Development of a Python Program Simulating Motions of Celestial Bodies

Jin Yan
Guangzhou Foreign Language School, Guangzhou, Guangdong, 511400, China
Email: lupo@edu.viax.org

Abstract. While the research of the celestial body is essential to understand the universe, it requires considerable costs in time and equipment. However, the computation methods can be used to analyze the astrophysics process in a simple approach. In this paper, a model of computing the motion of solar system based on Euler’s method is established. The model divided the process into infinite small time steps, and iteration is used to represent integration and update the position of planets over the process. The NumPy and matplotlib are used to visualize the motion. By the analysis of momentum and energy through the process and the application in the 2-body situation, the model is proved to be valid and accurate.

1. Introduction
Since the development of heliocentrism, astronomers are eager to track the paths of the planets revolving around the Sun. Before the stage of telescopes, people could only observe the pattern of stars through the naked eyes, thus people can only observe the stars near the Earth’s orbit. After that, the appearance of telescopes enables us to observe the clear vision of planets that are far away from Earth. Based on these observations, Newton’s law of universal gravitation and Kepler’s laws were developed to provide analytical solutions of planets’ orbits. However, neither analytical calculation nor direct observation is practical to get the motions of most planets in the universe. Observation through telescopes requires precise and expensive types of equipment and also long periods, and the complexity of the realistic system makes this problem have no analytical solutions. In the past few decades, with the improvements of computational powers, more and more physicists start to use computational tools to solve equations, do integrals, invert matrices, simulate physical processes of all kinds, etc. This paper presents a way to use a computational method to simulate the motion of the Sun and eight planets. [1] This method was first implemented for a simple Sun-Earth system and then extended to the full 8-planet system.

2. Methodology
Euler method is the key part of the simulation. It provides a way to use programming skills to calculate the motion of each planet using law of universal gravitation and Newton’s second law of motion.

2.1. Euler’s Method
Euler's Method is used to solve ordinary differential equations with initial values. Assuming the slope over an infinitely small interval of x will not change, the initial value can move a small step toward the direction of the end value using the value of the slope of that point. Then the new point can move toward the end value using the slope of the new point. After many steps, the end value will be
predicted. As long as the size of each step is small enough, the end value can be accurately estimated. Suppose we have a function \( f(x) \), and we know the value of \( f \) at a point \( x_0 \), and \( f'(x_0) \). If we want to find the value of \( f \) at a point close to \( x_0 \), say \( x_0 + h \), then we can approximate \( f(x_0 + h) \) by using the tangent line to the graph at the point \( (x_0, f(x_0)) \), as shown in Figure 1. The idea is that \( f'(x_0) \) estimates how much change there is in \( y \) for each change in \( x \); we know that \( x_0 \) increased by \( h \), so \( y \) should have increased by approximately \( hf'(x_0) \), so the new value should be \( f(x_0) + hf'(x_0) \).

![Figure 1. The demonstration of Euler’s method.](image)

2.2. Law of Universal Gravitation

The Law of Universal Gravitation states that everything in the universe, from planets to atoms, will attract other matters with a force that directly proportional to the mass of each matter and inverse proportional to the square of the distance between two matters, as shown in equation (1). The Gravity that eight planets and the Sun exerted on each other is the force that causes eight planets to revolve around the Sun. The Law of Universal Gravitation allows us to calculate the gravity between every pair of the planet at any time, given the mass of each planet, the distance between them, and the universal constant \( G \).

\[
F = G \frac{m_1 m_2}{r^2}
\]  

(1)

2.3. Newton’s Second Law of Motion

Newton’s Second Law of Motion connects the force exerted on one object with the motion of the object. It states that the net force experienced by the object is equal to the product of the mass of that object and the acceleration of that object, as shown in equation (2). So, after calculating the net force of one planet or the Sun through the Law of Gravitation, this law will provide the acceleration of that planet or the Sun.

\[
F = ma
\]  

(2)

In the simulation of eight planets and the Sun’s movements, no analytical solutions exist if each planet-to-planet gravitational force is taken into account explicitly. However, Euler’s method can be applied to this scenario to predict the motions numerically. The acceleration that can be computed according to Newton’s Second Law of Motion is the derivative of velocity. Given the initial velocity,
we can estimate the velocity of the planet at any time, as long as the time step is small enough. This process is illustrated in Figure 2.

Figure 2. The example to use Euler’s method to predict velocity.

Similarly, velocity is the derivative of displacement. Given the initial position, we can estimate the position of the planet at any time, which is illustrated in Figure 3. In the program, iteration is used to update the velocity and the position of each planet.

In the simulation program, updates in each time step are shown as:

For each time update:

\[
\text{Velocity} = \text{Velocity} + \text{Acceleration} \times dt \\
\text{Position} = \text{Position} + \text{Velocity} \times dt
\]

where \(dt\) is a time step

Figure 3. The example to use Euler’s method to predict position.
2.4. Memorization
CPU time and memory storage can be optimized. Originally, if n stars are involved in the process, the program has to do \( n^2 \) times of computation for each increase in time, since each planet exerts a force on every other planet. But according to Newton’s Third Law, the force exerted on planet A by planet B and the force exerted on planet B by planet A are a pair of interaction forces, which means that they have the same magnitude but the opposite direction. So, once the force on planet A exerted on planet B is computed, we can use the name of planets as key and the force with the opposite direction as value to store the key-value pair into the dictionary. When it’s the time to compute force exerted on planet B by planet A, we can just extract the value of the corresponding key, rather than do the calculation one more time. Thus, the computational times of n stars for an increase in time will be reduced to \( \frac{n(n-1)}{2} \). In this case, because the value of n is too small, the reduction in time may not be obvious, but it can be helpful when more stars are involved in the simulation of further researches.

3. Analysis

3.1. Simulation of 2-body model
The program was verified using a simple 2-body model because analytical solutions exist for this model. Numerical results computed by the program were compared with the analytical solution to examine the correctness of the program. The model will be correct as long as the paths computed by the analytical solution is consistent with the paths computed by this model.

The two bodies in this case have equal mass but opposite directions of speed rotate under the effect of gravitational force. To maintain the uniform circular motion, the centripetal force must be provided by gravitational force. And they will rotate with a velocity that is the same in magnitude and opposite in direction. According to the analytical solution calculated from circular motion and law of universal gravity, velocity=577.5m/s, mass=6*2024kg, the two bodies will do circular motions with identical orbits, whose radius are r=3*10^8m. As a result, if the motions computed by the program match the analytical solutions, it is reasonable to conclude that the method is correct and the program is properly implemented. See table 1 below for the parameters.

| Parameter                          | Value                      |
|------------------------------------|----------------------------|
| Velocity of a planet (m/s)         | 577.5                      |
| Mass of a planet (kg)              | 6*10^{24}                  |
| Radius of the path (m)             | 3*10^8                     |
| Time of circular motion (s)        | 1.64*10^6                  |
| Time step (s)                      | 500                        |
Figure 4. The 3-D simulation of the motion of the typical case of 2-body system.

Figure 5. The velocity-time graph of the typical case of 2-body.

In this case, each of the two planets is represented by the blue and red lines separately. Half of the period of the rotation is set as computation time. According to Figure 4, the paths of two planets are the semi-circle of the same orbit; Because the velocities of the two planets are the same in magnitude but opposite in direction and the masses of the two planets are identical, the total velocity will always be zero, which is consistent with the Figure 5. The application in this case further proves the validity of the model.

In addition to the typical case of 2-body with equal mass, the program is also tested by the case of the Earth-Moon system, which has m1>m2. The scientific data shows that the Moon will maintain circular motion around the Earth in the universe. This model can be applied in this case: if the Moon completes one revolution around the Earth over the period of Moon, which is 2.36*10⁶ s, in the actual situation, then the result computed by the model will be consistent with the real world situation, which can prove the validity of the model. See table 2 below for the parameters.
Table 2. The parameters of the simulation of the Earth-Moon system.

| Parameter                        | Value                      |
|----------------------------------|----------------------------|
| Initial velocity of the Moon (m/s) | 1.023*10^3                |
| Initial velocity of the Earth (m/s) | 0.0                       |
| Mass of the Moon (kg)            | 7.3*10^{23}               |
| Mass of the Earth (kg)           | 5.97*10^{24}              |
| Radius of the Path (m)           | 3.844*10^{8}              |
| Time of Circular Motion (s)      | 2.36*10^6                 |
| Time step (s)                    | 100                       |

Figure 6. The 3-D simulation of the motion of the Earth-Moon system.

In this case, the path of the Earth is represented by a blue line while the path of the Moon is represented by a grey line. According to Figure 6, the Moon maintains circular motion around the Earth over the period of Moon in the real situation. Because the Earth is only affected by the Moon in this case, the path of the Moon will be curved under the gravitation between the Earth and the Moon, as shown in Figure 6. So, the paths computed by the model are consistent with the paths in the actual situation.

3.2. Using The Law of Conservation of Momentum and The Law of Conservation of Energy to prove the accuracy of the simulation.

The revolution of eight planets and the motion of the Sun depend on the gravitational force they exerted on each other, and this paper neglects the effect of other planets and matters in the universe. Given the authentic solar system’s motions, we know that the orbits of all of the 8 planets should be approximately circular. For more general multi-body systems, although no analytical solutions for paths of the planets are available, conservation of momentum and energy of the whole system should still be obeyed. According to the Law of Conservation of Momentum, the total momentum of an isolated system is constant. So the momentum of the system should remain constant over iterations. Similarly, because ignoring the effect of other planets, the net work done by the other celestial objects
on the systems that contains eight planets and the Sun is zero. And the gravitational force inside the system is a conservative force, so the mechanical energy of the system is conservative. The analysis of the E-t graph and P-t graph can be used to show the accuracy of the model.

![Figure 7. The momentum-time graph of the solar system.](image)

![Figure 8. The energy-time graph of the solar system.](image)

While the location of each planet is updated at each time interval, the mechanical energy and the momentum of the system at each moment are computed and stored as well, which means the E-t graph and P-t graph can be easily drawn by python without many extensions of codes. According to Figure 7 and Figure 8, both the relationship between energy and time and the relationship between momentum and time are roughly linear with a slope of zero. (The reason for the little fluctuation of momentum and energy is the bias of the differential element method when dt is not small enough to decrease the computation time.) This shows that the energy and momentum do not change during the process and are consistent with The Law of Conservation of Energy and The Law of Conservation of Momentum, which proves the validity of the model.
Figure 9. The 3-D simulation of the motion of the solar system.

Figure 9 is the motion of the whole solar system simulated by the program. Parameters of bodies are obtained from Ref.[5]. As shown in the figure, the orbit of each of the planets is circular and centered at the Sun. The path of each body is represented by a curve with a different color. According to the analysis above, the paths of bodies shown in the figures above are valid.

4. Summary

4.1. Conclusion
In this paper, the model that used the finite element method is established. Based on this model, the simulation of the planetary movement of eight planets and the Sun is done using Python. To test the correctness of the model, it is first applied to the typical case of the 2-body movement. The result given by the model is consistent with the paths computed by the analytical solution, which shows that the differential element method can simulate the complex planetary movement of the solar system. Then momentum and the energy of the system are computed to check whether the model obeys the Law of Conservation of Momentum and the Law of Conservation of Energy.

4.2. Future Work

4.2.1. Simplicity. In this paper, we merely consider the effect of the eight planets and the Sun. Even though it makes the simulation process more simple, eight planets and the Sun are also influenced by the billions of other planets in the universe. It will be difficult to consider all the influence of other matters. So, there is a bias between the result of this paper and the real-world situation.

4.2.2. Complexity and Octree Method. The process of differential element method is simple to apply, but it will compute the effect of each planet of every other planet, which has the complexity of $O(N^2)$. If the number of planets increases, the computation time will increase in square extent. There are millions of planets in the universe, this method will be wasteful of time when it is used to solve problems that involve more bodies.

To eliminate the complexity, a more effective method, called Octree Method [3], is proposed. The Octree Method will create a cube that involves all of the planets. If the cube contains more than one planet, the large cube will be divided into eight smaller cubes. Repeat this process until all of the leaf nodes contain equal or less than one node. Then, the depth of the tree will be $\log_8 N$. Each layer will
contain $8^n$ of nodes. The construction of each node will require $8^{1-n}N$ of computation. As a result, each layer will require $8^n \times 8^{1-n}N$ of computation. The construction of the Octree Tree will require $O(8^N \log_8 N)$ of computation.

In the Octree Method, only the planets that are close enough with another planet need to be computed individually. The cluster of planets that are far enough will be viewed as a large planet in the computation. The way to identify whether a cluster of planets is far enough to be viewed as a whole is $\lambda$, which has an equation $\lambda = \frac{L}{R}$ (L is the size of node and R is the distance from this node to the planet). Starting from the first layer, if the node has $\lambda < \lambda_0$, then this node can be viewed as a big planet for simplification. Otherwise, the child nodes will be visited to check if $\lambda < \lambda_0$. In this way, the complexity of the program will be reduced to $O(N \log_8 N)$.

References
[1] J Reddy 2019 *Introduction to the Finite Element Method 4th Edition* Mc Graw Hill Education, College Station Texas
[2] M Newman 2013 *Computational Physics*
[3] H Freeman and D Meagher 1985 *Computer Architectures for Spatially Distributed Data* vol 18
[4] Samir Khan Christopher Williams Jimin Khim and 1 other contributed 2021 https://brilliant.org/wiki/eulers-method/
[5] Dr. David R. Williams 2019 https://nssdc.gsfc.nasa.gov/planetary/factsheet/