

Gravitational N-body Problem

Kamalesh Paluru
Omer Sipra

March 25, 2023

Summary of Contributions

Omer Sipra

- Worked primarily on documentation and theory for the N-body problem.

Kamalesh Paluru

- Implemented the theory and created a program to simulate an N-body system.

Interim Report

Classical 2-body Problem

We decided to start with the most basic case: The classical 2-body problem. Once we have the necessary classes, we can make them more general to support more complicated systems. The two base classes are **Body** and **System**:

- **Body**: This class represents a body in our N-body system.
- **System**: This class takes a list of **Body** objects and contains methods to modify an instance of the system (and the bodies in it).

Using OOP made it much easier to create a strong foundation and will help make it easier to generify the program.

`System.simulate(until=time)`

- This method mutates the body object in **System**, it hence returns the updated positions and velocities of the bodies.
- The equations used for the classical 2-body system are relatively simple, **simulate** algorithmically updates the positions and velocities of each body:
 - Repeats the following steps for every time differential in a given time interval:
 - Update the velocity of every body using:

$$\vec{v}_f = \vec{v}_i + d\vec{v}$$

$$\vec{v}_f = \vec{v}_i + \vec{a} \cdot dt$$

$$\vec{v}_f = \vec{v}_i + \frac{\vec{F}_{\text{net}}}{m} \cdot dt$$

Here; m is the mass of the body, \vec{v}_i is the initial velocity, \vec{F}_{net} is initially calculated by the constructor (and later updated) when **System** is initialized, and dt is the given time differential.

- Update the position of every body using:

$$\vec{x}_f = \vec{x}_i + d\vec{x}$$

$$\vec{x}_f = \vec{x}_i + \vec{v}_i \cdot dt$$

- Tick the clock:

$$t_f = t_i + dt$$

- Finally, we update the net force on each body using the updated positions and velocities.

Validation

- Unfortunately, the position of `earth` doesn't have the expected sequence of values. We highly suspect there is something wrong with the way the position/velocities are updated. `sun` however does seem to stay close to the origin as expected.
- Although we can't produce plots just yet (for the reason above), we have a strong foundation to build upon with these base classes.