Andean Cosmology School 2015
Practice session: lecture 2

Universidad de los Andes, Bogotá, Colombia

Introduction

The goal of this problem set is to get you familiarized with `GalSim` and `SExtractor`.

- References for `GalSim`: "`SExtractor` for Dummies", and the `SExtractor` manual.

- References for `GalSim`: "Quick Guide", and the `GalSim` paper (Rowe et al. (2014))

1. Basic Profiles in Galsim

(a) Create circular Gaussian, Exponential, De Vaucouleurs, Airy, and Moffat profiles.

(b) Rotate and shear each one of them by 45 deg and $\vec{e} = (0.2, 0.3)$, respectively.

(c) Convolve each one of them with a circular, Gaussian profile (a PSF).

(d) Draw each profile into a `galsim.Image` object, and print it to disk in a `fits` file. You may choose to create a multi-extension fit (one extension per profile), or to place the profiles in a grid in a single, large `fits` image.

(e) Add read noise and Poisson noise to the image.

(f) Look at your files in `DS9`. Play around with some parameters such as contrast, scale, etc.

For each one of the profiles above, choose parameters that you consider appropriate (*e.g.*, $\sigma = 0.8$ arcsec for a Gaussian). Also, feel free to explore the example scripts `demo1.py`, $\ldots$, `demo13.py` in the `examples` directory.

2. Unweighted second moments and multiplicative and additive biases

(a) Write a function in python that calculates the unweighted second moments for a surface brightness distribution. The input should be a `numpy` array, and it should return the values of Q_{11}, Q_{22}, Q_{12}.

(b) In `GalSim`, create noiseless images of Gaussian profiles with $\sigma = 0.8$ arcsec with ellipticities $e \in [(0.01, 0), (0.02, 0), ..., (0.09, 0), (0.1, 0)]$ and flux of 5000. Make a plot of $\Delta e \equiv e_{\text{true}} - e_{\text{measured}}$ vs e_{true}, where e_{measured} is obtained by a combination of the moments calculated by your routine above. Use the following definition of ellipticity:

$$e = \frac{Q_{11} - Q_{22} + 2iQ_{12}}{Q_{11} + Q_{22} + 2(Q_{11}Q_{22} - Q_{12}^2)^{1/2}} \tag{1}$$

Remember that $e = \sqrt{e_1^2 + e_2^2}$

(c) Fit a line to your data and report values for the multiplicative and additive biases m and c.

(d) Show that the unweighted quadrupole moments of a galaxy image convolved with a PSF are the sum of the moments of the PSF and the original image (no need of `GalSim`).

Although simple and useful for simple theoretical calculations, unweighted moments are not practical, given that they do not converge in the presence of noise or for some analytical profiles such as the Airy profile. `GalSim` has a more robust routine called `galsim.image.FindAdaptiveMom()` ("adaptive moments") that effectively works by iteratively fitting an elliptical Gaussian to the surface brightness profile of the object. Look at the docstring of that routine and the references therein (to the papers by Bernstein and Jarvis (2002) and Hirata and Seljak (2003))

3. Profiles at different signal to noise

(a) Create in `GalSim` an Gaussian profile with a given ellipticity $e = (0.02, 0.0)$ and flux of 5000. Then use the routine `galsim.Image.addNoiseSNR` to create copies of that profile at different SNR (signa-to-noise ratios): $[5, 10, 15, 20, 25, 45, 50, 60, 75, 100, 150]$. Like in the previous exercise, report Δe and plot it as a function of SNR. This time use `galsim.image.FindAdaptiveMom()` to get the ellipticities (if it does not converge, try increasing the parameter `max_mom2_iter`). Look at `demo5.py` or the docstring of `galsim.Image.addNoiseSNR` to see how to use it and to look at the definition of SNR. Do you see any biases at low SNR? Most of the galaxies in a photometric survey such as DES or LSST will be small and faint (*i.e*, low SNR), so it is crucial for a shape measurement method to provide an unbiased shear estimator for each galaxy.

4. Running `SExtractor` and creating a size-magnitude diagram

(a) Run `SExtractor` through the file `lsst_chip107.fits`. You will also need the files `sex.conf`, `default.params`, and `default.conv`. Make sure that the file `default.params` contains at least the parameters `MAG_AUTO` and `FLUX_RADIUS`. You can include other parameters such as position and second moments, if you want. `MAG_AUTO` lists the estimated magnitude of each detected source (this is only one of several methods that one can use in `SExtractor` to estimate magnitude). `FLUX_RADIUS` lists the estimated half-light radius (radius enclosing half of the light). Stars are point sources and should all have a similar small size (except stars that are saturated which appear to be larger).

(b) In `python`, make a size-magnitude diagram and identify regions containing stars, saturated stars, and galaxies. Choose approximate bounds and/or cuts for each one, and make your plot again with different color codings for the points in each region.

The image `lsst_chip107.fits` is an LSST simulation created in `GalSim` by using the script `lsst.yaml` in the in the `examples` directory. It is one of the 189 CCDs that will be part of the LSST focal plane. Creating the 189 images took abut 5.5h in my personal computer (MacBook pro), so remember to modify the script accordingly if you want to create only a subset of the focal plane.