Gallenspy

Gallenspy is an open source code created in python, designed for the mass profiles reconstruction in disc-like galaxies using the GLE. It is important to note, that this algorithm allow to invert numerically the lens equation for gravitational potentials with spherical symmetry, in addition to the estimation in the position of the source , given the positions of the images produced by the lens. Also it is important to note others tasks of Gallenspy as compute of critical and caustic curves and obtention of the Einstein ring.

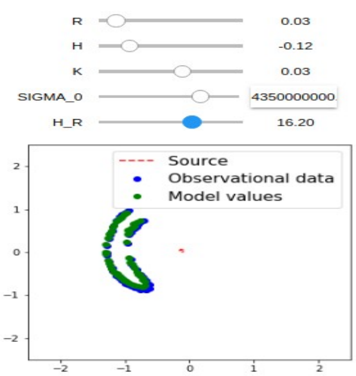
The main libraries used in Gallenspy are: **numpy** for the data hadling, **matplotlib** regarding the generation of graphic interfaces, **galpy** to obtain mass superficial densities, as to the parametric adjust with Markov-Montecarlo chains is taken into account **emcee** and for the graphics of reliability regions **corner** is used.

How to use Gallenspy

To start Gallenspy, it is important to give the values of cosmological distances in Kpc and critical density in SolarMass/Kpc² units for the critical density, which are introduced by means of a file named ***Cosmological_distances.txt***. On the other hand, it is the file ***coordinates.txt*** where the user must introduced the coordinates of the observational images and its errors respetively (in radians).(**Note:** for the case of a circular source it is present the file ***alpha.txt***, where the user must introduced angles value in radians belonging to each point of the observational images). These files mut be in each folder of Gallenspy which execute distinct tasks.

Source estimation

In the case of the estimation of the source Gallenspy let to the user made a visual fitting in the notebook **Interactive_data.ipynb** for a lens model of exponential disc in the folder **SourceAndLens_estimation**, where from this set of estimated parameters the user have the posibility of established the initial guess.



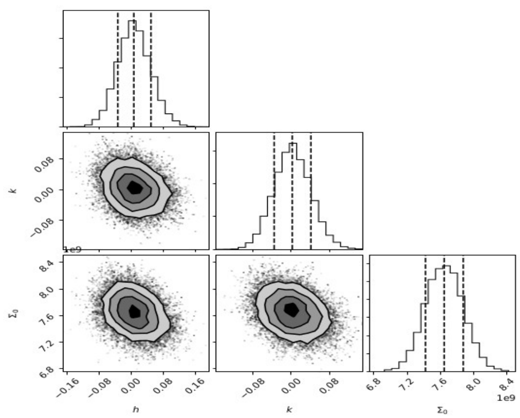
How **Interactive_data.ipynb** is an open source code, the user has the possibility of modify the parametric range in the follow block of the notebook.

```
interact(THETA,R=widgets.FloatSlider(min=0.01, max=0.2, step=0.01, value=0.15),
         H=widgets.FloatSlider(min=-0.2, max=0.2, step=0.01, value=-0.15),
         K=widgets.FloatSlider(min=-0.2, max=0.2, step=0.01, value=-0.15),
         H_R=widgets.FloatSlider(min=2, max=24, step=0.2, value=3),
         SIGMA_0 = widgets.FloatSlider(min=1e8, max=60e8, step=0.5e8, value=1e8));
```

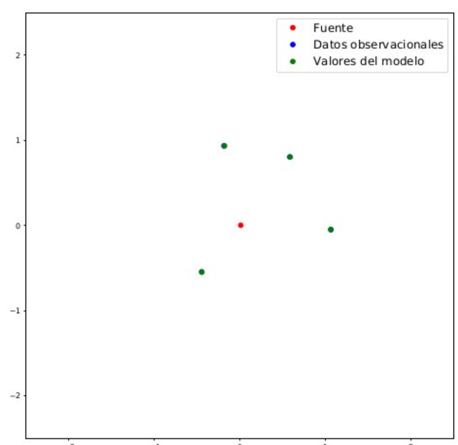
With the values of the visual fitting, the user must execute the file **source_lens.py** in the following way:

$ python3 source_lens.py

And in this point the user must consigning these values for the initial guess manually which were obtained of the visual fitting. Finally Gallenspy generate a files set denominated **parameters_lens_source.txt, contours_source_lens.pdf** and **fitting.pdf.**



contours_source_lens.pdf file



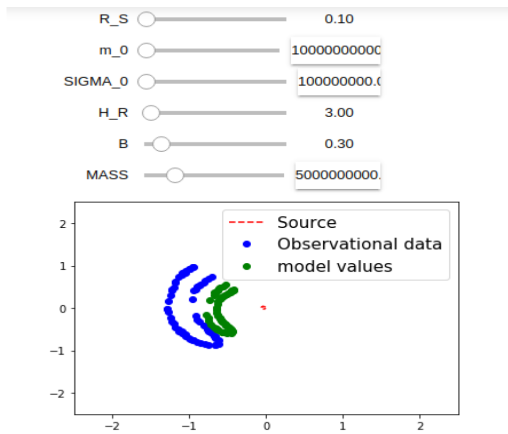
fitting.pdf file

The file **parameters_lens_source.txt** must be copy in **Einstein_Ring** and **Mass_reconstruction** folders .

Note: In the case of the circular source, the user must have present the value of the source radius, for distinct tasks of Gallenspy.

Mass reconstruction

In the case of mass reconstruction, Gallenspy let to the user made a visual fitting in the notebook **Interactive_data.ipynb** for a lens model with exponential disc, NFW and Miyamoto Nagai profiles in the folder **Mass_reconstruction**, where from this set of estimated parameters the user have the posibility of established the initial guess.



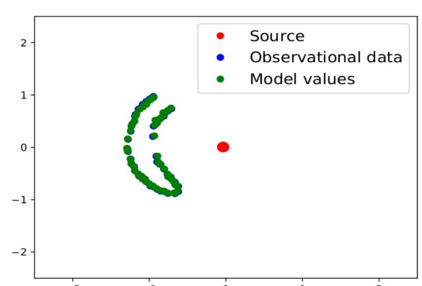
How **Interactive_data.ipynb** is an open source code, the user has the possibility of modify the parametric range in the follow block of the notebook.

```
interact(THETA,R_S=widgets.FloatSlider(min=0.1, max=60.1, step=1, value=0.1),
        m_0=widgets.FloatSlider(min=0.1e11, max=1e12, step=0.1e11, value=0.6),
        H_R=widgets.FloatSlider(min=2, max=30, step=0.2, value=3),
        SIGMA_0 = widgets.FloatSlider(min=1e8, max=1e10, step=0.5e8, value=1e8),
        B=widgets.FloatSlider(min=0.1, max=2, step=0.02, value=0.3),
        MASS=widgets.FloatSlider(min=0.1e10, max=2e10, step=0.05e10, value=0.5e10));
```

With the values of the visual fitting, the user must execute the file **parameters_estimation.py** in the following way:

$ python3 parameters_estimation.py

And in this point the user must consigning these values for the initial guess manually which were obtained of the visual fitting. Finally Gallenspy generate a files set denominated **parameters_MCMC.txt, contours.pdf** and **fitting.pdf.**



fitting.pdf file

The file **parameters_MCMC.txt** must be copy in the folder Mass_Estimation.

Mass estimation

For this task, the user must execute the file **mass_estimation.py** in the folder **Mass_Estimation**, in the following way:

$ python3 parameters_estimation.py

In this moment the user must introduce the radius (in arcs) for the calculus of the enclosed mass.

Finally Gallenspy generate a file denominated **mass_values.txt,** with the values of mass for each component of the galaxy.

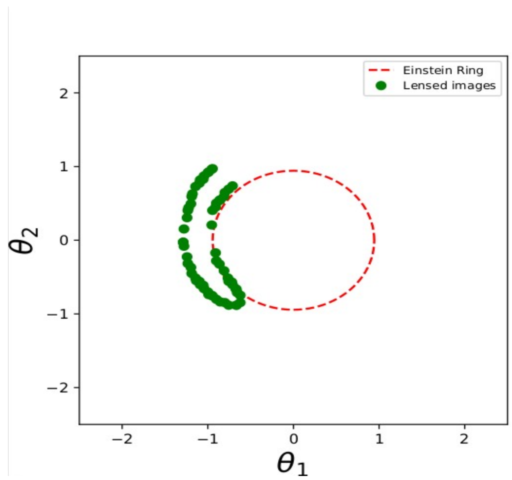
Einstein Ring

For this task, the user must execute the file **Einstein_ring.py** in the folder **Einstein_ring**, in the following way:

$ python3 Einstein_ring.py

Finally Gallenspy generate a file denominated **Einstein_radius.txt,** with the value of the Einstein radius and its errors respectively.

Other file generate in Gallenspy is **Einstein_radius_AND_images.pdf.**



The related pre-print reference: Mass reconstruction in disc like galaxies using strong lensing and rotation curves: The Gallenspy package.