NUR Hand-in Exercise 3

Angèl Pranger

March 24, 2025

Abstract

This document shows my solutions to hand-in exercise 3 of numerical methods in astrophysics.

1 Satellite galaxies around a massive central

General code used in this exercise is:

```python
import numpy as np
import matplotlib.pyplot as plt

def n(x,A,Nsat,a,b,c):
    return A*Nsat*((x/b)**(a-3))*np.exp(-(x/b)**c)

def readfile(filename):
    f = open(filename, 'r')
    data = f.readlines()[3:] #Skip first 3 lines
    nhalo = int(data[0]) #number of halos
    radius = []

    for line in data[1:]:
        if line[:-1]!='#':
            radius.append(float(line.split()[0]))

    radius = np.array(radius, dtype=float)
    f.close()
    return radius, nhalo #Return the virial radius for all the satellites in the file, and the number of halos

#Call this function as:
#radius, nhalo = readfile('satgals_m15.txt')


# Plot of binned data with the best fit (question 1b.4 and 1c)
# As always, feel free to replace by your own plotting routines if you want
xmin, xmax = 1e-4, 5. # replace by your choices
n_bins = 100 # replace by your binning
edges = np.exp(np.linspace(np.log(xmin), np.log(xmax), n_bins+1))

fig1b, ax = plt.subplots(3,2,figsize=(6.4,8.0))
for i in range(5):
    Nsat = 100 # replace by actual appropriate number for mass bin i
    x_radii = np.random.rand(10000) * (xmax-xmin) # replace by actual data for mass bin i
    Ntilda = np.sort(np.random.rand(n_bins)) * (xmax-xmin) # replace by fitted model for mass bin i integrated per radial bin
    binned_data=np.histogram(x_radii,bins=edges)[0]/Nsat
    row=i//2
    col=i%2
    ax[row,col].step(edges[:-1], binned_data, where='post', label='binned data')
    ax[row,col].step(edges[:-1], Ntilda, where='post', label='best-fit profile')
    ax[row,col].set(yscale='log', xscale='log', xlabel='x', ylabel='N', title=f"$M_h \\approx 10^{{{11+i}}} M_{{\\odot}}/h$")
ax[2,1].set_visible(False)
plt.tight_layout()
```

```
handles,labels=ax[2,0].get_legend_handles_labels()
plt.figlegend(handles, labels, loc=(0.65,0.15))
plt.savefig('my_solution_1b.png', dpi=600)

# Plot 1c (same code as above)
fig1c, ax = plt.subplots(3,2,figsize=(6.4,8.0))
for i in range(5):
    Nsat = 100 # replace by actual appropriate number for mass bin i
    x_radii = np.random.rand(10000) * (xmax-xmin) # replace by actual data for mass bin
    i
```

Q1.py