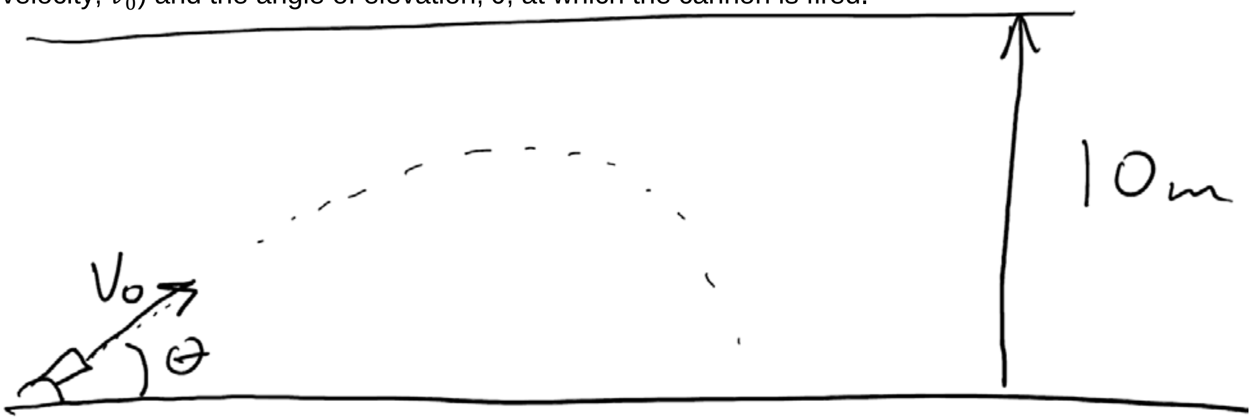
BIOE 198MI Biomedical Data Analysis. Spring Semester 2018.
Lab 2: Modelling Projectile Motion

Problem Statement

You are tasked with firing a cannonball inside of a range with 10 meter high ceiling with the objective of maximizing the distance that the cannonball travels. You can control the firing speed (initial velocity, v_0) and the angle of elevation, θ, at which the cannon is fired.



Assuming that the only external forces working on the projectile is gravity, how would you model the motion of the cannonball in MATLAB and find the optimal conditions?

Other Parameters:

$\vec{v}_{0,max} = 25$ m/s

Useful equations:

$$\vec{d} = \vec{v}_0 t + \frac{1}{2}\vec{a}\,t^2$$

Code Walkthrough

```
%% Modeling & Plotting Projectile Motion
```

Set initial parameters

$\vec{v}_0 = 25$ m/s, θ = 60^0, $(x_0,y_0) = (0,0)$ m,

Other variables:

$\vec{d}(t)$, t= ?, $\vec{a}$= ?

Approach: Break 2D problem into 2 1D problems

	x- component	y-component
$\vec{d}(t)$	$v_{0,x} \cdot t + \frac{1}{2}\overrightarrow{a_x} \cdot\ t^2$	$v_{0,y} \cdot t + \frac{1}{2}\overrightarrow{a_y} \cdot\ t^2$
$\vec{v}_0$	v_o cos θ	v_o sin θ
$\vec{a}$	0 m/s^2	g = -9.8 m/s^2

Now, let's look at the code. Put a breakpoint at line 38 and hit 'run'!
(lines 1-38 replicated below)

```
clear all; close all
%% Modeling & Plotting Projectile Motion
%Setting initial variables
v_0 = 25; % meters/second
theta = 60; % degrees

t=0:0.1:20; % time range for calculation (seconds)

% X-component
v_0x = v_0*cosd(theta); % calculating x component of initial velocity
a_x = 0;                % setting x component of acceleration

d_x = v_0x.*t + 0.5*a_x.*t.^2;    % calculating x positions

% Y-component
v_0y = v_0*sind(theta); % calculating y component of initial velocity
a_y = -9.8;             % setting y component of acceleration in m/s^2

d_y = v_0y.*t + 0.5*a_y.*t.^2;    % calculating y positions

% Zeroing points with negative y positions
 d_x(Y<0)=0;
 d_y(Y<0)=0;

% Plotting final results
figure(1)
plot(X,Y)
    xlabel('X position (m)')
    ylabel('Y position (m)')
        ax = gca;
        ax.FontSize = 18;

% All of this can be saved as a function like ProjectileMotion.m
```

Let's look at how a few of these variables are calculated:

```
t=0:0.1:20; % time range for calculation (seconds)
```

So, this is what t looks like:

t_0	t_1	…	t_{10001}

```
d_x = v_0x.*t + 0.5*a_x.*t.^2;    % calculating x positions
```

And since d_x is calculated by dot multiplication (i.e. element-by-element multiplication), it looks like this:

$d_x(t_0)$	$d_x(t_1)$	…	$d_x(t_{10001})$

where d_x has to be the same length as t.

How can you check if this is true? Type `whos` into the command line (or open up the workspace window) to check this.

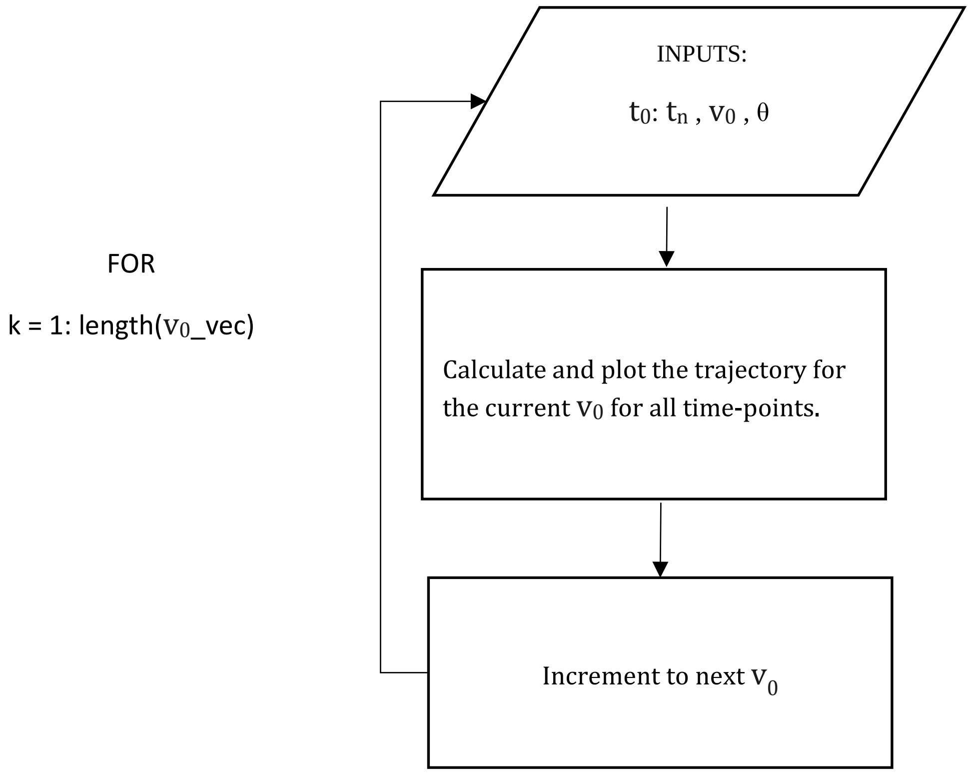
Is the final result that is displayed correct? How do you know?

```
%% Changing One Initial Parameter at a time (intro to for-loops)
```

What if we want to test v_0 from 1 m/s to 25 m/s?
Put a breakpoint at line 44 and hit 'run'!

```
v0_vec = 1:5:25; % setting a range of initial velocities
theta = 45;
```

For-loops are a useful tool for performing the same calculation multiple times and for recursive calculations.



```
%% Changing One Initial Parameter at a time (intro to for-loops)
```

Now, I want you to take the code that you have and plot trajectories for

$\vec{v}_0$ = 1 to 25 m/s, θ = 0 to 90^0

```
%looping through the range of initial velocities
for i = 1:length(v0_vec)
        figure(2)
        hold on
        v_0 = v0_vec(i); % setting v_0 each time

        [X,Y]=ProjectileMotion_fn(v_0,theta);
        %axis([0 1800 0 500])
end
title('Varying v_0')
    ax = gca;
    ax.FontSize = 18;

% Now we'll vary the initial angle
v0 = 25;
theta_vec = 0:5:90;

for i = 1:length(theta_vec)
        figure(3)
        hold on
        theta = theta_vec(i);

        [X,Y]=ProjectileMotion_fn(v_0,theta);
end
title('Varying Theta')
    ax = gca;
    ax.FontSize = 18;
```

How can we integrate varying both variables at once? I've given you the first few lines below:

```
%% Changing Two Parameters
% setting a range for both initial parameters
v0_vec = 1:5:25;
theta_vec = 0:5:90;
```

How would you continue from here?

```
%% Optimization (data/matrix visualization via imagesc)
```

If you look at lines 87 to 89 of version 2, you can see that I'm doing something a little extra here. What does it look like I'm doing and why?

Assignment:

Consider how accounting for drag would affect this problem.
Let's assume that the drag force, F_d, can be modeled as:

$$F_d = -b\cdot\vec{v}\ (t)$$

Where b is the drag coefficient. This leads to an acceleration due to drag that can be written as:

$$\vec{a} = -\frac{b}{m}\cdot\ \vec{v}\ (t)$$

For homework, add the impact of drag along both the x and y components to your existing model. Graph a similar series of modeled trajectories to model the impact of drag and determine what values would be need for v_0 with θ set at 45^0 to hit a target that is 8 meters away.

For the sake of simplicity, let $\frac{b}{m} = 1$, so that $\vec{a}(t) = -\vec{v}(t)$

Let's break this down like before:

Initial parameters

$\vec{v}_0$ = ?? m/s, θ = 60^0

Other variables:

t, $\vec{d}(t)$, $\vec{v}(t)$, $\vec{a}(t)$

Useful equations:

$$\vec{d} = \vec{v}_0 \cdot t + \frac{1}{2}\vec{a}\cdot t^2$$

$$\vec{v}^2 = \vec{v}_0^2 + 2\,\vec{a}\cdot\vec{d}$$

<u>Approach</u>: Break 2D problem into 2 1D problems

	x- component	y-component
$\vec{d}(t)$	$v_{0,x}t + \frac{1}{2}a_x(t)t^2$	$v_{0,y}t + \frac{1}{2}a_y(t)t^2$
$\vec{v}(t)$	$\sqrt{v_{0,x}^2 + 2a_x(t)d_x(t)}$	$\sqrt{v_{0,y}^2 + 2a_y(t)d_y(t)}$
$\vec{a}(t)$	$-v_x(t)$	$-g \pm v_y(t)$

Parameterizing the variables:

Since d, v, and a are all functions of time in this problem (i.e. they vary with time), we have to calculate them for all time points, which will look something like this:

$d_x(t_0)$	$d_x(t_1)$	…	$d_x(t_n)$

$v_x(t_0)$	$v_x(t_1)$	…	$v_x(t_n)$

$a_x(t_0)$	$a_x(t_1)$	…	$a_x(t_n)$

So, how do we go about this?
We know $d_x(t_0) = 0$ and $v_x(t_0) = v_{0,x}$. Let’s assume that $a_x(t_0) = 0$.
Now we have:

0	$d_x(t_1)$	…	$d_x(t_n)$

$v_{0,x}$	$v_x(t_1)$	…	$v_x(t_n)$

0	$a_x(t_1)$	…	$a_x(t_n)$

We can use the t_0 values to calculate the values for timepoint t_1:

$$v_x(t_1) = \sqrt{v_{0,x}^2 + 2a_x(t_0)d_x(t_0)}$$

$$a_x(t_1) = -v_x(t_0)$$

$$d_x(t_1) = v_{0,x}t_1 + \frac{1}{2}a_x(t_0)t_1^2$$

More generally, the values for each timepoint (t_n) can be calculated using the values for the previous timepoint (t_{n-1}).

$$v_x(t_n) = \sqrt{v_{0,x}^2 + 2a_x(t_{n-1})d_x(t_{n-1})}$$

$$a_x(t_n) = -v_x(t_{n-1})$$

$$d_x(t_n) = v_{0,x}t_{n-1} + \frac{1}{2}a_x(t_{n-1})t_n^2$$