ASEN 3802 Lab 3 Part 3

Andrew Patella, Niko Pappas, Dane Shedd, Lorien Hoshall

May 1, 2025

1 Part 3 Task 1: Effect of Angle of Attack on Coefficient of Lift

1.1 Deliverable: Coefficient of Lift (Plot)

Using your NACA airfoil generator, the provided vortex panel code, and your PLLT code, generate a plot of coefficient of lift versus angle of attack for the Cessna 180 aircraft

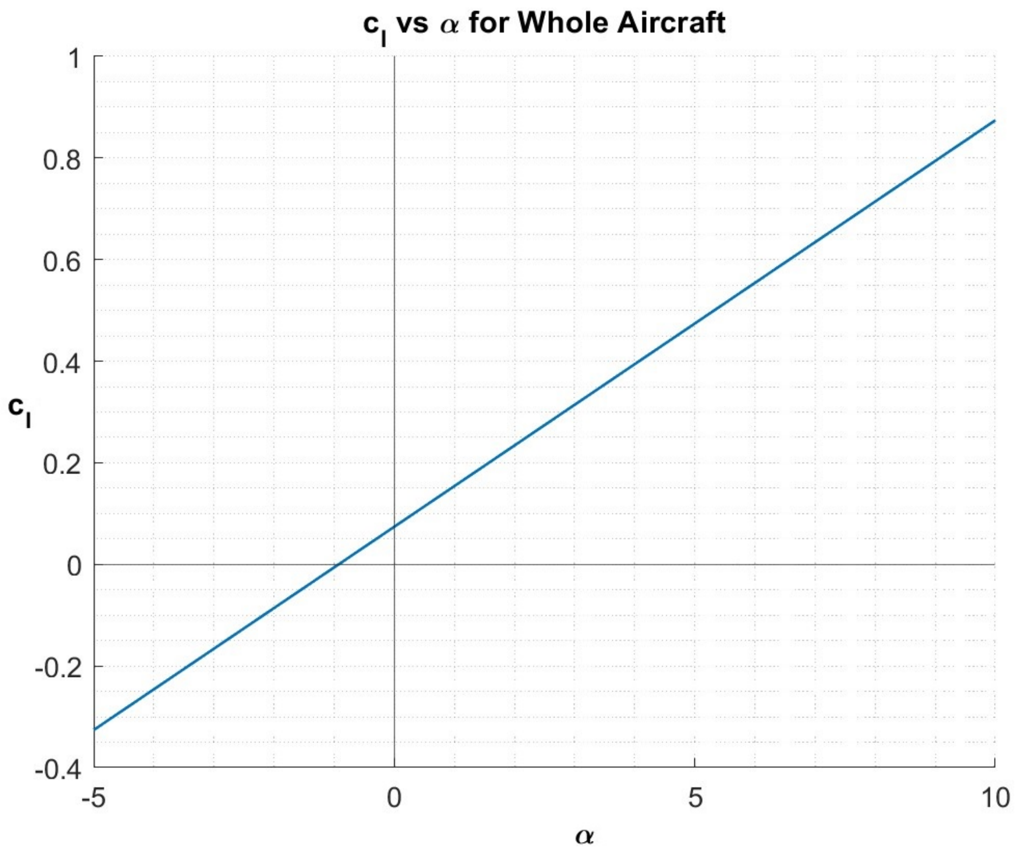

Figure 1: Coefficient of Drag versus Alpha for the Whole Aircraft

This plot shows the linear relationship between angle of attack and coefficient of lift. For the finite wing case, the lift slope is a little lower, at 0.08 [1/°]. This is because of the inefficiencies of the finite wing compared to the 2D airfoil. This plot was generated using the PLLT code generated in the last portion of lab and using the $\alpha_{L=0}$ values from the vortex panel code from part 1 of the lab. This model does not calculate stall, so the graph goes on linearly indefinitely, though this does not match reality.

2 Part 3 Task 2: Estimation of Profile Drag Coefficient

To simplify the modeling of the profile drag coefficient for the Cessna 180 aircraft, you can assume it to be equal to the sectional drag coefficient for the tip airfoil. With this in mind, construct a model of the sectional drag coefficient for the NACA 0012 airfoil (the tip airfoil) as a function of angle of attack leveraging the experimental data in Theory of Wing Sections. Generate a plot of the sectional drag coefficient predicted by your model versus angle of attack and also compare with experimental data with this plot. Provide a discussion on how you constructed your model with your submission, including any simplifying assumptions employed.

$$C_{D,0} = \int_{-b/2}^{b/2} c_d(y)c(y)dy$$

In order to calculate the Profile drag, we need the sectional drag for the airfoils. We used the NACA charts to pull three data points in the drag polars to fit a curve. We used a quadratic least squares model to simplify the analysis. This creates some error, since the experimental data is not quadratic, but it is close enough that we feel relatively confident in our results.

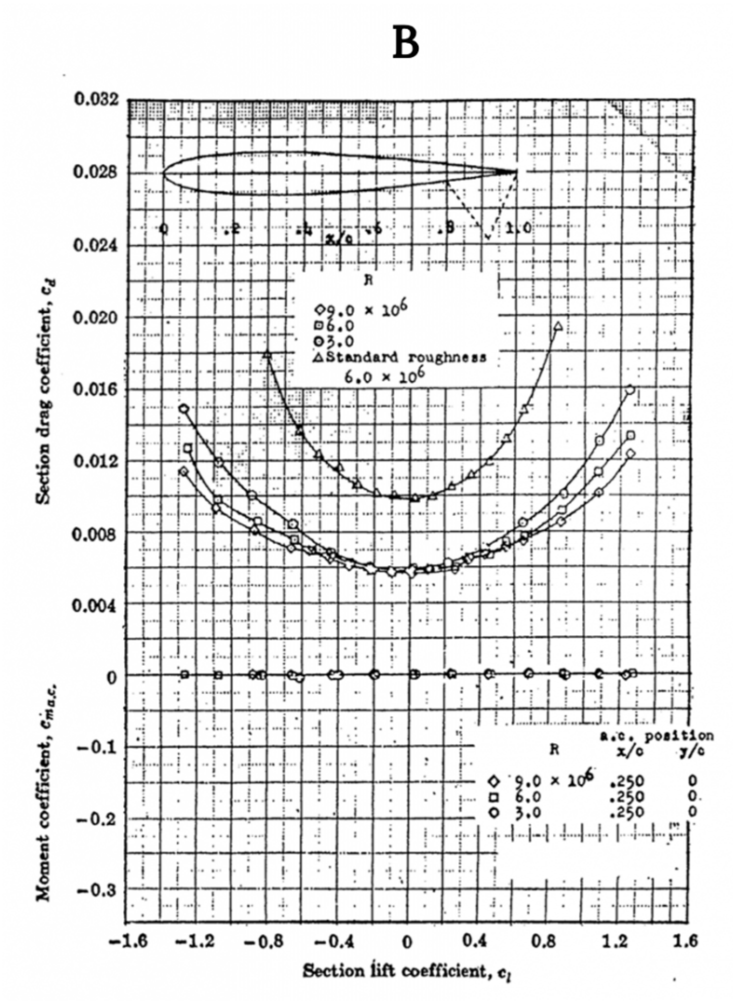

Figure 2: NACA 0012 Sectional Drag vs Coefficient of Lift

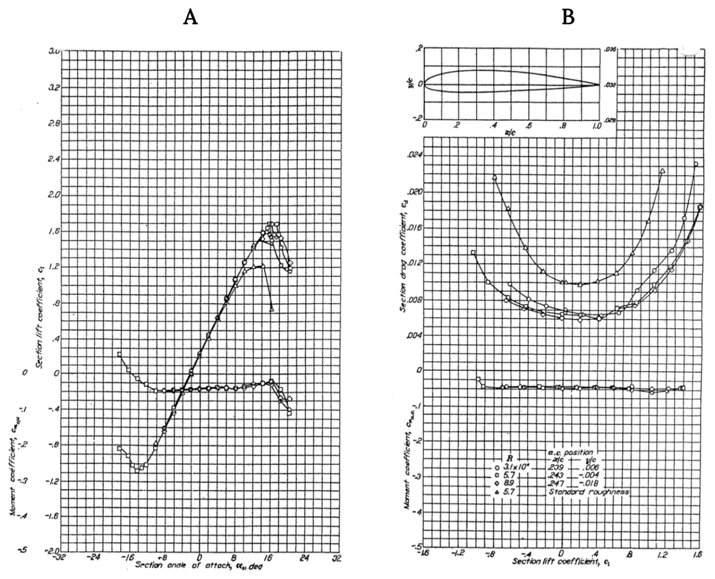

Figure 3: NACA 2412 Lift and Drag charts

From figures 2 and 3, we can get three points to form our quadratic fit. We took the left and right sides of the drag curve, as well as the minimum.

0012 c_L	0012 c_d	2412 c_L	2412 c_d
-0.8	0.018	-0.8	0.0218
0	0.01	0.2	0.009
0.85	0.0195	1.1	0.0225

Using these regressions, we can interpolate to get smooth curves that match the experimental data fairly well. With the results of interpolated data and the functions, in a for loop we calculated the sectional drag at the tip and root with varying alpha and used a function we wrote to calculate the integral for the profile drag. The inputs to the function are the dimensional quantities (c_t, c_r, b, S), the coefficient of drag at the tip and root $(c_{d,t}, c_{d,r})$ and a number n. This function takes the wingspan b and divides it into n linearly spaced points. At each point the integrand is calculated $(c_d(y_i)c(y_i))$ and then the total profile drag is calculated using the `trapz` function to integrate. This is done for each α_j and and the results are plotted, yielding figure 4. The code is in the appendix for reference.

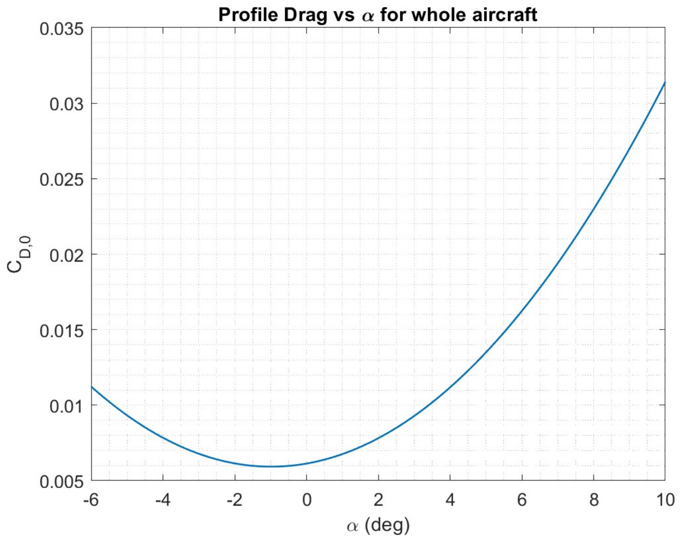

Figure 4: Profile drag for Cessna 180, assuming tip c_d is constant

This figure makes sense and matches what we expect for the behavior of the profile drag for the Cessna. Since we are assuming the NACA 0012 is constant across the wingspan in our calculation of c_d, it makes sense that the minimum profile drag is at $-1°$, where lift is minimized. This also matches the drag graph in figure 2. It makes sense that the drag increases on either side with angle of attack, because this matches the NACA airfoil charts, and it makes sense since as angle of attack increases, the pressure drag increases as well, which is a component of profile drag. There is a shift to the left by about $1°$ due to the geometric twist in the airfoil.

BONUS: Instead of assuming the profile drag coefficient to be the sectional drag coefficient for the tip airfoil, attain a model of the profile drag coefficient which takes into account the spanwise variation of the sectional drag coefficient

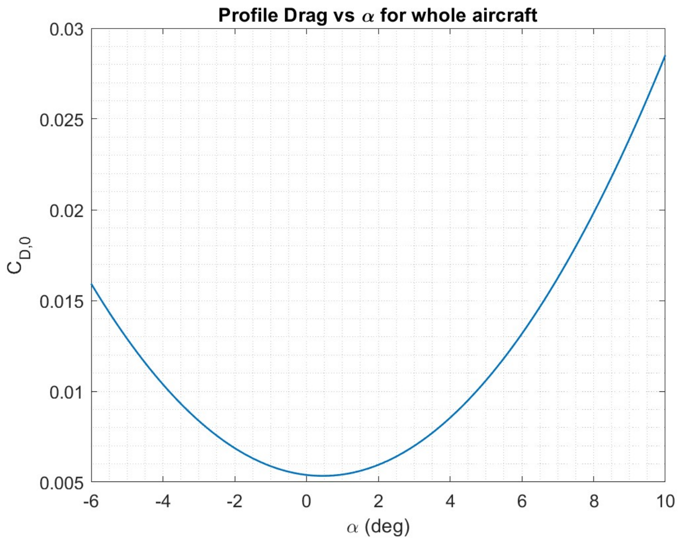

Figure 5: Extra Credit- Varying c_d along wingspan

In order to make this change, we assumed that the c_d value at any point y_i in the wingspan was varying linearly. This allowed us to plug in the different airfoils at the tip and root, and as in the PLLT code, vary them. This is a strong assumption to make, but this makes the programming much easier without any direct insight on how the sectional drag varies with position along the wing.

This graph is more unexpected than Figure 4. We see a minimum at $\alpha \approx 1.5°$, which doesn't really make sense since the profile drag should be minimized at a negative angle of attack when a symmetric airfoil is included. However, with closer examination of 3, we see that drag is not minimized at $c_l = 0$, but at $c_l \approx 0.2$. This slightly different data skews our results, but we only have that data to use so we trusted it.

3 Part 3 Task 3: Effect of Angle of Attack on Coefficient of Drag

Using your NACA airfoil generator, the provided vortex panel code, your PLLT code, and the profile drag coefficient model you constructed in Part 3 Task 2, plot the total drag coefficient, induced drag coefficient, and profile drag coefficient versus angle of attack for the Cessna 180 aircraft on the same figure.

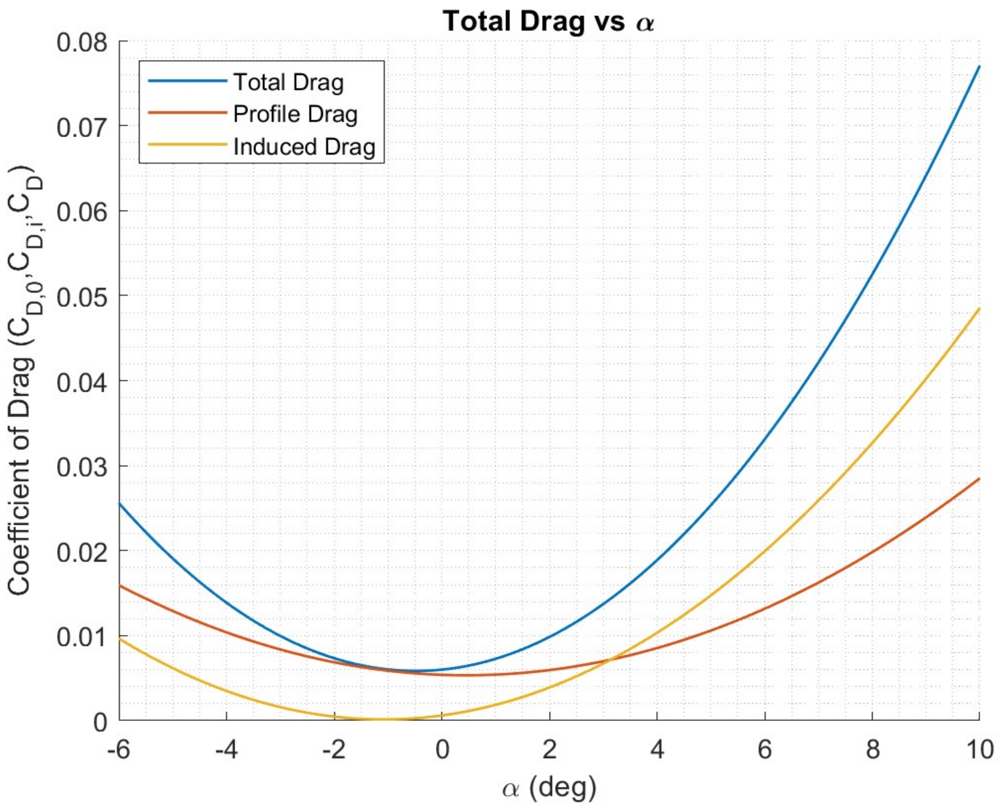

Figure 6: Drag Components versus Angle of Attack

This figure shows the dependence of drag and the primary contributor of drag as a function of angle of attack. This is made on the assumption that the profile drag is entirely pressure drag, neglecting skin fraction drag and other smaller contributors to drag. As the angle of attack increases, the induced drag increases and dominates the profile drag, which makes sense because induced drag is stronger with high alpha than profile drag. When there is no lift, at $\alpha_{L=0}$, the drag is entirely profile drag.

4 Part 3 Task 4: Effect of Airspeed on Thrust Required for Steady, Level Flight

Plot the thrust (in pounds of thrust) required for steady, level flight versus airspeed (in knots) ignoring the contributions of drag due to the fuselage and other non-wing parts of the Cessna 180 aircraft. In your calculations, assume the aircraft weights 2,600 lb and that the aircraft is flying at 10,000 feet on a standard day. What is the minimum thrust required for steady, level flight, and at what airspeed does this occur? Does this make sense to you? How would you expect these values to change if you were

to also account for the contributions of drag due to the non-wing parts of the aircraft? Hint: You may find it useful to first determine the thrust and airspeed required for steady, level flight as a function of angle of attack.
For the Steady, Level Flight, we can assume that $L = W$ and $T = D$. This means the aircraft is just flying steadily with no accelerations in any direction. This allows us to do the following simplifications, to solve for Thrust in terms of airspeed.

$$L = \frac{1}{2} C_L \rho_\infty V_\infty^2 S$$

$$V_\infty = \sqrt{\frac{2L}{C_L \rho_\infty S}}$$

This lets us find the velocities that we can fly at knowing our lift values. This will return complex values if $C_L < 0$, but this makes sense because an aircraft cannot be flying in steady, level conditions with negative lift. Now, using these velocities to solve for drag, we get:

$$D = C_D \frac{1}{2} \rho_\infty V_\infty^2 S$$

$$T = D$$

$$\boxed{T = C_D \frac{1}{2} \rho_\infty V_\infty^2 S}$$

Graphing this, we get:

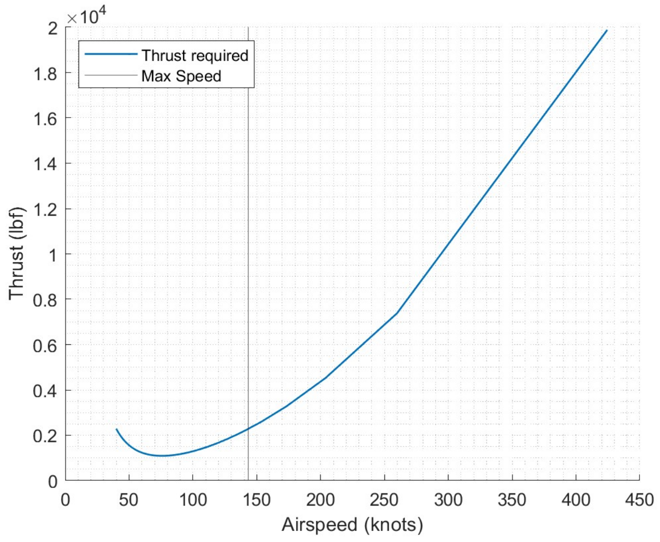

Figure 7: Theoretical Thrust versus Velocity

This graph is relatively accurate, but it does not take into account the limitations of the Cessna. This aircraft cannot generate 20,000 lbf of thrust, nor can it fly at 450 knots. Using the limitations of the aircraft we can truncate this graph, creating figure 8.

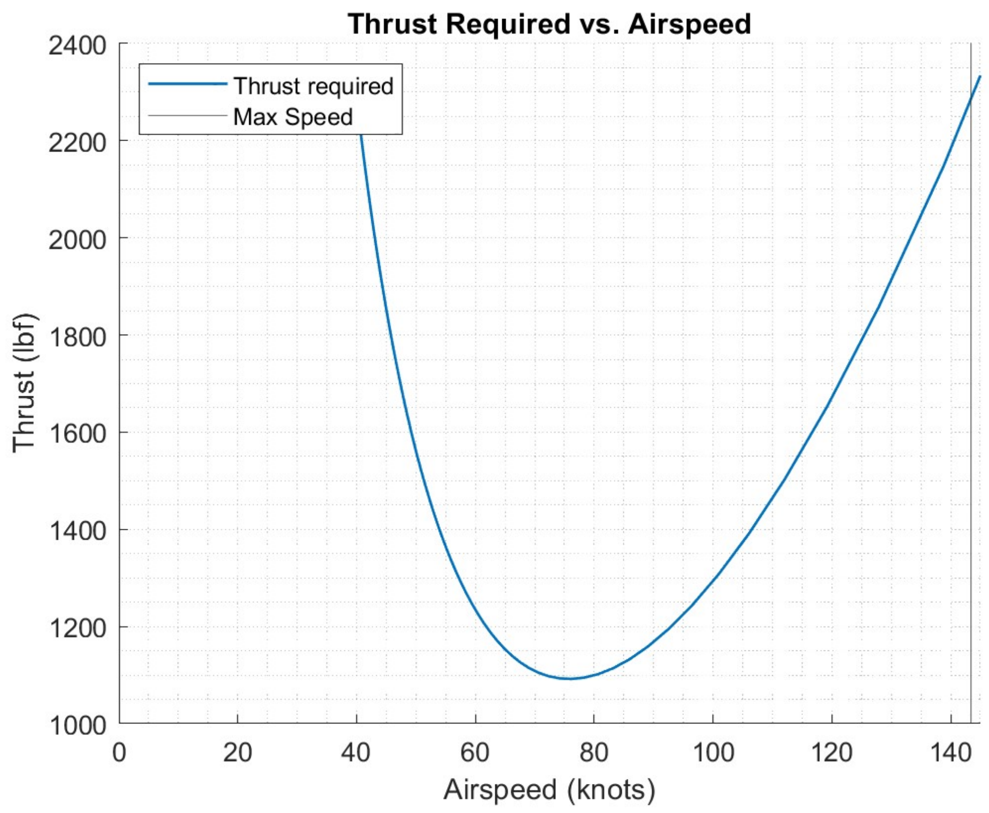

Figure 8: Thrust Required vs Airspeed for a Cessna 180

This graph correctly estimates the thrust required versus airspeed. As the airspeed increases, profile drag increases so more thrust is required to maintain steady flight. As the velocity drag decreases, more thrust is required to maintain steady flight since angle of attack is increasing and induced drag is increasing. On either end, there are limits where the airfoil reaches stall or where the engine meets its maximum output and thrust is overcome by drag.
Using the results, we can find the airspeed for minimum drag. This occurs at $V_\infty = 76.2$ knots, with a corresponding thrust of $T_{min} = 1092.1$ lbf. These results make sense. They are physically a minimum on the graph, but they are also within the bounds of what is reasonable for an aircraft like the Cessna to produce and experience. If the other components of the body were included in the drag calculation, the required thrust would increase since drag is increasing. If the thrust increases, the velocity should have to increase as well.

5 Appendix

5.1 Task 1

```
    close all; clear; clc;

%% Givens
b = 35 + 10/12; %ft
c_r = 5 + 4/12; %ft
c_t = 7 + 7/12; %ft
a0_t = 0.1185 * 180/pi;
a0_r = 0.1186 * 180/pi; % FROM PART 1 OF LAB
aero_t = 0;
aero_r = 0;
```

```
nums_root = [2,4,12];
nums_tip = [0,0,12];
geo_root = 0;
geo_tip = geo_root + 2*pi/180; %rad
N = 20;

%% PART 1: Coefficient of lift
airfoil_root = naca4series(nums_root,c_r,20);
airfoil_tip = naca4series(nums_tip,c_t,20);

alpha = linspace(-5,10,100);

for i = 1:length(alpha)
    cl_tip(i) = Vortex_Panel(airfoil_tip.xb,airfoil_tip.yb,alpha(i)+2);
    cl_root(i) = Vortex_Panel(airfoil_root.xb,airfoil_root.yb,alpha(i));
    [~,c_L(i),c_Di(i)] = PLLT(b,a0_t,a0_r,c_t,c_r,aero_t,aero_r,(alpha(i) + 2)*pi/180,alpha(i)*pi
        /180,N);
end

%Finding a0
lift_slope = (c_L(3)-c_L(2))/(alpha(3)-alpha(2))

% Plotting results
figure()
hold on;
grid minor
plot(alpha,c_L,'Linewidth',1)
xline(0)
yline(0)
title("c_l vs \alpha for Whole Aircraft")

figure()
hold on;
grid minor
plot(alpha,c_Di,"Linewidth",1)
xlabel("\alpha (deg)")
ylabel("c_d")
title("Induced Drag versus Alpha (From PLLT)")
```

5.2 Tasks 2 and 3

```
    close all; clear; clc;

%%% PART 3 TASK 2 %%%%%%%%%%%%%%%%%%

cr = 5 + 4/12; %ft
ct = 7 + 7/12; %ft
b = 35 + 10/12; %ft
S = 2*(0.5*(ct + cr)*b/2);

alpha = linspace(-6,10,100);


b = 35 + 10/12; %ft
c_r = 5 + 4/12; %ft
c_t = 7 + 7/12; %ft
a0_t = 0.1185 * 180/pi;
a0_r = 0.1186 * 180/pi;
```

```
aero_t = 0;
aero_r = 0;
nums_root = [2,4,12];
nums_tip = [0,0,12];
geo_root = 0;
geo_tip = geo_root + 2*pi/180; %rad
N = 20;
n = 10;

% Fitting to be able to model the cd by alpha
cl = [-0.8,0,.85];
cd = [0.018,0.01,0.0195];

% vector of C_L values to use, and associated fit output
cL_eval_tip = linspace(-0.8,0.85,50);
[P,~] = polyfit(cl,cd,2);
cd_eval_tip = polyval(P,cL_eval_tip);

clr = [-0.8,0.2,1.1];
cdr = [0.0218,.009,0.0225];

% vector of C_L values to use, and associated fit output
cL_eval_root = linspace(-0.8,1.1,50);
[P,~] = polyfit(clr,cdr,2);
cd_eval_root = polyval(P,cL_eval_root);


% NO VARIATION IN cd(y) (0012 the whole way)
for i = 1:length(alpha)
    % Finding cd at tip and root based on alpha
    cdt = spline(cL_eval_tip,cd_eval_tip,(alpha(i)-2)/(2*pi));
    cdr = spline(cL_eval_tip,cd_eval_tip,alpha(i)/(2*pi));
    Cd0(i) = profile_drag_coeff(ct,cr,cdt,cdr,b,S,n);
    [~,c_L(i),c_Di(i)] = PLLT(b,a0_t,a0_r,c_t,c_r,aero_t,aero_r,(alpha(i) + 2)*pi/180,alpha(i)*pi
        /180,N);
end

figure()
plot(alpha,Cd0,"Linewidth",1)
grid minor
xlabel("\alpha (deg)")
ylabel("C_{D,0}")
title("Profile Drag vs \alpha for whole aircraft")

% NO VARIATION IN cd(y) (0012 the whole way)
for i = 1:length(alpha)
    cdt = spline(cL_eval_tip,cd_eval_tip,(alpha(i)-2)/(2*pi));
    cdr = spline(cL_eval_root,cd_eval_root,alpha(i)/(2*pi));
    Cd0(i) = profile_drag_coeff(ct,cr,cdt,cdr,b,S,n);
    [~,c_L(i),c_Di(i)] = PLLT(b,a0_t,a0_r,c_t,c_r,aero_t,aero_r,(alpha(i) + 2)*pi/180,alpha(i)*pi
        /180,N);
end

figure()
plot(alpha,Cd0,"Linewidth",1)
grid minor
xlabel("\alpha (deg)")
ylabel("C_{D,0}")
```

```
title("Profile Drag vs \alpha for whole aircraft")


%% CALCULATING TOTAL DRAG

% Finding drag
CD = Cd0 + c_Di;

% Plotting
figure()
hold on
plot(alpha,CD,"Linewidth",1)
plot(alpha,Cd0,"Linewidth",1)
plot(alpha,c_Di,"Linewidth",1)
grid minor
xlabel("\alpha (deg)")
ylabel("Coefficient of Drag (C_{D,0},C_{D,i},C_D)")
title("Total Drag vs \alpha")
legend("Total Drag","Profile Drag","Induced Drag","location","northwest")

close all
%% CALCULATING THRUST

% Finding density
h = 10000; %ft
h = h/3.281; %m
[~,~,~,rho] = atmosisa(h);
rho = rho/515.4*32.17; %slug/ft^3
W = 26000; %lbs

% Finding Velocity
V = sqrt(2*W./(c_L*rho*S));
V = V(V==real(V));

% Finding Drag
CD = CD(33:end);
T = CD.*(0.5*rho*V.^2*S);

% Calculating minimum thrust and associated V
[Tmin, Imin] = min(T)
Vmin = V(Imin)/1.688

%plotting results
figure()
hold on
plot(V/1.688,T,"Linewidth",1)
grid minor
xlabel("Airspeed (knots)")
ylabel("Thrust (lbf)")
xline(143.4)
legend("Thrust required","Max Speed","Location","Northwest")

figure()
hold on
plot(V/1.688,T,"Linewidth",1)
grid minor
xlabel("Airspeed (knots)")
ylabel("Thrust (lbf)")
```

```
xline(143.4)
xlim([0,145])
legend("Thrust required","Max Speed","Location","Northwest")
title("Thrust Required vs. Airspeed")
```

5.3 Functions

5.3.1 Profile Drag Coefficient Function

```
    function Cd0 = profile_drag_coeff(ct,cr,cdt,cdr,b,S,n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This function calculates the profile drag coefficient for an airfoil
% given the profile drag and the root and tip, assuming trapezoidal wings.
% It also assumes that cd varies linearly across the wingspan between cdt
% and cdr

% INPUTS:
% ct - chord length of the tip
% cr - chord length at the root
% cdt - coefficent of profile drag at the tip
% crt - coefficient of profile drag at the root
% b - wingspan
% S - planform area
% n - number of integration points

% OUTPUTS:
% Cd0 - profile drag coefficient

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Function handles for c(y) and cd(y)
c = @(y) (ct-cr)/(-b/2)*abs(y) + cr;
cd =  @(y) (cdt-cdr)/(-b/2)*abs(y) + cdr;

% Points to evaluate at
y = linspace(-b/2,b/2,n);

% Integrand for trapz to work with
integrand = cd(y).*c(y);

% Profile drag coefficient
Cd0 = trapz(y,integrand)/S;
```

5.3.2 PLLT Function

```
    function [e,c_L,c_Di] = PLLT(b,a0_t,a0_r,c_t,c_r,aero_t,aero_r,geo_t,geo_r,N)

% INPUTS:
% b - span
% a0_t - lift slope at the wing tip
% a0_r - lift slope at the wing root
% c_t - chord length at the tip
% c_r - chord length at root
% aero_t - induced AoA at wing tip
% aero_r - induced AoA at wing root
% geo_t - geometric AoA at wing tip
% geo_r - geometric AoA at wing root
% N - number of Fourier Coefficients (terms)


% OUTPUTS:
% e - span efficiency factor
% c_L - coefficient of lift
% c_Di - induced drag coefficient


%% Calculating the geometric parameters as functions to use later
S = 2*(0.5*(c_t + c_r)*b/2);
AR = b^2/S;
c = @(y) (c_t-c_r)/(-b/2)*y + c_r; % Assuming linear change
a0 = @(y) (a0_t-a0_r)/(-b/2)*y + a0_r;
aero = @(y) (aero_t-aero_r)/(-b/2)*y + aero_r;
geo = @(y) (geo_t-geo_r)/(-b/2)*y + geo_r;

% Control points
theta0 = zeros(N,1);
for i = 1:N
    theta0(i) = i*pi/(2*N);
end

y0 = -b/2*cos(theta0);


%% Solving system of equations to find the Fourier Coefficients
M = zeros(N,N); %Matrix for linear system of equations

% Assigning the entries of B
for i = 1:N
    for j = 1:N
        M(i,j) = 4*b/(a0(y0(i))*c(y0(i)))*sin((2*j-1)*theta0(i)) + (2*j-1)*sin((2*j-1)*theta0(i))/
            sin(theta0(i));
    end
end

% Assigning the entries of b
rhs = zeros(N,1);
for i = 1:N
    rhs(i) = -aero(y0(i)) + geo(y0(i));
end

% Solving for the Fourier Coefficients
%A = M\rhs;
A = inv(M)*rhs;
% A1 should be 0.01261

%% Solving for the desired quantities
c_L = A(1)*pi*AR;
delta = 0;

for i = 2:N
    delta = (2*i-1)*(A(i)/A(1))^2 + delta;
end
e = 1/(1+delta);
```

```
c_Di = c_L^2/(pi*e*AR);

end
```

5.3.3 Vortex Panel Code

```
    function [CL] = Vortex_Panel(XB,YB,ALPHA)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input: %
% %
% XB = Boundary Points x-location %
% YB = Boundary Points y-location %
% ALPHA = AOA in degrees %
% %
% Output: %
% %
% CL = Sectional Lift Coefficient %
% improves efficiency by preallocating matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
% Convert to Radians %
%%%%%%%%%%%%%%%%%%%%%%
ALPHA = ALPHA*pi/180;
%%%%%%%%%%%%%%%%%%%%%
% Compute the Chord %
%%%%%%%%%%%%%%%%%%%%%
CHORD = max(XB)-min(XB);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Determine the Number of Panels %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = max(size(XB,1),size(XB,2))-1;
MP1 = M+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Preallocate Matrices for Efficiency %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X = zeros(1,M);
Y = zeros(1,M);
S = zeros(1,M);
THETA = zeros(1,M);
SINE = zeros(1,M);
COSINE = zeros(1,M);
RHS = zeros(1,M);
CN1 = zeros(M);
CN2 = zeros(M);
CT1 = zeros(M);
CT2 = zeros(M);
AN = zeros(M);
AT = zeros(M);
V = zeros(1,M);
CP = zeros(1,M);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Intra-Panel Relationships: %
% %
% Determine the Control Points, Panel Sizes, and Panel Angles %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
IP1 = I+1;
```

```
X(I) = 0.5*(XB(I)+XB(IP1));
Y(I) = 0.5*(YB(I)+YB(IP1));
S(I) = sqrt( (XB(IP1)-XB(I))^2 +( YB(IP1)-YB(I))^2 );
THETA(I) = atan2( YB(IP1)-YB(I), XB(IP1)-XB(I) );
SINE(I) = sin( THETA(I) );
COSINE(I) = cos( THETA(I) );
RHS(I) = sin( THETA(I)-ALPHA );
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inter-Panel Relationships: %
% %
% Determine the Integrals between Panels %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
for J = 1:M
if I == J
CN1(I,J) = -1.0;
CN2(I,J) = 1.0;
CT1(I,J) = 0.5*pi;
CT2(I,J) = 0.5*pi;
else
A = -(X(I)-XB(J))*COSINE(J) - (Y(I)-YB(J))*SINE(J);
B = (X(I)-XB(J))^2 + (Y(I)-YB(J))^2;
C = sin( THETA(I)-THETA(J) );
D = cos( THETA(I)-THETA(J) );
E = (X(I)-XB(J))*SINE(J) - (Y(I)-YB(J))*COSINE(J);
F = log( 1.0 + S(J)*(S(J)+2*A)/B );
G = atan2( E*S(J), B+A*S(J) );
P = (X(I)-XB(J)) * sin( THETA(I) - 2*THETA(J) ) ...
+ (Y(I)-YB(J)) * cos( THETA(I) - 2*THETA(J) );
Q = (X(I)-XB(J)) * cos( THETA(I) - 2*THETA(J) ) ...
- (Y(I)-YB(J)) * sin( THETA(I) - 2*THETA(J) );
CN2(I,J) = D + 0.5*Q*F/S(J) - (A*C+D*E)*G/S(J);
CN1(I,J) = 0.5*D*F + C*G - CN2(I,J);
CT2(I,J) = C + 0.5*P*F/S(J) + (A*D-C*E)*G/S(J);
CT1(I,J) = 0.5*C*F - D*G - CT2(I,J);
end
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Inter-Panel Relationships: %
% %
% Determine the Influence Coefficients %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
AN(I,1) = CN1(I,1);
AN(I,MP1) = CN2(I,M);
AT(I,1) = CT1(I,1);
AT(I,MP1) = CT2(I,M);
for J = 2:M
AN(I,J) = CN1(I,J) + CN2(I,J-1);
AT(I,J) = CT1(I,J) + CT2(I,J-1);
end
end
AN(MP1,1) = 1.0;
AN(MP1,MP1) = 1.0;
for J = 2:M
AN(MP1,J) = 0.0;
```

```
end
RHS(MP1) = 0.0;
%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for the gammas %
%%%%%%%%%%%%%%%%%%%%%%%%
GAMA = AN\RHS';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Tangential Veloity and Coefficient of Pressure %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for I = 1:M
V(I) = cos( THETA(I)-ALPHA );
for J = 1:MP1
V(I) = V(I) + AT(I,J)*GAMA(J);
end
CP(I) = 1.0 - V(I)^2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Solve for Sectional Coefficient of Lift %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
CIRCULATION = sum(S.*V);
CL = 2*CIRCULATION/CHORD;
end
```