# SimCADO – simcado.source.cluster() function

Dorian DEMARS

March 2020

SimCADO is a Python libray whose goal is to simulate an observation with the E-ELT telescope, from the source emission to the optical path of the telescope, and the MICADO detector (with or without Adaptative Optics).

This short report aims to explain how the simcado.source.cluster() function generates a star cluster.

```
cluster([mass, distance, half_light_radius])
```

## 1 simcado.source.Source() class

The `simcado.source.Source()` class places a point source in the source file with parameters :

```
simcado.source.Source(filename=None, lam=None,
    spectra=None, x=None, y=None, ref=None,
    weight=None, **kwargs)
```

Each point is given spectra : `spectra` is a series of reference spectra for the Source file, `ref` points towards one of the reference `spectra`, `lam` is the array representing the wavelength bins and `weight` is a scaling parameter for that specific point source : $f(x[i], y[i]) = spectra(ref[i]) * weight[i]$.

## 2 Stars in a cluster

The `simcado.source.cluster()`, **(1)** generates stars of various type according to the total `mass` of the cluster, **(2)** randomly places these stars in the field.

### 2.1 Stars types and spectra

Depending on the total mass cluster, the code reads *IMF_1E4.dat* (for cluster with $M_{\rm tot} < 10^4\,M_\odot$) or *IMF_1E5.dat* (for cluster with $M_{\rm tot} < 10^5\,M_\odot$). Note that the code will crash if asked a mass cluster $M_{\rm tot} > 10^5\,M_\odot$. These files are said to follow a Kroupa IMF

The first file contains a series of mass stars. The code will take all stars until the n-th according to :

$$nth = \frac{cluster\_mass}{10^4\,M_\odot} \times length(IMF\_1E4.dat)$$

This means that for a given cluster mass, the function will always generates the same stars but with different positions. Fig 1 shows the cumulative values contained in the *IMF_1E4.dat* file.

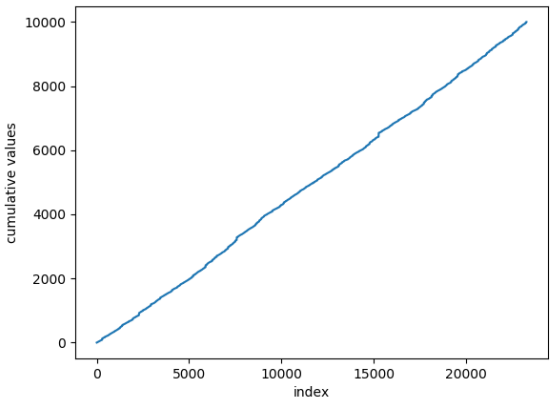


Figure 1: Cumulative sum of the content of *IMF_1E4.dat* file.

Each star's mass is associated to a stellar type, from O0V to M9V. As SimCADO contains typical `lam/spectra` couples for each of these stars and associates them.

The `weight` parameter is calculated from each star's typical absolute magnitude, converted to visual magnitude from m = $M_V$ + D, where D is the distance modulus D = $5 \cdot \log_{10}(\text{distance}) - 5$.

$$\text{weight} = 10^{-0.4\text{m}}$$

## 2.2 Stars positions

The position of stars in the cluster follow a Gaussian distribution, with a Half-Width-at-Half-Maximum (HWHM) is :

$$HWHM = \frac{half\_light\_radius}{distance}$$

$$\sigma = \frac{HWHM}{\sqrt{2ln(2)}}$$

The code generates the positions of the stars along x and y using `numpy.random.normal(0,` $\sigma$`,` $N_{stars}$`)`