CosmoTools18
Exercise on Cosmological N-body Simulations

Ruediger Pakmor

April 26, 2018

1 Cosmological structure formation

In this exercise, we construct initial conditions for a typical simulation of cosmic structure formation in the ΛCDM cosmology, including only dark matter.

We go through the following steps:

- Download and setup the GADGET4 code
- Construction of initial conditions (using NGENIC that is included in GADGET4)
- Running the simulation on a parallel computer with the GADGET4 code
- Visualizing the cosmic large-scale structure at different times
- Computing the halo mass function from the results of the friends-of-friends (FOF) group finder
- Calculating the fraction of mass that is in halos at different redshifts
- Finding the most massive halo
- Determining the density profile of the biggest cluster that has formed
- Plotting the matter powerspectrum at different redshfits

1.1 Download and setup the `GADGET4` code

Download the archive with the current development version of GADGET4 from `https://wwwmpa.mpa-garching.mpg.de/~rpakmor/NBodyExercise.tar.gz` . Note that this is an early development version and should not be distributed at the moment.

To install the code you need a few libraries installed, that are:

- MPI (any version)
- FFTW (version 3)
- GSL (any reasonably recent version)
- HDF5

To analyse the output you need to be able to load hdf5 files. If you want to use the python scripts to load snapshots, powerspectra and the output of the FOF halo finder that are provided with the code archive you downloaded you will need a python installation with numpy/matplotlib and h5py.

Gadget comes with 2 files that need to be configured, the *Config.sh* file in the gadget4 directory that contains options that need to be configured before the code is compiled and the *param.txt* that contains the runtime options for the code.

1.2 Generation of initial conditions

We would like to run a cosmological simulation with cosmological parameters $\Omega_{\mathrm{d}m} = 0.3$, $\Omega_\Lambda = 0.7$, $h = 0.7$, $\sigma_8 = 0.9$. As numerical parameters, we pick a boxsize $L = 150000\, h^{-1}\mathrm{k}pc$ and particle number $N_p = 64^3$ (or $N_p = 128^3$ if you have enough cores available). To be consistent with your pears, let's pick all the same random number seed – let's say 181170, such that the same realization is created. Pick a starting redshift of $z_{init} = 127$. Select Efstathiou's fitting formula for the initial power spectrum (by setting PowerSpectrumType to 1).

1.3 Run the problem with Gadget4

The next step is to run the simulation with the Gadget4 code. To this end you need to make sure that the configuration of its parameterfile and configfile are correct and that you selected a working systype in Makefile.systype (copy from its Template file, if there is no working systype for your system you may need to edit the Makefile directly and add one).

If you want to compile on the local RWTH cluster, you need to change to a more recent gcc compiler by executing

```
source /cvmfs/sft.cern.ch/lcg/external/gcc/6.2.0/x86_64-slc6/setup.sh
```

in your terminal. Then you load the mpi library with

```
module load openmpi-x86_64
```

and set

```
export SYSTYPE=Generic-gcc
```

Then you can compile the code with

```
make -j 4
```

On a Macbook use

```
export SYSTYPE=Darwin
```

instead.

Check in particular the system of units and that the gravitational softening length is of order $\sim 1/30$ of the mean particle spacing, fixed in comoving coordinates. With 4 compute cores, the simulation should take of order 0.5 hours for $N_p = 64^3$.

1.4 Let's take a look at it

In case something has gone wrong in a simulation, a simple simulation image will often be quite revealing, as the human eye is very good in spotting suspicious artefacts. While images are in no way conclusive, it's hence always a good idea to make some!

So construct a few slices through the simulation box. As a start, make simple dot plots where you show all the particles in a slab through the simulation box, projected down to a plane. For definiteness, project along the z-direction and show the slab $[0, L] \times [0, L] \times [0, L/5]$, where L is the boxsize.

How would you characterize the evolution of the cosmic structures?

1.5 Group finding

One of the most basic analysis usually carried out in cosmic structure formation simulations is group finding. The simplest and most widely used algorithm for this is FOF, which we apply to our simulation.

Plot the halo mass function $dn/dlogM$ for the halos found by FOF.

Then calculate the fraction of all mass contained in halos for the different snapshots. How does it evolve with time?

1.6 Structural properties of the largest halo

Make a dot-plot of the largest halo at the different output times.

Let's now study the internal structure of the most massive halo, and check how it evolves at late times. Write code to measure the spherically averaged density profile. This requires identification of the correct center of the halo. Don't use the center-of-mass of the particles that make up the FOF group (why?). Rather, iteratively find the densest point of the halo by shrinking a sphere that you re-center in each iteration onto its current center.

Use logarithmic bins in radius (say ~ 20), ranging from the softening length to about 3 times the virial radius. Include all particles in the simulation when you make the binning, not just those linked into the halo by the FOF method.

Overplot the density profiles in physical coordinates at the different output times, and interprete what you see.

1.7 Matter powerspectrum

Finally, if you have time left, we want to have a look at the matter powerspectrum of our simulation.

Use the provided python script to load the powerspectrum computed by the code. The powerspectrum is tabulated for 2000 bins, way too much for our poorly resolved simulation. To obtain a useful powerspectrum write a routine that rebins the original powerspectrum to a small number (say 50) bins. Think about which quantity is conserved in the rebinning. Then plot the powerspectrum at different times. How does it change?

If you got here and still have time left, try to calculate a powerspectrum of the halos instead of the total matter distribution. To this end write a new snapshot that contains one particle for each halo with its mass at its position. Then run Gadget with restart option 4 (Calculate a matter power spectrum) for your new snapshot. Compare the powespectrum of the halos with the matter powerspectrum of the full simulation.