N-Body project

Computational Astrophysics

ADVICE: using a low level programming language (such as for example C/C++) can significantly improve the performances of the code you are going to build up, especially in the second task, where time integration is involved. Higher level languages (such as Python) can be used, but be aware it will typically take way longer to complete a simulation, modulo optimization tricks (NumPy, ...).

Contents

1 Task1

1.1 Step 1

- Preliminarily, verify the form of the density function $\rho(r)$ by inferring it from the particle distribution and compare it with the analytical density function described in the original paper by Hernquist (from 1990 on Astrophysical Journal available on the web). Use Poissonian error bars[1] when comparing the numerical density profile with the analytical expected values.
- Note that the initial conditions are given in a system of units in which G=1. Assume reasonable units of length and mass for your calculations (units of velocity and time follow automatically from the assumption G=1) and discuss your choice.

1.2 Step 2

- Compute the direct N-Body forces between particles (note that the array potential[i] is not needed for this purpose). Start by assuming a softening of the order of the mean interparticle

[1]Poissonian error: the exercise requires to compare the Hernquist density profile with the density profile of the data. The simplest way to do that is to divide the space in spherical bins (shells) and count how many particles you have in all the bins, as you would do when building a histogram. Then you can compare these values to the expected values, i.e. the average amount of particles you would expect in each bin given the Hernquist density profile. In doing that you should consider that the number of particles you count in a given shell is a random variable that follows a Poisson distribution with λ = expected number of particles (average value). When you compare the 2 values you would need some error estimate to evaluate how similar they are, and the standard error you have is the standard deviation of the Poisson distribution, i.e. $\sqrt{\lambda}$. Please be careful about how you choose your bins, in order to have reasonable results.

separation[2] in the system, then repeat the force calculation by experimenting with different values of the softening and discuss your results.

- To check the direct force calculation result and its dependence on the softening choice, compare it with the analytical force expected based on the application of Newton's second theorem for spherical potentials[3] and plot the result (use the book "Galactic Dynamics" by Binney and Tremaine as main reference for the theoretical notions, in particular sec. 2.2 (most recent version of the book) or 2.1 (1987 version).

- Compute the relaxation timescale of the numerical model given the number of particles and the physical crossing timescale (use the half-mass radius R_{hm} and the circular velocity computed at the half-mass radius, $v_c = \sqrt{GM(R_{hm})/R_{hm}}$). Keeping in mind how the relaxation time formula is derived, do you expect varying the value of the gravitational softening to change the relaxation timescale? In particular, do you expect it to increase or decrease if the softening is increased above the interparticle separation? Can you explain why?

[2] Use the number of particles contained in the half-mass radius R_{hm} to estimate this value.

[3] If we assume spherical symmetry we have $\vec{F}(r) = -\frac{GM(r)}{r^3}\vec{r}$ and $M(r)$ is the mass included within the radius r.