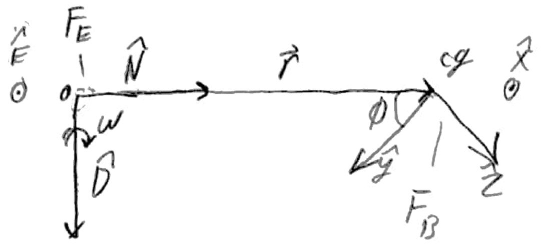
Samuel Mazurovskiy | ASEN 3128 - HW1 | due 9-12-19



$|\vec{r}| = 100\,m$
$|\vec{V}^E| = 10\,\frac{m}{s}$

1) $\vec{r}$ is the vector from o to the cg of the aircraft
$\vec{r}_E$ is the vector from o to the cg in E frame coordinates
$\vec{r}_E$ would be represented as $\begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix}[m]$

$\vec{r}_B$ is the vector from o to the cg in B frame coordinates
$\vec{r}_B$ would be represented as $\begin{bmatrix} 0 \\ -100\cos\phi \\ 100\sin\phi \end{bmatrix}[m]$

2) $\vec{V}^E = \frac{d}{dt}^E \vec{r}$ says the inertial velocity vector is equal to the E frame derivative of $\vec{r}$

$\vec{V}^E_E$ is the inertial velocity in E frame coordinates represented as $\vec{V}^E_E = [0\ 10\ 0]^T [\frac{m}{s}]$

$\vec{V}^E_B$ is the inertial velocity in B frame coordinates represented as $\vec{V}^E_B = [10\ 0\ 0]^T [\frac{m}{s}]$

3) $\vec{\omega}^{EB}$ is the angular velocity of the B frame as seen from E

$\vec{\omega}^{EB}_E$ is $\vec{\omega}^{EB}$ represented in E frame coordinates as $\begin{bmatrix} 0 \\ 0 \\ .1 \end{bmatrix}[\frac{rad}{s}]$

$\vec{\omega}^{EB}_E$ is $\vec{\omega}^{EB}$ represented in B frame coordinates as $\begin{bmatrix} 0 \\ .1\sin(\phi) \\ .1\cos(\phi) \end{bmatrix}[\frac{rad}{s}]$

4) $\frac{d^B}{dt}\vec{r}$ is the B frame derivative of $\vec{r}$

$\left(\frac{d^B}{dt}\vec{r}\right)_E$ is the B frame derivative of $\vec{r}$ in E frame coordinates represented as $[0\ 0\ 0]^T\left[\frac{m}{s}\right]$

$\left(\frac{d^B}{dt}\vec{r}\right)_B$ is the B frame derivative of $\vec{r}$ in B frame coordinates represented as $[0\ 0\ 0]^T\left[\frac{m}{s}\right]$

5) $\left(\frac{d^E}{dt}\vec{V}^E\right)_E$ is the E frame derivative of the inertial velocity in E frame coordinates represented as $[-1\ 0\ 0]\left[\frac{m}{s^2}\right]$

$\left(\frac{d^B}{dt}\vec{V}^E\right)_B$ is the B frame derivative of the inertial velocity in B frame coordinates represented as $[0\ 0\ 0]\left[\frac{m}{s^2}\right]$

6) $\frac{d^E}{dt}\vec{V}^E = \cancel{\frac{d^B}{dt}\vec{V}^E}^{\,0} + \vec{\omega}^{EB} \times \vec{V}^E$ $\quad \begin{vmatrix} \hat{N} & \hat{E} & \hat{D} \\ 0 & 0 & .1 \\ 0 & 10 & 0 \end{vmatrix} = -\hat{N}\left[\frac{m}{s^2}\right]$

$$\frac{d^E}{dt}\vec{V}^E = -\hat{N}\left[\frac{m}{s^2}\right]$$

7) $\vec{f} = m\vec{a}$ which is ~~is~~ a vector

$\vec{f}_E = m\left(\frac{d^E}{dt}\vec{V}^E\right)_E$ which is $[-m\ 0\ 0]^T[N]$

$\vec{f}_B = m\left(\frac{d^E}{dt}\vec{V}^E\right)_B$ which is $\begin{bmatrix} 0 \\ m\cos\phi \\ m\sin\phi \end{bmatrix}[N]$

8) $\vec{V}^E = \vec{W} + \vec{V}$

$\vec{V}^E = 10\hat{E}\left[\frac{m}{s}\right]$ $\vec{W} = 2\hat{N} + 3\hat{E} - 1\hat{D}\left[\frac{m}{s}\right]$

$\vec{V} = \vec{V}^E - \vec{W} = -2\hat{N} + 7\hat{E} + 1\hat{D}\left[\frac{m}{s}\right]$

$\boxed{\vec{V} = -2\hat{N} + 7\hat{E} + 1\hat{D}\left[\frac{m}{s}\right]}$

$\vec{V}_B = [-7 \quad -2\cdot -\cos\phi + \sin\phi \quad -2\cdot -\sin\phi + \cos\phi]$

$\boxed{\vec{V}_B = [7 \quad 2\cos\phi + \sin\phi \quad 2\sin\phi + \cos\phi]}$

Table of Contents

Header

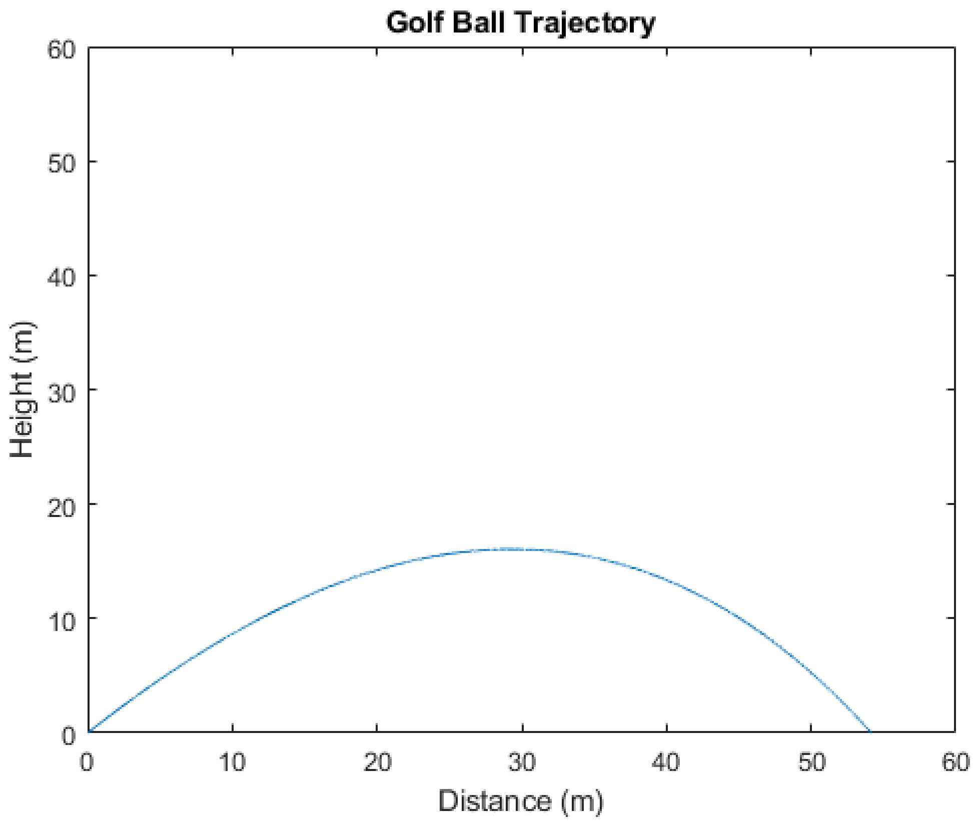
```
% Name: Samuel Razumovskiy
% Class: ASEN 3128
% Assignment: 1
% Due: 9/12/2019

clear,clc,close all
```

A) Checking if Results Make Sense

```
% Initial conditions for velocity, wind, and position
VN = 0;
VE = 20;
VD = -20;

N = 0;
E = 0;
D = 0;

WN = 0;
WE = 0;
WD = 0;

% Constants
rho = 1.225;
mass = 0.03;
area = pi*(.015)^2;
Cd = 0.6;
g = 9.81;
t = [0 5];

% Inits for ODE45
inits = [VN,VE,VD,N,E,D];

% ODE45
opt = odeset('maxstep',0.001);
[t, pos] = ode45(@(t,X)
 Integrate(t,X,WN,WE,WD,rho,mass,area,Cd,g),t,...
    inits,opt);

% Index of D positions below 0
```

```
ind = find(pos(:,6)<=0);

fprintf("The trajectory makes sense and seems to behave accurately\n")

% Ploting with up as positive so D pos is negative
figure(1)
plot(pos(ind,5),-pos(ind,6))
xlabel("Distance (m)")
ylabel("Height (m)")
title("Golf Ball Trajectory")
ylim([0 60])
```

The trajectory makes sense and seems to behave accurately



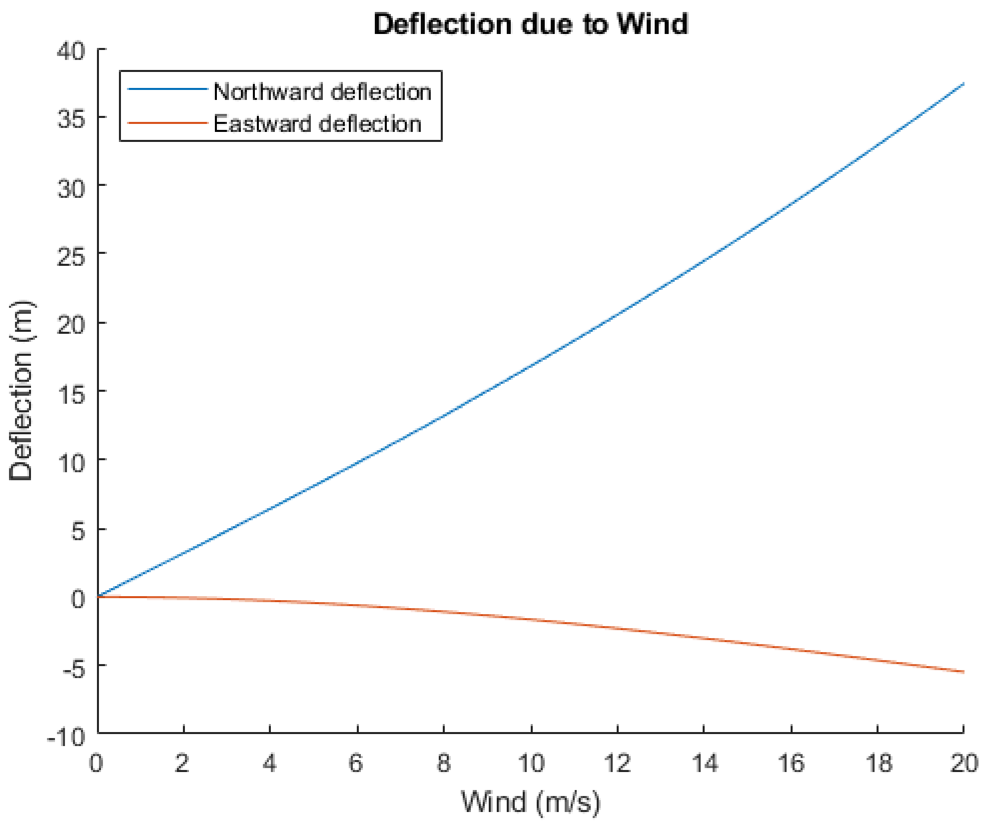
B) Checking Sensitivity to Wind

```
clear,clc,close all

% Initial conditions for velocity, wind, and position
VN = 0;
VE = 20;
VD = -20;

N = 0;
E = 0;
D = 0;
```

```
WN = linspace(0,20);
WE = 0;
WD = 0;

% Constants
rho = 1.225;
mass = 0.03;
area = pi*(.015)^2;
Cd = 0.6;
g = 9.81;
t = [0 5];
defN = zeros(1,100);
defE = zeros(1,100);

% Inits for ODE45
inits = [VN,VE,VD,N,E,D];

% Loop for a range of wind values
for i=1:length(WN)
    % ODE45
    opt = odeset('maxstep',0.001);
    [t, pos] = ode45(@(t,X)
 Integrate(t,X,WN(i),WE,WD,rho,mass,area,Cd,g),t,...
        inits,opt);
    % Storing deflection horizontally
    defN(i) = pos(end,4);
    defE(i) = pos(end,5);
end

% Normalizing deflection in the E
defE = defE-defE(1);

% Finding equation for wind vs. deflection
EquN = polyfit(WN,defN,1);
EquE = polyfit(WN,defE,1);

% Slope of the line
slopeN = EquN(1);
slopeE = EquE(1);

fprintf("The northward deflection in m per m/s of wind is %.3f m/(wind
 m/s)\n"...
    ,slopeN)
fprintf("While the eastward deflection was %.4f m/(wind m/s)",slopeE)

% Plot of the wind vs. deflection
figure(2)
hold on
plot(WN,defN)
plot(WN,defE)
xlabel("Wind (m/s)")
ylabel("Deflection (m)")
title("Deflection due to Wind")
```

```
legend("Northward deflection","Eastward
 deflection","location","northwest")
hold off
```

The northward deflection in m per m/s of wind is 1.856 m/(wind m/s)
While the eastward deflection was -0.2858 m/(wind m/s)

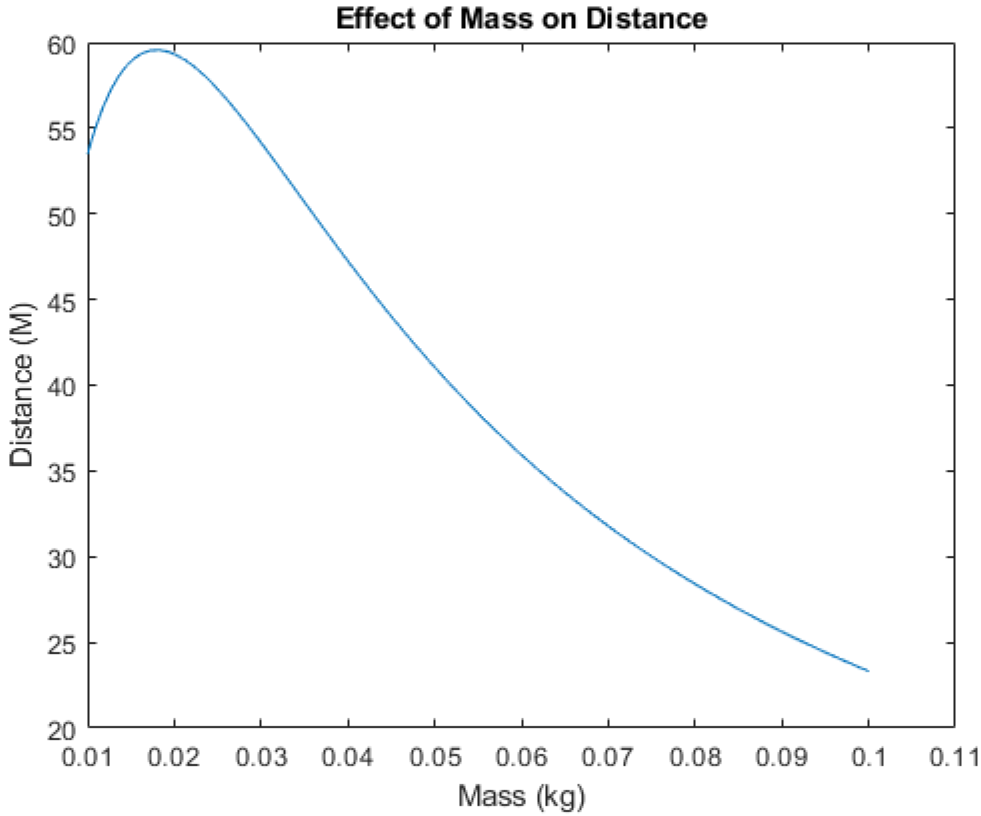


C) Finding How Mass Affects Distance

```
clear,clc,close all

% Range of Masses
m = linspace(0.01,.1);

% Initializing variable for max range
Dis = zeros(1,100);

% Looping for the range of masses
for i = 1:length(m)

    % Velocity dependent on the mass
    V = sqrt(12*2/m(i));

    % Initial conditions for velocity, wind, and position
    VN = 0;
    VE = V*cosd(45);
```

```
    VD = -V*cosd(45);

    N = 0;
    E = 0;
    D = 0;

    % Setting wind to 0 since only looking at distance
    WN = 0;
    WE = 0;
    WD = 0;

    % Constants
    rho = 1.225;
    mass = m(i);
    area = pi*(.015)^2;
    Cd = 0.6;
    g = 9.81;
    t = [0 5];

    % Inits for ODE45
    inits = [VN,VE,VD,N,E,D];

    % ODE45
    opt = odeset('maxstep',0.001);
    [t, pos] = ode45(@(t,X)
 Integrate(t,X,WN,WE,WD,rho,mass,area,Cd,g),t,...
        inits,opt);

    % Storing max range
    pos2 = pos(pos(:,6)<=0,5);
    Dis(i) = pos2(end);
end

% Maximum value of the different ranges and its index
[maxD, I] = max(Dis);

% Mass for the max distance
maxMass = m(I);

fprintf("A slightly lighter ball seems to get a better range\n")
fprintf("The maximum range found was %.3f m using a mass of %.3f kg
\n",maxD,maxMass)
fprintf("Which is less than the starting mass of 0.03 kg")

% Plotting mass vs. max distance
figure(3)
plot(m,Dis)
xlabel("Mass (kg)")
ylabel("Distance (M)")
title("Effect of Mass on Distance")
```

A slightly lighter ball seems to get a better range
The maximum range found was 59.560 m using a mass of 0.018 kg
Which is less than the starting mass of 0.03 kg



Published with MATLAB® R2018b

```
function [d_dt] = Integrate(t,X,WN,WE,WD,rho,mass,A,Cd,g)
% ODE45 function to integrate a flight path with drag

% Defining position and velocity although position is not used
VN = X(1);
VE = X(2);
VD = X(3);

N = X(4);
E = X(5);
D = X(6);

% Calculating the relative wind
VNrel = VN-WN;
VErel = VE-WE;
VDrel = VD-WD;

% Calculating the relative wind magnitude
Vrel = sqrt(VNrel^2+VErel^2+VDrel^2);

% Calculating the relative wind cosine directions
Nalpha = acos(VNrel/Vrel);
Ebeta = acos(VErel/Vrel);
Dgamma = acos(VDrel/Vrel);

% Calculating drag with relative wind
Drag = -Cd*rho*A*Vrel^2/2;

% Calculating acceleration
dVN_dt = Drag*cos(Nalpha)/mass;
dVE_dt = Drag*cos(Ebeta)/mass;
dVD_dt = Drag*cos(Dgamma)/mass + g;

% Applying velocity
dN_dt = VN;
dE_dt = VE;
dD_dt = VD;

d_dt = [dVN_dt;dVE_dt;dVD_dt;dN_dt;dE_dt;dD_dt];
end
```

Published with MATLAB® R2018b