The perihelion precession of Mercury: An exploration of numerical simulations in Python

Amir Zarandi

Numerical simulations are playing an increasingly important role in modern science, especially physics. In this paper, a numerical study of the famous perihelion motion of the planet Mercury (one of the prime observables supporting Einstein's General Relativity) is demonstrated as a test case for numerical simulations in Python. The paper includes details about the development of the code as well as a discussion of the visualization of the results.

First, we try plotting the orbit of Mercury by itself, affected only by the gravity of Sun, with no influence from the other planets. The starting point for this analysis is Newton's Second Law of motion:

$$\sum \vec{F}(\vec{r}_m) = M_m \ddot{\vec{r}}_m \tag{1}$$

Where M_m is the mass and $\vec{r}_m(t)$ is the position vector of Mercury. The only force acting on Mercury in this case is Sun's gravitational pull, thus, assuming Sun is located at the center of the coordinate system:

$$\sum \vec{F}(\vec{r}_m) = -\,GM_sM_m\frac{\vec{r}_m}{r_m^3} = M_m\ddot{\vec{r}}_m \tag{2}$$

Dividing both sides by M_m yields:

$$\ddot{\vec{r}}_m = -\,GM_s\frac{\vec{r}_m}{r_m^3} \tag{3}$$

where $G = 6.67 \times 10^{-11} m^3 kg^{-1} s^{-2} = 1.98 \times 10^{-29} AU^3 Kg^{-1} year^{-2}$ is the Newtonian constant of gravitation. The unit conversion for G is of crucial importance for our simulation since for any problem, both for the numerical treatment and the intuitive understanding, it is useful to select parameters in a "natural range." As such, we are aiming to express distances in astronomical units and time in years. In addition, $r_m = |\vec{r}_m|$ denotes the distance between Sun and Mercury.

Thus, Newton's Law of universal gravitation gives us an ordinary differential equation in this case for which we can use the Odeint function from the integrate module of the SciPy library to evaluate numerically.

One key assumption we make here is that all the planets in the solar system have angular momentum enough to maintain their observed orbits with the corresponding semi-major axis and eccentricity. Doing so allows us to determine the initial conditions for this differential equation if we also assume the planets lay exactly on their perihelion at $t = 0$. Note that the trajectories do not depend on where the orbit of Mercury in the simulation is started

In assuming an elliptical orbit for all the planets in the solar system, we have sun as one of the foci, meaning perihelion and aphelion, as the nearest and farthest points on the orbit are on the major axis, as are both foci and the center. The mathematical characteristic of the ellipse then allows us to calculate the perihelion distance using the semi-major axis and the eccentricity as so:

$$\rho_p = a(1 - e) \tag{4}$$

Setting the x-axis of our coordinate system as the semi major axis of the starting orbit gives us $x(0) = \rho_p$ and $y(0) = 0$, thus the $\vec{r}_m(0) = \langle \rho_p , 0 \rangle$.

At perihelion and aphelion, the velocities are purely tangential, hence conservation of angular momentum yields:

$$\rho_p v_p = \rho_a v_a \tag{5}$$

$$a(1 - e)v_p = a(1 + e)v_a \tag{6}$$

Conservation of energy (potential plus kinetic) then gives us:

$$-\frac{GM_s}{\rho_p} + \frac{v_p^2}{2} = -\frac{GM_s}{\rho_a} + \frac{v_a^2}{2} \tag{7}$$

$$-\frac{GM_s}{\rho_p} + \frac{v_p^2}{2} = -\frac{GM_s}{a(1 + e)} + \frac{v_a^2}{2} \tag{8}$$

Solving the system of equations given by Equations 6 and 8 for v_p gives:

$$v_p = \left(\frac{GM_s}{\rho_p}(1 + e)\right)^{1/2} \tag{9}$$

As mentioned earlier, this velocity is tangential, meaning it's entirely in the y-axis direction in the case of our coordinate system at $t = 0$. This means $v_x(0) = 0, v_y(0) = v_p$, hence $\dot{\vec{r}}_m(0) = \langle 0, v_p \rangle$

There's of course nothing special about Mercury in this case and these initial conditions are true for any planet starting its motion from the perihelion. Hence, as we later add more planets to our system, we'll use Equation 4 and 9 time and time again to obtain initial conditions for our differential equations. As a result, it is handy to use Python functions to structure the program and hold repeating code ("DRY" — Don't Repeat Yourself).

Next, we transform the equation of motion for Mercury into a function that can be handled by the ODE solving machinery. Since this equation is independent of the mass of Mercury, it can be generalized to any one planet system. We employ the linear algebra submodule of NumPy for ease of calculation when getting the magnitude of our vectors.

Given an array containing x, y, vx, and vy in the respectuive order, this function return the derivatives, i.e. vx, vy, ax, ay. The reason being that SciPy can only handle first order differential equations as opposed to the second order one we have, so we break it down into two coupled first order ordinary differential equations.

To do so, we have first defined a position vector from sun to the corresponding planet, then used the differential equation to obtain the second time derivatives. The first time derivatives are given by the vx and vy in the initial array.

Running this code for 120 million iterations over the course of 120 years (each iteration being around 30 seconds), gives us a graph like that in Figure 1.

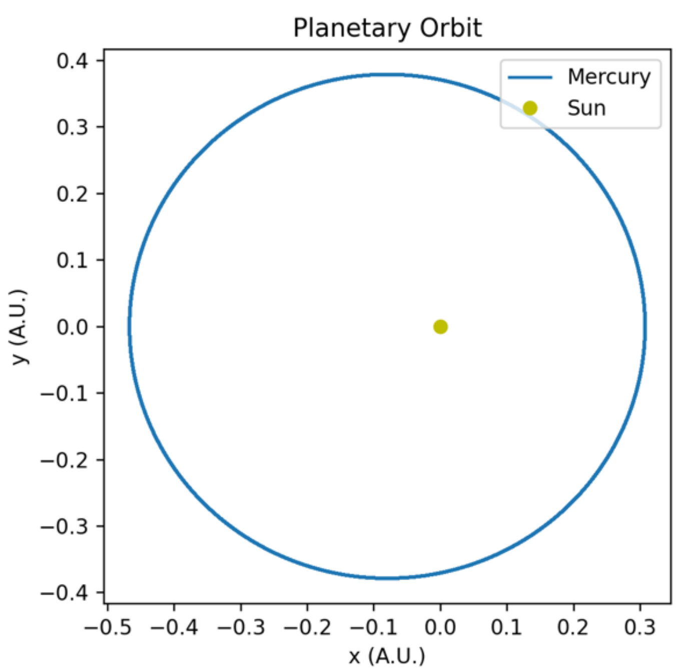

Figure 1: Orbit of Mercury under no influence from other planets.

Mercury's orbit in this case, as expected, is slightly elliptical, with sun as one of the foci. We're not expecting any precession in this case as the orbit is not affected by anything other than Sun, however, it's good practice to develop the algorithm we'll use to track the perihelion in more complicated cases. Any precession we'll see in this case will be due to the error in solving the differential equation numerically.

To begin, we define a distance function, which, iterating through x and y coordinates, calculates the angular distances from the origin and returns a one-dimensional array containing those. Next, we use the arglreextrema function from the signal submodule of SciPy Library to find the indices of the local minima in an array containing polar distances

Note that arglreextrema returns a tuple containing an array, thus why we must take the zeroth index as our target array. This part is by far the most time-consuming part of the code, and it's advised to run the code for a smaller number of time steps just to avoid running into memory issues. This function seems to have a large time complexity as switching from 120 million to 12 million iterations reduces its runtime from around 7 minutes to 30 seconds.

Perihelion of the orbits of mercury, as the point of closest approach will correspond to the local minima in the array of angular distances. Using the indices from our function, not only can we reduce the position-velocity array of mercury to position array of the perihelion, but we can also reduce the timestamp array to the timestamp of the perihelion.

It's worth mentioning that at first, I tried using the fact that the velocity along Sun-planet axis at the perihelion is zero, and goes from negative to positive in that vicinity, however neglecting the fact that the Sun-planet axis moves as the perihelion precesses, I mistakenly looked at the velocities in the x direction. Needless to say, this yielded faulty presession angles. Switching to an algorithm that looked at the local minima of the array of distances seemed to be the more sensible option in this case.

We then define an angle function to iterate through the perihelion position array and calculate the angle the perihelion makes with the x-axis. Using all these functions, we can define a function that given a solution array to the differential equations, returns the times at which the perihelion happens, the angle and the angular distance of each perihelion.

Plotting the trajectory of the perihelion over time in polar coordinates gives us Figure 2a. This graph tells us that the angle and distance are approximately the same over time. A closer examination however gives us better results. Figure 2b shows the angle over time plotted to arcsec accuracy.

We can define a best fit function to find the slope of the line of the best fit to this graph. Doing so, yields a slope of -0.24 corresponding to a precession of 24 arcseconds per century. As mentioned before, this is all due to arithmetic error, which also tells us that our future results will have some error in an order of 20 arcsecs or smaller.

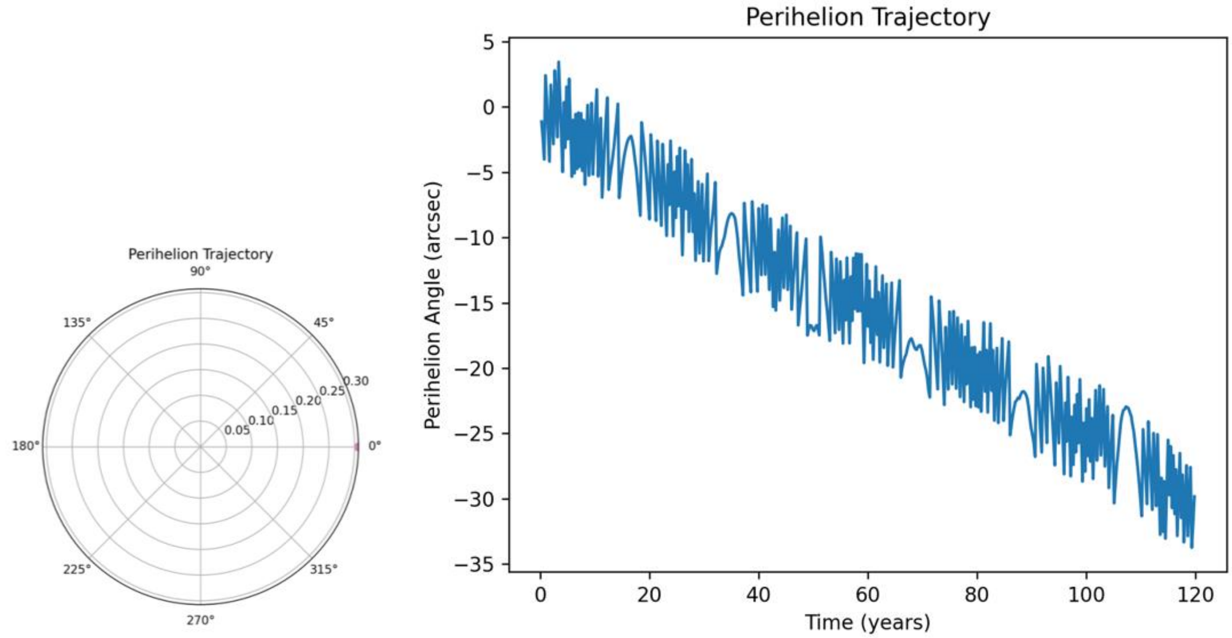

Figure 2: Perihelion of Mercury under the influence of Sun alone: a) polar coordinates b) angle versus time.

Now, we would like to shift our focus to multi-planetary systems. Newton's second law for a system of two planets (Mercury plus some planet p) would yield:

$$\sum \vec{F}(\vec{r}_m) = M_m \ddot{\vec{r}}_m \tag{10}$$

$$\sum \vec{F}(\vec{r}_p) = M_p \ddot{\vec{r}}_p \tag{11}$$

The forces acting on each planet in this case includes the gravitational pull of sun and the other planet. Hence, we get the following equations:

$$M_m\, \ddot{\vec{r}}_m = -GM_s M_m \frac{\vec{r}_m}{r_m^3} - GM_p M_m \frac{\vec{r}_m - \vec{r}_p}{|\vec{r}_m - \vec{r}_p|^3} \tag{12}$$

$$M_p\, \ddot{\vec{r}}_p = -\, GM_s M_p \frac{\vec{r}_p}{r_p^3} - GM_p M_m \frac{\vec{r}_p - \vec{r}_m}{|\vec{r}_p - \vec{r}_m|^3} \tag{13}$$

Once again, dividing through by the mass of each planet would yield:

$$\ddot{\vec{r}}_m = -GM_s \frac{\vec{r}_m}{r_m^3} - GM_p \frac{\vec{r}_m - \vec{r}_p}{\left|\vec{r}_m - \vec{r}_p\right|^3} \tag{14}$$

$$\ddot{\vec{r}}_p = -GM_s \frac{\vec{r}_p}{r_p^3} - GM_m \frac{\vec{r}_p - \vec{r}_m}{\left|\vec{r}_p - \vec{r}_m\right|^3} \tag{15}$$

The initial conditions for these differential equations would again be the perihelion and the perihelion velocity of each planet as we assume they both start their orbit at the perihelion. Here we make another crucial assumption about our celestial bodies: the starting position of the system, the semi major axis and the perihelion at t = 0, for all planets lay on the x-axis.

Notice how the effect the second planet will have on mercury is proportional to its mass and inversely proportional to the square of its distance from mercury. A quick analysis yields that this ratio of mass to the square of distance gives us these top three planets: Jupiter > Venus > Earth. As we later see though, Venus has a bigger impact on precession of Mercury than Jupiter. It can be shown that the combined effect of all the other planets on precession of mercury is less than 10 arcsecs (through many pointless hours of coding) and for simplicity purposes can be ignored.

We begin the next part of our analysis by adding Venus to the system. This gives us the planetary orbits shown in Figure 3.

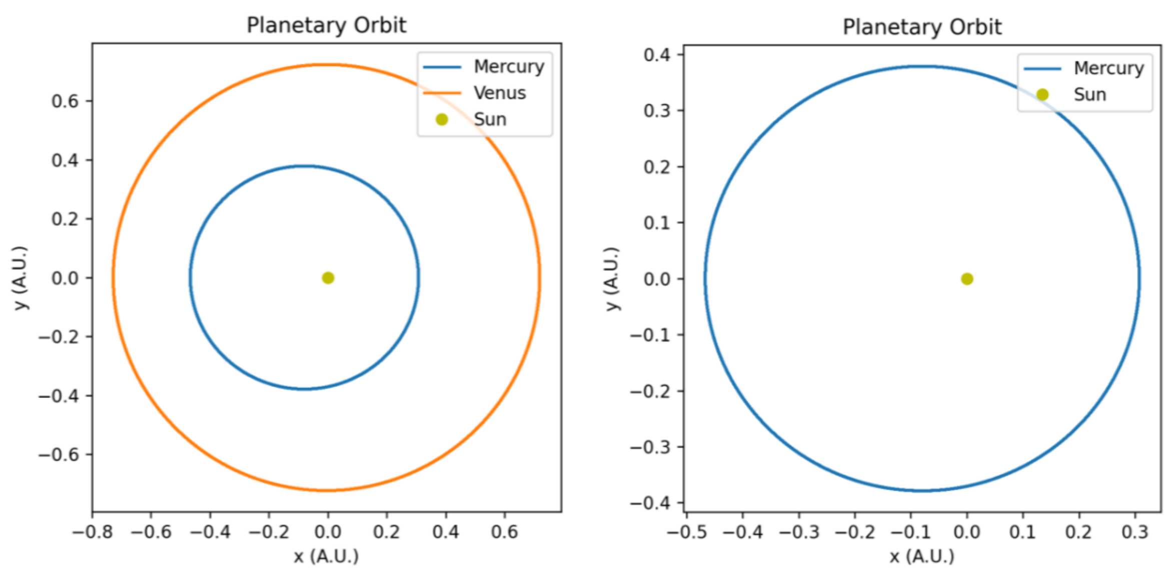

Figure 3: Orbit of Mercury under the influence of Sun and Venus: a) both orbits b) Mercury alone.

Once again, we look at the trajectory of the perihelion over time. The best fit line to the graph in Figure 4b has a slope of 2.63, signaling a precession of 263 arcseconds per century.

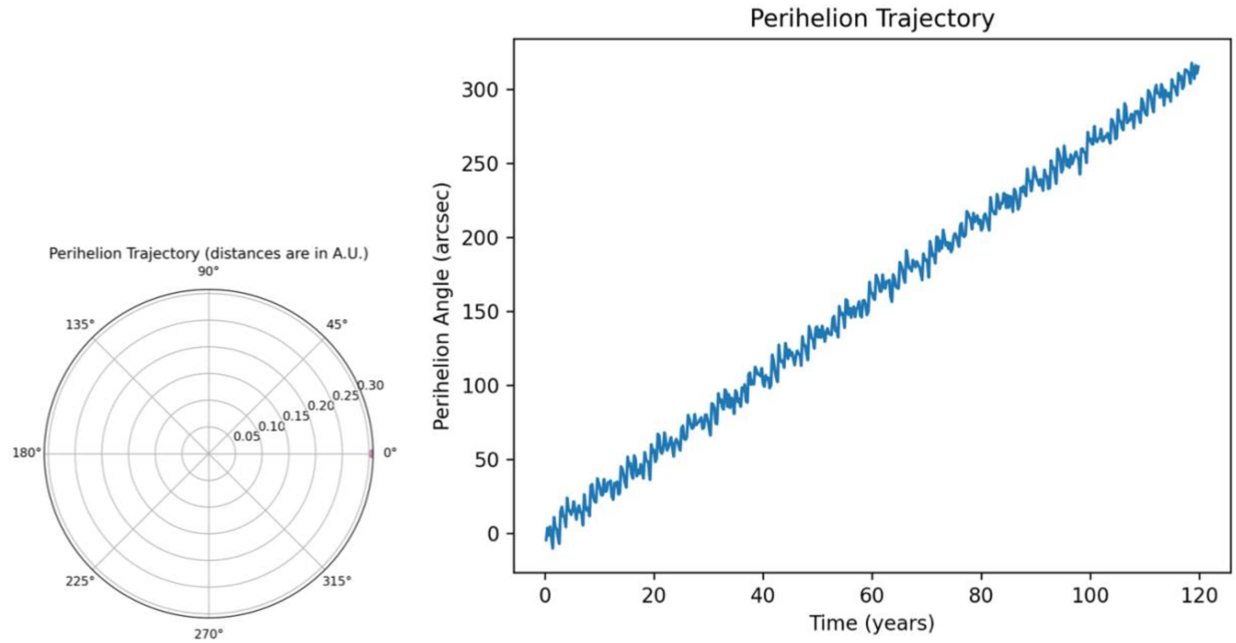

Figure 4: Perihelion of Mercury under the influence of Sun and Venus: a) polar coordinates b) angle versus time.

We continue this procedure for Earth and Jupiter. The equations of motion for a three-planet system (m, p1 and p2) are in a similar fashion to the previous ones:

$$\ddot{\vec{r}}_m = -GM_s \frac{\vec{r}_m}{r_m^3} - GM_{p1} \frac{\vec{r}_m - \vec{r}_{p1}}{\left|\vec{r}_m - \vec{r}_{p1}\right|^3} - GM_{p2} \frac{\vec{r}_m - \vec{r}_{p2}}{\left|\vec{r}_m - \vec{r}_{p2}\right|^3} \tag{16}$$

$$\ddot{\vec{r}}_{p1} = -\,GM_s \frac{\vec{r}_{p1}}{r_{p1}^3} - GM_m \frac{\vec{r}_{p1} - \vec{r}_m}{\left|\vec{r}_{p1} - \vec{r}_m\right|^3} - GM_{p2} \frac{\vec{r}_{p1} - \vec{r}_{p2}}{\left|\vec{r}_{p1} - \vec{r}_{p2}\right|^3} \tag{17}$$

$$\ddot{\vec{r}}_{p2} = -\,GM_s \frac{\vec{r}_{p2}}{r_{p2}^3} - GM_m \frac{\vec{r}_{p2} - \vec{r}_m}{\left|\vec{r}_{p2} - \vec{r}_m\right|^3} - GM_{p1} \frac{\vec{r}_{p2} - \vec{r}_{p1}}{\left|\vec{r}_{p2} - \vec{r}_{p1}\right|^3} \tag{18}$$

Adding Earth yields the orbits in Figure 5 and the perihelion in Figure 6. A best fit line of 3.51 gives us a total precession of 351 arcsecs per century.

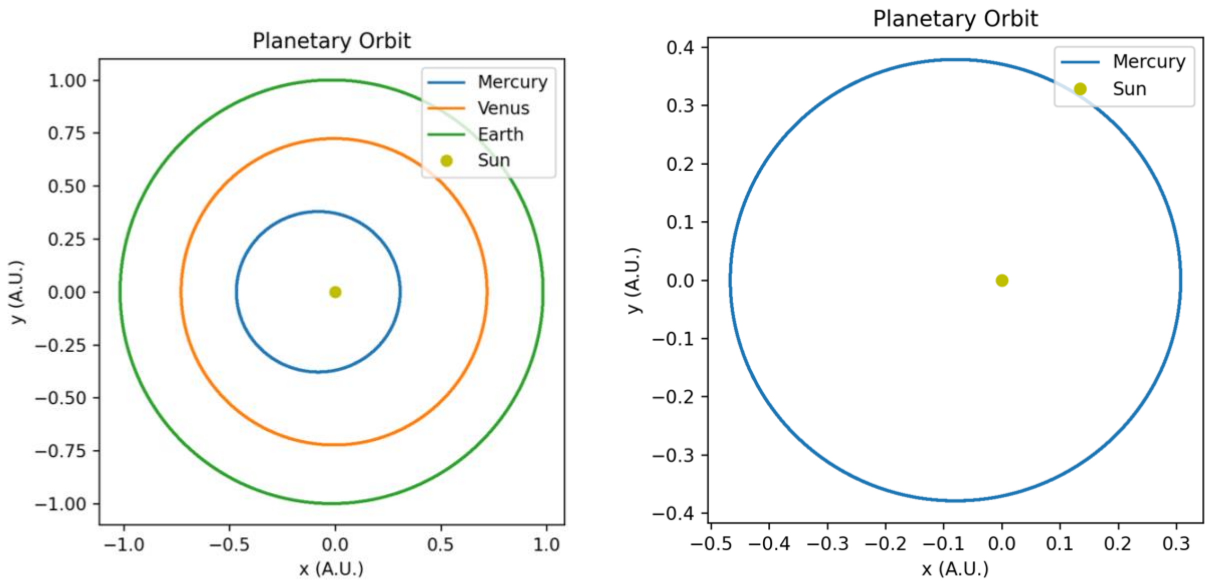

Figure 5: Orbit of Mercury under the influence of Sun, Venus, and Earth: a) all three orbits b) Mercury alone.

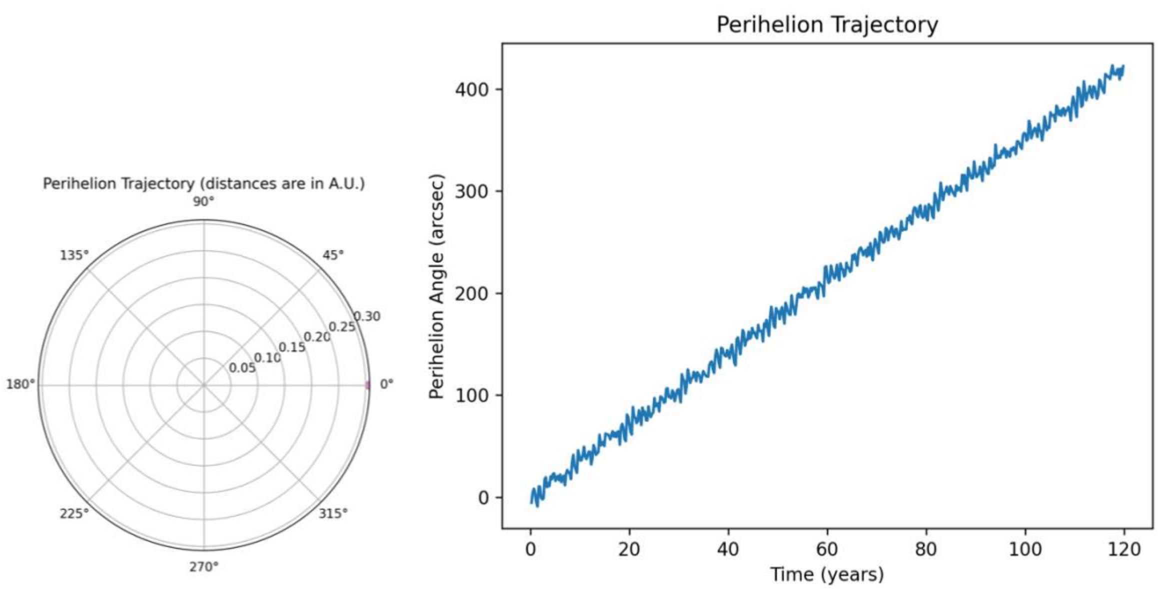

Figure 6: Perihelion of Mercury under the influence of Sun, Venus, and Earth: a) polar coordinates b) angle versus time.

For a four-body system, we get a similar set of equations:

$$\ddot{\vec{r}}_m = -GM_s \frac{\vec{r}_m}{r_m^3} - GM_{p1} \frac{\vec{r}_m - \vec{r}_{p1}}{\left|\vec{r}_m - \vec{r}_{p1}\right|^3} - GM_{p2} \frac{\vec{r}_m - \vec{r}_{p2}}{\left|\vec{r}_m - \vec{r}_{p2}\right|^3} - GM_{p3} \frac{\vec{r}_m - \vec{r}_{p3}}{\left|\vec{r}_m - \vec{r}_{p3}\right|^3} \tag{19}$$

$$\ddot{\vec{r}}_{p1} = -GM_s\frac{\vec{r}_{p1}}{r_{p1}^3} - GM_m\frac{\vec{r}_{p1}-\vec{r}_m}{\left|\vec{r}_{p1}-\vec{r}_m\right|^3} - GM_{p2}\frac{\vec{r}_{p1}-\vec{r}_{p2}}{\left|\vec{r}_{p1}-\vec{r}_{p2}\right|^3} - GM_{p3}\frac{\vec{r}_{p1}-\vec{r}_{p3}}{\left|\vec{r}_{p1}-\vec{r}_{p3}\right|^3} \tag{20}$$

$$\ddot{\vec{r}}_{p2} = -GM_s\frac{\vec{r}_{p2}}{r_{p2}^3} - GM_m\frac{\vec{r}_{p2}-\vec{r}_m}{\left|\vec{r}_{p2}-\vec{r}_m\right|^3} - GM_{p1}\frac{\vec{r}_{p2}-\vec{r}_{p1}}{\left|\vec{r}_{p2}-\vec{r}_{p1}\right|^3} - GM_{p3}\frac{\vec{r}_{p2}-\vec{r}_{p3}}{\left|\vec{r}_{p2}-\vec{r}_{p3}\right|^3} \tag{21}$$

$$\ddot{\vec{r}}_{p3} = -GM_s\frac{\vec{r}_{p3}}{r_{p3}^3} - GM_m\frac{\vec{r}_{p3}-\vec{r}_m}{\left|\vec{r}_{p3}-\vec{r}_m\right|^3} - GM_{p1}\frac{\vec{r}_{p3}-\vec{r}_{p1}}{\left|\vec{r}_{p3}-\vec{r}_{p1}\right|^3} - GM_{p2}\frac{\vec{r}_{p3}-\vec{r}_{p2}}{\left|\vec{r}_{p3}-\vec{r}_{p2}\right|^3} \tag{22}$$

Adding Jupiter yields the orbit in Figure 7 and the perihelion in Figure 8.

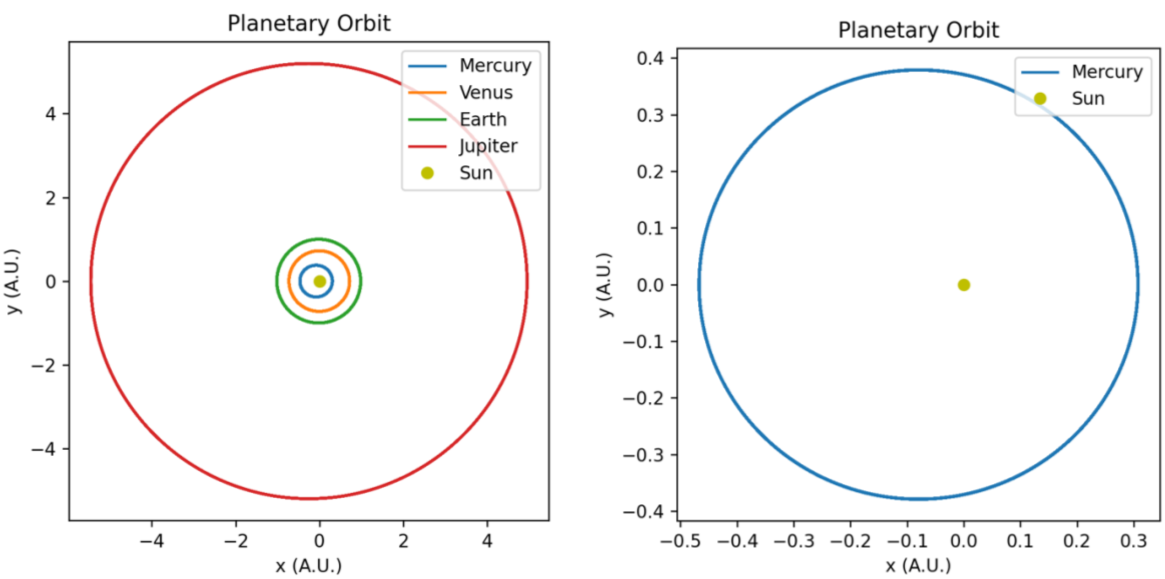

Figure 7: Orbit of Mercury under the influence of Sun, Venus, Earth, and Jupiter: a) all four orbits b) Mercury alone.

Not so surprisingly, Jupiter has a noticeable effect on the oscillations of the perihelion angle-time graph. These oscillations are around 12 years, corresponding to the period of Jupiter's orbit. We made sure to have roughly the integer number of Jupiter orbits in this case to get an accurate best fit line since an unsymmetrical graph would largely impact the slope. A best fit line of 5.42 gives us a total precession of 542 arcsecs per century.

We have arrived at a discrepancy of around 34 arcsecs. As it was shown historically, this cannot be resolved by classical mechanics, and we need to take into account corrections made to the distances by general relativity.

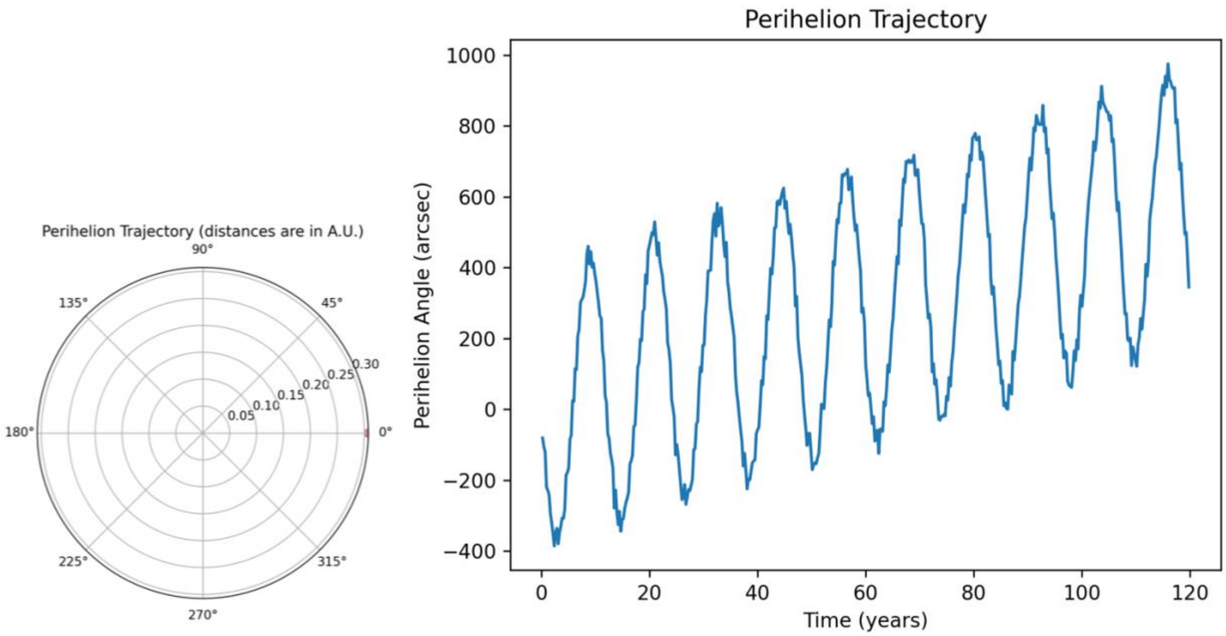

Figure 8: Perihelion of Mercury under the influence of Sun, Venus, Earth, and Jupiter: a) polar coordinates b) angle versus time.

An approximate form would be to apply some relativistic correction factor α to any vector magnitude in Equations 19 through 22. This would result in:

$$\ddot{\vec{r}}_m = -GM_s \frac{\vec{r}_m}{r_m^3}\left(1+\frac{\alpha}{r_m^2}\right) - GM_{p1}\frac{\vec{r}_m-\vec{r}_{p1}}{\left|\vec{r}_m-\vec{r}_{p1}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_m-\vec{r}_{p1}\right|^2}\right) - GM_{p2}\frac{\vec{r}_m-\vec{r}_{p2}}{\left|\vec{r}_m-\vec{r}_{p2}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_m-\vec{r}_{p2}\right|^2}\right) - GM_{p3}\frac{\vec{r}_m-\vec{r}_{p3}}{\left|\vec{r}_m-\vec{r}_{p3}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_m-\vec{r}_{p3}\right|^2}\right) \tag{23}$$

$$\ddot{\vec{r}}_{p1} = -GM_s \frac{\vec{r}_{p1}}{r_{p1}^3}\left(1+\frac{\alpha}{r_{p1}^2}\right) - GM_m\frac{\vec{r}_{p1}-\vec{r}_m}{\left|\vec{r}_{p1}-\vec{r}_m\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p1}-\vec{r}_m\right|^2}\right) - GM_{p2}\frac{\vec{r}_{p1}-\vec{r}_{p2}}{\left|\vec{r}_{p1}-\vec{r}_{p2}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p1}-\vec{r}_{p2}\right|^2}\right) - GM_{p3}\frac{\vec{r}_{p1}-\vec{r}_{p3}}{\left|\vec{r}_{p1}-\vec{r}_{p3}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p1}-\vec{r}_{p3}\right|^2}\right) \tag{24}$$

$$\ddot{\vec{r}}_{p2} = -GM_s \frac{\vec{r}_{p2}}{r_{p2}^3}\left(1+\frac{\alpha}{r_{p2}^2}\right) - GM_m\frac{\vec{r}_{p2}-\vec{r}_m}{\left|\vec{r}_{p2}-\vec{r}_m\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p2}-\vec{r}_m\right|^2}\right) - GM_{p1}\frac{\vec{r}_{p2}-\vec{r}_{p1}}{\left|\vec{r}_{p2}-\vec{r}_{p1}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p2}-\vec{r}_{p1}\right|^2}\right) - GM_{p3}\frac{\vec{r}_{p2}-\vec{r}_{p3}}{\left|\vec{r}_{p2}-\vec{r}_{p3}\right|^3}\left(1+\frac{\alpha}{\left|\vec{r}_{p2}-\vec{r}_{p3}\right|^2}\right)$$

$$\ddot{\vec{r}}_{p3} = -GM_s \frac{\vec{r}_{p3}}{r_{p3}^3}\left(1 + \frac{\alpha}{r_{p3}^2}\right) - GM_m \frac{\vec{r}_{p3} - \vec{r}_m}{\left|\vec{r}_{p3} - \vec{r}_m\right|^3}\left(1 + \frac{\alpha}{\left|\vec{r}_{p3} - \vec{r}_m\right|^2}\right) - GM_{p1} \frac{\vec{r}_{p3} - \vec{r}_{p1}}{\left|\vec{r}_{p3} - \vec{r}_{p1}\right|^3}\left(1 + \frac{\alpha}{\left|\vec{r}_{p3} - \vec{r}_{p1}\right|^2}\right) - GM_{p2} \frac{\vec{r}_{p3} - \vec{r}_{p2}}{\left|\vec{r}_{p3} - \vec{r}_{p2}\right|^3}\left(1 + \frac{\alpha}{\left|\vec{r}_{p3} - \vec{r}_{p2}\right|^2}\right) \quad (25)$$

(26)

The corrected vector value can be defined as a Python function. Although we now $\alpha \approx 1.1 \times 10^{-8} AU^2$, we would first like to see what correction factor would yield the experimental precession of 575 arcseconds per century. The effect of this relativistic correction is directly proportional to α, a linear regression between alpha and precession angle yields the graph in Figure 9.

Thus, our calculations yield some alpha of $0.9 \times 10^{-8} AU^2$. Plugging in the real value of alpha, we get a precession of 580 arcseconds century. This is an error of less than one percent and we have already predicted an error in order of 3 percent or less based on calculations alone. Considering the effect of other planets in the solar system, the precession of Sun itself, the 3D nature of the orbits, there's multiple sources of error. All and all, the presented numerical simulation gives us a very good estimate of the precession of Mercury.

In conclusion, this particular numerical simulation of the Newtonian analysis of Mercury's motion based on a simplified model leads to the conclusion that Newtonian gravity would predict Mercury to precess by 542 arcseconds per century. As expected, there exists some discrepancy that can be attributed to General Relativity, which when taken into account, gives us a total precession of 580 arcseconds per century, reasonably close to the experimental results. Finally, I would like to say it's been so awesome to work on a problem a Nobel Laureate sweated over!

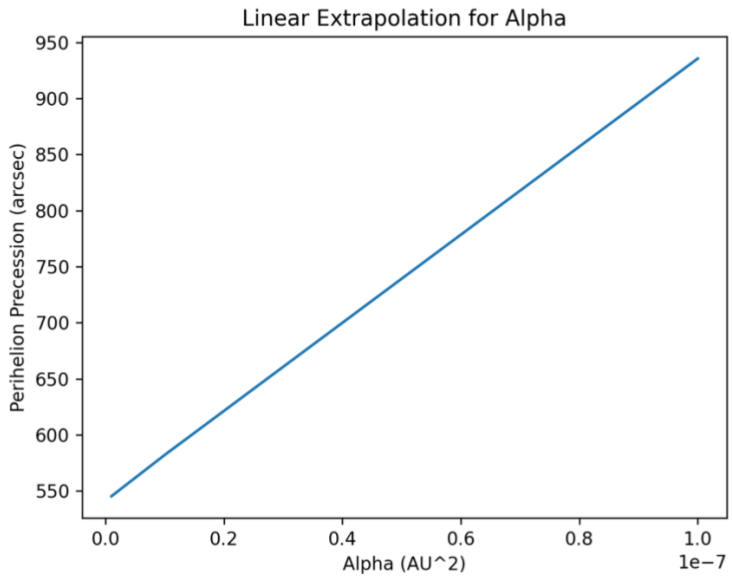

Figure 9: Precession versus correction factor.

I wish to acknowledge the support of the Lawrence University Physics Department and specially Professor Martin.