# Hand-in Exercise 2

## Deadline: Oct 30, 23:59

**Read these instructions carefully before you start coding.**

This second hand-in exercise covers material from **lectures 4 through 6** (up to and including minimization). Unless noted otherwise, you are expected to code up your own routines using the algorithms discussed in class and **cannot use special library functions** (except exp()). Extremely simple functions like `arange`, `linspace` or `hist` (without using advanced features) are OK. When in doubt, write your own, or ask us. You can use the routines you wrote for the tutorials, however, your routines must be written **by yourself**. Codes will be checked for blatant copying (with other students as well as other sources), routines which are too similar get **zero points**.

For every main question you should write a **separate program**. We **must** be able to run everything with **a single call to a script** `run.sh`[1], which downloads any data (data needed for an exercise **may not** be included in your code package but needs to be retrieved at runtime), runs your scripts and generates a PDF containing all your source code and outputs **in the following format**:

- Per main question, the code of any shared modules.
- Per sub-question, an explanation of what you did.
- Per sub-question, the code specific to it.
- Per sub-question, the output(s) along with discussion/captions.

Your code may be handed however you'd like, for example by emailing a zip file or by sharing a github repository. If you organize your code in multiple folders, `run.sh` should be **in the top folder**. Have a fellow student test that your code works by running ./run.sh to make sure there are no permission errors! Exercises that are not run with this single command or do not have their code and output in the PDF get **zero points** (this includes solutions in Jupyter notebooks).

Ensure that your code runs to completion on the `pczaal` computers using `python3`(!). Codes that do not get **zero points**. Your code should have a total run time of **at most 10 minutes**, solutions generated will not be checked beyond this limit. If, **during testing**, a part of your code takes long to run, remember that you can run it once and read in its output (saved to file) in the rest of your code – however, the rules as stated above still apply to whatever you hand in at the end (everything run with a single command, all outputs produced on the fly, total runtime limit, etc). It is possible to test your own code on the pczaal remotely by using ssh, you can directly log into a pczaalXX computer (XX is between 00 and 21) using `ssh -XY [username]@pczaalXX.strw.leidenuniv.nl` to test your code.

For all routines you write, **explain** how they work in the comments of your code and **argue** your choices! Similarly, whenever your code outputs something **clearly indicate** next to the output/in the PDF what is being printed. This includes discussing your plots in their captions.

If a part of your code **does not run**, explain what you did so far and what the problem was and still include that part of the code in the PDF for possible partial credit. If you are unable to get a routine to work but you need it for a follow-up question, use a library routine in the follow-up (and clearly indicate that and why you do so).

Each sub-question starts with a reference to a relevant tutorial ($T$) and/or a reference to the relevant lecture ($L$). For example, *[L2,T2.3]* means the question is related to lecture 2 and question 3 in tutorial 2. Your previously written code for these will be a great starting point!

[1]See the example here: `https://github.com/Fonotec/NURSolutionTemplate`.

1. **Dark Matter Halo**
Performing simulations of isolated disk galaxies is a common practice in astrophysics. To run these simulations we need to make initial conditions that are representable for the dark matter distribution of galaxies. In this exercise we will make initial conditions for a dark matter halo of an isolated galaxy. Most generally the dark matter profile is well-fitted by an NFW profile, based on the results of many N-body simulations. The NFW profile is given by:

$$\rho(r) = \frac{\rho_0}{x(1+x)^2}. \tag{1}$$

Here $x = r/r_s$, with $r_s$ the scale radius. For most practical circumstances it is not necessary to use a NFW profile, but we can use a Hernquist profile. The advantage of this is that an Hernquist profile converges faster and therefore the sample we probe is smaller and thus numerically more stable. In general the Hernquist profile has the form:

$$\rho(r) = \frac{M_{\rm dm}}{2\pi}\frac{a}{r(r+a)^3}. \tag{2}$$

For this exercise we will assume that $M_{\rm dm} = 10^{12}$ and $a = 80$ kpc.

(a) (6 points) *[L4,T4.3,T4.6]* Write a random number generator that returns a random floating-point number between 0 and 1. At minimum, use some combination of an (M)LCG and a 64-bit XOR-shift. We'll test its quality using the following methods:

1. have your code plot sequential random numbers against each other in a scatter plot ($x_{i+1}$ vs $x_i$) for the first 1000 numbers generated.
2. have your code generate 1,000,000 random numbers and plot the result of binning these in 20 bins 0.05 wide and plot Poisson uncertainties to see if your RNG behaves good enough.
3. Calculate the Pearson correlation coefficient $r_{x_i x_{i+1}}$ and $r_{x_i x_{i+2}}$ for your random numbers using 100,000 numbers and show it in your answers.

**Evaluate** the outcome of these tests and change your RNG parameters if necessary. Make sure you use a fixed seed and set it only once for your entire program. The seed value should be the first output when we run your code.

(b) (5 points) *[L4,T4.4]* Use this functional form to generate a radial distribution of particles, do this for $10^6$ particles. Make a plot showing the enclosed fraction of particles at a certain radius and compare this with the expected amount of enclosed fraction of mass. *Hint: You need to calculate the cumulative distribution function, do not forget to convert the density profile to a 3D density profile for r.*

(c) (2 points) *[L4,T4.2]* Generate a 3D distribution of $10^3$ particles in a Hernquist profile and make a 3D scatter plot showing this distribution. Additionally make a plot of the random angels $\phi$ and $\theta$, explain how you generated random numbers in $\phi$ and $\theta$.

(d) (6 points) *[L4,T4.1]* Numerically calculate $\mathrm{d}\rho(r)/\mathrm{d}r$ at $r = 1.2a$. Output the value found alongside the analytical result, both to at least 12 significant digits. Choose your differentiation algorithm such as to get them as close as possible.

(e) (5 points) *[L5,T5.2]* In galaxy formation we often want to calculate the $R_\Delta$ and $M_\Delta$ values where $\Delta$ is the amounts of time we exceed the critical density $\rho_{\rm c} = 8.5 \cdot 10^{-27}$ kg/m$^3 \approx 150$ M$_\odot$/kpc$^3$. This means that $R_\Delta$ is the radius at which $\rho_{dm}(r) = \Delta\rho_c$. Find the value of $R_{200}$, $R_{500}$ and their corresponding masses $M_{200}$ and $M_{500}$ (use units of $M_\odot$ and kpc).

(f) (6 points) *[L6,T6.2,T6.5]* The potential of an asymmetric Hernquist potential in 2D is given by:

$$\Phi = -\frac{GM_{\rm dm}}{\sqrt{(x-1.3\text{ kpc})^2 + 2(y-4.2\text{ kpc})^2} + a}. \tag{3}$$

Starting from ($-1000$ kpc, $-200$ kpc) find the minimum of this potential. Make a plot of the number of iterations versus the distance from the final point.

2. **Satellite galaxies around a massive central – part 2**
Recall the number density satellite profile from the first assignment:

$$n(x) = A \langle N_{\mathrm{sat}} \rangle \left(\frac{x}{b}\right)^{a-3} \exp\left[-\left(\frac{x}{b}\right)^{c}\right]. \tag{4}$$

Here $x$ is the radius relative to the virial radius, $x \equiv r/r_{\mathrm{vir}}$, with $x < x_{\mathrm{max}}$, and $a$, $b$ and $c$ are free parameters controlling the small-scale slope, transition scale and steepness of the exponential drop-off, respectively. $A$ normalizes the profile.

Throughout this exercise, take $a = 2.4$, $b = \mathbf{0.25}$, $c = 1.7$, $x_{\mathrm{max}} = 5$, $\langle N_{\mathrm{sat}} \rangle = 100$ and $A = 256/(5\pi^{3/2})$.

(a) (6 points) *[L6,T6]* The number of satellites in the infinitesimal range $[x, x + \mathrm{d}x)$ is given by $N(x)\mathrm{d}x = n(x)4\pi x^2 \mathrm{d}x$. Find the maximum of $N(x)$ for $x \in [0, 5)$ using a maximization algorithm that is not equal to the maximization algorithm used in exercise 1. Output both the $x$ and $N(x)$ at the maximum.

(b) (5 points) *[L4,T4.4,T4.5,T5.4]* We want to generate 3D satellite positions such that they statistically follow the satellite profile in equation (4); that is, the probability distribution of the (relative) radii $x \in [0, 5)$ should be $p(x)\,\mathrm{d}x = N(x)\,\mathrm{d}x / \langle N_{\mathrm{sat}} \rangle$. Use one of the methods discussed in class to sample this distribution. Generate 10,000 points and make a log-log plot showing $N(x)$ and a histogram of your 10,000 sampled points. Use 20 logarithmically-spaced bins between $x = 10^{-4}$ and $x = x_{\mathrm{max}}$ and don't forget to divide each bin by its width. Check if both profiles agree with each other.

(c) (3 points) *[L5,T5.3]* Select 100 random satellite galaxies from (b) in a way that is guaranteed to:

1. select every galaxy with equal probability;
2. not draw the same galaxy twice;
3. not reject any draw.

Next sort the 100 drawn galaxies from smallest to largest radius and plot the number of galaxies within r, use a xlog plot with x range of $x = 10^{-4}$ to $x = x_{\mathrm{max}}$.

(d) (3 points) *[L5,T5.3]* Take the radial bin from (b) containing the largest number of galaxies. Using sorting, calculate the median, 16th and 84th percentile for this bin and output these values. Next, divide the 10,000 points up into 100 haloes each containing 100 galaxies and make a histogram of the number of galaxies in this radial bin in each halo (so 100 values). Bin $i$ of this histogram should be centered on integer $i$ (so first bin around 0, second around 1, etc). Plot this histogram, and over-plot the Poisson distribution (using your code from hand-in 1 or `numpy`) with $\lambda$ equal to the mean number of galaxies in this radial bin.
*Hint: If the histogram and the Poisson distribution don't seem to match within a reasonable uncertainty, something's wrong!*

Most recent changes:
*Oct 24:* The value of $b$ in question 2 should be 0.25, not 0.3. 2(d) referred to a question that was removed and has been clarified.