Modeling and Simulation of Three-Body Problem

Zibo Wang

May 2019

1 Introduction

The three-body problem is the problem of taking the initial positions and velocities (or momenta) of three point masses, and solving for their subsequent motion according to Newton's laws of motion and Newton's law of universal gravitation. A three-body problem in general has no closed-form solution, an expression that can be evaluated in a finite number of operations.

The most common way to approach a three-body is numerical methods. In this project, I went through one numerical method with following scheme:

1. Based on the current position of the objects, calculate forces acting on these objects.
2. Update momenta of objects using the forces over a small time interval.
3. Calculate position of objects at the end of the time interval, using the momenta.
4. Repeat above steps.

Details of this scheme will be discussed in the next section.

2 Method

From Newton's law of universal gravitation, force between two objects are

$$F = G\frac{m_1 m_2}{r^2}, \tag{2.1}$$

where G is the gravitational constant ($6.674 \times 10^{-11} N \cdot (m/kg)^2$), m_1 is the first mass, m_2 is the second mass, and r is the distance between two objects. In vector form, the equation (2.1) can be rewritten as

$$\begin{aligned} \vec{F}_{21} &= -G\frac{m_1 m_2 \hat{r}_{12}}{|\vec{r}_{12}|^2} \\ &= -G\frac{m_1 m_2 \vec{r}_{12}}{|\vec{r}_{12}|^3}, \end{aligned} \tag{2.2}$$

where $\vec{F}_{21}$ is force applied on object 2 exerted by object 1, $\vec{r}_{12}$ is the is the distance between objects 1 and 2, i.e. $\vec{r}_{12} = \vec{r}_2 - \vec{r}_1$, and $\hat{r}_{12}$ is the unit vector from object 1 to 2. Therefore, if positions of three objects a given, we are able to calculate net force on each object on that moment.

Newton's second law of motion states that the acceleration of a body is proportional to the force applied,

$$\vec{F} = m\vec{a}. \tag{2.3}$$

A more professional form for this law is

$$\vec{F} = \frac{d\vec{P}}{dt}, \tag{2.4}$$

where $\vec{P}$ is the momentum vector, and acceleration is expressed in term of the rate of change of momentum. This can be approximated as

$$\vec{F} \approx \frac{\Delta \vec{P}}{\Delta t}, \tag{2.5}$$

for a small time interval Δt. Rearrange the above equation gives change of momentum in the time interval

$$\Delta \vec{P} \approx \vec{F} \Delta t. \tag{2.6}$$

Therefore, if at time t, the momentum of an object is known to be $P(t)$, its momentum after a small time interval Δt, is approximately

$$\vec{P}(t + \Delta t) \approx \vec{P}(t) + \vec{F} \Delta t. \tag{2.7}$$

Using the momentum of the object, we are able to calculate its velocity

$$\vec{v} = \frac{\vec{P}}{m}. \tag{2.8}$$

Then if at time t, the object is at position $\vec{r}(t)$, we can use this velocity approximate the final position of the object after the small time interval

$$\vec{r}(t + \Delta t) \approx \vec{r}(t) + \vec{v} \Delta t. \tag{2.9}$$

Therefore, in theory, given initial positions and velocities (or momenta) of n objects, we are able to find the net force on each object. Then we are able to approximate their momenta, velocities, and positions after a small enough time interval. Then using the new positions to repeat the steps and simulate their motion.

3 Result

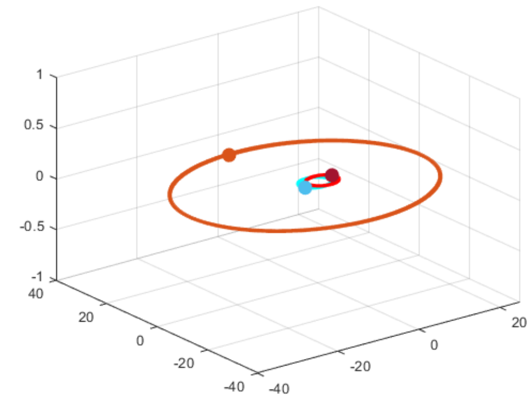
Based on the idea of the preview section. I wrote a MATLAB function that takes masses, positions, and momenta of three object, and simulate their motion.

I tried initial conditions of several stable solution. Fig. (1a) shows a typical binary star system with a planet. Because the mass of the planet is much small compared to mass of a star, and the planet is far away from two stars, it has a very small influence on the binary star system, and the system is stable. Fig. (1b) is the solution of The Sitnikov problem. The third object is placed in the center of mass of a binary star system, with velocity (or momentum) perpendicular to the plane of the binary star system. In this case, the third object and the binary system just oscillate up and down about the center of mass of the whole system. Fig. (1c) shows the Figure-Eight solution, in which three bodies share a figure-eight orbit.

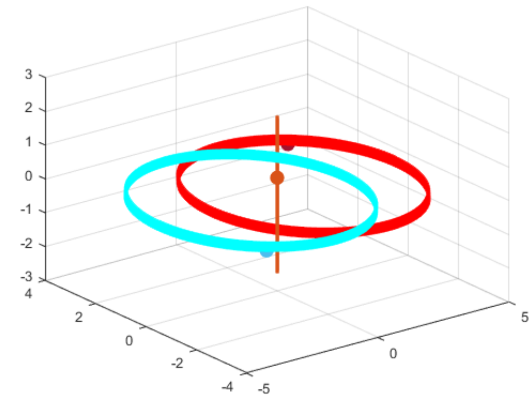
The same function can also be used to test the stability of a system. Fig. (1d) shows the Pythagorean three-body problem: three masses in the ratio 3:4:5 are placed at rest at the vertices of a 3:4:5 right triangle. This system would have an eventual escape. As shown in the figure, the orange body after experiencing a gravitational slingshot, escapes from the system, and leaves a stable binary system.

4 Future Work

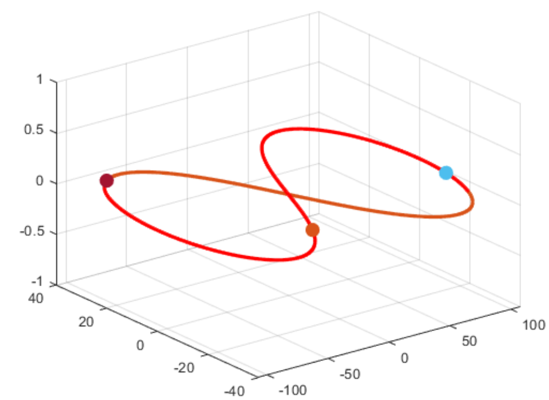
My function currently works with only three bodies, but the same idea can be applied to an n-body system. I am also conceiving about an algorithm to identify stable solutions, by recognizing the distance between two bodies are always within certain value, or the difference between the state (position and momenta vectors) of a system at certain time t_1 and its state at a different time t_2 is less than some tolerance values.



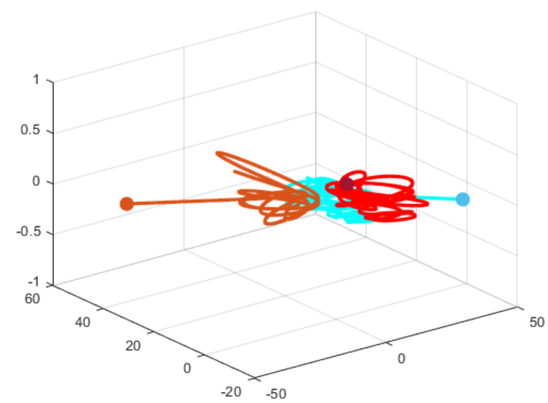
(a) A binary star system with a planet.



(b) The Sitnikov problem.



(c) The Figure-Eight solution.



(d) The Pythagorean three-body problem.

Figure 1: Simulations of three bodies.

Reference

[1] Casselman, Bill. "A new solution to the three body problem - and more.", American Mathematical Society http://www.ams.org/publicoutreach/feature-column/fcarc-orbits1

[2] "Sitnikov problem." Scholarpedia http://www.scholarpedia.org/article/Sitnikov_problem

[3] Makino, Jun. "5.8 Three Bodies on a Figure Eight." *The Art of Computational Science.* http://www.artcompsci.org/msa/web/vol_1/v1_web/node45.html

[4] Allain, Rhett. "This is the only way to solve the three-body problem."