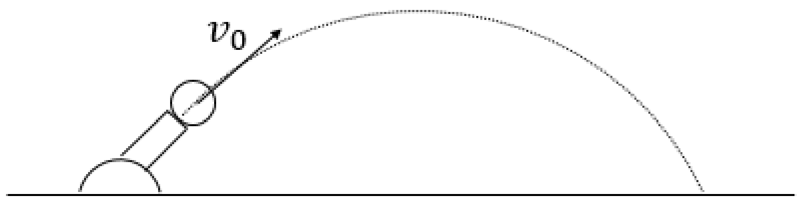
BIOE 198MI Biomedical Data Analysis. Spring Semester 2019.
Lab 2: Modelling Projectile Motion

Problem Statement
You are tasked with firing a cannonball the objective of maximizing the distance that the cannonball travels. You can control the angle of elevation, θ and initial velocity, v_0.



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?

A. PLOT TRAJECTORY AT A FIXED INITIAL ANGLE

Useful equations:

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

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
$\vec{d}(t)$	$v_{0,x} \cdot t + \frac{1}{2} a_x \cdot\ t^2$	$v_{0,y} \cdot t + \frac{1}{2} a_y \cdot\ t^2$
$\vec{v}_0$	$v_0 \cos\theta$	$v_0 \sin\theta$
$\vec{a}$	$0\ m/s^2$	$-9.8\ m/s^2$

Strategy to plot the trajectory:

$$y(x) \Rightarrow \begin{cases} x = x(t) \\ y = y(t) \end{cases}$$

We transform our equation into a parametric equation. For given time *t*, we can calculate the corresponding *x* and *y*, then plot them in the figure.

```
%% Modeling & Plotting Projectile Motion
clear all; close all
%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 velecotiy
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

% Truncate the trajectory such that the curve is above the horizontal line
 d_xplot=d_x(d_y>=0);
 d_yplot=d_y(d_y>=0);

% Plotting final results
figure(1)
plot(d_xplot,d_yplot)
    xlabel('X position (m)')
    ylabel('Y position (m)')
hold on
```

There is another way of plotting this figure, we can try it and compare it with the first figure.
How to do that?

Here is my code:

```
%% Another way of plotting it
%Setting initial variables
v_0 = 25; % meters/second
theta = 60; % degrees

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

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

d_x=0:1:60; % assign the range of displacement in x direction
```

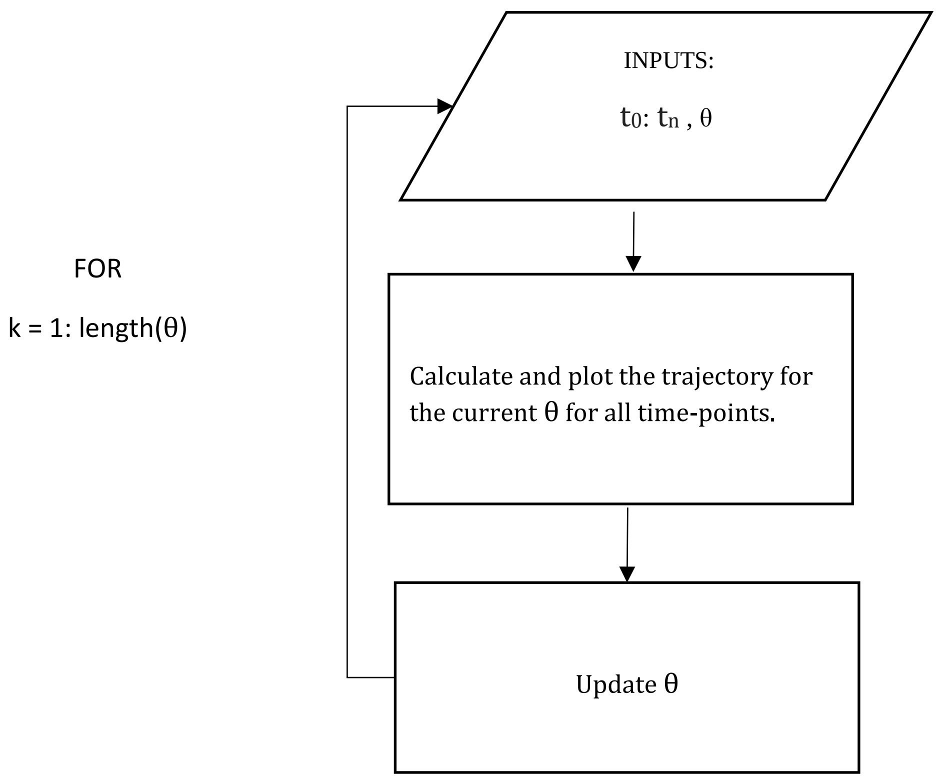
```
                    % calculate the displacement in y direction(fill this
            line)

% Truncate the trajectory such that the curve is above the horizontal line
 d_xplot=d_x(d_y>=0);
 d_yplot=d_y(d_y>=0);

% Plotting final results
figure(1)
plot(d_xplot,d_yplot,'o')
    xlabel('X position (m)')
    ylabel('Y position (m)')
hold off
```

Which method is good, why?

B. OPTIMIZATION ON INITIAL ANGLE

Still use the same code in part A(first method). Changing initial angle and see what you get.
Try $\theta = 20^{\circ}, 40^{\circ}, 60^{\circ}, 80^{\circ}$.

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



Here is the code:

```
clear all; close all
%Setting initial variables
v_0 = 25; % meters/second

t=0:0.1:20; % time range for calculation (seconds)
a_x = 0;                % setting x component of acceleration
a_y = -9.8;             % setting y component of acceleration in m/s^2
theta_range=0:1:89;%%varying theta from 0 to 89
dist=zeros(1,length(theta_range));%%store the distance value for each theta

for i=1:length(theta_range)

                           %get current theta value
                           % calculating x component of initial velocity
                           % calculating y component of initial velecotiy
                           % calculating x positions
                           % calculating y positions

    % Truncate the trajectory such that the curve is above the horizontal
line
    d_xplot=d_x(d_y>=0);
    d_yplot=d_y(d_y>=0);

    dist(i)=max(d_xplot);
end


figure(2)

%plot thet result and see how dist change with respect to theta(fill this
line)
hold on
```

What kind of shape is it? Can you explain it with mathematical expression?

C. OPTIMIZATION ON INITIAL VELOCITY

Similar process as part B. The only difference is that, instead of varying theta, we are going to vary v_o.

```
clear all; close all
%Setting initial variables
theta = 60; % meters/second

t=0:0.1:20; % time range for calculation (seconds)
a_x = 0;                % setting x component of acceleration
a_y = -9.8;             % setting y component of acceleration in m/s^2
v0_range=10:1:30;       % varying theta from 10 to 30

%% finish the rest of your code and plot your result
```

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 18 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 = 45^0

Other variables:

t, $\vec{d}(t)$, $\vec{v}(t)$, $\vec{a}(t)$, $g = -9.8m/s^2$

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

	x- component	y-component
$\vec{d}(t)$	$-v_{0,x}e^{-t} + v_{0,x}$	$g * t - \left(v_{0,y} - g\right) * (e^{-t} - 1)$
$\vec{v}(t)$	$v_{0,x}e^{-t}$	$g + (v_{0,y} - g)e^{-t}$
$\vec{a}(t)$	$-v_x(t)$	$g - v_y(t)$

Hint:

1. Use similar code in part C to get the right answer.

2. Searching range of $\vec{v}_0$ is $[10,30]$

3. Since it is difficult to get exact the same answer. After you get *dist* vector which has the same meaning in part C. We can use the following code to get the final answer.

```
[value,position]=min(abs(dist-18));
v0_ans=v0_range(position);
```