Hands-on Numerical Project

Due: Thursday, December 7, in class

Goal: Run and analyze a cosmological simulation with Gadget-4
Steps:

- Set up with the makefile and parameter file
- Run the simulations
- Post-processing using Python codes

The official Gadget documentation can be found here: `https://wwwmpa.mpa-garching.mpg.de/gadget4/`

Assignment:

- Write a structure finder to identify halos using a Friends-of-Friends linking algorithm, e.g. `http://online.itp.ucsb.edu/online/gclusters11/halo/pdf/More_Halo_GalaxyClusters_KITP.pdf`
- Identify the **3rd** most massive halo at $z = 0$
- Plot the density profile of this halo
- Compare this to the NFW prediction
- plot a histogram for the halos mass function

You should work in groups of 3 students. There is nothing to hand in, but you should prepare a short presentation ~ 15 minutes to present in class on December 7.

1 Download the pre-packed setup from Canvas

→ Contains: code, I/O reader, simple post-processing example
→ Requires: C, make, FFTW, GSL, HDF5, MPI, Python (Numpy, PyTables, ...)
Content:

- gadget4: folder contains all files to build Gadget
  - Config.sh : contains options for compilation
  - Makefile.systype : choose machine for which to compile

  - Makefile: chooses correct path and option file in buildsystem, depending on Makefile.systype
  - buildsystem: contains files containing paths to libraries and compiler options
  - src: folder containing code

- output: empty folder for snapshots
- outputs.txt : file contains scale factors for snapshots
- params.txt : contains run time options
- fof.py : can be extended for the FOF algorithm, contains function to read snapshot file

In this setup, we do not need initial conditions, which is different than many simulations (e.g. Illustris). The initial conditions are made during runtime since we have set in the Config file how many particles, how large a box, and the cosmological density.

2 Understanding the parameter file

The parameter file, `param.txt`, sets runtime options without needing to recompile the code. A full documentation can be found here: **`https://wwwmpa.mpa-garching.mpg.de/gadget4/05_parameterfile/`**. In the downloaded file we added comments as explanation after the % symbol. The units for each parameter match the units set in System of Units in cgs section in the parameter file.

3 Understanding the config file

Unless specified otherwise, the code will compile with the options specified in `Config.sh`. If any of these options are changed, the code must be recompiled. Ideally, one should call `make clean` and a new `make`. A full documentation can be found unter: **`https://wwwmpa.mpa-garching.mpg.de/gadget4/04_config-options/`**.

4 Building the code

Enter the code folder `gadget4`. The most important part is the `Makefile.systype` file, that allows you to the set the variable `SYSTYPE`. As you can see in `Makefile`, this variable is used to load a file containing the correct library paths and another file containing the exact compiler options. For a LINUX system set `SYSTYPE="Generic-gcc"` and for a Mac `SYSTYPE="Darwin"`. To compile properly, you must have `gsl`, `hdf5`, and `fftw3` installed.

To build the code, type in the command line within the `gadget4` folder:
`make`

and the output should resemble:

```
Build configuration:
SYSTYPE: "Generic-gcc"
CONFIG: Config.sh
EXEC: Gadget4

...

build/fo/fof_findgroups.o build/fof/fof_nearest.o
build/fof/fof_io.o build/fof/foftree_build.o build/ngenic/ngenic.o
build/ngenic/power.o build/ngenic/grid.o build/compile_time_info.o
build/compile_time_info_hdf5.o build/version.o -lm
-L/opt/homebrew/Cellar/hdf5/1.14.1/lib -lhdf5 -lz
-L/opt/homebrew/Cellar/gsl/2.7.1/lib -lgsl -lgslcblas
-L/opt/homebrew/Cellar/fftw/3.3.10_1/lib -lfftw3f -o Gadget4
```

There will likely be many warnings during compilation, but this is fine as long as there are no errors and the compilation completes, i.e. an executable is generated. If the library paths cannot be found, check the imported library file (e.g. for `SYSTYPE="Generic-gcc"` it is `buildsystem/Makefile.gen.libs`) and add the correct paths.

5 Run the code

To run the code we leave the code folder and create a symlink with `ln -s gadget4/Gadget4`. We now have to decide how many MPI Tasks to use. This depends on the number of CPU cores your computer has. To start the simulation we call `mpirun -np 4 ./Gadget4 param.txt` , where 4 can be replaced by the number of cores. This starts the program `Gadget4`, which uses the paramter file `param.txt` . The simulation will now start to print ouput after each time step, that should look like this:

```
Sync-Point 316, Time:  0.0563892, Redshift:  16.7339, Systemstep:
0.000228555, Dloga:  0.00406141, Nsync-grv:  884736, Nsync-hyd:
0
```

The simulation ends at `Time = 1`. Since the simulation will take several hours, it is possible to stop and restart the simulation. In param.txt we can set the option `CpuTimeBetRestartFile` (set to 1800 at the moment). This means the code will write

restart files every 30 minutes, which contain all information to restart the simulation. To restart from the latest restartfiles, we have to start the code with `mpirun -np 4 ./Gadget4 param.txt 1` (important! use the same number of cores as used to write the restartfile).

6 Understand and read the output

The code creates several files. The important ones are the three snapshot files in the `output` folder. They contain the full data of all particles in the simulation at the scale factors we set in `outputs.txt` . We will concentrate in this project on the universe at redshift $z = 0$, which means the file `snapshot_002.hdf5`. It is written in the `hdf5` format, that can be read easily in python with the `h5py` package. To make the reading easier, we already added the file `fof.py` that can be used to read the snapshot. The positions are in Mpc, which means between 0 and 40. The mass of the dark matter particles is the same.

7 Instruction for the assignment

- As a first step write a FoF algorithm. As the linking lenth use 0.2 times the average particle distance. Run your FoF algorithm on the last snapshot to create a catalog of halos at $z = 0$.
- Since the two largest haloes undergo a merger, identify the 3rd largest halo at $z = 0$ and plot its radial density profile. Between 30kpc and 1Mpc the profile should be approximately desribed by an NFW profile. Fit an NFW profile to the simulation data. In addition, estimate the value for $R_{200,\mathrm{crit}}$ and find the concentration parameter for this halo.
- Create a histogram of the halo masses, similar to a halo mass function. Since one requires some amount of particles to resolve a halo, only take into account halos with at least 20 particles.

8.902 Astrophysics II
Fall 2023