MockGals

Make mock stars and galaxies in a FITS image with noise.
Manual for version 0.1.

Mohammad Akhlaghi

This manual is for `mockgals`, a program to make mock astronomical objects in a FITS image and add the appropriate noise.

Table of Contents

1 Quick start

Once `mockgals` is installed, if you simply run `./mockgals` in the installed directory, 45 mock galaxies and 5 stars will be randomly positioned in a `FITS` image of size $200 * 200$.

2 An introduction.

Making mock galaxies is very important in the process of understanding our data. `mockgals` was initially made with this exact intent. Certain astronomical targets, for example elliptical galaxies, are very sharp in their central regions, this makes a simple calculation of the profile in the center of each pixel unrealistic for such cases. The main advantage of `mockgals` is that it integrates the central parts of profiles until a given accuracy. It does this without any sorting or ordering and in a very fast manner.

A summary of the advantages of `mockgals` includes:

1. Integration of the center of the profile.
2. Very efficient in CPU usage, resuling in a very fast processing.
3. Written in the C programming language, which is easy to understand and modify or contribute to by any interested user.
4. [To be added] Can make profiles in any dimentions.

3 Installation

`mockgals` relys on only 3 packages: `GSL` (for mathematical functions), `FFTW` (for convolution) and `cfitsio` (for reading to and from FITS files).

4 Configuring mockgals.

Currently configuration paramters into `mockgals` can only be given through single letter commandline options. The options are explained below.

5 Mock paramters

The parameters of the profiles you want to build into a FITS image are fed to it though a table. In this section we will review how `mockgals` reads that table and how best to prepare it.

6 How mockgals works

In this section we will give a complete review on how `mockgals` works and how this is implemented in its coding in order to facilitate your reading of the code and possibly modifying it and making it better.