AIRPLANE ANALYSIS TERM PROJECT

Canadair / Firefighter

CL-415 / DHC-515

Kim Tran

MEEN 4120

Aerospace Fundamentals

Spring 2023

Table of Contents

List of Figures

List of Tables

Background

The DHC-515 Firefighter, also known as the Canadair CL-415 and Bombardier 415, is a Canadian amphibious waterbomber aircraft designed and built by Canadair, which by the time of production became a division of Bombardier Aerospace. The aircraft production program was then subsequently built by Viking Air and De Hallivand Canada, where it received its current name from the CL-515 (an updated underproduction version of CL-415) The aircraft is primarily used as a firefighter and search and rescue plane, with the ability to land and take off from both land and water.

The CL-415/DHC-515 was first produced in 1993 as a successor to the CL-215, with a total of 90 aircraft built until production ended in 2015. The aircraft is powered by two Pratt & Whitney Canada PW123AF turboprop engines, holding a takeoff power of 2,380 shp each, giving it a maximum speed of 223 mph (359 km/h) and a range of over 1,500 miles (2,414 km).

One of the most notable features of the DHC-515 is its firefighting capability. The aircraft is equipped with a large tank, which can hold up to 1,620 gallons (6,132 liters) of water or 13,536lbs (6,140 kg) of weight. It can scoop water from a nearby water source while flying at low altitudes, then drop the water onto a fire to help extinguish it. This makes it a vital tool in fighting wildfires, particularly in areas where access to water is limited.

The DHC-515 has been used in firefighting and search and rescue operations around the world, including in Canada, the United States, Greece, Italy, Spain, and Australia. It has been praised for its effectiveness in fighting wildfires and saving lives in emergency situations.

Despite its success, the DHC-515 faced some challenges during its development and fielded use. One of the major challenges was the high cost of maintenance due to the harsh environment it operates in. Another challenge was the need for highly skilled pilots to operate the aircraft due to its unique amphibious capabilities. Of the 90 aircraft built, seven had been reportedly removed from service as a result of several accidents by December 2007.

In terms of dimensions, the DHC-515 has a wingspan of 93 feet 11 inches (28.6 meters), a length of 65 feet (19.82 meters), and a height of 29 feet 3 inches (8.9 meters). Its empty weight is approximately 28,400 pounds (102,880 kg), its maximum takeoff weight from land is 43,850 pounds (19,890 kg), and its maximum takeout weight from water is 37,850 pounds (17,170 kg).

In summary, the De Hallivand Canada DHC-515 / Canadair CL-415 is a critically important aircraft for firefighting and search and rescue operations around the world, with its unique amphibious capabilities and firefighting tank system. Despite some challenges during its development and fielded use, the DHC-515 has proven to be an effective and reliable tool for saving lives and protecting communities from wildfires.

Aircraft Data

Data Type	Value	Unit	Reference
Gross weight	43850	lbs	https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Empty weight	28400	lbs	https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Wing area	1080	ft	https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Wing span	93.92	ft	https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Oswald Spanwise Efficiency	0.83	n/d	https://calculator.academy/oswald-efficiency-factor-calculator/#f1p1|f2p0
Zero lift drag coefficient	0.012	n/d	AbbottDoenhoff_TheoryOfWingSections
Sweep	0	deg	https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Propulsive efficiency	0.52	n/d	https://ppg.e-props.fr/calculator_PROPS.php?language=en
Number of engines	2		https://winair.ca/wp-content/uploads/2018/05/Everything-that-You-Need-to-Know-about-the-Canadair-CL-415-Infographic-Image.png
Max Engine Power	2380	hp	https://en.wikipedia.org/wiki/Pratt_%26_Whitney_Canada_PW100
Max Engine Thrust	0	lbs	https://en.wikipedia.org/wiki/Pratt_%26_Whitney_Canada_PW100
Specific Fuel Consumption	0.47	lb/hp*hr	https://en.wikipedia.org/wiki/Pratt_%26_Whitney_Canada_PW100

Airfoil	NACA4418		Orrico_SJoseph_Technical_Report; http://airfoiltools.com/airfoil/details?airfoil=naca4418-il; 3rd-place-undergraduate-team-multi-mission-amphibian

Table 1: Aircraft Data

Level Flight Speed Sweeps for various standard day altitudes

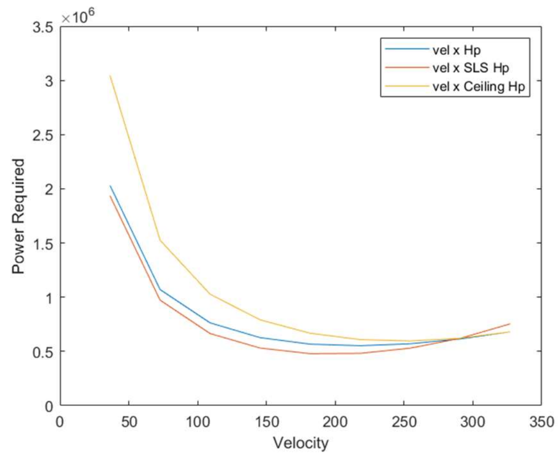


Figure 1: Power Required vs. Various Air Speeds

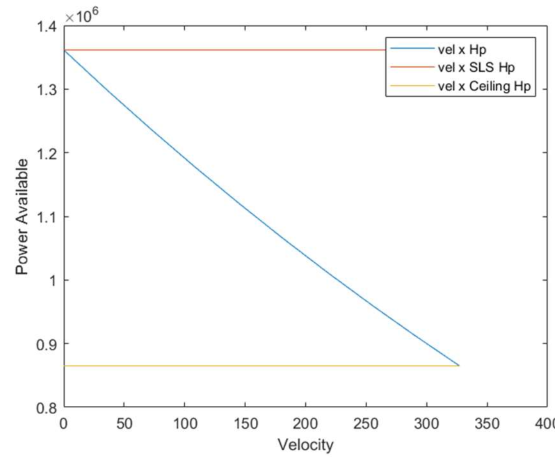


Figure 2: Power Available vs. Various Air Speeds

In Figure 2, it can be observed that the power required decreases dramatically at first, but then starts to increase slowly as air speed increases. In Figure 3, the power available depends on the altitude of the aircraft. At 0ft, the power available is maximized but when the aircraft is at 14,700ft, the power available is very limited.

This trend can be explained by knowing that the power level required to maintain level light will initially increase with airspeed due to the increased lift to drag required. When the L/D ratio of the wings reach a certain airspeed, the induced drag decreases and thus lowers the power required to maintain level flight. When flying at higher speeds, the increase in parasitic drag causes the power required to increase again and to peak in the power curve.

The minimum airspeed will occur where the point on the power curve is at its lowest, while the maximum airspeed will occur when the point on the power curve is where induced drag is at its lowest because this is where the maximum power output is required to maintain level flight.

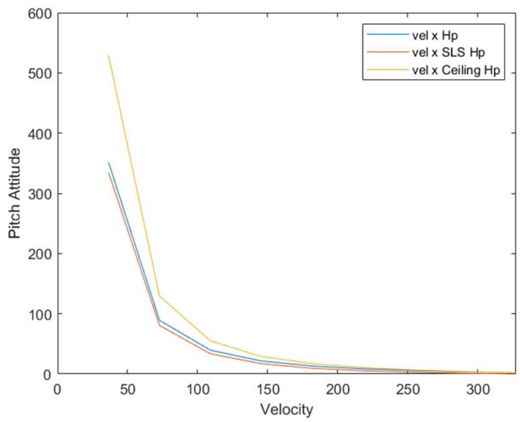


Figure 3: Pitch Attitude vs. Velocity

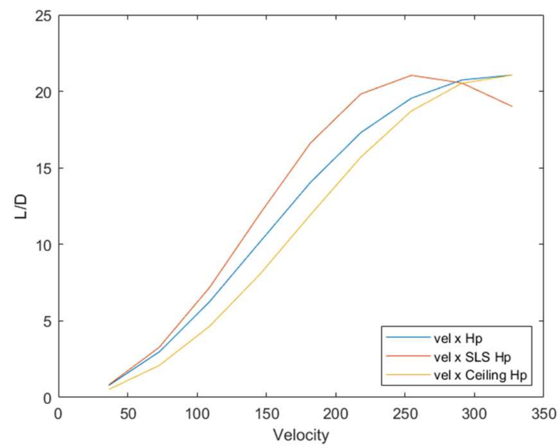


Figure 4: L/D vs. Velocity

The relationship between pitch attitude and velocity can be found in figure 4. The pitch attitude will decrease with increasing air speeds due to the lift generated by the wings, this can be seen in Figure 5. In order to maintain stability, the pitch attitude must decrease to preventing the aircraft from pitching up too much.

Aerodynamic Efficiency

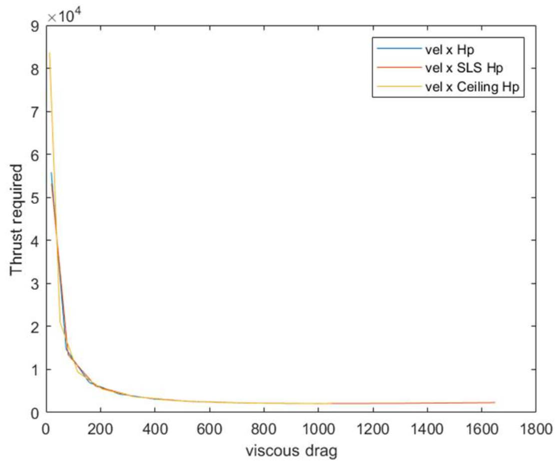


Figure 5: Thrust Required vs. Viscous Drag

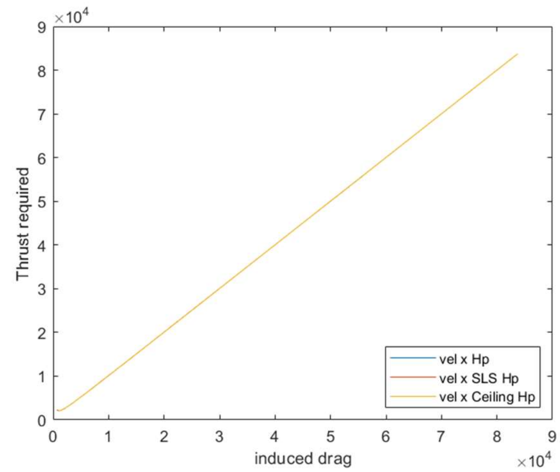


Figure 6: Thrust Required vs. Induced Drag

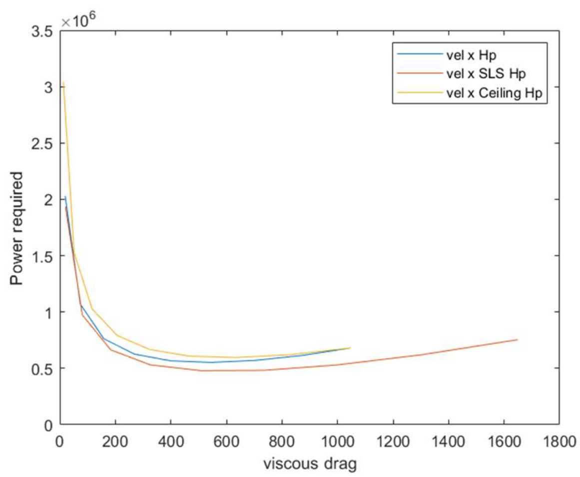


Figure 7: Power Required vs. Viscous Drag

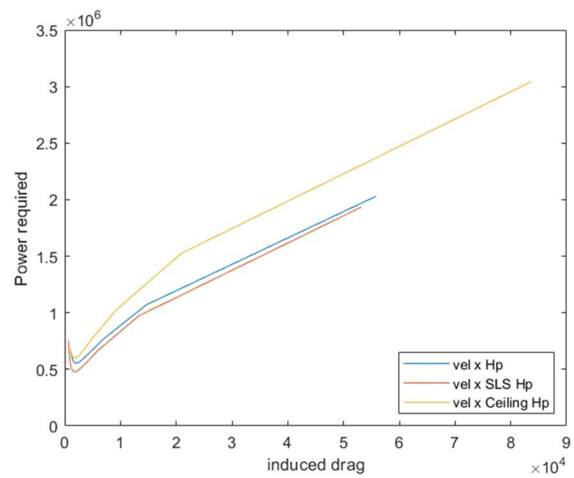


Figure 8: Power required vs. Induced Drag

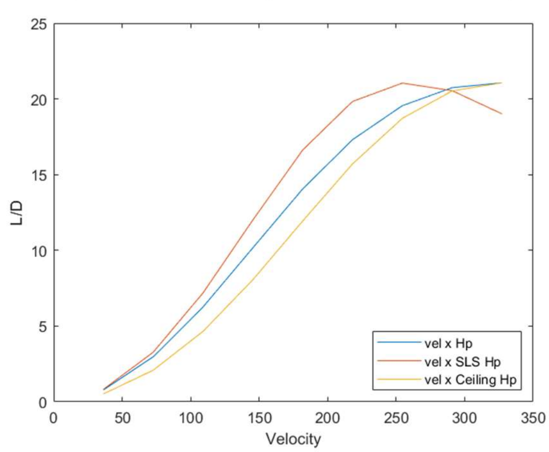


Figure 9: 3D Lift to drag ratio

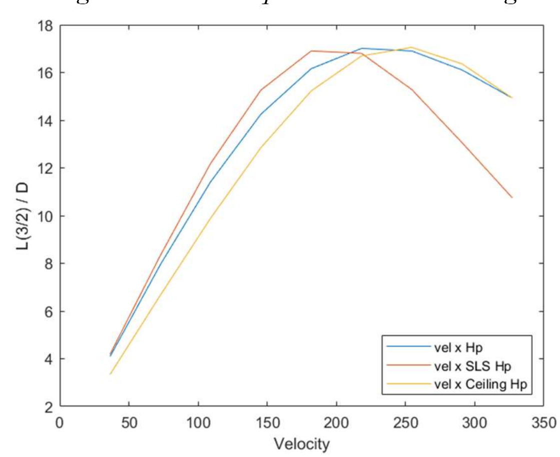


Figure 10: 3D Lift$^{3/2}$ to Drag Ratio

The relationship between the thrust required and power required to overcome viscous and induced drag to the 3D L/D ratio is that they are all related to the efficiency of an aircraft in flight. It can be observed that as airspeed increases, thrust and power both decrease rapidly in the beginning, but then gradually increase with airspeed. Thrust and power will generally increase linearly against induced drag. At an airspeed corresponding to the max 3D L/D ratio, the lift generated by the wings will be at its maximum, resulting in efficient flight conditions. The 3D $L^{3/2}/D$ ratio operates on the same principle but takes into account the weight of the aircraft. When the power and thrust is at its minimum, so is the 3D L/D ratio and 3D $L^{3/2}/D$ ratio.

Rate of Climb

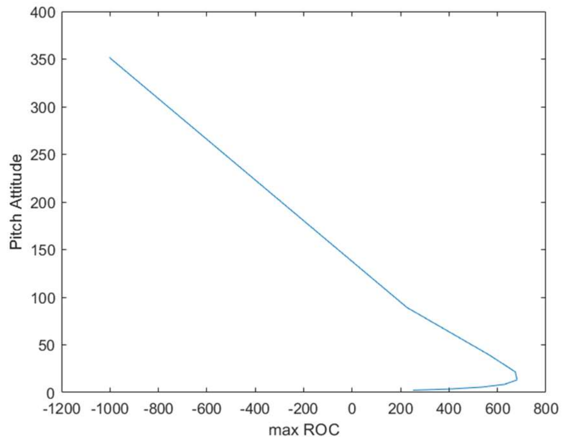


Figure 11: Pitch Attitude vs. Max ROC

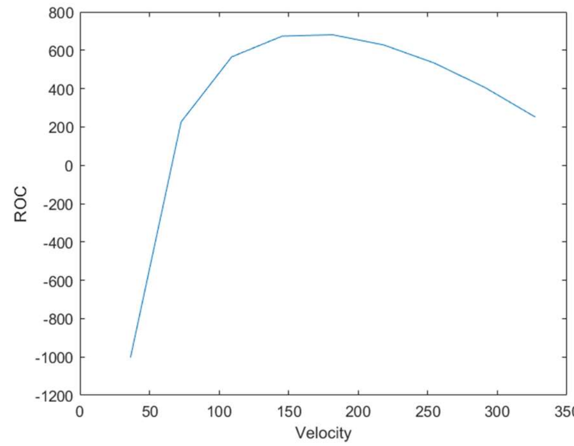


Figure 12: ROC vs. Velocity

It can be observed from Figure 11 that pitch attitude is decreasing and the ROC is increasing, this is due the weight of the airplane causing it to descent. When the aircraft is at an optimal pitch attitude (as seen at around 25 degrees), there will be increased drag therefore reducing the ROC. In Figure 12, the maximum rate of climb occurs when flying at an airspeed that produces the highest lift to drag ratio because the aircraft is generating the most amount of lift for the least amount of drag. The absolute ceiling is observed at around 350 degrees and the service ceiling at around 150 degrees.

Range

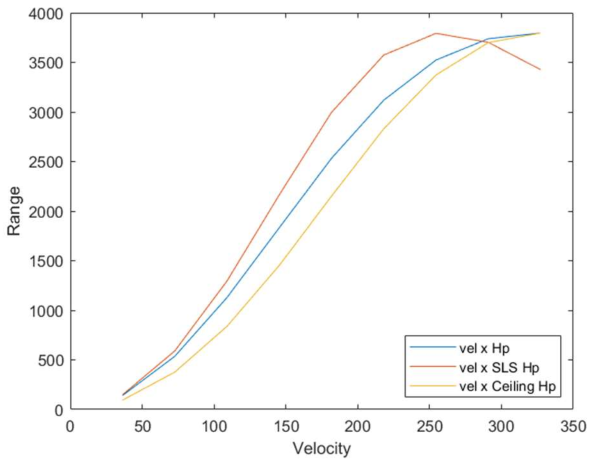
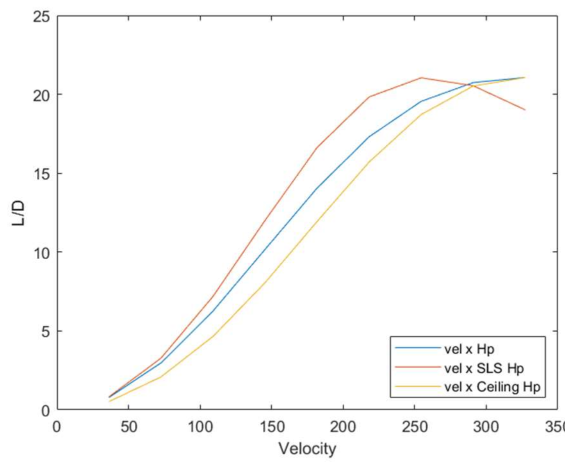


Figure 13: Range vs. Velocity



3D Lift to Drag Ratio

In order to fly at a maximum range, the aircraft generally needs to go at an airspeed that maximizes the lift to drag ratio, which determines how efficient the aircraft is in using fuel over distance traveled. By maximizing this ratio, we can get the highest range for a given amount of fuel by using the lowest possible specific fuel consumption. Some other factors that can increase the maximum range is utilizing the largest possible propulsive efficiency and having the largest fuel weight and lowest empty weight.

Endurance

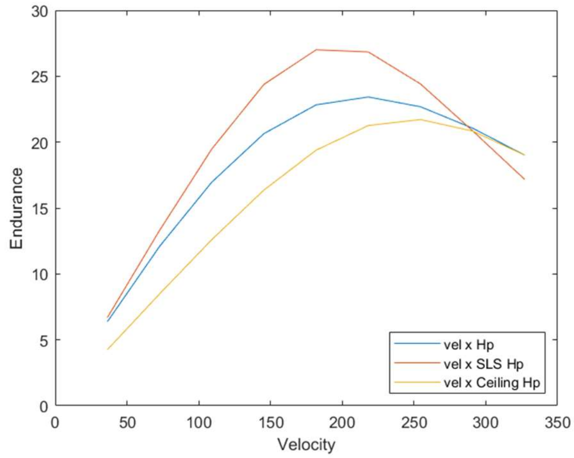
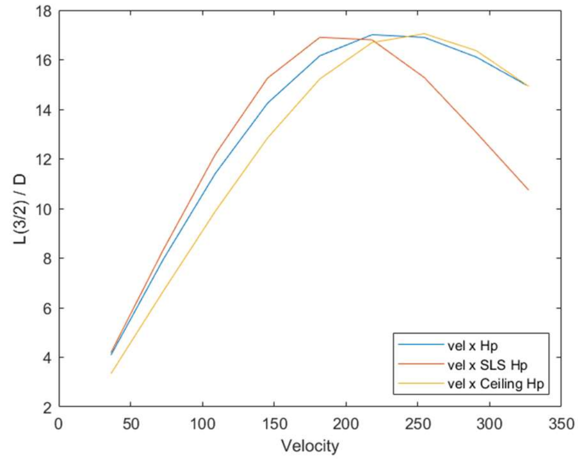


Figure 14: Endurance vs. Velocity



3D Lift$^{3/2}$ to Drag Ratio

In general, in order to achieve maximum endurance, the aircraft would need to travel at an airspeed that minimizes the fuel consumption rate which is the speed where the maximum 3D lift$^{3/2}$ to drag ratio occurs. It will need to travel slower than the airspeed for maximum range. This means that the plane will need a high ratio of largest fuel weight and lowest empty weight, as well as fly as low as possible to increase air density.

References

A., Abbott Ira H, and Von Doenhoff Albert E. Theory of wing sections: Including a summary of

"Canadair CL-415." Wikipedia, April 2, 2023. https://en.wikipedia.org/wiki/Canadair_CL-415.

Lednicer, David. "Airfoils of U S and Canadian Aircraft." Airfoils. Accessed May 11, 2023. http://www.aerofiles.com/airfoils.html.

"Pratt & Whitney Canada PW100." Wikipedia, April 27, 2023. https://en.wikipedia.org/wiki/Pratt_%26_Whitney_Canada_PW100.

"Pratt & Whitney Canada PW123 Turboprop Engine." Homepage. Accessed May 11, 2023. https://airandspace.si.edu/collection-objects/pratt-whitney-canada-pw123-turboprop-engine/nasm_A19900255000.

"Propeller Efficiency Calculator." E. Accessed May 11, 2023. https://ppg.e-props.fr/calculator_PROPS.php?language=en.

"Propeller Specification." Propeller Specification | Viking's Aerial Firefighter. Accessed May 11, 2023. https://aerialfirefighter.vikingair.com/firefighting/specifications/propeller-specification.

airfoil data. New York: Dover, 2015.

Serdar A., Mehmet S., Alperen S., Muzaffer T.k AIAA – 2017 Multi Mission Amphibian (MMA) Undergraduate Team Design Competition

Haley K., Brendan W., Jackson W. AIAA 2021-2022 Undergraduate Aerial Firefighter Aircraft Final Design Report

"Turboprop." Wikipedia, April 7, 2023. https://en.wikipedia.org/wiki/Turboprop.

WinAir. "Everything That You Need to Know about the Canadair CL-415 - Winair (Aviation Management Software)." WinAir. Accessed May 11, 2023. https://winair.ca/blog/everything-need-know-canadair-cl-415/.

Appendix A. MATLAB Summary

<table>
<tr><th>Function</th><th>Summary</th></tr>
<tr><td><pre>function [alo] = Angle0Lift(Data)
% angle for zero lift
alo = intrp(0,Data.cl,Data.alfa);

%cl0_position = find(Data.cl==0);
%alo = Data.alfa(cl0_position);

end</pre></td><td>Function to determine the zero lift angle</td></tr>
<tr><td><pre>function [aStall] = AngleStall(Data)
% return stall angle
clmax = -100;
for i = 1:length(Data.alfa)
    if Data.cl(i) > clmax
        aStall = Data.alfa(i);
        clmax = Data.cl(i);
    end
end
end</pre></td><td>Function to determine the stall angle</td></tr>
<tr><td><pre>function [rho] = getDensity(h)
% atmosphere model for temp, pressure, density
% inputs
%   h: pressure altitude (ft)
% output
%   rho: density (slug/ft^3)
R = 1716;   % gas constant
hf = convlength(h,'ft','m');
[T, a, P, rho] = atmosisa(hf);
T   = convtemp(T,'K','R');
%a   = convvel(a,'m/s','ft/s');
%rho = convdensity(rho,'kg/m^3','slug/ft^3');
P   = convpres(P,'Pa','psf');
rho = P/(R*T);

end</pre></td><td>Function to find the density from a given pressure altitude in ft</td></tr>
<tr><td><pre>function [T] = getTemperature(h)
% atmosphere model for temp, pressure, density
% inputs
%   h: pressure altitude (ft)
% output
%   T: temperature (deg R)
R = 1716;   % gas constant
hf = convlength(h,'ft','m');
[T, a, P, rho] = atmosisa(hf);
T   = convtemp(T,'K','R');</pre></td><td>Function to find the temperature from a given pressure altitude in ft</td></tr>
</table>

``end```	
``function [out] = intrp(in,X,Y)``` ```% interpolate a 2-d function``` ```% in: input value of x``` ```% X: input vector``` ```% Y: output vector``` ```if in <= X(1)``` ``` out = Y(1);``` ```elseif in >= X(length(X))``` ``` out = Y(length(X));``` ```else``` ``` for i = 1:length(X)-1``` ``` if (in > X(i)) && (in <= X(i+1))``` ``` out = Y(i) + (in-X(i))*(Y(i+1)-Y(i))/(X(i+1)-``` ```X(i));``` ``` end``` ``` end``` ```end``` ```end```	Function to interpolate a matrix for a range of values
``function [a0] = slope2D(Data,angle)``` ```% 2-d lift curve slope``` ```% add your function details here``` ```b = abs(angle);``` ```a0 = ((intrp(b,Data.alfa,Data.cl)) - (intrp(-``` ```b,Data.alfa,Data.cl)))/(2*b);``` ```end```	Function to find the 2-D lift curve slope given an angle
``function [a3D] = slope3D(a0,Data)``` ```% 3D airfoil lift curve slope/deg``` ```AR = ((Data.b)^2) / (Data.S);``` ```a3D = a0 / (1+((57.3*a0) / (pi*Data.e*AR)));```	Function to find the 3-D lift curve slope given an angle
``function [a] = Sound(T)``` ```% determine speed of sound``` ```% input T: temperature (deg R)``` ```% output a: speed of sound in ft/sec``` ```a = sqrt(1.4*1716*T);``` ```end```	Function to find the speed of sound given a temperature in Rankin
``function [R] = aerodynamics(v,Hp,Data)``` ```% aerodynamics``` ```% inputs``` ```% v velocity (ft/sec)``` ```% Hp pressure altitude (ft)``` ```% aircraft data (Data)``` ```% output R, result structure```	Function to find several aerodynamic properties given velocity, pressure

<pre>% R.V: velocity (ft/sec) % R.M: Mach number (n/d) % R.q: dynamic pressure (psf) % R.CL: lift coefficient (n/d) % R.aStall: stall angle (deg) % R.Thet: pitch attitude (deg) % R.CD: drag coefficient (n/d) % R.LD: lift to drag ratio % R.LD3b2: lift^3/2 / drag (n/d) % R.LD0p5: lift^1/2 / drag (n/d) % R.Di: induced drag (lb) % R.Do: profile viscous drag (lb) % R.TR: thrust required (lbs) % R.PR: power required (ft lb/sec) % R.RC: climb rate (fpm) % R.PA: power available (ft lb/sec) % R.TA: thrust available (lb) % R.Range: Range (miles) % R.End: endurance (hours) R.Hp = Hp; R.V = v; rho0 = getDensity(0); % SLS density rho = getDensity(Hp); % density t = getTemperature(Hp); % temperature c = Sound(t); % speed of sound R.M = R.V/c; % Mach number AR = Data.b^2/Data.S; % aspect ratio R.q = rho*R.V^2/2; % dynamic pressure psf R.CL = Data.Wg/(R.q*Data.S); % lift coefficient alo = Angle0Lift(Data); % angle for 0 lift R.aStall = AngleStall(Data); % stall angle (deg) if R.M < 1.0 % subsonic a0 = slope2D(Data,6)./sqrt(1-R.M^2); % 2-d lift curve slope /deg a3D = slope3D(a0,Data); % 3-d lift curve slope /deg CDi = R.CL^2/(pi()*Data.e*AR); % induced drag coefficent R.Thet = R.CL/a3D + alo; % pitch attitude (deg) else % supersonic a3D = 4./sqrt(R.M^2-1); % 3-d lift curve slope / rad R.Thet = R.CL/a3D; % pitch attitude (rad) rr = AR*sqrt(R.M^2-1); CDi = 4.*R.Thet^2*(1-1/(2*rr))/sqrt(R.M^2-1); % supersonic induced drag R.Thet = R.Thet * 180 / pi(); % pitch attitude (deg) end R.CD = Data.CD0 + CDi; % fuselage + induced drag coefficent R.LD = R.CL/R.CD; % lift / drag R.LD3b2 = R.CL^1.5/R.CD; % lift^1.5 / drag</pre>	attitude, and airplane data

<pre>R.LD0p5 = R.CL^0.5/R.CD; % lift^0.5 / drag R.Di = CDi *R.q*Data.S; % induced drag (lbs) R.Do = Data.CD0*R.q*Data.S; % viscous profile drag (lbs) R.TR = Data.Wg/R.LD; % thrust required (lbs) R.PR = R.TR*R.V; % power required (ft lb/sec) if strcmpi(Data.Type,'propeller') R.PA = Data.N*Data.hpmax*Data.eta*550.*rho/rho0; % propeller power available (ft lbs / sec) R.TA = 0.0; R.Range = Data.eta*R.LD*log(Data.Wg/Data.We)*550.*3600/(Data.sfc*5280); % miles R.End = Data.eta*R.LD3b2*sqrt(2*rho*Data.S)*(1/sqrt(Data.We)- 1/sqrt(Data.Wg))*550./ Data.sfc; % hours elseif strcmpi(Data.Type,'turbine') R.TA = Data.N*Data.tmax*rho/rho0; % turbine thrust available (lbs) R.PA = R.TA*R.V; % turbine power available (ft lbs / sec) R.Range = 2*sqrt(2/(rho*Data.S))*R.LD0p5*(sqrt(Data.Wg)- sqrt(Data.We))*3600/ (Data.sfc*5280); % miles R.End = R.LD*log(Data.Wg/Data.We)/Data.sfc; % hours end R.RC = (R.PA - R.PR)*60/Data.Wg; % climb rate, fpm end</pre>	
<pre>function [Data] = Tran(type) Data = struct; Data.name = 'DHC-515'; % name Data.sign = 'subsonic'; % call sign, or designation Data.Type = 'propeller'; % propulsion type - 'propeller' or 'turbine' Data.Wg = 43850.; % gross weight - (lbs (max takeoff weight)) Data.We = 28400.; % empty weight - (lbs (gross weight - payload - fuel)) Data.S = 1080.; % wing area, (ft2) Data.b = 93.92; % wing span, (ft) [93ft 11in] Data.e = 0.83; % Oswald spanwise efficiency, n/d Data.CD0 = 0.012; % airplane fusealge drag coefficient, n/d Data.Sweep = 0.0; % wing sweepback angle (deg) Data.eta = 0.52; % propulsive efficiency, n/d Data.N = 2; % number of engines Data.hpmax = 2380.0; % max engine power (hp), sea level standard static Data.tmax = 0; % max engine thrust (lb), sea level standard static % turboprop engines produce little thrust because all the % power goes to turn propeller shaft Data.sfc = 0.470; % specific fuel consumption; prop (lb / hp hr); turbine(lb / lb thrust hr)</pre>	Function that contains data structure for airplane.

Data.Airfoil = '4418'; % airfoil NACA Section Number (NACA 4418) Data.Na = 12; % # alfa table entries, ordered min to max Data.alfa = [-12 -10 -8 -6 -4 -2 0 2 4 6 8 10]; Data.cl = [-0.83 -0.63 -0.45 -0.25 0 0.15 0.35 0.55 0.7 0.9 1.02 1.12]; end	

Table 2: MATLAB Summary

Appendix B. MATLAB Routines

```
Data = Tran('DHC-515');


% create matrix for different velocity and altitudes
vel = linspace(0,327.173,10);
Hpres = linspace(0,14700,10);


% velocity changing with altitude
R1 = aerodynamics(vel(1),Hpres(1),Data);
R2 = aerodynamics(vel(2),Hpres(2),Data);
R3 = aerodynamics(vel(3),Hpres(3),Data);
R4 = aerodynamics(vel(4),Hpres(4),Data);
R5 = aerodynamics(vel(5),Hpres(5),Data);
R6 = aerodynamics(vel(6),Hpres(6),Data);
R7 = aerodynamics(vel(7),Hpres(7),Data);
R8 = aerodynamics(vel(8),Hpres(8),Data);
R9 = aerodynamics(vel(9),Hpres(9),Data);
R10 = aerodynamics(vel(10),Hpres(10),Data);


% @ SLS attitude
    V1 = aerodynamics(vel(1),Hpres(1),Data);
    V2 = aerodynamics(vel(2),Hpres(1),Data);
    V3 = aerodynamics(vel(3),Hpres(1),Data);
    V4 = aerodynamics(vel(4),Hpres(1),Data);
    V5 = aerodynamics(vel(5),Hpres(1),Data);
    V6 = aerodynamics(vel(6),Hpres(1),Data);
    V7 = aerodynamics(vel(7),Hpres(1),Data);
    V8 = aerodynamics(vel(8),Hpres(1),Data);
    V9 = aerodynamics(vel(9),Hpres(1),Data);
    V10 = aerodynamics(vel(10),Hpres(1),Data);
% @ Absolute ceiling
    V1max = aerodynamics(vel(1),Hpres(10),Data);
    V2max = aerodynamics(vel(2),Hpres(10),Data);
    V3max = aerodynamics(vel(3),Hpres(10),Data);
    V4max = aerodynamics(vel(4),Hpres(10),Data);
    V5max = aerodynamics(vel(5),Hpres(10),Data);
    V6max = aerodynamics(vel(6),Hpres(10),Data);
    V7max = aerodynamics(vel(7),Hpres(10),Data);
    V8max = aerodynamics(vel(8),Hpres(10),Data);
    V9max = aerodynamics(vel(9),Hpres(10),Data);
    V10max = aerodynamics(vel(10),Hpres(10),Data);
```

```
% Create matrix for values from aerodynamics.m
outputVmax = [V1max V2max V3max V4max V5max V6max V7max V8max V9max V10max]
outputV = [V1 V2 V3 V4 V5 V6 V7 V8 V9 V10]
output = [R1 R2 R3 R4 R5 R6 R7 R8 R9 R10]
PR_plot = [output.PR];
PRV_plot = [outputV.PR];
PRVmax_plot = [outputVmax.PR];
PA_plot = [output.PA];
PAV_plot = [outputV.PA];
PAVmax_plot = [outputVmax.PA];
Thet_plot = [output.Thet];
ThetV_plot = [outputV.Thet];
ThetVmax_plot = [outputVmax.Thet];
LD_plot = [output.LD];
LDV_plot = [outputV.LD];
LDVmax_plot = [outputVmax.LD];



% plots
    plot(vel,PR_plot) % power required
    hold on;
    plot(vel,PRV_plot)
    hold on;
    plot(vel,PRVmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'});
    xlabel('Velocity');
    ylabel('Power Required');


    plot(vel,PA_plot) % power availiable
    hold on;
    plot(vel,PAV_plot)
    hold on;
    plot(vel,PAVmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'});
    xlabel('Velocity');
    ylabel('Power Available');


    plot(vel,Thet_plot) % pitch attitude
    hold on;
    plot(vel,ThetV_plot)
    hold on;
    plot(vel,ThetVmax_plot)
    hold off;
```

```
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'});
    xlabel('Velocity');
    ylabel('Pitch Attitude');


    plot(vel,LD_plot) % lift to drag ratio
    hold on;
    plot(vel,LDV_plot)
    hold on;
    plot(vel,LDVmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');
    xlabel('Velocity');
    ylabel('L/D');
```

```
% Aerodynamic Efficiency
TR_plot = [output.TR]; % thrust required
TRV_plot = [outputV.TR];
TR_Vmaxplot = [outputVmax.TR];
Di_plot = [output.Di]; % induced drag
DiV_plot = [outputV.Di];
DiVmax_plot = [outputVmax.Di];
Do_plot = [output.Do]; % vicous drag
DoV_plot = [outputV.Do];
DoVmax_plot = [outputVmax.Do];


    plot(Di_plot,TR_plot) % TR vs. Di
    hold on;
    plot(DiV_plot,TRV_plot)
    hold on;
    plot(DiVmax_plot,TR_Vmaxplot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');
    xlabel('induced drag');
    ylabel('Thrust required');


    plot(Do_plot,TR_plot) % Do vs. TR
    hold on;
    plot(DoV_plot,TRV_plot)
    hold on;
    plot(DoVmax_plot,TR_Vmaxplot)
    hold off;
```

```
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','northeast');
    xlabel('viscous drag');
    ylabel('Thrust required');


    plot(Di_plot,PR_plot) % Di vs. PR
    hold on;
    plot(DiV_plot,PRV_plot)
    hold on;
    plot(DiVmax_plot,PRVmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');
    xlabel('induced drag');
    ylabel('Power required');


    plot(Do_plot,PR_plot) % Do vs. PR
    hold on;
    plot(DoV_plot,PRV_plot)
    hold on;
    plot(DoVmax_plot,PRVmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','northeast');
    xlabel('viscous drag');
    ylabel('Power required');
```

```
% Rate of climb


RC_plot = [output.RC]; % rate of climb matrix
plot(vel,RC_plot)
xlabel('Velocity');
ylabel('ROC');


plot(RC_plot,Thet_plot) % ROC vs. Pitch attitude
xlabel('max ROC');
ylabel('Pitch Attitude');
```

```
% Range
Range_plot = [output.Range]; % range matrix
RangeV_plot = [outputV.Range];
```

```
RangeVmax_plot = [outputVmax.Range];
plot(vel,Range_plot) % vel vs. range
hold on;
plot(vel,RangeV_plot)
hold on;
plot(vel,RangeVmax_plot)
hold off;
xlabel('Velocity');
ylabel('Range');
legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');
```

```
% Endurance
End_plot = [output.End]; % endurance matrix
EndV_plot = [outputV.End];
EndVmax_plot = [outputVmax.End];
plot(vel,End_plot) % velocity vs. endurance
hold on;
plot(vel,EndV_plot)
hold on;
plot(vel,EndVmax_plot)
hold off;
xlabel('Velocity');
ylabel('Endurance');
legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');


LD3b2_plot = [output.LD3b2];
LD3b2V_plot = [outputV.LD3b2];
LD3b2Vmax_plot = [outputVmax.LD3b2];
    plot(vel,LD3b2_plot)
    hold on;
    plot(vel,LD3b2V_plot)
    hold on;
    plot(vel,LD3b2Vmax_plot)
    hold off;
    legend ({'vel x Hp', 'vel x SLS Hp', 'vel x Ceiling Hp'},
'Location','southeast');
    xlabel('Velocity');
    ylabel('L(3/2) / D');
```