COSMOLOGICAL N-BODY SIMULATIONS

A "worked example" of how to use cosmological N-body simulations to investigate structures formed in a cold dark matter-dominated Universe.

1. Introduction

Cosmological simulations play a leading role in studies of structure formation in the Universe. These simulations have helped establish the leading ΛCDM paradigm, which is based on the idea that the Universe is filled with cold dark matter (CDM) and whose expansion history is dominated now by some form of "dark energy" or "cosmological constant" (Λ).

Direct numerical simulations are often the only tools available to investigate the complex non-linear evolution of structure in the Universe. The following exercises are intended to give you a flavor of the kind of cosmological studies that N-body simulations can be applied to. The overall aim is to generate, evolve, and analyze a model ΛCDM Universe using realistic initial conditions and state-of-the-art codes. This model will be analyzed in order to answer detailed questions about the statistical properties of the nonlinear distribution of cold dark matter throughout a representative region of the Universe.

Because they are state-of-the-art, the numerical methods we shall use are fairly complex, and describing them in detail is well beyond the scope of the current exercise. Instead, the emphasis is on developing insight and intuition through a worked example, as an illustration of the kinds of problems that presently attract the attention of researchers in the field.

For simplicity, the examples we will work on deal exclusively with the dark matter component, which is the most amenable to traditional N-body treatment. The role of baryons, crucial as it is to the development of the observable Universe, will (sadly) be neglected. The conceptual differences in the evolution of collisional and collisionless fluids ("hydrodynamics" vs "stellar dynamics") result in different numerical approaches which would take much longer to explore meaningfully than the time available during this course.

2. Tools

We shall use a number of numerical tools that various researchers have developed and made freely available over the years. These are not the only ones available nor perhaps (in the opinion of some experts) the most efficient, versatile and powerful, but I have chosen them because they do not rely on commercially-licensed software, because they are easy to install, and because they are reasonably well documented.

To build a numerical model of the present-day non-linear distribution of cold dark matter in the Universe we need: (a) an initial-conditions generator, and (b) a cosmological N-body code to evolve these initial conditions into the non-linear regime. The suite of tools we will explore is completed with (c) a number of visualization and analysis tools. These will serve as a starting point for students to develop the analysis tools needed to complete these exercises. All necessary codes are available at

http://www.astro.uvic.ca/~jfn/astr508/probsets/NbodySim.tar.gz

Copy this tarball somewhere, unpack, and install the various tools in the /Software directory.

IMPORTANT: Although in principle all of these tools should run in any Linux-based system, each system has site-specific implementation issues. I have made sure that the codes run in our parallel cluster "LLAIMA". This will provide some valuable experience for those of you interested in running codes in a massively-parallel environment. You may need to request a username/password to use this system. Please contact the Research Computing Facility (http://rcf.uvic.ca/). Mention my name if you are asked for a reference. Consult the LlaimaNotes.txt file for some tips. You may wish to use LLAIMA to set up and run the simulations, but it might be easier if you bring the data to your own workstation for analysis and visualization.

2.1 Initial Conditions

Two options are available.

(a) To use the package COSMICS *(~/NbodySim/Software/ICs/COSMICS/),* developed by Ed Bertschinger at MIT (http://ascl.net/9910.004). This is a suite of FORTRAN programs that may be used to computer transfer functions and cosmic microwave background (CMB) anisotropy for a variety of cosmological models. It also generates Gaussian random initial conditions suitable for nonlinear structure formation simulations. It is a bit more flexible than option (b), but it generates files whose format and units need to be converted to those readable by the N-body code that we shall use, GADGET.

(b) To use the N-GENIC (*(~/NbodySim/Software/ICs/N-GenIC)* package written by Volker Springel. This allows one to generate files that are directly readable by the GADGET code.

The worked example below adopts (a) but feel free to use (b) instead if you wish.

2.2 Cosmological N-body Code

We shall use the (excellent!) N-body code GADGET-2, written by Volker Springel of the Max-Planck Institute for Astrophysics (Springel, V. 2005, MNRAS, 364, 1105, http://www.mpa-garching.mpg.de/gadget). This is a massively parallel TreeSPH code able to evolve a collisionless fluid with traditional N-body methods and an ideal gas using the Smooth Particle Hydrodynamics (SPH) technique. The cosmological treatment marries a tree-based approach with a particle-mesh (PM) technique, with optimal results in terms of efficiency at various stages of the evolution of a system. We shall use only the N-body capabilities of this excellent and versatile numerical tool. You may find a copy at *~/NbodySim/Software/GADGET/*. I have used the version 2.0.5 of this code but there is an updated version 2.0.7 that you may try as well. Either should work fine for our purposes.

2.3 Analysis Tools

Again, two options.

(a) GADGET comes with a few IDL scripts that illustrate how to read the data output. Feel free to use those and adapt them (or write your own).

Or,

(b) use the tools developed by the HPCC group of the University of Washington (The "N-body Shop"), available from (http://wwwhpcc.astro.washington.edu/tools/tools.html).

These are stand-alone C programs developed for the analysis and visualization of numerical simulations. For these exercises, we shall use codes available at *~/NbodySim/Software/Analysis/pkdtools*

- **TIPSY**: Theoretical Image Processing System
- **SMOOTH:** Code for calculating mean-field quantities in N-body simulations, for example the density at every particle position.
- **FOF:** A friends-of-friends group finder for N-body simulations.

One of the challenges that arise from using a variety of sources is that each of these codes deals with data using its own formats and units. I will make available some simple tools to facilitate the exchange of data between these codes. However, I encourage you to look at these only as examples and not to treat them just as "black boxes". Much may be learnt from reverse-engineering these few simple examples. In particular, you may find the following useful:

- **COSMICS2GDT***(~/NbodySim/Software/Analysis /cosm2gdt/)***:** a simple code that allows for transformation of format and code units between the COSMICS and GADGET codes.
- **GHEAD***(~/NbodySim/Software/Analysis /G_head/)***:** a simple code that reads a GADGET snapshot file and outputs some basic information from its header. This may be used to check quickly the redshift of a snapshot, the integrity of the file, its endianness, etc.
- **GADGET2TIPSY***(~/NbodySim/ Software/Analysis /gadget2tipsy/)***:** allows for switching between the formats and conventions used by GADGET and TIPSY. SMOOTH and FOF, as well as other tools from the N-body Shop use binary TIPSY format as input.
- **SM** (available system-wide on Linux boxes) is a useful interactive plotting program written by Robert Lupton and Patricia Monger

(http://www.astro.princeton.edu/~rhl/sm/). It is *not* free but is available at most astronomical institutions.

Finally, you may need to develop some analysis tools on your own. Feel free to use any programming language you feel comfortable with. No one will check your programming; only the results!

EXERCISE 1

Aim: to generate initial conditions appropriate for the ΛCDM paradigm on a representative box.

Procedure: After familiarizing yourself with the COSMICS package (i.e., read the manual!), use the tool GRAFIC to generate positions and velocities for 10^6 particles (a 100x100x100 grid) in a cubic box of size 100 Mpc/h on a side. Adopt a *flat* geometry, and choose a simple fit to the CDM transfer function to modify the scale-invariant spectrum. Experienced and/or adventurous students may want to tinker with the LINGER tool, which allows for much more complex transfer functions, including the effects of baryons, relativistic species, and more.

You will need to choose values for the matter density parameter, Ω_m, for the vacuum energy density parameter, Ω_Λ, and for the present value of Hubble's constant, H_0. Ignore the contribution of baryons and other forms of matter-energy. Use the results from the Planck team to set these parameters.

Questions:

1.1) List the parameters used to initialize the model. How did you ensure a flat geometry?
1.2) How did you normalize the power spectrum? (Hint: what value of σ_8 did you choose and why?)
1.3) Why is the box 100 Mpc/h in size? What determines this size? A qualitative answer would suffice. Bonus for a quantitative one.
1.4) As in the previous question, but referring to the redshift at which the initial conditions are set. How is this chosen? What would you need to do to change the initial redshift?
1.5) Visualize the positions of the particles. You may do this using TIPSY (see exercise 2), or you can use any plotting program you wish. Because GRAFIC works displacing a cubic grid, the outline of the grid is still easily recognizable. How would you use the perturbed positions and velocities to "undo" the displacements and to return each particle to its unperturbed grid position? Try it and show graphically that you recover the unperturbed cubic grid.
1.6) Discuss possible artifacts that may result from the choice of a cubic grid. Can you think of ways to remediate or at least ameliorate them? (Hint: you may want to consult the MAKEGLASS compiler option in the GADGET code).

EXERCISE 2:

Aim: to port the GRAFIC output to GADGET format.

Procedure: write a simple code that reads the output from GRAFIC (p3m.dat) and turns it into GADGET format. Use units of Mpc/h for the positions, of 10^{10} Msun/h for masses and of km/s for velocities.

Hint: look at the code in *~/NbodySims/Software/Analysis/cosm2gdt/* for a worked example but do *not* take its choices for granted! Please make sure to write all the particles in species #2. Gadget treats species#1 as gas particles.

Questions:

2.1) Describe the computation of the particle mass.

2.2) Although rescaling the positions to the new GADGET units is trivial, scaling the velocities is less so. According to COSMICS2GDT, $\mathbf{V}_{\mathrm{GADGET}}=\mathbf{V}_{\mathrm{COSMICS}}/\sqrt{a}$, where a is the expansion factor. Is this correct? Why? Work out this conversion in detail.

EXERCISE 3

Aim: to evolve a 100 Mpc/h box into the non-linear regime (up to z=0) using the cosmological code GADGET.

Procedure:

a) Read the user's guide for GADGET-2 and the accompanying paper, available from *http://www.mpa-garching.mpg.de/gadget*
b) Compile the code GADGET-2 so as to evolve a periodic box in comoving coordinates. Choose PMGRID=256, or 128 if memory available is short. Enable DOUBLEPRECISION and SYNCHRONIZATION.
c) Edit the *input.param* file to choose parameters appropriate for your run. (Hint: use the template provided in *~/NbodySim/TestRun/input.param* and modify the necessary parameters, after inspecting them all!) Set the spatial resolution of the force calculations to be 1/40$^{\mathrm{th}}$ of the mean interparticle spacing.
d) Edit the *~/NbodySim/TestRun/mpi-batch* file to choose parameters appropriate for your run. Or use *mpirun* interactively if batch queues are not available in your system.
e) Choose output parameters so that the code outputs about 11 snapshots between the initial redshift and $a=1$, for z=10,9,…,1,0.
f) Run it!

Questions:

3.1) Describe the choices of parameters in *input.param.* Which ones did you change from the template provided and why?

3.2) Can you think of an easy way to speed up the run time by a factor of 2 or 4 without compromising the numerical accuracy of the evolution? What would you need to sacrifice to get this? Describe how and run the code again. Check that your speedup is actually about what you expect. Make sure to save the snapshots and output files (e.g., run everything in a different directory!) of these runs, you may wish to use them later.

EXERCISE 4

Aim: to analyze the runs in order to explore the statistics of the non-linear evolution of dark matter.

Procedure:

a) In order to utilize the N-body Shop tools most effectively, first transform all snapshots to TIPSY format. Feel free to adapt the tool GADGET2TIPSY provided or write your own.
b) Note that the units preferred by the N-body Shop tools are quite different from those used by GADGET. In particular, positions are now given in a box of unit length, the total mass of the box equals Ω_m (so that densities are in units of the critical density for closure). What are the natural units of velocity that this implies, in units where the gravitational constant is G=1? Check the velocity conversion proposed in GADGET2TIPSY. Is this consistent with what you find?
c) If you have used GADGET2TIPSY for the conversion, the output is an ASCII file (useful for easy manipulation with various plotting programs such as SM). You will need to convert this into binary format in order to use the N-body Shop tools. This can be done by writing a simple C program, or by using the TIPSY-TOOL *ascii2bin* in *~/NbodySim/Software/Analysis/pkdtools/.* Alternatively, you can do this within TIPSY using the commands "*openascii 'ascii-file-name'* " and "*readascii 'binary-file-name'* ".
d) Once this is done, you are ready now to run some analysis tools. Start with running FOF on the z=0 snapshots, using a linking length equal to 20% of the mean inter-particle separation in the box, periodic boundary conditions, and a minimum size of 32 particles for the group. Follow up by running SMOOTH in order to estimate the local density of dark matter at the location of each particle.

Questions:

Answer 4.1 and at least ONE group of questions below.

4.1) **Overall Properties.** Visualize the groups identified by FOF using TIPSY. (Hint: use the *readarray* and *xarray* commands.) Focus on the location of the most massive groups: where are they located in the cosmic "web"? Zoom into some of them, "box" them, and rotate them to appreciate their 3D shape. Do they appear to be dynamically relaxed? Are more massive systems larger in size? More symmetric or less?

4.2) **Halo Mass Function**. Compute and plot the z=0 differential mass function of FOF groups and compare them with published results (see, e.g., Jenkins et al Jenkins et al. 2001, MNRAS 321, 372 or Springel et al. 2005, Nature 435, 629). How does the mass function change with time? Evaluate and plot it at z=1, 2, etc. When do the most massive clusters ($M>10^{14}$ M_{sun}/h) begin to form? How complete do you think is your inventory of groups at the low mass end? Design and carry a test to assess the minimum halo mass for which your mass function is reasonably accurate. Show this graphically.

4.3) **Halo Assembly History.** Using the groups identified by FOF at various times trace the formation history of halos. It is customary to define the "formation time" of a system when its most massive progenitor has a mass equal to half the final mass of the system. Write a code to estimate the formation redshift of a system; compute it and correlate this with halo mass at z=0. Which systems formed earliest/latest? What is the minimum mass for which you can estimate reliably the formation time? What problems have you encountered in the computation? What is the typical assembly time of halos of 10^{12} M_{sun}/h? These are thought to be the hosts of galaxies like the Milky Way.

4.4) **Halo Internal Structure.** Write a code that picks all particles in a group and resets them to its center of mass reference frame. Use physical rather than comoving units. (Do not forget to add the Hubble flow velocities!) . Compute for each system the total mass, potential energy, kinetic energy, and the angular momentum vector. For many questions we are interested in analyzing the equilibrium properties of a sample of halos, so it is important to weed the sample out of systems that are obviously out of equilibrium. One simple possibility is to use the virial theorem: how would you construct an "equilibrium estimator" to single out halos out of equilibrium? (Hint: In a relaxed system 2K+W=0.) Are there any "unbound" (E>0) groups? If so, discuss why FOF found these groups.

4.5) **Halo Equilibrium State.** Another possibility is to use departures from spatial symmetry, since halos undergoing rapid transformations (due to, typically, merging) will be inevitably distorted. How would you construct an "equilibrium estimator" that uses the spatial symmetry of the system? (Hint: in a well-relaxed halo the densest region will generally coincide with the center of mass.) Compare the results of both estimators. Which is most effective at picking out unrelaxed systems? Under what conditions will one estimator indicate equilibrium but not the other? Combine both in order to define a sample of systems in equilibrium.

4.6) **Halo Rotation**. Let us now explore the rotational properties of halos. Consider the *specific* angular momentum as a function of halo mass. Is there a net correlation? Does this mean that more massive halos "rotate faster" than their lower mass counterparts? Assuming that the halo of the Milky Way has the same specific angular momentum as the stellar disk, what can you say about the mass of the halo of the Milky Way? (Hint: you may approximate the angular momentum of an exponential disk with a flat rotation curve by $2 \times r_d \times V_{rot}$, where r_d is the exponential scalelength and V_{rot} is the rotation speed.)

Optional. When was the net rotation of a halo acquired? Trace the evolution of the specific angular momentum as a function of time for the particles making up the 5 most massive halos at z=0. Compare the evolution of angular momentum with the assembly of the mass. When are the rotation properties of a halo set? (Hint: you may find it useful to use TIPSY to plot the positions of the particles of a halo at the various characteristic times of the angular momentum evolution.)

In cosmology it is customary to characterize rotation by using a dimensionless rotation parameter, defined as $\lambda = J\,|E|^{1/2} / GM^{2.5}$, where E is the total energy of the system, J is the angular momentum, and M is the total mass. Discuss why this is a better way of assessing the rotation content of a halo and derive the value of λ corresponding to a particle in circular motion in a Kepler potential. Now compute λ for all halos identified by FOF.

Optional. Compute and plot λ vs halo mass. Do more massive halos "rotate faster" than their low mass counterparts? Compute the distribution of λ and fit it to a lognormal function. What is the mean and dispersion of this distribution? What does this imply for the role of rotation in the dynamical support of dark matter halos? Compare the distribution of λ parameters for equilibrium and unrelaxed halos. What differences can you point out?

4.7) **Mass Profile.** Let us look now at the spatial distribution of dark matter within halos. Compute and plot the spherically-averaged density profile and circular velocity profile of the 10 most massive dark matter halos. You may wish to use bins equally spaced in log(radius) and to plot all quantities in logarithmic units. Compare the shape of the density and circular velocity profiles of various halos and consider the choice of center. Is the center of mass the most appropriate choice of center in order to compute the halo mass profiles? Check how much the profiles change when a density-weighted center of mass is used instead. (Hint: you may compute densities at the location of each particle using SMOOTH.)

Discuss how you would determine the innermost radius for which the profile is accurate. At this point you may wish to look at the tests you ran in response to question 3.2.

Consider now the scaling properties of the mass profiles. Instead of using physical units for density, radius and velocity, scale all densities to the critical density for closure (ρ_{crit}), all radii to the virial radius of each halo and circular velocities to the virial velocity of each halo. (Define the virial radius as the radius where the mean inner density is 200 ρ_{crit}. The virial velocity is the circular velocity at that radius.) Plot again the density and circular velocity profiles. Comment on the similarity (or otherwise) of profiles scaled this way.

4.8) **Velocity Profiles**. Is the dynamics of dark matter halos dominated by radial or tangential motions? Verify this by computing the radial velocity of each particle around the most massive halo deemed to be near equilibrium. Extend this analysis to a distance of 5 virial radii from the center. (Note that for this you will need to consider all particles in the simulation rather than those assigned to the group by FOF.) Plot radial velocity versus radius. Identify the turnaround radius, the first infall region, and the region where velocities are dominated by the Hubble flow. Repeat for a few of the most massive halos. Discuss the meaning of the definition of the virial radius within this context.

Compute the velocity dispersion in the radial and tangential direction in radial bins. Is the halo velocity distribution isotropic? Does the anisotropy depend on radius?