ENGG1001: Programming for Engineers

> Assignment 1 (10 Marks)

Due: Friday 26th March, 4pm

Introduction

Legend has it that in 1971, two college students at Yale University used a large slingshot (made with eight 2-metre lengths of surgical rubber) to launch a loaf of frozen bread over their college master's house. The approximate trajectory of the loaf is shown below.

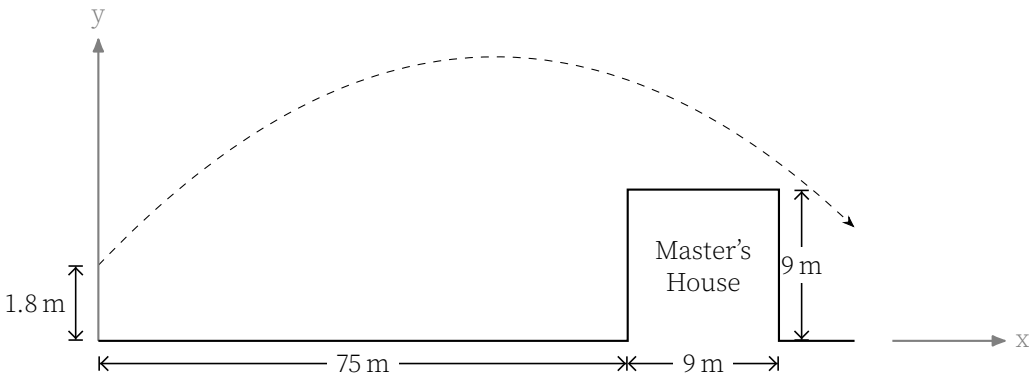


Figure 1: *Approximate trajectory (not to scale)*

The equations of motion that the bread satisfied are:

$$
\begin{aligned}
x(t_0 + \Delta t) &= x_0 + v_{x0}\Delta t + \tfrac{1}{2}a_x\left(\Delta t\right)^2 \\
y(t_0 + \Delta t) &= y_0 + v_{y0}\Delta t + \tfrac{1}{2}a_y\left(\Delta t\right)^2 \\
v_x(t_0 + \Delta t) &= v_{x0} + a_x\Delta t \\
v_y(t_0 + \Delta t) &= v_{y0} + a_y\Delta t
\end{aligned}
$$

where x is the distance travelled in the horizontal direction, y is the distance travelled in the vertical direction, v_x is the velocity in the horizontal direction, v_y is the velocity in the vertical direction, a_x is the acceleration in the horizontal direction, a_y is the acceleration in the vertical direction, the subscript $_{-0}$, denotes a value at time t_0 and Δt is the time interval at which successive new values of x, y, v_x and v_y are computed. $a_x = 0\,\mathrm{m\,s^{-2}}$ and $a_y = -9.81\,\mathrm{m\,s^{-2}}$ (i.e. a_y is the acceleration due to gravity).

Task 1

You need to write a program that computes the trajectory of the bread. When this program is invoked a user must be able to select one of the following three options:

- i for specifying initial conditions,
- c for computing and plotting the trajectory, or
- q for quitting the program.

A sample printout of the functioning program is given below.

```
>>> main()

Please specify one of the following options:
     'q' - quit
     'i' - input initial conditions
     'c' - compute and plot trajectory
i
Input launch height (in meters): 1.8
Input launch velocity (in meters/second): 30
Input the launch angle (in radians): .6
Input height of master's house (in meters): 9
Input distance to house's furthest wall (in meters): 84
Input time increment for modelling (in seconds): .1

Please specify one of the following options:
     'q' - quit
     'i' - input initial conditions
     'c' - compute and plot trajectory
c

Please specify one of the following options:
     'q' - quit
     'i' - input initial conditions
     'c' - compute trajectory
q
```

You can initialise a vector in Python with the following type of command:

```
>>> x_vec = ( )
>>> x_vec
( )
```

You can add a new value to the vector with the following type of command:

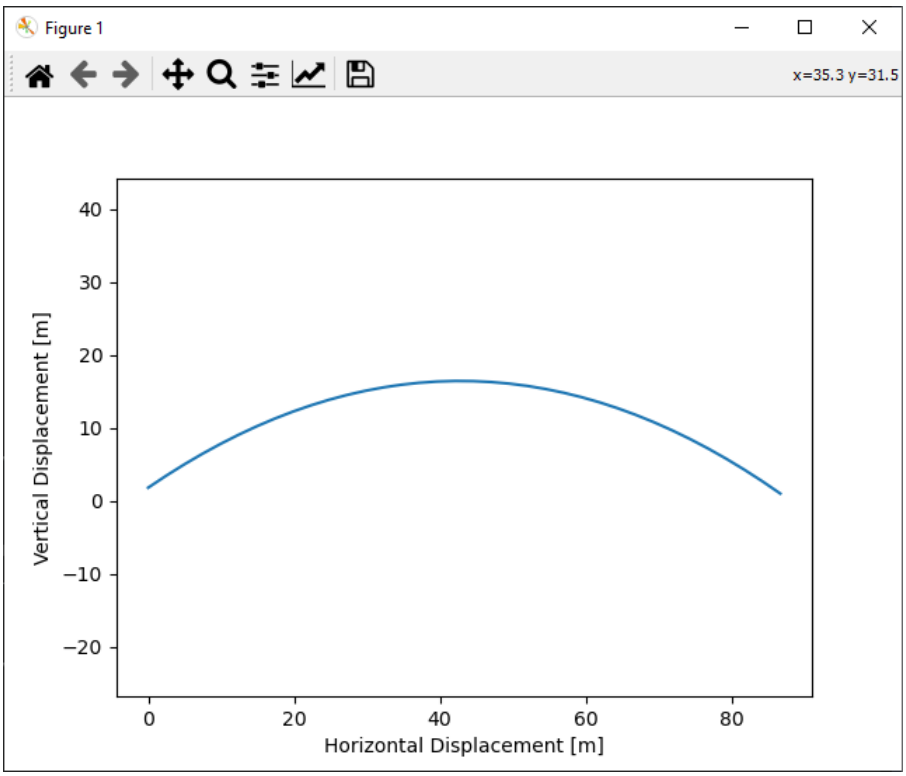
```
>>> x_vec += (5,)
>>> x_vec
(5,)
```

You must round all values returned by the kinetic model to two decimal places.

It is not a requirement of the assignment to plot out your trajectory, but you are advised to do so in order to check the plausibility of your results. You can use the following commands to plot the trajectory:

```
>>> import matplotlib.pyplot as plt
>>> plt.plot(x_vec, y_vec)
>>> plt.xlabel('Horizontal displacement (m)')
>>> plt.ylabel('Vertical displacement (m)')
>>> plt.show( )
```

If you plot the trajectory corresponding to the initial conditions above you would obtain a graph with the shape shown below (though the scaling of the axes may be different). Before submitting your assignment make sure you remove any plot commands as they can disrupt the automatic testing procedure. You must implement your solution using the provided form for the equations. That is, you must compute the trajectory one sample at a time.



Note that you can assume the user will always enter the `i` option before entering the `c` option.

Task 2

Modify your program so that it takes in a fourth command option: `o`, which causes the program to determine and print out the optimum launch angle. A sample printout when the `o` option is entered is given below:

```
>>> main()

Please specify one of the following options:
     'q' - quit
     'i' - input initial conditions
     'c' - compute trajectory
     'o' - print out optimum launch angle
o
0.85
```

Again, you can assume the user will have entered the `i` option before entering the `o` option. Note that in the sample output above, 0.85 is not the correct value for the optimal angle; you need to write code which will produce the correct optimal angle. You cannot simply print out an answer - there must be code written to determine the optimal angle. You must also round the value for optimal angle to two decimal places.

Writing Your code

You must download the file, `a1.py`, from Blackboard, and write your code in that file. When you submit you assignment **you must submit only one file and it must be called `a1.py`**. Do not submit any other files or it can disrupt the automatic code testing program which will grade your assignments.

Design

Your are expected to use good programming practice in your code and you must incorporate at least the following functions.

`main`

This function handles the interaction of your program with the user and manages the calls to the other functions you write. It also handles the plotting. It is invoked with the following command

```
>>> main()
```

Parameters

- `None` Takes no parameters

Returns

- `None` Does not return anything

`get_initial_conditions`

This function prompts the user to input the launch height, the launch velocity, the launch angle, the master's house height, the distance to the furthest wall of the master's house and the time increment for modelling the equations of motion. It is assumed that the initial horizontal position is 0.

Parameters

- `None` Takes no parameters

Returns

- `x0` The initial horizontal position
- `y0` The initial vertical position
- `v0` The launch velocity
- `theta0` The launch angle
- `ax` The acceleration in the x direction
- `ay` The acceleration in the y direction
- `house_height` The height of the master's house
- `house_distance` The distance to the furthest wall of the house
- `delta_t` The time increment used to implement the modelling

`kinetic_model`

This function takes in the relevant initial conditions as arguments and uses the equations of motion to calculate and return vectors of time-series results.

Parameters

- `x0` The initial horizontal position
- `y0` The initial vertical position
- `v0` The launch velocity
- `theta0` The launch angle
- `ax` The acceleration in the x direction
- `ay` The acceleration in the y direction
- `house_height` The height of the master's house
- `house_distance` The distance to the furthest wall of the house
- `delta_t` The time increment used to implement the modelling

Returns

- `x_vec` Vector of x-coordinates
- `y_vec` Vector of y-coordinates
- `vx_vec` Vector of horizontal velocities
- `vy_vec` Vector of vertical velocities Note that the vector of y co-ordinates must not include any negative values. i.e. do not include points with co-ordinates that are "below the ground".

Assessment and Marking Criteria

Functionality Assessment — 7 Marks

The functionality will be marked out of 7. Your assignment will be put through a series of tests and your functionality mark will be proportional to the number of tests you pass. If, say, there are 25 functionality tests and you pass 20 of them, then your functionality mark will be $^{20}/_{25} \times 7$.

You will be given the functionality tests before the due date for the assignment so that you can gain a good idea of the correctness of your assignment yourself before submitting. You should, however, make sure that your program meets all the specifications given in the assignment. That will ensure that your code passes all the tests. Note: Functionality tests are automated and so string outputs need to exactly match what is expected.

Code Style Assessment — 3 Marks

The style of your assignment will be assessed by one of the tutors, and you will be marked according to the style rubric provided with the assignment. The style mark will be out of 3.

Assignment Submission

You must submit your completed assignment electronically through Blackboard. The only file you submit should be a single Python file called `a1.py` (use this name – all lower case). This should be uploaded to Blackboard > Assessment > Assignment 1. You may submit your assignment multiple times before the deadline – only the last submission will be marked.

Late submission of the assignment will not be accepted. In the event of exceptional personal or medical circumstances that prevent you from handing in the assignment on time, you may submit a request for an extension. See the course profile for details of how to apply for an extension.

Requests for extensions **must** be made no later than 48 hours prior to the submission deadline. The application and supporting documentation (e.g. medical certificate) **must** be submitted to the ITEE Coursework Studies office (78-425) or by email to enquiries@itee.uq.edu.au. If submitted electronically, you **must** retain the original documentation for a minimum period of six months to provide as verification should you be requested to do so.