Investigation into numerical modelling of ballistics using iterative methods in Python

Kacper Slodek
School of Physics, University of Bristol
(Dated: April 2023)

Numerical modelling through the Euler method has been used to show the trajectories of ballistic motion in one and two dimensions. The role of drag on the motion of a launched cannonball was investigated showing the dependence of displacement on air resistance. Furthermore, the angle of launch needed to hit a target was computed for various cases of wind resistance.

I. INTRODUCTION

A. Ballistics

Projectile motion is a form of movement of an object projected in a gravitational field; the topic of objects moving in Earth's gravitational field was studied since ancient times. One of the most prominent early contributors to the study was Aristotle [1], who attempted to describe any motion by postulating that if an object is moving, another thing must be moving it. Using these theories, and using his own observations, an Italian engineer Niccolò Fontana Tartaglia [2] illustrated what happens to a projectile when it is launched from a point on the ground. This theory divides the motion of the object into three distinct parts: a linear part after the projectile is launched, named the 'violent motion', a curved path where the projectile loses altitude, called the 'mixed' or 'crooked' motion and finally a portion of the trajectory where the projectile falls back to the ground with no horizontal velocity as illustrated in Figure 1.

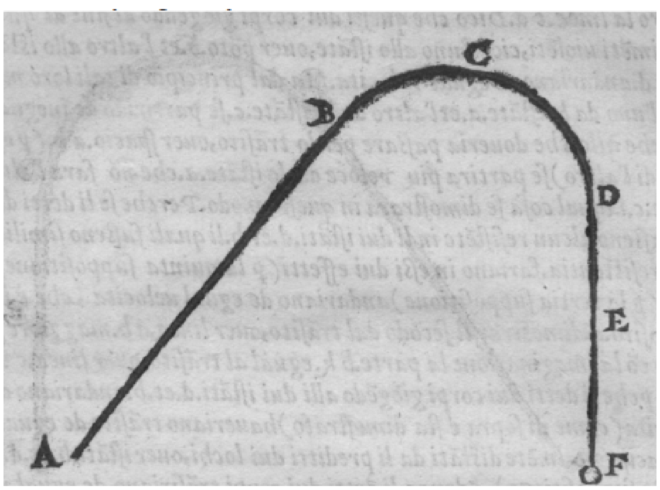

FIG. 1. Tartaglia's attempt to illustrate the motion of a projectile as fired from a cannon, the trajectory is sub-divided into three parts: 'violent', 'mixed' and 'natural' motions. [3]

Aristotle's attempt to explain why a projectile does not continue infinitely upwards, but rather falls to the ground, included two main factors: an 'impulsion' that gets used up to divide the air where the projectile makes its path, and 'reciprocal interaction' between the object and the medium where some power gets dissipated [4]. Even with the limited knowledge of science at the time, Aristotle predicted the mechanisms of friction between a launched object and the air around it.

The importance of projectile motion traces back to as early as 1346 when the English used them at the Battle of Crécy [5]. The understanding of how the cannonball moves through the air and the ability to predict with confidence where it will land was crucial from then on during numerous battles all over the world. Figure 1 portrays the theory of a projectile falling vertically downwards at the end of its flight (the 'natural' motion), as this was what many soldiers experienced - the cannonballs appeared as though they came directly from above; the case was similar to a hail of arrows in battles before cannons were invented [6].

The first person to accurately describe the shape of a trajectory of a launched object was Galileo who postulated that if a projectile has a constant horizontal velocity, the path traced out when it travels is a parabola [7]:

$$z = z_0 - gt^2, \tag{1}$$

where z and z_0 are the horizontal and initial displacements of the object in metres, g is the acceleration due to gravity in metres per second squared, and t is the time in seconds. This theory however does not take air resistance into account, which depending on the friction coefficient, can change the shape of the trajectory in a major way.

B. Numerical modelling

Nowadays, the immense computational power at the fingertips of anyone who possesses a modern computer can confirm whether Tartaglia's or Galileo's theories about the motion of an object moving under gravity are correct. The average CPU (Central Processing Unit) speed, as of 2022 is $\approx$ 4GHz in a quad-core processor [8]; that is four different processing units, each executing four billion operations each second. With these specifications, and some programming knowledge, the motion of a projectile can be modelled numerically, or a differential equation can be solved in a matter of seconds and a graph of suitable variables can be plotted and output to the user. Euler's method was used by Paola La Rocca and Francesco Riggi in this manner to model numerous initial conditions to verify the various theories postulated by curious scientists over the years [9]. Comparing Figure 2 to Figure 1 shows that Tartaglia was not too far from the true shape of the trajectory, with the 'violent' and 'mixed' parts of his motion theory being not too dissimilar from the numerical solution of Rocca and Riggi.

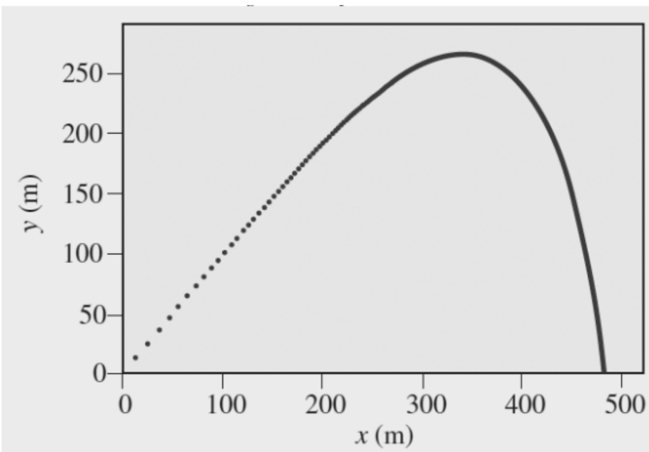

FIG. 2. Trajectory of a cannonball assuming air drag force proportional to the square of the velocity. Initial speed $u = 400ms^{-1}$, launch angle $\theta = 45°$. From [9].

II. THEORY

Newton's second law of motion postulates:

$$F = ma = m\frac{dv}{dt} = m\frac{d^2r}{dt^2}, \tag{2}$$

where m = mass of object, v = velocity of object, a = acceleration of object, r = displacement of object. Therefore, for a projectile accelerating under constant gravity, g:

$$F_{grav} = -mg\mathbf{k}, \tag{3}$$

where k is a unit vector normal to the surface of the Earth. Realistically, for a ballistic model, a force of drag needs to be included. This can be given by:

$$F_{drag} = -b\sqrt{v_x^2 + v_y^2 + v_z^2}\begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}. \tag{4}$$

Where b = a constant depending on the shape and size of the object and the composition of air.

If the object is moving through the air at velocity v, and experiences a force due to wind that has velocity w, using Galilean transformation it can be shown that in the frame of the Earth, the cannonball moves with velocity $v - w$. Therefore, the force on the projectile due to the wind is:

$$F_{wind} = -b|v - w|^2\frac{v - w}{|v - w|}. \tag{5}$$

With these amendments made to the motion of the projectile, equation 2 can be expressed as such:

$$ma = F_{grav} + F_{wind} = -mg\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ - b\sqrt{(v_x(t) - w_x)^2 + (v_y(t) - w_y)^2 + v_z^2}\begin{pmatrix} v_x(t) - w_x \\ v_y(t) - w_y \\ v_z(t) \end{pmatrix}. \tag{6}$$

Consider a cannonball dropped from a height z_0 with initial velocity $v_{z_0} = 0$, which falls with no wind present. This means $v_x = v_y = 0$. Therefore, we can use equation 6, and using only the z component:

$$m\frac{dv_z}{dt} = -mg + bv_z^2. \tag{7}$$

This equation then becomes:

$$\frac{dv_z(t)}{dt} = -g - b\sqrt{v_z(t)^2}v_z(t). \tag{8}$$

The odd form of taking the square root of the square of the velocity is vital as otherwise, the value would always be positive, meaning the simulated cannonball would 'fall' upwards instead of downwards indefinitely. Next let us consider the projectile in two dimensions with an initial launch velocity: the motion of the projectile with initial velocity in the x-z plane from ground level can be described using:

$$m\frac{dv_x(t)}{dt} = -\frac{mg}{u_T^2}\sqrt{v_x(t)^2 + v_z(t)^2}v_x(t), \tag{9}$$

and

$$m\frac{dv_z(t)}{dt} = -\frac{mg}{u_T^2}\sqrt{v_x(t)^2 + v_z(t)^2}v_z(t) - mg. \tag{10}$$

Furthermore, the relationship between the launch angle, θ and the range of the projectile will be studied. If the velocity of an object happens in the x-z plane, then the initial velocity can be described as:

$$\mathbf{v} = \begin{pmatrix} v_{x_0} \\ v_{z_0} \end{pmatrix} = v_m \begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix}, \tag{11}$$

where v_m is the muzzle or exit velocity of the projectile (the initial velocity).

III. METHOD

All of the following modelling has been completed using Python 3. There were two main methods used in the experiment: the iterative Euler method for numerical solutions of equations used for plotting graphs, and an analytical tool to evaluate intricate integrals in a short time.

A. Analytical Integration and Solving

The analytical integration makes use of a SymPy library which needs to be imported to Python. An example pseudocode shows how this simple function can output an integrate an expression in Figure 3.

```
symbols = 'symbols used in expression'
integral = 'some expression'
var = 'variable to integrate with respect to'
output = CALL Function integrate with integral, var
```

FIG. 3. Pseudocode for the SymPy integration function. The output can then be manipulated so that the result is output into LaTeX format.

This was very useful in the analytical solution of equation 7 later in the report.

Another use of analytical methods provided with the SymPy library is the ability to, for a given expression and its output, solve for a variable of the user's choice. How this is done is shown is pseudocode in Figure 4. This was used for comparison between analytical model found by the integration function, and Galilean model.

```
DECLARE expr = 'equation'
DECLARE eqOut = 'output of equation'
DECLARE solution = CALL solve with expr, eqOut
```

FIG. 4. Pseudocode for the SymPy solving function. The output can then be manipulated as any number.

B. Euler Method

Euler method was used as a good approximation to variables with values changing over time, to solve a first-order differential equation instead of calculating a value of a function at a time t, the time interval T was divided into N small fixed increments Δt, such that $T = N\Delta t$. Firstly, for an interval $t \in [0, T]$, a set of time variable values was discretised with $N+1$ points labelled t_i with $i = [0, 1, 2, ..., N]$. Next, noting that velocity is the rate of change of position with respect to time, and acceleration being the rate of change of velocity with respect to time, the following can be written for the Euler method [10]:

$$v_{i+1} = v_i + \frac{dv}{dt}\,\Delta t_i\ , \tag{12}$$

Where v = vertical velocity of an object and $\frac{dv}{dt}$ is the acceleration of the object. This approximation works by adding small increments to the current velocity, equal to the rate of change of the velocity multiplied by time. Similarly, for the position we have:

$$z_{i+1} = z_i + v_i\,\Delta t_i\ , \tag{13}$$

where z is the vertical displacement of an object. The computational power comes into play when we consider this method in a case where the number of increments is large, which would not be possible to calculate manually. Using the principle of looping in programming can make this task simple: the process starts with initializing sets of N elements for each variable that will be changing (time, position and velocity in this case) with 0 values. This is done to ensure there will be an equal number of elements for the plots when the calculation is complete. The program then iterates through values from the set i mentioned earlier and applies equations 8 & 9 to the sets of variables. A simplification of how this works in pseudocode can be seen in Figure 4.

IV. RESULTS

A. Theoretical analysis of the role of drag

Applying the method described in section 3.A to equation 7, yielded the following result:

$$v_z = -u_T \tanh\frac{gt}{u_T}, \tag{14}$$

where u_T is the terminal velocity of the object and is equivalent to $\sqrt{\frac{mg}{b}}$. Integrating using the same method once again yielded the expression for vertical displace-

```
DECLARE N = number of divisions
DECLARE T = time scale
DECLARE t = ARRAY[0:N]
DECLARE V = ARRAY[0:N]
DECLARE Z = ARRAY[0:N]
DECLARE dt = T/N
DECLARE a(t) = acceleration expression
for i = 0 to N
    t[i] = i * dt
    v[i] = v[i-1] + a(t[i]) * dt
    z[i] = z[i-1] + v[i] * dt

CALL plot with t, z
```

FIG. 5. Pseudocode for Euler method implemented with a simple model for acceleration, velocity and position with respect to time. The last line calls a plot function from the MatplotLib library that can be imported into Python; in this example the position z, is plotted against time t.

ment:

$$z = z_0 - \frac{u_T^2}{g} \ln(\cosh(\frac{gt}{u_T})), \tag{15}$$

where z_0 is the initial vertical displacement of the object. Figure 5 shows the plot of vertical velocity against time for different values of terminal velocity u_T, clearly showing that this value is indeed the maximum value the velocity can take, confirming the model from equation 12.

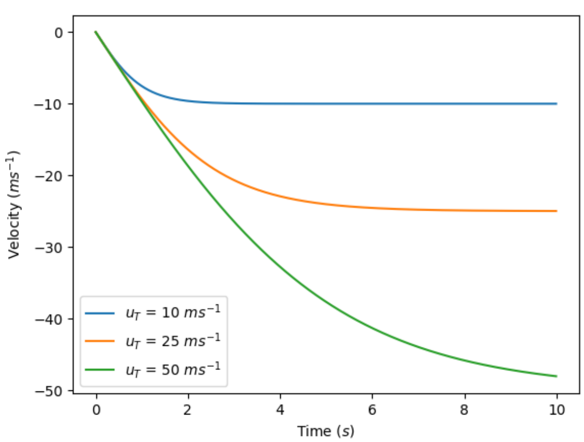

FIG. 6. Plots of the velocity of an object under gravity and air resistance, modelled for different values of terminal velocity. The plot shows how the velocity approaches the terminal velocity in each case, albeit at a slower rate for higher values of u_T.

The plot of horizontal displacement of the object with time can be seen in Figure 6; this uses the model established with equation 13. These calculations enabled

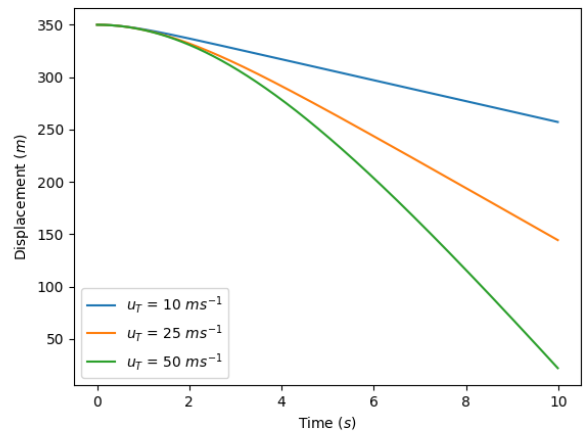

FIG. 7. Plots of the displacement of an object under gravity and air resistance, modelled for different values of terminal velocity with initial displacement $z_0 = 350m$.

the comparison between the analytically solved model and the Galilean models postulated in equation 1. The method of solving equations described in section 3.A was used to calculate the value of time where the object would hit the ground ($z = 0$), with the same set of initial conditions $z_0 = 500m$, mass $m = 20kg$ and $u_T = 100ms^{-1}$ in cases of both models of ballistic motion. The resultant difference between the two times was found to be $\Delta t = 0.842s$. This shows that there is a significant difference in the motion between the true model which includes air resistance, compared to the model postulated by Galileo.

B. Numerical Study of the Role of Drag on a Dropped Cannonball

To study the role of drag numerically, the Euler method described in Section 2.B was applied to equations 8 & 10, with the initial conditions $z_0 = 500m$ and $v_{z_0} = 0ms^{-1}$. Equation 8 needed to be discretised first, following the process described in section 3.A:

$$v_{z_{i+1}} = v_{z_i} + \Delta t(-g - (g/u_T^2)\sqrt{v_{z_i}^2}v_{z_i}) \tag{16}$$

The time scale, T was chosen to be the analytical solution computed in the previous section, since the initial displacement was the same to illustrate the full trajectory of the object. The plots of vertical displacement against time and vertical velocity against time can be seen in Figure 8. Since the Euler method is an approxi-

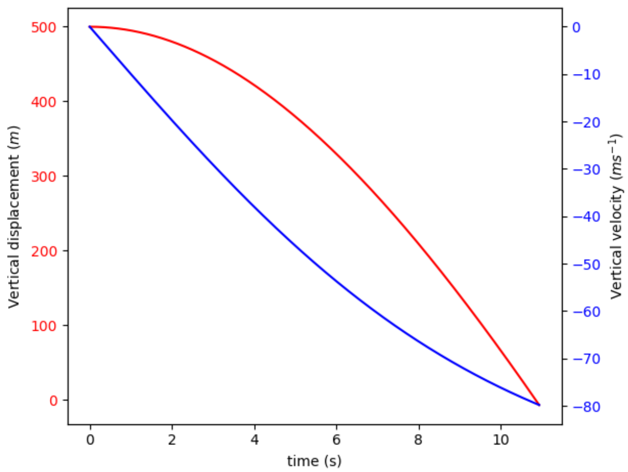

FIG. 8. Numerical modelling of the discretised equation of a cannonball with air resistance with the initial conditions $z_0 = 500m$, $u_T = 100ms^{-1}$ and $m = 20kg$, using the Euler method. The displacement is shown as the red line with velocity being the blue line.

mation of the true solution to the problem of motion of a cannonball, the global error, defined as the difference between the true solution to the equation: $z = 0$ and the last value calculated by the approximation: z_N, then the global error is given by:

$$\epsilon = z_N. \tag{17}$$

It can be shown that the global error is approximately proportional to the small increments used in the Euler method, Δt, when it is small. To verify this, the Euler method was run for different numbers of divisions of the time scale, and hence different values of Δt. The results show that the global error is indeed a linear function of Δt when Δt is small, which can be seen in Figure 9.

C. Numerical Study of the Role of Drag on a Launched Cannonball

To investigate how a projectile behaves if it is launched at some initial velocity in two dimensions with gravity and air resistance, the Euler method and discretisation of expressions was applied to equations 9 and 10 such that the velocity can be described using the following equations:

$$v_{z_{i+1}} = v_{z_i} - \Delta t \left(\frac{g}{u_T^2} \sqrt{v_{x_i}^2 + v_{z_i}^2}\, v_{z_i} + g \right) \tag{18}$$

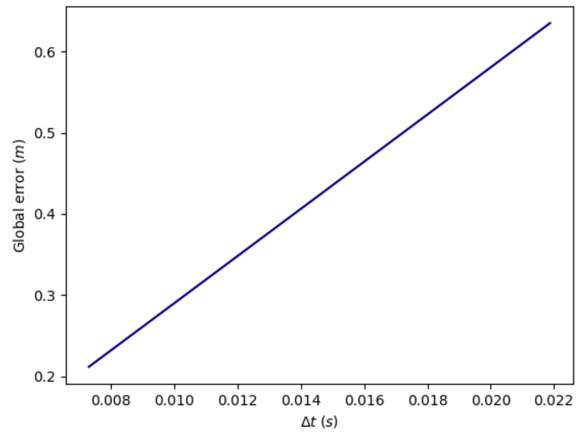

FIG. 9. The global error as the difference between the analytically calculated height and the Euler approximation-calculated height showing a clear linear relationship for small value of Δt, and hence large values of N, the number of divisions of the time scale.

and

$$v_{x_{i+1}} = v_{x_i} - \Delta t \frac{g}{u_T^2} \sqrt{v_{x_i}^2 + v_{z_i}^2}\, v_{x_i}. \tag{19}$$

These were then modelled with the initial conditions of $v_{z_0} = 80ms^{-1}$ and $v_{x_0} = 20ms^{-1}$, $u_T = 100ms^{-1}$. The horizontal and vertical displacement as a function of time can be seen in Figure 10, with the trajectory of the launched object in Figure 11.

To determine the time-of-flight of the projectile, as well

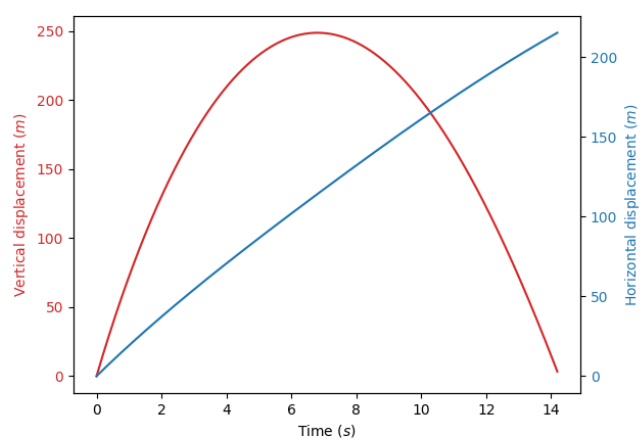

FIG. 10. Numerically modelled horizontal and vertical displacement of an object launched at initial horizontal and vertical velocities of $v_{z_0} = 80ms^{-1}$ and $v_{x_0} = 20ms^{-1}$ respectively.

as its range, the code used in producing Figures 9 & 10

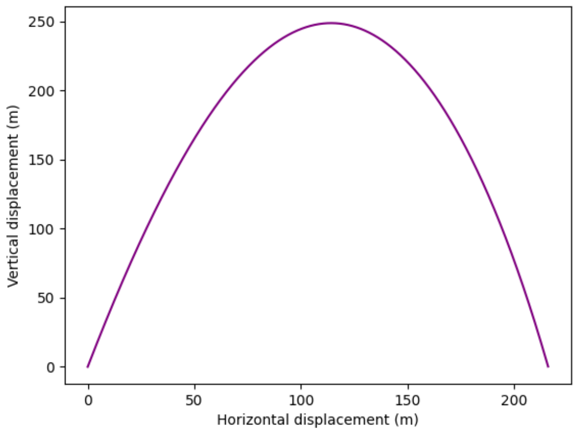

FIG. 11. Numerically modelled trajectory of a launched object with initial velocities $v_{z_0} = 80ms^{-1}$, $v_{x_0} = 20ms^{-1}$ and terminal velocity $u_T = 100ms^{-1}$.

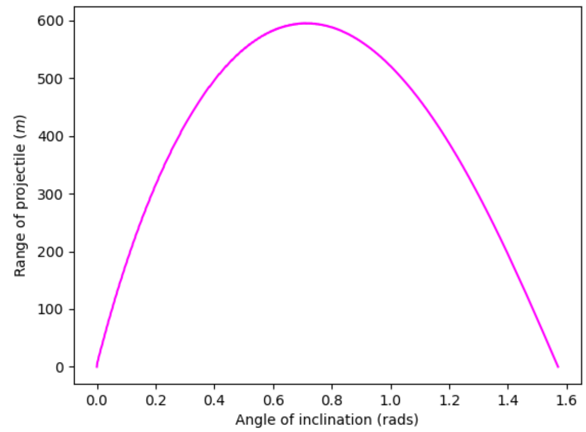

FIG. 12. Numerically modelled range of a projectile launched at muzzle velocity $v_m = 100ms^{-1}$ as a function of the launch angle θ. The maximum range occurs at $\theta \approx 0.7076$ radians.

was modified to return values of time and horizontal displacement for values of vertical displacement close to 0 (where the projectile hits the ground). This resulted in an estimated time-of-flight $t = 14.27s$ and the range of the projectile as $x = 216.4m$.

D. Predicting the Landing Point of a Cannonball

In this section, the muzzle velocity, v_m was held constant, and the launch angle, θ was varied to investigate its influence on the range of the projectile using equations 11, 18 and 19 and using the incremental Euler method to obtain a value of range for varying launch angles for the initial conditions $v_m = 100ms^{-1}$ and $u_T = 100ms^{-1}$. This yielded the plot in Figure 12; it can be seen that the optimal angle (maximum range) occurs around $\theta = \frac{\pi}{4}$ radians. However, upon further investigation, this angle was computed to be $\theta_{max} \approx 0.7076$ rad (0.077 rad difference). This clearly shows the importance of the launch angle on the range of the projectile. The next course of action was to determine what launch angle results in the projectile 'hitting' a target set at some horizontal displacement, x_{hit} away from the starting position. The target was positioned at $x_{hit} = 250m$ and the simulation was ran to determine the angles for which the range was close to the target range. This resulted in two different launch angles for which the range matched the target range: $\theta_1 = 0.149$ radians and $\theta_2 = 1.348$ radians. Finally, to combine all the ideas introduced in this report, a projectile travelling in two dimensions, with gravity and air resistance was modelled, taking into account the launch velocity and the wind velocity. In this model, the muzzle velocity was once again $v_m = 100ms^{-1}$ and a wind resistance was added with the horizontal velocity of the wind was taken as $w_x = -20ms^{-1}$. Using equation 6, and applying the discretisation and Euler method iterative calculation yielded the following equations to be used:

$$v_{x_{i+1}} = v_{x_i} - \Delta t \frac{g}{u_T^2} \sqrt{(v_{x_i}+20)^2 + (v_{z_i})^2}(v_{x_i}+20), \quad (20)$$

and

$$v_{z_{i+1}} = v_{z_i} - \Delta t \left(\frac{g}{u_T^2} \sqrt{(v_{x_i}+20)^2 + (v_{z_i})^2}(v_{z_i}) + g \right). \quad (21)$$

This time the target was 'put' at $x_{hit} = 300m$ and the simulation was run yet again to determine the angle of launch for which the range is the target horizontal displacement. This resulted in the two launch angles being computed as $\theta_1 = 0.204$ radians and $\theta_2 = 1.144$ radians.

V. DISCUSSION

Comparison of Figures 11 & 1 shows how the understanding of launched projectiles changed over the course of history. A direct comparison between Tartaglia's subdivisions of ballistic motion and the numerical modelling of this motion can be observed; drawing a parallel between the 'violent' part of the motion with the part of the trajectory where the initial velocity has the most influence of the path. Next, the 'mixed' motion where the gravitational deceleration brings the vertical velocity of the projectile from positive, through zero, and finally into negative values. Finally, the part that deviates the most, which is the 'natural' motion postulated by the early cannon men: the modelling has shown that this part of the motion, whilst steeper, definitely does not appear to be a straight, vertical line, which was thought to be the model seen in many sources after Tartaglia's theory crafting. Moreover, the trajectory (for the chosen initial conditions) is more akin to the one postulated by Galileo Galilei - a parabolic-appearing trajectory, which can be seen in Figure 13, indeed, it looks like the result of the numerical modelling is just a scaled-down version of the Galilean model. However, the Euler method used was

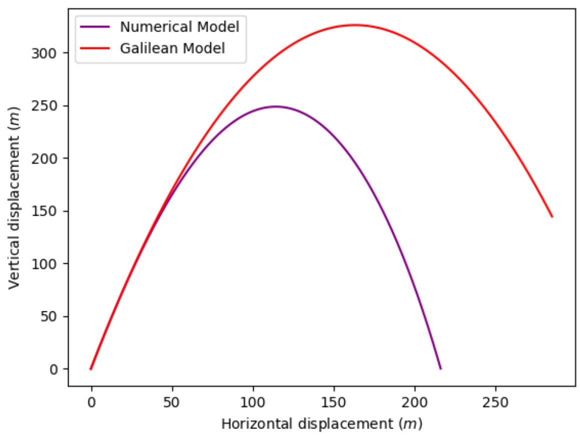

FIG. 13. Comparison of the trajectories of the numerical model from this experiment and the Galilean model of projectile motion.

just an approximation, which comes with an error, it was shown that global error is approximately proportional to the step size, when the step is small, and hence number of divisions is large. Additionally, it can be shown that the local error is proportional to the square of the step size [11], creating inevitable discrepancies between the true solutions to the investigated equations. It was however good enough to illustrate the shapes of trajectories in this experiment. Additionally, another source of error could have come from the value of wind speed being held constant and only in one direction. Naturally, in a real-world scenario, the wind speed would act in all three dimensions and it would change over time. Equations 20 and 21 could be modified to include the changing wind velocity such that wind velocity would be a function of time $\mathbf{w} = \mathbf{w}(t)$. This could be also included in the code, calculating the new wind speed for each increment of the Euler Method. Moreover, the terminal velocity was assumed to be constant in all the models, when in reality it depends on the shape of an object. If the object was a cannonball, and therefore a sphere, the expression for its terminal velocity is given by [12]:

$$u_T = \frac{2\rho g r^2}{9\eta}, \tag{22}$$

where r is the radius of the sphere, ρ is its density, and η is the viscosity of the medium (air in our case). This could be added to the code as a more accurate computation of the terminal velocity, additionally, the dependence of the trajectory on the radius of the cannonball would be better illustrated, providing a more accurate model. Another issue with the over-simplified model used was that it was assumed that the drag coefficient was constant, which impacted the force of air resistance in the model. In a real-world scenario, the drag coefficient has a complicated formula, depending on the Reynolds number, Re [13]:

$$Re = \frac{\rho u L}{\mu}, \tag{23}$$

where L is the characteristic length of an object, μ and ρ are the dynamic viscosity and density of air respectively and u is the flow speed of air. This can then be used to calculate the force of drag on a sphere using the following equation [14]:

$$F_{drag} = -\frac{b\rho v^2 A}{2}, \tag{24}$$

where A is the cross-sectional area of the sphere. This equation could be applied to the model in the code, giving a more accurate representation of the trajectory of a cannonball, further increasing the accuracy of the model. Furthermore, to increase the precision of the model, the value of time increment, Δt, could be reduced to much smaller values than those used in the model. Since the time complexity of some parts of the program were $O(n^2)$, meaning that if there were N divisions of the time scale, then the interpreter would have to execute N^2 lines of code, leading to even more operations performed by the processor (since Python is a High Level Language, a single line of code maps to multiple processor operations). This would greatly prolong the time of execution of the program, leading to immense inefficiency at some value of N. This could be remedied by optimisation of code, for example using a technique called 'loop tiling' [15]. Finally, the biggest possible improvement to the method of this investigation would be to use another, more accurate iterative algorithm. One good example is the Runge-Kutta Method [16], which instead of taking the rate of change at one point, like the Euler method, uses four different increments, at four different points of the rate of change of a variable and calculates their weighted average. This provides a more sophisticated approximation, which could greatly benefit the accuracy of the model.

VI. CONCLUSION

The aim of this investigation was to inspect several aspects of ballistic motion through numerical modelling and the use of the iterative Euler method. Firstly, an analytical solution to a simple, one-dimensional problem about the role of drag was found through an integration function, and the role of the terminal velocity on both displacement and velocity of an object was shown in Figures 6 & 7. Secondly, the role of drag on a dropped object was modelled numerically, resulting in the plot in Figure 8, showing the velocity and vertical displacement of said object. Furthermore, it was shown that the global error between the analytical and the numerical solution is approximately linearly dependent on the step size, Δt, and hence inversely proportional to the number of divisions of the time scale, N. Furthermore, the role of drag on a cannonball launched in 2D was investigated and the results presented in Figure 10 shows that in opposition to the Galilean model, the horizontal displacement of an object is not a linear function of time. Finally, the landing point of a cannonball was predicted for a set of initial conditions with, and without wind accounted for, which was represented in Figures 11 and 12.

REFERENCES

[1] Grant E. *The Foundations of Modern Science in the Middle Ages: Their Religious, Institutional, and Intellectual Contexts.* Cambridge University Press, Cambridge, 1996, pp. 58–63.

[2] Tartaglia N. *The newly discovered invention of Nicolò Tartalea of Brescia, most useful for every theoretical mathematician, bombardier, and others, entitled 'New Science'. In: Drake S Drabkin IE (eds) Mechanics in Sixteenth Century Italy: Selections from Tartaglia, Guido Ubaldo and Galileo.* University of Wisconsin Press, Madison, 1969, pp. 63–97.

[3] Tartaglia N. *Nova Scientia: Three bookes of Colloquies Concerning the Arte of Shooting in Great and Small Peeces of Artillerie.* 1588.

[4] Hussey E. *Aristotle's mathematical physics: A reconstruction.* Clarendon Press, Oxford, 1991, p. 235.

[5] Bonaparte L. N. "History of the cannon". In: *Colburn's United Service Magazine* 2 (1849), pp. 180–195.

[6] Hannam J. *God's Philosophers: How the Medieval World Laid the Foundations of Modern Science.* Icon Books, London, 2009, pp. 136–139.

[7] Naylor R.H. "Galileo: The search for the parabolic trajectory". In: *Annals Science* 33 (1976), pp. 153–172.

[8] Tina Tan. URL: `https://www.minitool.com/news/what-is-a-good-processor-speed.html`.

[9] La Rocca P. Riggi F. "Projectile motion with a drag force: Were the Medievals right after all?" In: *Physics Education* 44 (2009), pp. 398–402.

[10] John C. Butcher. *Numerical Methods for Ordinary Differential Equations.* John Wiley Sons, New York, 2003, p. 45. ISBN: 978-0-471-96758-3.

[11] Kendall A Atkinson. *An Introduction to Numerical Analysis.* 2nd ed. John Wiley Sons, New York, 1989. ISBN: 978-0-471-50023-0.

[12] George B. Arfken et al. "chapter 6 - NEWTON'S LAWS". In: *International Edition University Physics.* Ed. by George B. Arfken et al. Academic Press, 1984, pp. 100–127. ISBN: 978-0-12-059858-8. DOI: `https://doi.org/10.1016/B978-0-12-059858-8.50011-X`. URL: `https://www.sciencedirect.com/science/article/pii/B978012059858850011X`.

[13] A. Sommerfeld. "A Contribution to Hydrodynamic Explanation of Turbulent Fluid Motions". In: *International Congress of Mathematicians* 3 (1908), pp. 116–124.

[14] Barnes W. McCormick. *Aerodynamics, Aeronautics, and Flight Mechanics.* John Wiley Sons, Inc., New York, 1979, p. 24. ISBN: 0-471-03032-5.

[15] Steven Muchnick. *Advanced Compiler Design Implementation.* Morgan Kaufmann. ISBN: 978-1-55860-320-2.

[16] C. Runge. "Ueber die numerische Auflsung von Differentialgleichungen". In: *Mathematische Annalen* 46 (June 1895), pp. 167–178. DOI: `10.1007/bf01446807`. URL: `https://doi.org/10.1007/bf01446807`.