Draft version January 23, 2020
Typeset using LATEX **preprint** style in AASTeX63

Modeling Lab 2: The Kepler Problem

Kathlyn M. Purcell[1]

[1] *Villanova University*

ABSTRACT

To evaluate Kepler's Problem, we must implement Newton's Laws of Motion and Gravitation. Equation 1 (Newton's Second Law of Motion) can be substituted into equation 2 (Newton's Law of Gravitation) to create a working equation to discover the orbital size of the two body system (equation 3).

$$F = ma \tag{1}$$

$$F = -\frac{G * M * m}{r^2}\frac{\vec{r}}{r} \tag{2}$$

For our project, we want to study the position of a small object orbiting a larger object. We can do this by solving for $\vec{r}$ in Newton's Second Law of Motion by using the processes described in equations 3 and 4:

$$\ddot{\vec{r}} = -\frac{GM}{r^2}\frac{\vec{r}}{r} \tag{3}$$

$$\vec{r} = -\int\int \frac{GM}{r^2}\frac{\vec{r}}{r}dtdt \tag{4}$$

There is a major problem, in that this integral cannot be calculated the traditional way. In order to study the path of the orbit for any multiple body system a creative solution must be implemented. The integral can be evaluated several ways, including iterating through Newton Raphson, or using Runga Kutta techniques. First, an iterative solution is used to plot the orbit of one period for the Halley's Comet orbit. This technique results in a constant orbit, where no matter the timescale, the orbit remains constant and free from major error. Later, the Runga Kutta method is implemented. Here the path of the object slightly changes over time, resulting in a non-uniform orbit dependent on the timescale. Both correctly determine the orbit of the two-body system, but the solution in Runga Kutta allows for minute changes in the elliptical orbit.

1. INTRODUCTION TO NEWTON RAPHSON ITERATION METHOD

In order to implement the iterative solution, the position projected onto a circular orbit, then iterated to find a value close to the true elliptical value. In this way the solution is not a direct integration of equation 4, but it is an accurate calculated proxy for the true radius and angle of orbit. Eccentric anomaly is used to understand the position on the projected circular orbit. Eccentric anomaly is the angle from the origin of the system to the position of the object in its circular orbit.

1.1. *Mean Anomaly*

To calculate this value, mean anomaly is used (equation 5) to measure the time of the orbit in respect to its period.

$$M = 2\pi \frac{(t - t_p) mod P}{P} \tag{5}$$

Here M is mean anomaly, P is period, t is the current time, and t_p is time at perihelion. The ( )mod P / P presents the results in a fraction of a period. This allows the program to run faster, without being bogged down with unnecessary multiples of the period over time. This also results in a constant calculation for the orbital size. No matter the time period over which this function is run (as long as its more than one period) the resulting orbit will be equivalent to any subsequent ones. This make for no processing of the orbit over time or error accumulation.

This calculation was programed through a while loop. The time is set to 0 initially and run for every time value in one period of the orbit of Halley's comet. The list is 3963 units long or 76 years in weeks; one period for Halley's comet. Obviously, the while loop resulted in a list for M(t) that was 3963 units long.

1.2. *Eccentric Anomaly*

After mean anomaly is found, the Newton Raphson method is used to calculate the most accurate value for the elliptical position of the orbit. This was briefly mentioned in the abstract. Further explained, in order to calculate the position of the object, equation 4 cannot be integrated to find the answer. Newton Raphson method is used by projecting the position of the object onto a circular orbit then calculated. This position is iterated through equation 10 to reduce the distance between the circular and elliptical orbits to find the true value of the object.

Equation 7 is used as the primary function. This primary function supplies the components given in equation 8 and 9, which are reconstructed through the Newton Raphson method (equation 6) to result in equation 10: our equation to determine the eccentric anomaly.

$$f(x_0) = f'(x_0)(x_0 - x_1) \tag{6}$$

$$M = E - esinE \tag{7}$$

$$f(E) = E - esinE - M \tag{8}$$

$$f'(E) = 1 - ecosE \tag{9}$$

$$E_(n + 1) = E_n - \frac{E_n - esinE_n - M}{1 - ecosE_n} \tag{10}$$

In order to program this, I made a while loop in another nested while loop. The nested loop ran the function with the respective M as the first E_n value. After the first run through the loop I set E_n to E_{n+1} I continued this loop 100 times to ensure that it found an accurate value for the iterated elliptical orbit. When it found the 100th iteration for the respective M and E_{n+1} then I stepped up the M value and ran it in the primary while loop until all M values were solved for (this took less than one second to run even for [accurate] 100 periods so I felt comfortable with this loop constraint). Again, this resulted in a 3963 value list.

1.3. *True Anomaly*

To find θ, equation 11 was used. The value we are solving for is called the true anomaly as it is the value for the elliptical orbit of the two-body system, where the eccentric anomaly was the angle for the circular orbit.

$$tan(\theta/2) = (\frac{1+e}{1-e})^{1/2} tan(E/2) \tag{11}$$

Again, a while loop was used to timestep over the whole time scale and it resulted in a 3963 unit list.

1.4. *Radius*

True anomaly was used in equation 12 to solve for the radius of the second body over the entire time scale of the period. This results in a function of the radius of the orbit based on the objects position in respect to the more massive body and the point of perihelion. Again, resulting in a list of 3963 units, giving final values of θ and r over the entire period of Halley's comet orbit in timesteps of weeks.

$$r(\theta) = \frac{a(1-e^2)}{1+ecos\theta} \tag{12}$$

2. RESULTS OF ITERATIVE METHOD

In this question, Halley's orbit was plotted as a function of x and y in Figure 1, 2 and in polar coordinates in Figure 3. This sows a true representation of Halley's comet's orbit of perihelion distance of 0.587AU and aphelion distance of 34.98 AU with a y $\pm$ value of 4.53 AU in cartesian coordinates. In polar coordinated the orbit spans from 0 to 180 degrees and has a max radius of 34.98 AU.

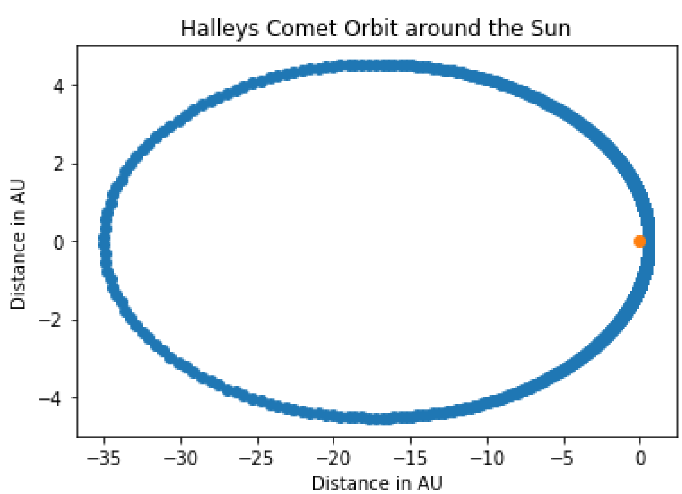

Figure 1. Halley's Comet Orbit (Blue) around the Sun (orange, origin), *Made through PYPLOT*

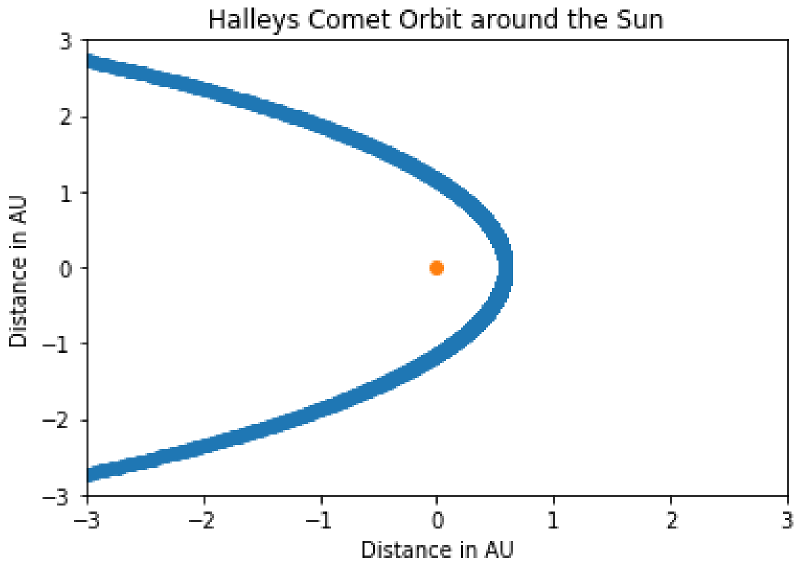

Figure 2. Zoomed Version of Halley's Orbit with Respect to the Sun, *Made through PYPLOT*

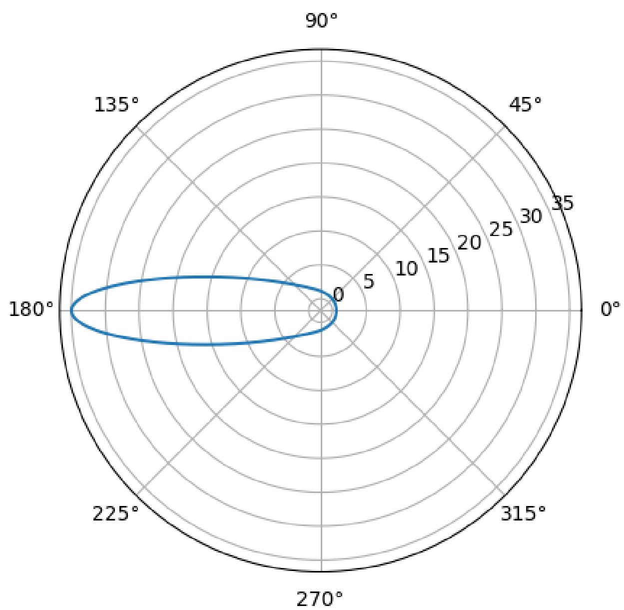

Figure 3. Halley's Orbit Plotted in Polar Coordinates, *Made through PYPLOT*

At this instance, the attached movie *halley_orbit.avi* should be referenced to see an animated version of the cartesian plot with respect to time. This shows the plotted orbit of Halley's comet, while also animating a moving version of the comet's orbit for the reader to understand that the orbit is truly a function of time, not simply a elliptical line. This notion comes into play in the next addition to the project where other objects are added. The stationary images of their orbits may suggest that the orbits overlap and the objects are constantly interacting, but it will come to light that interaction between the objects is a very scarce occurrence.

2.1. *Adding Other Objects*

When these figures and animations were made in order to understand Halley's Comet and its orbit with respect to the sun, further measures were taken to understand its orbit with respect to other nearby objects. The rest of the relevant solar system was plotted to understand Halley's orbit. This was shown again in cartesian coordinates (Figure 4 & 5) and in polar coordinates (Figure 6 & 7). The planets Earth, Mars, Jupiter, Saturn and Uranus were plotted because they were relevant to the size of the orbit for Halley's comet. Mercury and Venus were left out due to focus on Earth at near-Sun orbits and Neptune was neglected because Halley's orbit was inside the radius of the orbit of Neptune.

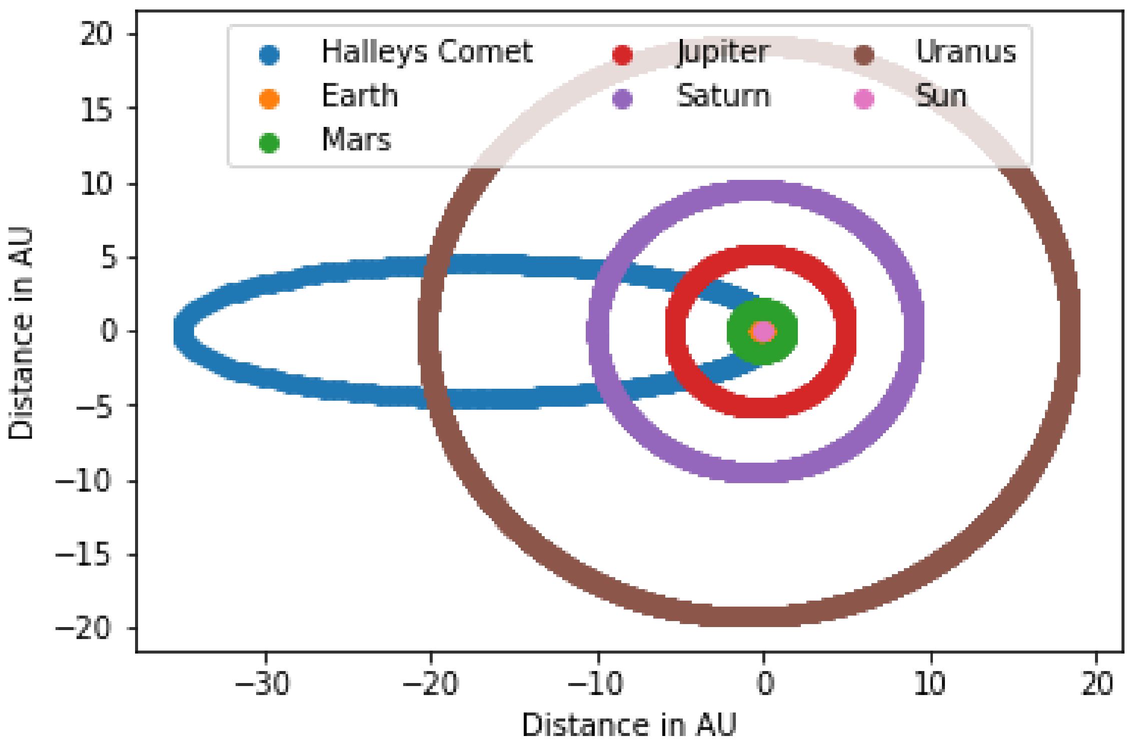

Figure 4. Halley's Orbit Plotted with Other Solar System Planets, *Made through PYPLOT*

2.2. *Near Object Occurrence Rates*

As stated in the previous section, the stationary plots of the orbital paths of the planets and comet would make the reader infer that their interactions between the objects are common. This is not the case and this section the aim is to quantify these interactions. This occurs by determining the number of times the x and y coordinates of Haley's comet are equal to the x and y coordinates of the other planets.

This is done for each planet for each value of t (the timestep) through a function of if loops, which each test if the x AND y coordinates are equivalent for a given t in each of the planets. If any of the planet's coordinates match that of Halley's comet then their respective indicator is increased by one unit. This is then nested in a larger while loop that increases the t value until it reaches a maximum,

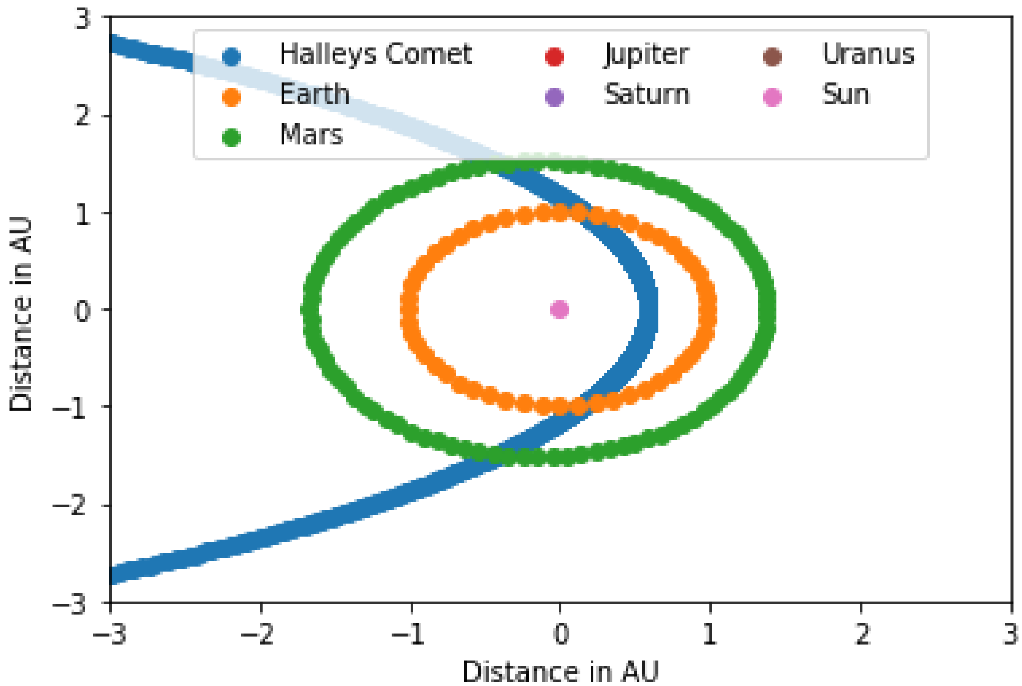

Figure 5. Halley's Orbit with Other Solar System Planets Zoomed to the Sun, *Made through PYPLOT*

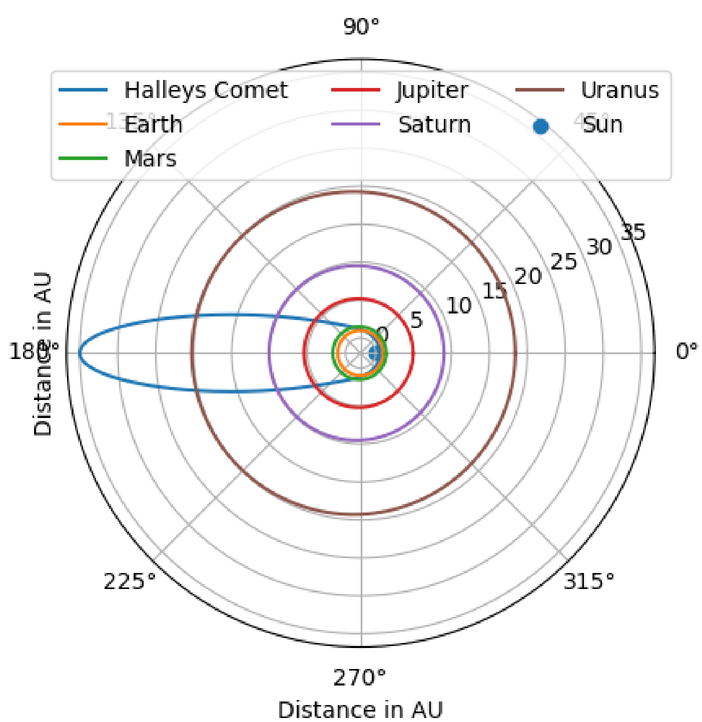

Figure 6. Halley's Orbit with Other Solar System Objects in Polar Coordinates, *Made through PYPLOT*

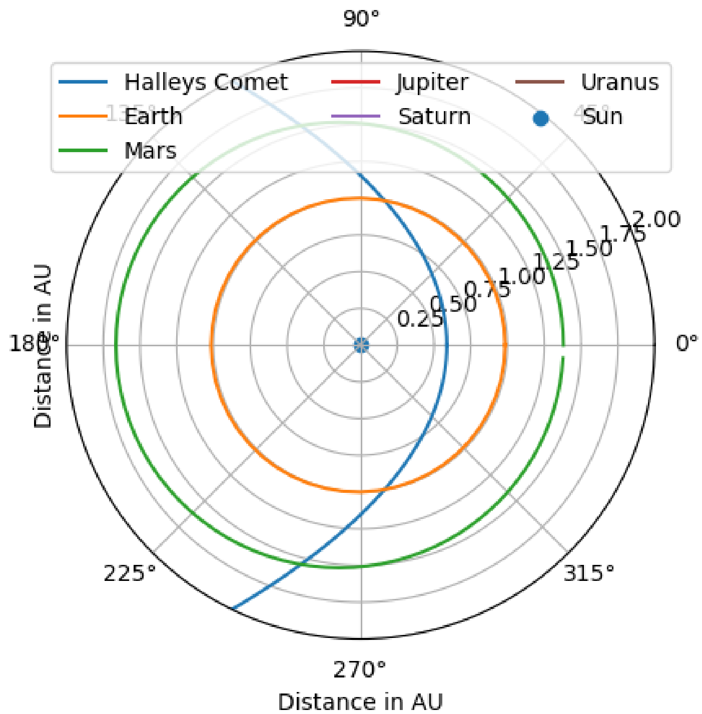

Figure 7. Halley's Orbit (Zoomed with Respect to the Sun) in Polar Coordinates, *Made through PYPLOT*

for this test is was one period of Halley's Comet (hereby HC). This was t=3963 which stands for the number of weeks in the period of 76 years.

Table 1 is the result of a test of direct impact with the planets. It is clear that none of the planets are directly hit in only one period of HC. It should be noted here that this simulation is derived from an instance where all objects begin at their perihelion point and are then set into their respective orbits from that position. All of the objects run until a time of 3963 weeks so, for example, Mars has a period of 98 weeks, so Mar's orbital path is run approximately 40 times until one orbit of HC is complete. This direct impact simulation was run over one HC time period, then 10 periods then 100, and finally 1000 HC periods or 76,000 years. **Table 1 is a reflection of 1000 years as well as 76 years.** This assures the reader that no direct impact is inevitable. All this is still to say, there are still near-object occurrences to investigate.

Object	Constraint Distance (AU)	Occurance Rate
Earth	0	0
Mars	0	0
Jupiter	0	0
Saturn	0	0
Uranus	0	0

Table 1. HC Occurance Rate of Direct Impact with Solar System Objects over One HC Period

Object	Constraint Distance (AU)	Occurance Rate
Earth	0.3	107
Mars	0.45	33
Jupiter	1.56	0
Saturn	2.85	0
Uranus	5.7	26

Table 2. HC Occurance Rate of Near Object Passes Determined By the Constraint Distance (On either side of the location of the object) Over one HC Period

Since a direct impact situation resulted in zero instances of interaction between the planets and HC, the simulation was run again but instead constraining the impact coordinates to be exactly equal, some liberty was given. This relaxation of the constraints reflected the definition for near-Earth objects. This is a range from 1.3 to .97 AU allowing for 0.3 AU on either side of the planet to constitute a recognized Near Earth Occurrence (NEO). This constraint works for Earth but had to be calibrated for the larger and further planets like the gas giants.

First a relationship depending on mass was attempted but this resulted in constraint values for Jupiter of 95.4 AU which is well outside of our solar system, rendering every point in HC's orbital path to be a Near Jupiter Orbit.

After this, an orbital radius function was developed that took into account not the mass of the planet but its distance from the sun. This resulted in Table 2. Also in Table 2 is the occurrence rate for a Near Planetary Occurrence for each planet with respect to its constraint distance. Here it should be noted that while Mars had a larger constraint distance, it experienced less Near Mars Occurrences (NMO) than Earth due to the fact that the perihelion distance of HC (0.587 AU) was closer to the NEO zone (1.3-.97AU) that it was to the larger NMO(1.95-1.05 AU) zone.

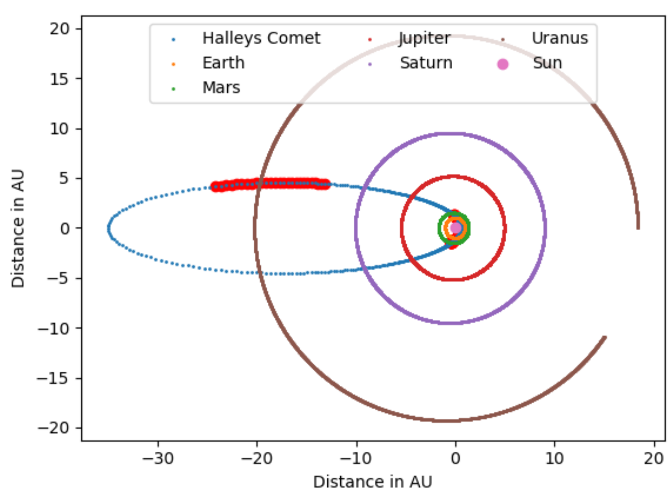

Figure 8. Halley's Orbit with Primarily NUO Shown in Red, *Made through PYPLOT*
It should be noted that the orbit of Uranus is not complete in this plot. The simulation was run over one period of HC while the period of Uranus is longer. This has no impact on the data presented.

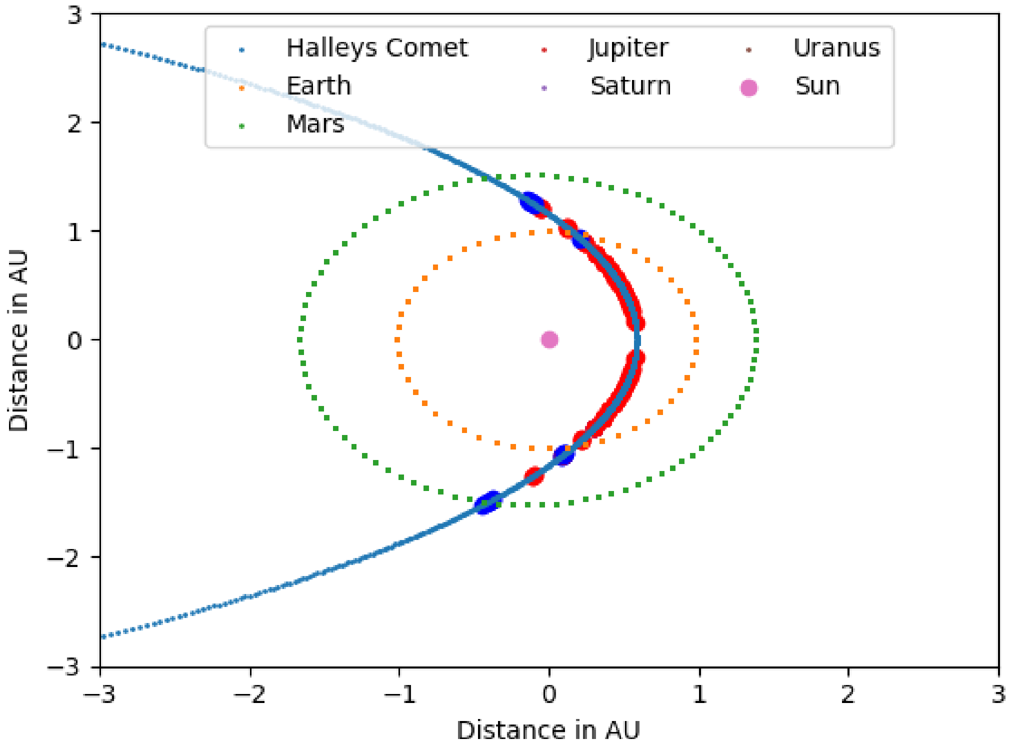

Figure 9. Halley's Orbit (Zoomed with Respect to the Sun) With Red Dots Signifying NEO and Blue Dots Signifying NMO, *Made through PYPLOT*

Figure 8 displays primarily the Near Uranus Occurrences, plotted in red, while Figure 9 zooms into the Sun and gives a better indication of the NEO and NMO. These are significant because any one of them could be too close to the planet and jostle the orbital stability of HC causing it to enter an unexpected orbital path.

While this is a simulation of the random occurrence that all objects would align at their perihelion points and then begin their orbits, it still gives important insight to the dangers of impact (Table1), or more likely, orbital course readjustment (Table 2) that the planets could have on the state of HC's orbit. A Near Uranus Occurrence may seem far off or not very likely, but a simple nudge by the large planets gravitational field can send HC on a crash course to Earth where, with an unstable orbit, the NEO rate is much higher. This large and fast moving comet could wreak havoc on the atmosphere of Earth with only one too-close-for-comfort NEO.

3. RUNGA KUTTA 3RD ORDER

In question 2 the goal was to describe the variables in a Runga Kutta of the 3rd Order. For the 2nd Order the values of $\gamma_1 + \gamma_2 = 1$ and $\beta = \alpha = 1$. This is the representation of the Huene solution to the 2nd Order Runga Kutta (RK). The following equations describes the process of finding the solution for RK 3rd order and solving for the variables.

n=3:

$$y(x+h) = y(x) + \gamma_1 k_1 + \gamma_2 k_2 + \gamma_3 k_3 \tag{13}$$

$$k_1 = hf(x,y) \tag{14}$$

$$k_2 = hf(x+\alpha_1 h, y+\beta_1 k_1) \tag{15}$$

$$k_3 = hf(x+\alpha_2 h, y+\beta_2 k_1+\beta_2 k_2 \tag{16}$$

$$y(x+h) = y(x)+\gamma_1 hf+\gamma_2 hf(x+\alpha_1 h, y+\beta_1 k_1)+\gamma_3 hf(x+\alpha_2 h, y+\beta_2 k_1+\beta_2 k_2 \tag{17}$$

$$y(x+h) = y(x)+\gamma_1 hf+\gamma_2 hf(x+\alpha_1 h, y+\beta_1 hf)+\gamma_3 hf(x+\alpha_2 h, y+\beta_2 hf+\beta_2 hf(x+\alpha_1 h, y+\beta_1 hf)) \tag{18}$$

$$\begin{split} y(x+h) = y(x)+\gamma_1 hf+\gamma_2(hf+\alpha_1 h^2 f_x+\beta_1 h^2 f_y f)+\gamma_3(hf+\alpha_2 h^2 f_x+\beta_2 h^2 f_y f+\beta_2 h^2 f' \\ +\beta_2\alpha_1 h^3 f_{xx}+\beta_1\beta_2 h^3 f_{yy} f) \end{split} \tag{19}$$

$$\begin{split} y(x+h) = y(x)+(\gamma_1+\gamma_2+\gamma_3)hf+(\gamma_2\alpha_1+\gamma_3\alpha_2)h^2 f_x+(\gamma_2\beta_1+\gamma_3\beta_2)h^2 f_y f+(\gamma_3\beta_2)h^2 f' \\ +(\gamma_3\beta_2\alpha_1)h^3 f_{xx}+(\gamma_3\beta_2\beta_1)h^3 f_{yy} f \end{split} \tag{20}$$

$$TAYLOR..SERIES y(x+h) = y(x)+hf(x,y)+1/2h^2 f'+1/6h^3 f''+... \tag{21}$$

$$\gamma_1+\gamma_2+\gamma_3 = 1 \tag{22}$$

$$\gamma_2\alpha_1+\gamma_3\alpha_2 = 1/2 \tag{23}$$

$$\gamma_2\beta_1+\gamma_3\beta_2 = 1/2 \tag{24}$$

$$\gamma_3\beta_2 = 1/2 \tag{25}$$

$$\gamma_3\beta_2\alpha_1 = 1/6 \tag{26}$$

$$\beta_2 = \frac{1}{2\gamma_3} \tag{27}$$

$$\gamma_3\frac{1}{2\gamma_3}\alpha_1 = 1/6...\alpha_1 = 1/3 \tag{28}$$

$$\gamma_3\frac{1}{2\gamma_3}\beta_1 = 1/6...\beta_1 = 1/3 \tag{29}$$

$$\gamma_2\beta_1 + 1/2 = 1/2...\gamma_2\beta_1 = 0...\gamma_2 = 0 \tag{30}$$

$$Here..I..got..stuck..so..I..admit..I..consulted..the..Huen..Method..for..the..3^{rd} \\ ..order..to..solve..for..only..\gamma_1 = 1/4 \tag{31}$$

$$1/4 + 0 + \gamma_3 = 1...\gamma_3 = 3/4 \tag{32}$$

$$3/4\beta_2 = 1/2...\beta_2 = 2/3 \tag{33}$$

$$0 + 3/4\alpha_2 = 1/2...\alpha_2 = 2/3 \tag{34}$$

Here because I used the Huene Method to unlock only the value for γ_1, the result was a full solution to the variables in the Huene Method of the Runga Kutta 3rd Order. These values were checked against the published Huene Method as well as against themselves in their respective 6 equations to ensure total accuracy for this problem. The Runga Kutta method was used in the next question but adapted to the fourth order to ensure the accuracy of predicting orbits for Halley's Comet and other objects.

4. INTRODUCTION TO RUNGA KUTTA 4TH ORDER METHOD

This question implemented the use of the 4th Order of the Runga Kutta method. This was adapted from pre-published programs utilizing this complicated 4th order iteration. This project was based on the program given on StockOverflow at *https://stackoverflow.com/questions/51575544/two-body-problem-scipy-integrate-rk45-gives-broadcasting-error-and-scipy-integr* and its solutions.

This program was a definition of the RK solution to the two body problem, initialization of variables, and a solution run with scipy.integrate.OdeInt. First, the initial conditions were defined as such: a = 17.788 AU (HC), x0= 0.587 AU (HC), y0=0 AU, vx0=0 AU per s and equation 35 for v_{y0}:

$$v_{y0} = math.sqrt[(GM) * \frac{2}{x_0} - \frac{1}{a}]AU/s \tag{35}$$

The initial conditions were appended to an array of 3963(again the time period and number of steps based on the number of weeks in HC orbit) called Y0. After this the four necessary equations were derived, they are described as such in equations [36,-7,-8,-9] where r is defined as in equation 40:

$$\dot{y}[0] = \dot{[x]} = v_x \tag{36}$$

$$\dot{y}[1] = \dot{[y]} = v_y \tag{37}$$

$$\dot{y}[2] = \dot{v_x} = -\frac{GMx}{r^3} \tag{38}$$

$$\dot{y}[3] = \dot{v_y} = -\frac{GMy}{r^3} \tag{39}$$

$$r = math.sqrt(y[0]^2 + y[1]^2) \tag{40}$$

5. RESULTS OF RUNGA KUTTA 4TH ORDER METHOD

This was run through OdeInt (equation 41) which developed the necessary solution for the 4th order RK method. Where the x and y values were given in solution[:,0] and solution[:,1] respectively. This function was run over a timescale of one period and the period was divided into 100,000 sections, ensuring a smooth plotted curve for the result. These values were plotted (Figure 10) to understand the orbit and then animated as given in rk_100000_halley.avi which is attached.

$$solution = odeint(twoBody, Y0, np.linspace(0, 3963, 100000), args = (mu,)) \tag{41}$$

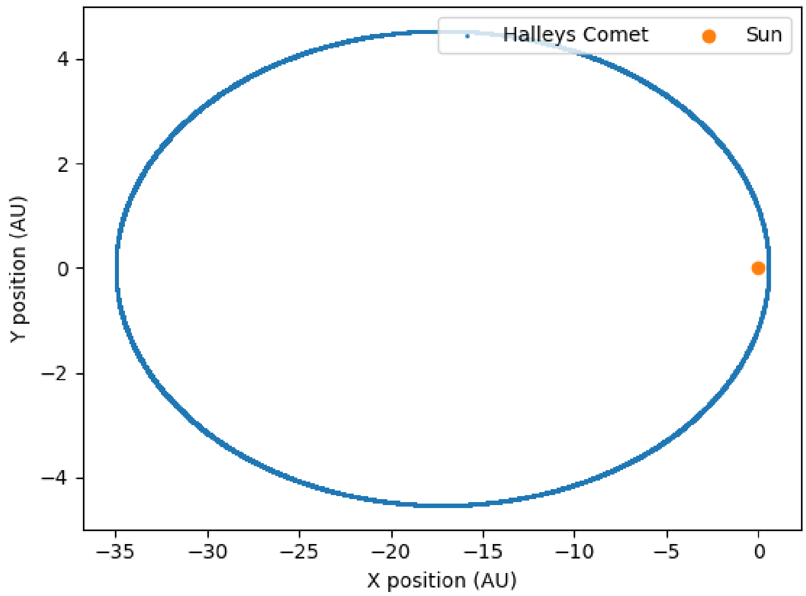

Figure 10. Halley's Orbit Developed with Runga Kutta 4th Order Method, *Made through PYPLOT*

I then compared this model (RK), for predicting the orbit of HC, with the first model (Iterative, Newton Raphson). This is displayed in Figure 11 as well as in Figure 12. Figure 11 shows the plot of the two values compared with each other. This means the r value for iterative over time minus the r value for Runga Kutta as a function of time. Figure 12 shows that there is almost no error between the two orbits when they are plotted over each other. It is clear that the plots overlap in figure 12 but the plot in figure 11 shows a large accumulation of error. This can be described by the fact that the iterative solution shows the orbit of HC over constant timesteps, while RK method shows the timestep of an integration. This makes the orbit plot and animate normally but when compared to corresponding t values for iteration it becomes incongruent. The integration shows jumps before and after the 'time scale' for the r value but since it jumps forward and back in a linear progression it 'time accurate' when animated. This allows the orbit and animation to be correct but still result in a large accumulation of error when compared time step by timestep with the iterative solution that solves for the orbit linearly. As the timescale is increased the error would also increase as more of the Runga HC orbit is 'filled in'. This can be seen more clearly in rk_500_halley.avi and rk_5000_halley.avi of a

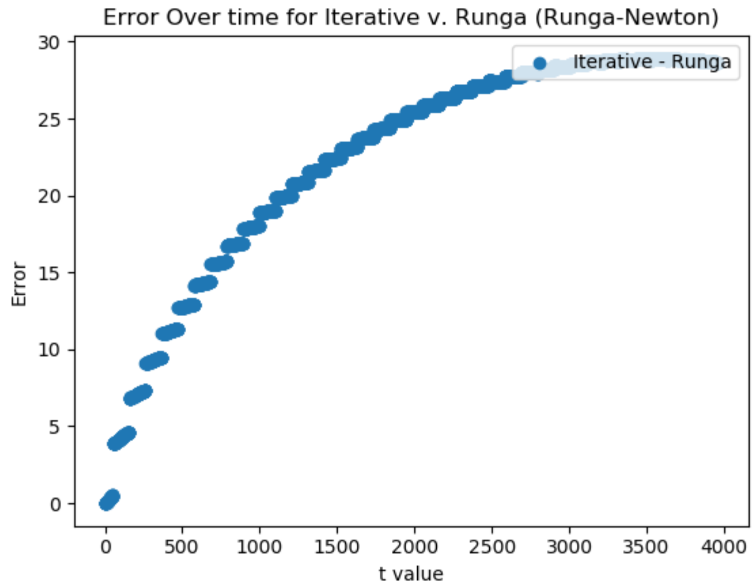

Figure 11. Halley's Orbit Error Compiled over Time between Newton Raphson and Runga Kutta 4th Order Methods, **Error (y-axis) in AU**, *Made through PYPLOT*

period of 3963 weeks divided in 500 and 5000 sections respectively. This 'filling in' is not apparent in rk_100000_halley.avi because it occurs in steps around the orbit so it appears to calculate linearly over the tiny timestep increments while 500 and 5000 steps aren't enough to mask this jumping.

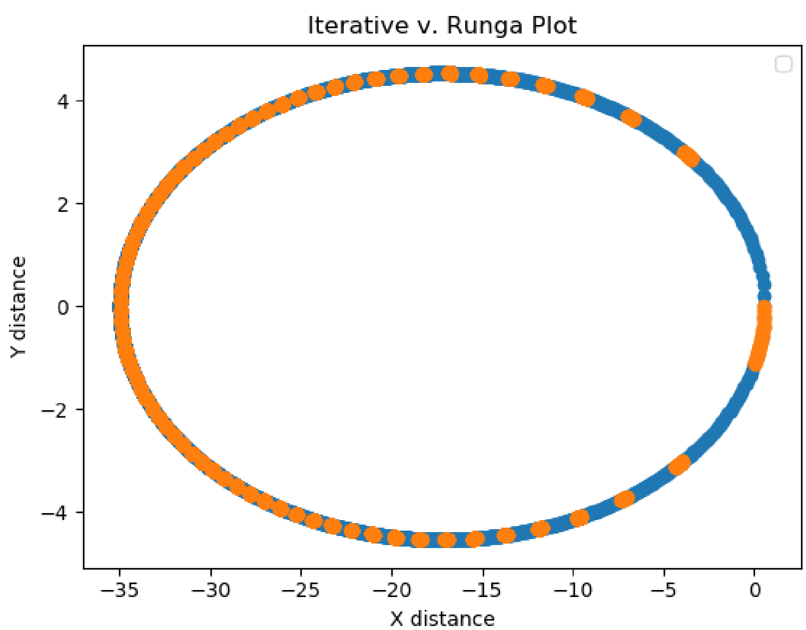

Figure 12. Halley's Orbit comparison between Newton Raphson and Runga Kutta 4th Order Methods, **X and Y axis both in AU**, *Made through PYPLOT*

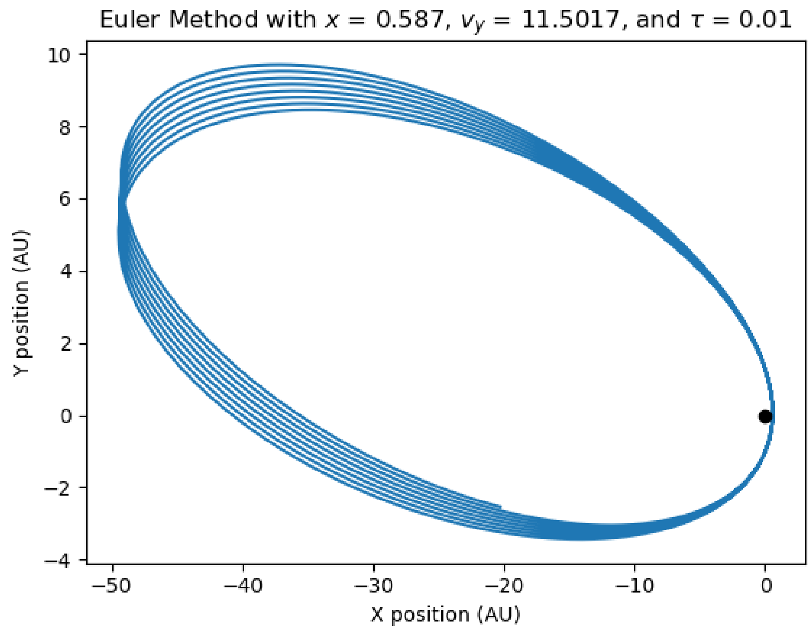

Figure 13. HC Orbit over 1000 years Modeled with Euler's Method, *Made through PYPLOT*

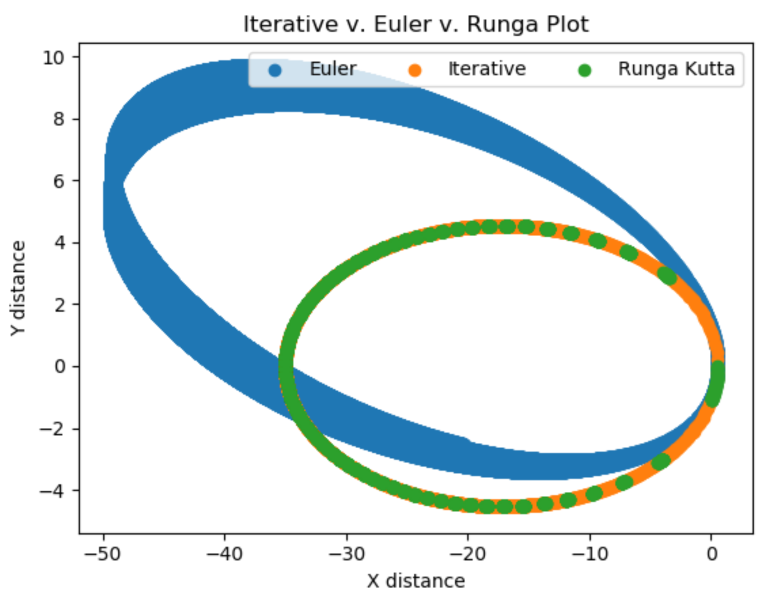

Figure 14. Halley's Orbit comparison between Euler Method & Newton Raphson and Runga Kutta 4th Order Methods, **X and Y axis both in AU**, *Made through PYPLOT*

6. INTRODUCTION AND RESULTS FOR OTHER INTEGRATORS

After the Runga Kutta 3rd and 4th order and Newton Raphson method, another was introduced: The Euler Method. This is a method that uses the initial conditions for the orbit (which are known from the 4th order Runga Kutta) and it steps forward once per timescale in order to predict the orbit of HC. This was initially used in place of the Runga Kutta method but was quickly refused due to its error accumulation over a small timescale. This predicted HC orbit is given in Figure 13. This is a proper elliptical orbit, but it is important to notice the size and procession of the orbit, which are inaccurate. The procession is shown with time dependence in the attached file: *euler_1000years_halley.avi*. Both of these attachments are over a period of 1000 years, just over 13 periods of HC.

After each period the orbit of HC shifts under the Euler method which results in a significant error accumulation in comparison to both Newton and Runga methods over a very short timescale. The scale of the error is shown in Figure 14. The main reason that the Euler method fails for the HC orbit, is that HC is elliptical while Euler method works primarily on circular orbits. Since the velocity at perihelion is so large due to the centripetal velocity at such a quick angle, the Euler method attempts to describe the velocity as a 'circular' orbit, failing at making it circular and ultimately producing a very large (longer and wider) and high velocity inaccurate orbit.

7. CONCLUSION

Even though the integral given in equation 4 is non-integrable, there are still many ways to solve for the position of a two body system. These ways include an iterative Newton Raphson method as well as the 3rd and 4th order Runga Kutta method (along with a precarious Euler method). It was determined that the Newton Raphson and the Runga Kutta 4th Order method were the most accurate to model the HC orbit, these axis and timescale models were checked against published values with little discrepancies. The Euler method was determined to be quite inaccurate compared to both the other methods and the published values of the HC orbit. This project was successful in modeling the orbit of HC to an accurate figure not only once but twice through Newton Raphson and Runga Kutta 4th order.

ACKNOWLEDGMENTS

This paper would like to acknowledge support from many helpful, extensive emails with our very own Andrej Prša. A special thank you should also go out to all others in this class for their support and knowledge including over all others Alicia, Gus, Alex, Lindsay and Tyler for support both morally and computationally. Of course, a huge thank you to Stack Overflow.