Exploring Realistic Projectile Motion

Computational Physics Final Project Report

Author: James Lund

December 10, 2024

1 Introduction and Theory

Projectile motion is motion experienced by a body or particle (a projectile) that is projected at an angle to the horizontal ground with some initial velocity in the presence of a gravitational field. The trajectory of the projectile under the influence of only gravity is purely parabolic. However, considering the influence of other forces such as drag, and the Coriolis force gives a much more realistic model of the trajectory of a projectile. This project focuses on simulating realistic projectile motion on a rotating body with an atmosphere in three dimensions in Python. This provides a more accurate model of how projectiles move through the air, the study of which has significant applications in several fields. The main goal of the project is to analyze how these forces impact the projectile trajectory, range, and flight time. For simplicity, the rotating body in this case is Earth.

Drag is a force that acts on a projectile in such a way that it opposes the relative motion of the projectile. Hence, this is essentially air resistance defined by

$$F_d = Bv^2, \tag{1}$$

where B is a constant characterizing the drag effect and v is the velocity of the projectile. The Coriolis force is an example of a fictitious force that only arises in non-inertial, rotating frames of reference. Fictitious forces are ones that observers in a non-inertial frame think they feel due to the acceleration. This force is responsible for the development of rotating storms that spin anti-clockwise in the Northern Hemisphere and clockwise in the Southern Hemisphere. To find an expression of the Coriolis force, a transformation of Newton's second law to a reference frame rotating about a fixed axis, with angular velocity, $\underline{\omega}$ is required. Consider the diagram below:

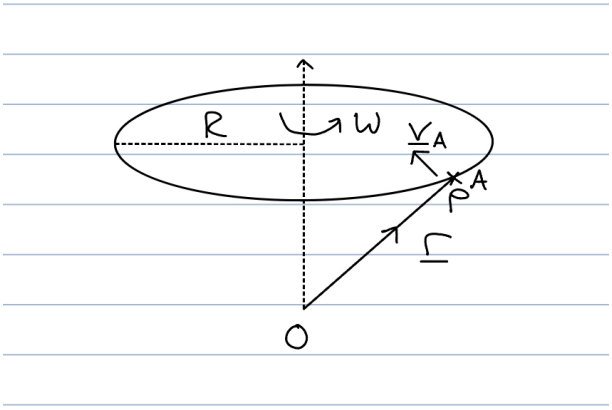

Figure 1: Diagram showcasing rotational motion of an object about a central point of origin O.

Suppose at some point in time, an object A is at point P. P has coordinate, $\underline{\mathbf{r}}$, relative to the origin, O, in an inertial frame. In this frame, A moves with velocity vector, $\underline{\mathbf{V}_A}$. P is stationary in the rotating frame but rotation at angular velocity, ω, in the inertial frame is observed. As seen in the inertial frame, the speed of P is given by

$$|V_P| = R\omega = r\omega \sin\theta. \tag{2}$$

So the direction of $\underline{\mathbf{V}_P}$ is into the page in figure 1 and thus $\underline{\mathbf{V}_P}$ can be defined as

$$\underline{\mathbf{V}_P} = \underline{\omega} \times \underline{\mathbf{r}}. \tag{-2}$$

In the inertial frame, the motion of A is

$$\underline{\mathbf{V}_A} = \frac{d\underline{\mathbf{r}}}{dt}. \tag{4}$$

A experiences a "fictional" speed say $\frac{\partial r}{\partial t}$, the rate of change of $\underline{\mathbf{r}}$ as measured in the rotating frame. The speeds in the two frames are related by

$$\underline{\mathbf{V}_A} = \underline{\mathbf{V}_P} + \frac{\partial \underline{\mathbf{r}}}{\partial t} = \underline{\omega} \times \underline{\mathbf{r}} + \frac{\partial \underline{\mathbf{r}}}{\partial t}. \tag{5}$$

Now differentiate the above with respect to time to give:

$$\frac{d\underline{\mathbf{V}_A}}{dt} = \underline{\omega} \times \frac{d\underline{\mathbf{r}}}{dt} + \frac{d}{dt}\left(\frac{\partial \underline{\mathbf{r}}}{\partial t}\right) = \underline{\omega} \times \underline{\mathbf{V}_A} + \frac{d}{dt}\left(\frac{\partial \underline{\mathbf{r}}}{\partial t}\right) = \underline{\mathbf{a_A}} \tag{6}$$

Substituting the equation for $\underline{\mathbf{V}_A}$ into the above gives

$$\underline{\mathbf{a_A}} = \underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}} + \frac{\partial \underline{\mathbf{r}}}{\partial t}) + \frac{d}{dt}\left(\frac{\partial \underline{\mathbf{r}}}{\partial t}\right). \tag{-5}$$

Dealing with the last term:

$$\underline{\mathbf{V}_A} = \underline{\omega} \times \underline{\mathbf{r}} + \frac{\partial \underline{\mathbf{r}}}{\partial t}. \tag{8}$$

Now apply equation (5) with $\underline{\mathbf{r}}$ replaced by $\frac{\partial \underline{\mathbf{r}}}{\partial t}$ so

$$\frac{d}{dt}\left(\frac{\partial \underline{\mathbf{r}}}{\partial t}\right) = \underline{\omega} \times \frac{\partial \underline{\mathbf{r}}}{\partial t} + \frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2}. \tag{9}$$

Hence equation 7 becomes:

$$\underline{\mathbf{a_A}} = \underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}} + \frac{\partial \underline{\mathbf{r}}}{\partial t}) + \underline{\omega} \times \frac{\partial \underline{\mathbf{r}}}{\partial t} + \frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2} = \frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2} + \underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}}) + 2\underline{\omega} \times \frac{\partial \underline{\mathbf{r}}}{\partial t}. \tag{10}$$

Writing $\frac{\partial \underline{\mathbf{r}}}{\partial t} = \underline{\mathbf{V}_A}'$ and $\frac{\partial^2 \underline{\mathbf{r}}}{\partial t^2} = \underline{\mathbf{a_A}}'$, as the velocity and acceleration respectively in the rotating frame, equation 10 becomes

$$\underline{\mathbf{a_A}}' = \underline{\mathbf{a_A}} - \underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}}) - 2\underline{\omega} \times \underline{\mathbf{V}_A}'. \tag{11}$$

Multiplying the above equation by the mass gives the externally applied force as seen in the inertial frame, $\underline{\mathbf{F}}$, and the apparent force, $\underline{\mathbf{F}'}$, in the rotating frame as experienced by A:

$$\underline{\mathbf{F}'} = \underline{\mathbf{F}} - 2m\underline{\omega} \times \underline{\mathbf{V}'} - m\underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}}). \tag{12}$$

[1] The $-2m\underline{\omega} \times \underline{\mathbf{V}'}$ term is the Coriolis force that acts perpendicular to $\underline{\mathbf{V}'}$ as observed in the rotating frame and perpendicular to $\underline{\omega}$ in the plane of rotation. This is speed dependent and zero if the object is at rest in the rotating frame. The $-m\underline{\omega} \times (\underline{\omega} \times \underline{\mathbf{r}})$ is another fictitious force, the centrifugal force that acts radially outwards. This is position dependent in the rotating frame. For much larger rotating frames of reference such as a rotating planet, the Coriolis force dominates the centripetal force and thus can be neglected.

The Runge-Kutta method, explained in the methodology section, is used to solve for the

equations of motion for a projectile under the influence of gravity, drag and the Coriolis forces. The equation of motion that is solved is:

$$\underline{\mathbf{F}} = m\underline{\mathbf{a}} = m(\underline{\mathbf{g}} + 2\underline{\mathbf{v}} \times \underline{\omega}) - B\underline{\mathbf{v}}^2 \tag{13}$$

where m is the projectile mass in kilograms, $\underline{\mathbf{g}}$ the acceleration due to gravity, $9.81ms^{-2}$, and B is a constant characterizing the drag effect due to air resistance. $\underline{\mathbf{v}}$ is the velocity of the projectile and $\underline{\omega}$ the angular velocity of the rotating body about its rotational axis taken to be the North Pole-South Pole axis. The following figure defines the coordinate system for three dimensional projectile motion:

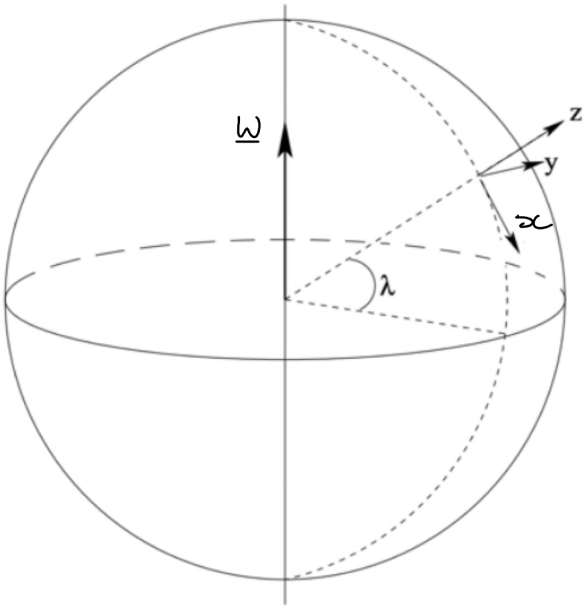

Figure 2: The coordinate system used in the project where the body is rotating clockwise about its axis of rotation.

where the x and y axes define the surface of the Earth which can be treated locally as flat and z axis points radially outward from the centre of the rotating body. λ is the latitude in degrees on the surface of the rotating body at which the projectile is launched from.

2 Methodology

Using a computer, the NumPy and MatplotLib libraries were employed in Python in a Jupyter Notebook to help produce the simulation. Then multiple functions were created to ensure that the code runs as efficiently as possible. An acceleration function was created that computes

the total acceleration from gravity, drag and the Coriolis forces. For simplicity the projectile is assumed to have a mass of one kilogram so this total acceleration is thus the total force from equation (13). Functions for the second and fourth order Runge-Kutta methods were created to calculate the trajectory of a projectile for different latitudes and launch angles. These are numerical methods for solving initial value problems for ordinary differential equations which in this case is solving the equation of motion. For the second order Runge-Kutta, initial acceleration is calculated and then midpoint estimates for the velocity and position are calculated. The acceleration is then recalculated at the midpoint and then the velocities and positions are updated at a later time. This results in a much more accurate calculation of the trajectory of the projectile compared with using the Euler-Cromer method which does not use midpoint estimates. The second order Runge-Kutta is set up by:

$$x(t + \Delta t) = x(t) + f(x', t')\Delta t \tag{14}$$

where x' and t' are given by:

$$x' = x(t) + \frac{1}{2} f(x(t), t)\Delta t \tag{15}$$

$$t' = t + \frac{1}{2}\Delta t \tag{16}$$

[2] Here, $\frac{dx}{dt}$ is approximated as the value $f(x', t')$ where t' is the midpoint time and x' is the approximated value of x at t'. Δt is the time step and $f(x(t), t)$ is the differential equation to be solved. For the fourth order Runge-Kutta, the equations become more complex since there are four steps involved [3] whereas there are only two for the second order method. Visually, the numerical methods work as follows:

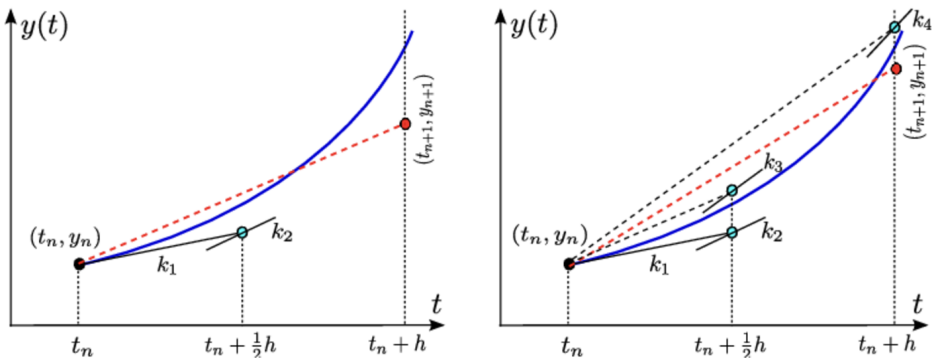

Figure 3: From left to right: Runge-Kutta second and fourth order methods showing the steps.[4]

Hence this gives a much more accurate and stable trajectory calculation for larger timesteps. Despite the accuracy however, these methods are very computationally expensive. A simulation function was created that employs either one of the second and fourth order Runge-Kutta methods to return the computed positions in three dimensions. A plotting function was created to produce an interactive plot of the trajectory in three dimensions from the positions calculated in the simulation function.

Various parameters which can be altered are defined in a new code cell. These include the values of B, the coefficient of drag, the gravitational acceleration, g, the angular velocity of the rotating body, $\underline{\omega}$, and the latitude in degrees. The Earth's angular velocity is roughly 7.3×10^{-5} rads^{-1} found by dividing 2π by 86400, the number of seconds in a day. Initial conditions can be defined and altered such as the launch velocity, the launch angle, and the starting position. The initial velocity vector is defined in such a way that the projectile is launched in the eastward direction relative to the Earth's surface

$$\underline{\mathbf{v_0}} = (v_0 \cos\theta, 0, v_0 \sin\theta) \tag{17}$$

where v_0 is the launch velocity, and θ, the launch angle. This simulates the effect of launching a projectile in a direction parallel to the Earth's surface at different latitudes. Finally, a timestep, Δt and total simulation time, T in seconds, was defined in order for the simulation to run and be plotted. $\Delta t = 0.01$ and $T = 20$ in this case. Once the projectile motion was plotted, three more functions were created to plot the maximum range and altitude of the projectile as a function of the latitudes and the launch angles. The 4th order Runge-Kutta method is employed here to accurately find the most optimal set of launch angles and latitudes for range and altitude maximization. From all these different plots, projectile motion in the presence of drag and Coriolis forces can be analyzed in great depth.

3 Results and Analysis

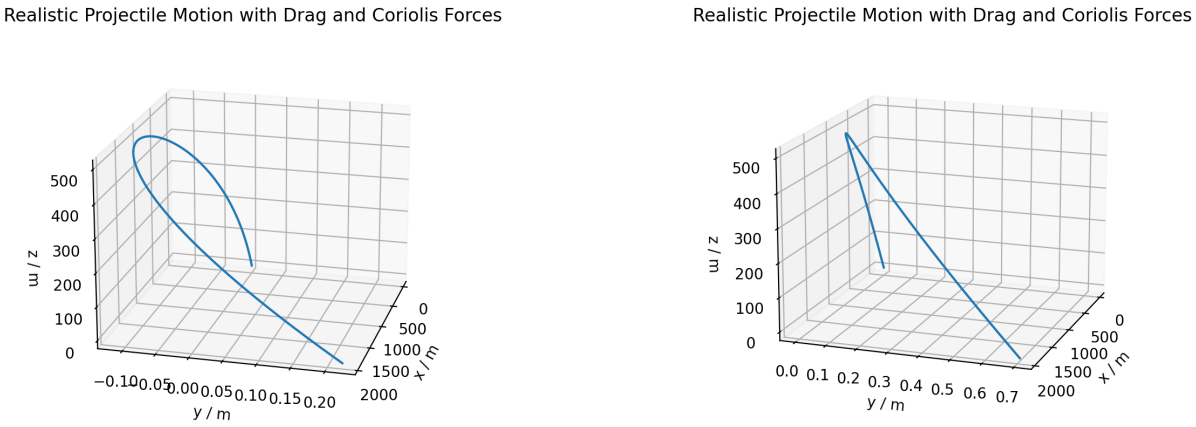

Figure 4: From left to right: realistic projectile motion on Earth at latitudes of 15° and 30° respectively with an initial velocity of 100 ms^{-1}.

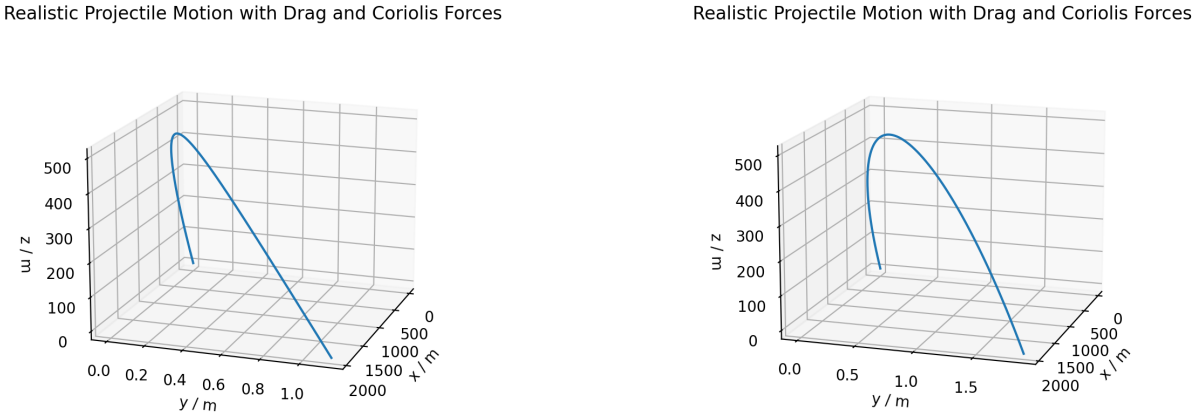

Figure 5: From left to right: realistic projectile motion on Earth at latitudes of 45° and 90° respectively with an initial velocity of 100 ms^{-1}.

From figures 4 and 5, the deflection of the projectile increases with increasing latitude. The final y coordinate increases with latitude, implying a greater deflection. This is consistent as the Coriolis force is directly proportional to the angular velocity, $\underline{\omega}$, which is largest at the poles and smallest, almost zero, at the equator. Hence the force is maximized at the poles and zero at the equator.

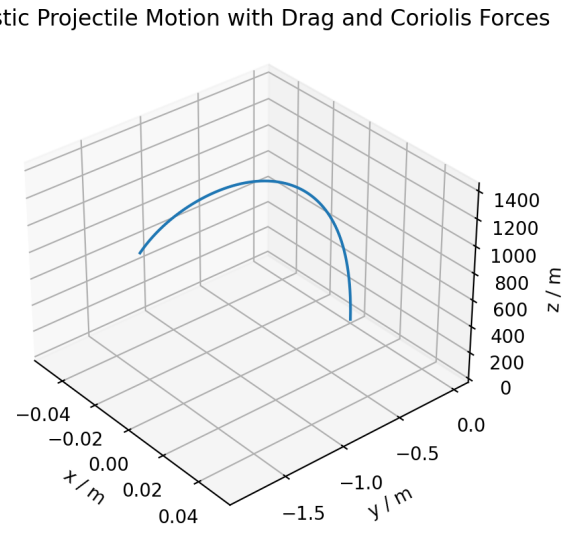

Figure 6: Realistic projectile motion at a latitude of 0° with an initial velocity of $100\text{m}s^{-1}$.

Zero Coriolis force results in the trajectory of the projectile not being deflected in space. This is seen above in figure 6 where drag is the only force now acting on the projectile.

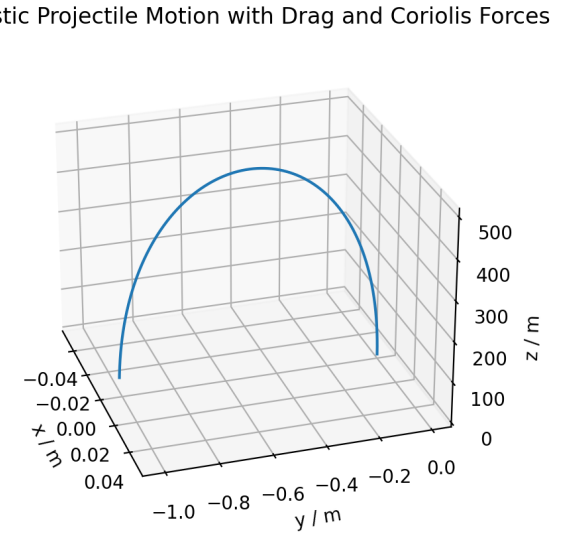

Figure 7: Realistic projectile motion at a latitude of 0° with an initial velocity of $100\text{m}s^{-1}$ and no acting drag force.

Setting the drag coefficient $B = 0$, at the equator produces symmetric parabolic projectile motion in figure 7 with no deflection as expected with idealized case.

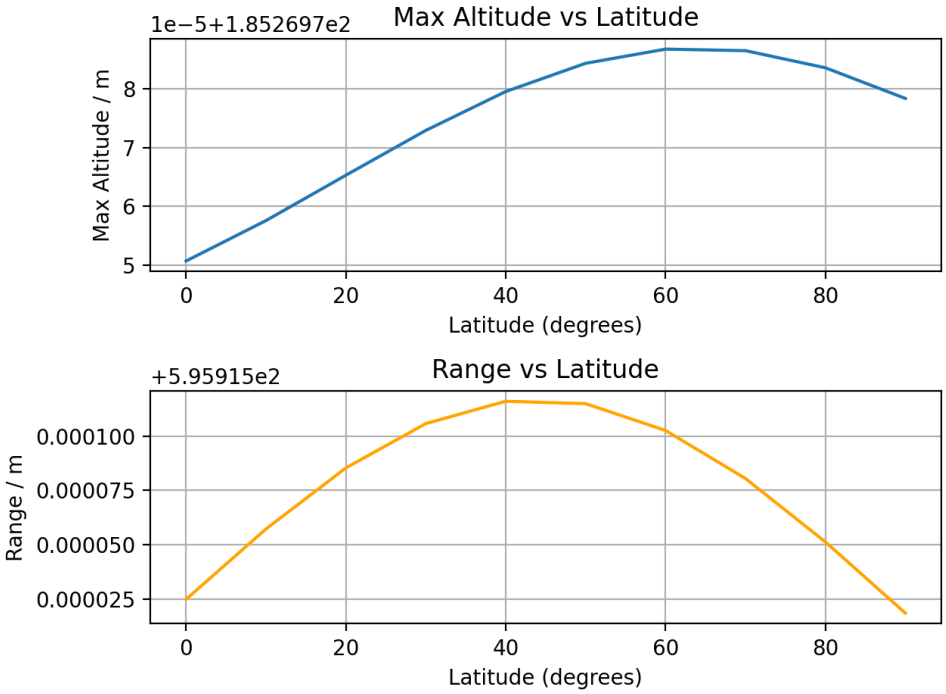

Figure 8: Maximum altitude and range as a function of launch latitude with $v_0 = 100$ ms^{-1}.

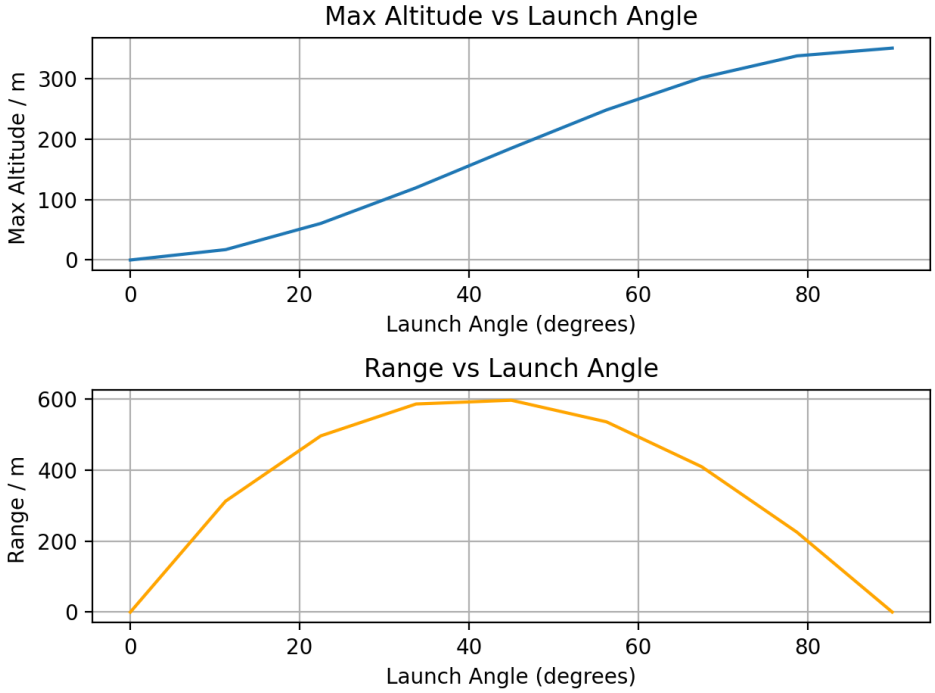

Figure 9: Maximum altitude and range as a function of launch angle with $v_0 = 100$ ms^{-1}.

From figure 8, the altitude is maximized at a latitude of roughly 70° and the range is maximized at a latitude of 40°. From figure 9, the altitude is maximized at a launch angle of 90° as expected. The range is maximized at a launch angle of 45°, which is the same as that from projectile motion in the absence of air resistance. These relationships between the maximum altitude and maximum range with the latitude and launch angle appear to be sensical. The projectile trajectories are significantly altered in the presence of drag and Coriolis compared to the idealized case.

However, due to time constraints mainly with debugging the simulation code, these relationships could not be verified further. The scope of the analysis was limited to a set of the same initial conditions as a result of time. Furthermore, with more time generally, the simulation could be refined further to produce more plots showing realistic projectile motion with heavier projectiles and larger angular velocity vectors on other rotating bodies. The initial conditions could also be varied and other factors such as accounting for the surface velocity due to the rotating body and the variation air density with altitude were not explored. Future work could explore these areas in more detail to further improve the accuracy and realism of the simulation.

4 Conclusion

To conclude, a simulation modelling realistic projectile was successfully developed in Python, incorporating both drag and Coriolis forces. The second and fourth order Runge-Kutta methods were used to solve for the equations of motion and from this, the model shows that the projectiles are subject to deflection at non-zero latitudes, altering the trajectory. This showcases the importance of considering other external forces in modelling projectile motion in the real world. On the other hand, due to time constraints, further aspects were not investigated including factoring in surface velocity due to rotating. Future work would integrate these further refinements to the simulation, in order to produce a model of projectile motion as realistic as possible. Despite this, the project establishes the ground work for simulating and gaining insights into realistic projectile motion.

References

[1] A. Persson, "The coriolis effect," *History of Meteorology*, vol. 2, pp. 1–24, 2005.

[2] N. Giordano and H. Nakanishi, *Computational physics*. Pearson/Prentice Hall, 2006.

[3] D. K. Zhang, "Discovering new runge-kutta methods using unstructured numerical search," 2019.

[4] M. Fadlisyah, *A Rewriting-Logic-Based Approach for the Formal Modeling and Analysis of Interacting Hybrid Systems*. PhD thesis, 09 2014.