A. Pillepich
Block Course Cosmology
Heidelberg University

Problem Sheet #1 (Hand out: Wed => hand in: Fri)

0. Rationale and instructions

One of the most insightful and practical ways to "experience" the Universe on intermediate-to-large spatial scales is by inspecting the output of cosmological "N-body" simulations.

We will therefore refer to examples of such simulations and learn how to extract useful information from them in relation to the distribution and evolution of matter on large scales, the structure of dark-matter haloes, the demographics of dark-matter haloes, etc etc.

To complete the Problem Sheets of this course, you will need to be familiar with Python, you will need a computer and you will need an internet connection. If any of these is missing, please reach out to the instructor(s).

Each problem sheet will have to be completed on a Jupiter Notebook, via the Jupiter Lab below.

You may not use Astropy or similar astronomy/cosmology-related libraries. But, you can use them to verify your personal solutions. You may use other foundational libraries such as NumPy, Matplotlib, h5py, etc

For each problem, in the hand out, we expect the following contents:

- complete text of the problem;
- actual code — reproducible;
- solutions, with plots, images, maps, tables, numerical values — as required;
- comments and thoughts written down in English and with complete sentences about the process and the solutions.

Keep all your solutions properly organised, in your Jupiter Notebook. Most of the problem sheets will be in connection to one another.

The handouts consist of both a Jupiter notebook file and a pdf, to be emailed to pillepich@mpia.de and ravishankar@mpia.de

Note:
If you are a beginner to programming and/or Python, you can read an introductory Python Tutorial, followed by a NumPy Tutorial, Matplotlib Tutorial (for plotting).

1. Setting up the working environment

We will describe what N-body simulations are during the lectures. In the meanwhile, we can get familiar with them from a purely practical viewpoint and uncover basic properties of their output.

We need therefore to be ready with some practical tasks:

0.1 Register to access the data of the Illustris and IllustrisTNG simulations:
http://www.tng-project.org/data/

0.2 Once approved, ask for access to the Jupiter Lab Workspace:
https://www.tng-project.org/data/lab/

0.3 Get familiar with the ideas behind the Illustris and TNG simulations and get familiar with their data infrastructure:
http://www.illustris-project.org/
http://www.tng-project.org/

0.4 Get familiar with hdf5 file format: all Python distributions can handle hdf5 files via available I/O routines: get familiar with h5py Tutorial. Note: to inspect hdf5 from a terminal, you will need proper installation.

0.5 Get familiar with the structure of the datafiles from the simulations. Obtain and get familiar with the basic I/O functions needed to work with this simulation data:
https://github.com/illustristng/

NOTE: we will mostly use dark-matter only or N-body only or gravity only realizations of such simulations. These have suffix "-Dark". Moreover, we will mostly use the lowest-resolution versions to facilitate data handling: e.g. TNG100-3-Dark, Illustris-3-Dark and TNG300-3-Dark. For these, we can download directly the virtual 'simulation.hdf5' container file. The low-resolution simulations are "hidden" from the main webpage https://www.tng-project.org/data/, but can be reached via properly tweaking the web address.

1. The distribution of matter on large scales.

1.1 Load the mass and coordinates of all the dark-matter haloes from the TNG100-3-Dark simulation at z=0 and make a map (an image) of their distributions in the simulated box in as a 2D projection along a random axis of the simulation:

- Plot one point per halo
- Plot one circle per halo with the size of the circle proportional to the halo mass: what are the ranges of masses simulated in TNG100-3-Dark in solar masses?
- Plot different halo maps with different minimum cuts in halo mass or different bins in halo mass

1.2 Load the mass and coordinates of dark-matter particles from the TNG100-3-Dark simulation at z=0 and make an image of the matter distribution in such box in a random orientation:

- Plot one point per particle (!) Can you see anything?

- Make a 2D histogram of the dark-matter particles along a random simulation axis. Choose different pixel sizes to get an idea.
- Optional: Make a smoothed 2D histogram by smoothing with a 2D Gaussian filter
- What densities are we looking at? In Msun/Mpc^2? How would you describe such density field?
- Optional: Now load the dark-matter particles at different times i.e. different z and inspect how the density distribution evolves as the Universe ages.

NOTE: no need to use the illustris_python for this.

2. The distribution of matter within haloes.

2.1 Load the mass and coordinates of the dark-matter particles that belong to a given halo from the TNG100-3-Dark simulation at z=0. For example, choose the 5th object in the Group list or the 10th, or whichever:

- Make an image of such halo, i.e. make a map of its dark-matter particles in a random projection (use a 2D histogramming). What densities are we looking at in Msun/kpc^2? How would you describe the density field?
- Compare different projections of the same halo, i.e. the same halo projected along different simulation axes. Is it spherical?
- Compare the dark-matter density maps of haloes of very different masses.

NOTE: no need to use the illustris_python for this.

Basic Input:

- There are three main types of output files from simulations: snapshots or particle data, halo or galaxy catalogs, merger trees. We will work with the first two.
- The snapshot with the highest number is the one at lower redshift. z=0 is snap 099.
- Haloes are also called Groups — these are identified with the FoF technique
- There are many ways to define the mass of a halo: here use Group/GroupMass
- There is only one type of material in TNG100-3-Dark: dark-matter. It is also called PartType1
- Use the input online to understand the units of the simulation data. We want to work in solar masses (Msun), kilo parsecs (kpc) or megaparsecs (Mpc). Little “h” is the value of the Hubble Parameter: its adopted value is in the Header file ("HubbleParam").

NOTE: the tools developed to solve this Problem Sheet will be needed for Problem Sheets #3 and #4