PHYS1321 Computational Methods in Physics
December 9, 2016

Time Evolution of Our Solar System with Earth-Asteroid Collisions

Gerardo Ayes, Nick Hopkins
University of Pittsburgh

ABSTRACT

Aims. To simulate our solar system and introduce asteroids to monitor effect they have on the system.

Methods. Python code available via GitHub. Odeint was mainly used to generate array data.

Results. Various simulations with two test cases: Ceres sized asteroid and Moon sized asteroid.

1. Introduction

The problem that we tackled was creating a simulation of the solar system and simulating collisions of astronomical objects with the earth. In order to do this we needed to encode the physics involved. Attractive forces due to gravity are what generate the orbits of celestial bodies. Newton's law of gravitation states that the force of attraction between two objects is directly proportional to the product of their masses and the gravitational constant, while being inversely proportional to the square of their separation distance. This results in the equation: $F_{ji} = -\frac{GM_iM_j}{r^2}$ where M_i and M_j are the masses of the objects and r is the separation distance between the two. This equation is the foundation of what allows us to understand how objects affect one another due to gravity. The collisions and even the orbits of the planets were described using this equation.

It would have arguably been more fun to simulate asteroid collisions with Earth that sent it out to Jupiter's orbit or that ejected it out of our solar system than those presented below. However, in the spirit of being scientific the cases discussed were chosen while keeping in mind more plausible situations given our knowledge of the history of our own solar system.

2. Code Description

The complete repository is available on GitHub[1] with the structure being such that there are several files aptly named to reflect the step in our process for which they were created. Later we unified them to produce a final code that satisfied our initial objectives, described above. Among these is "entire-solar-system-simulation" which begins by defining the relevant constants in our solar system. The format chosen for the organization of this data was, since there are 9 main objects, for 0 to represent the Sun, 1 for Mercury, and continuing on until reaching 8 for Neptune. The constants needed were mass M_i, average orbital distance D_i, and average orbital velocity V_i. The gravitational constant and their respective orbital periods were also established in order to have appropriate time scales when testing the code.

Next was the "genOrbitalMotion" function, this is called by the function in the same cell, "orbitalMotion" with the built in odeint from the scipy module. The duty of the former is to tell odeint what to do during each time step and this was generalized to n objects based off of the initial conditions. When specifying yarr[2] and t[3] the orbital periods mentioned earlier proved most useful and N specified the resolution, how many iterations odeint would go through. From there plotting was through the matplotlib.pyplot module with a legend for ease of tracking each body.

Subsequently there was the animation function. Originally I had not attempted an animation before and thus supposed that

[1] https://github.com/nah91/Destabilization

[2] initial conditions going as: x, v_x, y, v_y

[3] time array with length N

either creating png files and saving generated images to them and subsequently displaying them in a gif or creating a figure and updating the data shown using the arrays generated by the odeint function would work. It turns out that the latter with an mp4 file as the result is the one I got to work first. Thanks to a code found online[4] it was a matter of specifying the os path which solved the issue.

Afterward, the final function I truly needed was not a new one, but rather a modification of the original genOrbitalMotion function to take into account collisions. The simplest method seemed to be to directly alter the function as opposed to try a separate collision function to be called later. The solution was to add an if statement after calculating the separation distance between bodies i and j and to update the position and velocity in both x and y of object j to that of i. Finally, test cases can be implemented varying the mass and radius of a ninth body, namely an asteroid, setting initial conditions such that it collides with Earth and affects its orbit significantly.

3. Analytic Solution

The major calculation throughout our project was that of the acceleration of body i due to the forces caused by the gravitational fields of the other bodies considered $\sum_{j\neq i}^{n} F_{ji}$ where n was the total number of bodies in the system. For example, when considering the solar system's effect on Earth the for loops through i and j in "genOrbitalMotion" calculate the acceleration $\sum_{j\neq 3}^{8} -\frac{GM_j}{r^2}$. Through each iteration of j, this value is added to the sumx and sumy terms to be assigned to the Ax and Ay arrays for assignment into ansarr, which will persist.

4. Test Cases

It may be useful to keep in mind that all the following simulations were run with initial positions at $\theta = \pi$ and thus the planets' positions line up along the customary -x axis and their velocities take them into counterclockwise orbits.

The first test case discussed was that of an asteroid with the mass of Ceres (~9.393e20 kg) with a velocity defined such that its kinetic energy is equal to the difference between the orbits of Mars and Earth: $KE = \frac{1}{2}mv^2 = PE_{Mars} - PE_{Earth}$ The results were slightly surprising as the expected result was an overlap between their orbits after the collision. However, Earth's orbit was affected much less than that. Fig. **??** depicts Earth's orbit with vs. without this asteroid collision. This case is interesting because Ceres is an actual object in our solar system, classified as a dwarf planet, and is an interesting thought experiment.

The second test case explored was an asteroid with the mass of our Moon and $KE_{asteroid} = \frac{1}{2}mv^2 = PE_{Earth}$. This is similar to what Astronomers theorize resulted in the Earth-Moon system we take for granted today. Out original hope was to simulate debris from each collision by randomizing the number of objects resulting (between preset values to limit computing time) and using conservation of linear momentum in x and y as constraint for the last object's parameters. However, the number of objects running through odeint in each timestep seems to be restricted by the initial conditions[5].

5. Full Physical System

The entire solar system was considered[6] in a simplified simulation. The Sun and eight planets[7] were placed in realistic orbits and made to all revolve in the same direction. This is simplified because we did not take into account the moons or confirm that our integration method is stable over a long period. Of course, the period of Jupiter's orbit is relatively small and this should not affect results.

6. Results

The initial conditions in Table 1 generated the plots shown in the Figures.

7. Discussion

Figures 3-5 are in regards to the Earth-Moon collision test case. In Figure 3 you it can be seen that the change in the x and y

4 http://josephrenaud.com/post-drop/2016/data-animation-with-python

5 Almost as if conservation of the number of objects is assumed.

6 The planets but not their moons, although could be trivially added in since "orbitalMotion" takes in n objects. Would need to adjust collision parameter established by 'Sep' variable and add masses into Mass array (same with Radius array).

7 Pluto is currently considered a dwarf planet, although it may yet have its revenge!

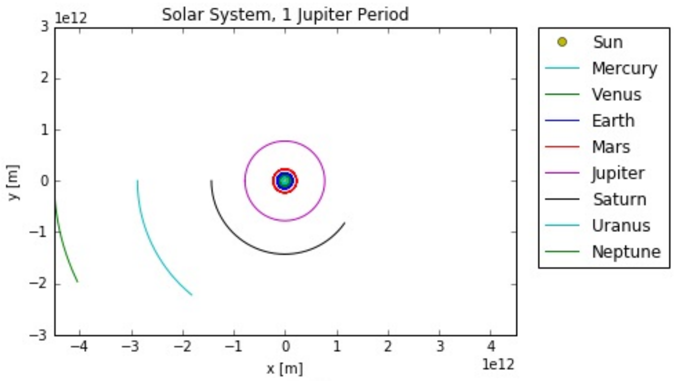

Fig. 1. Orbits of planets in our solar system.

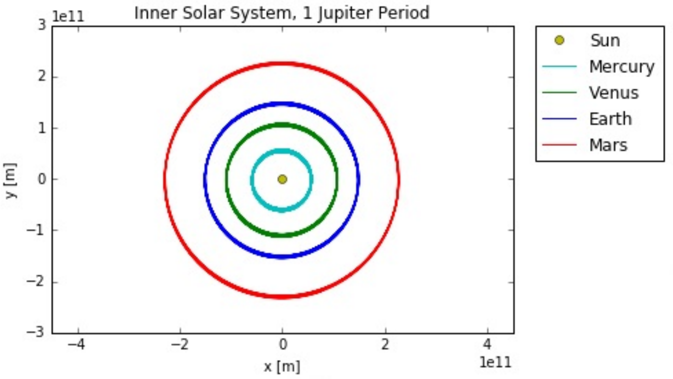

Fig. 2. Orbits of planets in our solar system with a better view of Earth.

positions of the Earth's orbit over time still have a similarity to one another, but vastly different from the Earth's original orbit. It was surprising to see how little the Earth-Ceres asteroid collision affected the Earth's orbit compared to the Earth-Moon collision. Since Ceres is considered a dwarf planet and the largest asteroid in the belt between Mars and Jupiter, it was first thought that a collision between it and the Earth would be more impactful on the Earth's orbit. Instead we can see in figures 4 and 5 that the relative position as well as the component velocities of the Earth's orbit barely shift after the collision. This is unlike the Earth-Moon collision where figures 6 and 7 show how the positions as well as the velocities of the components of the Earth's orbit are very askew after the collision.

8. Future Directions

Currently the solar system that was constructed lasts long enough for the collisions but we would like to make the orbits of the planets more stable over a longer period of time. We would like to include elastic collisions to the test pool as well as being able to simulate debris after a collision. Currently we run the test cases assuming the collisions are fully inelastic and have no debris post-collision. With debris, others could even do a moon formation simulation if desired. We managed to get a working animation of the solar system so we could eventually include an animation of a collision test case as well. If others were to continue this project, they could attempt to find more interesting initial conditions[8]. They could also include collision simulations of asteroids with the other planets of the solar system or even asteroid-asteroid collisions.

9. Acknowledgments

Thanks to Seth Roffé for his help generalizing the function "orbitalMotion" to n objects.

Special thanks to Dr. Michael Wood-Vasey for providing us with the experience and knowledge to complete this project as well as assisting with taking into account collisions.

10. Code

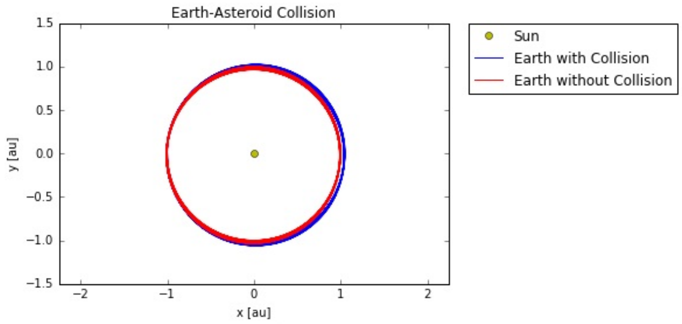

Fig. 3. Orbit of Earth with vs. without a collision, asteroid with mass of Ceres.

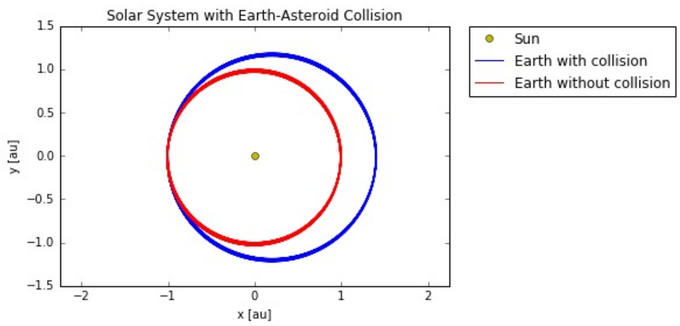

Fig. 4. Orbit of Earth with vs. without a collision, asteroid with mass of Moon.

[8] Perhaps the destabilization of Earth's orbit to the degree that it collides with Mars or Venus and causes a chain reaction of collisions.

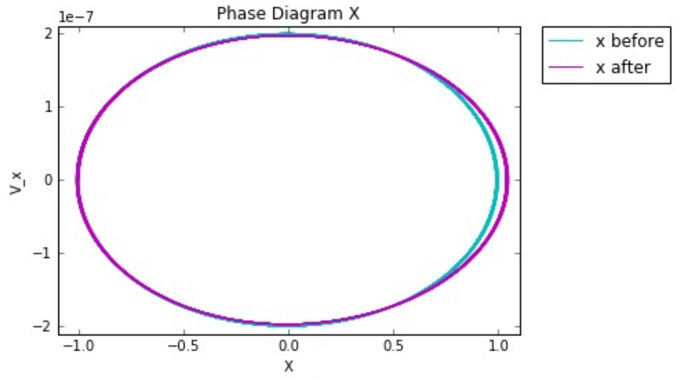

Fig. 5. $\frac{\dot{x}}{x}$ represents phase space. A closed loop indicates conservation of energy, and the elongation to the right is a reflection of the change in Earth's orbit due to Ceres size asteroid.

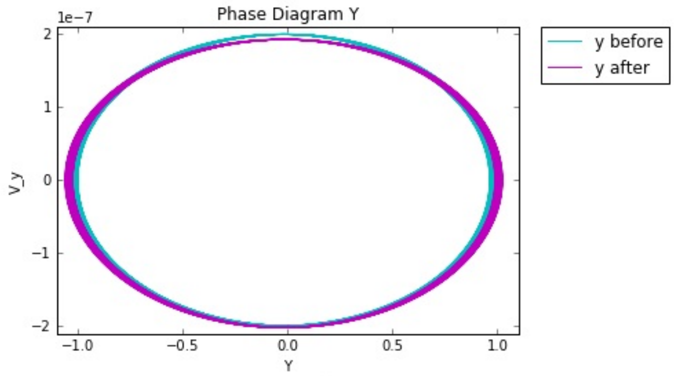

Fig. 6. $\frac{\dot{y}}{y}$ represents phase space. A closed loop indicates conservation of energy, and the elongation to the right is a reflection of the change in Earth's orbit due to Ceres size asteroid.

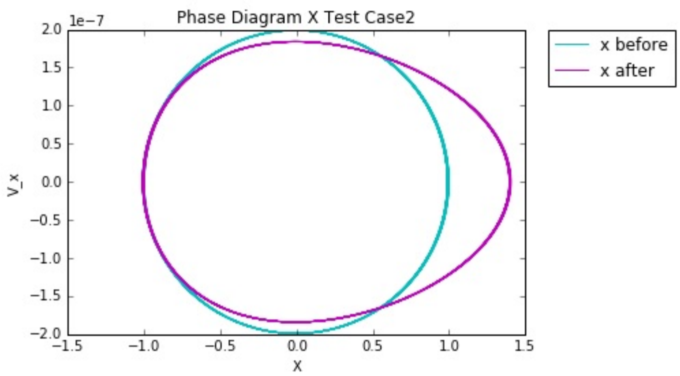

Fig. 7. $\frac{\dot{x}}{x}$ represents phase space. A closed loop indicates conservation of energy, and the elongation to the right is a reflection of the change in Earth's orbit due to Moon size asteroid.

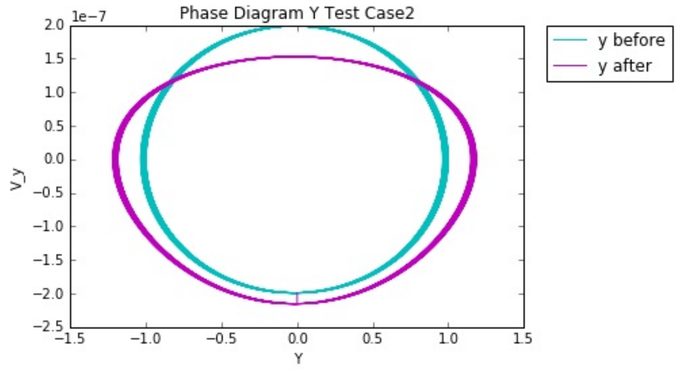

Fig. 8. $\frac{\dot{y}}{y}$ represents phase space. A closed loop indicates conservation of energy, and the elongation to the right is a reflection of the change in Earth's orbit due to Moon size asteroid.

Table 1. The conversion factor of meters to au is α = 6.6846E-12. Orbital velocity = $\sqrt{\frac{GM_0}{D_i}}$

Name of Celestial Body	Period(s)	Mass(kg)	Radius(au)	Orbital Distance to Sun (au)	Velocity (au/s)
Sun	NA	1.99E+30	(6.955E+8)α	NA	NA
Mercury	7.60E+06	3.30E+23	(2.439E+6)α	(5.79E10)α	(47882)α
Venus	1.94E+07	4.87E+24	(6.052E+6)α	(1.082E11)α	(35026)α
Earth	3.15E+07	5.97E+24	(6.3781E+6)α	(1.496E11)α	(29788)α
Mars	5.94E+07	6.42E+23	(3.397E+6)α	(2.279E11)α	(24134)α
Jupiter	3.74E+07	1.90E+27	(6.9911E+7)α	(7.786E11)α	(13057)α
Saturn	9.15E+08	5.69E+26	(6.0268E+7)α	(1.433E12)α	(9624)α
Uranus	2.65E+09	8.68E+25	(2.5559E+7)α	(2.873E12)α	(6797)α
Neptune	5.20E+09	1.03E+26	(2.4764E+7)α	(4.495E12)α	(5434)α
Ceres	NA	9.39E+20	(4.73E+5)α	(1.496E11)α	1.53E-05
Moon-Sized Asteroid	NA	7.34E+22	(1.7371E+6)α	(1.496E11)α	1.53E-06