PiTP 2009: “Computational Astrophysics” Lectures on Collisionless Dynamics and SPH
Exercise Set 1

Volker Springel

July 12, 2009

1 Cosmological structure formation: Into the future

In this exercise, we construct initial conditions for a typical simulation of cosmic structure formation in the ΛCDM cosmology, including only dark matter. For fun, we will run the simulation not only until $z = 0$, but also a bit into the future to see what happens when the Universe becomes dark energy dominated.

We go through the following steps:

- Construction of initial conditions
- Running the simulation on a parallel computer with the GADGET2 code
- Visualizing the cosmic large-scale structure at different times
- Analyzing the cosmic energy budget
- Finding halos with a friends-of-friends (FOF) group finder
- Determining structural properties (density profile and velocity anisotropy) of the biggest cluster that has formed
- Measuring σ_8 in different ways

1.1 Generation of initial conditions

Download a version of the IC-code N-GenIC that I have written at some point (from the web-site of the school). It is controlled with a simple ASCII parameterfile, which you should modify accordingly for the task at hand. An example parameterfile is included in the code distribution.

We would like to run a cosmological simulation with cosmological parameters $\Omega_{\rm dm} = 0.3$, $\Omega_\Lambda = 0.7$, $h = 0.7$, $\sigma_8 = 0.9$. As numerical parameters, we pick a boxsize $L = 150000\, h^{-1}$kpc and particle number $N_p = 128^3$ (or $N_p = 64^3$ if you are in a hurry and the computers are busy). To be consistent with your pears, let's pick all the same random number seed – let's say 123456, such that the same realization is created. Pick a starting redshift of $z_{\rm init} = 63$. Select Efstathiou's fitting formula for the initial power spectrum, and feed the code a Cartesian grid for the unperturbed particle distribution.

The IC code is MPI-parallel, and is run with the parameterfile as argument. (However, as the problem size is still small, you may also run it in serial.)

1.2 Run the problem with Gadget2

The next step is to run the simulation with the Gadget2 code. To this end you need to configure its parameterfile and makefile correctly. Most of this should be straightforward, but here are a few suggestions:

Pick the TreePM gravity solver, with a suggested mesh-size of PMGRID = 256^3 for the 128^3 particle set-up. We would like to run the simulation out to a scale factor of $a = 8$. (Aside: Calculate how many billion years this is into the future). I'd suggest to create output dumps at scale factors 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0.

Since we would like to look at the evolution of the energies among other things, ask the code to create measurements of the potential energy (this costs a bit additional computational expense), by setting the switch COMPUTE_POTENTIAL_ENERGY. Use something like TimeBetStatistics=0.125 to set the frequency of these outputs.

Pick a gravitational softening length of order $\sim 1/35$ the mean particle spacing, fixed in comoving coordinates. With 8 compute cores, the simulation should take of order 2 hours, so this is not completely negligible any more. Exchange data with fellow students or reduce the particle number to $N_p = 64^3$ if you are impatient or computer resources are momentarily congested.

Further note: Recently, I actually fixed a minor issue in the public release of the Gadget2 code, which under certain circumstances could negatively affect the late-time accuracy of simulations that are run far into the future – like the one we are trying here. It is therefore recommended to fetch the new version (2.0.4) from Gadget's web-site.

1.3 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 at late times?

1.4 Cosmic energies

Among other log-files, the code should have produced an ASCII file, *energy.txt*. It contains global measurements of kinetic energy and potential energy that we want to look at next. The third column gives the total potential energy, and the forth column contains the total kinetic energy in peculiar motion – these are the two energy terms in the cosmological Hamiltonian. Make a plot of the kinetic energy and the energy in the peculiar gravitational potential as a function of scale-factor (log scales). Disregard the highest redshift output, as the zero point of the potential energy will not go accurately to zero for $z \to \infty$ due to discreteness effects, unlike expected for the continuum. What's your interpretation of the results? Does this match your expectations?

Explain why both the kinetic energy and the potential energy scale $\propto a$ at high redshift.

Also make a plot of the total energy as a function of scale factor. Try to use the Layzer-Irvine equation for roughly checking energy conservation in the simulation. To this end, predict the expected total energy change relative to the initial time by numerically integrating the cosmic energy equation based on the measured values for kinetic and potential energy. Overplot this expected energy as symbols for each of the times in your *energy.txt* file.

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.

You can download (a rather old serial) version of a FOF group finder I once wrote from the school's web-site, or if you prefer, you can use one of the group finders that have been made publicly available (for example from the U. of Washington's N-body shop, or AHF from Alexander Knebe).

Let's use a standard linking length of 0.2 times the mean particle spacing. If you use the code I provide, note that its parameters are hard-coded in the beginning of the file *main.c*. Check that these settings are ok, and then apply the group finder to all the outputs you got. (Three parameters are expected, the output directory, the base name of the snapshot files, and the number of the dump that you want to operate on.)

Once the group finder has run, check its results by plotting the cumulative mass function at the different output times. (Optional: Overlay predictions from the Press-Schechter mass function formalism.)

Make a dot-plot of the largest halo at the different output times. (The provided FOF code includes IDL read-statements for the group catalogue it produces. For convenience, the group finder also produces an output file where the coordinates of the particles that make up a group are listed separately for each group, such that you can access them conveniently.)

1.6 Structural properties of the largest halo

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. (Alternatively, you could use the minimum of the potential.) 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. Also measure the radial (σ_r^2) and tangential velocity dispersion, $\sigma_t^2 = (\sigma_\phi^2 + \sigma_\theta^2)/2$, in each shell (in terms of physical velocities), and the velocity anisotropy parameter $\beta(r) = 1 - \sigma_t^2/\sigma_r^2$ for the halo. Interprete the results for the time evolution.

1.7 Checking the normalization of the simulation

Finally, we would like to check how well the simulation has reproduced the desired normalization in terms of the linearly extrapolated rms fluctuations σ_8 in spheres of radius $8\,h^{-1}$Mpc.

Implement two different methods for measuring σ_8 from the $z = 0$ output of the simulation:

- Write code that bins the particles onto a grid, and then convolves the density fluctuation field with a top-hat kernel of radius $8\,h^{-1}$Mpc. Do this by setting up the kernel in real-space. This is simple if you observe that parts of the kernel will sit in all the 8 corners of the real-space field, for the usual FFT storage convention. Also, make sure that the discretized kernel you construct is normalized to unity. Then carry out the convolution with discrete FFT transforms. Finally measure the rms of the density field to determine σ_8.

- Measure for a large number of random coordinates the enclosed mass density contrast around these random points. Use this to estimate σ_8, even though this is obviously not a computationally efficient approach.

Give reasons why the result for σ_8 you obtain is not exactly equal to the input value used for constructing the initial conditions.