Aerodynamics of a Cambered Airfoil ASEN 2002 September 28, 2022

Rebecca Rivera[*], Elliott Tung[†], Patrick Paluszek[‡], Connor O'Reilly [§]
University of Colorado - Boulder

This report covers the experiments and analysis done to observe how the angle of attack and free stream velocity affect the pressure, lift, and drag coefficients of a cambered airfoil. First, the surface pressure distribution was integrated, then, using the measured data and the equations for the lift and drag coefficients, values for each coefficient were computed. Measurements were taken at a variety of angles of attack, and at three different airspeeds. Based on the graphs created using the computed coefficients, it can be observed that as the angle of attack increases, both the lift and drag coefficients initially decrease but then increase linearly until they reach the stall point. Also, by comparing the plots it is observed that airspeed has little affect on the lift and drag coefficients but it does affect the accuracy of the measurements. Plots for the higher airspeeds matched those produced by NACA for the same airfoil.

Nomenclature

C_p	pressure coefficient
μ_∞	free-stream viscosity
V_∞	free-stream velocity
ρ_∞	free-stream density
p_∞	free-stream pressure
S	surface area
c	chord length
L	lift force
D	drag force
P	pressure
C_L	coefficient of drag
C_D	coefficient of lift
l	lift forces per unit span
d	drag forces per unit span
C_l	sectional coefficient of lift
C_d	sectional coefficient of drag
c	chord

[*]SID: 106306495
[†]SID: 106968675
[‡]SID: 106967379
[§]SID: 107054811

I. Introduction

The purpose of this lab was to observe the pressure distribution over a cambered Clark Y-14 airfoil and see how the angle of attack affects the lift and drag forces across the airfoil. To determine the lift and drag coefficients and the pressure coefficient, the differential pressure was measured across the airfoil at 32 angles of attack and three velocities at each angle. This showed how the lift and drag forces on the airfoil vary with the angle of attack. Data was used from all the groups in the lab section which allowed each group to take measurements at four angles of attack. The data found was then used to find at which angle the airfoil produced the most lift and at what angle stall occurred.

II. Experimental Setup and Measurement Techniques

In order to get the proper measurements, it is necessary to perform the experiment in a wind tunnel. In this case the ITLL Low-Speed Wind Tunnel was used with the Clark Y-14 airfoil mounted vertically, spanning the entire test section. The airfoil is equipped with 19 static pressure ports that are spread out over the entire top and bottom of the airfoil to take measurements of the pressure at different sections. Figure 1 shows the location of the ports.

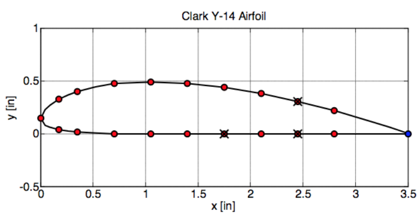

Figure 1. Schematic of Clark Y-14 Airfoil With Ports

Clark Y-14 Airfoil			Scanivalve Port #
Port #	x [in]	y [in]	
1	0	0.14665	1
2	0.175	0.33075	2
3	0.35	0.4018	3
4	0.7	0.476	4
5	1.05	0.49	5
6	1.4	0.4774	6
7	1.75	0.4403	7
8	2.1	0.38325	8
9	2.45	0.308	Not Connected
10	2.8	0.21875	9
11	3.5	0	Theoretical (Not Physical)
12	2.8	0	10
13	2.45	0	Not Connected
14	2.1	0	11
15	1.75	0	Not Connected
16	1.4	0	12
17	1.05	0	13
18	0.7	0.0014	14
19	0.35	0.0175	15
20	0.175	0.03885	16
Wind Tunnel Static Pressure Ring			Reference

Figure 2. Measurement Port Configuration

The procedure started by first checking to see if the pressure tubes for the pitot-static tube were connected to the pressure transducer and that the pressure tubes connected to the airfoil model were connected to the Scanivalve pressure scanner. Because there are only 16 channels for the pressure scanner, ports 9, 13, and 15 were not connected. There is no port at the trailing edge of the airfoil so the pressure at that point had to be estimated using the two nearest ports on top and bottom and then averaging the two extrapolated pressures.

After connecting all the tubes to the proper ports, the airfoil had to be aligned to the first assigned angle of attack. Once everything was hooked up and ready to go, measurements were taken at 9 m/s, 17 m/s, and 34 m/s. The data was recorded and the procedure was repeated for the next three angles of attack. All groups performed the same procedure but at different angles of attack to obtain data for all 32 angles.

III. Post-processing and Calculation of Force Coefficients

The raw measurements from each group were compiled and averaged so that they could be used to make the necessary calculations. Every group took 60 measurements at each angle of attack and of those 60 measurements, 20 were taken at each velocity. The averages were then organized by velocity and angle of attack in a manner that they could be easily referenced.

Major sources of uncertainty include experimental error and assumptions made. The angle of attack was set manually so human error must be taken into consideration. This would affect the measurements and therefore the results of the calculations. In addition, assumptions made during the calculations can cause a discrepancy between the actual and calculated values. For example, one assumption made is that the airfoil perfectly spans the test section and therefore the flow over the airfoil is two-dimensional. It is obvious that this is not the case in the wind tunnel since the airfoil has a root and a tip which may cause a small disturbance in the pressures measured.

We believe that the estimate for the pressure at the trailing edge is fairly accurate because extrapolating from the two bottom ports leads directly to the theoretical port because those three ports have an x location of 0 in. The extrapolation from the top two ports is less accurate and would be the cause of uncertainty in the estimated value, however, by averaging the two extrapolated pressures we ended up with a good approximation of the actual pressure at that point.

IV. Airfoil Static Pressure Coefficient Distribution

The pressure coefficient was calculated by using the averaged data in Eq. 1.

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} = \frac{p - p_\infty}{q_\infty} \tag{1}$$

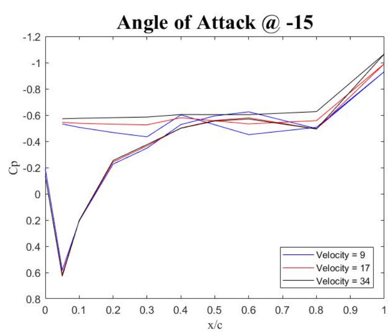

Figure 3. Clark Y-14 Airfoil Pressure Coefficient Distribution

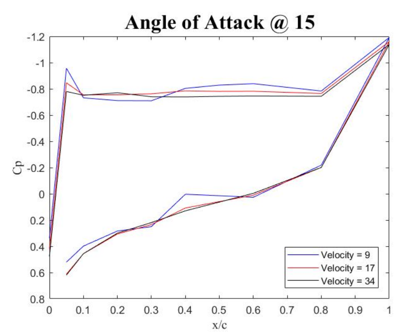

Figure 4. Clark Y-14 Airfoil Pressure Coefficient Distribution

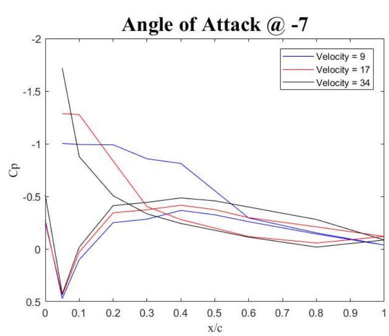

Figure 5. Clark Y-14 Airfoil Pressure Coefficient Distribution

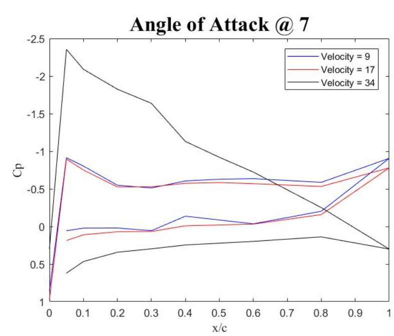

Figure 6. Clark Y-14 Airfoil Pressure Coefficient Distribution

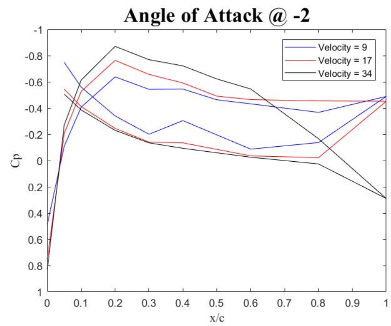

Figure 7. Clark Y-14 Airfoil Pressure Coefficient Distribution

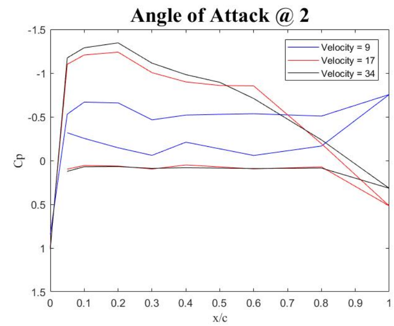

Figure 8. Clark Y-14 Airfoil Pressure Coefficient Distribution

The pressure distributions at different speeds do have different curves. At both 17 and 34 m/s shown in Fig. 7 and Fig. 8, they are essentially the same, but at 9 m/s, the pressure distribution looks premature. The pressure distribution seems to follow the same correlation at low speeds from Aerodynamics Lab 1. Where at lower speeds, the graphs do not look at as what they should because the uncertainty is greater at lower speeds, thus causing the pressure distribution at 9 m/s to make sense.

At lower angles of attack, the pressure distributions essentially follow the same trend. As the angle of attack increases, the falling pressure gradient increases in magnitude after the absolute maximum pressure coefficient is reached. Which also causes a by-product of increased lift coefficient because the area between the top and bottom curves has increased.

From observing Fig. 3 through Fig. 8, it seems that the maximum pressure distribution difference between the top and bottom surfaces of the airfoil occurs somewhere within the first quarter chord (25 percent chord). In Fig. 8, it is achieved at at 20 percent while in Fig. 4, it is achieved at 6 percent chord. This maximum occurs earlier and earlier as the angle of attack is increased. This tells us that the Clark Y-14 airfoil's pitching moment is ever so increasing from the quarter chord as the angle of attack is increased. We can expect that this Clark Y-14 airfoil's pitching moment will increase as the angle of attack is increased.

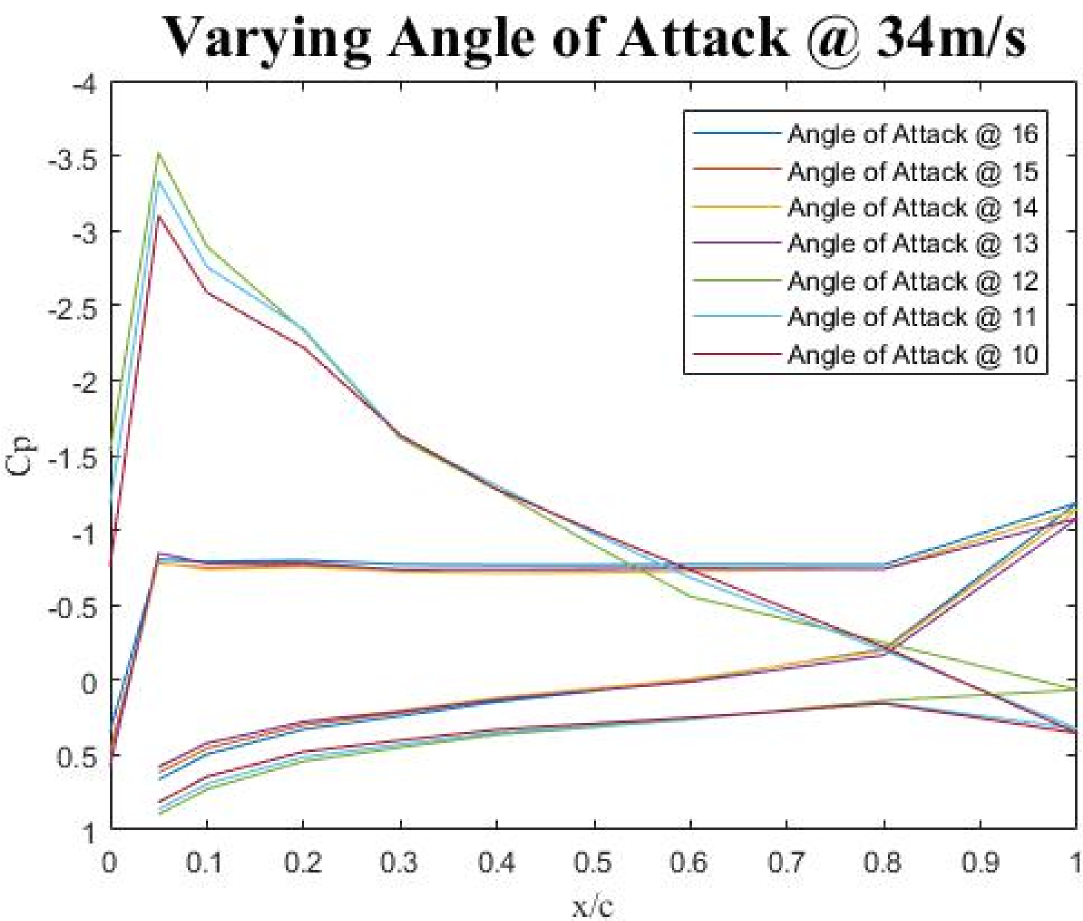

Figure 9. Evidence of Flow Separation

There is evidence that flow separation has occurred. At 13 degrees angle of attack, the maximum pressure distribution difference between the top surface and the bottom surface drops to a much lower value compared to the angles of attack lower than 13 degrees, as observed from Fig. 9. Increasing the angle of attack past 13 degrees will result in nearly the same pressure distribution. This tells us that flow separation causes a minimum coefficient of lift to occur passed 13 degrees of attack since the area between the curves stays nearly constant.

V. Lift and Pressure Drag Coefficients

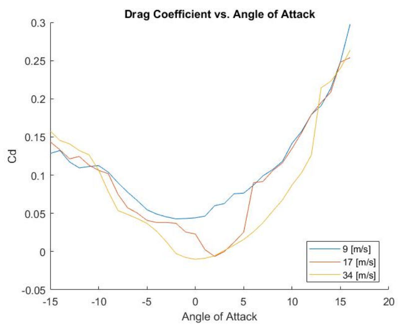

Figure 10. Clark Y-14 Airfoil Pressure Coefficient Curve

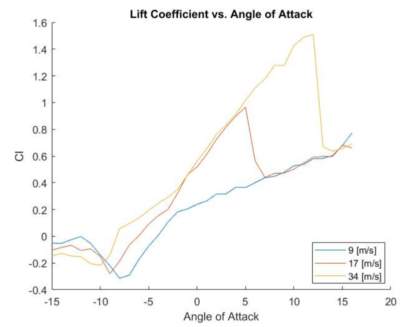

Figure 11. Clark Y-14 Airfoil Pressure Coefficient Curve

The lift and drag coefficients of the airfoil vary with angle of attack as seen in Fig. 7 and Fig. 8. The tested angles of attack were -15 to 16 degrees, incrementing by one degree. Fig. 7 shows how the drag coefficient starts at values around 0.15 and as the angle of attack is increased, the drag coefficient first gradually decreases to a minimum value of just under zero and then gradually increases to its maximum value which is between 0.25 and 0.3. The lift coefficient starts just under 0 and briefly decreases until it reaches an angle of attack of approximately -9 or -10 degrees. Then, the lift increases almost linearly until sharp leading edge stall can be observed. The angle of attack where stall occurs varies greatly with the free stream velocity. At 34 m/s stall occurs at angle of attack of about 12 degrees and at 17 m/s stall can be observed at only 5 degrees. The 9 m/s data doesn't appear to stall at the tested angles of attack. The type of stall the airfoil experiences is based on the free stream velocity and shape of the airfoil.

The maximum lift coefficient varies with the tunnel velocity. At 9 m/s the maximum lift coefficient is just under 0.8 at 16 degrees angle of attack. The maximum lift coefficient at 17 m/s is slightly less than one at a significantly lower angle of attack, approximately 5 degrees. The maximum lift coefficient happens at 34 m/s and is approximately 1.5 at 13 degrees.

At zero angle of attack the lift coefficient is 0.43 based on the curves from 17 and 34 m/s. The value is slightly lower at 9 m/s. The lift coefficient is not zero at zero angle of attack because of the camber of the airfoil. The lift coefficient is zero at angles of attack -9 to -5 degrees depending on the free stream velocity. The curve only passes through zero as the lift coefficient begins its linear increase.

Our experimental results for the airfoil tested at 17 m/s and 34 m/s accurately match the slope of the linear lift region of the NACA results for the Clark Y-14 airfoil. However, the experimental data shows the airfoil reaching stall at angles of attack lower than that of the NACA results shown in Fig. 12. At 9 m/s the results are much less accurate and the linear region is a very poor representation of the coefficient of lift. Looking at both the lift and drag plots, the 17 and 34 m/s curves are very similar and much more accurate and the 9 m/s curve is the out-lier. This is because measurements on the airfoil are much less accurate at lower air speeds.

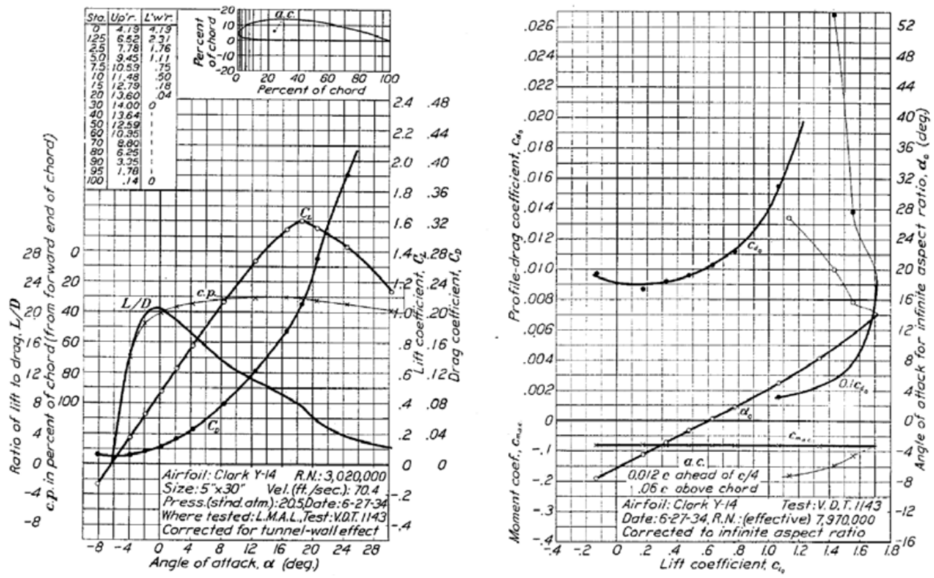

Figure 12. Aerodynamic Coefficients

The following equations were implemented into MATLAB to solve for and plot the coefficients of lift and drag:

$$n_i = -\frac{1}{2}(p_i + p_{i+1})\Delta x_i \tag{2}$$

$$a_i = \frac{1}{2}(p_i + p_{i+1})\Delta y_i \tag{3}$$

$$n = -\sum_{i=1}^{n} \frac{1}{2}(p_i + p_{i+1})\Delta x_i \tag{4}$$

$$a = \sum_{i=1}^{n} \frac{1}{2}(p_i + p_{i+1})\Delta y_i \tag{5}$$

$$C_n = \sum_{i=1}^{n} \frac{1}{2}(C_{p_i} + C_{p_i+1})\frac{\Delta x_i}{c} \tag{6}$$

$$C_a = \sum_{i=1}^{n} \frac{1}{2}(C_{p_i} + C_{p_i+1})\frac{\Delta y_i}{c} \tag{7}$$

$$C_l = C_n cos\alpha - C_a sin\alpha \tag{8}$$

$$C_d = C_n sin\alpha + C_a cos\alpha \tag{9}$$

VI. Conclusion

After calculating the lift, drag, and pressure coefficients from the pressures recorded on the airfoil, a variety of important concepts can be noted. As the angle of attack increases from zero, the maximum difference between the top and bottom curves of the pressure distributions increase until an angle of attack of 13 degrees; any angle of attack including 13 and above will be decreased and the same. Thus increasing the lift coefficient till the stall point. The lift coefficient shows a linearly increasing trend until the stall point as the angle of attack is increased. The drag coefficient, however, decreases, reaches a minimum, and then increases to a maximum as the angle of attack increases from -15 to 16 degrees. When calculating the coefficients, these values may have a value of uncertainty. Experimental as well as initial assumptions can cause these calculated values to be slightly off from the actual values.

References

[1]Anderson, J. D., Introduction to flight, New York, NY: McGraw Hill Education, 2016.

[2]"$ASEN 2002 Aerodynamics Experimental Laboratory 1 : Calibration of the ITLL Low - Speed Wind Tunnel,$"$Web, 2017$.

Acknowledgments

We would like to acknowledge the valuable assistance we received from the teaching assistants and lab assistants that helped us with both the data collection process and with understanding crucial points in the analysis.

VII. Appendix

A. Code

```
% Calculate lift and drag coefficients for every velocity and angle of
% attack

clear
clc

% read in data
[vel9, vel17, vel34] = latlab();

c = 0.0889; % chord length [m], 3.5 [in]
% input x and y locations of ports
x = [0 0.175 0.35 0.7 1.05 1.4 1.75 2.1 2.8 3.5 2.8 2.1 1.4 1.05 0.7 0.35
   0.175];
y = [0.14665 0.33075 0.4018 0.476 0.49 0.4774 0.4403 0.38325 0.21875 0 0 0
    0 0 0.0014 0.0175 0.03885];

%% 9 [m/s]
angles = -15:16;
for i = 1:numel(angles) % calculate Cl and Cd for each angle of attack
    % calculations for ports 1-9 (before trailing edge)
    for j = 1:8
        % ports 1-9
        % average the pressure between each port
        Pavg(j) = (1/2)*(vel9(i,(j+6))+vel9(i,(j+7))); % Pavg(i) = (1/2)*(
           P(i)+P(i+1));
        % calculate the normal force between each port
        n(j) = -Pavg(j)*(x(j+1)-x(j));
        % calculate the axial force between each port
        a(j) = Pavg(j)*(y(j+1)-y(j));
    end
    % calculations for ports 9-11 (between 9,10 and 10,11)
    Pavg(9) = (1/2)*(vel9(i,15)+P11);
    n(9) = -Pavg(9)*(x(10)-x(9));
    a(9) = Pavg(9)*(y(10)-y(9));

    Pavg(10) = (1/2)*(P11+vel9(i,17));
    n(10) = -Pavg(10)*(x(11)-x(10));
    a(10) = Pavg(10)*(y(11)-y(10));

    % calculations for ports 11-17 (after trailing edge)
    for j = 10:16
        % ports 11-17
        % average the pressure between each port
        Pavg(j) = (1/2)*(vel9(i,(j+6))+vel9(i,(j+7))); % Pavg(i) = (1/2)*(
           P(i)+P(i+1));
        % calculate the normal force between each port
        n(j) = -Pavg(j)*(x(j+2)-x(j+1));
        % calculate the axial force between each port
        a(j) = Pavg(j)*(y(j+2)-y(j+1));
    end

    % avg between port 17 and 1
```

```
    Pavg(17) = (1/2)*(vel9(i,22)+vel9(i,7));
    % normal and axial forces between ports 17 and 1
    n(17) = -Pavg(17)*(x(17)-x(1));
    a(17) = Pavg(17)*(y(17)-y(1));

    % sum normal and axial forces over all ports to get totals
    N = sum(n);
    A = sum(a);

    % non-dimensionalize the total forces
    Cn = N/(dynamicP*c);
    Ca = A/(dynamicP*c);

    % transfer to freestream coordinate system for each angle of attack
    Cl(i) = Cn*cosd(angles(i)) - Ca*sind(angles(i));
    Cd(i) = Cn*sind(angles(i)) - Ca*cosd(angles(i));

end

%% 17 [m/s]

%% 34 [m/s]

% Housekeeping


clc
clear
close all

[sohelpmegod] = why_do_the_matlab_gods_hate_me();

% Read in data
data1 = xlsread('AirfoilPressure_S011_G01.csv');
data3 = xlsread('AirfoilPressure_S011_G03.csv');
data5 = xlsread('AirfoilPressure_S011_G05.csv');
data7 = xlsread('AirfoilPressure_S011_G07.csv');
data9 = xlsread('AirfoilPressure_S011_G09.csv');
data11 = xlsread('AirfoilPressure_S011_G11.csv');
data13 = xlsread('AirfoilPressure_S011_G13.csv');
data15 = xlsread('AirfoilPressure_S011_G15.csv');

% Organize data in matrix
% range of angle of attack is -15:16 [degrees]
% tested velocities are 9,17,34 [m/s]

angleN15 = data15(1:60,1:23);
angleN14 = data13(1:60,1:23);
angleN13 = data11(1:60,1:23);
angleN12 = data9(1:60,1:23);
angleN11 = data7(1:60,1:23);
angleN10 = data5(1:60,1:23);
angleN9 = data3(1:60,1:23);
angleN8 = data1(1:60,1:23);
angleN7 = data15(61:120,1:23);
angleN6 = data13(61:120,1:23);
```

```
angleN5 = data11(61:120,1:23);
angleN4 = data9(61:120,1:23);
angleN3 = data7(61:120,1:23);
angleN2 = data5(61:120,1:23);
angleN1 = data3(61:120,1:23);
angle0 = data1(61:120,1:23);
angle1 = data15(121:180,1:23);
angle2 = data13(121:180,1:23);
angle3 = data11(121:180,1:23);
angle4 = data9(121:180,1:23);
angle5 = data7(121:180,1:23);
angle6 = data5(121:180,1:23);
angle7 = data3(121:180,1:23);
angle8 = data1(121:180,1:23);
angle9 = data15(181:240,1:23);
angle10 = data13(181:240,1:23);
angle11 = data11(181:240,1:23);
angle12 = data9(181:240,1:23);
angle13 = data7(181:240,1:23);
angle14 = data5(181:240,1:23);
angle15 = data3(181:240,1:23);
angle16 = data1(181:240,1:23);

% average values of select columns in 'angle...' for rows in increments of
    20
% average dynamic pressure and diff. pressure at all ports,
% store averages in 1x21 matrices for each velocity and angle of attack

avgsN15_9 = sum(angleN15(1:20,:))/20;
avgsN15_17 = sum(angleN15(21:40,:))/20;
avgsN15_34 = sum(angleN15(41:60,:))/20;

avgsN14_9 = sum(angleN14(1:20,:))/20;
avgsN14_17 = sum(angleN14(21:40,:))/20;
avgsN14_34 = sum(angleN14(41:60,:))/20;

avgsN13_9 = sum(angleN13(1:20,:))/20;
avgsN13_17 = sum(angleN13(21:40,:))/20;
avgsN13_34 = sum(angleN13(41:60,:))/20;

avgsN12_9 = sum(angleN12(1:20,:))/20;
avgsN12_17 = sum(angleN12(21:40,:))/20;
avgsN12_34 = sum(angleN12(41:60,:))/20;

avgsN11_9 = sum(angleN11(1:20,:))/20;
avgsN11_17 = sum(angleN11(21:40,:))/20;
avgsN11_34 = sum(angleN11(41:60,:))/20;

avgsN10_9 = sum(angleN10(1:20,:))/20;
avgsN10_17 = sum(angleN10(21:40,:))/20;
avgsN10_34 = sum(angleN10(41:60,:))/20;

avgsN9_9 = sum(angleN9(1:20,:))/20;
avgsN9_17 = sum(angleN9(21:40,:))/20;
avgsN9_34 = sum(angleN9(41:60,:))/20;
```

```
avgsN8_9 = sum(angleN8(1:20,:))/20;
avgsN8_17 = sum(angleN8(21:40,:))/20;
avgsN8_34 = sum(angleN8(41:60,:))/20;

avgsN7_9 = sum(angleN7(1:20,:))/20;
avgsN7_17 = sum(angleN7(21:40,:))/20;
avgsN7_34 = sum(angleN7(41:60,:))/20;

avgsN6_9 = sum(angleN6(1:20,:))/20;
avgsN6_17 = sum(angleN6(21:40,:))/20;
avgsN6_34 = sum(angleN6(41:60,:))/20;

avgsN5_9 = sum(angleN5(1:20,:))/20;
avgsN5_17 = sum(angleN5(21:40,:))/20;
avgsN5_34 = sum(angleN5(41:60,:))/20;

avgsN4_9 = sum(angleN4(1:20,:))/20;
avgsN4_17 = sum(angleN4(21:40,:))/20;
avgsN4_34 = sum(angleN4(41:60,:))/20;

avgsN3_9 = sum(angleN3(1:20,:))/20;
avgsN3_17 = sum(angleN3(21:40,:))/20;
avgsN3_34 = sum(angleN3(41:60,:))/20;

avgsN2_9 = sum(angleN2(1:20,:))/20;
avgsN2_17 = sum(angleN2(21:40,:))/20;
avgsN2_34 = sum(angleN2(41:60,:))/20;

avgsN1_9 = sum(angleN1(1:20,:))/20;
avgsN1_17 = sum(angleN1(21:40,:))/20;
avgsN1_34 = sum(angleN1(41:60,:))/20;

avgs0_9 = sum(angle0(1:20,:))/20;
avgs0_17 = sum(angle0(21:40,:))/20;
avgs0_34 = sum(angle0(41:60,:))/20;

avgs1_9 = sum(angle1(1:20,:))/20;
avgs1_17 = sum(angle1(21:40,:))/20;
avgs1_34 = sum(angle1(41:60,:))/20;

avgs2_9 = sum(angle2(1:20,:))/20;
avgs2_17 = sum(angle2(21:40,:))/20;
avgs2_34 = sum(angle2(41:60,:))/20;

avgs3_9 = sum(angle3(1:20,:))/20;
avgs3_17 = sum(angle3(21:40,:))/20;
avgs3_34 = sum(angle3(41:60,:))/20;

avgs4_9 = sum(angle4(1:20,:))/20;
avgs4_17 = sum(angle4(21:40,:))/20;
avgs4_34 = sum(angle4(41:60,:))/20;

avgs5_9 = sum(angle5(1:20,:))/20;
avgs5_17 = sum(angle5(21:40,:))/20;
```

```
avgs5_34 = sum(angle5(41:60,:))/20;

avgs6_9 = sum(angle6(1:20,:))/20;
avgs6_17 = sum(angle6(21:40,:))/20;
avgs6_34 = sum(angle6(41:60,:))/20;

avgs7_9 = sum(angle7(1:20,:))/20;
avgs7_17 = sum(angle7(21:40,:))/20;
avgs7_34 = sum(angle7(41:60,:))/20;

avgs8_9 = sum(angle8(1:20,:))/20;
avgs8_17 = sum(angle8(21:40,:))/20;
avgs8_34 = sum(angle8(41:60,:))/20;

avgs9_9 = sum(angle9(1:20,:))/20;
avgs9_17 = sum(angle9(21:40,:))/20;
avgs9_34 = sum(angle9(41:60,:))/20;

avgs10_9 = sum(angle10(1:20,:))/20;
avgs10_17 = sum(angle10(21:40,:))/20;
avgs10_34 = sum(angle10(41:60,:))/20;

avgs11_9 = sum(angle11(1:20,:))/20;
avgs11_17 = sum(angle11(21:40,:))/20;
avgs11_34 = sum(angle11(41:60,:))/20;


avgs12_9 = sum(angle12(1:20,:))/20;
avgs12_17 = sum(angle12(21:40,:))/20;
avgs12_34 = sum(angle12(41:60,:))/20;

avgs13_9 = sum(angle13(1:20,:))/20;
avgs13_17 = sum(angle13(21:40,:))/20;
avgs13_34 = sum(angle13(41:60,:))/20;

avgs14_9 = sum(angle14(1:20,:))/20;
avgs14_17 = sum(angle14(21:40,:))/20;
avgs14_34 = sum(angle14(41:60,:))/20;

avgs15_9 = sum(angle15(1:20,:))/20;
avgs15_17 = sum(angle15(21:40,:))/20;
avgs15_34 = sum(angle15(41:60,:))/20;

avgs16_9 = sum(angle16(1:20,:))/20;
avgs16_17 = sum(angle16(21:40,:))/20;
avgs16_34 = sum(angle16(41:60,:))/20;

mat9 = [avgsN15_9; avgsN14_9; avgsN13_9; avgsN12_9; avgsN11_9; avgsN10_9;
   avgsN9_9; avgsN8_9; avgsN7_9; avgsN6_9; avgsN5_9; avgsN4_9; avgsN3_9;
   avgsN2_9; avgsN1_9; avgs0_9; avgs1_9; avgs2_9; avgs3_9; avgs4_9;
   avgs5_9; avgs6_9; avgs7_9; avgs8_9; avgs9_9; avgs10_9; avgs11_9;
   avgs12_9; avgs13_9; avgs14_9; avgs15_9; avgs16_9  ];
mat17 = [avgsN15_17; avgsN14_17; avgsN13_17; avgsN12_17; avgsN11_17;
   avgsN10_17; avgsN9_17; avgsN8_17; avgsN7_17; avgsN6_17; avgsN5_17;
   avgsN4_17; avgsN3_17; avgsN2_17; avgsN1_17; avgs0_17; avgs1_17;
```

```
    avgs2_17; avgs3_17; avgs4_17; avgs5_17; avgs6_17; avgs7_17; avgs8_17;
    avgs9_17; avgs10_17; avgs11_17; avgs12_17; avgs13_17; avgs14_17;
    avgs15_17; avgs16_17  ];
mat34 = [avgsN15_34; avgsN14_34; avgsN13_34; avgsN12_34; avgsN11_34;
    avgsN10_34; avgsN9_34; avgsN8_34; avgsN7_34; avgsN6_34; avgsN5_34;
    avgsN4_34; avgsN3_34; avgsN2_34; avgsN1_34; avgs0_34; avgs1_34;
    avgs2_34; avgs3_34; avgs4_34; avgs5_34; avgs6_34; avgs7_34; avgs8_34;
    avgs9_34; avgs10_34; avgs11_34; avgs12_34; avgs13_34; avgs14_34;
    avgs15_34; avgs16_34  ];



ra9 = mat9(:,1:15);
da9 = sohelpmegod(1,:)';
na9 = mat9(:,16:23);

ra17 = mat17(:,1:15);
da17 = sohelpmegod(2,:)';
na17 = mat17(:,16:23);

ra34 = mat34(:,1:15);
da34 = sohelpmegod(3,:)';
na34 = mat34(:,16:23);

umat9 = [ra9 da9 na9];
umat17 = [ra17 da17 na17];
umat34 = [ra34 da34 na34];

clear
clc

% Read in data
data1 = xlsread('AirfoilPressure_S011_G01.csv');
data3 = xlsread('AirfoilPressure_S011_G03.csv');
data5 = xlsread('AirfoilPressure_S011_G05.csv');
data7 = xlsread('AirfoilPressure_S011_G07.csv');
data9 = xlsread('AirfoilPressure_S011_G09.csv');
data11 = xlsread('AirfoilPressure_S011_G11.csv');
data13 = xlsread('AirfoilPressure_S011_G13.csv');
%data15 = xlsread('AirfoilPressure_S011_G15.csv');


% Organize data in matrix
% range of angle of attack is -15:16 [degrees]
% tested velocities are 9,17,34 [m/s]

%angleN15 = data15(1:60,1:23);
angleN14 = data13(1:60,1:23);
angleN13 = data11(1:60,1:23);
angleN12 = data9(1:60,1:23);
angleN11 = data7(1:60,1:23);
angleN10 = data5(1:60,1:23);
angleN9 = data3(1:60,1:23);
angleN8 = data1(1:60,1:23);
%angleN7 = data15(1:60,1:23);
```

```
angleN6 = data13(61:120,1:23);
angleN5 = data11(61:120,1:23);
angleN4 = data9(61:120,1:23);
angleN3 = data7(61:120,1:23);
angleN2 = data5(61:120,1:23);
angleN1 = data3(61:120,1:23);
angle0 = data1(61:120,1:23);
%angle1 = data15(121:180,1:23);
angle2 = data13(121:180,1:23);
angle3 = data11(121:180,1:23);
angle4 = data9(121:180,1:23);
angle5 = data7(121:180,1:23);
angle6 = data5(121:180,1:23);
angle7 = data3(121:180,1:23);
angle8 = data1(121:180,1:23);
%angle9 = data15(181:240,1:23);
angle10 = data13(181:240,1:23);
angle11 = data11(181:240,1:23);
angle12 = data9(181:240,1:23);
angle13 = data7(181:240,1:23);
angle14 = data5(181:240,1:23);
angle15 = data3(181:240,1:23);
angle16 = data1(181:240,1:23);

% average values of select columns in 'angle...' for rows in increments of
    20
% average dynamic pressure and diff. pressure at all ports,
% store averages in 1x21 matrices for each velocity and angle of attack

%avgsN15_9 =

function [sohelpmegod] = extrapolation_please()
%{

   Purpose: Function will linear extrapolate using the pressure readings
      from
            ports 8 and 9 and then ports 11 and 12 and then average the
               two results to
            use for the theoretical value of pressure at x = 3.5
   Inputs:  Data sheet
   Outputs: values for the extrapolated value of pressure at x = 3.5 for
      each group.
                         will return
%}

%initialize matrix
finals = zeros(240,16);
newfin = zeros(240,8);
results = zeros(12,8);
sohelpmegod = zeros(3,32);
%start calculations

for endmeplease = 1:8

        if endmeplease == 1
              whole_lotta_data = xlsread('AirfoilPressure_s011_G01.csv')
```

```
                ;
        end
        if endmeplease == 2
                whole_lotta_data = xlsread('AirfoilPressure_s011_G03.csv')
                    ;
        end
        if endmeplease == 3
                whole_lotta_data = xlsread('AirfoilPressure_s011_G05.csv')
                    ;
        end
        if endmeplease == 4
                whole_lotta_data = xlsread('AirfoilPressure_s011_G07.csv')
                    ;
        end
        if endmeplease == 5
                whole_lotta_data = xlsread('AirfoilPressure_s011_G09.csv')
                    ;
        end
        if endmeplease == 6
                whole_lotta_data = xlsread('AirfoilPressure_s011_G11.csv')
                    ;
        end
        if endmeplease == 7
                whole_lotta_data = xlsread('AirfoilPressure_s011_G13.csv')
                    ;
        end
        if endmeplease == 8
                whole_lotta_data = xlsread('AirfoilPressure_s011_G15.csv')
                    ;
        end

%scan pressures 8 and 10
data_8 = whole_lotta_data(:,14) ;
data_10 =  whole_lotta_data(:,15);

%scan pressures for port 12 and 14

data_12 = whole_lotta_data(:,16);
data_14 = whole_lotta_data(:,17);

%declare x values for calculations
x8 = 2.1;% in
x10 = 2.8;% in
x11 = 3.5;% in
x12 = 2.8;% in
x14 = 2.1;% in
for h = 1:240
        finals(h,(endmeplease*2)-1) = (((x11-x10)*(data_10(h,1)-data_8(h
            ,1)))/(x10-x8)) + data_10(h,1);
        finals(h,endmeplease*2) = (((x11-x12)*(data_12(h,1)-data_14(h,1)))
            /(x12-x14)) + data_12(h,1);
end
end
for i = 1:8
    newfin(:,i) = finals(:,(i*2)-1)+finals(:,i*2);
```

```
end


for z = 1:8
   results(1,z) = sum(newfin(1:20,z))/20;
   results(2,z) = sum(newfin(21:40,z))/20;
   results(3,z) = sum(newfin(41:60,z))/20;
   results(4,z) = sum(newfin(61:80,z))/20;
   results(5,z) = sum(newfin(81:100,z))/20;
   results(6,z) = sum(newfin(101:120,z))/20;
   results(7,z) = sum(newfin(121:140,z))/20;
   results(8,z) = sum(newfin(141:160,z))/20;
   results(9,z) = sum(newfin(161:180,z))/20;
   results(10,z) = sum(newfin(181:200,z))/20;
   results(11,z) = sum(newfin(201:220,z))/20;
   results(12,z) = sum(newfin(221:240,z))/20;

end

sohelpmegod(:,1:8) = results(1:3,8:-1:1);
sohelpmegod(:,9:16) = results(4:6,8:-1:1);
sohelpmegod(:,17:24) = results(7:9,8:-1:1);
sohelpmegod(:,25:32) = results(10:12,8:-1:1);
end

% Housekeeping


cp9 = (umat9(:,7:23))./umat9(:,5);
cp17 = umat17(:,7:23)./umat17(:,5);
cp34 = umat34(:,7:23)./umat34(:,5);

c = 3.5;
x = [0 0.175 0.35 0.7  1.05 1.4 1.75 2.1 2.8 3.5 2.8 2.1 1.4 1.05 0.7 .35
    0.175];
xc = x/c;

figure(1)
plot(xc,cp9(1,:),'b')
hold on
plot(xc,cp17(1,:),'r')
plot(xc,cp34(1,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
legend('location','southeast')
title ('Angle of Attack @ -15','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off

figure(2)
plot(xc,cp9(31,:),'b')
hold on
plot(xc,cp17(31,:),'r')
plot(xc,cp34(31,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
```

```
legend('location','southeast')
title ('Angle of Attack @ 15','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off

figure(3)
plot(xc,cp9(9,:),'b')
hold on
plot(xc,cp17(9,:),'r')
plot(xc,cp34(9,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
title ('Angle of Attack @ -7','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off

figure(4)
plot(xc,cp9(23,:),'b')
hold on
plot(xc,cp17(23,:),'r')
plot(xc,cp34(23,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
title ('Angle of Attack @ 7','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off

figure(5)
plot(xc,cp9(14,:),'b')
hold on
plot(xc,cp17(14,:),'r')
plot(xc,cp34(14,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
title ('Angle of Attack @ -2','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off

figure(6)
plot(xc,cp9(18,:),'b')
hold on
plot(xc,cp17(18,:),'r')
plot(xc,cp34(18,:),'k')
legend('Velocity = 9','Velocity = 17','Velocity = 34')
title ('Angle of Attack @ 2','FontName','Times','FontSize',20)
xlabel('x/c','FontName','Times','FontSize',12)
ylabel('Cp','FontName','Times','FontSize',12)
set(gca,'Ydir','reverse')
hold off
```