TP4 Exercise 5

Intro to N-body Cosmological Simulations

https://obswww.unige.ch/lastro/misc/TP4/doc/rst/Exercices/Ex05.html

Main idea

To be able to compare our **observations** of the *large-scale universe* to **theory**, we need to try and simulate the distribution and evolution of matter across cosmic time.

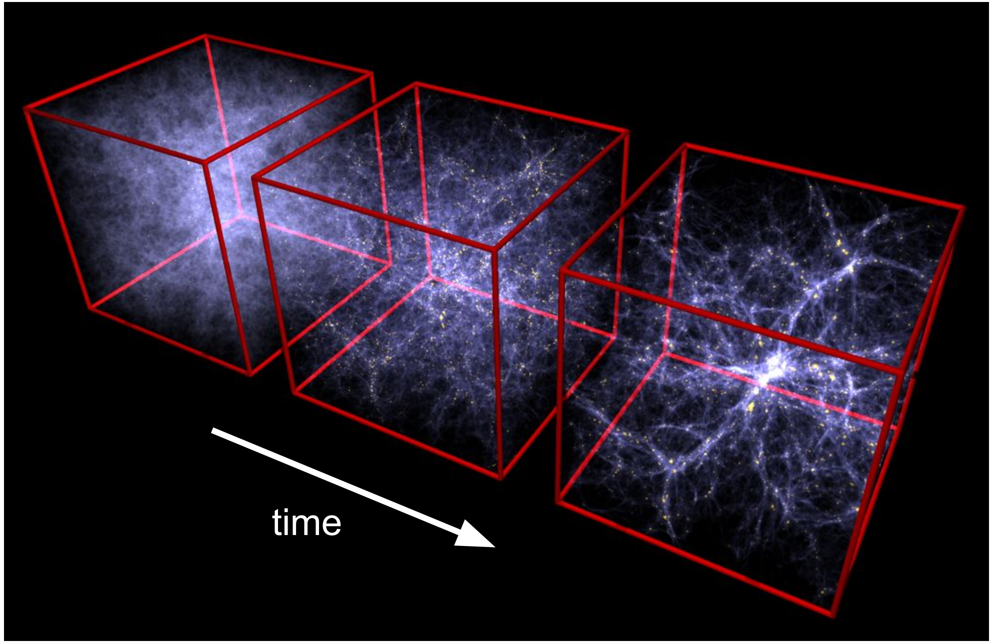
What this effectively means is

(1) build a huge virtual box in our computers

(2) fill it with a ton of massive particles

(3) let them evolve according to the laws of physics--like *gravity*.

Credit : Volker Springel



Considerations

- Solving the equations of motion for each particle is *computationally expensive*
  → tradeoff between accuracy and computability
  → specialized high-performance computing clusters are necessary

Considerations

- Solving the equations of motion for each particle is *computationally expensive*
  → tradeoff between accuracy and computability
  → specialized high-performance computing clusters are necessary

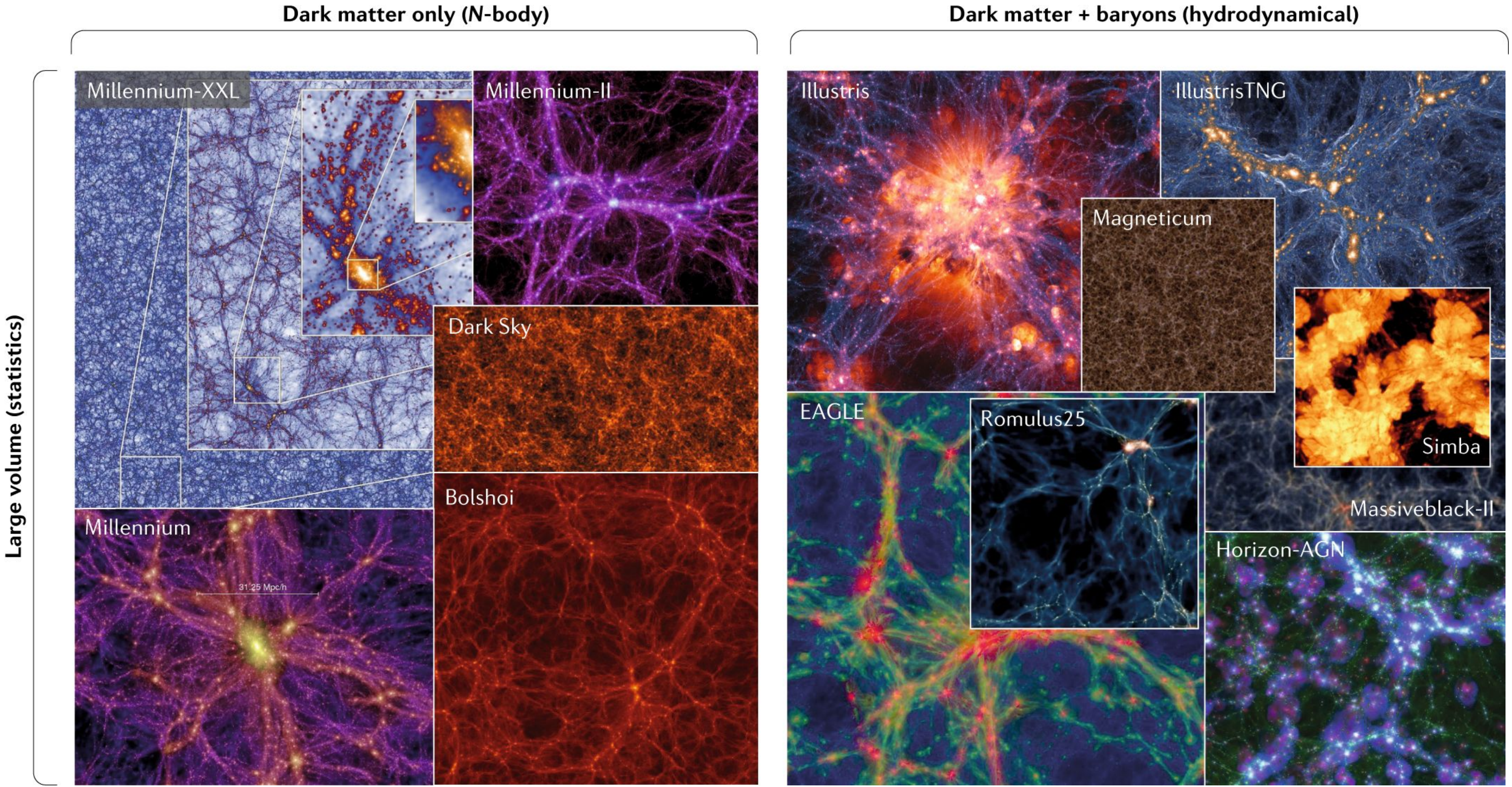
- What does an N-body 'particle' mean ?
  → depends on the simulation / resolution
  → typically a 'mass unit' on the order of 10^9 solar masses

Considerations

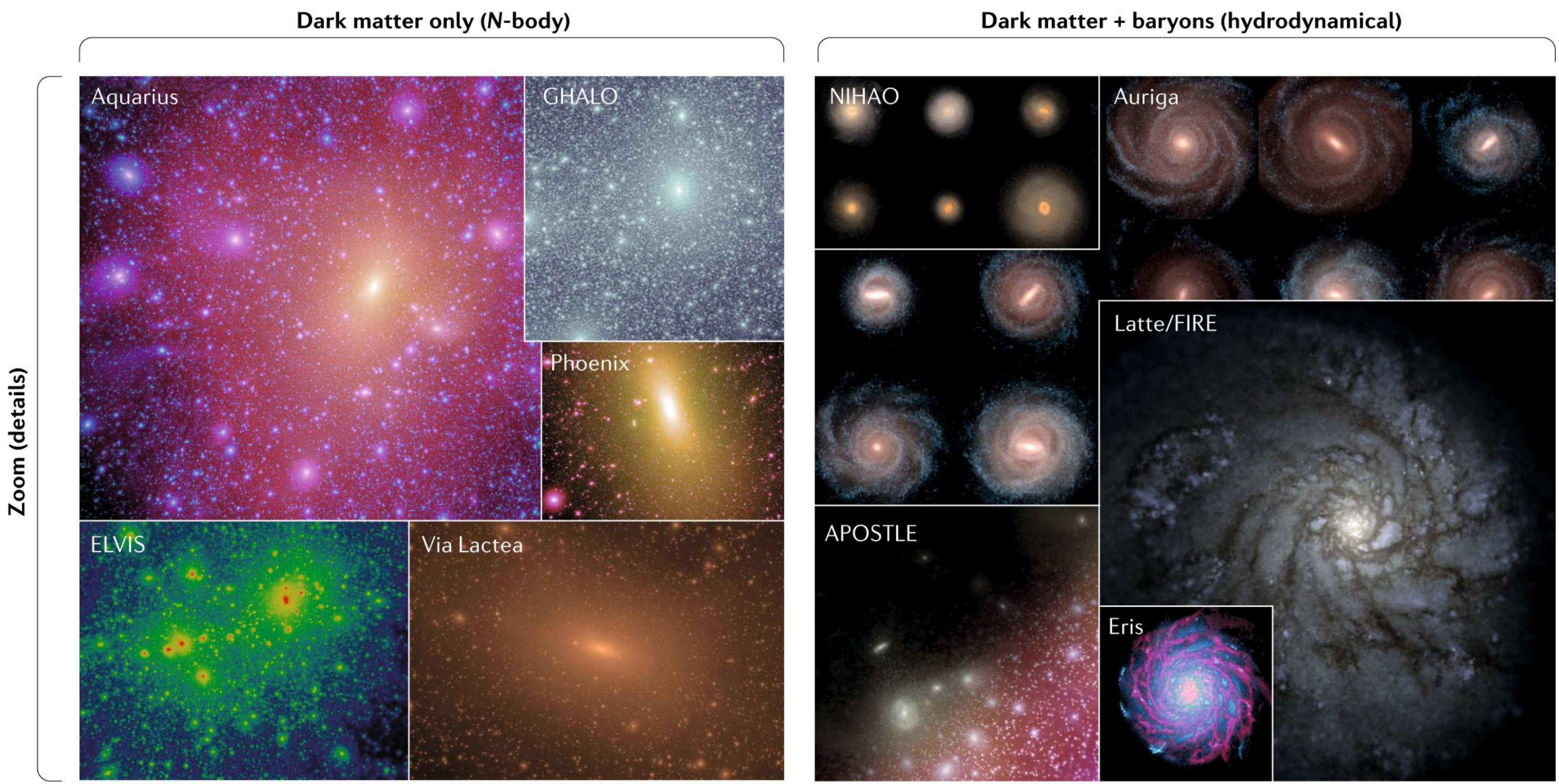
- Solving the equations of motion for each particle is *computationally expensive*
  → tradeoff between accuracy and computability
  → specialized high-performance computing clusters are necessary

- What does an N-body 'particle' mean ?
  → depends on the simulation / resolution
  → typically a 'mass unit' on the order of 10^9 solar masses

- Different kinds of simulations for different science
  → DM-only : largest scales (historically first)
  → Hydrodynamical : includes baryons (now possible with improving technology)



Credit : Vogelsberger et al. *Nature Reviews Physics* 2, 42 (2020)



Credit : Vogelsberger et al. *Nature Reviews Physics* 2, 42 (2020)

Your task

Run a small, DM-only simulation and visualise / analyse the results.

1. Generate the initial conditions (i.e. initial particle positions and velocities) using a public code called *N-GenIC*

Your task

Run a small, DM-only simulation and visualise / analyse the results.

1. Generate the initial conditions (i.e. initial particle positions and velocities) using a public code called *N-GenIC*

2. Run the simulation from a very early time (i.e. high redshift) up to today using another public code called *Gadget-2*

Your task

Run a small, DM-only simulation and visualise / analyse the results.

1. Generate the initial conditions (i.e. initial particle positions and velocities) using a public code called *N-GenIC*

2. Run the simulation from a very early time (i.e. high redshift) up to today using another public code called *Gadget-2*

3. Analyse what you get :
   (i) Using public and in-house codes (Yves Revaz)
   → Make a video of the output snapshots (i.e. the state of the system at different fixed times)
   → Run a DM halo finder on the snapshots (V. Springel)

   (ii) Write your own code to study DM halo statistics and their mass profiles

Practicalities

You will use

`lesta` : the computing cluster at the UNIGE Department of Astronomy.

https://www.astro.unige.ch/wiki/IT/doc/astroge/lesta

`/scratch/<username>/` : special directory on `lesta` where you'll run the sim.

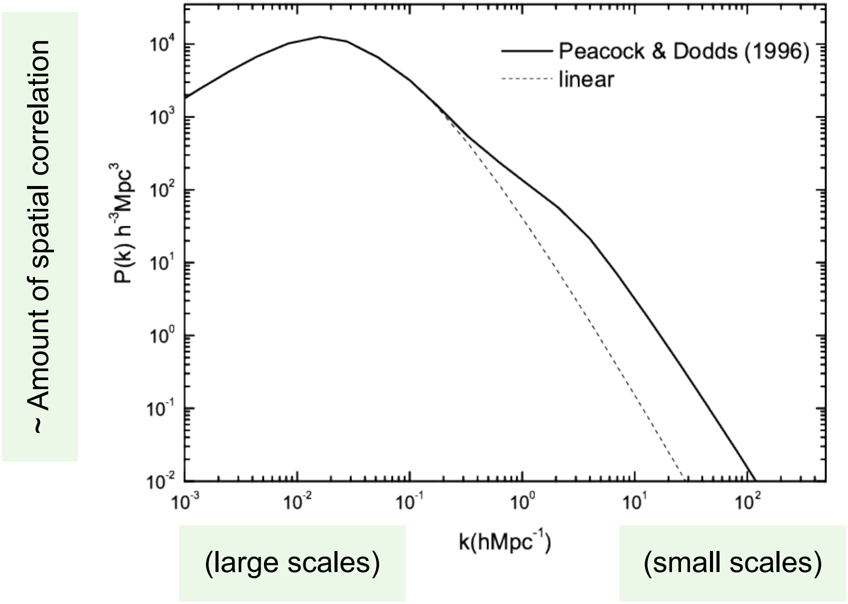
Do not run in your home directory !

`$ sbatch` privileges : jobs are submitted to the system using this command.

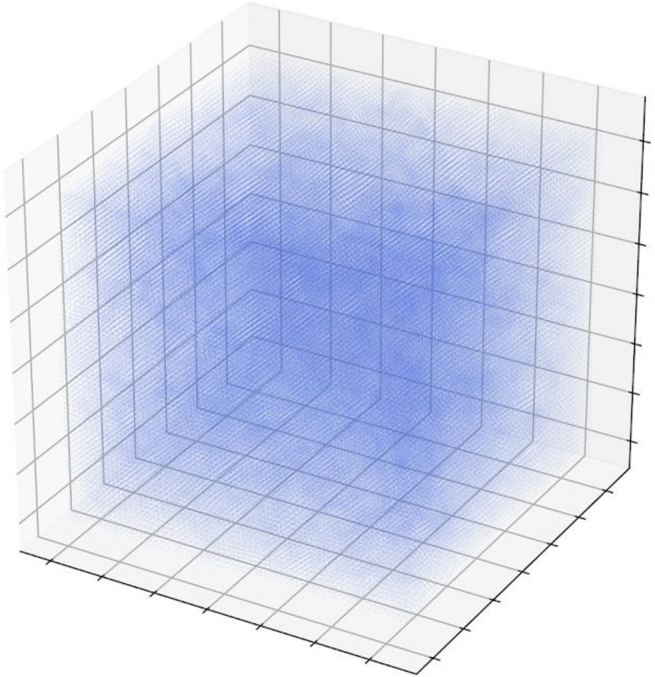
Email astro-it-support@unige.ch if you don't have these already.

1. Initial conditions

Credit : Wang et al. *A&A* 536, A96 (2011)



Matter power spectrum



https://blog.tensorflow.org/2020/03/simulating-universe-in-tensorflow.html

1. Initial conditions

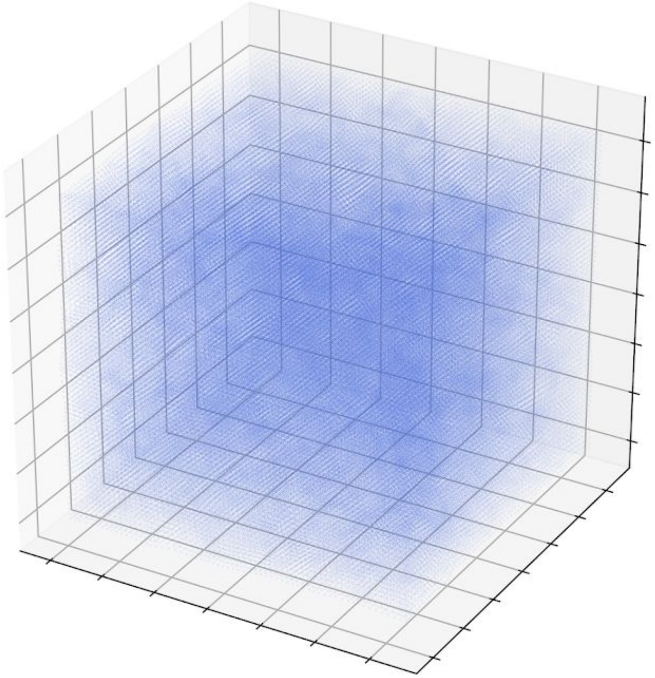
`N-GenIC` will do this for you (**C** code)

-- download the code

-- set it up to compile properly on your system

-- run it to produce the ICs

-- understand the cosmological parameters and their units



https://blog.tensorflow.org/2020/03/simulating-universe-in-tensorflow.html

2. Simulation

`Gadget-2` is the simulation code (also in **C**)

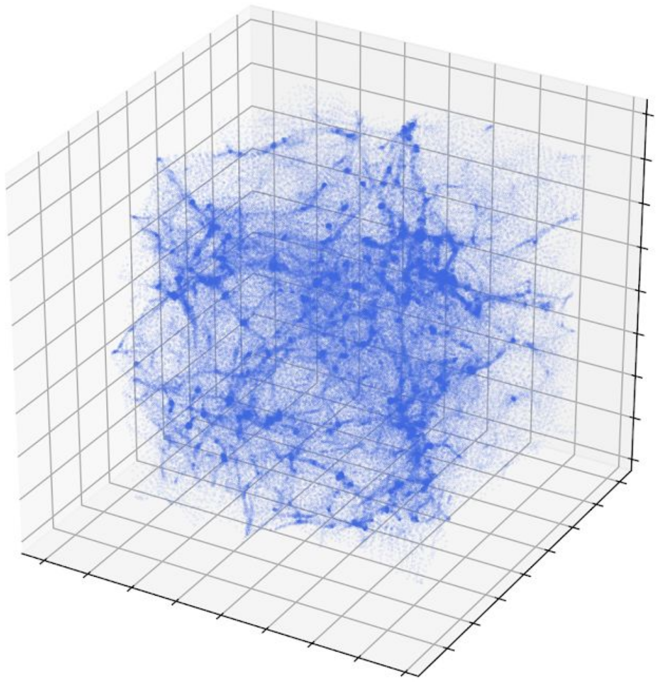
-- download the code

-- set it up to compile properly on your system

-- define the start and end time

-- define the softening length

-- launch the simulation to run in parallel using `slurm` (Simple Linux Utility for Resource Management)



https://blog.tensorflow.org/2020/03/simulating-universe-in-tensorflow.html

3. Visualisation + Analysis

`gmkgmov` within the `pNbody` **python** package

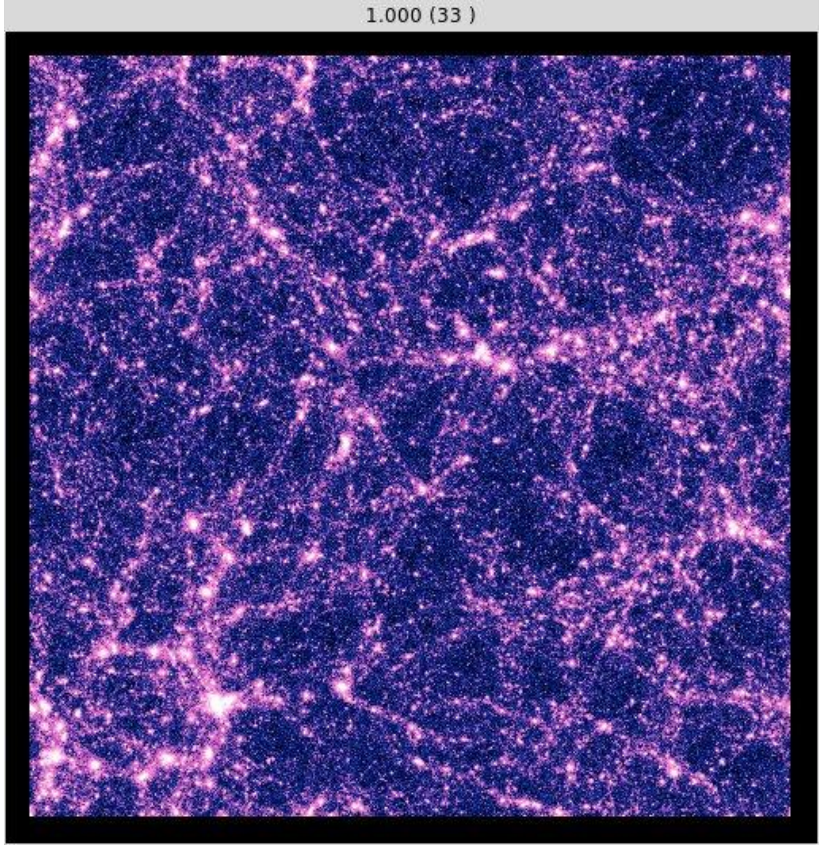
-- already included in `clastro` module

-- collects your snapshots and assembles them into a movie

`gmov` within the `pNbody` **python** package

-- already included in `clastro` module

-- displays your movie



3. Visualisation + Analysis

`FoF_Special` will find halos (**C** code)

-- download the code

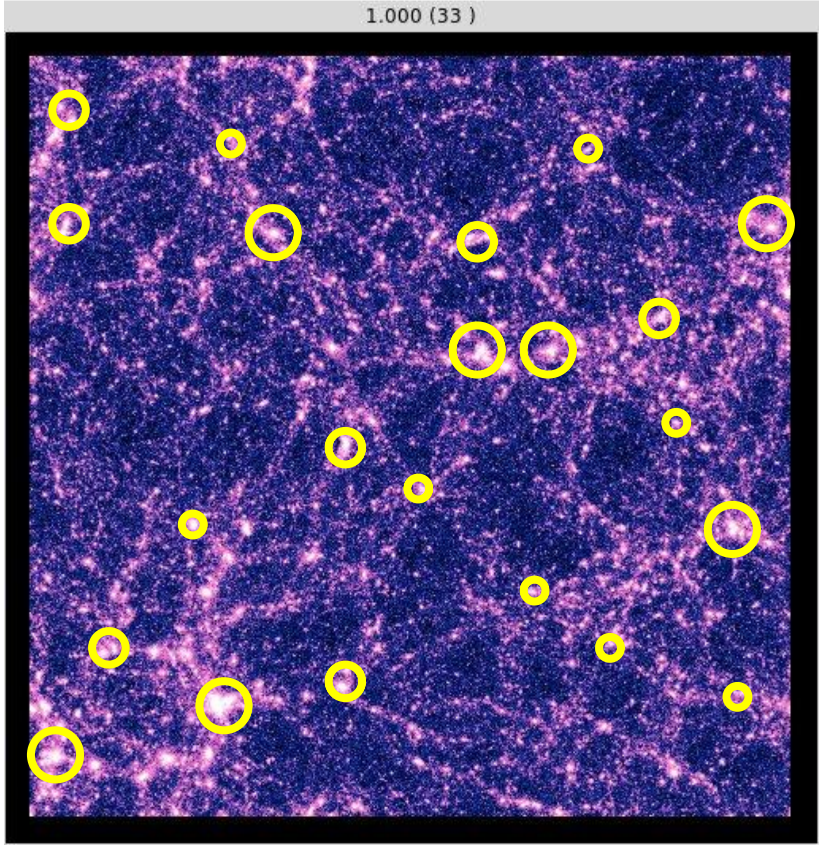
-- compile and run it on your snapshots

-- get a halo catalogue to analyse

`FoFlib` will process your halo catalogues and make them user-friendly (**python**)

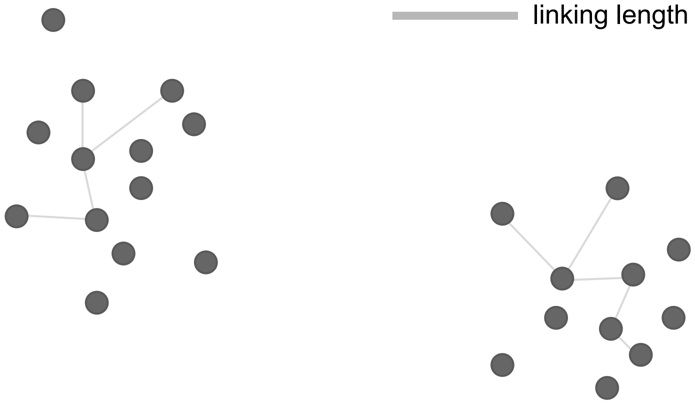
-- download the code

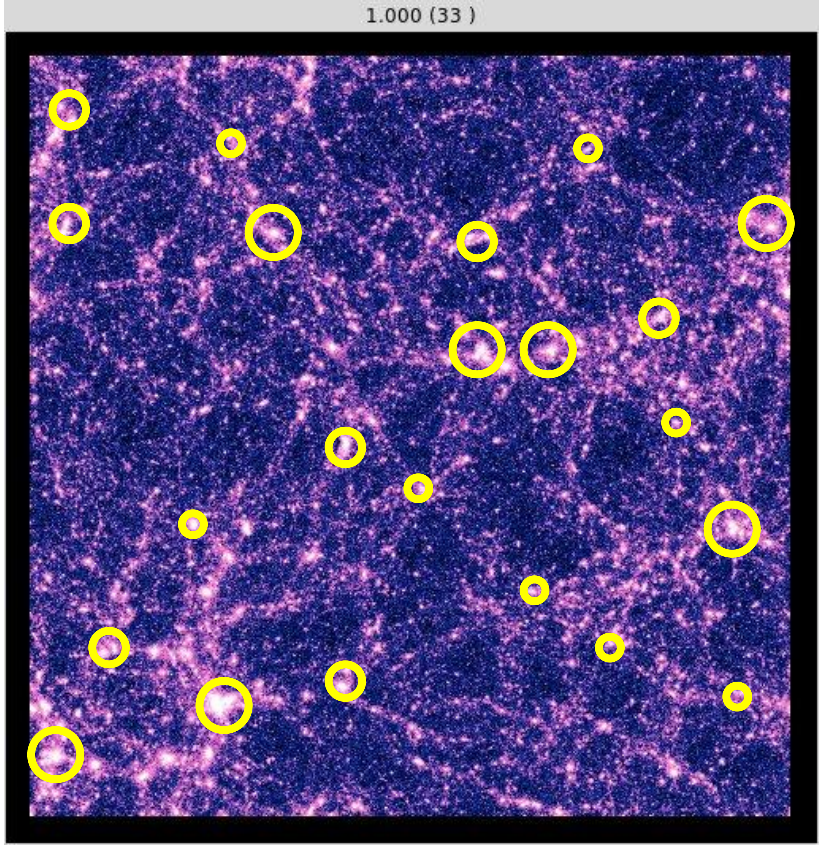
-- use its methods in <u>your own code</u>



3. Visualisation + Analysis

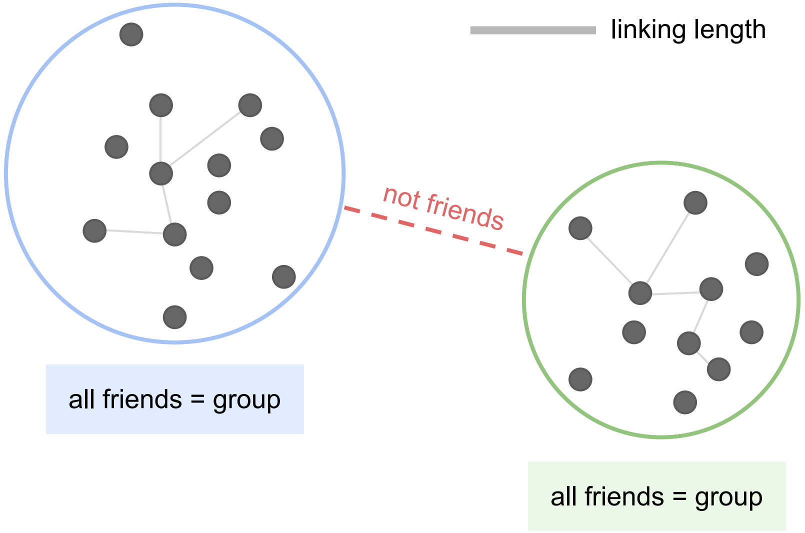
FoF = Friends of Friends

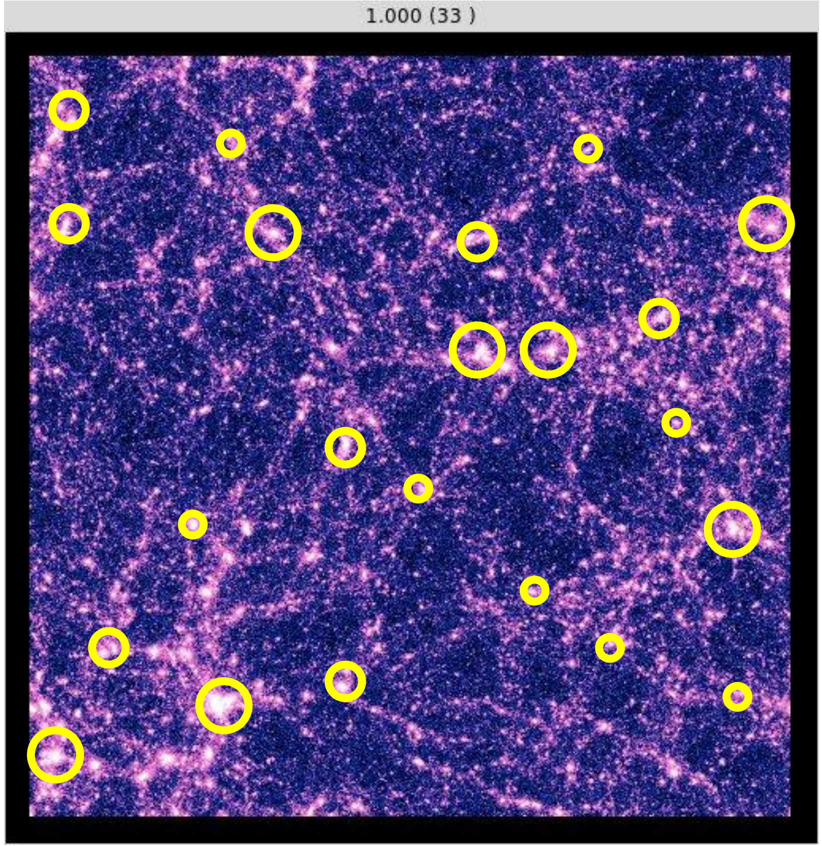




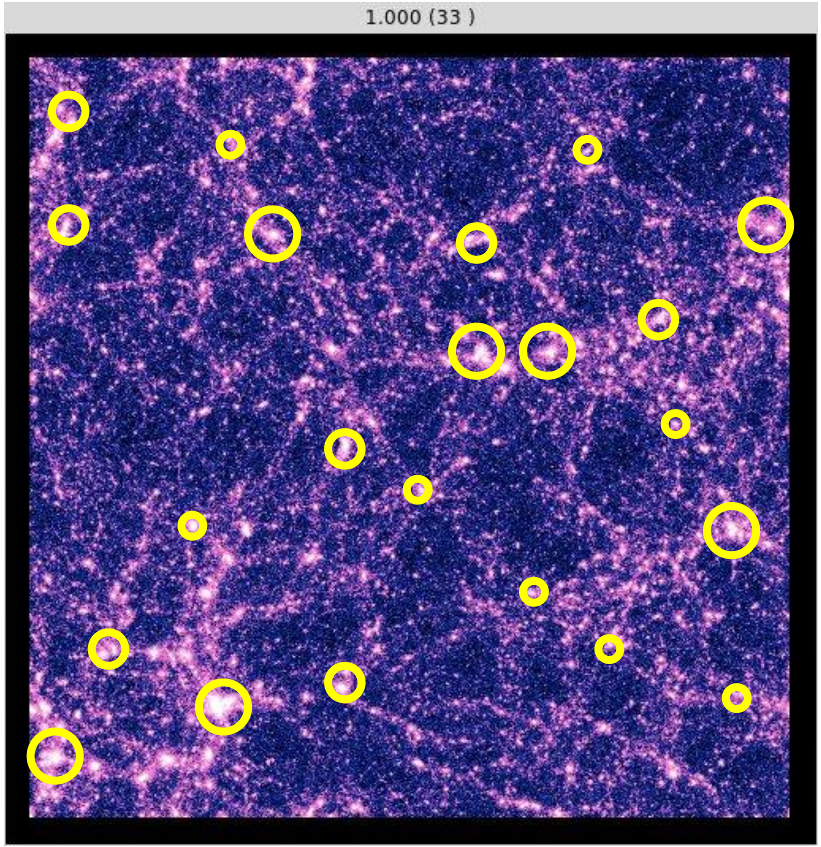
3. Visualisation + Analysis

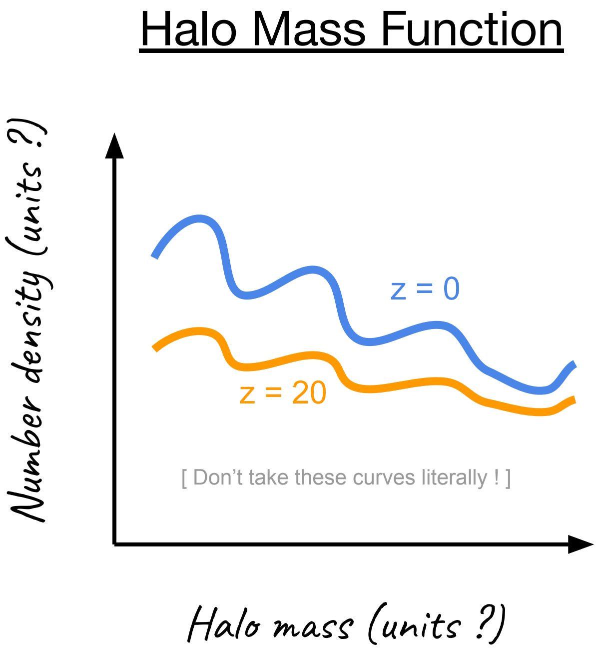
FoF = Friends of Friends



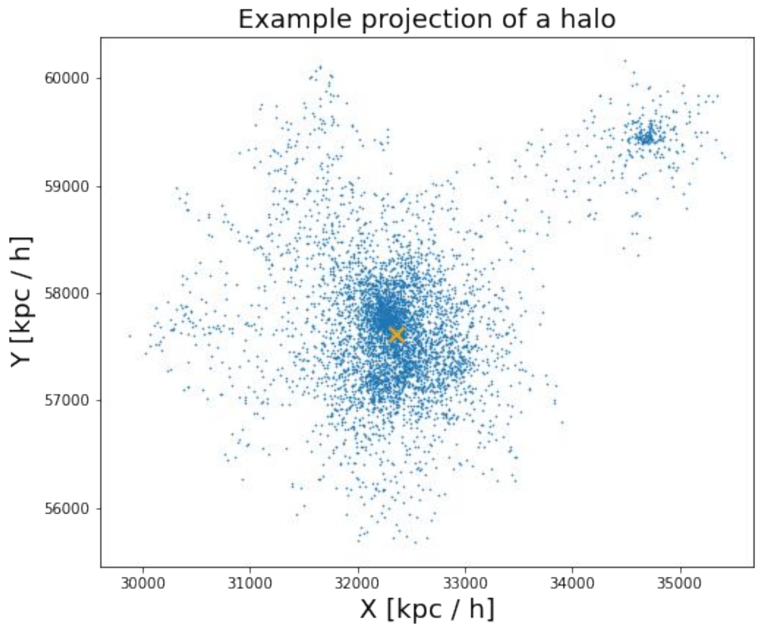


3a. Statistical properties of DM halos





3b. Halo density profiles



- Compute r_{200} and M_{200}
- Plot