Analysis of Mercury's Perihelion Precession: A Numerical Approach Using Second-Order Runge-Kutta Method*

Escobar Matzir, Ricardo José Manuel, 202002342[1, †]

[1] *Escuela de Ciencias Físicas y Matemáticas, Universidad de San Carlos de Guatemala, Zona 12, Guatemala.*

This work analyzes Mercury's perihelion precession using a numerical approach based on the second-order Runge-Kutta method implemented in Python 3. Historically, this problem could not be explained using Newton's theory of gravitation. It was not until Albert Einstein formulated his theory of general relativity that this phenomenon was finally understood. To address this issue without resorting to the complex calculations of relativity, we start with a relativistic correction to Newton's law of gravitation and implement it using the numerical method. The results show that the precession rate calculated with this method aligns with actual observations, highlighting the effectiveness of using a numerical approach.

I. INTRODUCTION

Mercury's perihelion precession has been a historical challenge for physics. Although Kepler's laws and Newton's theory of gravitation describe planetary orbits with remarkable accuracy, these theories failed to explain certain anomalies in Mercury's orbit, particularly the slow shift of its perihelion. In the mid-19th century, astronomer Urbain Le Verrier identified this problem and attempted to solve it by proposing the existence of an unknown planet. However, the definitive solution came in the early 20th century when Albert Einstein formulated his general theory of relativity, which finally provided a satisfactory explanation for this phenomenon.

In this work, we study the slow shift of Mercury's perihelion using a numerical approach that incorporates relativistic corrections to Newton's law of gravitation. We employ the second-order Runge-Kutta method implemented in Python to solve the equations of motion. This approach allows us to simulate the planet's orbit and quantify the rate of its perihelion precession. By comparing the value obtained from the simulation with the observed value of 43 arcseconds per century, we find a good agreement.

Through this study, we aim not only to validate Einstein's general theory of relativity but also to demonstrate the utility of numerical methods in analyzing complex physics problems where analytical approaches become exceedingly difficult to apply.

II. BACKGROUND

A. Mercury's Perihelion Precession

Kepler (1609), in one of his laws, established that planets move in elliptical orbits around the Sun, with the Sun located at one of the foci. A few years later (1687), Newton formulated his Law of Universal Gravitation, which predicted this result with great precision. However, despite its accuracy, it failed to explain certain anomalies in planetary motion, such as Mercury's perihelion precession.

Mercury's perihelion precession refers to the slow shift in the position of the point in its orbit closest to the Sun, known as the perihelion. Planets orbit the Sun in ellipses, but these ellipses are not fixed in space; in Mercury's case (the most notable example), the ellipse slowly rotates around the Sun, as shown in Figure (1). Thus, the perihelion moves by a certain angle $2\pi\varepsilon$ along a circle with respect to its previous perihelion. [2].

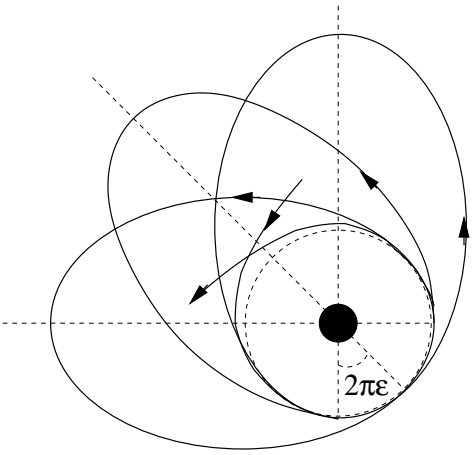

Figure 1: Illustration (exaggerated) of Mercury's perihelion precession. Image obtained from Janssen's book. [2, p. 174].

This anomaly was observed by the French astronomer Urbain Jean Joseph Le Verrier between 1845 and 1859 [3]. A large part of this irregularity can be explained by the interactions of other bodies in the solar system. However, there remains a residual effect of a shift of 43 arcseconds per century in the perihelion [3].

Various proposals emerged to explain this anomaly. One of them was by Le Verrier, who believed that this small precession could be caused by an unknown planet (named Vulcan) or by a cloud of asteroids, as other planetary anomalies (such as those of Uranus and Neptune) had been explained in a sim-

* computational physics
† e-mail: ricardoemf03@gmail.com

ilar manner [3]. However, this hypothesis was quickly discarded after performing calculations and finding that they did not match observations.
This and other anomalies motivated Albert Einstein (1915) to formulate his general theory of relativity, which expanded our understanding of gravity by describing it as the curvature of space-time caused by the presence of massive objects. Einstein's theory finally provided an explanation for this phenomenon.

In this work, we will avoid resorting to the complex analytical calculations of general relativity and instead approach the problem through relativistic corrections (derived from the theory of general relativity). Newton's inverse-square law of gravitation is modified by adding an extra term to the force, as shown in equation (1), taking $\alpha = 1.1 \times 10^{-8}\text{AU}^2$ (in astronomical units). This correction is very weak compared to Newtonian gravitational force since its effects only become significant in contexts where the gravitational intensity is extremely high, such as near massive bodies or in very close orbits around them (as is the case for Mercury). However, this small correction is sufficient to explain the phenomenon of Mercury's perihelion precession.

$$\vec{F} = -\frac{GM_S M_M}{r^2}\left(1+\frac{\alpha}{r^2}\right)\frac{\vec{r}}{r} \tag{1}$$

For practical purposes, we resort to numerical methods to model Mercury's orbit. These methods involve solving the equations of motion (5) using computational techniques so that the gravitational forces acting on the planet are represented with high precision. In particular, we use the second-order Runge-Kutta method to integrate the equations of motion, implemented in Python 3.

B. Second-Order Runge-Kutta Method

Like Euler's method, the Runge-Kutta method is used to solve first-order differential equations of the form (2). However, this method improves upon Euler's method by providing greater accuracy when calculating the value of a dependent variable.

$$\frac{dy}{dx} = f(x,y) \tag{2}$$

In Euler's method, the function y is approximated in each interval $[x_n, x_{n+1}]$ by a straight line such that

$$y_{n+1} = y_n + f(x_n, y_n)\Delta x$$

where the slope of the line is $f(x_n, y_n)$, which corresponds to the derivative y, or the slope of the tangent line at the point x_n.

This method is slightly modified to obtain the Runge-Kutta method. In this case, we consider the midpoint of the interval $[x_n, x_{n+1}]$, which corresponds to $x = x_n + \frac{1}{2}\Delta x$, where the value of y_{n+1} is calculated. This calculation is performed using Euler's method, resulting in the following relationship:

$$y_{n+1} = y_n + \frac{1}{2}k_1, \qquad k_1 = f(x_n, y_n)\Delta x$$

so that now, the slope at the midpoint $\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_1\right)$ is given by

$$\frac{k_2}{\Delta x} = f\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_1\right)$$

This slope is then used to generate the new corrected value of y_{n+1} corresponding to the point x_{n+1}, again using Euler's method [1].

$$y_{n+1} = y_n + k_2 \tag{3}$$

which explicitly corresponds to

$$y_{n+1} = y_n + f\left(x_n + \frac{1}{2}\Delta x, y_n + \frac{1}{2}k_1\right)\Delta x \tag{4}$$

This result shows us that, in fact, Euler's method is a particular case of the first-order Runge-Kutta method. Furthermore, if we continue with this idea of approximating y_{n+1} at intermediate points of the previous method, we will obtain higher-order Runge-Kutta methods, which become increasingly accurate but also more complex.

C. Implementation of the Runge-Kutta Method for Mercury's Orbital Equations of Motion

Having discussed the problem of Mercury's perihelion and the numerical method to be used, the next step is to deduce the Runge-Kutta algorithm from the force equation (1). It is important to note that the numerical method is specified for first-order differential equations. Although the force equation consists of a system of two second-order differential equations (one for x and another for y), we can convert it into a system of four first-order differential equations, as shown in equation (5). In this way, equation (1) is divided into four coupled first-order differential equations: two that describe Mercury's orbital positions and two that describe its velocities in the xy-plane.

$$\begin{aligned} \tfrac{dx}{dt} &= v_x; & \tfrac{dv_x}{dt} &= -\tfrac{GM_S M_M}{r^3}\left(1+\tfrac{\alpha}{r^2}\right)x \\ \tfrac{dy}{dt} &= v_y; & \tfrac{dv_y}{dt} &= -\tfrac{GM_S M_M}{r^3}\left(1+\tfrac{\alpha}{r^2}\right)y \end{aligned} \tag{5}$$

Now that we have determined the equations of motion, the next step is to apply the Runge-Kutta method (equation 3) to these equations, obtaining the following:

$$\begin{aligned} x_{n+1} &= x_n + k_2 & v^x_{n+1} &= v^x_n + k_4 \\ y_{n+1} &= y_n + k'_2 & v^y_{n+1} &= v^y_n + k'_4 \end{aligned} \tag{6}$$

con valores de $k's$

$$
\begin{aligned}
k_1 &= v_n^x \Delta t & k_3 &= -\frac{GM_S}{r_n^3}\left(1+\frac{\alpha}{r_n^2}\right) x_n \Delta t \\
k_2 &= \left(v_n^x + \tfrac{1}{2}k_3\right)\Delta t & k_4 &= -\frac{GM_S}{R_n^3}\left(1+\frac{\alpha}{R_n^2}\right)\left(x_n + \tfrac{1}{2}k_1\right)\Delta t \\
k_1' &= v_n^y \Delta t & k_3' &= -\frac{GM_S}{r_n^3}\left(1+\frac{\alpha}{r_n^2}\right) y_n \Delta t \\
k_2' &= \left(v_n^y + \tfrac{1}{2}k_3'\right)\Delta t & k_4' &= -\frac{GM_S}{R_n^3}\left(1+\frac{\alpha}{R_n^2}\right)\left(y_n + \tfrac{1}{2}k_1'\right)\Delta t \\
r_n &= \sqrt{x_n^2 + y_n^2} & R_n &= \sqrt{\left(x_n + \tfrac{1}{2}k_1\right)^2 + \left(y_n + \tfrac{1}{2}k_1'\right)^2}
\end{aligned}
$$

D. Application of the Numerical Method in Python 3

1. *Initial Conditions*

Python is a high-level, interpreted programming language known for its clear and readable syntax, which facilitates the rapid and efficient development of applications. For this reason, we use this language to implement our numerical method.

Before creating the program, it is necessary to define the initial conditions and values. The initial position is defined at the point farthest from the Sun, and the initial velocity is calculated using the conservation of angular momentum and energy between the initial point and the point $(0,b)$, where b is the semi-minor axis of the ellipse. These values are described in equation (7), where a is Mercury's semi-major axis and e is its eccentricity. Thus, ea is the distance between the Sun and the center of the ellipse. Meanwhile, the semi-minor axis in terms of the semi-major axis and the eccentricity is $b = a\sqrt{1-e^2}$.

$$
\begin{aligned}
x_0 &= (1+e)a, & y_0 &= 0 \\
v_0^x &= 0, & v_0^y &= \sqrt{\tfrac{GM_S}{a}\tfrac{1-e}{1+e}}
\end{aligned}
\tag{7}
$$

Since a computational numerical method is used, an appropriate unit system was chosen for handling the data. Astronomical units (1 AU = 149,597,828.68 km, the average distance from Earth to the Sun) were used for distances, and Earth years were used for time. These values for Mercury and the Sun, as well as the number of iterations in the numerical method and the time step, are as follows:

$$
\begin{aligned}
&a = 0.39 \text{ UA} \\
&e = 0.206 \\
&GM_S = 4\pi^2 \text{ UA}^3/\text{year}^2 \\
&\Delta t = 0.0001 \\
&N = 20000
\end{aligned}
$$

Finally, equation (8) was used to describe the distance between Mercury and the Sun (with θ being the angle between the horizontal and Mercury's position vector), and equation (9) was used for its velocity. In equation (8), we note that when $\theta = 0$, the planet is at its farthest point, whereas when $\theta = 180^\circ$, it is at its closest point to the Sun.

$$
r' = \frac{(1-e^2)a}{1-e\cos\theta} \tag{8}
$$

$$
\frac{dr'}{dt} = \frac{xv_x + yv_y}{r'} \tag{9}
$$

Note the use of r' (primed) instead of r to avoid confusion with the magnitude of the position vector relative to the origin of the ellipse.

Since the actual value of α is very small, Mercury's perihelion precession rate is also very small and cannot be measured numerically over a limited time period. For this reason, we use the α values described in Table (I) and, based on them, calculate Mercury's perihelion precession rate. Using the precession rates for different α values, we extrapolate the data to find the actual precession rate corresponding to $\alpha = 1.1 \times 10^{-8}\text{AU}^2$.

Table I: Values of α used in the simulation (in UA^2)

α	0.0008	0.001	0.002	0.004

2. *Implementation in Python*

The first step is to create a function `RungeKutta2(···)`, which takes as parameters the initial conditions (7), the number of iterations, the step size, and the value of α to be used. This function returns the position and velocity data for Mercury's orbit at each time step over a total time range of $T = N\Delta t = 2$ years. It also provides the values of (8) and (9) at each time step.

From this data, we can calculate the perihelion precession rate of Mercury for a given α. This is done considering the following: The value of dr'/dt equals zero each time Mercury passes through its farthest and closest points to the Sun, the aphelion and perihelion, respectively. With this in mind, we can trace an angle θ_p between the horizontal axis (x-axis) and Mercury's position vector at each instant where $dr'/dt = 0$. Assuming the aphelion is initially on the positive x-axis, this will give angles of $\theta_p \geq 180^\circ$ and $\theta_p \leq 90^\circ$, corresponding to the perihelion and aphelion, respectively.

We filter the angles $\theta_p \geq 180^\circ$, which correspond to the perihelion with the horizontal, and record the time instants when they occur. These data are then fitted to a linear model of the form $\theta_p(t) = mt + c$, where m and c are the fitting constants. Finally, $d\theta_p/dt = m$ represents the perihelion precession rate of Mercury (given in degrees/year) for the corresponding value of α.

This process is repeated for all the values in Table (I). Once the precession rate has been obtained for each α value from the table, these data are extrapolated (also through a linear fit) to determine the actual perihelion precession rate of Mercury corresponding to $\alpha = 1.1 \times 10^{-8}$ AU^2.

III. RESULTS

A. Calculation of the Precession Rate for $\alpha = 0.0008$ AU^2

In the previous section, we discussed the approach to addressing the problem. In this section, we present the results obtained for an initial value of α. After integrating the equations of motion using the numerical method, we plotted Mercury's positions (Figure 2) to visualize its trajectory. As expected, we observed an elliptical trajectory with a slight displacement of the ellipse. This supports the predictions made by Kepler.

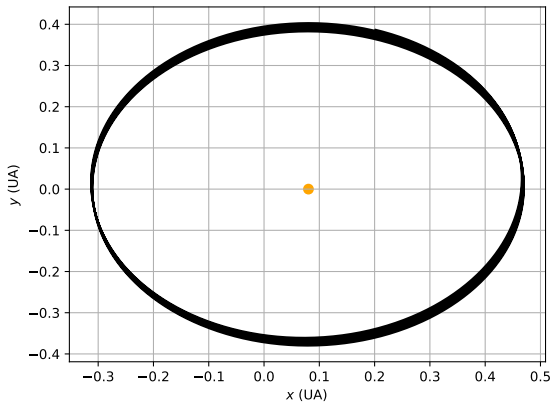

Figure 2: Mercury's orbit around the Sun for $\alpha = 0.0008$

According to the setup, the distances between Mercury and the Sun must exhibit a maximum and a minimum point, corresponding to the aphelion and perihelion, respectively. This is evident in Figure (3), where we clearly observe an oscillatory behavior. The extreme values are approximately $(1+e)a \approx 0.47$ AU (70 million km) for the farthest point and $(1-e)a \approx 0.31$ AU (46 million km) for the closest point. These data verify the established initial conditions as well as the proper functioning of the numerical method used. The observed extreme distances align with the expected values for the aphelion and perihelion, confirming the validity of the approach taken.

Since the objective is to find the perihelion precession angle, the maximum and minimum points in Figure (3) correspond to the zeros of the graph in Figure (4), obtained based on equation (9). This allows us to determine the times when Mercury passes through its farthest and closest points. For each of these times, we calculate the angle θ_p between the horizontal axis and Mercury's position vector at that instant. Once this is done, we filter the angles $\theta_p \geq 180°$, which represent the perihelion deviation from the horizontal. These results are presented in Figure (5), where a linear behavior is observed. The linear fit to the points is as follows:

$$\theta_p(t) = (8.39 \pm 0.04)t + (180.02 \pm 0.05) \tag{10}$$

Therefore, the perihelion precession rate of Mercury for a value of $\alpha = 0.0008$ AU^2 is:

$$\frac{d\theta_p}{dt} = 8.39 \pm 0.04 \ \frac{\text{degrees}}{\text{year}} \tag{11}$$

The next step is to determine the precession rate for the other α values listed in Table (I).

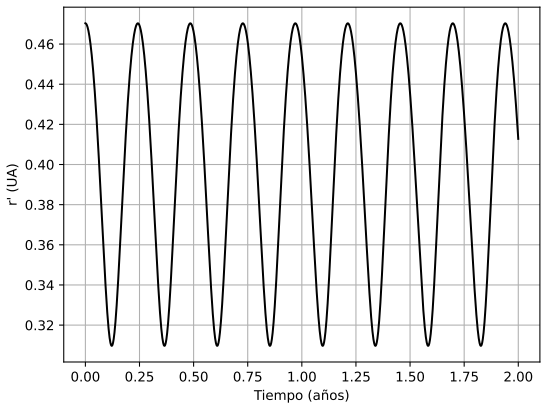

Figure 3: Distance between Mercury and the Sun as a function of time.

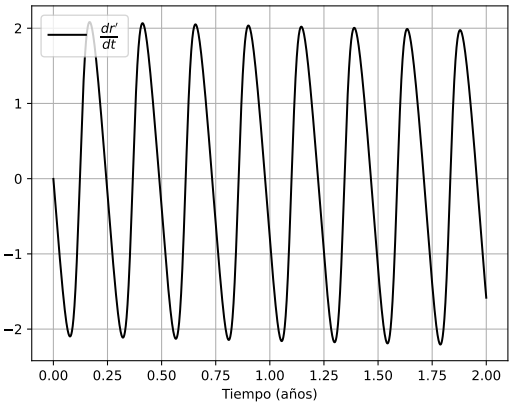

Figure 4: Rate of change of distances between Mercury and the Sun with respect to time.

B. Calculation of Mercury's Actual Perihelion Precession Rate

In the previous section, we found that the perihelion precession rate is $d\theta_p/dt = 8.39 \pm 0.04$ (degrees/year) for a value of $\alpha = 0.0008$ AU^2. In this section, we repeated the procedure from the previous section for the other α values listed in Table (I). Following this process, we determined the precession rates described in Table (II). For example, for a value of $\alpha = 0.004$ AU^2, the corresponding precession rate is 45.378 ± 0.044 degrees/year.

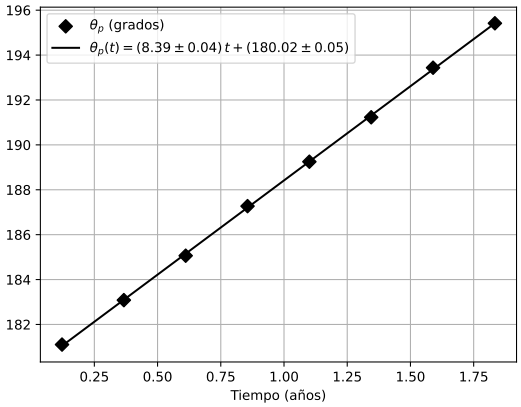

Figure 5: Angle of deviation of the perihelion with respect to the horizontal as a function of time.

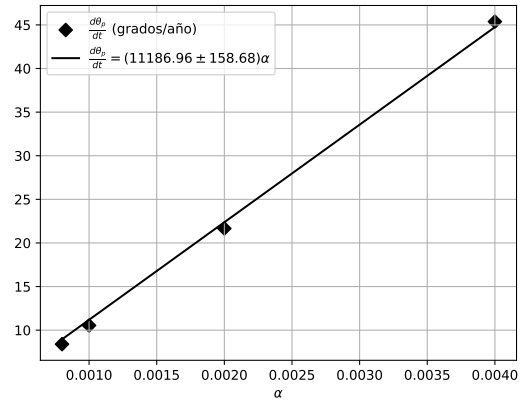

Figure 6: Mercury's perihelion precession rate for each value of α.

Figure (6) shows the values from Table (II) plotted on a graph. These values also exhibit a linear behavior as a function of α. In other words, the precession rate is proportional to the value of α. Performing a linear fit, we obtain the following:

$$\frac{d\theta_p}{dt} = (11186.96 \pm 158.68)\alpha \tag{12}$$

Substituting the actual value $\alpha = 1.1 \times 10^{-8}$ AU^2 into equation (12), converting degrees to arcseconds and years to centuries, we obtain a precession value of:

$$\frac{d\theta_p}{dt} = 44.3 \pm 0.6 \text{ second of arc/century} \tag{13}$$

Table II: Mercury perihelion precession rate for each value of α

α (UA^2)	$d\theta_p/dt$ (grados/año)
0.0008	8.395 ± 0.043
0.001	10.540 ± 0.041
0.002	21.659 ± 0.046
0.004	45.378 ± 0.044

IV. CONCLUSIONS

As stated by Kepler in his first law, Mercury's orbit follows an ellipse with the Sun at one of its foci. This is evident in Figure (2), which shows the path of its orbit. Furthermore, it is observed that due to the relativistic correction, this ellipse does not remain fixed in space but rotates slowly around the Sun. This behavior is precisely what is expected according to the theory of general relativity.

The results have shown that, as the relativistic correction parameter α is varied, the perihelion precession rate increases linearly. We have determined that Mercury's actual perihelion precession rate is 44.3 ± 0.6 arcseconds per century (equation 13), which is in good agreement with the observed value of 43 arcseconds per century, as also predicted by Einstein's theory of general relativity.

This also demonstrates that the second-order Runge-Kutta method is an effective tool for numerically simulating Mercury's motion and estimating the precession of its perihelion, which cannot be explained by Newtonian mechanics.

Finally, we emphasize the importance of relativistic corrections in the analysis of planetary motion and show how even small adjustments to gravitational forces can have a significant effect on long-term orbital behavior. Additionally, the use of computational tools provided us with a practical and precise approach to addressing a problem that historically posed a challenge for physics.

V. ANEXO

The code used in this simulation as well as other details can be found in the following link.

[1] Juan Diego Chang. Física computacional. Notas de clase en Física computacional, Escuela de Ciencias Físicas y Matemáticas, USAC, 2024. Accedido en: Noviembre 2024.

[2] Bert Janssen. *Teoría de la Relatividad General*. Granada, España, 2013. Dpto de Física Teórica y del Cosmos, Edificio Mecenas, Campus de Fuente Nueva, 18071 Granada, España. Contacto: bjanssen@ugr.es.

[3] Jesús López Zavala, Elizabeth Fraire Bonilla, Juan Diego Pérez Ruiz, Simón Rodríguez Rodríguez, and Jaime Burgos García. Precesión del perihelio de mercurio, 2023.