Table of Contents

Milestone 1 Parts 1 and 2

```
close all;
clc;
clear;

%%Importing data and creating a lift curve slope by plotting alpha vs.
 CL:
%Import Tempest UA data table first, then redo the import data command
 with
%the different file name.
```

Variables:

```
e = 0.9;
AR_Tempest = 16.5; %For the Tempest UA, given from the characteristics
AR_Boeing = 7;
%a1 is the 3D Lift Curve slope for the Tempest. a0 is the 2D Lift
 Curve
%slope for the Tempest.
%a2 is the 3D Lift Curve slope for the Boeing. a02 is the 2D Lift
 Curve
%slope for the Boeing.
```

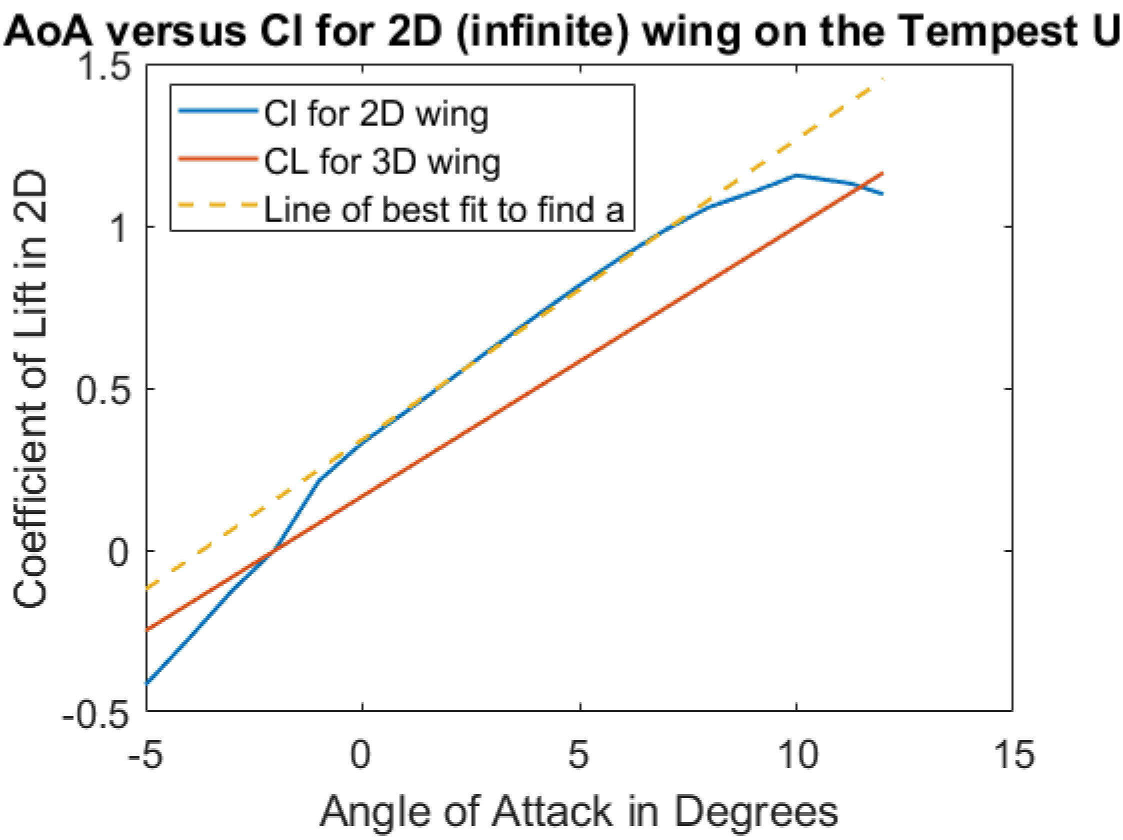
Part 1 Getting the Lift Curve Comparison CL vs AoA

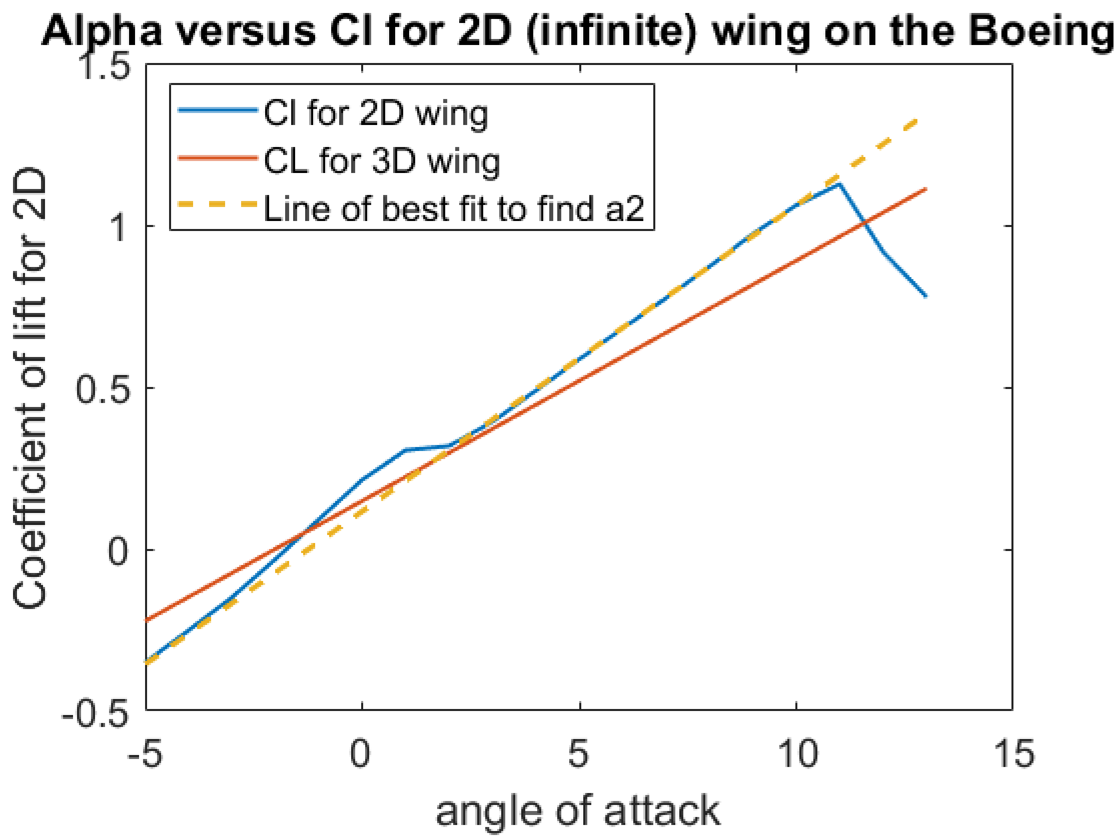
```
                    %Getting CL vs. AoA for the TempestUA

Tempest = xlsread('TempestAirfoilData.xlsx', 'A1:C18');
% Tempest = readtable('TempestAirfoilData.xlsx');
 a_Tempest = Tempest(:,1);
 Cl_Tempest = Tempest(:, 2);
 Cd_Tempest = Tempest(:,3);
 %Finding a0:
 slope1 = polyfit(a_Tempest(6:14), Cl_Tempest(6:14),1);
 m = slope1(1);
```

```
 b = slope1(2);
 model1 = a_Tempest.*m + b;
 %a0 = mean(model1);
 %will assume the average of the polyfit line will be a0. To make it
 more
 %specific get the values of two points and get the slope that way
 later.
 a0 = m; %(a0 is basically the slope of the polyfit that we found.)

 %Get a1 (3D Lift curve slope) for the Tempest
 a1 = a0 / (1 + ((57.3*a0)/(pi*e*AR_Tempest)));
 %Now calculating the 3D CL for the finite wing: AoA at L = 0 is -2
 CL_Tempest = a1*(a_Tempest(:)-a_Tempest(4)); %Iterated through all of
 the AoAs in the dataset.

figure(1)
plot(a_Tempest, Cl_Tempest, 'Linewidth', 1.5); %Line of best fit to
 find a0.
hold on
plot(a_Tempest, CL_Tempest, 'Linewidth', 1.5); %Plotting 3D CL.
plot(a_Tempest, model1, '--', 'Linewidth', 1.5)
set(gca,'Fontsize',15)
title('AoA versus Cl for 2D (infinite) wing on the Tempest UA');
xlabel('Angle of Attack in Degrees');
ylabel('Coefficient of Lift in 2D');
legend('Cl for 2D wing', 'CL for 3D wing', 'Line of best fit to find
 a', 'Location', 'Northwest');
hold on

                   % Getting CL vs. AoA for the Boeing
Boeing = xlsread('BoeingAirfoilData.xlsx', 'A1:C19');
a_Boeing = Boeing(:,1);
Cl_Boeing = Boeing(:,2);
Cd_Boeing = Boeing(:,3);
%Finding a0:
 slope2 = polyfit(a_Boeing(8:16), Cl_Boeing(8:16),1);
 m2 = slope2(1);
 b2 = slope2(2);
 model2 = a_Boeing.*m2 + b2;
%%3D: CL vs. AoA PLEASE NOTE a02 AND a2 are the slopes for the boeing!
%----
a02 = m2; %(run of 1 so nothing in denominator.)
 %Get a (3D Lift curve slope)
 a2 = a02 / (1 + ((57.3*a02)/(pi*e*AR_Boeing)));
 %Now calculating the 3D CL for the finite wing: AoA at L = 0 is -2
 CL_Boeing = a2*(a_Boeing(:)-a_Boeing(4)); %Iterated through all
 of the AoAs in the dataset. AoA when L=0 is when the lift curve
 intercepts the x-axis.


figure(2)
% plot(a_Tempest, model2, '--', 'Linewidth', 1.5) Getting vector
 error.
plot(a_Boeing,Cl_Boeing, 'Linewidth',1.5)
```

```
hold on
plot(a_Boeing, CL_Boeing, 'Linewidth', 1.5)
plot(a_Boeing, model2, '--', 'Linewidth', 2)
set(gca,'Fontsize',15)
title('Alpha versus Cl for 2D (infinite) wing on the Boeing')
xlabel('angle of attack')
ylabel('Coefficient of lift for 2D')
legend('Cl for 2D wing', 'CL for 3D wing', 'Line of best fit to find
 a2', 'Location', 'Northwest');
hold off
```

Warning: Could not start Excel server for import, 'basic' mode will be used.
Refer to HELP XLSREAD for more information.
Warning: Could not start Excel server for import, 'basic' mode will be used.
Refer to HELP XLSREAD for more information.





Part 2 Drag Polar Comparison (C_D vs. C_L) Plotting the same figure on

a) the 3D finite wing drag polar (use equation 3 from the lab doc) b) the whole aircraft drag polar (equation 4-9) c) The drag polar provided in the aircraft data file. (see data set)

```
%a)
CD_wing_Tempest = Cd_Tempest + (CL_Tempest.^2)/(pi*e*AR_Tempest);
CD_wing_Boeing = Cd_Boeing + (CL_Boeing.^2)/(pi*e*AR_Boeing);


%b)
e0Tempest = 1.78*(1-(0.045*AR_Tempest^0.68))-0.64;
%  e0Boeing = 1.78*(1-(0.045*AR_Boeing^0.68))-0.64;
e0Boeing = 4.61*(1-(0.045*AR_Boeing^0.68))*(cos(37.5)^0.15)-3.1;

S_wet_Tempest = 2.04;
S_ref_Tempest = 0.63;
S_wet_Boeing = 2535;
S_ref_Boeing = 511;

Cfe_Tempest = 0.0055; %Tempest
Cfe_Boeing = 0.0030;
 CD_min_Tempest = Cfe_Tempest*(S_wet_Tempest/S_ref_Tempest);
 CD_min_Boeing = Cfe_Boeing*(S_wet_Boeing/S_ref_Boeing);
```

```
% CL_minD_Tempest = 0.1662;
% CL_minD_Boeing = 0.08465;

[~,i3] = min(CD_wing_Tempest);
[~,i4] = min(CD_wing_Boeing);

CL_minD_Tempest = CL_Tempest(i3);
CL_minD_Boeing = CL_Boeing(i4);

k1Temp = 1/(pi*e0Tempest*AR_Tempest);
k1Boeing = 1/(pi*e0Boeing*AR_Boeing);

 WholeAircraftPolarTempest = (CD_min_Tempest + k1Temp*(CL_Tempest -
 CL_minD_Tempest).^2);
 WholeAircraftPolarBoeing = (CD_min_Boeing + k1Boeing*(CL_Boeing -
 CL_minD_Boeing).^2);



%c)Graphing the "Truth Data" of the finite wing.
 DPolarTruthTemp = xlsread('TempestTruthData.xlsx', 'A1:C18'); %Drag
 polar truth data
  CL_TruthTempest = DPolarTruthTemp(:,2);
  CD_TruthTempest = DPolarTruthTemp(:,3);

  DPolarTruthBoeing = xlsread('BoeingTruthData.xlsx', 'A1:B25'); %Drag
 polar truth data
  CL_TruthBoeing = DPolarTruthBoeing(:,1);
  CD_TruthBoeing = DPolarTruthBoeing(:,2);



 % fix labels and legend and split the drag graphs to Tempest and
 Boeing.



% Plotting the different drags for Boeing and Tempest:
```

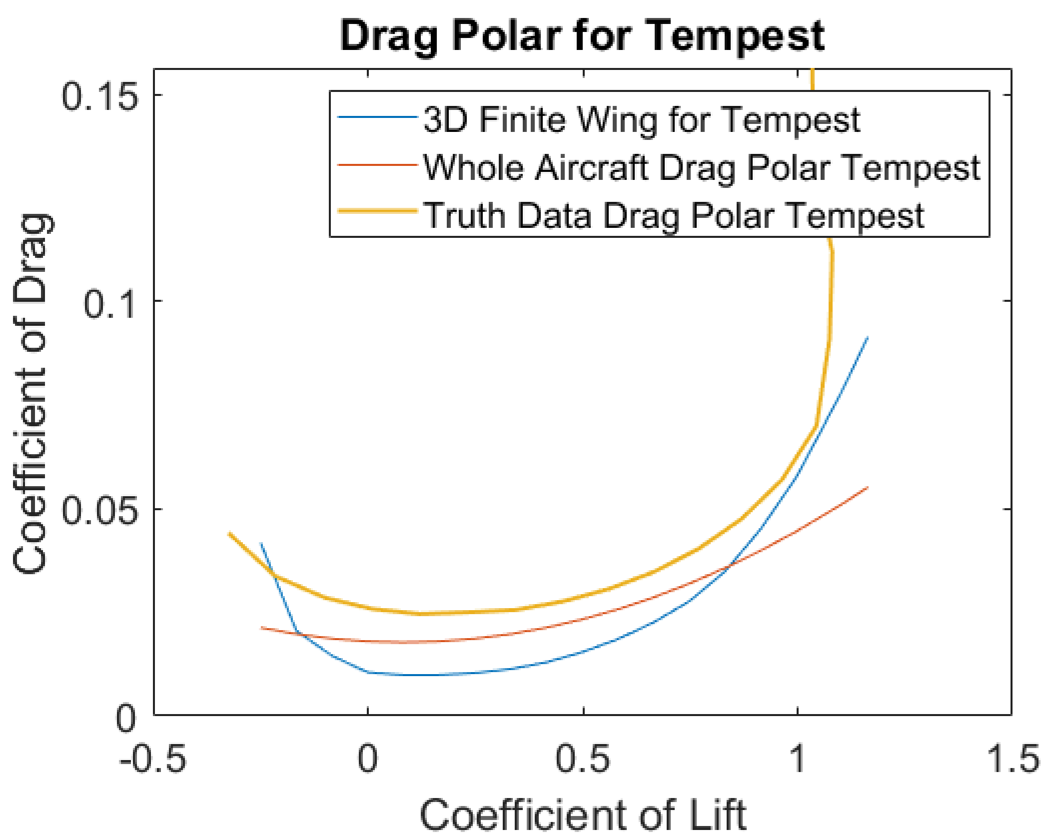
```
Warning: Could not start Excel server for import, 'basic' mode will be
 used.
Refer to HELP XLSREAD for more information.
Warning: Could not start Excel server for import, 'basic' mode will be
 used.
Refer to HELP XLSREAD for more information.
```

Tempest:

Wing drag for Tempest

```
 figure(3)
```

```
plot( CL_Tempest, CD_wing_Tempest)
hold on
 % Full aircraft drag polar for Tempest:
 plot(CL_Tempest, WholeAircraftPolarTempest)
%Truth data for Tempest:
 plot(CL_TruthTempest, CD_TruthTempest, 'Linewidth', 1.5);
 set(gca,'Fontsize',15)
 title('Drag Polar for Tempest')
 legend('3D Finite Wing for Tempest','Whole Aircraft Drag Polar
 Tempest ','Truth Data Drag Polar Tempest')
 xlabel('Coefficient of Lift')
 ylabel('Coefficient of Drag')
 hold off
```



Boeing:

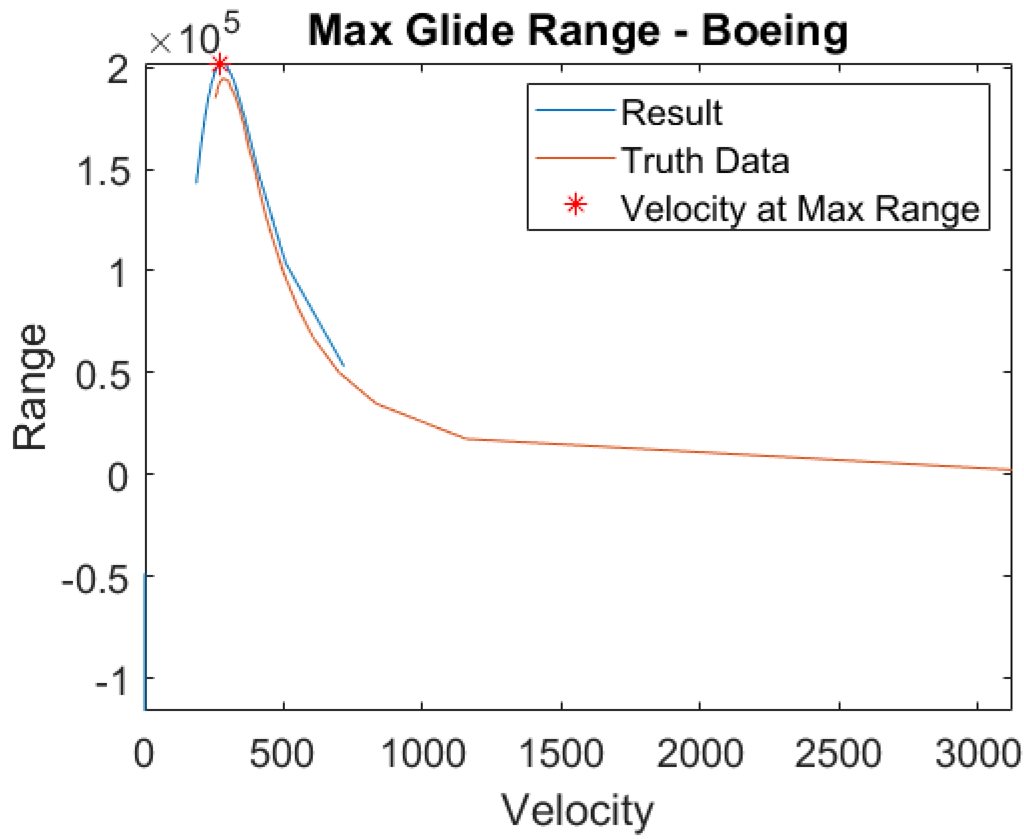
Wing drag for Boeing

```
 figure(4)
 plot(CL_Boeing, CD_wing_Boeing)
 hold on
 % Full aircraft drag polar for Boeing:
 plot(CL_Boeing, WholeAircraftPolarBoeing)
 % Truth data for Boeing:
  plot(CL_TruthBoeing, CD_TruthBoeing, 'Linewidth', 1.5);
 set(gca,'Fontsize',15)
```

```
 title('Drag Polar for Boeing')
 legend('3D Finite Wing for Boeing','Whole Aircraft Drag Polar Boeing
','Truth Data Drag Polar Boeing')
 xlabel('Coefficient of Lift')
 ylabel('Coefficient of Drag')
 hold off



 %Error Calculations

Error1 = CD_TruthTempest(1:end)-WholeAircraftPolarTempest(1:end);
Error2 = CD_TruthBoeing(1:19)-WholeAircraftPolarBoeing(1:19);

abs(mean(Error1))
abs(mean(Error2))


%L/D Calculations

LoverD_Tempest =  CL_Tempest./WholeAircraftPolarTempest;
LoverD_Truth_Temp = CL_TruthTempest./CD_TruthTempest;
LoverD_Boeing = CL_Boeing./WholeAircraftPolarBoeing;
LoverD_Truth_Boeing = CL_TruthBoeing./CD_TruthBoeing;

%L/D vs AoA - Tempest
plot(a_Tempest,LoverD_Tempest)
hold on
plot(a_Tempest,LoverD_Truth_Temp)
set(gca,'Fontsize',15)
title('L/D vs AoA - Tempest')
legend('Result','Truth Data')
xlabel('Angle of Attack')
ylabel('L/D')
hold off

max(LoverD_Tempest)
max(LoverD_Boeing)
max(LoverD_Truth_Temp)
max(LoverD_Truth_Boeing)


%Range & Endurance Factors

h_tempest = 1500; %m
h_Boeing = 10668; %m
W_tempest = 62.78; %N GTOW = 6.4kg
W_Boeing = 3706634.37564; %N GTOW = 833,000lb
rho_temp = 1.0581; %kg/m^3 @std atm 1500m
rho_Boeing = 0.38035; %kg/m^3 @std atm 35000ft



%Boeing
```

```
%%MAX GLIDE RANGE - @(L/D)max CD0 = kCL^2
v_gliderange_Boeing = real(sqrt(W_Boeing./
(.5*rho_Boeing*CL_Boeing*S_ref_Boeing)));
v_gliderange_Truth_Boeing = real(sqrt(W_Boeing./
(.5*rho_Boeing*CL_TruthBoeing*S_ref_Boeing)));
R_Boeing = h_Boeing*LoverD_Boeing;
R_truth_Boeing = h_Boeing*LoverD_Truth_Boeing;

[~,i] = max(R_Boeing);

plot(v_gliderange_Boeing,R_Boeing)
hold on
plot(v_gliderange_Truth_Boeing,R_truth_Boeing)
plot(v_gliderange_Boeing(i),R_Boeing(i),'r*','MarkerSize',8);
title('Max Glide Range - Boeing')
set(gca,'Fontsize',15)
legend('Result','Truth Data','Velocity at Max Range')
xlabel('Velocity')
ylabel('Range')
hold off
v_range_max_Boeing = v_gliderange_Boeing(i)

vmax_glide_truth =
 V1(CL_TruthBoeing,LoverD_Truth_Boeing,W_Boeing,rho_Boeing,S_ref_Boeing)


%Max Powered Range

v_range_max_Boeing1 =
 V2(CL_Boeing,CD_min_Boeing,W_Boeing,rho_Boeing,S_ref_Boeing,k1Boeing)
v_range_max_truthBoeing =
 V2(CL_TruthBoeing,CD_min_Boeing,W_Boeing,rho_Boeing,S_ref_Boeing,k1Boeing)

%Max Powered Endurance
%Same as Max Glide Range (L/Dmax)



%Tempest:
%%MAX GLIDE RANGE - @(L/D)max CD0 = kCL^2
v_gliderange_tempest = real(sqrt(W_tempest./
(.5*rho_temp*CL_Tempest*S_ref_Tempest)));
R = h_tempest*LoverD_Tempest;
R_truth = h_tempest*LoverD_Truth_Temp;

[~,i] = max(R);

plot(v_gliderange_tempest,R)
hold on
plot(v_gliderange_tempest,R_truth)
plot(v_gliderange_tempest(i),R(i),'r*','MarkerSize',8);
title('Max Glide Range - Tempest')
set(gca,'Fontsize',15)
```

```
legend('Result','Truth Data','Velocity at Max Range')
xlabel('Velocity')
ylabel('Range')
hold off
v_range_max = v_gliderange_tempest(i)


%Max Powered Range - Prop/Tempest
%Same as Max Glide Range


%Max Powered Endurance - Tempest: 3CD0 = kCL^2

V_Emax_Tempest =
 V3(CL_Tempest,CD_min_Tempest,W_tempest,rho_temp,S_ref_Tempest,k1Temp)
%Truth data uses calculated CD_min
V_Emax_Tempest_Truth =
 V3(CL_TruthTempest,CD_min_Tempest,W_tempest,rho_temp,S_ref_Tempest,k1Temp)


ans =

    0.0277


ans =

    0.0113


ans =

   23.3761


ans =

   18.9492


ans =

   19.1149


ans =

   18.2626


v_range_max_Boeing =

  271.0175
```

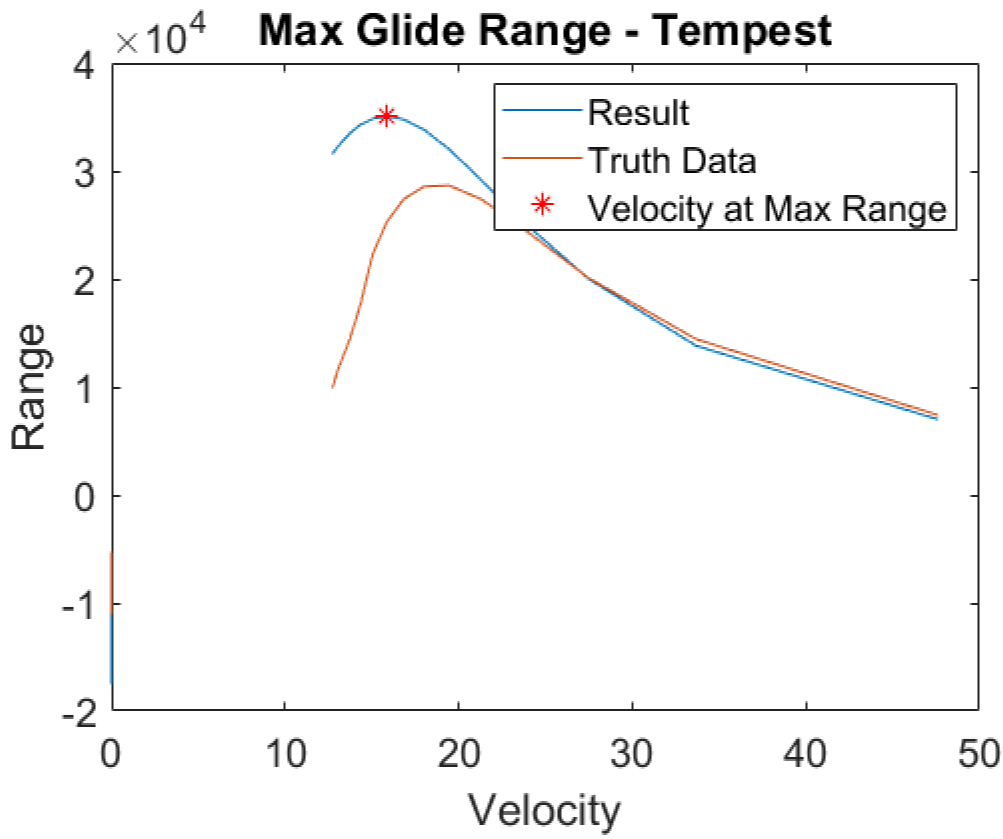


Functions

```
function [velocity] = V1(CL_Vec,LoD_Vec,Weight,rho,area)%L/D,CL,
 Wieght, alt, rho, Wing Area
[~,i] = max(LoD_Vec);
CL_Glide= CL_Vec(i);
velocity = real(sqrt(Weight./(.5*rho*CL_Glide*area)));
end

function [V_best] = V2(CL_Vec,CD_min,Weight,rho,area,k) %CD_min,
 CL_vec, Wieght, alt, rho, Wing Area
%Finds the CL closes to CD0 = 3kCl^2
    %CD0 = 3kC^2
    %CD0 - (3kC^2) = 0;
    diffVec = abs(CD_min-(3*k*(CL_Vec.^2)));
    [~,i]=min(diffVec);
    CL = CL_Vec(i);
    V_best = real(sqrt(Weight./(.5*rho*CL*area)));
end

function [V_best] = V3(CL_Vec,CD_min,Weight,rho,area,k)%CD_min,
 CL_vec, Wieght, alt, rho, Wing Area
%Finds the CL closes to 3CD0 = kCl^2
    %3CD0 = 3kC^2
    %3*CD0 - (kC^2) = 0;
```

```
    diffVec = abs((3*CD_min)-(k*(CL_Vec.^2)));
    [~,i]=min(diffVec);
    CL = CL_Vec(i);
    V_best = real(sqrt(Weight./(.5*rho*CL*area)));
end
```

```
vmax_glide_truth =

  284.1494


v_range_max_Boeing1 =

  358.5224


v_range_max_truthBoeing =

  373.4506


v_range_max =

   15.8703


V_Emax_Tempest =

   12.7246


V_Emax_Tempest_Truth =

   13.2020
```



Published with MATLAB® R2019b