BPhO Computational Challenge 2024

Umair Malik

August 1, 2024

Abstract

This paper describes my thoughts and actions in regards to this year's Computational Challenge, which focused primarily on the physics of Projectile Modelling. The tasks progressed steadily in complexity and accuracy, painting an increasingly more developed depiction of projectile dynamics; revealing its use in real-world applications such as high-level sports, civil engineering, and even in the military, as explored by the extension task, which incorporated intercontinental launches.

The project itself was completed using a backend of Python to generate the required data reliably, using a framework of the NumPy and Math libraries for the bulk of the tasks, with additional support from SciPy for the more complex mathematical functions, namely integration. The frontend utilised Streamlit, a visualisation library that allowed for interactivity from the user, through slider and input widgets.

The equations and techniques were obtained from Dr A. French's resources on eclecticon.info and from his broadcasts through the BPhO organisation.

Contents

1 Task 1 - Projectile Motion Model

The first task was very simplistic as it used a drag-free projectile model that did not account for the variable acceleration that would be experienced by an object in motion, however it is still reasonable accurate - using the following equations:

$$x = u_x t$$
$$y = h + u_y t - \tfrac{1}{2} g t^2$$
$$v_x = u \cos\theta$$
$$v_y = u \sin\theta - gt$$
$$v = \sqrt{v_x^2 + v_y^2}$$

These could be used to procedurally generate the position of the projectile over a period of time. Using a while loop to verify that the projectile is above 0m, the program will provide data as arrays that can be processed to produce graphics such as in Fig. 1.

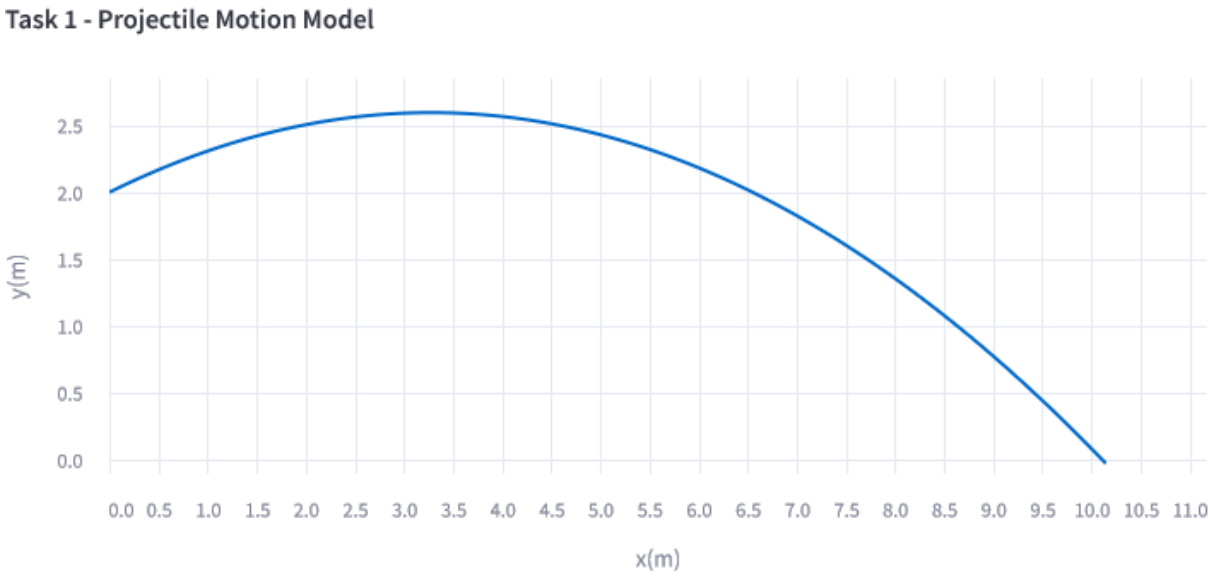

Figure 1:

2 Task 2 - Projectile Apogee

Moving onto the second task, we develop the model into a more rigorous device, using a more exact "analytical" model. We calculate the max range using

$$R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)$$

and then dividing the distance from the origin into equal slices. With these x coordinates, the corresponding y values can be calculated with the formula

$$y = h + x\tan\theta - \frac{g}{2u^2}\left(1 + \tan^2\theta\right)x^2$$

Furthermore, the apogee can be found using the following equations:

$$x_a = \frac{u^2}{g}\sin\theta\cos\theta$$

$$y_a = h + \frac{u^2}{2g}\sin^2\theta$$

This data is processed and displayed in Fig. 2.

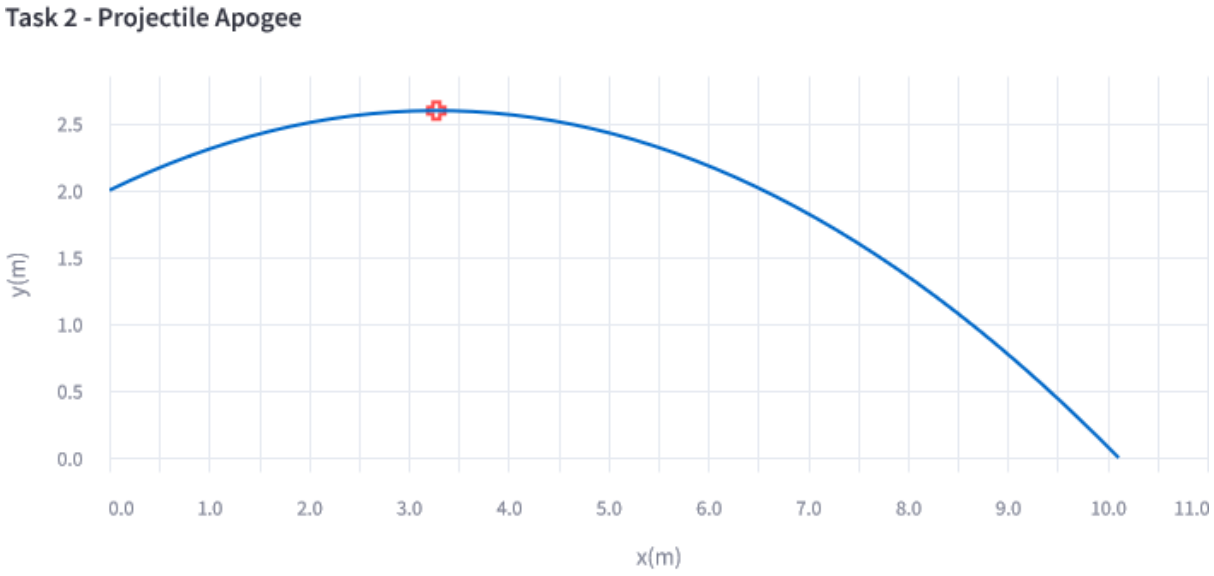

Figure 2:

3 Task 3 - Projectile to hit point

Task 3 involves taking an input of a target x and y coordinate, with the aim of finding the different extremes of trajectories to reach there. We can find the highest and lowest paths the ball can take by finding the roots of the equation

$$y = h + x\tan\theta - \frac{g}{2u^2}\left(1 + \tan^2\theta\right)x^2$$

We can rearrange it and set one side to 0 as such:

$$a\tan^2\theta + b\tan\theta + c = 0$$

Where:

$$a = \frac{g}{2u^2}x^2$$
$$b = -x$$
$$c = y - h + \frac{gx^2}{2u^2}$$

With this, we can use the quadratic formula to find the solutions for the angles,

$$\theta_{\pm} = \tan^{-1}\left(\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}\right)$$

Giving us two angles that correspond to the high ball and the low ball. Their trajectories can be generated in a similar manner to Task 2, using a range of x values and mapping them to their respective y value by using the first formula on this page.
To get the trajectory that has the lowest initial velocity, we can use the equations:

$$u_{min} = \sqrt{g}\sqrt{y + \sqrt{x^2 + y^2}}$$

And:

$$\theta = \tan^{-1}\left(\frac{y + \sqrt{x^2 + y^2}}{x}\right)$$

Obtaining the velocity and the angle respectively, which are used to plot the third trajectory, as shown in Fig. 3.

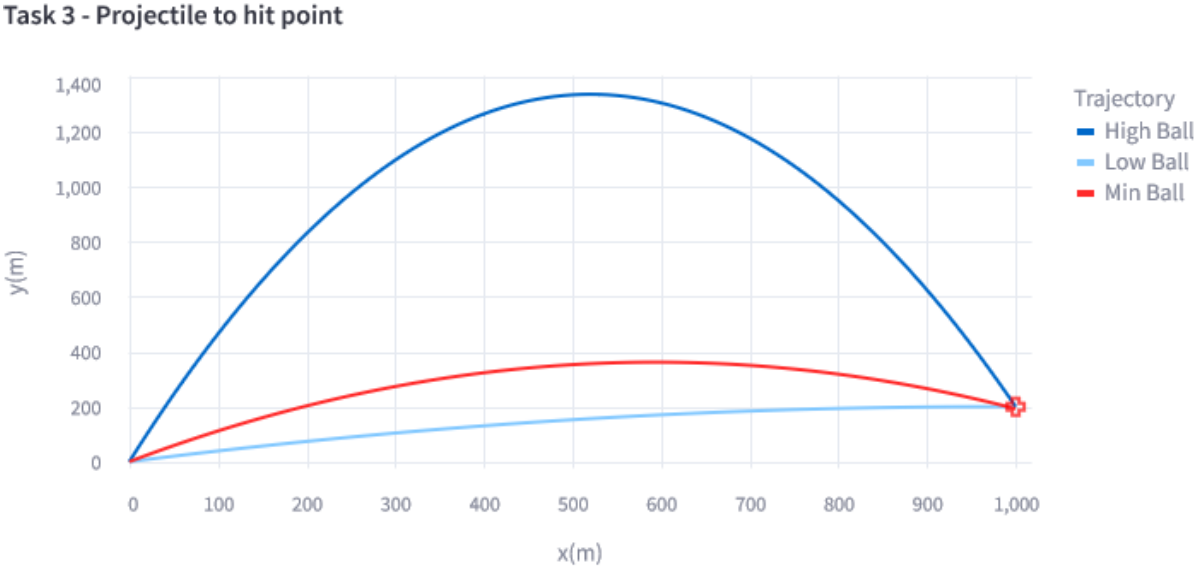

Figure 3:

4 Task 4 - Maximising Distance

The fourth task requires us to create a projectile model that compares the trajectory of a set of given inputs and the trajectory of one maximising horizontal distance whilst using the same launch height and launch speed.

In order to work out the path that maximises distance, we can first see what happens when the launch height is set to 0. In that case, range can be worked out using:

$$R = \frac{u^2}{g}\sin(2\theta)$$

Meaning that distance is maximised when $\sin(2\theta) = 1$, thus $\theta = 45°$.
However, this needs to extended outside of the case where launch height is zero, so we must use the general equation for range with is:

$$R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)$$

To find the optimal angle, the first differentiation must equal 0 as so:

$$\frac{d}{d\theta}\left(\frac{Rg}{u^2}\right) = 0$$

Which works out to be that:

$$\theta = \sin^{-1}\left(\frac{1}{\sqrt{2 + \frac{2gh}{u^2}}}\right)$$

With this angle, we can use the following equation to map out the curve

$$y = h + x\tan\theta - \frac{g}{2u^2}\left(1 + \tan^2\theta\right)x^2$$

Which we can display as Fig. 4.

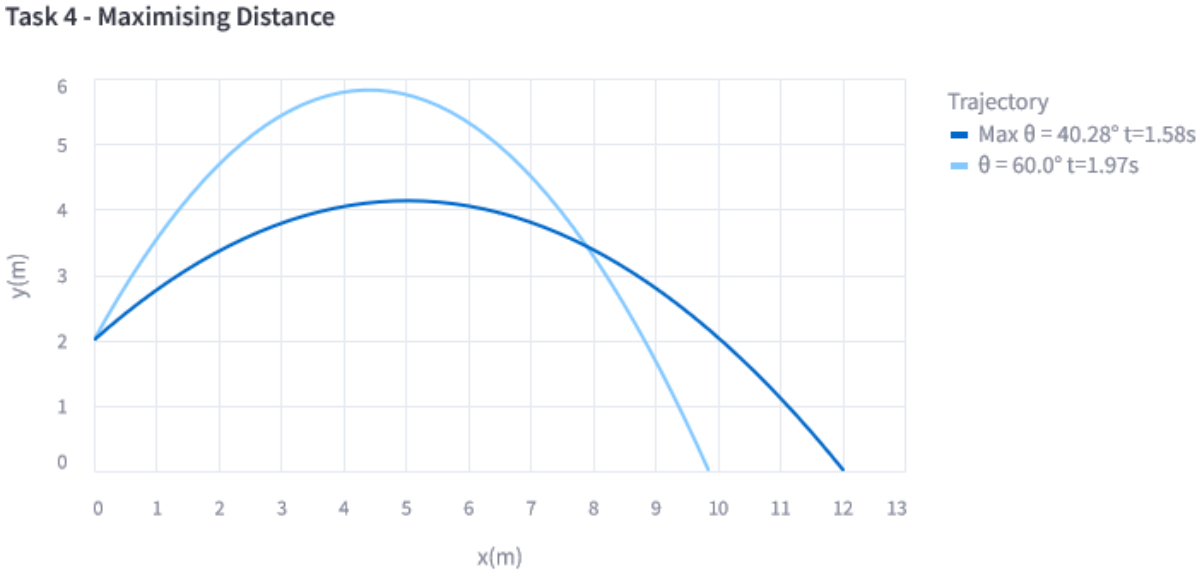

Figure 4:

5 Task 5 - Bounding Parabola

Moving on to the fifth task, we aim to build upon the work of task three, where we calculated three trajectories that pass through a given point: high ball, low ball and minimum initial velocity. The first two obtained by finding where the differentiation of the following equation is equal to 0:

$$y = h + x\tan\theta - \frac{g}{2u^2}\left(1 + \tan^2\theta\right)x^2$$

Minimum velocity and its angle are found using use the equations:

$$u_{min} = \sqrt{g}\sqrt{y + \sqrt{x^2 + y^2}}$$

And:

$$\theta = \tan^{-1}\left(\frac{y + \sqrt{x^2 + y^2}}{x}\right)$$

Furthermore, for this task we bring in the maximum distance equation from task 4, which is found via:

$$\theta = \sin^{-1}\left(\frac{1}{\sqrt{2 + \frac{2gh}{u^2}}}\right)$$

Finally, we need to create a bounding parabola that encompasses our first four curves. This can be done by finding the positive discriminant of this equation:

$$gx^2\tan^2\theta - 2u^2x\tan\theta + 2u^2y + gx^2 = 0$$

Which becomes:

$$y = \frac{u^2}{2g} - \frac{g}{2u^2}x^2$$

This needs to be then adjusted for the offset caused by our launch height, which transforms the equation in to:

$$y = \frac{u^2}{2g} - \frac{g}{2u^2}x^2 + h$$

Giving us our formula for the bounding line, letting us plot out Fig. 5.

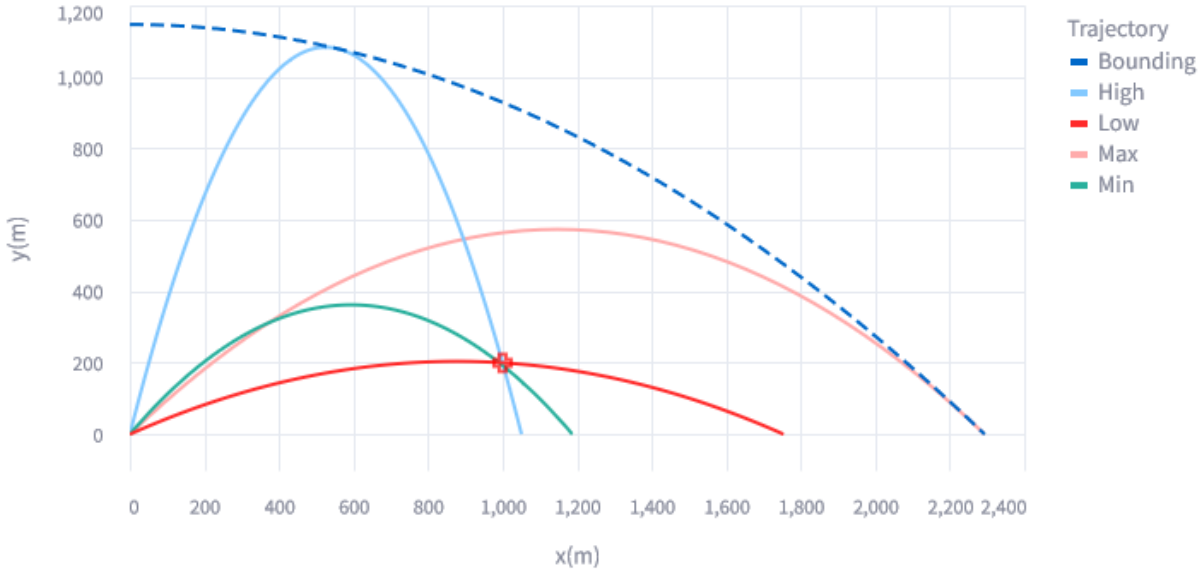

Figure 5:

6 Task 6 - Distance Traveled

The sixth task gets us to find the distance travelled by the projectile and then investigate how that changes as the angle increases, with an aim of finding the maximised distance travelled.
On the special case that the launch height is zero, we can use this simplified equation:

$$s = \frac{u^2}{g}\left(\ln\left(\frac{1+\sin\theta}{\cos\theta}\right)\cos^2\theta + \sin\theta\right)$$

However, not all of our projectiles can be fit into this generalisation, thus we need to integrate the equation for our projectile:

$$y = h + x\tan\theta - \frac{g}{2u^2}\left(1+\tan^2\theta\right)x^2$$

This can be done through this general equation:

$$s = \int_0^R \sqrt{(dx)^2 + (dy)^2}$$

Through some algebraic manipulation we can get:

$$s = \int_0^R \sqrt{1 + \left(\tan\theta - \frac{gx}{u^2}(1+\tan^2\theta)\right)^2}\,dx$$

With R being the range of the projectile calculated with this equation:

$$R = \frac{u^2}{g}\left(\sin\theta\cos\theta + \cos\theta\sqrt{\sin^2\theta + \frac{2gh}{u^2}}\right)$$

We can then display these distances from our integrations along side the plots of the trajectories, as seen bellow in Fig. 6.

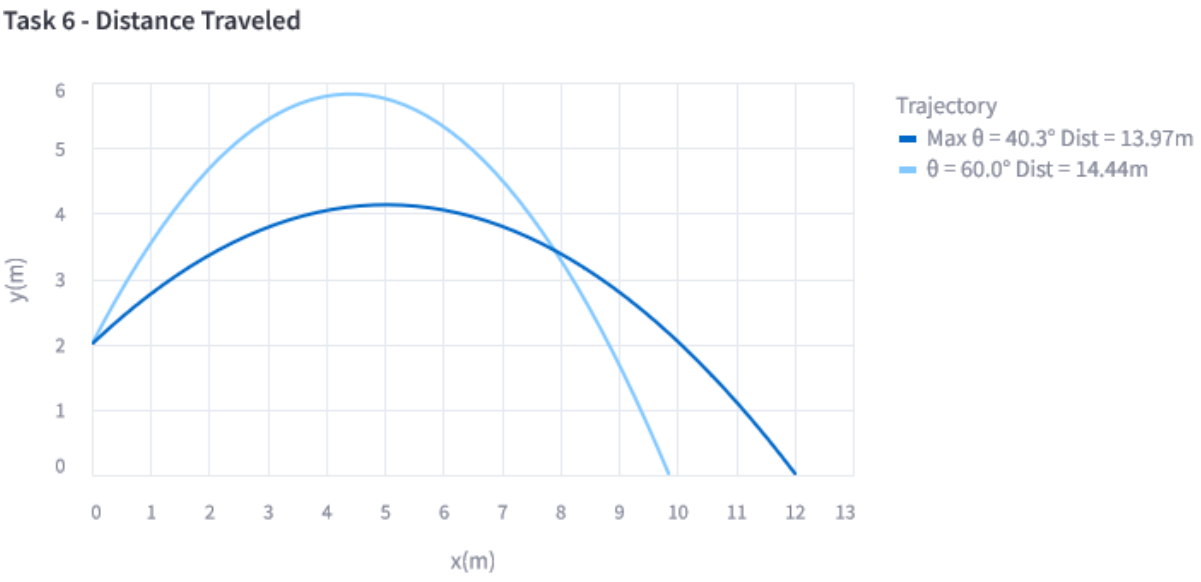

Figure 6:

7 Task 7 - Min-max points

Task 7 involves the investigation of the relationship of a projectile's range over a period of time and how that relationship changes as launch angle changes.
We can see that for a certain range of angles, there is a local maximum in range, followed by a local minimum that precedes the graph diverging into a mathematical infinity.
First, we need to find the range of the projectile for a given co-ordinate, which we can use this definition of range:

$$r^2 = x^2 + y^2$$

By substituting in the equations for x and y, we can get the following equation for range:

$$r = \sqrt{u^2t^2 - gt^3u\sin\theta + \frac{1}{4}g^2t^4}$$

We can find these significant points of maximum or minimum using the second derivative of the function, thus the following equation:

$$t_{\pm} = \frac{3u}{2g}\left(\sin\theta \pm \sqrt{\sin^2\theta - \frac{8}{9}}\right)$$

$$\theta \geq \sin^{-1}\left(\frac{2\sqrt{2}}{3}\right)$$

This means that at ≈ 70.5, instead of getting two points of significance, we only get one: an inflection point. Furthermore, at angles smaller that 70.5 there are no stationary points at all.

We can now plot out the range in respect to time, with the second equation on this page. Going further, we can also plot a graph of y against x whilst also having the stationary points being plotted as seen in the 2 figures below.

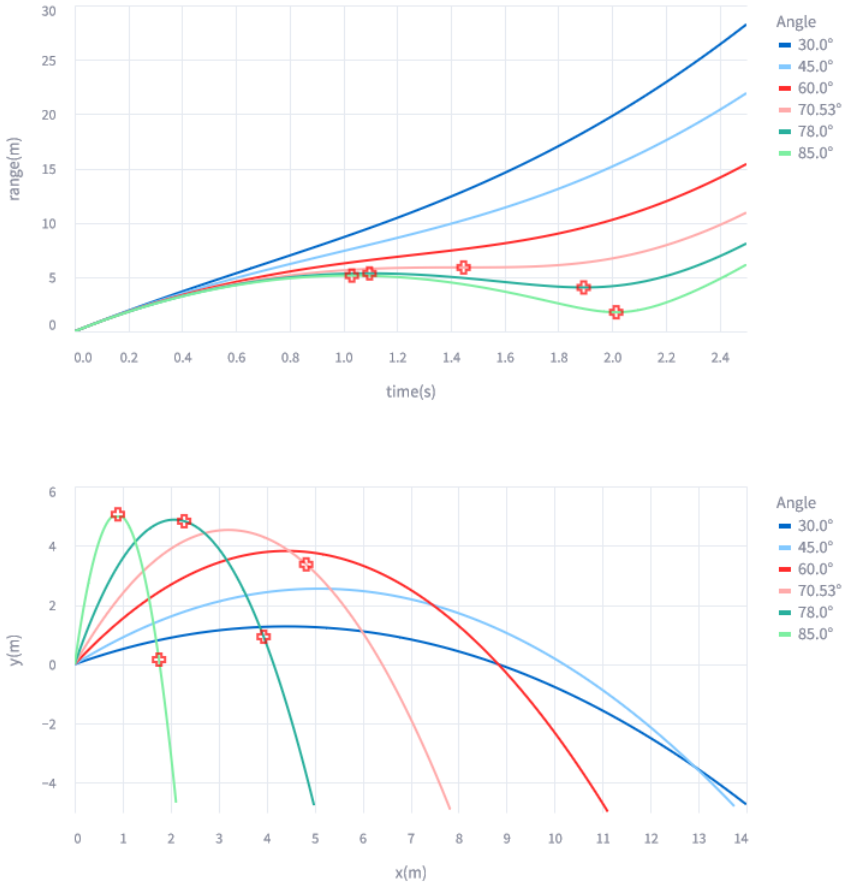

Figure 7:

8 Task 8 - Projectile Bounce

In Task 8, we are set with the goal of plotting the motion of the projectile as it bounces up and down on the floor.

We will use the velocity Verlet algorithm - a method to calculate the position and velocity of a particle over a period of time.
The horizontal and vertical positions are calculated with the following formulas:

$$x(t + \Delta t) = x(t) + v_x(t) \cdot \Delta t + \frac{1}{2} a_x(t) \cdot (\Delta t)^2$$
$$y(t + \Delta t) = y(t) + v_y(t) \cdot \Delta t + \frac{1}{2} a_y(t) \cdot (\Delta t)^2$$

We can then update the vertical and horizontal velocities using the following equations:

$$v_x(t + \Delta t) = v_x(t) + \frac{1}{2}(a_x(t) + a_x(t + \Delta t)) \cdot \Delta t$$
$$v_y(t + \Delta t) = v_y(t) + \frac{1}{2}(a_y(t) + a_y(t + \Delta t)) \cdot \Delta t$$

When the projectile hits the ground i.e. $y <= 0$, we need to adjust the velocity of the object with the following equation:

$$v_y(t) = -C \cdot v_y(t)$$

With C being the coefficient of restitution, which is defined as the ratio between the speed of the object after impact and the speed of the object before impact.
With this data, we can plot out the path of the bounding projectile as seen in Fig. 8.

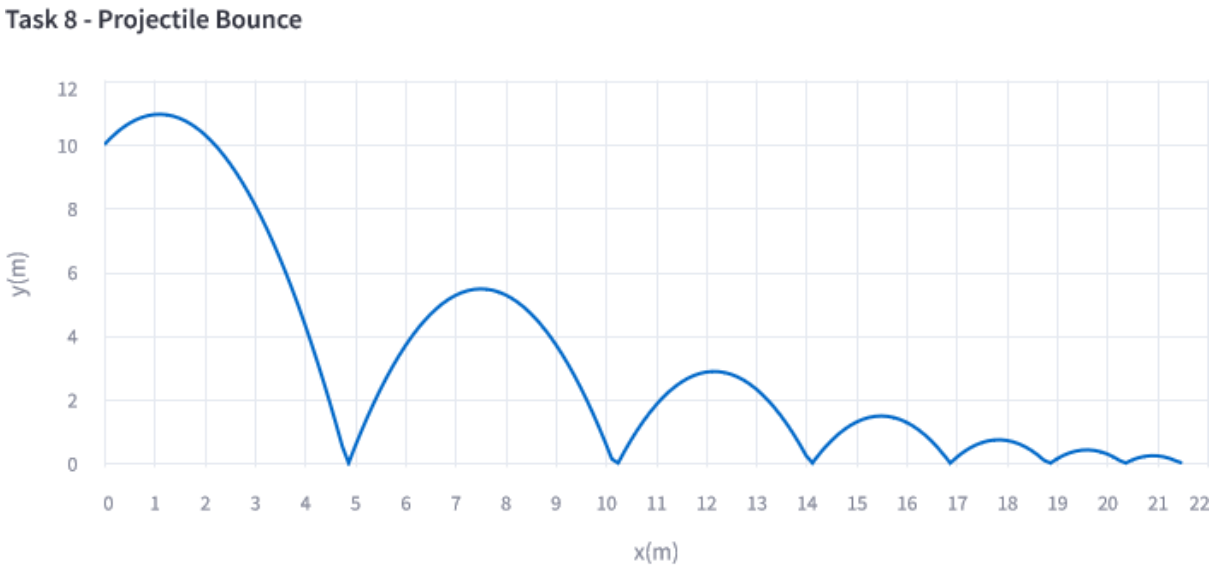

Figure 8:

9 Task 9 - Air resistance model

For the final of the nine main tasks, we have to incorporate air resistance into our projectile model. This means that we now have to deal with variable acceleration. The first thing to calculate is the air resistance factor, which we can do with this equation:

$$k = \frac{\frac{1}{2}C_D \rho A}{m}$$

We can use this factor to calculate the effect that air resistance will have on both the x acceleration and the y acceleration, as seen in these equations:

$$a_x = -\frac{v_x}{v_r} k v_r^2$$
$$a_y = -g - \frac{v_y}{v_r} k v_r^2$$

With v_r being the resultant velocity after the combination of the x and y velocities.
Now that we have the adjusted accelerations we can use the same Verlet equations from the previous task to calculate the positions:

$$x(t + \Delta t) = x(t) + v_x(t) \cdot \Delta t + \frac{1}{2} a_x(t) \cdot (\Delta t)^2$$
$$y(t + \Delta t) = y(t) + v_y(t) \cdot \Delta t + \frac{1}{2} a_y(t) \cdot (\Delta t)^2$$

Which we can then plot onto a graph with the trajectory without air resistance as such in Fig. 9:

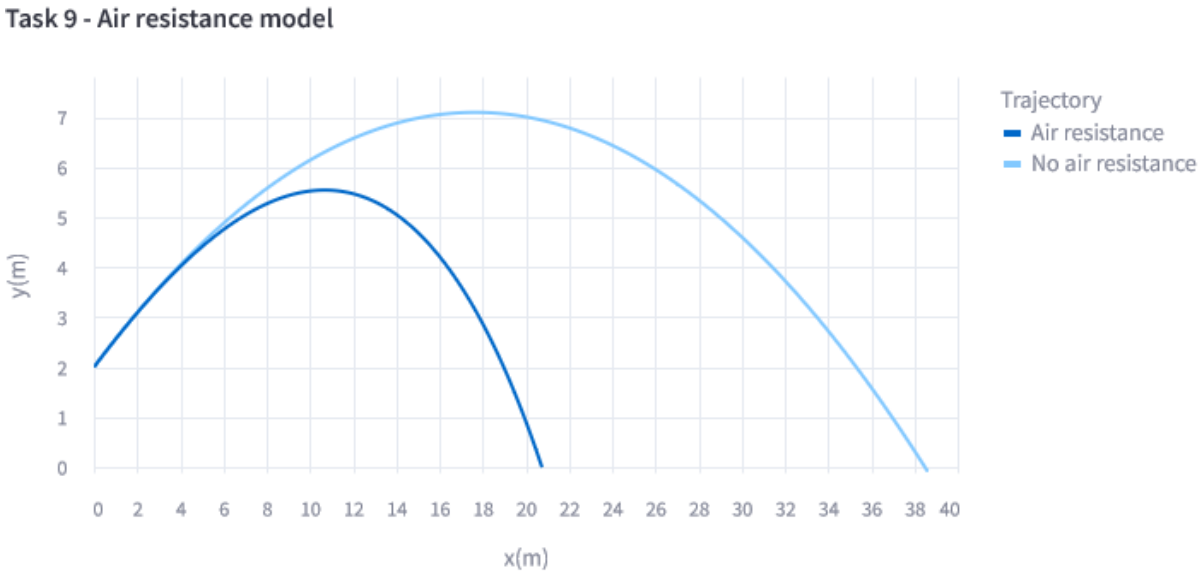

Figure 9: