Jia Xiu Sai 08/04/2019

PROJECT PLAN

Outline
Simulation programs are used in engineering and science to investigate how complex systems will react to changing variables without doing the experiment in real-life. This saves time, cost and most importantly, it is extremely safe as the engineers and scientist do not need to carry out the experiments. In this project, the simulation will focus on a sphere object that is fired out of an air cannon at different speeds. It will show the changing speed, acceleration and distance with both a spreadsheet and animated graphics from the variable parameters input by the user. This will study what effect the angle, initial speed and weight of the projectile will have on the variables using a set of equations.

Research
The speed of the projectile object can be written in a vector (x and y) with different values acting in different directions. This can be calculated from the initial velocity using trigonometry. The velocity acting in the horizontal component is affected only by the air resistance and determines how fast/far the object will travel horizontally. On the other hand, the gravitational force and air resistance acting on the object affects the vertical component of the velocity and it determines how long the projectile will stay in the air. Figure 1 below shows a rough draft of how the simulation will look like and the forces acting on the ball include drag, gravitational force and velocity.

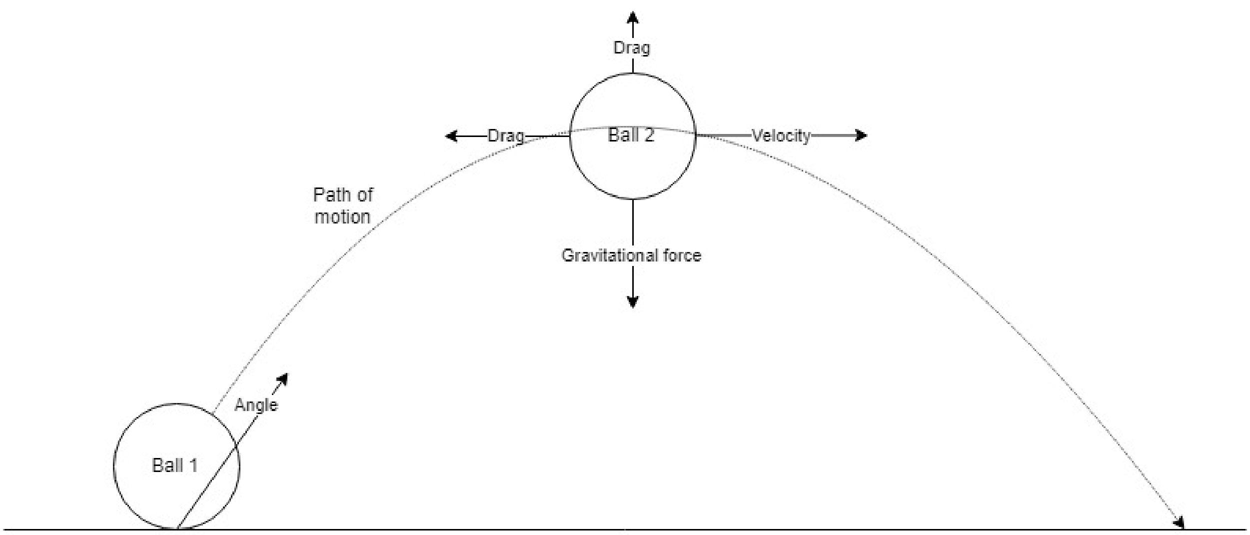


Figure 1: Rough sketch of simulation.

System Parameters/ Variables
In this program, the user will input these parameters:
- Mass of the projectile fired in kilograms.
- The initial speed of the projectile in meters per seconds.
- The initial angle of the projectile in degrees
    - This is converted into radians by the program but all data of angle will be displayed in degrees.
    - This value will only be limited to between 0 and 90 degrees.
- Drag coefficient

The fixed and calculated variables are:

- X and Y coordinates of:
    - Projectile position
- Total simulation run time in seconds
    - Depends on how long the projectile stays in the air ($y > 0$).
- Step in seconds

Arrays/lists are used for:

- X, Y and final velocity coordinates (v_x, v_y & v)
- X, Y and final of acceleration coordinates (a_x, a_y & a)
- X, Y and final of distance coordinates (s_x, s_y & s)
- Time (t)
- Step (i)

The menu in the main function will include options:

1. Input parameters
2. Show current parameters
3. Calculate data
4. Save data as a CSV file
5. Plot graph
    a. Velocity (m/s) against time (sec)
    b. Height (m) against distance travelled (m)
6. Run simulation
7. Quit program

Figure 1 below shows an estimation of how the main program will work.

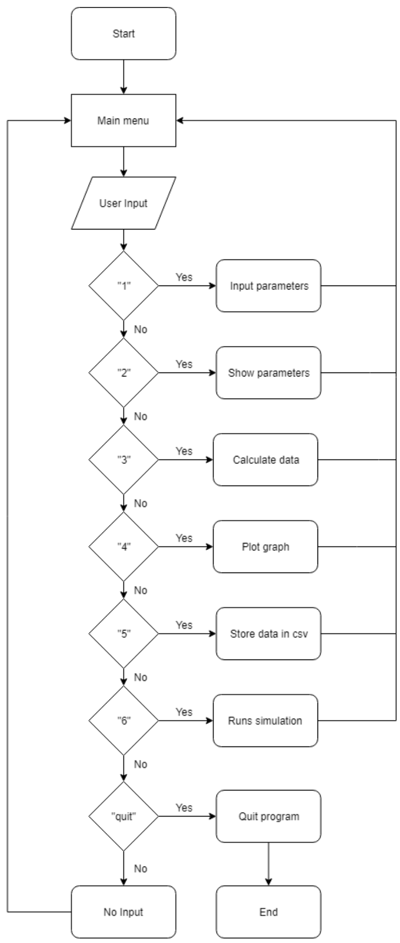
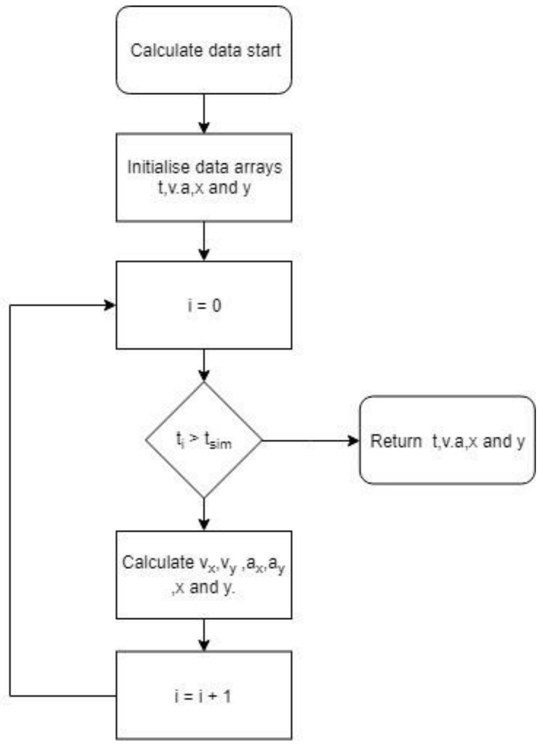


Figure 2 below shows an estimation of how the calculate function will work.



Equation/ Formulas

These equations will be used in the calculate data function:

1. Gravitational Force
    1. Mass × Gravitational constant (9.81) × Height

2. Air resistance
    1. $\frac{\text{Drag coefficeint} \times \text{Air density} \times \text{Frontal area} \times (\text{Current velocity})2}{2}$
    2. Frontal area $= \pi r^2$
    3. Air density at sea level is 1.225 kg/m^3 According to the International Standard Atmosphere (ISA)

3. Initial velocity (v)
    1. Velocity-x (v_x) = Speed × Cos (angle)
    2. Velocity-y (v_y) = Speed × Sin (angle)

4. Current acceleration (a)
    1. Acceleration-x (a_x) = -Drag (x-axis)
    2. Acceleration-y (a_y) = Gravitational force - Drag (y-axis)

5. Next velocity (v)
    1. Next velocity (v_x) = Current velocity-x - Acceleration-x (a_x)
    2. Next velocity (v_y) = Current velocity-y - Acceleration-y (a_y)

6. Next distance (s)
    1. Next distance-y (s_x) = Current distance-x + (Next velocity (v_x) x Step)
    2. Next distance-x (s_y) = Current distance-y + (Next velocity (v_y) x Step)

The basic algorithm for computing the simulation data is as follows:

1. Set initial time to zero.
2. The initial x and y velocity is calculated from the initial velocity and angle from user input using equation 3.
3. Store initial values in lists.
4. Loop until reaches total simulation time:
    a. Calculate gravitational force using equation 1.
    b. Calculate the frontal area and air resistance using equation 2.
    c. Calculate current acceleration using equation 4.
    d. Calculate next velocity using equation 5.
    e. Calculate next distance using equation 6.
    f. Add results to lists.

Problem Composition

The requirement for this task are that it:

- Must be written using Python programming language.
- Must be portable in Microsoft Windows.
- Must include the ability for user input.
- Must include a range of data types.
- Must include at least three user defined functions.
- Must use the import function.
- Must display data in tabular format.
- Must generate at least one graph.
- Must use at least three relevant physics equations.

Notes/ Ideas

Coding stages of this program:

1. Structure of the menu with function.
2. Fill in equations and calculations.
3. Convert the data into visuals
    a) Display key variables.
    b) Coordinates of projectile where it reaches the highest X and Y coordinates.
    c) A ruler that is variable to the values from the calculation.
4. Graphs
    a) Velocity against time.
    b) Displacement x-coordinate against Displacement y-coordinate
5. User Input
6. Display user input
7. Save as a CSV file
8. Crash prevention and testing

Future development can include features into this program like the ability to insert parameter of the wind, ball bouncing after reaching the ground and an overall simulation which visualise all angle with intervals of 10 degrees instead of angle input. Furthermore, a ruler in the visual presentation can include that changes appropriately depending on the projectile's x and y coordinates.

References

1. Hall,N. (2015) 'Drag of a Sphere' National Aeronautics and Space Administration. [Online] Available: https://www.grc.nasa.gov/www/k-12/airplane/dragsphere.html

2. Wikipedia (2019) ‘International Standard Atmosphere’ [Online] Available: https://en.wikipedia.org/wiki/International_Standard_Atmosphere
3. Khan Academy. (n.d.) ‘What is 2D projectile motion?’ [Online] Available: https://www.khanacademy.org/science/physics/two-dimensional-motion/two-dimensional-projectile-mot/a/what-is-2d-projectile-motion
4. Khan Academy. (n.d.) ‘What are velocity components?’ [Online] Available: https://www.khanacademy.org/science/physics/two-dimensional-motion/two-dimensional-projectile-mot/a/what-are-velocity-components