Final Project

Date: Aban 8th, Due: Aban 15th

This homework is designed to give you practice with writing functions to solve problems. You will just be graded on whether your scripts produce the correct output, but not necessarily on how efficiently they're written.

Q1. 1. Throwing a ball.

1. Below are all the steps you need to follow, but you should also add your own meaningful comments to the code as you write it.
    a. Start a new file in the MATLAB Editor and save it as throwBall.m
    b. At the top of the file, define some constants (you can pick your own variable names)
        i. Initial height of ball at release = 1.5 m
        ii. ii. Gravitational acceleration = 9.8 m/s2
        iii. Velocity of ball at release = 4 m/s
        iv. Angle of the velocity vector at time of release = 45 degrees
    c. Next, make a time vector that has 1000 linearly spaced values between 0 and 1, inclusive.
    d. If x is distance and y is height, the equations below describe their dependence on time and all the other parameters (initial height h, gravitational acceleration g, initial ball velocity v, angle of velocity vector in degrees q). Solve for x and y:
        i. $x(t) = v\cos(q\frac{\pi}{180})t$
        ii. $y(t) = h + v\sin(q\frac{\pi}{180})t - \frac{1}{2}gt^2$
    e. Approximate when the ball hits the ground.
        i. Find the index when the height first becomes negative (use find).
        ii. The distance at which the ball hits the ground is value of x at that index
        iii. Display the words: The ball hits the ground at a distance of X meters. (X is the distance you found in part ii above)
    f. Plot the ball's trajectory
        i. Open a new figure (use figure)
        ii. Plot the ball's height on the y axis and the distance on the x axis (plot)
        iii. Label the axes meaningfully and give the figure a title (use xlabel, ylabel, and title)
        iv. Hold on to the figure (use hold on)
        v. Plot the ground as a dashed black line. This should be a horizontal line going from 0 to the maximum value of x (use max). The height of this line should be 0. (see help plot for line colors and styles)
    g. Run the script from the command window and verify that the ball indeed hits the ground around the distance you estimated in e-ii. You should get something like this:

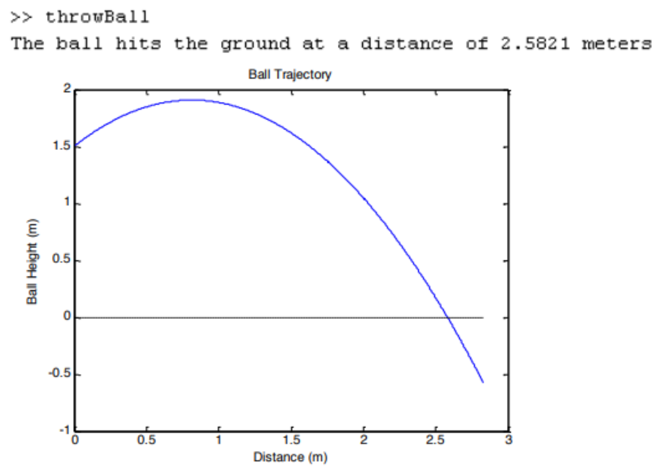

Figure 1- Figure for question 1

Q2. 2. Function

Turn your script for question 1 into a function. Add a function declaration that takes v and theta as inputs and returns the distance at which the ball hits the ground: distance=throwBall(v,theta).

We generally don't want functions to plot things every time they run, so remove the figure command and any plotting commands from the function. To be able to simulate a wide range of v and theta, make the time go until 10sec and add an if statement that will display the warning 'The ball does not hit the ground in 10 seconds' if that turns out to be the case. Also, if the ball doesn't hit the ground in 10 seconds, you should return NaN as the distance. To test your function, write a script testBall.m to throw the ball with the same velocity v but different angles and plot the distance as a function of angle theta. The plot should look something like the figure below. You will need to run throwBall within a loop in order to calculate the distances for various thetas.

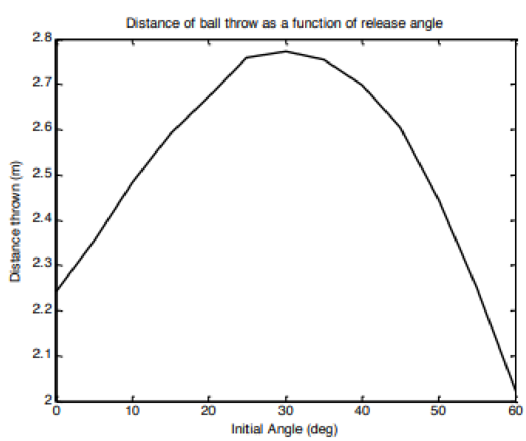

Figure 2- Figure for question 2

Reference: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-057-introduction-to-matlab-january-iap-2019/index.htm