Chapter 18

Stellar density in a globular cluster

For this assignment we will look at the globular cluster 47 Tucanae (or NGC 104). The data is available from the NASA Hubble space telescope archive.
Goals for this week:

- demonstrate an ability to design and implement simple programs,
- access data in FITS files, and run some data analysis on them (fitting and creating periodograms in this case).

This exercise will be graded as follows, you can earn at most 9 points by completing the exercise and handing in on time. Your grade g, awarded on a scale from 1 to 10, is calculated as follows:$g = 1 + s$, where s are the points you earned. For this exercise the points are awarded for the following things:

(3 points) For proper programming style and a programs that are clearly structured.

(2 points) For the assignment in Section 18.1.

(2 points) For the assignment in Section 18.2.

(2 points) For the assignment in Section 18.3.

18.1 Part 1: Showing FITS image data with sky-coordinates

In this part of the assignment you will download a FITS image and create a figure showing it, plotted along the correct axes (i.e. using sky coordinates).

- Go to the Mikulski Archive for Space Telescopes (MAST) Hubble pages and search for NGC 104. Have a look at the results page and see what is available.
- **Assignment:** For the purposes of this exercise you should download the following FITS file: `https://archive.stsci.edu/cgi-bin/mastpreview?mission=hst&dataid=IC2R02050` .
- **Assignment:** Use the FITS functionality from Astropy to open the FITS file. Access the image data in the FITS file and create a plot of it using Matplotlib as we have done before. You will have to adjust the range of values that are shown by Matplotlib. Look at the documentation of `pyplot.imshow`, there are relevant keyword arguments to adjust this range.
- The image you just created has no sky coordinates along the axes, instead it has pixel coordinates. Astropy contains a module `astropy.wcs` which can deal with the *World Coordinate System*, which allows you to create plots with proper sky-coordinates.

- **Assignment:** Read through the documentation for the `astropy.wcs` module, available on `http://docs.astropy.org/en/stable/wcs/index.html`. It shows you how you can add the sky coordinates to a plot of the image data. You should look through the FITS header and confirm that it contains a WCS section. It is that section that describes how the image pixels are projected on the sky.
- **Assignment:** Create a plot of the globular cluster NGC 104 that shows the stars in the cluster and has axes labeled with sky coordinates. Make sure your plot also shows the masked regions (so do not remove the `NaN`-values yet).

18.2 Part 2: Detecting stars

In this part of the exercise you will use the `photutils` package to detect the stars in the image. Unless noted otherwise you should perform the following analysis using pixel coordinates.

- **Assignment:** Install the `photutils` package.
- **Assignment:** The `photutils` package expects data with the background subtracted and without `NaN` values. Use Numpy to mask `NaN`-values and determine the median of the image. Subtract the median from the image and after that replace the `NaN`-values with zero.
- **Assignment:** Use `DAOStarFinder` from `photutils` package to detect the star positions in the image.
- **Assignment:** Plot the detected stars over the image of the NGC 104 globular cluster.

18.3 Part 3: Finding the cluster center

In this section of the assignment you will perform some simple analysis (histograms and function fits) to find the approximate cluster center. In this part of the exercise we will again perform our analysis in pixel coordinates.

- **Assignment:** Use the `numpy.histogram` function to create a two histograms. One for the x-positions of the detected stars, and one for the y-positions. (We ignore the distortion in the image, and assume the cluster to be circular.) Use on the order of 40 bins for each of these histograms. Create an array with bin centers from the bin edges returned by `numpy.histogram`.
- **Assignment:** Read the documentation to the `scipy.optimize.curve_fit` function that allows you to fit functions to data.
- **Assignment:** Create a function for a one-dimensional Gaussian add a term to be able to fit a constant background $Ae^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} + b$. *As a hint, you can use the following function definition as a start:*

```
def gaussian(x, A, mu, sigma, b):
    pass
```

- **Assignment:** For both the X and Y star position histograms perform a fit. Fit the Gaussian model you created to the bin values and bin centers. Provide an initial set of parameters for the `curve_fit` fucntion using the `p0` keyword argument. You can estimate values for `A`, `mu`, `sigma` and `b` values from a plot of the histograms. Provide plots of your histograms with the fitted functions drawn over them, like the plots in Figure 18.1.
- *Tip: The fits will be better if you skip the first and last few bins in the histograms. These bins are lower artificially because they are partially masked. Because of this they throw off the fitting algorithm.*

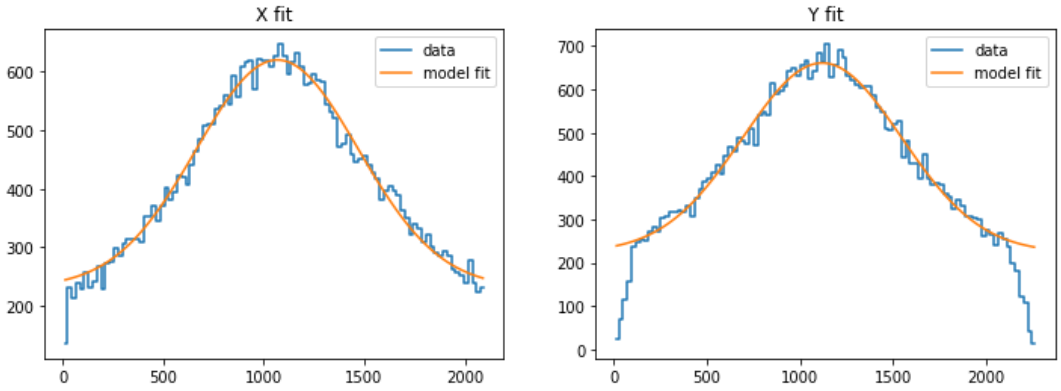

Figure 18.1: Example of the fits to X and Y star position histograms.

- **Assignment:** Finally create a histogram of the pixel radii (from the cluster center) at which the stars were detected. Explain the break in this histogram that is present at a radius of about 1000 pixels (put the answer to this question in your report).

18.4 Handing in

Check that the following things are included in your submission:

- a report describing your data sources and methods,
- plots you were asked to create,
- all code you wrote to perform your data analysis (able to be executed),
- any ancillary files needed to run your programs.

Furthermore, check that your code conforms to proper style and that it has a clear structure. Check that your code does not depend on hard coded paths, as those will not be present on the computer used for grading — make everything work with relative paths (and structure the tarball that you hand in such that the program will run without having to move files around).