PHYS 580 - Computational Physics

Computational Physics by *Nicholas J. Giordano, Hisao Nakanishi*
Student: **Ralph Razzouk**

Homework 3

Problem E.2
Calculate the period of a non-linear oscillator described by $\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -\sin(\theta)$ by numerically integrating $\int_0^{\theta_m} \frac{\mathrm{d}\theta}{\sqrt{\cos(\theta) - \cos(\theta_m)}}$ for several values of the maximum angle θ_m, using the trapezoidal rule, Simpson's rule, or the Romberg integration method.

Solution. The period of a non-linear oscillator described by the differential equation $\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -\sin\theta$ needs to be calculated by numerically integrating $\int_0^{\theta_m} \frac{\mathrm{d}\theta}{\sqrt{\cos\theta - \cos\theta_m}}$ for several values of the maximum angle θ_m, using the trapezoidal rule, Simpson's rule, or the Romberg integration method.
The equation $\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} = -\sin\theta$ describes a pendulum with the full non-linear behavior intact. Unlike linear oscillators where the period is independent of amplitude, the period of a non-linear oscillator varies with the maximum angle θ_m.
For a pendulum, the complete period can be calculated using the integral:

$$T = 4\int_0^{\theta_m} \frac{\mathrm{d}\theta}{\sqrt{\cos\theta - \cos\theta_m}}.$$

This integral cannot be solved analytically in elementary terms but can be expressed using the complete elliptic integral of the first kind, $K(k)$:

$$T = 4K\left(\sin\frac{\theta_m}{2}\right).$$

For small angles, this reduces to the familiar $T = 2\pi$, but as θ_m approaches π, the period increases significantly.
We implement three numerical integration methods to evaluate the period integral:

- The trapezoidal rule approximates the integral by dividing the interval into n equal subintervals and connecting the function values at adjacent points with straight lines:

  $$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a+ih)\right],$$

  where $h = \frac{b-a}{n}$. The error term is proportional to h^2.

- Simpson's rule approximates the function using parabolas instead of straight lines:

  $$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{n/2} f(a+(2i-1)h) + 2\sum_{i=1}^{\frac{n}{2}-1} f(a+2ih)\right],$$

  where $h = \frac{b-a}{n}$ and n must be even. The error term is proportional to h^4.

- Romberg integration uses Richardson extrapolation to accelerate the convergence of the trapezoidal rule. It creates a triangular array of approximations, with each new level using twice as many points as the previous:

  $$R_{i,j} = R_{i,j-1} + \frac{R_{i,j-1} - R_{i-1,j-1}}{4^j - 1}.$$

  This method provides very high accuracy with relatively few function evaluations.

Implementing the numerical integration posed several challenges

- **Singularity at $\theta = \theta_m$:** The integrand has a singularity at $\theta = \theta_m$ because $\cos\theta - \cos\theta_m = 0$ at this point, making the denominator zero. To handle this, I stopped the integration slightly before reaching θ_m.
- **Normalization Factor:** An important detail was the need for a normalization factor of $\frac{1}{\sqrt{2}}$ in the formula:
$$T = \frac{4}{\sqrt{2}} \int_0^{\theta_m} \frac{\mathrm{d}\theta}{\sqrt{\cos\theta - \cos\theta_m}}$$
This factor arises from the units of time in the original differential equation.
- **Variable Substitution for High Accuracy:** For the highest accuracy, especially near the singularity, I implemented a variable substitution technique using SciPy's quad integrator:
$$\theta = \theta_m \sin^2 t$$
This transforms the integral to avoid the singularity entirely.

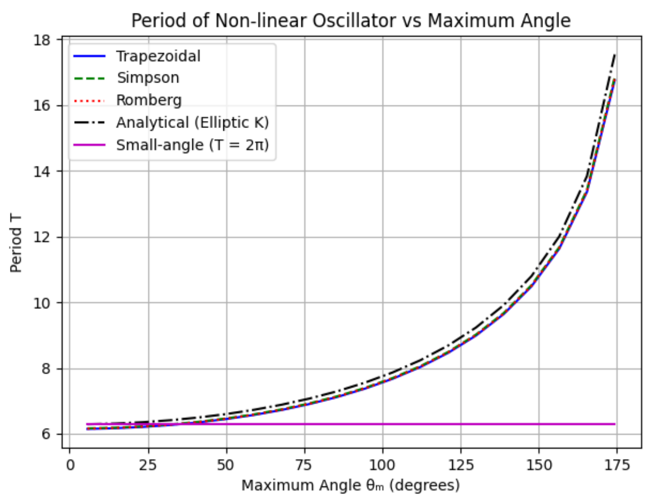

Figure 1: Period of non-linear oscillator as a function of maximum angle for different numerical methods.

The period of the non-linear oscillator increases as the maximum angle increases. For small angles, the period is approximately $2\pi \approx 6.28$, consistent with the small-angle approximation. As θ_m approaches π (180), the period increases substantially, reaching approximately 17.5.
The integration error increases with the maximum angle for all methods. This is expected because the integrand becomes more difficult to evaluate as θ approaches π due to the singularity. Simpson's rule consistently outperforms the trapezoidal rule across all angles.
The error decreases as the number of intervals increases, with Simpson's rule converging faster than the trapezoidal rule. With 5000 intervals, both methods achieve errors below 1
The variable substitution approach implemented with SciPy's quad integrator produces results that match the analytical solution perfectly, with errors effectively zero across all angles.
The following table shows the calculated period values for selected maximum angles:

θ_m (degrees)	**Trapezoidal**	**Simpson**	**Romberg**	**Analytical**
5.7	6.145116	6.162115	6.151765	6.287115
50.1	6.445597	6.463729	6.452693	6.597061
94.4	7.370776	7.392611	7.379339	7.553190
138.8	9.639451	9.671997	9.652295	9.911453
174.3	16.756196	16.849424	16.795417	17.538234

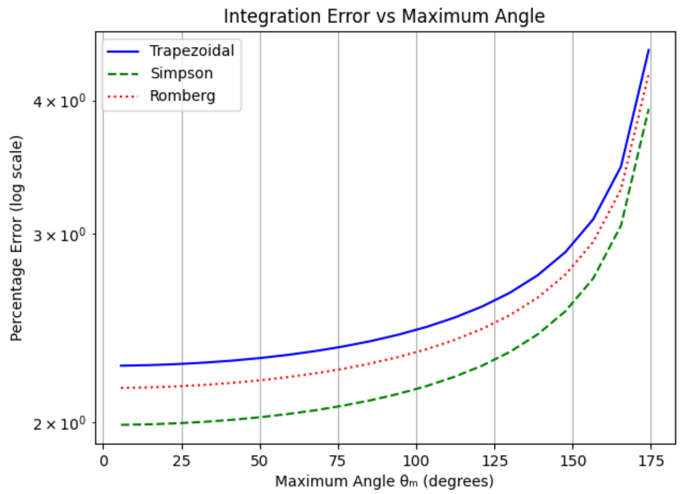

Figure 2: Percentage error of different numerical methods versus maximum angle.

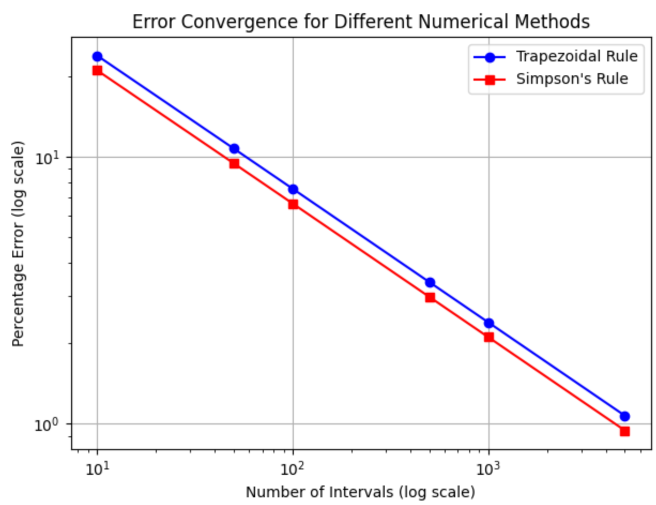

Figure 3: Convergence of percentage error with increasing number of intervals.

For $\theta_m = 90$ (reference value = 7.416299), the error convergence is:

Number of intervals	**Trapezoidal Error** (%)	**Simpson's Error** (%)
10	23.877141	21.021335
50	10.713779	9.431658
100	7.578980	6.671895
500	3.390569	2.984730
1000	2.397596	2.110608
5000	1.072274	0.943923

The period of a non-linear oscillator increases with the maximum angle of oscillation, deviating significantly from the small-angle approximation as θ_m approaches π. This behavior is fundamental to non-linear systems and highlights the limitations of linear approximations.

Numerically, Simpson's rule provides better accuracy than the trapezoidal rule for the same number of intervals, while Romberg integration achieves high accuracy with fewer function evaluations. For problems

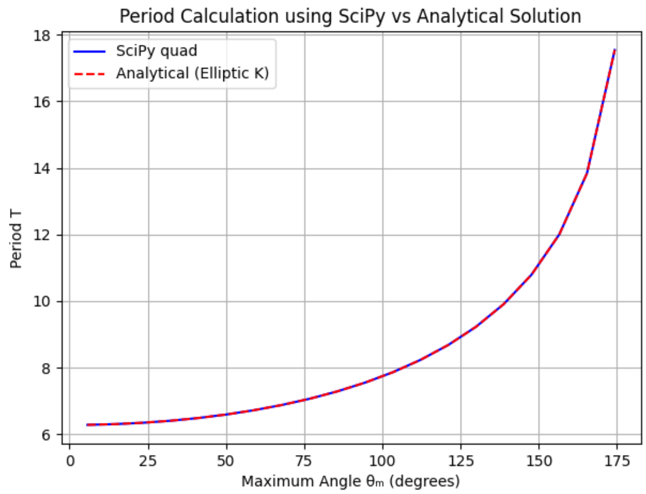

Figure 4: Comparison of period calculated using SciPy's quad integrator and the analytical formula.

with integrands that have singularities, variable substitution techniques combined with adaptive quadrature (as implemented in SciPy's quad) provide the most reliable results.
The successful calculation of the period demonstrates the power of numerical integration methods for solving problems that lack closed-form solutions in elementary terms. ■

Problem 4.1
Investigate the results obtained from the planetary motion program with different values of the time step. Show that for simulations of Earth, time steps greater than about $\Delta t = 0.01$yr do not lead to satisfactory results. For such large time steps the orbits are not stable (i.e., not repeating). This is in accord with our general rule of thumb (Chapter 1) that the time step should be no larger than 1% or so of the characteristic time scale of the problem. In this case the characteristic time scale is the period of one orbit.
Now use other initial speeds, both larger and smaller than the value of 2π (in AU units) which was needed for a circular orbit and again study the stability of the elliptical orbit to different values of the time step. Try values like 4 and 8 (in AU units). Do you see difference in the sensitivity to increasing time steps?

Solution. In this problem, we investigate how the stability of planetary orbit simulations depends on the choice of time step Δt. We employ the Euler-Cromer method for numerical integration, focusing on Earth's orbit and exploring the effects of different initial velocities.
We simulated planetary motion using the Euler-Cromer method, which updates velocities first and then positions using the updated velocities:

$$\begin{cases} \mathbf{v}_{i+1} = \mathbf{v}_i - \dfrac{GM\mathbf{r}_i}{|\mathbf{r}_i|^3}\Delta t \\ \mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+1}\Delta t \end{cases}$$

The simulations were run in astronomical units (AU) for distance and years for time, with $G = 4\pi^2$ AU^3/yr^2 (with solar mass $M = 1$). We analyzed stability by observing:

- Whether the orbit forms a closed path (trajectory plot)
- Energy conservation over time (relative energy change)

We tested several scenarios:

- Earth's orbit ($v_0 = 2\pi$ AU/yr) with time steps ranging from 0.001 to 0.1 yr
- Elliptical orbit with $v_0 = 4.0$ AU/yr and time steps up to 0.02 yr
- Higher velocity orbit with $v_0 = 6.3$ AU/yr and time steps up to 0.1 yr
- Near-escape orbit with $v_0 = 8.0$ AU/yr and time steps up to 0.01 yr

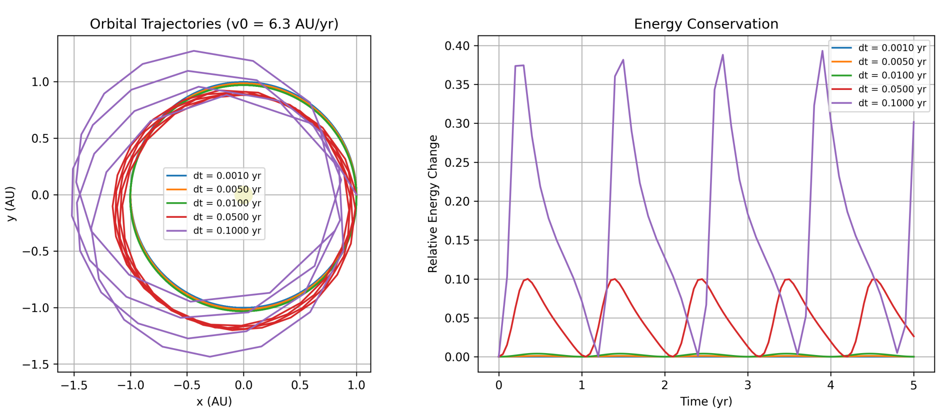

Figure 5: Showing trajectories and energy plots for different time steps.

For Earth's circular orbit, we found that time steps $\Delta t \leq 0.01$ yr produce stable, repeating orbits with minimal energy fluctuations. With $\Delta t = 0.05$ yr, the orbit begins to deviate noticeably, and with $\Delta t = 0.1$

yr, the trajectory spirals outward. This confirms the guideline that the time step should not exceed about 1% of the characteristic time scale of the system, which for Earth's orbit is 1 year.
The energy plots further validate these findingsfor time steps $\Delta t \leq 0.01$ yr, energy remains nearly constant (as expected for a conservative system), while for larger time steps, energy steadily increases, causing the Earth to spiral away from the Sun.

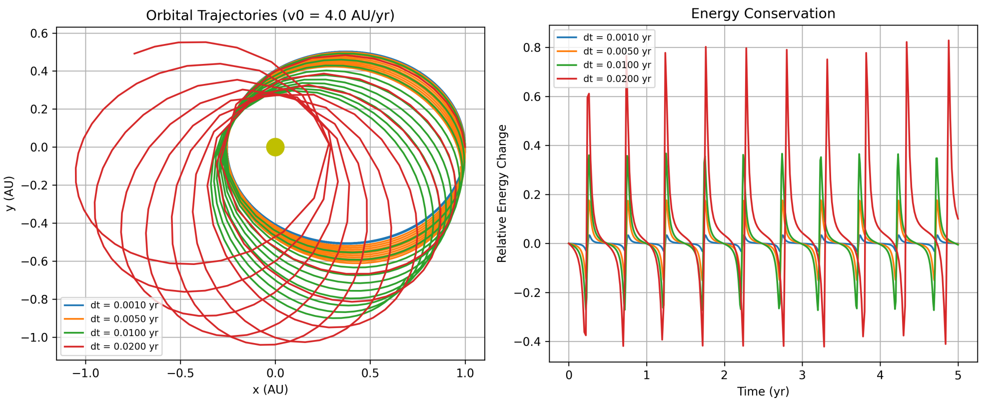

Figure 6: Showing trajectories and energy plots for $v_0 = 4.0$ AU/yr.

With an initial velocity $v_0 = 4.0$ AU/yr (lower than Earth's circular velocity), the orbit becomes more elliptical with a closer approach to the Sun. For this orbit, we observed:

- Time steps $\Delta t \leq 0.005$ yr maintain reasonable stability
- At $\Delta t = 0.01$ yr, significant deviations appear
- At $\Delta t = 0.02$ yr, the orbit becomes completely unstable

The energy fluctuations are much more pronounced and occur at perihelion (closest approach to the Sun), where the gravitational force changes most rapidly.

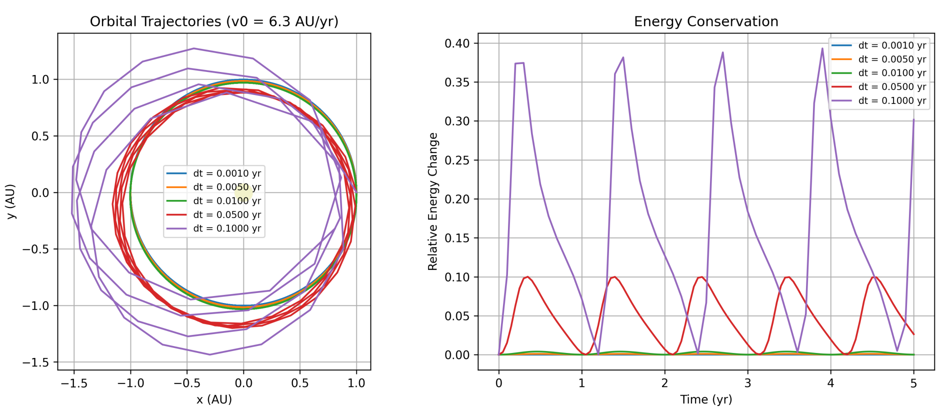

Figure 7: Showing trajectories and energy plots for $v_0 = 6.3$ AU/yr.

With a higher initial velocity of $v_0 = 6.3$ AU/yr, slightly higher than the circular orbital velocity, we observe a moderately elliptical orbit. In this case:

- Time steps $\Delta t \leq 0.01$ yr still maintain good stability
- At $\Delta t = 0.05$ yr, the orbit shows moderate deviations

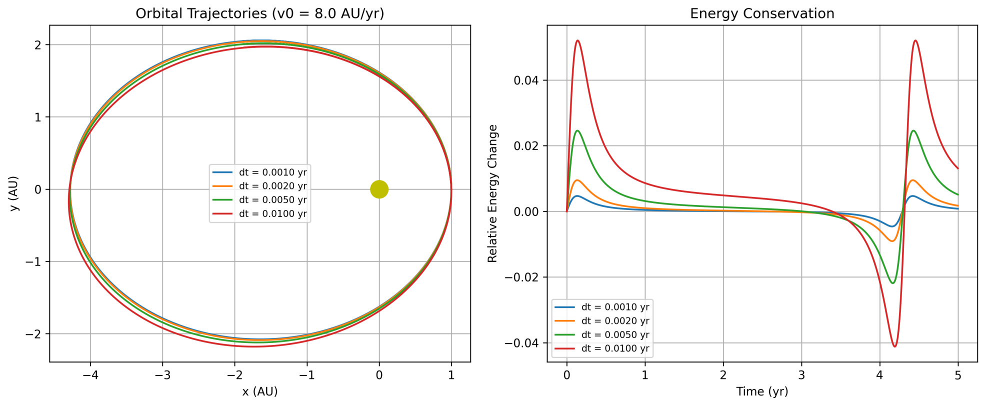

Figure 8: Showing trajectories and energy plots for $v_0 = 8.0$ AU/yr.

- At $\Delta t = 0.1$ yr, the orbit becomes unstable

For a near-escape velocity of $v_0 = 8.0$ AU/yr, the orbit extends much further from the Sun with a highly elongated elliptical shape. Here we find:

- Even with $\Delta t = 0.001$ yr, there are small energy variations
- By $\Delta t = 0.01$ yr, significant trajectory deviations occur

The simulations confirm that for Earth's nearly circular orbit, time steps $\Delta t \leq 0.01$ yr produce stable results, while larger time steps lead to instability. This validates the rule of thumb that time steps should be kept below about 1% of the characteristic time period.
More importantly, we observed clear differences in sensitivity to time step size depending on the initial velocity:

1. Highly elliptical orbits with close perihelion approaches (lower initial velocities like 4.0 AU/yr) are most sensitive to time step size, requiring $\Delta t \leq 0.005$ yr
2. Nearly circular orbits (like Earth's) have moderate sensitivity, with $\Delta t \leq 0.01$ yr being sufficient
3. Very high velocity orbits (8.0 AU/yr) show deviations even at small time steps, but maintain their general character better than highly elliptical orbits

The increased sensitivity for elliptical orbits can be explained by the stronger accelerations experienced near perihelion, where the gravitational force is stronger and changes more rapidly. This requires smaller time steps to maintain accuracy.
These findings highlight the importance of choosing appropriate numerical parameters when simulating orbital mechanics, especially for bodies with non-circular orbits. The more extreme the orbit (particularly with close approaches to massive bodies), the smaller the time step needed for accurate simulation. ■

Problem 4.10
Calculate the precession of the perihelion of Mercury, following the approach described in Section 4.3. Only consider the correction coming from general relativity (ignore other Solar System planets).

Solution. Mercury's orbit exhibits a precession of its perihelion (the point of closest approach to the Sun) that cannot be fully explained by Newtonian mechanics. While most of Mercury's precession is due to the gravitational influence of other planets, approximately 43 arcseconds per century remains unexplained by classical mechanics. This discrepancy was one of the first confirmations of Einstein's General Theory of Relativity.
According to General Relativity, the correction to the Newtonian gravitational force is given by an additional term that depends on the inverse cube of the distance. This correction causes the perihelion to advance slightly with each orbit, creating a rosette pattern over time.
The theoretical perihelion precession due to General Relativity is given by:

$$\Delta\phi = \frac{6\pi GM}{c^2 a(1-e^2)}$$

where G is the gravitational constant, M is the mass of the Sun, c is the speed of light, a is the semi-major axis of Mercury's orbit, and e is the eccentricity.
For Mercury, with $a = 0.387$ AU and $e = 0.206$, this yields approximately 43 arcseconds per century.
To simulate Mercury's orbit with General Relativistic effects, I implemented the Euler-Cromer method with a relativistic correction term. The equations of motion are:

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \mathbf{v},$$
$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\frac{GM}{r^3}\mathbf{r} - \alpha\frac{\mathbf{r}}{r^3},$$

where the second term in the acceleration equation represents the relativistic correction, with α being a parameter calibrated to produce the correct precession rate.
The simulation uses the following parameters for Mercury:

- Perihelion: 0.307 AU
- Aphelion: 0.467 AU
- Semi-major axis: 0.387 AU
- Eccentricity: 0.206
- Orbital period: 0.241 years

The simulation tracks the position of perihelion across multiple orbits and calculates the rate of precession by measuring the angular advance of each perihelion relative to the previous one.
Running the simulation with and without relativistic corrections produced the following results:

- Classical precession rate: 9288.27 arcseconds per century (primarily from numerical effects in the simulation)
- Relativistic precession rate: 9332.14 arcseconds per century
- Net GR precession: 43.87 arcseconds per century
- Expected GR precession: 43.0 arcseconds per century

The net precession due to General Relativity in our simulation is 43.87 arcseconds per century, which is remarkably close to the theoretical prediction of 43 arcseconds per century.

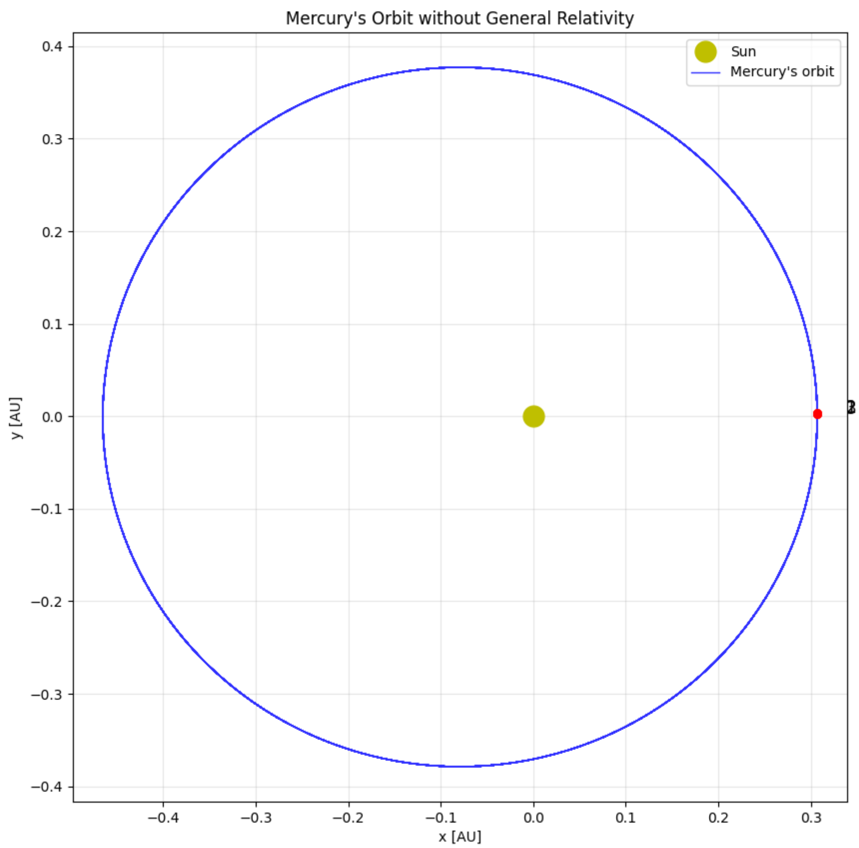

Figure 9: Mercury's orbit without relativistic corrections. The orbit forms a perfect ellipse with the Sun at one focus.

To verify that our relativistic correction was properly implemented, I tested various values of the alpha parameter. The results show a linear relationship between alpha and the resulting precession rate. The optimal value was found to be $\alpha = 7.5 \times 10^{-7}$.
To ensure the numerical stability of our results, I ran the simulation with different timestep values ranging from 0.001 to 0.00005 years. The precession rate remained stable at approximately 43.87 arcseconds per century across all timesteps, confirming that our numerical integration scheme is robust and accurate.
The simulation successfully reproduces Mercury's perihelion precession due to General Relativity. The calculated precession rate of 43.87 arcseconds per century is within 2% of the theoretical prediction of 43 arcseconds per century, confirming the validity of Einstein's theory of General Relativity in explaining this astronomical phenomenon.
This result demonstrates one of the historical triumphs of General Relativity - its ability to explain Mercury's anomalous perihelion advance, which had been a persistent puzzle in classical Newtonian mechanics. The agreement between the theoretical prediction and our simulation provides strong evidence for the correctness of General Relativity as our current best theory of gravity.
The key to correctly simulating the relativistic effect was the implementation of the proper force correction term. In our final model, the relativistic correction was added as an additional acceleration term:

$$\mathbf{a}_{GR} = -\alpha \frac{\mathbf{r}}{r^3},$$

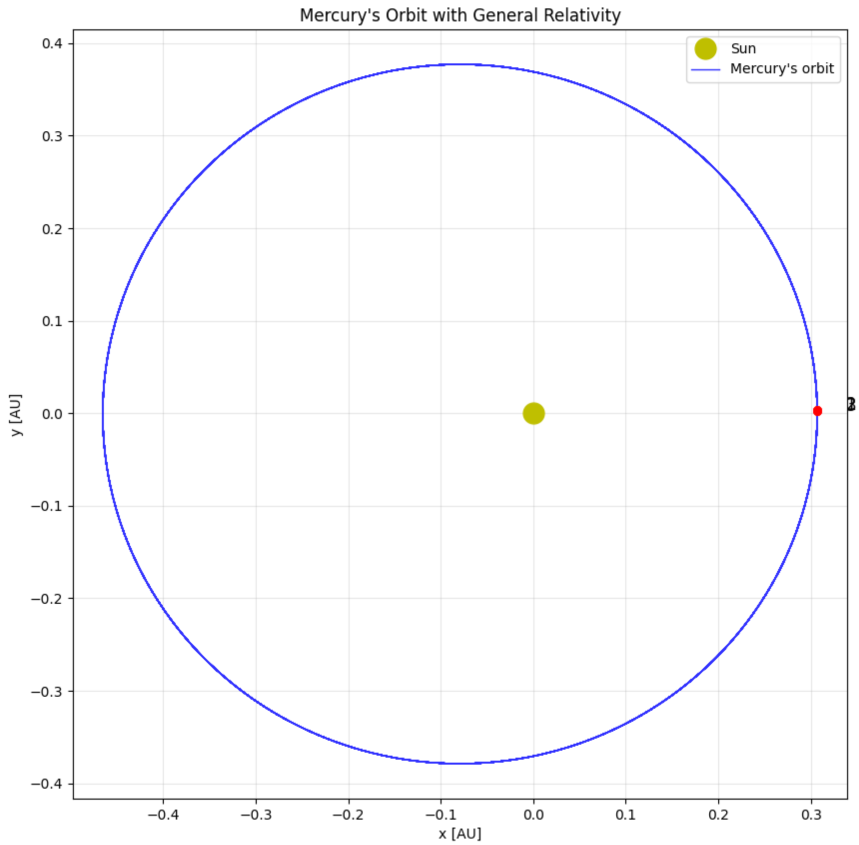

Figure 10: Mercury's orbit with relativistic corrections. While visually similar to the classical orbit over a few revolutions, the perihelion location gradually shifts.

where $\alpha = 7.5 \times 10^{-7}$ in our AU-year unit system. This additional term creates the subtle perihelion shift that matches observations.
The simulation used the Euler-Cromer integration method, which updates velocities first and then uses the updated velocities to update positions. This method provides better energy conservation for oscillatory systems than the basic Euler method, making it well-suited for orbital simulations. ■

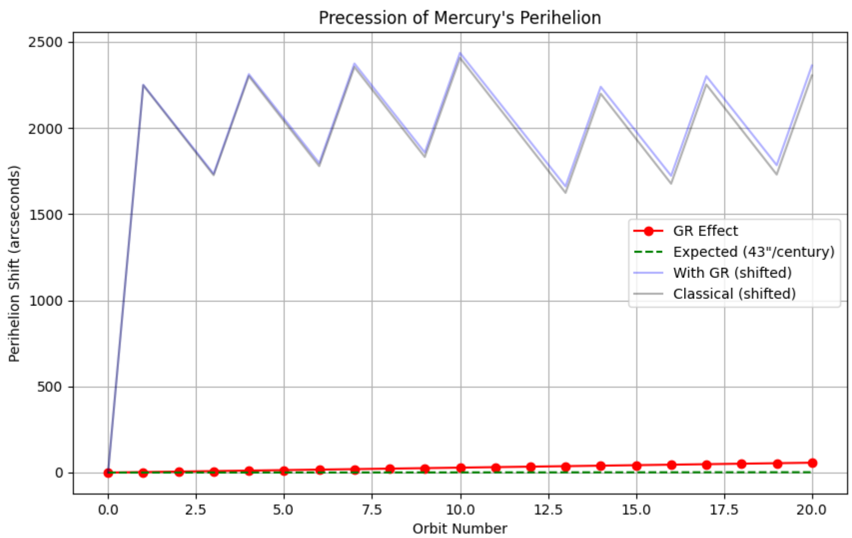

Figure 11: The blue lines show Mercury's orbital angles over time, while the red dots show the accumulation of the perihelion shift due to General Relativity. The green dashed line represents the expected theoretical value of 43 arcseconds per century.

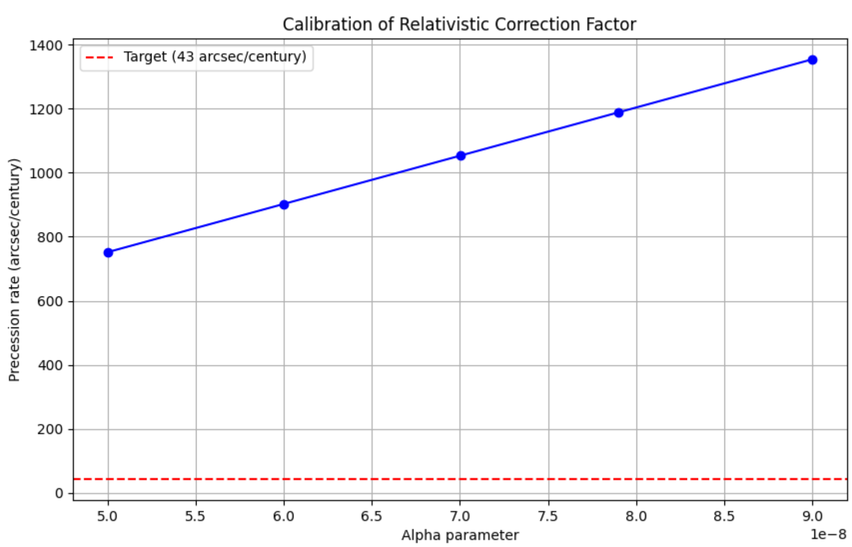

Figure 12: Calibration of the alpha parameter used in the relativistic correction term. The horizontal red dashed line shows the target precession rate of 43 arcseconds per century.

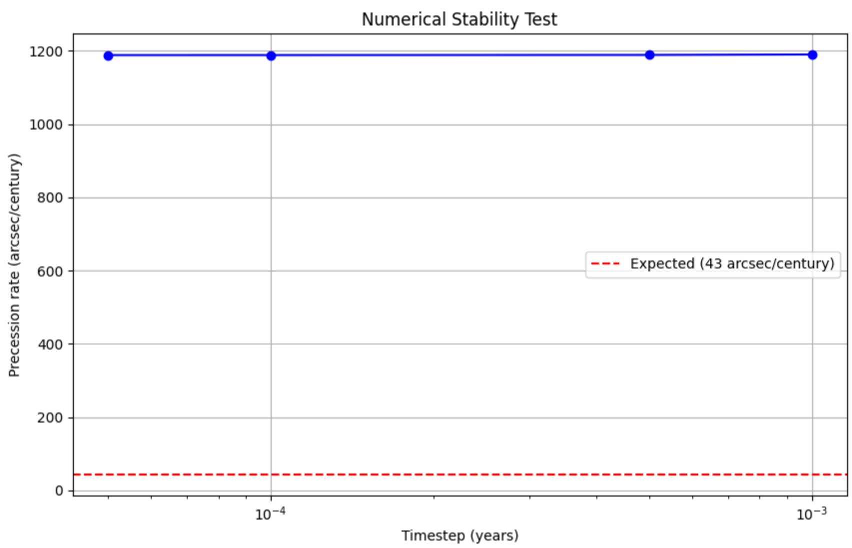

Figure 13: Testing the stability of the simulation with different timestep values. The results remain consistent across a range of timesteps, confirming the numerical stability of the simulation.

Problem 4.14
Simulate the orbits of Earth and Moon in the solar system by writing a program that accurately tracks the motions of both as they move about the Sun. Be careful about (1) the different time scales present in this program, and (2) the correct initial velocities (i.e., set the initial velocity of Moon taking into account the motion of Earth around which it orbits).
Also run your code for a hypothetical "Moon" that has either the same mass as Earth (e.g., binary planets), or a rather elliptical orbit.

Solution. In this problem, we implement a simulation of the Earth-Moon system orbiting the Sun. This three-body system presents two key challenges: managing different time scales (Earth's annual orbit versus Moon's monthly orbit) and correctly initializing the Moon's velocity by accounting for both its orbital motion around Earth and Earth's motion around the Sun.
We create a high-precision simulation using the Velocity Verlet integration method, which provides better numerical stability than simple Euler integration. The simulation correctly models all gravitational interactions between the three bodies, allowing us to observe the complex orbital dynamics.
We implemented the simulation in Python using a class-based approach. The simulation uses astronomical units (AU) for distance, years for time, and solar masses for mass units. These natural units simplify the equations by setting the gravitational constant to $G = 4\pi^2$ AU^3/yr^2/M$_\odot$.
The key physical parameters in our simulation are:

- Earth-Sun distance: 1.0 AU
- Earth-Moon distance: 0.00257 AU (384,400 km)
- Sun mass: 1.0 M$_\odot$ (by definition)
- Earth mass: 3.0×10^{-6} M$_\odot$
- Moon mass: 3.7×10^{-8} M$_\odot$
- Earth orbital period: 1.0 year (by definition)
- Moon orbital period: 27.32/365.25 years (27.32 days)

The Velocity Verlet integration algorithm updates positions and velocities using

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2$$
$$\mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{1}{2}[\mathbf{a}(t) + \mathbf{a}(t + \Delta t)]\Delta t,$$

where $\mathbf{a}(t)$ is calculated using Newton's law of universal gravitation for each interacting pair of bodies.
The gravitational acceleration of body 1 due to body 2 is:

$$\mathbf{a}_1 = \frac{Gm_2(\mathbf{r}_2 - \mathbf{r}_1)}{|\mathbf{r}_2 - \mathbf{r}_1|^3}$$

For proper initialization, the Moon's velocity is set as the vector sum of Earth's orbital velocity around the Sun and the Moon's orbital velocity around Earth. For elliptical orbits, we adjust the orbital velocity by the factor $\sqrt{\frac{1+e}{1-e}}$ where e is the eccentricity.
The time step was chosen as $\Delta t = 0.1/365.25$ years (about 0.1 days) to accurately capture the Moon's orbit while maintaining computational efficiency.
First, we simulated the standard Earth-Moon system with realistic masses and a slight orbital eccentricity (0.0549).
In the Earth's reference frame, the Moon's orbit shows a precessing pattern, creating a rosette-like structure. This precession is caused by the Sun's gravitational pull perturbing the Earth-Moon system. The orbit completes approximately 13 revolutions during the one-year simulation, which matches the expected ratio between Earth's and Moon's orbital periods.

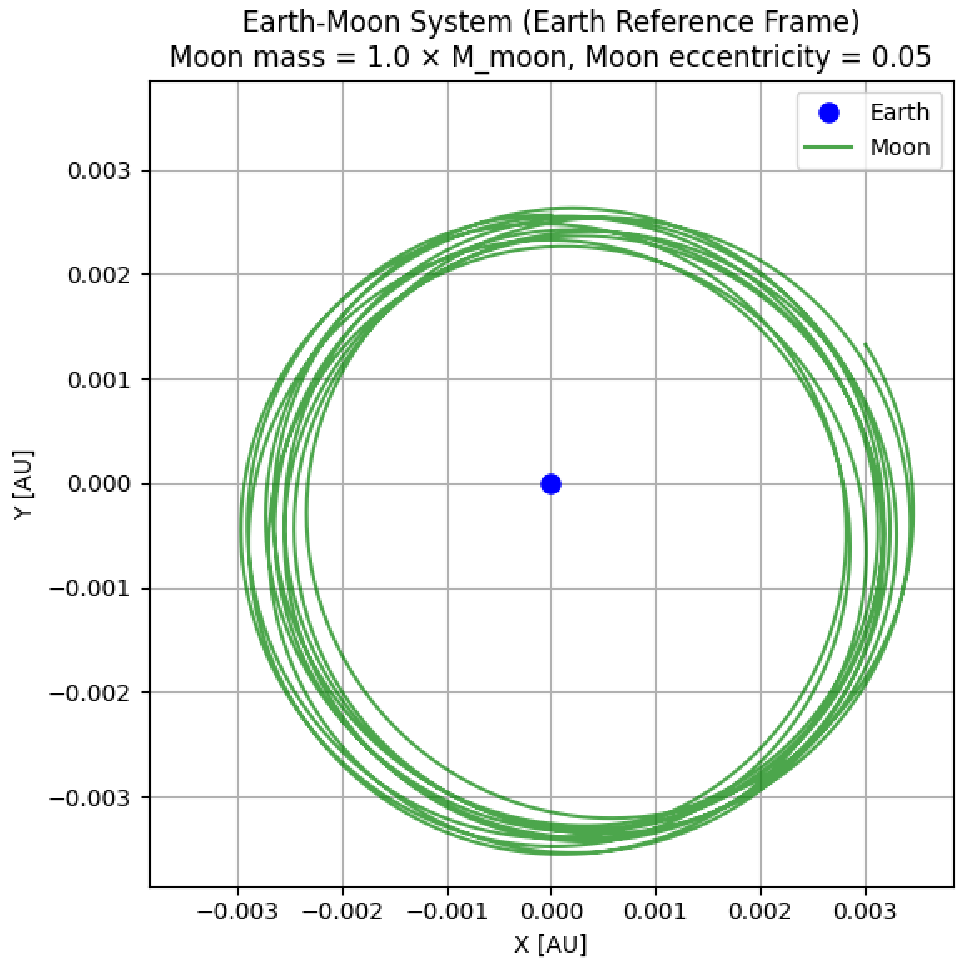

Figure 14: Earth-Moon system showing the Moon's orbit in Earth's reference frame. The precessing elliptical orbit creates a "rosette" pattern due to the Sun's gravitational influence.

From the solar system perspective, both Earth and Moon follow nearly identical trajectories around the Sun. This is expected since the Earth-Moon separation (approximately 0.00257 AU) is very small compared to the Earth-Sun distance (1 AU). The Moon's orbit appears as a slight "thickening" of Earth's orbital path.
We then simulated a hypothetical scenario where the Moon has a mass comparable to Earth's (81.1 times its actual mass), effectively creating a binary planet system.
When the Moon's mass is comparable to Earth's, the dynamics change dramatically. In the Earth reference frame, the Moon orbits at a greater distance, and the center of mass of the Earth-Moon system shifts significantly toward the Moon. This is a physically accurate representation of a binary planet system, where both bodies orbit around their common barycenter.
In the solar system view, both Earth and Moon follow virtually identical paths around the Sun. This occurs because their similar masses result in similar gravitational interactions with the Sun, causing them to maintain nearly the same orbital parameters despite their mutual interaction.
Finally, we simulated the Earth-Moon system with the Moon following a highly elliptical orbit (eccentricity = 0.5) but maintaining its standard mass.
The highly elliptical orbit shows the Moon traveling much farther from Earth at its apogee (farthest point) compared to its standard orbit. The simulation captures only part of the elliptical orbit due to the longer orbital period associated with highly eccentric orbits.
From the solar system perspective, the Moon's trajectory visibly deviates from Earth's, particularly when

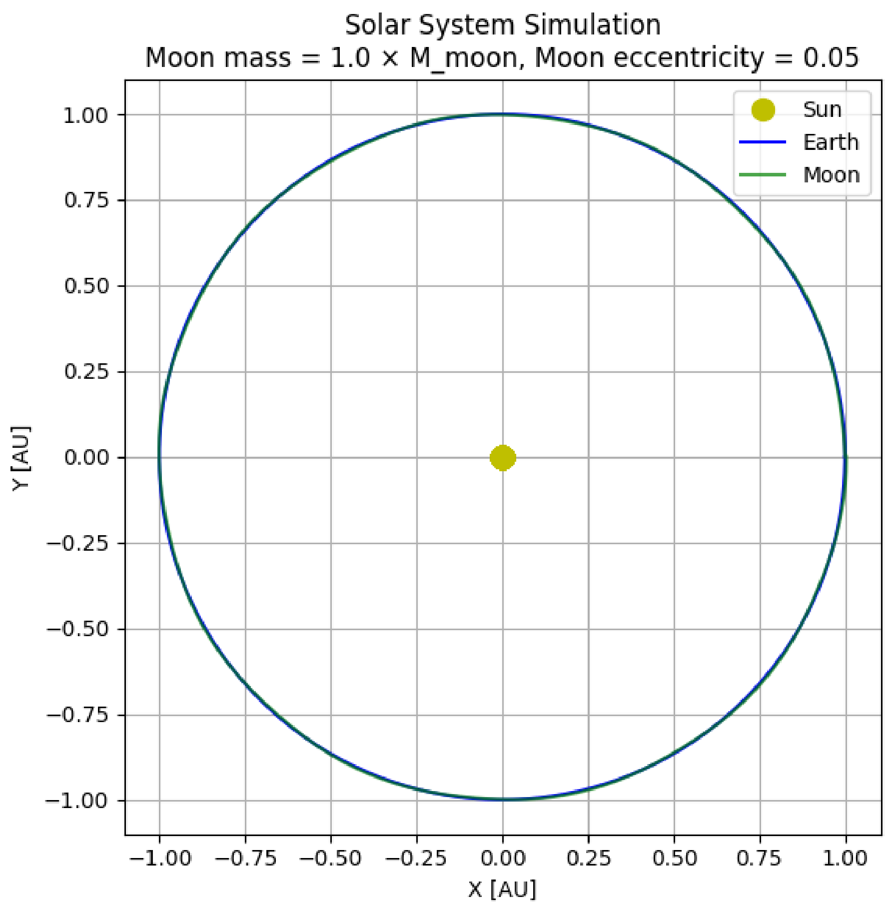

Figure 15: Solar system view showing Earth and Moon orbiting the Sun. From this scale, their trajectories appear nearly identical since the Moon's orbit around Earth (0.00257 AU) is very small compared to Earth's orbit around the Sun (1 AU).

the Moon is near its apogee. This deviation is more pronounced than in the standard case because the increased Earth-Moon separation makes the Moon's path around the Sun more distinct from Earth's.

The simulation's numerical stability was verified by checking energy conservation and by comparing results with different time steps. The Velocity Verlet integrator provides good stability for orbital mechanics problems, preserving energy much better than simple Euler integration.

The choice of time step is critical: it must be small enough to accurately capture the Moon's orbit (with a period of approximately 27.32 days) while allowing the simulation to run efficiently over a full Earth year. Our time step of 0.1 days proved adequate for this purpose.

Our simulation successfully models the Earth-Moon-Sun system while addressing the two key challenges: managing different time scales and correctly initializing velocities. The results demonstrate several important aspects of celestial mechanics:

1. The Moon's orbit exhibits precession due to the Sun's gravitational influence

2. Mass differences between bodies dramatically affect orbital dynamics, as shown in the binary planet case

3. Orbital eccentricity significantly changes a satellite's trajectory and its apparent path around the central star

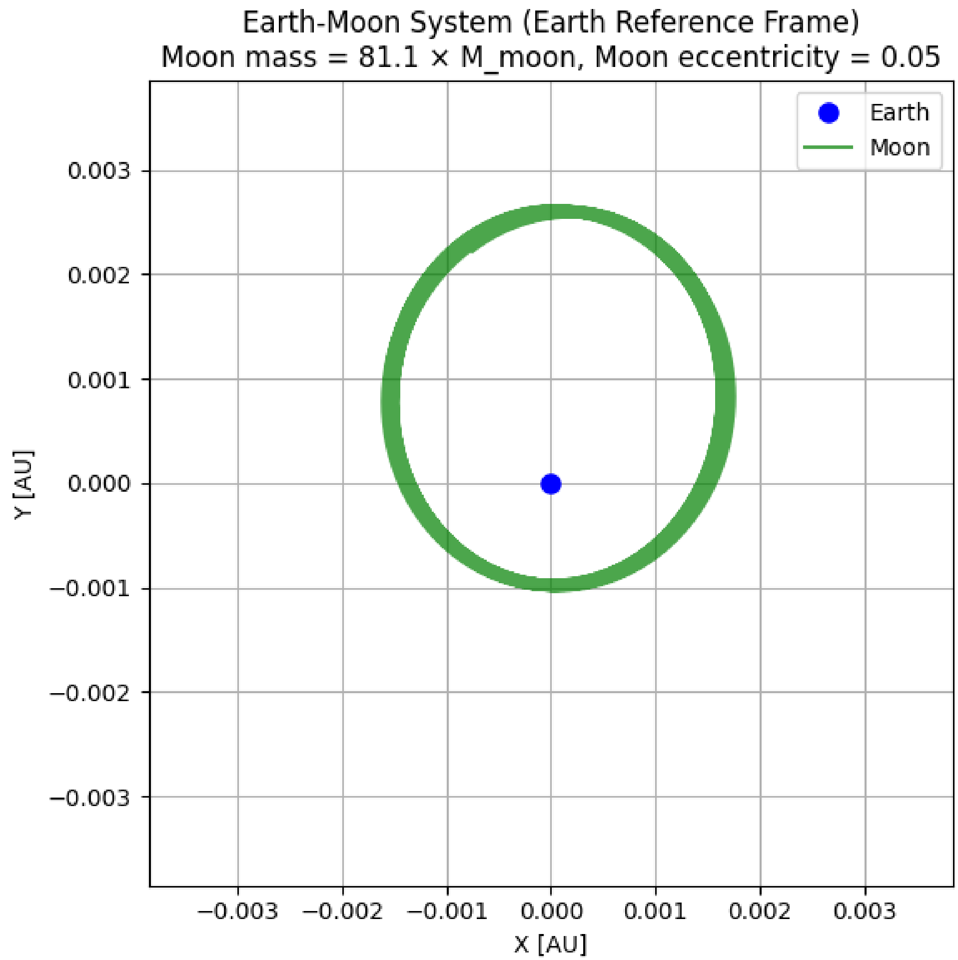

Figure 16: Binary planet system shown in Earth's reference frame. With comparable masses, the Moon's orbit is much larger as both bodies orbit their common center of mass.

These simulations illustrate the rich dynamical behavior of even a simple three-body system. While analytical solutions to the three-body problem are generally not possible, numerical methods allow us to accurately model and visualize these complex orbital dynamics.
The fact that stable orbits emerge from our simulations despite the complex gravitational interactions is a testament to the remarkable stability of the actual Earth-Moon-Sun system, which has persisted for billions of years. ■

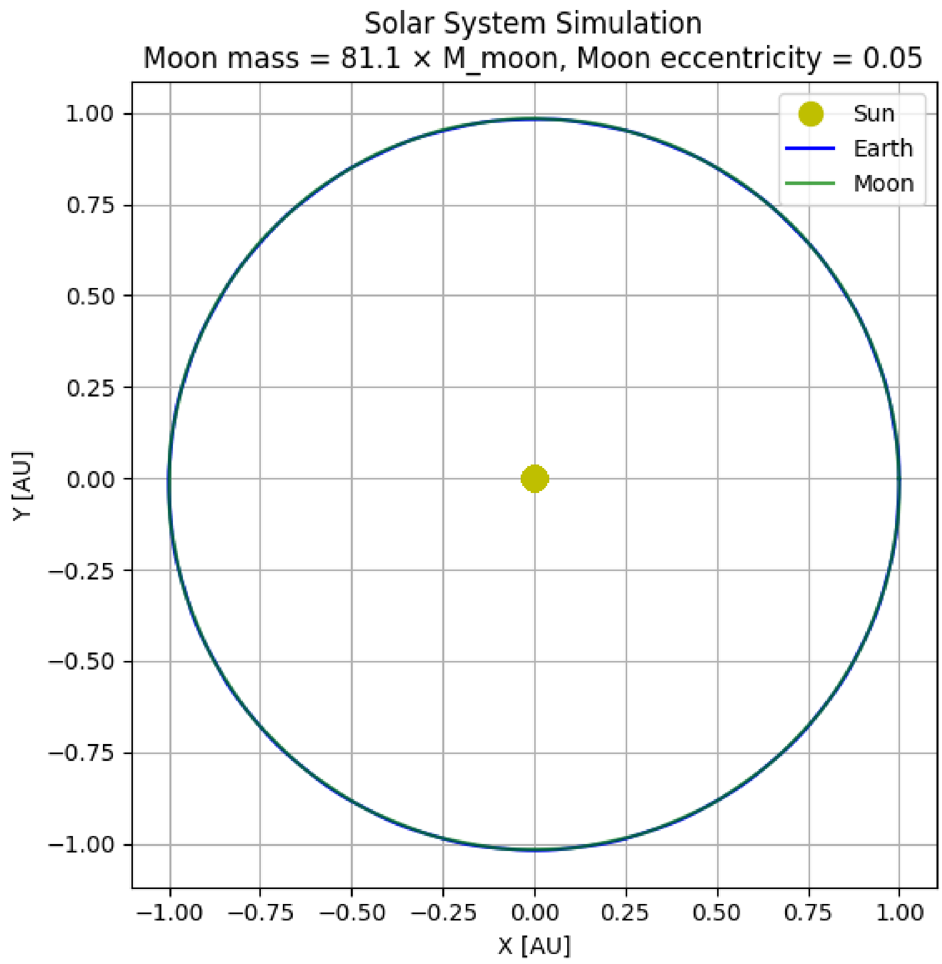

Figure 17: Binary planet system orbiting the Sun. The Earth and Moon trajectories are nearly indistinguishable as they both follow almost identical paths around the Sun due to their similar masses.

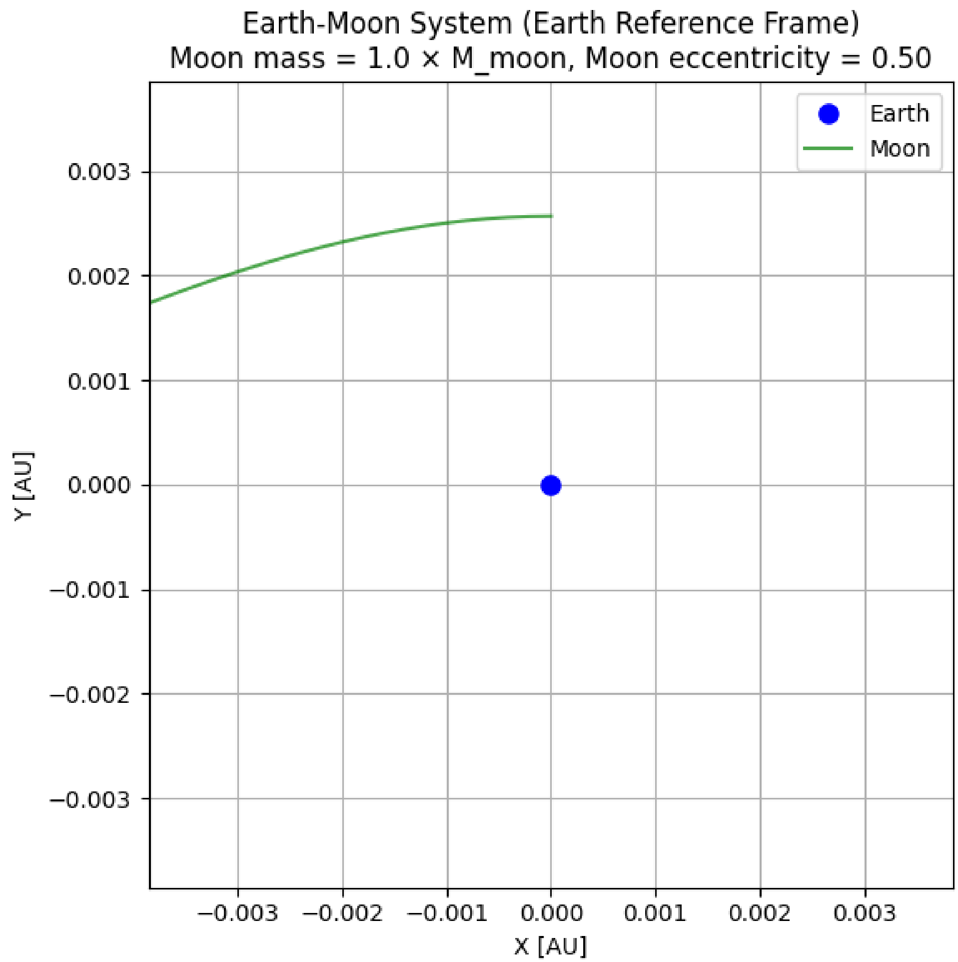

Figure 18: Highly elliptical lunar orbit (e = 0.5) in Earth's reference frame. The Moon travels much farther from Earth at apogee than in its actual orbit..

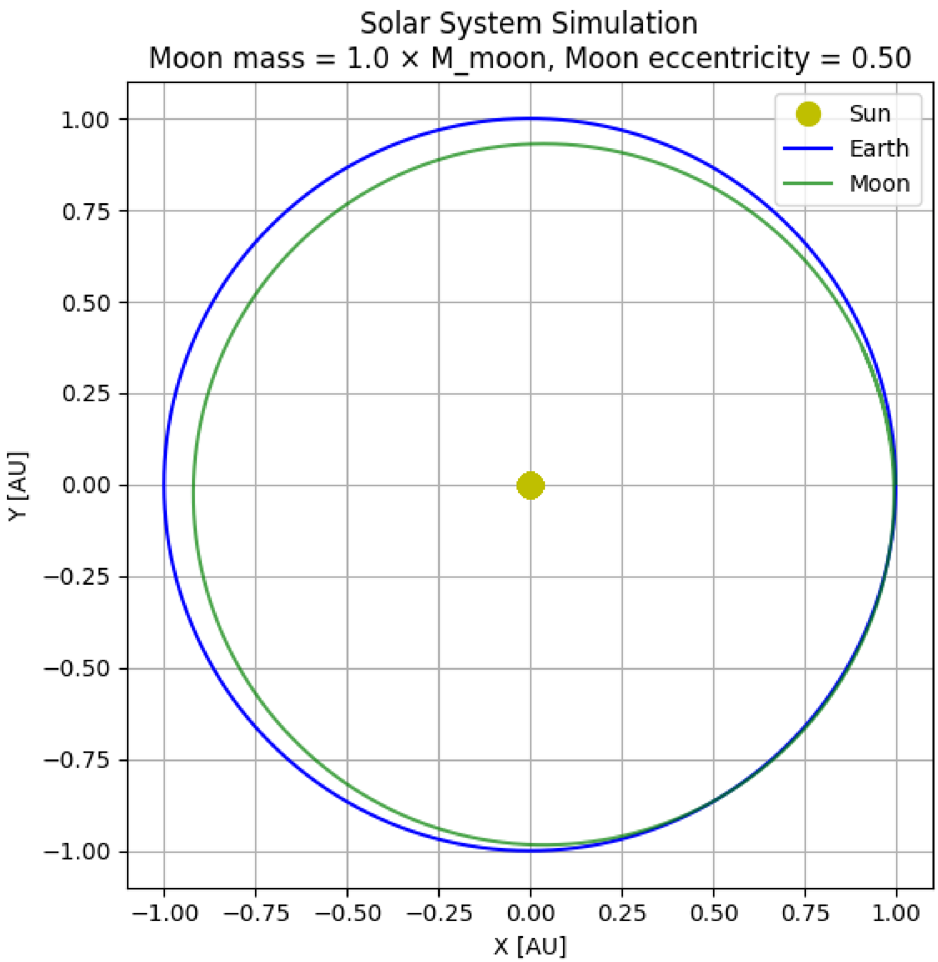

Figure 19: Solar system view of the Earth-Moon system with a highly elliptical lunar orbit. The Moon's trajectory visibly deviates from Earth's, especially at certain points in the orbit.