Orbital Integration of Earth and Jupiter.

Name: Liam Artair Peacock

Module: PY2105

Location: Lab 201, Department of Physics, UCC

Abstract:

The objective of this experiment was to simulate the orbits of the Earth and Jupiter around the Sun. As well as this, to calculate the evolution of the angular momentum of the Earth and Jupiter. This report intends to verify theoretical expressions for the components of the orbits of Earth and Jupiter around the sun, and the evolution of the angular momentum of the Earth and Jupiter with time.

This was done by writing programs in Python to answer the questions as written on the final take home exam sheet.

The experiment was partially successful as the orbits of Earth and Jupiter were simulated correctly, however, the momentums calculated were not accurately calculated. A closer look at the programs written must be conducted to improve the results obtained in the plots.

Table of Contents Page

Introduction:

A good understanding of the orbits of planets is important, particularly the understanding of our planet the Earth is pivotal for numerous technologies to function. Some examples include the Global Positioning System (GPS) satellites, and the International Space Station (ISS). The angular momentum of planets can be affected by other planet's influence. Jupiter is one such planet which affects the Earth's angular momentum.

The following experiment was conducted to simulate the orbits of the Earth and Jupiter around the Sun. As well as this, to calculate the evolution of the angular momentum of the Earth and Jupiter. This report intends to verify theoretical expressions for the components of the orbits of Earth and Jupiter around the sun, and the evolution of the angular momentum of the Earth and Jupiter with time.

Experimental Procedure:

Part A:

For the setup of this derivation, see [fig.1]. From Newton's Law of Gravitation, we have the following formula:

$$F_G = \frac{GM_sM_E}{r^2} \quad (1)$$

Where $G = 6.6743 \times 10^{-11}\ N\,m^2\,kg^{-1}$ is the gravitational constant, the stellar mass, in this case the sun is $M_s = 1.9891 \times 10^{30}\ kg$, $M_E = 5.972 \times 10^{24}\ kg$ is the mass of the Earth, and $r = 1\ AU$ is the radius of the Earth in Astronomical Units.

Applying Newton's 2nd Law, we arrive at the following equations:

$$\frac{d^2x}{dt^2} = \frac{F_G, x}{M_E} \quad (2)$$

$$\frac{d^2y}{dt^2} = \frac{F_G, y}{M_E} \quad (3)$$

These 2nd order differential equations are the x and y components of the acceleration of the Earth. Simplifying (2) and (3) give:

$$\frac{d^2x}{dt^2} = \frac{GM_s\cos(\theta)}{r^2} \quad (3)$$

$$\frac{d^2y}{dt^2} = \frac{GM_s\sin(\theta)}{r^2} \quad (4)$$

Where θ is the angle. One can use this angle in its current form and set the initial conditions in terms of π. However, this angle can also be converted into positions if one wishes to deal with positions instead using the following expressions for the x and y components the angles for each are:

$$\cos(\theta) = \frac{x}{r} \quad (5)$$

$$\sin(\theta) = \frac{y}{r} \quad (6)$$

Where x and y are the x and y positions of the Earth. Substituting (5) and (6) into (3) and (4) respectively, we arrive at the following expressions:

$$\frac{d^2x}{dt^2} = \frac{GM_sx}{r^3} \quad (7)$$

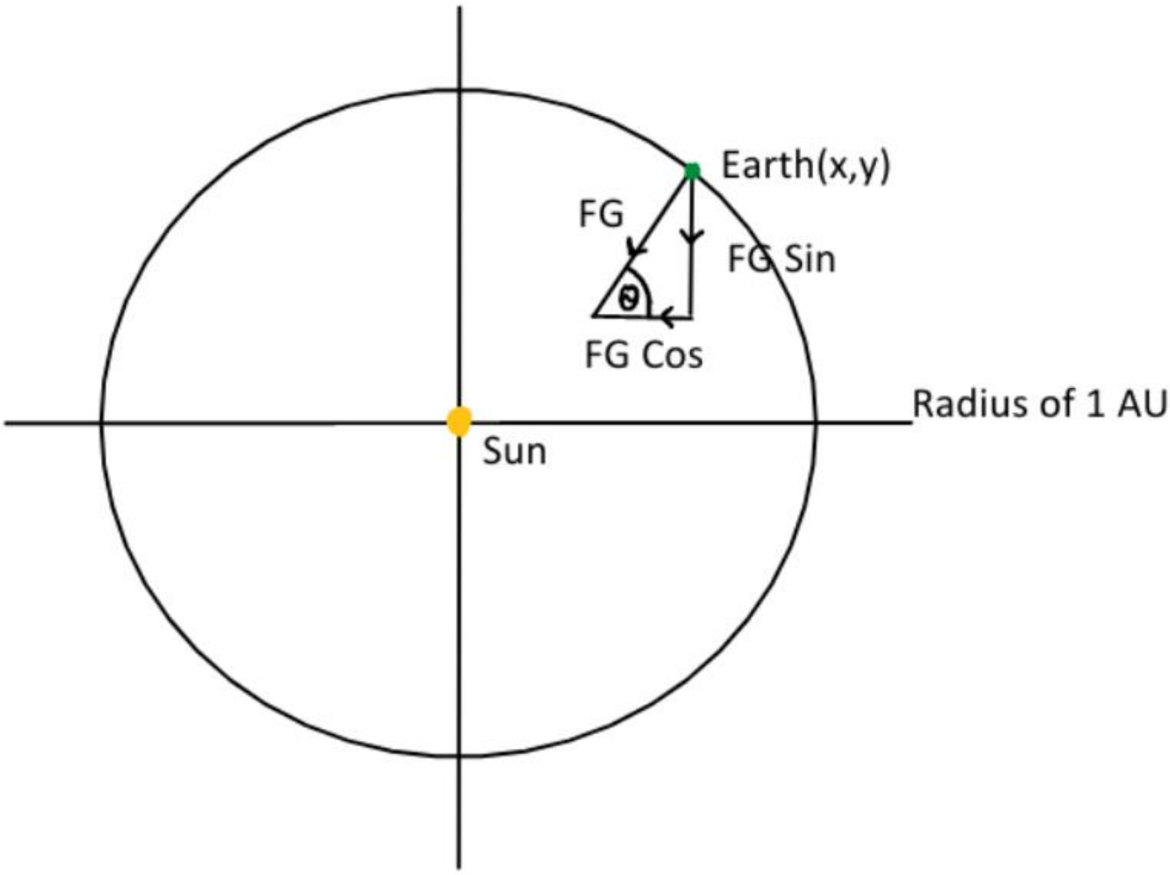
$$\frac{d^2y}{dt^2} = \frac{GM_sy}{r^3} \quad (8)$$

If we let $\alpha = \frac{dx}{dt}$, and $\delta = \frac{dy}{dt}$, then differentiating α and δ with respect to t will let us convert our 2nd order differential equation into a 1st order differential equation. This then gives:

$$\frac{d\alpha}{dt} = -\frac{GM_s x}{r^3} \quad (9)$$

$$\frac{d\delta}{dt} = -\frac{GM_s y}{r^3} \quad (10)$$

The minus sign in the expression arises as the forces are acting toward the Sun and as such are by convention negative, see [fig.1]



[Fig.1] Diagram of the setup in Part A. The orbit of Earth is assumed to be a unit circle as the radius is 1 AU. The components of F_G show the negative direction toward the Sun indicated by the arrows. The Earth is represented with its pair of x and y coordinates. The Sun is located at the centre $(0, 0)$.

Part B:

For this part, the objective was to simulate the orbit of the Earth using the Euler method. To start, the values for G, M_S, M_E, and r are initialised as described in Part A. The max time that the simulation was run for was set to 10 years, and a list named dt contained the three timesteps used, $0.1, 0.01,$ and 0.001.

For the initial conditions, this algorithm used the position setup described in equations (9) and (10) so the initial x and y position were set at $1\ AU\ and\ 0\ AU$ respectively. The initial velocity in the y direction was calculated using the following equation:

$$v_K = \sqrt{\frac{GM_s}{r^2}} \quad (11)$$

Where v_K is the Keplerian velocity. For the x velocity, as the x position was at the max distance from the Sun, setting the x velocity to be anything other than 0 would cause the Earth to travel further than radius.

To implement the Euler method, two range based for loops were used. The first for loop would initialise the first elements of the lists to hold the x and y positions as well as the x and y velocities. This loop also plotted the elements of the lists after the second loop had concluded. The second loop would first calculate the distance between the Earth and the Sun:

$$r = \sqrt{{x_0}^2 + {y_0}^2} \quad (12)$$

Where x_0 and y_0 are the initial x and y positions respectively. If one assumes the orbit of Earth is perfectly circular, there would be no need to update this position.

Then, the x and y velocities are updated using:

$$vx_{i+1} = vx_0 - \left(\frac{GM_s x_0}{r^3}\right)(h) \quad (13)$$

$$vy_{i+1} = vy_0 - \left(\frac{GM_s y_0}{r^3}\right)(h) \quad (14)$$

Where vx_0 and vy_0 are the initial x and y velocities, vx_{i+1} and vy_{i+1} are the updated x and y velocities, and h is the current timestep.

The positions were then updated using the following formula:

$$x_{i+1} = x_0 - (vx_{i+1})(h) \quad (15)$$

$$y_{i+1} = y_0 - (vx_{i+1})(h) \quad (16)$$

Where x_{i+1} and y_{i+1} are the updated x and y positions.

After both the loops had concluded, the title, x and y labels, legend, and grid were then added to the graphs and displayed.

Part C:

As in Part B, firstly the same constants and initial conditions were defined and set. After this, a function called "AccelerationGravity" was defined to calculate and return the acceleration. Then, like Part B, the Euler method was implemented; however, this time it was implemented inside a function named "SimulateOrbit".

Inside this function, the time for the plots was calculated using the np.linspace() function from the numpy library. This function generates evenly spaced values for time between the interval of the max time the simulation runs for, and the number of steps used.

Two arrays were defined (r and v) to hold the position and velocities in vector form. After this, a for loop would call "AccelerationGravity", and then use the following expression to update the velocity vector:

$$v_1 = v_0 + (a_0)(h) \quad (17)$$

Where v_0 is the initial velocity vector, a_0 is the first acceleration vector calculated by "AccelerationGravity", and v_1 is the updated velocity vector.

Using this updated vector, the position vector was then updated as follows:

$$r_1 = r_0 + (v_1)(h) \quad (18)$$

Where r_0 and r_1 are the initial and updated position vectors respectively.

The linear momentum was then calculated using the following equation:

$$p = M_E v_1 \quad (19)$$

Where M_E is the mass of the Earth.

Finally, the loop calculated the angular momentum and then added it to a list. The angular momentum was calculated using the function np.cross() from the numpy library:

$$L = np.cross(r_1, p) \quad (20)$$

This function computes the cross product of the position vector and the linear momentum.

After this loop concluded, "SimulateOrbit" returned the time and momentum lists. To plot the graphs, a for loop was used to assign the time and angular momentum lists to the returned values of the "SimulateOrbit" function.

Part D:

Orbit:

As described in the previous parts, the constants were defined. In this part, a function was defined to calculate the position and velocity as well as update them (r). A second function was then defined named "RungeKuttaSecondOrder" which calculated $k1$ and $k2$ of the Runge-Kutta 2nd order method using the following two equations:

$$k1 = \big(f(t,r)\big)(h) \quad (21)$$

$$k1 = \big(f(t+h, r+k1)\big)(h) \quad (22)$$

Where t is time, h is the current time step, $f(t,r)$ is the function used to calculate the x and y position, as well as the x and y velocity, (see Part D Orbits python file). This function then returns the value of the following expression:

$$r + 0.5(k1 + k2) \quad (23)$$

Where the 0.5 is a result of using Heun's method which is letting $a1 = 0.5$ [1], and r is the universal array that at certain indices contains the specific quantities of x and y position as well as the x and y velocities.

A for loop then sets the initial time and calculates the x and y positions and updates the lists containing the x and y positions so they can be plotted later. The plotting is taken care of using a second for loop that plots the elements of the lists mentioned in this paragraph, as well as a yellow-coloured dot to represent the sun.

Momentum:

As per the previous parts, the constants were defined. As described in Part C, the function "SimulateOrbit" was defined under the new name of "SimulateOrbit_euler". As well as this function, a new function named "SimulateOrbit_rk2" was defined to implement the Runge-Kutta 2nd order method using Heun's method.

"SimulateOrbit_rk2", similarly to "SimulateOrbit_euler", used np.linspace() to generate evenly spaced values for time between the interval of the max time the simulation runs for, and the number of steps used.

The position and velocity are once again stored as two vectors using arrays (r and v). A for loop was then used that calculates the angular momentum as follows:

$$L = np.cross\big(r, (M_s v)\big) \quad (24)$$

Where instead of calculating the linear momentum first as per equation (19), the angular momentum is calculated in one step my taking the linear momentum as an input to the function. When using Runge-Kutta in this context, $k1$ and $k2$ must be calculated twice in the form of $k1_r$, $k1_v$, $k2_r$, and $k1_v$ respectively; the equations for which are as follows:

$$k1_v = AccelerationGravity(r) \ (25)$$

$$k1_r = v \ \ (26)$$

$$k2_v = AccelerationGravity(r + (k1_r h)) \ \ (27)$$

$$k2_r = v + (k1_v h) \ \ (28)$$

Where “AccelerationGravity” is the same function that was defined in Part C, h is the timestep and the subscript r and v correspond to the position and velocity respectively.

After this, “SimulateOrbit_rk2 updates the values for the position and velocity vectors as follows:

$$r_1 = r_0 + \left(\frac{k1_r}{2} + \frac{k2_r}{2}\right)(h) \ \ (29)$$

$$v_1 = v_0 + \left(\frac{k1_v}{2} + \frac{k2_v}{2}\right)(h) \ \ (30)$$

Where $r_0$and v_0 are the initial position and velocity, and $r_1$and v_1 are the updated position and velocity respectively.

Finally, the plots were created using a for loop which called “SimulateOrbit_rk2” and “SimulateOrbit_euler” and added the values returned to separate lists. These values were then used in the plots. The plots were separated by subplots for each timestep.

Part E:

The setup for this part is almost identical to Part A, the difference being that Jupiter is also orbiting the Sun, see [fig.2]. The force between Earth and Jupiter (F_{EJ}) is described as follows:

$$F_{EJ} = \frac{GM_E M_J}{{r_{EJ}}^2} \ \ (31)$$

Where $M_J = 1.898 \times 10^{27}\ kg$ is the mass of Jupiter, and r_{EJ} is the distance between the Earth and Jupiter. For the orbital radius of Jupiter, (r_J) it was taken to be a value of $5.2\ AU$. In the same manner as Part A, Newton’s 2nd Law can be applied:

$$\frac{d^2x}{dt^2} = \frac{F_{EJ}, x}{M_E} \ \ (32)$$

$$\frac{d^2y}{dt^2} = \frac{F_{EJ}, y}{M_E} \ \ (33)$$

Simplifying the two equations above:

$$\frac{d^2x}{dt^2} = \frac{GM_J \cos{(\theta_{EJ})}}{{r_{EJ}}^2} \ \ (34)$$

$$\frac{d^2y}{dt^2} = \frac{GM_J \sin(\theta_{EJ})}{{r_{EJ}}^2} \quad (35)$$

Where θ_{EJ} is the angle between Earth and Jupiter. The above two equations were the ones used in the orbit procedure discussed in this part. However, for the momentum portion of this part, one can continue this derivation as follows to represent the angles in terms of position.

Using equations (5) and (6) where $r = r_{EJ}$, this results in the two equations below:

$$\frac{d^2x}{dt^2} = \frac{GM_J\,(x_E - x_J)}{{r_{EJ}}^3} \quad (36)$$

$$\frac{d^2y}{dt^2} = \frac{GM_J\,(y_E - y_J)}{{r_{EJ}}^3} \quad (37)$$

Where x and y are the positions of Jupiter. Letting $\alpha = \frac{dx}{dt}$, and $\delta = \frac{dy}{dt}$, then differentiating α and δ with respect to t will let us convert our 2nd order differential equation into a 1st order differential equation. This then gives:

$$\frac{d\alpha}{dt} = -\frac{GM_J\,(x_E - x_J)}{{r_{EJ}}^3} \quad (38)$$

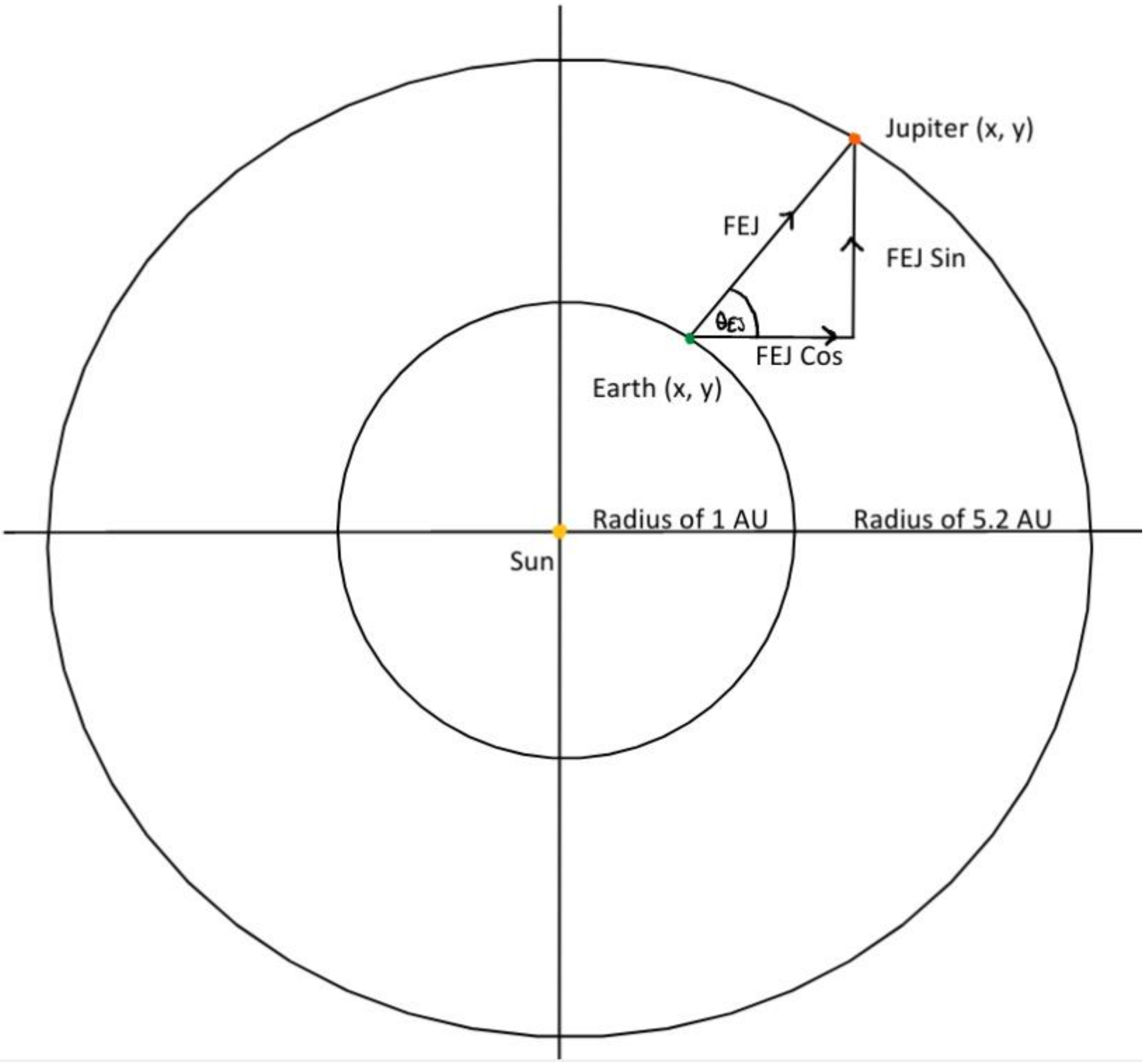
$$\frac{d\delta}{dt} = -\frac{GM_J\,(y_E - y_J)}{{r_{EJ}}^3} \quad (39)$$

The minus sign in the expression arises as the forces are acting toward Jupiter and as such are by convention negative, see [fig.2]

Finally, combining equations (9) and (10) with (38) and (39), we arrive at the final expression for x and y components of the velocity of Earth:

$$\frac{d\mathrm{v}_{Ex}}{dt} = -\frac{GM_s\,x_E}{{r_E}^3} - \frac{GM_J\,(x_E - x_J)}{{r_{EJ}}^3} \quad (40)$$

$$\frac{d\mathrm{v}_{Ey}}{dt} = -\frac{GM_s\,y_E}{{r_E}^3} - \frac{GM_J\,(y_E - y_J)}{{r_{EJ}}^3} \quad (41)$$



[Fig.2] Diagram of the physical setup for Part E. Similarly, to the setup in Part A see [fig.1], the Earth orbits the Sun; however, Jupiter is also orbiting the Sun. As Jupiter's gravitational force is greater than Earth, Earth is also being attracted to Jupiter as well, as indicated by the arrows on the components.

Orbit:

The values for the constants G, M_S, M_E,M_J, r_E, and r_J were initialised as described in Part B and per the final take home exam sheet. Similarly, to Part D, a function named "RungeKutta" was defined to calculate the orbits using the Runge-Kutta 2nd order method employing Heun's method.

Unlike the method described in equations (9) and (10) of Part A, this part uses the method of the angles instead of position.

This function would first create lists to hold the x and y positions of the Earth and Jupiter respectively. A for loop would then calculate the x and y position of the earth using the following equations and respectively then add these values to the respective lists:

$$x_E = r_E \cos(\theta_E) \quad (42)$$

$$y_E = r_E \sin(\theta_E) \quad (43)$$

$$x_J = r_J \cos(\theta_J) \quad (44)$$

$$y_J = r_J \sin(\theta_J) \quad (45)$$

Next, the distance from Earth to Jupiter (r_{EJ}) was then calculated via the following equation:

$$r_{EJ} = \sqrt{(x_E - x_J)^2 + (y_E - y_J)^2} \quad (46)$$

Where x_E and y_E are the x and y positions of the Earth, and x_J and y_J are the x and y positions of Jupiter respectively.

After this, the acceleration between the Sun and Jupiter, between the Sun and Earth, and between Earth and Jupiter were calculated as follows:

$$a_{SJ} = \frac{GM_s}{{r_J}^2} \quad (47)$$

$$a_{SE} = \frac{GM_s}{{r_E}^2} \quad (48)$$

$$a_{EJ} = \frac{GM_s}{{r_{EJ}}^2} \quad (49)$$

The accelerations in the x and y directions of for the Earth and Jupiter were then calculated using the following equations:

$$a_{xE} = -(a_{SE})(\cos(\theta_E)) + (a_{EJ})(\cos(\theta_J)) \quad (50)$$

$$a_{yE} = -(a_{SE})(\sin(\theta_E)) + (a_{EJ})(\sin(\theta_J)) \quad (51)$$

$$a_{xJ} = -(a_{SE})(\cos(\theta_j)) + (a_{EJ})(\cos(\theta_E)) \quad (52)$$

$$a_{yJ} = -(a_{SE})(\sin(\theta_J)) + (a_{EJ})(\sin(\theta_E)) \quad (53)$$

Where θ_E and θ_J are the angles of the Earth and Jupiter respectively.

Then, the $k1$ values for the angle and velocity of Earth and Jupiter respectively were calculated:

$$k1_{\theta_E} = (h)(v_E) \quad (54)$$

$$k1_{v_E} = (h)(a_{xE}) \quad (55)$$

$$k1_{\theta_J} = (h)(v_J) \quad (56)$$

$$k1_{V_J} = (h)(a_{xJ}) \quad (57)$$

Where $k1_{\theta_E}$ and $k1_{\theta_J}$are the $k1$ values for the angles of Earth and Jupiter, and $k1_{v_E}$ and $k1_{V_J}$ are the $k1$ values for the velocities of Earth and Jupiter respectively.

For the $k2$ values for the angle and velocity of Earth and Jupiter, they were calculated as follows:

$$k2_{\theta_E} = (h)\left(v_E + \frac{k1_{v_E}}{2}\right) \ (58)$$

$$k2_{v_E} = (h)\left(a_{xE} + \frac{k1_{v_E}}{2}\right) \ (59)$$

$$k2_{\theta_J} = (h)\left(v_J + \frac{k1_{v_J}}{2}\right) \ (60)$$

$$k2_{v_E} = (h)\left(a_{xJ} + \frac{k1_{v_J}}{2}\right) \ (61)$$

Finally, the updated values of the angles of Earth and Jupiter as well as the velocities of Earth and Jupiter:

$$\theta_{E_1} = \theta_{E_0} + \ k2_{\theta_E} \ (62)$$

$$v_{E1} = v_E + \ k2_{v_E} \ (63)$$

$$\theta_{J_1} = \theta_{E_0} + \ k2_{\theta_E} \ (64)$$

$$v_{J1} = v_J + \ k2_{v_J} \ (65)$$

Where $\theta_{E_1}$and θ_{J_1} are the updated angles of Earth and Jupiter, and v_{E1} and v_{J1} are the updated velocities of Earth and Jupiter respectively.

The function "Runge-Kutta" returns these updated values and the second for loop enumerates through the time steps and plots the points in the lists of the x and y values of the Earth and the x and y values of Jupiter respectively.

Momentum:

Similarly, to the Orbit section of this part, all constants were defined in the same manner and a function was defined to implement the Runge-Kutta method named "RungeKutta". For the initial conditions, the velocities of Earth and Jupiter were set as their respective Keplerian velocities (see equation (11) for the Earth; for Jupiter, the r value was replaced with r_J) The initial angles for the Earth and Jupiter were 0 radians and π radians respectively so the two planets would start on the opposite sides of the Sun.

The x and y positions of the Earth and Jupiter were the calculated as follows:

$$x_E = (r_E)(\cos(\theta_E)) \ \ (66)$$

$$y_E = (r_E)(\sin(\theta_E)) \ \ (67)$$

$$x_J = (r_E)\big(\cos(\theta_J)\big) \ \ (68)$$

$$Y_J = (r_E)(\sin(\theta_E)) \ \ (69)$$

After this, the distance between Earth and Jupiter was then calculated using equation (42).

Then, the accelerations of Earth and Jupiter were calculated using the following expressions:

$$a_E = \left(\frac{GM_J}{{r_{EJ}}^3}\right)(r_E) \quad (70)$$

$$a_J = \left(\frac{GM_E}{{r_{EJ}}^3}\right)(r_J) \quad (71)$$

Following this, the velocities of Earth and Jupiter were updated as follows:

$$v_E = v_{E0} + (h)(a_E) \quad (72)$$

$$v_J = v_{J0} + (h)(a_J) \quad (73)$$

Where v_{E0} and v_{J0} are the initial velocities of the Earth and Jupiter.

The angles of Earth and Jupiter were updated before calculating the angular momentum as such:

$$\theta_{E+1} = \theta_{E0} + \frac{h + v_E}{r_E} \quad (74)$$

$$\theta_{J+1} = \theta_{J0} + \frac{h + v_J}{r_J} \quad (75)$$

Where θ_{E0} and θ_{J0} are the initial angles of the Earth and Jupiter.

The angular momentum of the Earth and Jupiter was then calculated as follows:

$$L_E = M_E r_E v_E \quad (76)$$

$$L_J = M_J r_J v_J \quad (77)$$

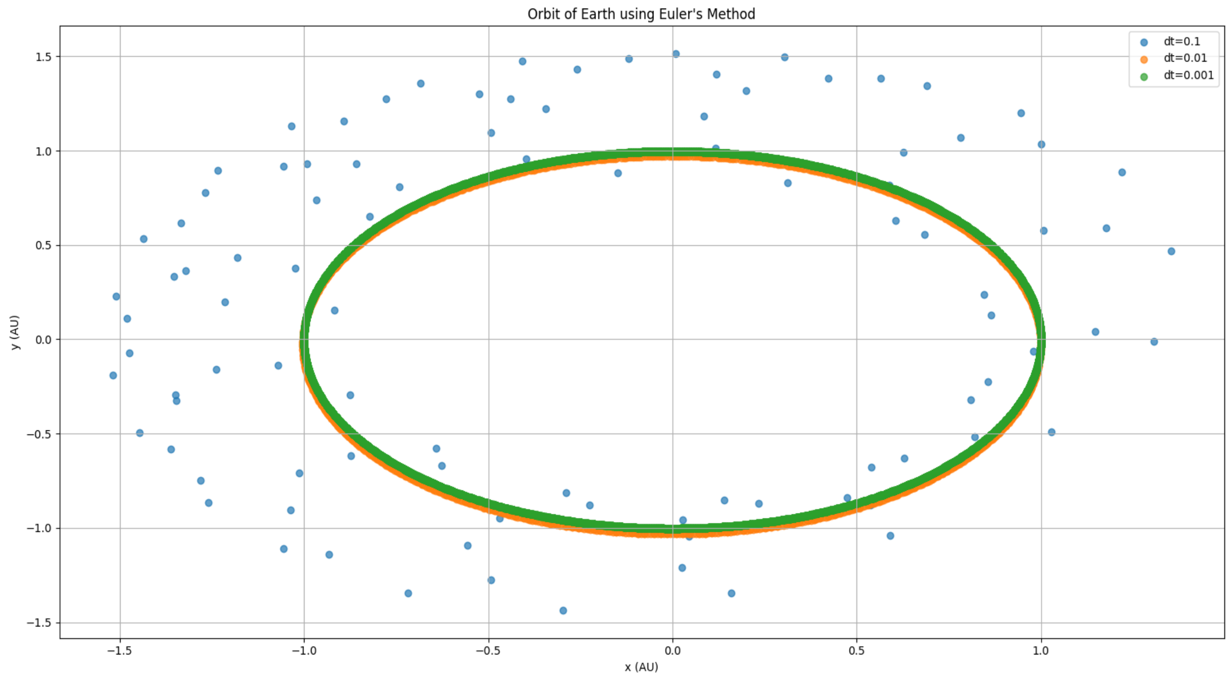
“RungeKutta” then returns the L_E, L_J, and T which was calculated by multiplying the timestep in use by the number of steps.

A for loop was then used to create two subplots of the angular momentum values versus time.

Results:

Part B:

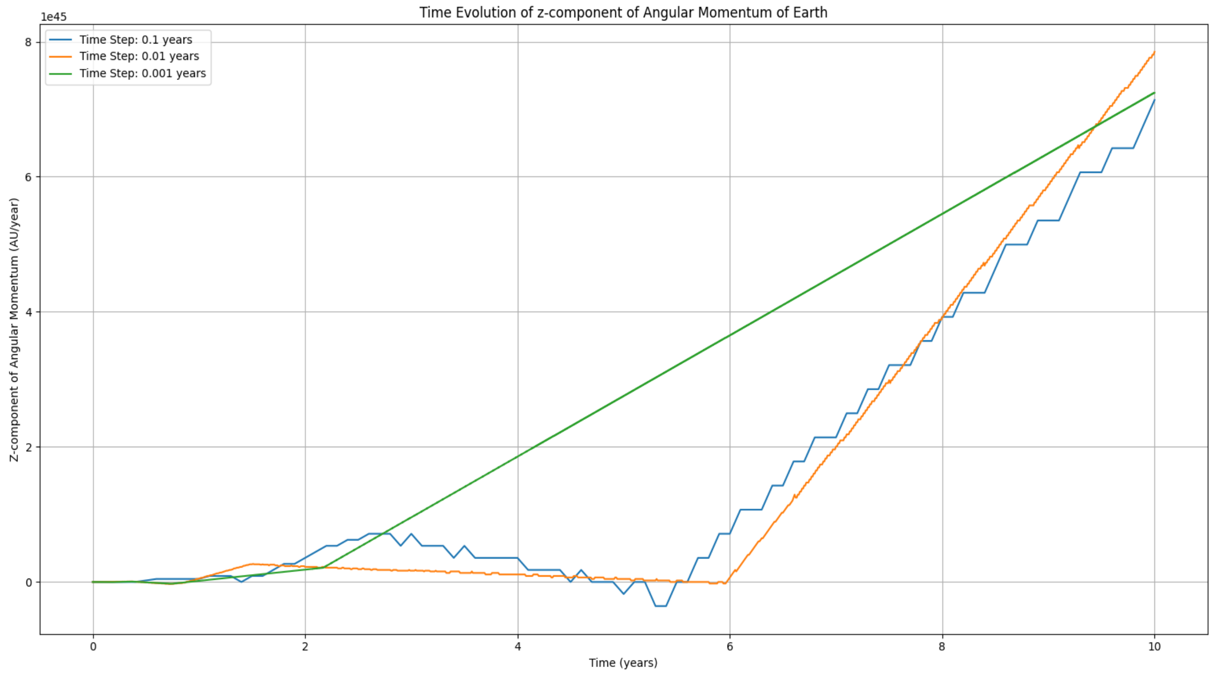
As can be seen in [fig.3], as one lowers the timestep, the accuracy of the plotted orbit improves significantly. The largest timestep, which corresponds to the blue dots, barely represents a coherent orbit, while the green curve properly shoes an orbit as it’s the smallest timestep used.



[Fig.3] Plot of the orbit of the Earth around the Sun using Euler's method. The blue dots correspond to the largest time step, the orange curve corresponds to the middle time step, and the green curve corresponds to the smallest time step. As expected, the largest timestep produces the least accurate plot.

Part C:

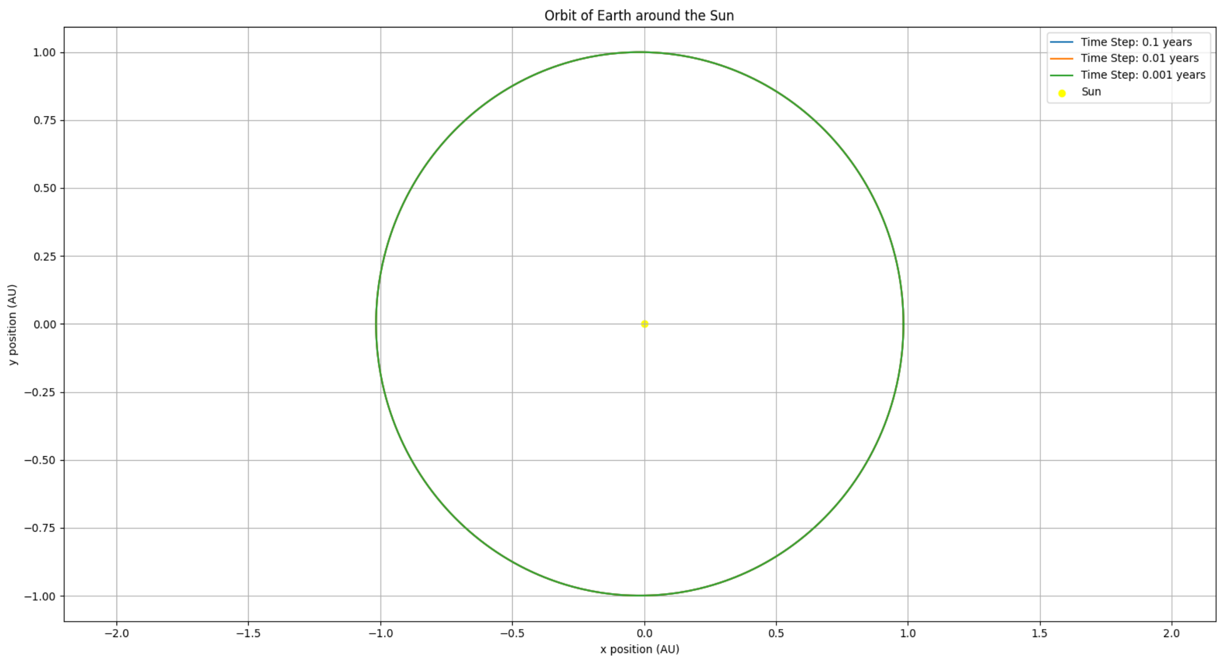
When viewing the plot in [fig.4], one can clearly see that the z- component of the momentum of the Earth fluctuates as it does, but then increases linearly after 2 years. As mentioned, the Earth's angular momentum does fluctuate, but not to the extent that the plot seems to suggest.



[Fig.4] Plot of the evolution of the z-component of the angular momentum with time in years. As can be seen, the angular momentum fluctuates as time increases. This is concerning as there is fluctuations in the angular momentum, the angular momentum of the z-component remains mostly constant.

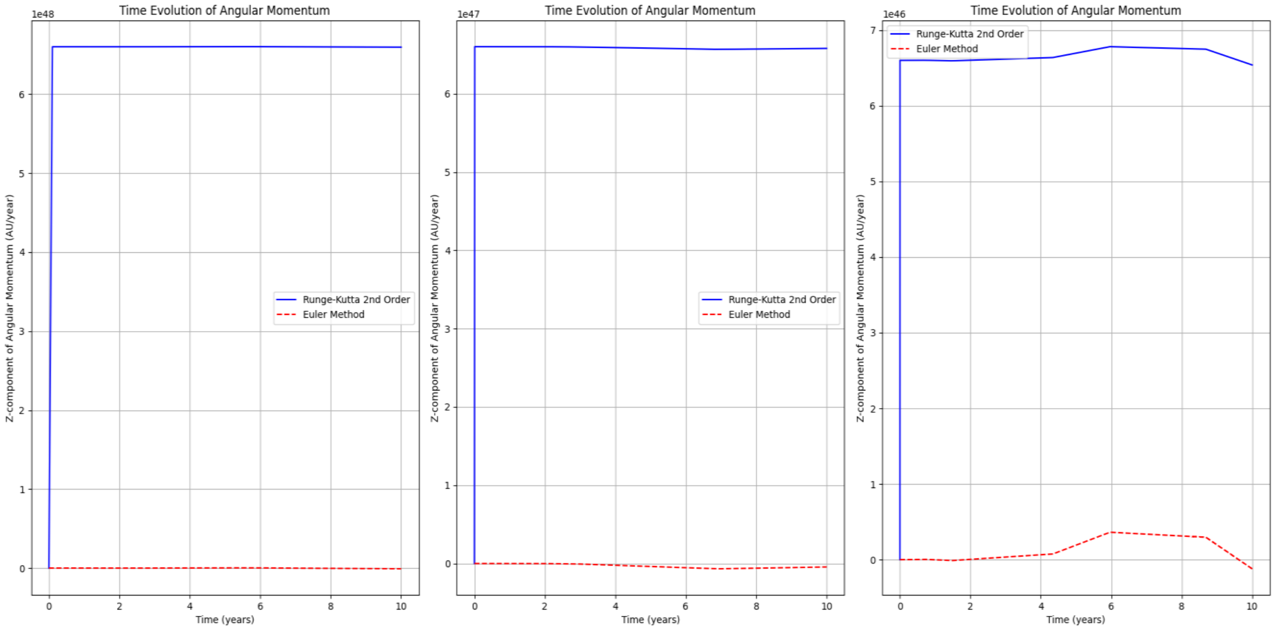
Part D:

As can be seen, the Runge-Kutta method produces a perfectly circular orbit for the Earth. A problem with this plot, however, is that the plots calculated for the different timesteps all have the same values.



[Fig.5] Plot of the orbit of the Earth around the Sun using the Runge-Kutta 2nd order method. The yellow dot at the centre is the sun. The orbit is calculated to be a circle; however, the plot only shows the smallest timestep.

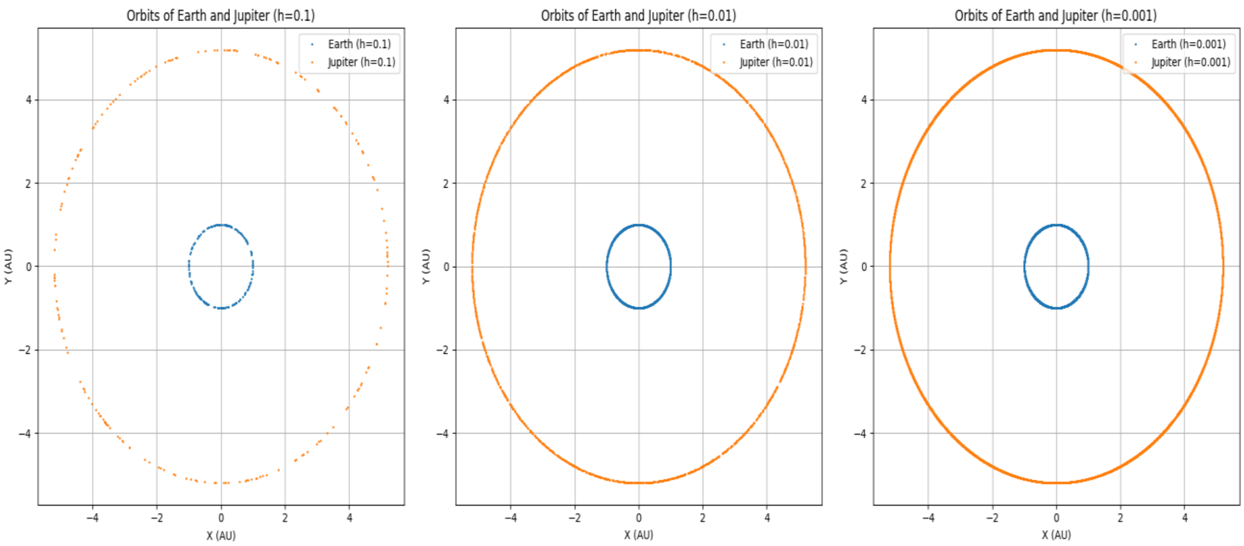
As the timestep used for the Runge-Kutta and Euler methods decreases (decrease from left to right), fluctuations are introduced in increasing magnitudes. These fluctuations are expected as was discussed in Part C; however, in these plots, the momentum remains mostly constant. The momentum of the Earth of the Euler method should not be that close to zero.



[Fig.6] Plot of the time evolution of the angular momentum of the z-component over the timescale of 10 years. This plot used the Runge-Kutta 2nd order method which is the blue line, and the Euler method is the dotted red line. As the time step becomes smaller (right most graph) the momentum begins to noticeably fluctuate in both the Runge-Kutta and the Euler plots.

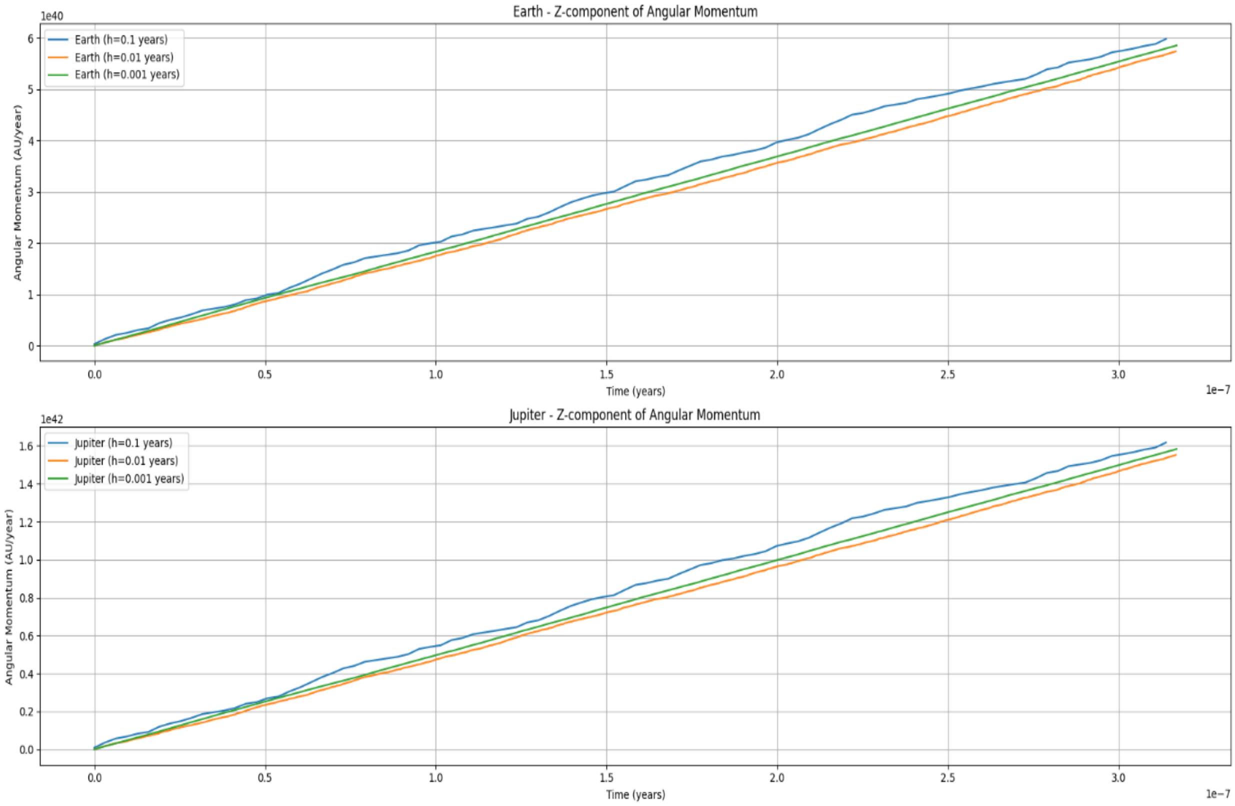
Part E:

Similarly, to Part D, the Runge-Kutta method produces a perfectly circular orbit for the Earth and Jupiter. As the timestep becomes smaller, the orbit becomes more defined.



[Fig.7] Plot of the orbits of Earth and Jupiter around the Sun using Runge-Kutta 2nd order method. As expected, the orbit of Jupiter is much larger than that of the Earth's orbit. A problem with this graph is that only the smallest time step was plotted as all three timesteps produced the same values.

As the timestep decreases, fluctuations in the lines are reduced. These fluctuations are expected as was discussed in Part C; however, in these plots, the momentum seems to establish an increasing linear relationship with time. The momentum of Earth and Jupiter should not be increasing this rapidly in this sort of a timespan. When considering the values of the y axes of both plots, we see that the Earth's momentum is much greater than the momentums calculated for Jupiter which also should not be the case.



[Fig.8] Plot of the time evolution of the angular momentum of the z-component for both Earth and Jupiter over the timescale of 10 years. While both plots only display up to 3 years, a max limit of 10 years for the simulation was set. The momentum plotted shows a linearly increasing relationship with time. As mentioned in [fig.6] this should not be the case.

Discussion:

When discussing these programs, one must remember that assumptions are being made, the primary assumption being that the orbits are perfectly circular. The orbits of Earth and Jupiter actually follow an elliptical shape. As such, this inaccurate depiction of the orbit of both planets affects the angular momentum calculations in Parts C, D, and E. If the orbits were instead calculated without this assumption the momentum values and plots of the momentum's evolution for both Earth and Jupiter.

When calculating the momentum, there are no definitive values for the angular momentum for Earth and Jupiter as it fluctuates. The order of magnitude of these values for Earth however is $\sim \times 10^{33}$ and $\sim \times 10^{38}$ for Jupiter [2]. However, when investigating the magnitudes of the angular momentums displayed in [fig.4], [fig.6], and [fig.8], it can be seen that the respective magnitudes are $\times 10^{45}$, $\times 10^{46} - 10^{48}$, and $\times 10^{40} - 10^{42}$. These values are must greater than the approximate magnitudes. Most likely this is due to the scaling of the graph. As three different plots are being generated and plotted on the same graph, a large range of values would lead to a scaling that would skew results.

Another important point to consider with this project is that the simulations and calculations were preformed under the assumption that the setup is in 2 dimensions while the solar system is in 3 dimensions. For this reason, these orbits and momentums do not match the actual solar system. While these models are obviously only an approximation of a difficult system to accurately model.

When considering the difference between the Euler and Runge-Kutta 2nd order methods, the biggest difference is the error which they produce. The Euler method is prone to large errors when using large timesteps. Runge-Kutta is still prone to these errors when using large timesteps but is more accurate and reliable than Euler's method.

Conclusion:

The experiment partially successful as the orbits of Earth and Jupiter were simulated correctly, however, the momentums calculated were not accurately calculated. A closer look at the programs written must be conducted to improve the results obtained in the plots.

References:

[1] PY2015 lecture notes, ODE1

[2] Approximate orders of magnitude

<https://www.zipcon.net/~swhite/docs/astronomy/Angular_Momentum.html#:~:text=Based%20on%20this%20calculation%2C%20Jupiter's,Earth%2C%20which%20is%207.1e33. >

- Nicholas G, Hisao N "Computational Physics" (Pearson/Prentice Hall, 2nd edition, 2006)