse.

TODO: Here I plan to more or less copy the explanations that are in `input.xml` already.

unit-mass Dorem

unit-distance ipsum

unit-intensity dolum

5.2 Executing the code

TODO

6 Bibliography

7 Acknowledgments