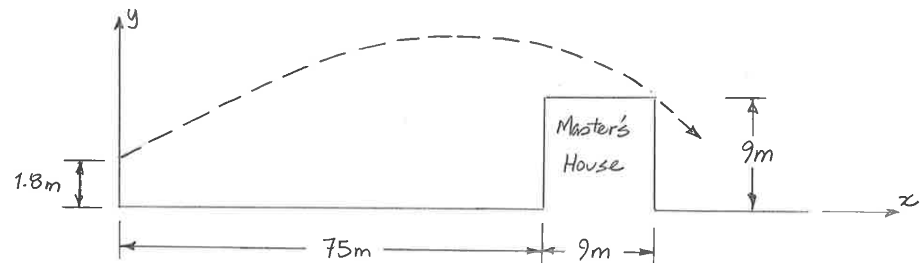
The University of Queensland
School of Mechanical and Mining Engineering

MECH2700 Engineering Analysis I (2015)

Assignment 1: Computational Mechanics

Introduction

Legend has it that, in 1971, two college students at Yale University used a large slingshot (made with 8 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.



Write a Python program to compute the trajectory assuming that

- there is no wind.
- the drag coefficient of the loaf of bread is the same as a similarly-sized sports ball.
- the bread leaves the slingshot at a height of 1.8 metres above the ground.

Then use your program to determine the optimum firing angle, with respect to the horizontal, and the minimum launch velocity to just clear the master's house.

Theory

The continuous motion of an object in the x, y-plane can be described by specifying the object's position (x, y) and velocity (v_x, v_y) as functions of time, t. If we know the forces (F_x, F_y) acting on the object then we can compute the state of motion for all times by integrating the equations of motion, shown here in Cartesian coordinates.

$$m\,a_x = \Sigma F_x \quad ; \quad m\,a_y = \Sigma F_y \tag{1}$$

The object is assumed to have a fixed mass, m, concentrated at its centroid and the components of acceleration are defined as

$$a_x = \frac{d^2x}{dt^2} \quad ; \quad a_y = \frac{d^2y}{dt^2} \tag{2}$$

In your Applied Mechanics course in first year, you usually had a simple functional form for the force components and integrated the equations of motion analytically to obtain a closed-form solution. For most situations that you encounter in your engineering practice this analytic approach will not be feasible [1] and you will have to resort to some form of numerical scheme to integrate the equations of motion approximately. This is the basis of nearly all simulation packages. In this assignment, you will develop a Python program to compute a discrete approximation the ballistic trajectory of a projectile.

Task 1 Read Section 14.6 *Numerical Solutions* of an old version of Bedford & Fowler's Engineering Mechanics text. You may have this book from your Applied Mechanics course or find it in the library. In case the numbering is different in your copy of the text, Chapter 14 is labelled "Force, Mass and Acceleration".

Task 2 Show that the state of motion of the object at $t = t_0 + \Delta t$ (*i.e.* a short time after some arbitrary time t_0) can be approximated by

$$\begin{aligned} x(t_0 + \Delta t) &= x_0 + v_{x0}\Delta t + \frac{1}{2}a_{x0}\Delta t^2 && (3) \\ y(t_0 + \Delta t) &= y_0 + v_{y0}\Delta t + \frac{1}{2}a_{y0}\Delta t^2 && (4) \\ v_x(t_0 + \Delta t) &= v_{x0} + a_{x0}\Delta t && (5) \\ v_y(t_0 + \Delta t) &= v_{y0} + a_{y0}\Delta t && (6) \end{aligned}$$

where the subscript 0 indicates a value at $t = t_0$.

The simulation

For our trajectory simulation, let's assume that the frozen loaf of bread has a mass $m = 0.7\,\text{kg}$ and has an effective diameter $d = 15\,\text{cm}$ (to be used for determining the frontal-area of the loaf in flight).

At $t = 0$, the loaf leaves the slingshot located at (x_0, y_0) with a speed of $v_0\,\text{m/s}$ and a flight-path angle, θ_0, with respect to the horizontal. These are the initial conditions.

While in flight, the only forces acting on the load are self-weight because of the Earth's gravity ($g = 9.81\,\text{m/s}^2$) and aerodynamic drag. The magnitude of the aerodynamic drag is given by the expression

$$F_{drag} = \frac{1}{2}\rho\, v^2\, A_{front}\, C_{drag} \tag{7}$$

where $\rho = 1.2\text{kg/m}^3$ is the ambient air density, v is the speed of the projectile, A_{front} is the frontal area of the loaf (*i.e.* its cross-sectional area) and C_{drag} is a dimensionless coefficient that has been determined by experiment. If we specify our quantities in metres, kilograms and seconds, F_{drag} will be in Newtons. The direction of the aerodynamic drag force is opposite to that of the projectile's velocity.

Task 3 Draw a free-body diagram showing the forces acting on the loaf of bread while it is in flight.

Task 4 Write expressions for the acceleration components a_x and a_y. Each of these may be built up as a set of expressions.

[1] Possibly because you do not have the forcing functions in analytic form or because they are too complicated to integrate.

Task 5 The drag coefficients of sports balls have been measured by a number of people and published in various journal papers. Locate some of these publications and determine a suitable value of C_{drag} for the loaf of bread.

Task 6 Write a Python function that computes the trajectory of the loaf once it has left the slingshot and continues until it strikes the ground or hits the Master's house. Your program should start with the initial state as discussed above and with $x_0 = 0.0$, $y_0 = 1.8$ and guessed values of v_0 and θ. It should then integrate the equations of motion with small increments of time and save the loaf's position and velocity at discrete points in time. Terminate the integration process when the loaf has hit the ground, ignoring the presence of the house.

Task 7 For $C_{drag} = 0.0$, integrate the equations of motion analytically to determine an exact solution for the ideal case of no air resistance. Demonstrate that your program agrees with your analytic solution.

Task 8 Determine a time-step that gives a sufficiently accurate estimate of the trajectory for the case of nonzero drag. Use the time of flight and downrange distance travelled as the test criteria. Explain why you consider this to be good enough.

Task 9 Use your program to determine optimum initial velocity and flight angle for the loaf to just clear the Master's house as shown in the figure. This could be considered a computational experiment and, although the search for the optimum initial conditions could be automated, doing a manual search is fine for this exercise.

Task 10 Compute and plot the flight path for your optimum initial conditions.

Submission Notes and Assessment Criteria

This is an individual assignment, however, discussing your formulation and code development with your peers is a useful activity, especially when working out details of how to formulate code from the mathematical model. Submit a report, detailing your analysis, code and results for Tasks 2 to 10. This report should be brief but well structured, so that the tutor can easily interpret your results. Include your Python code as and produce a single PDF file for your report. It is fine to scan hand-written elements (such as equations and schematic figures) and include them in your report. There is no good reason to add errors to your report through sloppy typesetting. The plot in task 10, however, should be computer-generated and appropriately labelled. As well as the report PDF, you will need to submit a zip-file containing all of your Python code. For assessment of your work, the tutors will be looking for answers to the following questions.

- Have you done what the tasks require? Note that your report should be well structured and tidy so that the tutor can easily find your answers. Your plots having correct titles and axes labels is quite helpful.
- Are your answers accurate?
- Is your code well-structured and clearly written, with suitable documentation comments?
- Does your code correspond to the results shown in your report?