

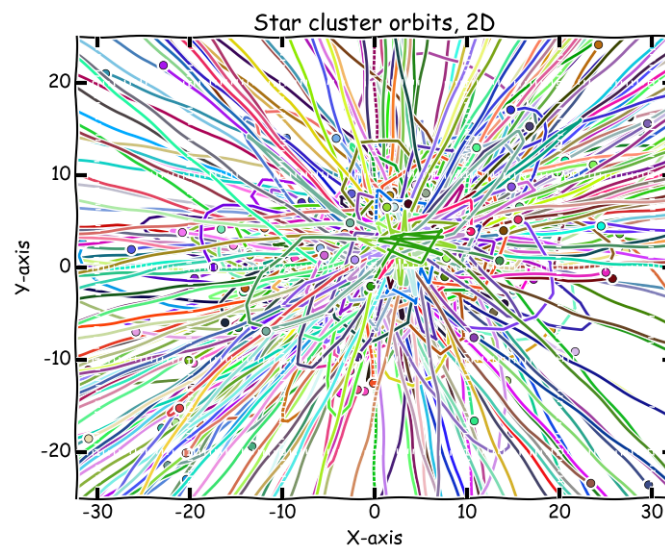


5. Continuation on Astronomical project - N -body simulation of an open galactic cluster

Due date **Nov 9th, 2016**

Magnus Christopher Bareid
un: magnucb

December 22, 2016



Abstract

This project is a continuation on a previous project given to the students this semester, in which the students were to gravitationally and numerically simulate the Solar System.

For the current project, the students are to simulate an N -body system, unrelated to the Solar system, and investigate properties of the system as it is simulated to reflect on the model's physical validity.

Contents

1	Introduction	1
2	Underlying math	1
2.1	Newton's laws	1
2.2	Numerical round-off correction	1
2.3	Time and gravitational constant	1
3	Programming	2
3.1	Update of main	2
3.2	Update of Velocity Verlet	2
3.3	Time steps	2
4	Experiments	2
4.1	Equilibrium without numerical round-off correction results	3
4.2	Equilibrium without numerical round-off correction discussion	3
5	Appendix	4
5.1	Code and GitHub	4

N-body simulation of an open galactic cluster

1 Introduction

For this project I will review the mathematical equations related to Newtonian gravitation and other relevant equations.

It is worth noting that the differential integration method that I utilize is the Verlet method, as opposed to the Euler or RungeKutta4 method which were mentioned in the previous astronomical project.

The physical properties which I will investigate through this project are:

- energy conservation in an N -body system and its related equilibrium,
- numerical instabilities from round-off corrections,
- validity of the virial theorem,
- radial distribution of stellar bodies in a gravitationally bound system.

2 Underlying math

The most important mathematical expression for an assignment investigating properties of a gravitational system, will be the equations gravitational force exchange.

2.1 Newton's laws

The force equation between two objects, will be as per Newton's law of gravity:

$$\vec{F}_{G_{i,j}} = -G \frac{m_i m_j}{r_{i,j}^3} \vec{r}_{i,j} \quad (1)$$

where i and j denotes which objects' force, in the system of objects' forces, are being measured, the m s represents their masses, r is the displacement between the two objects, and G is the Newton's gravitational constant.

Furthermore, Newton's third law of motion comes into place here, as objects are being iterated over, such that

$$\vec{F}_{G_{j,i}} = -\vec{F}_{G_{i,j}} \quad (2)$$

and the force experienced by the second object of the two equals the same force that the first object experiences, only in the opposite direction.

2.2 Numerical round-off correction

At a later point in the project, there is introduced a smoothing factor that helps deal with numerical round-off problems, modifying the force equation thusly:

$$\vec{F}_{G_{i,j}} = -G \frac{m_i m_j}{r_{i,j}^2 + \varepsilon^2} \frac{\vec{r}_{i,j}}{r_{i,j}} \quad (3)$$

wherein the new quantity ε represents the aforementioned smoothing factor, with which different magnitudes of values will be experimented.

2.3 Time and gravitational constant

Then it is the matter of the system's time scale. It seems arbitrarily that I should measure time in years for a problem that isn't related to Earthern years. Instead, time is defined through, and normalized by, an analytical expression for an N -body system's duration of collapse τ_{crunch} (or τ_c for short). The mathematical background and elaboration of τ_c is based on a topic of cosmology referred to as the "Spherical Top Hat Model" [3], but its formalism is already given in the project's details. Simply put, it describes the analytical time-frame in which a homogeneous fluid collapses in on itself to form a singularity.

From the expression of τ_c I then pull the gravitational constant, dependent on the system's density ρ_0 , and by extension the cluster's limiting radius R_0 .

$$\begin{aligned} \tau_c &= \sqrt{\frac{3\pi}{32G\rho_0}} \\ \Rightarrow G &= \frac{3\pi}{32\tau_c^2 \rho_0} , \\ \rho_0 &= \frac{\sum_i m_i}{\frac{4}{3}\pi R_0^3} , \\ \Rightarrow G &= \frac{1}{8} \frac{\pi^2 R_0^3}{\tau_c^2 \sum_i m_i} \end{aligned}$$

however, as I measure time as an iterated increment of τ_c , its value is normalized to 1, yielding:

$$G = \frac{1}{8} \frac{\pi^2 R_0^3}{\sum_i m_i} \quad (4)$$

3 Programming

The previous project utilized some small-scale general relativity, which at the scales of this current project is obsolete. Instead, the force equation is rewritten according to equation 1 and 3.

The layout of the class structures largely remain the same, and the main program basically runs the same way, with a few changes.

3.1 Update of main

Command line functionality is modified to take in the system's number of bodies \mathbf{N} , a time step size \mathbf{dt} , correction factor ε , and duration of the simulation.

Next is the random number generation modules initialization, with a uniform distribution of positions for the cluster's stellar bodies, and a gaussian distribution of mass between the stellar bodies, with a mean of 10 Solar masses and a standard deviation of 1 Solar mass.

The cluster's instance is then initialized with a given value for ε , as the calculation of forces and energies between stellar bodies is a function of this class.

The stellar bodies now need to be created. To ensure that the cluster's bodies are created in a spherically uniform distribution, the coordinates system in which they are first initialized is spherical, and then these are translated into cartesian coordinates. The bodies are then created with a gaussianly randomized mass value as described prior.

Now that the cluster's total mass is measured, the gravitational constant G is initialized as per equation 4, the datafile is opened and the numerical integration begins, producing the clusters' bodies' movement and energies unto the data.

3.2 Update of Velocity Verlet

As stated previously, the numerical integrator of the system utilizes the Velocity Verlet method. It should be noted that the previous project's utilization of this algorithm was extremely crudely implemented. For the purpose of this project, however, the algorithm has been cleaned up:forklare

```

1 void Verlet::integrateOneStep(Cluster &system){
2   system.calculateForcesAndEnergy();
3   // Velocity Verlet
4   for (CelestialBody &body :system.bodies()) {
5     // half-step computation
6     body.velocity += (m_dt/2.0)*(body.force / body.mass);
7     body.position += body.velocity*m_dt;
8   }
9   system.calculateForcesAndEnergy();
10  for (CelestialBody &body :system.bodies()) {
11    // the new velocity
12    body.velocity += (m_dt/2.0)*(body.force / body.mass);
13  }

```

This improvement of the code yields an efficiency boost at about 20% less time for a standard run of the program with 100 stellar bodies and 4000 time steps to iterate.

3.3 Time steps

Choosing a decent time increment's value, \mathbf{dt} , is essential to ensure that the program proceeds ahead in the best possible way.

To find try out a few resolutions, I constructed some initial test runs of the program. The initial parameters were set to $\mathbf{N} = 100$ stellar bodies, $\mathbf{R}_0 = 20$ light years for the stellar bodies' radial distribution's upper displacement limit, and trying out a few time resolutions of \mathbf{dt} being a varyingly small fraction of τ_c .

Having tried out a few time resolutions, I settled on $\mathbf{dt} = 0.001\tau_c$, as this seemed to yield a smooth accuracy for the integration and still be time efficient.

4 Experiments

This section holds all the experiments that were performed with the code. Where not otherwise specified, parameters \mathbf{R}_0 and \mathbf{dt} are set to 20 and $0.001\tau_c$ respectively.

4.1 Equilibrium without numerical round-off correction results

Simulating an $N = 100$ body simulation with no round-off correction term yielded this development.

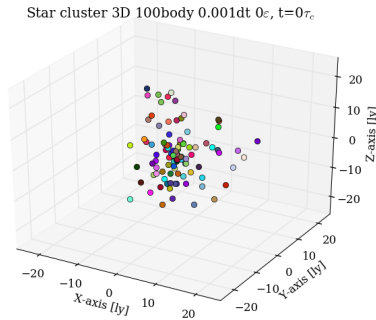


Figure 1: $N = 100$ body cluster simulation at $t = 0\tau_c$.

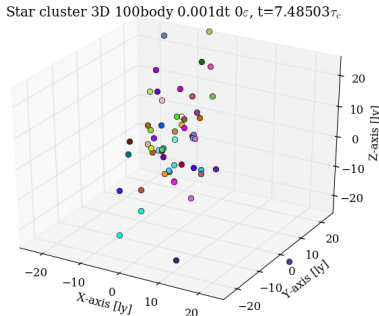


Figure 2: $N = 100$ body cluster simulation at $t \approx 7.5\tau_c$.

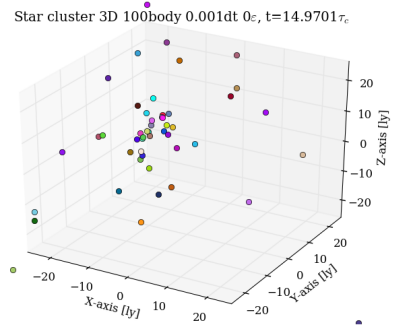


Figure 3: $N = 100$ body cluster simulation at $t \approx 15\tau_c$.

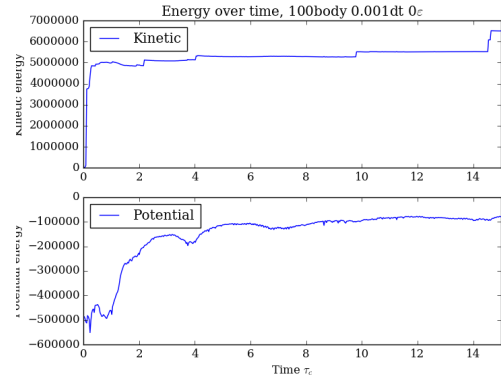


Figure 4: $N = 100$ body cluster's energy development for a duration of $t \approx 15\tau_c$.

4.2 Equilibrium without numerical round-off correction discussion

Having seen the development of the cluster by plots and animation, it seems that the cluster almost immediately ejects some of its contents, and as the rest of the structure collapses even more material is flung sideways and beyond at regular intervals when the stellar distribution seems to pulse, as indicated by the sudden increases in kinetic energy from figure 4.

It is clear that no matter for how long a duration one might choose for this set of initial conditions, a lot of stellar bodies are ejected and flung across space, effectively never to be bound by the cluster's potential well again.

One might consider the aforementioned apparent motion of pulsation to be the energies of the system reaching an equilibrium point,

which is the ideal for a gravitationally bound system. However, as such a large quantity of bodies that initially belonged to the cluster are ejected, then this seems a poor equilibrium for the initial system - even though the remaining core bodies seem to waltz around peacefully enough.

5 Appendix

5.1 Code and GitHub

All my code is located at this address:

<https://github.com/magnucb/p5>

References

- [1] Jensen, M 2016,
CodeName,
viewed 9th of December 2016,
URL
- [2] M. Joyce, B. Marcos, and F. Sylos Labini
2010,
Cold uniform spherical collapse revisited
viewed 9th of December 2016,
<https://arxiv.org/pdf/1011.0614v1.pdf>
- [3] M. Dijkstra,
AST4320: Cosmology and Extragalactic
Astrophysics,
viewed th of December 2016,
https://dl.dropboxusercontent.com/u/16515147/AST4320/LecturenotesAST4320_Nov30.pdf