Modelling the Orbit of a Probe around the Moon

Sarah Nicole Straw

14th October 2024

Abstract

Using python to numerically integrate a system of ordinary differential equations, formed from equating gravitational and centripetal forces for both the probe and the moon and setting up their respective equations of motion, the state of the system in terms of positional and velocity vectors were evolved across an array of time values and plotted. The time period for the modeled moon was 99.44% accurate.

1 Introduction

This program solves the equations of motion for two bodies in orbit which relate to each other. Modelling the moon's orbit of the earth and a lunar probe orbiting the moon, the gravitational field of the earth and moon effect the probe, extending the complexity of it's motion. This demonstrates the application of rotational mechanics to large scale orbitals and how gravitational force can be combined with centripetal force to model orbiting bodies.

2 Theory and Methods

To form the equations of motion for the moon the earth was fixed at the origin, meaning only the gravitational force applied on the moon was taken account of due to the relative size of the moon being only around 1% of the earth's mass. The entire system was then formed in the earth's frame of reference and plotted relative to the earth. The motion of the probe took account of both the gravitational attraction of the moon and that of the earth, taking the initial values in the moon's frame of reference and translating them to the earth's frame to plot.

2.1 Moon's Orbit Calculation

Equation of motion for the Moon:

$$M_m \ddot{r}_m = -\frac{M_e M_m G}{|r_m|^3} r_m \tag{1}$$

Here Mm represents the mass of the moon, Me is the mass of the earth, $G$is the gravitational constant and r_m is the positional vector of the moon from the origin, where the earth is fixed. $r_m = (x_m, y_m)$ is the components of the moon's position which are plotted as they evolve with time.

Differential Equations of Motion in Cartesian components:

$$\frac{dx_m}{dt} = v_{m,x}; \quad \frac{dy_m}{dt} = v_{m,y}; \quad \frac{dv_{m,x}}{dt} = \frac{M_e G x_m}{({x_m}^2 + {y_m}^2)^{\frac{3}{2}}}; \quad \frac{dv_{m,y}}{dt} = \frac{M_e G y_m}{({x_m}^2 + {y_m}^2)^{\frac{3}{2}}} \tag{2}$$

These equations are listed as numbered functions to be solved within the 'Derivatives' function in the code.

To find the relation between initial radius of the moon's orbit to initial velocity the gravitational attractive force applied to the moon is equated with the required centripetal force for it's orbital motion. From this the mass of the moon is cancelled out and the relation becomes $r_m {v_m}^2 = GM_e$

. Inputing $r_m = 3.844 \cdot 10 * *8$ m, the radius of the moon's orbit of earth, and setting $v_m = \sqrt{\frac{GM_e}{|r_m|}}$ produces the orbital motion of the moon.

2.2 Probe's Orbit Calculation

Equation of motion for the Moon:

$$M_p \ddot{r}_p = -\frac{M_e M_p G r_p}{|r_p|^3} - \frac{M_m M_p G r_{pm}}{|r_{pm}|^3} \tag{3}$$

Here M_p represents mass of the probe, however this doesn't effect the motion of itself. The first part shows the gravitational force from the earth acting on the probe, the second part shows attraction from the moon.

Differential Equations of Motion in Cartesian components:

$$\frac{dx_p}{dt} = v_{p,x}; \quad \frac{dy_p}{dt} = v_{p,y}; \quad \frac{dv_{p,x}}{dt} = \frac{M_e G x_p}{({x_p}^2 + {y_p}^2)^{\frac{3}{2}}} - \frac{M_m G x_{pm}}{({x_{pm}}^2 + {y_{pm}}^2)^{\frac{3}{2}}}; \quad \frac{dv_{p,y}}{dt} = \frac{M_e G y_p}{({x_p}^2 + {y_p}^2)^{\frac{3}{2}}} - \frac{M_m G y_{pm}}{({x_{pm}}^2 + {y_{pm}}^2)^{\frac{3}{2}}} \tag{4}$$

These are added to the list of functions to be solved within the 'Derivatives' function in the code. The state of the system is extended to include the x and y components of displacement and velocity of the probe and these are evaluated across the same time array as that of the moon, so the resulting array of time evolved positions can be plotted together to show the moon and probe's motion.

The initial values for the probe were related by $r_{pm} v_{pm}^2 = GM_m$, where r_{pm} is radius of orbit from the moon and v_{pm} is velocity of the probe in circular motion in the moon's frame of reference.

3 Explanation of Code

3.1 Part 1: Moon's Orbit

First a function was set up to calculate the derivatives for the orbit of the moon displaced in equation (2). Naming this function 'Derivatives' and listing its variables, starting with the independent variable time and then the dependent variable state, which is a tuple dependent variables of the moon's displacement and mass. Extra variables are then listed, these being the masses of the moon and earth as well as the gravitational constant.

The derivatives in equation (2) are then computed in a specific order that will be followed throughout the code, and labelled f1, f2, f3 and f4. These are put in a tuple to be returned by the function.

```python
def derivatives(time,state,Me,Mm,G):


    """
    Function to compute the derivatives for the orbit of a body about the origin
    ----------
    time : independent variable, floating point, not used. (s)
    state : Tuple of floats containing (xm, ym, vmx, vmy).
    Me : mass of the earth (kg)
    Mm : mass of the moon (kg)
    G : gravitational constant
    -------
    Returns: tuple of derivatives (dxm/dt,dym/dt,dvmx/dt,dvmy/dt).
    """

    xm,ym,vmx,vmy = state # state of dependent variables at some time t
    f1 = vmx # dxm/dt = vmx
    f2 = vmy # dxy/dt = vmy
    rm = np.sqrt(xm*xm+ym*ym) #components of radius of orbit
    f3 = -Me*G*xm/rm**3 # dvmx/dt = f3
    f4 = -Me*G*ym/rm**3 # dvmy/dt = f4
    return (f1,f2,f3,f4)
```

Figure 1: Derivatives Function for Moon's Orbit

Next initial conditions were set up for each dependent variable as well as defining the constants. G, M_e, M_m and r_m were found by online research. The moon's orbit was to start on the postive x axis, so x_m0 was set to the value of the moon's orbital radius. At this point in the moon's anticlockwise orbit vertical velocity is at it's maximum, so $v_{my}0$ is set to equation $v_m = \sqrt{\frac{GM_e}{|r_m|}}$.

Next an array of times are made to evaluate the equations to, this is done using np.linspace. Here 10,000 time points are produced uniformly between the minimum and maximum times given.

A tuple of minimum and maximum time is set up and an initial state tuple in the same order as the state tuple is made with the initial value of each dependent variable.

To solve the equations of motion scipy.integrate.solve ivp is used, which numerically integrates a system of ordinary differential equations given an initial value. Inputting our differential equations (2) with the initial values tuple, solve ivp integrates the equations returning x_m, y_m, v_{mx} and v_{my} evaluated at each time point in the array we provided, and these variables are stored in the same order as they are in the state tuple.

```
t_interval = (t_min, t_max) # time interval tuple

initial_state = (xm0,ym0,vmx0,vmy0) # in same order as the state tuple

results = si.solve_ivp(derivatives, t_interval, initial_state, t_eval=teval, args=(Me, Mm, G), atol=1e-4, rtol=1e-4)
```

Figure 2: Solving the equations of motion for the array of times given.

To plot the results the array of x_m and y_m values must be plotted. These are extracted from the results object, within which is results.y, a 2d array of four rows, being the four state variables, and 10,000 columns, being the array of time points each variable is to calculated to. The first two rows contain the arrays of x_m and y_m, so these are extracted and plotted against eachother to show the orbit of the moon. Aspect ratio is set to 1 to ensure the y and x axes are proportionate.

```
x_values = results.y[0] # extracting the array of x_m values from results
y_values = results.y[1] # extracting the array of y_m values from results
ax.plot(x_values,y_values, label='Moon', color = '#7c7aff') # Make the plot
plt.legend(loc='upper right', fontsize='large')
plt.show()
```

Figure 3: Extracting results data and plotting

3.2 Part 2: Probe's orbit of the Moon

To include the motion of the probe, the probe's differential motion equations (4) were added to the list of function in the derivatives function. As the probes initial conditions are given in the moon's frame of reference, equations are given to relate x and y components of displacement in the moon's frame to the earth's frame of reference. The state tuple is also extended to include the dependent variables for the probe in the earth's frame.

```
ypm = yp-ym # m, definition of y component of rpm positional vector
xpm = xp-xm # m, definition of x component of rp positional vector
rpm = np.sqrt(xpm*xpm+ypm*ypm) # radius of orbit from probe around the moon from its components
rp = np.sqrt(xp*xp+yp*yp) # distance from probe to earth from components of rm and rpm

f5 = vpx # dxp/dt = vpx
f6 = vpy # dyp/dt = vpy
f7 = -Me*G*xp/rp**3 - Mm*G*xpm/rpm**3 # dvpx/dt = f7
f8 = -Me*G*yp/rp**3 - Mm*G*ypm/rpm**3 # dvpy/dt = f8

return (f1, f2, f3, f4, f5, f6, f7, f8)
```

Figure 4: Differential Equations of Motion for Probe added to the Derivatives function

Initial conditions are then established. Those for the probe are in the moon's frame of reference, such as lunar altitude r_{pm}. Equations adding the components for the moon and the probe from the moon are established using positional vector analysis to give the initial values for each dependent variable for the probe in the earth's frame of reference. The initial state tuple is extended to include the initial values for the dependent variables of the probe in the earth's frame.

```
# Probe initial conditions in Moon's frame of reference:

rpm = 8.0e6 # m, radius of the probe's orbit around the moon
xpm0 = rpm # m, initial distance from probe to moon
ypm0 = 0.0 # m, initially probe sits along x-axis to moon so y0=0
vpmx0 = 0.0 # m/s, initial x component of velocity of probe relative to the moon
vpmy0 = np.sqrt(G*Mm/abs(rpm)) # m/s, initial velocity along y axis

# Probe initial conditions in Earth's frame of reference:

yp0 = ym0+ypm0 # m, initial y coordinate of probe
xp0 = xm0+xpm0 # m, initial x coordinate of probe
vpx0 = vmx0+vpmx0 # m/s, initial x component of probe's velocity
vpy0 = vmy0+vpmy0 # m/s, initial y component of probe's velocity
```

Figure 5: Initial conditions translations between the frame of the Moon and the Earth

To solve the derivatives the scipy.integrate.solve ivp is used again with the new extended derivatives function and initial state tuple. The results object now has an array of eight rows, being the 4 state variables for the moon and probe, and 10,000 columns for the array of time points each variable is to calculated to. Rows 4 and 5 contain the array of x_p and y_p values to be extracted and plotted alongside the moon.

```
x_values_m = results.y[0] # extracting the array of x_m values from results
y_values_m = results.y[1] # extracting the array of y_m values from results
x_values_p = results.y[4] # extracting the array of x_p values from results
y_values_p = results.y[5] # extracting the array of y_p values from results

#plt.xlim(0.0,1.2e5)
#plt.ylim(1.0e5,1.6e5)
ax.plot(x_values_m,y_values_m, label='Moon', color = '#7c7aff')
ax.plot(x_values_p,y_values_p, label = 'Probe', color = '#FF7a90')
ax.legend(loc='upper right')
plt.show()
```

Figure 6: Extracting results for the Moon and Probe plot

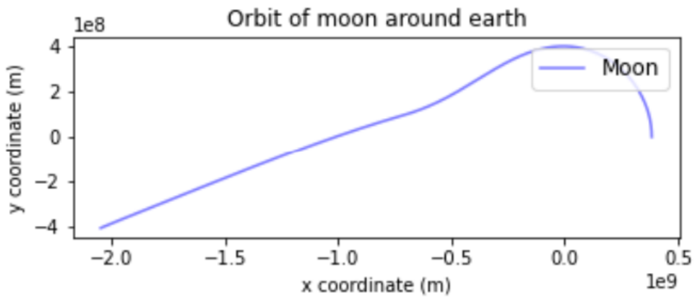
4 Results and Discussion

4.1 Moon Model

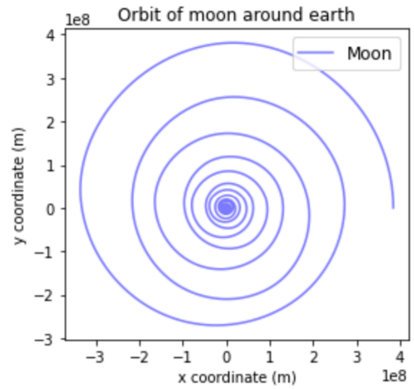
By varying the numerical parameters of the model, for example the tolerance, it's accuracy can be increased. Tolerance can be set with atol, absolute tolerance, being the correct decimal places a value will be taken to, and rtol, relative tolerance, being the number of correct digits a value will be taken to. rtol has more effect on the data, so keeping atol constant at the default value of 1e-6, rtol will be varied.

Here tolerance appears to effect the energy transfer between gravitational potential and kinetic energy of the moon. The tolerances $1e-2$ and $1e-3$ show the moon gaining kinetic energy as gravitational potential is transferred to it, spiralling into the earth at the origin. This is clearly inaccurate and so these tolerances cannot be selected for the model. However at $etol = 1e-4$ the orbits totally superimpose showing a frozen orbit, accurate to the moon's orbit in reality.

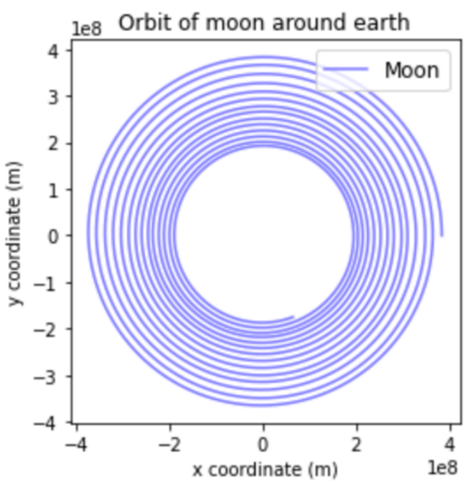
Here the absolute distance from the moon to the earth is plotted along time for the four models each with a different tolerance, as if the spiraling orbits were rolled out. The blue plot shows the lowest tolerance most accurate to reality, with the others curling away from this value over time.



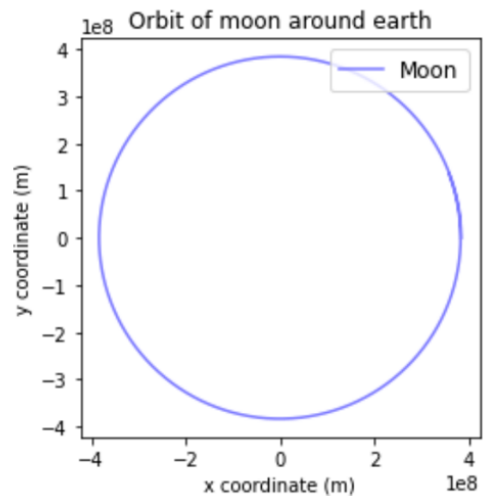
(a) Moon's orbit with $rtol = 1$



(b) Moon's orbit $rtol = 1e-2$



(a) Moon's orbit with $rtol = 1e-3$



(b) Moon's orbit $rtol = 1e-4$

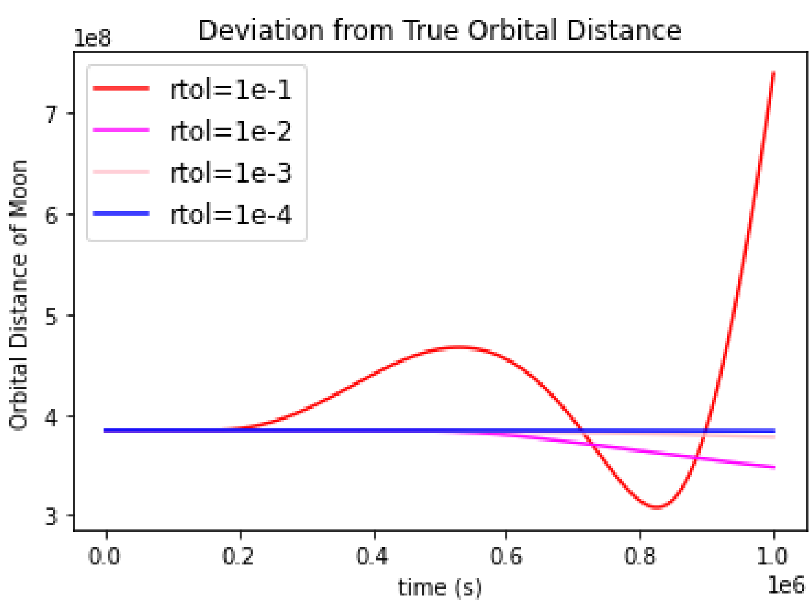
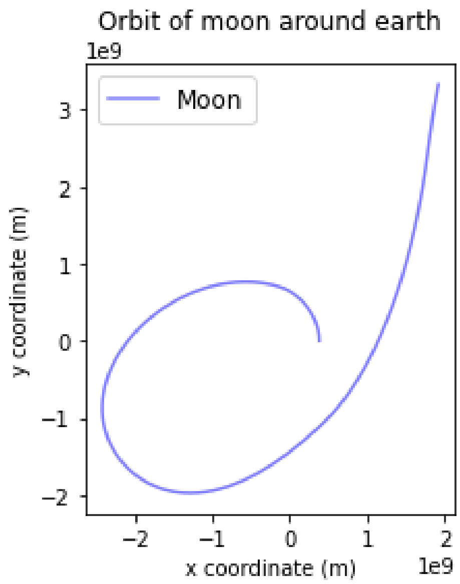


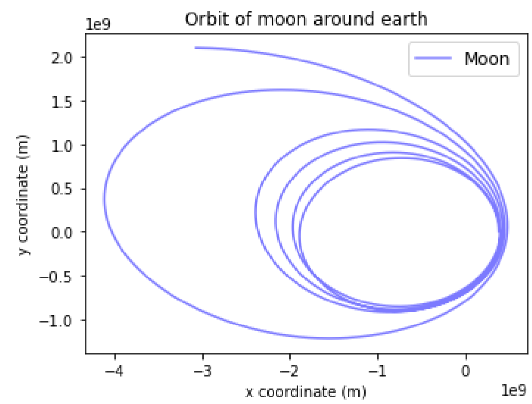
Figure 9: Deviation from Orbital Path for varying tolerances

4.2 Elliptical Orbits

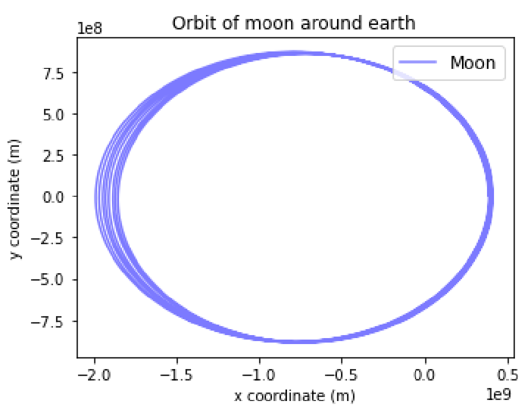
By increasing the initial velocity of the moon beyond the value calculated from the initial distance, being the moon's actual distance from the earth, the orbit it follows becomes elliptical. First increasing initial velocity by 300m/s then varying tolerance to find a tolerance perform well for both circular and elliptical orbits.



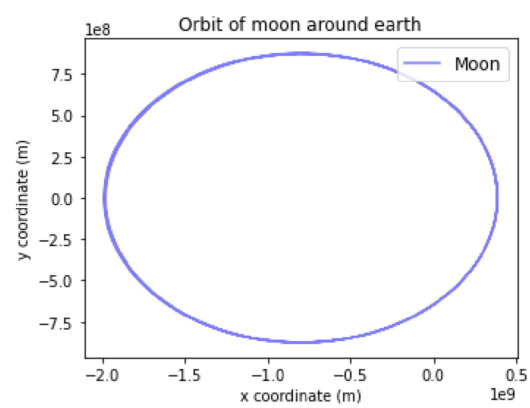
(a) Elliptical orbit for rtol=1e-1



(b) Elliptical orbit for rtol=1e-2



(c) Elliptical orbit for rtol=1e-3



(d) Elliptical orbit for rtol=1e-4

It can be seen that the orbital path furthest from the centre of mass of the Earth is where the lower tolerances begin to deviate from the true path. This shows a higher tolerance is required to model elliptical orbits.

Clearly a absolute tolerance of $1e-4$ performs well for both elliptical and circular orbits.

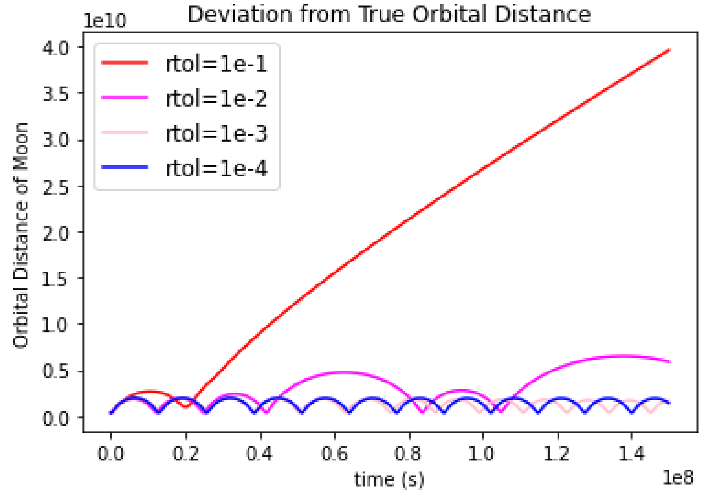
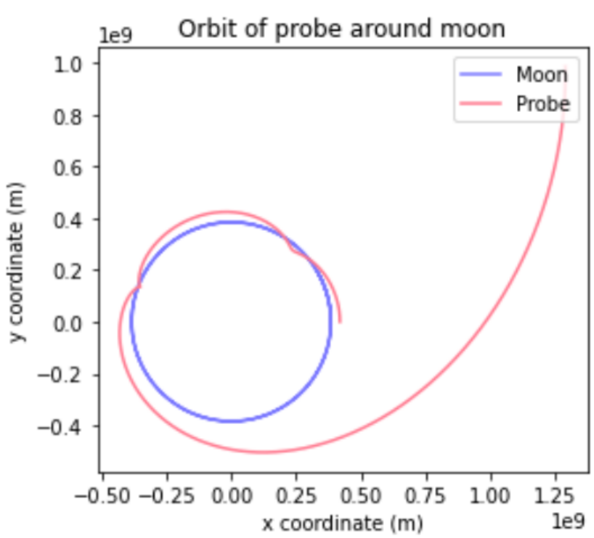


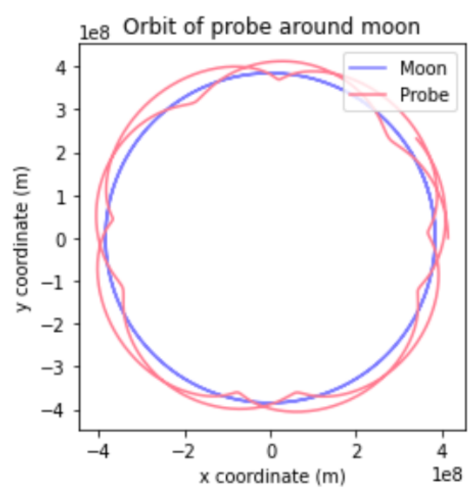
Figure 11: Deviation from Elliptical path for varying tolerances

4.3 Probe and Moon Model

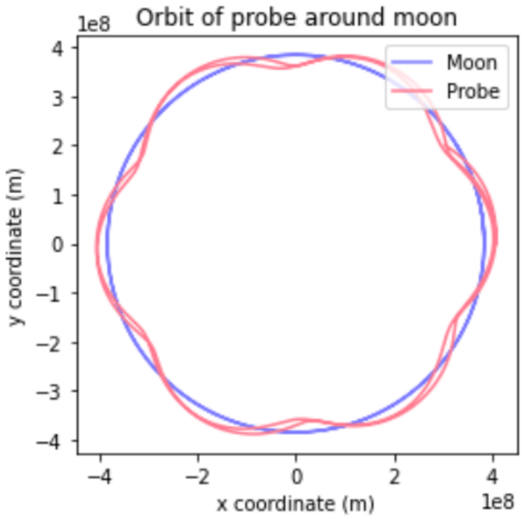
To tests the probe model the lunar altitude of orbit around the moon is varied across the magnitude 10^7 m, above which the probe escapes the orbit. This fits expectation as the initial velocity the moon has will be too much for the gravitational field of the earth to change to keep it in a stable orbit. Altitudes below this either spiral into the moon, when the gravitational attraction outweighs the centripetal force required to keep the moon in a stable orbit, or have such small time periods of orbit relative to that of the moon that the time for computing their motion exponentially grows, often resulting in the computer crashing.



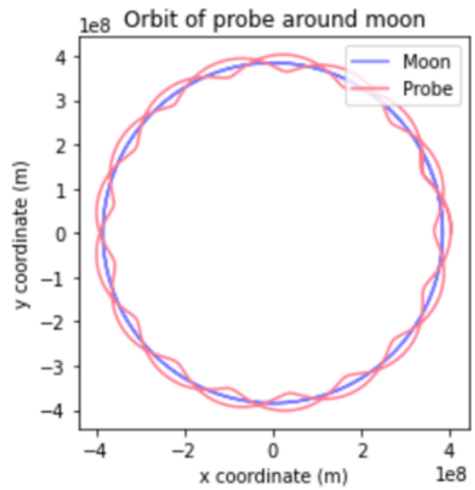
(a) Probe orbital radius set to $35e6$



(b) Probe orbital radius set to $30e6$



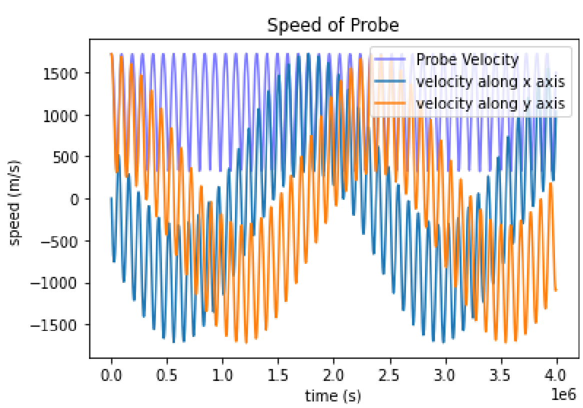
(a) Probe orbital radius set to $25e6$



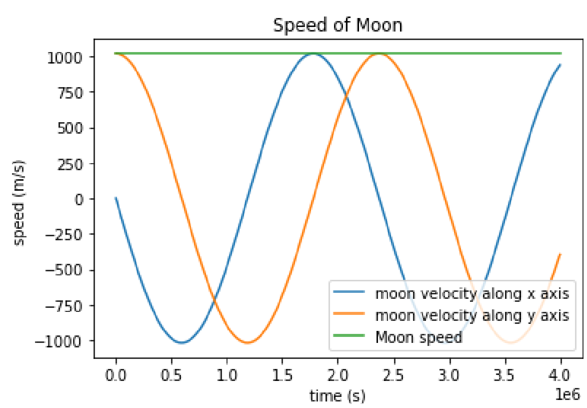
(b) Probe orbital radius set to $20e6$

As the probe's initial altitude decreases the time period of its orbit around the moon decreases. the maximum time value here was set to plot two full orbits of the sun to examine how the probe's path changes with rotations. It can be seen that as the probe's initial altitude decreases its path difference from one orbit of the sun to the next changes. From relatively low with $rpm = 30e6$ to almost perfectly superimposed. However the next decrease causes the paths to be perfectly out of sync. An altitude of $10e6$m is chosen to model the probe.

The gravitational force applied by the earth on the probe is much smaller than that of the moon despite the mass of the earth being two orders of magnitude higher than the moon. The average distance between the probe and the earth is two orders of magnitude higher as well, however this separation is squared in the gravitational force equation and so dominates the effect. In this model force from the moon has a magnitude 28.2 times more than that of the earth.



(a) x and y components of probe velocity



(b) x and y components of moon velocity

The speed of the probe is not constant in it's orbit, unlike the moon. This is as the probe has forces acting on it from more than one body. Taking a snippet of the full orbit, it can be shown how the speed of the moon is constant whereas the probe has a constant average speed but actual speed oscillates in the earth's frame. This fits with physical expectation, as when the probe is at the part of its orbit where it's moving towards the earth the gravitational attraction it experiences will accelerate it along its path, whereas when it's moving away it will decelerate. As the orbit is symmetrical it also fits with expectation that the average speed of the probe is constant, and oscillates with the time period of the probes orbit around the moon.

Overall the final plot was made with $rtol = 1e-4$, and $rpm = 8.0e6$. Limiting the axes to the top right to show more clearly the orbit, this plot is reached:

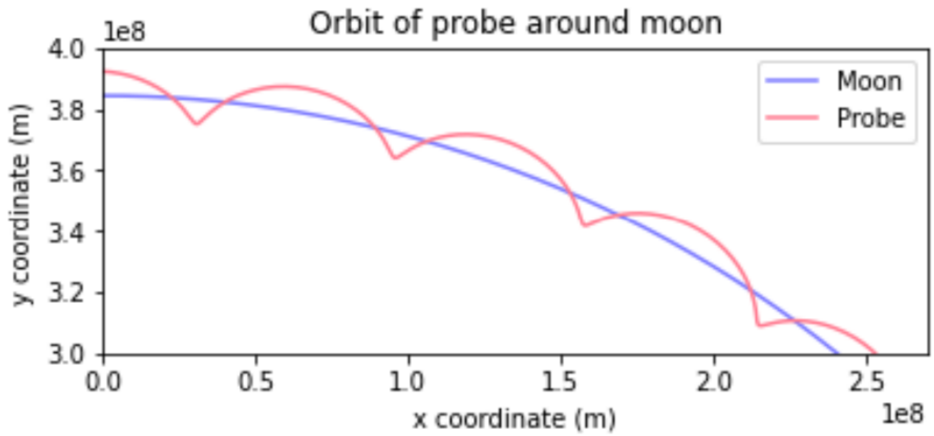


Figure 15: Probe's orbit around the Moon

To test the accuracy of the model the time period for the moon to orbit the earth can be calculated to be $T = 2\pi\sqrt{\frac{rm^3}{GMe}}$, found to 2371877.064 s. The moons true time period is 27.3 days. The accuracy of the model time period is calculated from these values to be 99.44%. An accuracy this high makes it very convincing the model is physically correct and not a numerical artifact. The remaining 0.56% inaccuracy could be due to the small deviation left by the tolerance level, or by other bodies like the sun having a gravitational effect on the moon not accounted for in this model.

5 Conclusion

The model is highly accurate to the moon's orbit. For modeling the probe its mass is irrelevant to its motion, while other parameters like initial velocity and distance vectors can be specified in the mode to model the motion of the probe. To improve the gravitational effect of the probe on the moon and earth would be included, as well as the gravitational forces from other bodies in the solar system.