

Due Dec 5th, 2014

Pick one of the 4 projects below, write a report similar to HW6 and prepare a 10-15 min presentation to be given in class on Dec 5, Dec 10 or Dec 12 (we will decide the order of the presentations later in class). The projects are described in some detail in the next pages.

You can work in small groups if you prefer (2-3 people) but each of you have to produce an independent report and presentation (or a section of a longer report/presentation). If you work in a group you are expected to produce a commensurately larger amount of work than if you work on your own (the grade will reflect the level of effort).

- **Project 1: Your first cosmological simulation of a black hole!**
- **Project 2: Parallelization of an N-BODY code using OPENMP**
- **Project 3: Parallelization of an N-BODY code using CUDA**
- **Project 4: 3D volume rendering**

PROJECT 1: YOUR FIRST COSMOLOGICAL SIMULATION OF A BLACK HOLE! (also known as secondary infall problem)

Write a report and prepare a class presentation on the results of your numerical simulations of dark matter accretion around a black hole at rest in an expanding Universe. This problem is quite difficult for an undergraduate level course: do not worry too much if your results do not agree with the expected solution!

In order to solve this problem you need to generalize the integration package you wrote for PS5 so that it can solve the N -body problem in an expanding Universe assuming spherical geometry and fixed time steps. The following ODEs should be integrated in order to solve the problem (see also my handwritten notes):

$$\frac{dx}{dt} = (1+z)v_p \quad (1)$$

$$\frac{dv_p}{dt} = -H(t)v_p - (z+1)^2 \frac{G\Delta M}{x^2}. \quad (2)$$

These equations describe the motion of spherical shells of dark matter around the black hole in comoving coordinates: $v_p(x)$ is the peculiar velocity of a shell at comoving distance x from the black hole and $\Delta M(< x) = M(< x)(t) - M(< x)(t = t_i)$ is the difference between the mass within x at time t and the original mass at time $t = t_i$ when the Universe was homogeneous (constant density). To convert the distances in physical units r use the formula: $r = x/(1+z)$, where the redshift z depends on time as $z+1 = (t/t_0)^{-2/3}$ and where $t_0 = 13$ Gyr is the age of the Universe. The Hubble parameter in a flat matter dominated Universe is: $H = H_0(1+z)^{3/2} = H_0(t/t_0)^{-1}$, where $H_0 = 70$ km/s.

Alternatively you can integrate the equation as a function of redshift z rather than time t (it may be easier). Using the relationship $dz/dt = -H(1+z)$ you get the following equation:

$$\frac{dx}{dz} = -\frac{v_p}{H(z)} \quad (3)$$

$$\frac{dv_p}{dz} = \frac{v_p}{1+z} + \frac{z+1}{H(z)} \frac{G\Delta M}{x^2}. \quad (4)$$

Assume initial conditions as follows: start the simulation at redshift $z_i = 3000$ and assume that the dark matter has constant density $\rho = 3.8 \times 10^{-29}$ g/cm³. Remember to place a black hole of mass $M_{BH} = 1 M_\odot$ at $x = 0$. If I failed to provide enough explanations to solve the problem feel free to ask me. Have fun!

Hints: The dark matter should create a spherical halo around the black hole with mass $M(< r) \propto r^{0.75}$ and have a total mass within the halo radius that increases with decreasing redshift as $M \propto 1/(1+z)$.

PROJECT 2: PARALLELIZATION OF N-BODY CODE USING OPENMP

Parallelize using OPENMP the N-body code you wrote for HW6. Use the N-body code to simulate an interesting astrophysical problem: a globular cluster of stars, a collision between two galaxies, collapse of dark matter in an expanding universe, etc. Plot the execution time as a function of the number of particles N and as a function of the number of OPENMP threads (for a fixed N). Produce a movie of the simulation.

PROJECT 3: PARALLELIZATION OF N-BODY CODE USING CUDA

Rewrite part or all the PP N-body code you wrote for HW6 to be run on GPUs. You can use CUDA programming language (Nvidia GPUs) or any other programming language for GPUs. Repeat PS6 and compare the performance of CPU and GPU as you increase the number of N-body particles. The essential part of the code that needs to run on GPU is the PP part (direct summation of particle forces).

PROJECT 4: 3D RENDERING OF VOLUMES

Learn how to use a visualization program like YT, Visit, ParaView, etc. Obtain a 3D data set from a simulation (from me or visualize your own simulation) and produce different types of visualization: slices, volume rendering, 3D contour plots. Produce a movie of a fly through the simulation or rotate the volume slowly.