UiO : Institute of Theoretical Astrophysics
University of Oslo

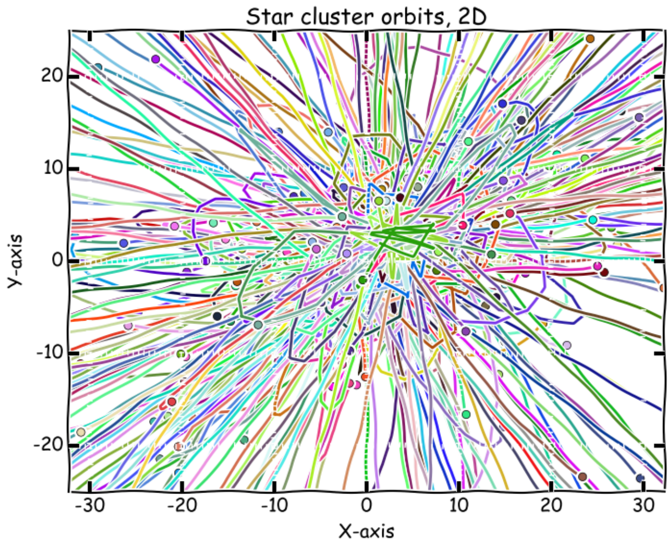
Fys3150/4150 - Computational Physics

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

Due date **Nov 9th, 2016**

Magnus Christopher Bareid
un: magnucb

December 15, 2016



Abstract

Empty void

Contents

N-body simulation of an open galactic cluster

1 Introduction, or picking up where we left off with the Solar system

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.

For this purpose, we may review the mathematical equations related to Newtonian gravitation, and other relevant equations.

It is worth noting that the differential integration method we will now utilize, will be the Verlet method, as opposed to the Euler or RungeKutta4 method.

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:

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

where i and j denotes which objects' force, in the system of objects' forces, are being measured, the ms 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

$$\vec{F}_{G_{j,i}} = -\vec{F}_{G_{i,j}} \tag{2}$$

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:

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

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 we should measure time in years for a problem that doesn't connect to years in any way. 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), from which we then pull the gravitational constant, dependent on the system's density ρ_0, and by extension the cluster's limiting radius R_0. The mathematical background and elaboration of τ_c is based on a topic of cosmology referred to as the "Spherical Top Hat Model" , but its formalism is aready given in the project's details.

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

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

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

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 **N**, a time step size **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 betwen 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 firstinitialized is spherical, and then these are translated into cartesian coordinates. The bodies are then created with a gaussially 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:

```
void Verlet::integrateOneStep(Cluster &system){
system.calculateForcesAndEnergy();
// Velocity Verlet
for (CelestialBody &body :system.bodies()) {
  // half-step computation
  body.velocity += (m_dt/2.0)*(body.force / body.mass);
  body.position += body.velocity*m_dt;
}
system.calculateForcesAndEnergy();
for (CelestialBody &body :system.bodies()) {
  // the new velocity
  body.velocity += (m_dt/2.0)*(body.force / body.mass);
}}
```

This improvement of the code yields an efficiency boost at about 30% less time for a standard run of the program with 100 stellar bodies and 4000 time steps to iterate.

3.3 Time steps

For the 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 a time resolution of $\mathbf{dt} = 0.001\tau_c$, just to start off.

4 Appendix

4.1 Code and GitHub

All my code is located at this address:
`https://github.com/magnucb/p5`

References

[1] Jensen, M 2016, `CodeName`, viewed $something^{th}$ of December 2016, `URL`.

[2] Jensen, M 2016, Computational Physics Lecture Notes Fall 2015:

13.2.2: Canonical Ensemble &

13.3.1: Ising Model and Phase Transitions in Magnetic Systems,

viewed *something*th of December 2016, `URL`