N-body Simulation Tutorial

**Introduction**

Today, we are using a small direct N-body integrator to simulate few-body interactions like three-body interactions. We will observe their motions and learn about the dynamical interactions between stars in globular clusters, galaxies and in the Universe.

**About the N-body Simulator**

The N-body simulator we use today is based on the ***leapfrog*** method for calculating the forces between the stars, their velocities and positions as a function of time. In theory, if we know the masses, initial positions and velocities of all the stars, we can calculate their velocities and positions using Newton’s second law .

For example, to approximate and , if the mass of one star is , and the total net force on the star is , the acceleration of the star is

.

Taking small time intervals , the new velocity at time can be written as

,

where is the initial velocity of the star. And the new position at time is

The leapfrog method calculates the velocities at half the time step , and the positions at one time step . In doing this, the simulation is more stable and it is easier to conserve the total energy of the stars. And yes, total energy and angular momentum of the system should be conserved.

All the codes are coded in ***Python***.

**Running the N-body Simulator**

*Step 1*: Go to

*Step 2*:Open the **Nbody\_plot.ipynb** file, on the menu at the top, click **cell**, and then click **Run all**.A box asking for input will show, type in **example\_binary**. This is an folder that contains an output file made for the dynamical interactions of two point-like stars, both of 1 solar mass. Watch the animation.

What can you see? You have a binary star system! The two stars orbit each other while moving together at a certain direction. They move at that direction because their total initial momentum points toward that direction.

*Step 3*: Now you are going to simulate a three-body interaction. Go to **Home,** open file **N-body\_simulator\_leapfrog.ipynb**, click **cell**, then click **Run all**. A few boxes will show asking for input, type in the numbers as follow,

Number of particles: 3

Number of steps (divisible by 100): 1000

Number of dimensions (2 or 3): 3

Specify the timestep: 0.01

Randomly generate masses?: yes

Randomly generate particle positions?: yes

Your input of the number of particles decides how many stars are going to be in the simulation. Number of steps tells the code how many time steps to take. Number of dimensions sets the dimension of the simulation, 2 is 2-dimensional (x & y axis), and 3 is 3-dimentional (x, y & z axis). Time step is from above in **About the N-body Simulator.** The smaller the time step , the more accurate the simulation is. For this step, let’s generate the masses and initial star positions randomly (the code will pick the masses and initial star positions for you within a range). If you put in no in both “Randomly generate masses?” and “Randomly generate particle positions”, you will need to manually put in the masses and initial positions of all the stars. We will do this later.

After you put in all the numbers, the code will run, and it will finish in 1-2 mins. It will generate a new folder with year-date-time as the folder name.

Now go back to **Nbody\_plot.ipynb**. Repeat step 1, but this time type in the new folder name instead of example\_binary. Watch the animation.

What’s different between the three-body interactions and the two-body interactions? It might be hard to see, but if we ignore the overall straight-line motion of the binary star, then it appears that the two stars are orbiting each other. Their orbit can be described by simple equations, like the Kepler’s Law. However, if there are three stars in a system,

**Questions**

1. If there are two stars that are stationery in space with masses m1=5 solar mass and m2=1 solar mass. The separation between them are 0.005 pc.
2. If you live on a planet with three stars (e.g. three Suns), will you be able to see