INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

AE 211(A) : INCOMPRESSIBLE FLOWS
TAKE HOME MID-SEM EXAM

Group No. 4
Atharv Mishra (180158)
Chada Sahith Reddy (180202)
Dipak Miglani(180248)
Irene Grace Karot Polson(180308)
Simran Singh (180765)

DATE: 19-02-2020

INTRODUCTION

Source Panel and Vortex Panel methods are a vital component in the analysis of 2D shapes in a flow. To get an essence of the same, we have done the analysis of airfoils and collected related attributes and present to you the same.

In this report we have tried to explain the observed phenomena using numerical calculations. We have obtained some relations in the process and have plotted the streamlines for both the cases.
all the codes we used and the results we obtained while solving the problem. We have also indicated the source codes wherever necessary .

Dedicated to *: Airfoils (literally!)*

SOURCE PANEL METHOD

While analyzing the airfoil using the Source panel method to calculate the potential flow around an arbitrary airfoil, we worked with a matlab code[1]. We changed the shape of the body under study to that of an airfoil whose equation is given by the following formula[a];

The airfoil is defined as a pair of parametric equations for $X(\theta)$ and $Y(\theta)$ for $\theta = \langle 0, 2\pi \rangle$,

$$X(\theta) = 0.5 + 0.5\frac{|\cos\theta|^B}{\cos\theta}, \quad (1)$$

$$Y(\theta) = \frac{T}{2}\frac{|\sin\theta|^B}{\sin\theta}(1 - X^P) + C\sin(X^E\pi) + R\sin(X2\pi), \quad (2)$$

We will be using the above equation as our model of airfoil for the source panel method.
In this we can vary the parameter theta to get the x and y coordinates of our required airfoil which we have picked by assigning B=2, T=0.2, C=0.05, P=1,E=1,R=0.The resultant airfoil has the following shape.

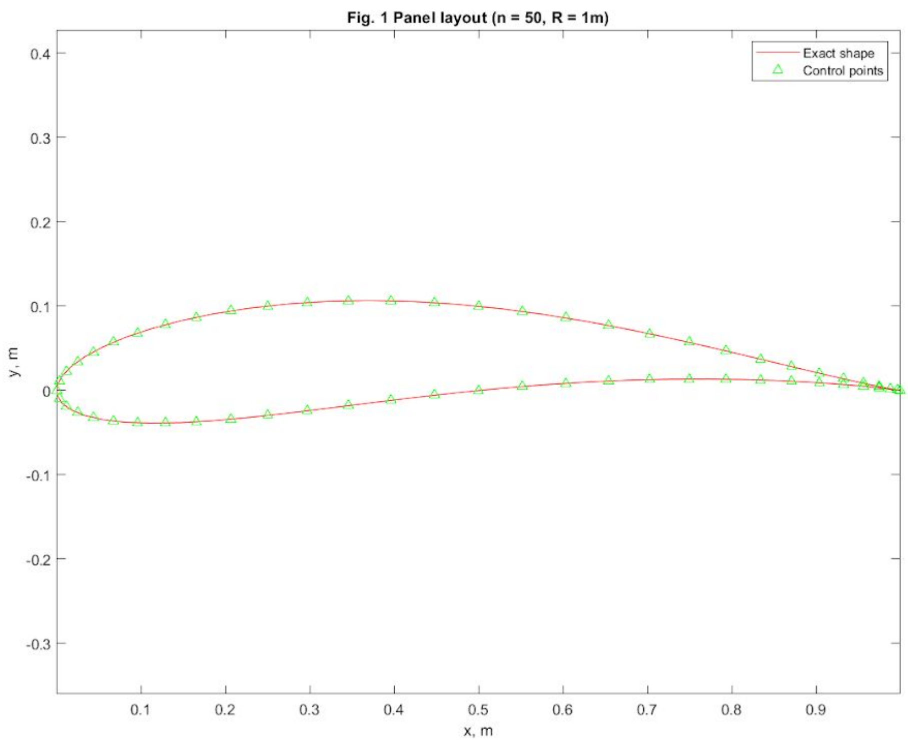


Figure displaying the airfoil obtained (in red) and control points (in green)

This is followed by the geometric calculations which were already provided to us. We use the calculated data to find other required parameters and results. Also, we did not particularly make panels but rather took this plotted airflow as the panel itself. We did this in order to get better results as more panels mean better calculations. Additionally, the V_inf was taken small so as to avoid magnification of error in further calculations due to assumptions.

At this point, we have calculated enough data to analyse the graphs between C_p Vs x/C and Vs (Tangent Velocity at the airfoil) Vs x/C where x is the length in the direction of the airfoil and C is the Chord length. The code for the same are simple and as illustrated below :

```
box on;
figure(2)    %Plotting conX vs Cp
plot(conX,Cp,'r');
grid;
title('Fig. 2 Comparison of C_p and X/C');
xlabel('X/C'); ylabel('C_p');

figure(3)    %Plotting conX Vs Vs
plot(conX, (Vs),'r');
grid;
title('Fig. 2 Comparison of V_s and X/C');
xlabel('X/C'); ylabel('V_s, m/s')
```

Figure showing the code for plotting the above mentioned graphs

The calculated value of the sum of all sources comes out to be -0.006808. This value should have been zero ideally. However, due to crude assumptions while calculating, we obtain it non zero (approximately zero).

The graphs obtained are as follows :

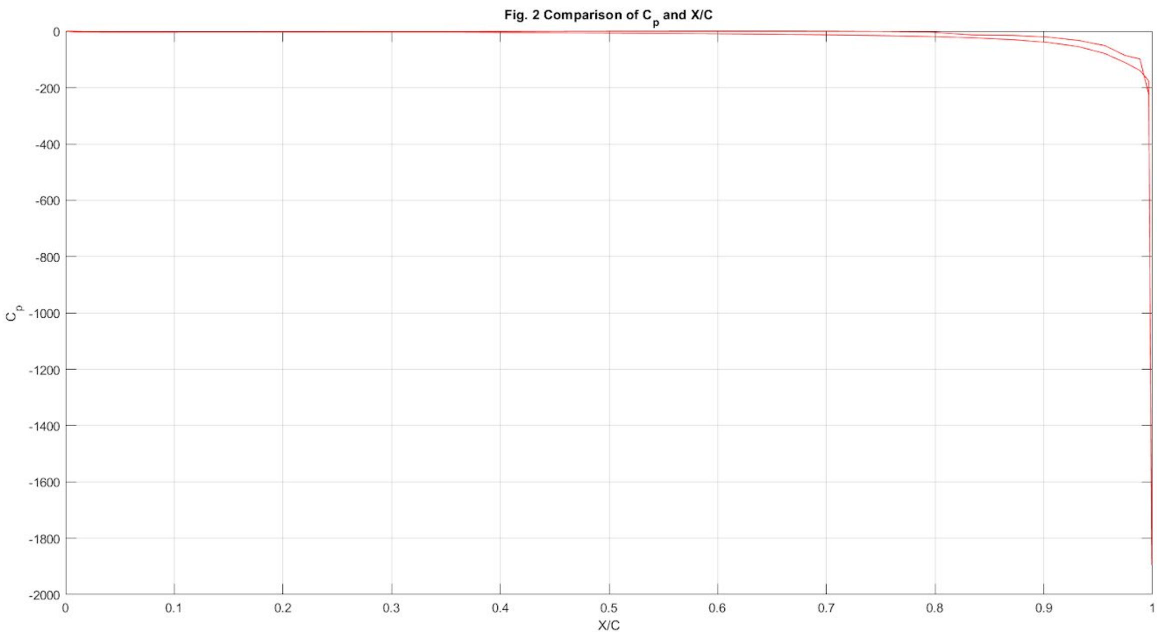


Figure depicting relation between C_p and x/C.

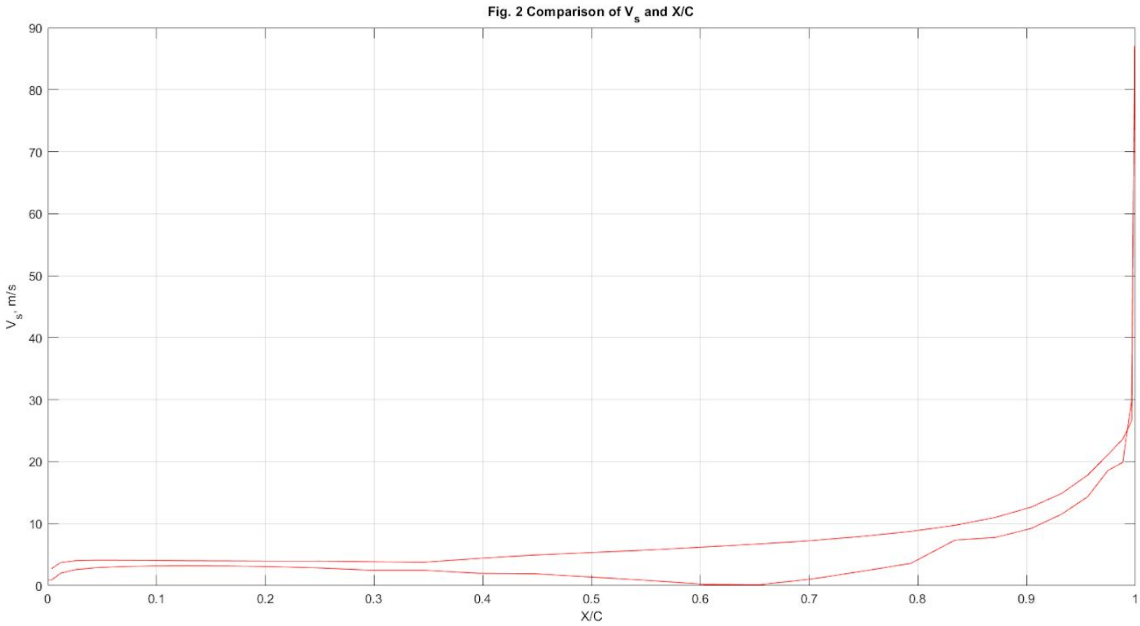


Figure depicting relation between Vs and x/C

In the first graph, on careful observation we observe that there are two curves corresponding to the coefficient of pressure at a given x/C. These are on the upper and lower part of the airflow. Also, both of these are always in the negative region, which clearly shows the lift of the airflow so obtained is zero. So, this satisfies the no circulation condition.

In the second graph, we plotted the tangential velocity against x/C. We observe that the value of Vs tends to singularity which depicts the Kutta condition.

Now we move on to calculating the potential function in the space nearby space. To serve this purpose, we take a region extending from $0<x<1$ and $-0.5<y<0.5$. We took a scale of 0.01 and hence found out the required potential function at a gap of 0.01 in both x and y direction. The calculation of the potential function is done by using the following formula :

$$\phi \;=\; -\frac{1}{2\Pi}\Sigma\,\lambda_j \int_j ln(r)_{ij}\,ds$$

Where, λ_j = strength per unit length of jth panel

r_{ij} = distance of jth panel's control point from the point in consideration

ds = differential panel length

Additionally, we made a 2D array by name Q to store the values of the potential functions. The code to achieve the following is depicted below :

```
scale = 0.01;
Q = zeros(101,101);  %We will use this 2D array to save the values of potential function

%Loop to calculate the potential function at points in the space
for i=0:100
    for j=-50:50
        potential = 0;
        if 0 == inpolygon(scale*i,scale*j,X,Y)
            for k = 1:n
                %Calculating the potential function
                potential= potential + (lambda(k)/(2*pi))*log((sqrt((scale*i-conX(k))^2 + (scale*j-conY(k))^2)))*S(k);
            end
        end
        Q(i+1,j+51) = potential;
    end
end
```

Figure depicting the code snippet to calculate the potential function Q

We use the function i*npolygon()* to make sure that the value of Q is calculated only at the points which lie outside the airfoil. Also, three nested loops were used to calculate the value of Q at different points due to all the panels as shown in the figure above. Following this, we calculated the x and y components of the velocity of the flow for each point in the grid. As the length of each panel is considerably small (due to large number of panels), we just divided the change in the values of the potential function with the change in length in each direction. Also, we did summation instead of integrating owing to the same reasoning as before. The obtained values were stored in arrays named U and V. The code for the same is as shown below :

```
U = zeros(100, 100);        %To store the x component of the streamlines
V = zeros(100, 100);        %To store the y component of the streamlines

for i = 1:100
    for j = -49:50
        if 0 == inpolygon(scale*i,scale*j,X,Y)  %To avoid making streamlines inside the airfoil
            U(i,j+50) = V_inf + (Q(i+1, j+50) - Q(i, j+50))/scale;  %differentiating the
            V(i,j+50) = (Q(i, j+51) - Q(i, j+50))/scale;            %potential function
        else
            U(i,j+50) = 0;      %put the values inside the airflow to zero
            V(i,j+50) = 0;      %to optimize the code
        end
    end
end

figure(4)
hold on;
plot(X,Y,'g','LineWidth', 2);   %plotting the airfoil
for i = 1:100
    for j = -49:50
        if 0 == inpolygon(scale*i,scale*j,X,Y)
            v = [U(i,j+50),V(i,j+50)];
            M = 50*norm(v);
            %To make the arrows of the same size
            %Plotting arrows using the quiver function
            quiver(i*scale, j*scale, (U(i,j+50)/M), (V(i,j+50)/M),'color',[0 0 0]);
        end
    end
end
```

The last and the final part of the code shown as the second part of the figure above concluded the source panel method by plotting arrows of uniform size in the direction of the velocity in the coordinate plane. The size of the arrows was made uniform so as to be better able to depict the variation in the direction of velocity. For the same reason, alpha was taken as zero and the value for V_infinity was set to 2 m/s. The final results so obtained are shown below :

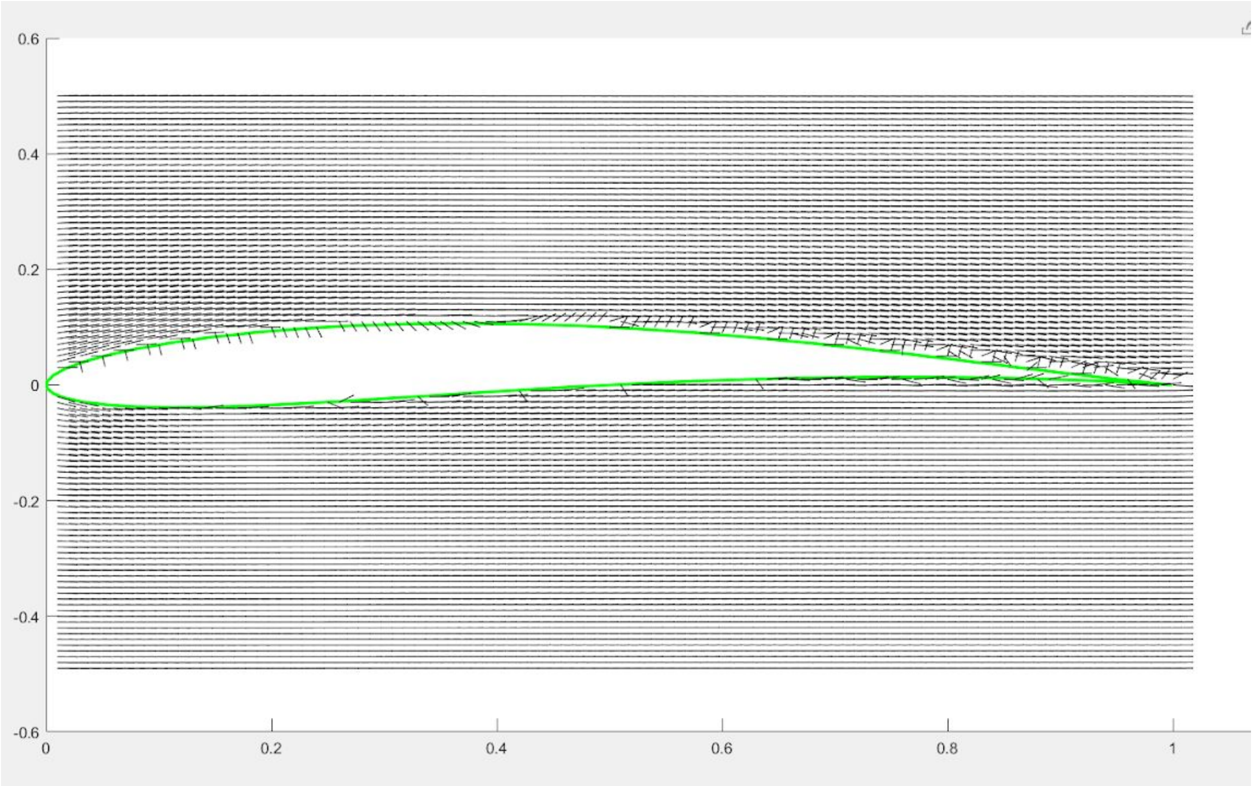


Figure showing the final result

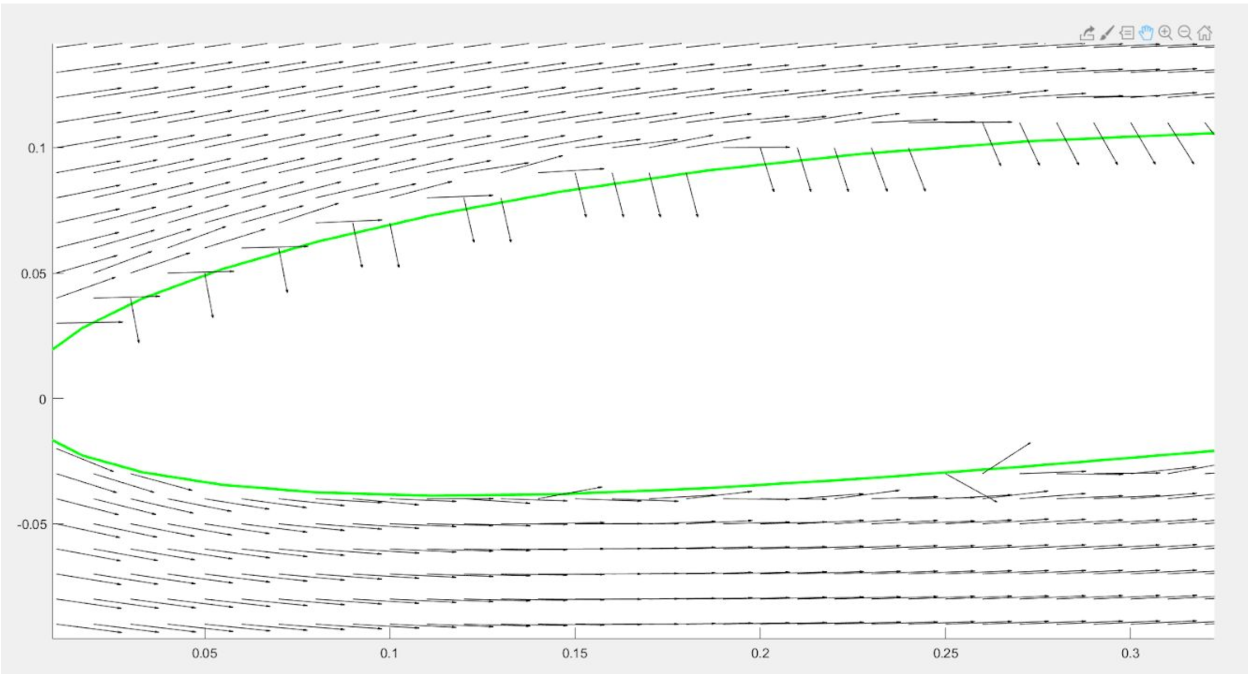


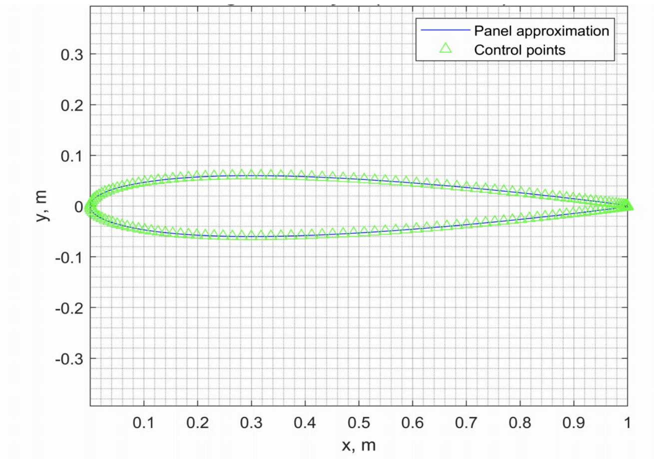
Figure showing a zoomed in version of the final result

Please note that the arrows perpendicular to the airfoil at the boundary (of the airfoil) show that the boundary conditions which were imposed earlier in effect. These should have been eliminated when using the function *inpolygon()* but persisted due to the inaccuracies of the function. The direction of the flow came out to be pretty accurate otherwise.This only shows the efficiency of the source panel method.

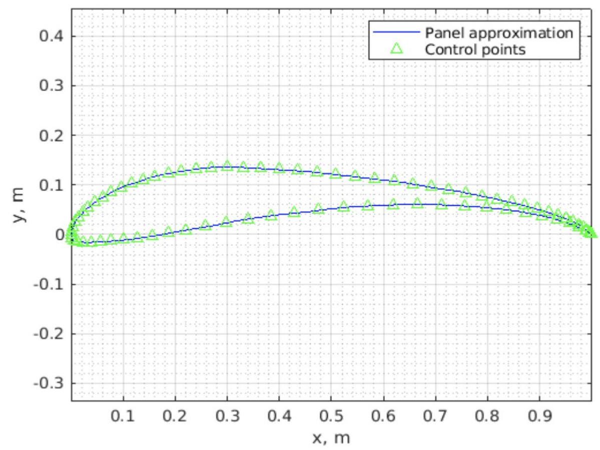
Vortex Panel Method

While analyzing the airfoil using vortex panel method to calculate the potential flow around an arbitrary airfoil, we worked with a matlab code[2] based on the theory from from Kuethe and Chow - "Foundations-of-Aerodynamics".

1.) Replacement of the airfoil by vortex panels of linearly varying strength. We take 160 such vortex panels for further analysis.Since the number of panels used is large, the parameters like pressure distribution and coefficient of lift are close to the exact solution.



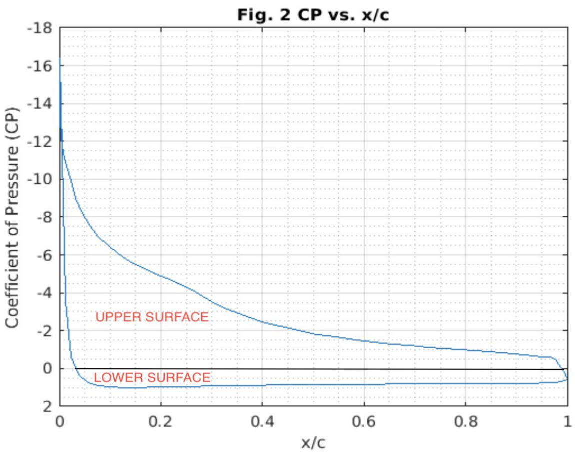
(a) NACA0012 : Symmetric Airfoil

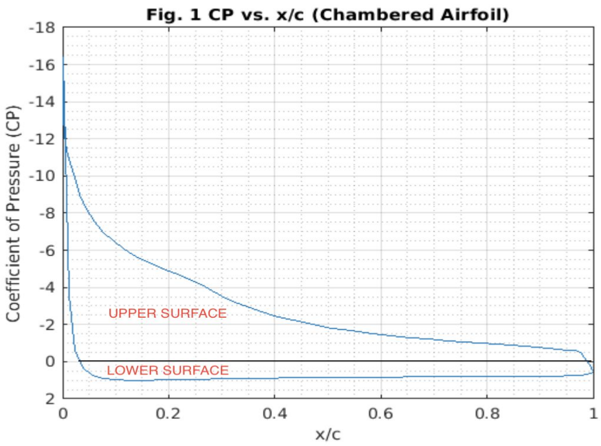


(b) S1223 : Cambered Airfoil

2.) The coefficient of pressure is calculated by the equation $C_p = \frac{p - p_\infty}{1/2 \rho U_\infty^2}$.

We have calculated CP for a fixed value of U_∞ and a fixed value of angle of attack. The value of CP changes with x because different panels have different values of CP. For the upper surface we observe that the value of CP < 0 which implies that $p_{upper} < p_\infty$ whereas for the lower surface of the airfoil, we observe that the value of CP > 0 which implies that $p_{lower} > p_\infty$. These two observations suggest that $p_{lower} > p_{upper}$ for the airfoil, which will produce an upward force viz Lift which is in accordance with the real case.
The absolute value of CP decreases with x on the upper surface as well as the lower surface and for both, the symmetric and the cambered airfoil.





The code given below calculates the value of CP for every panel.

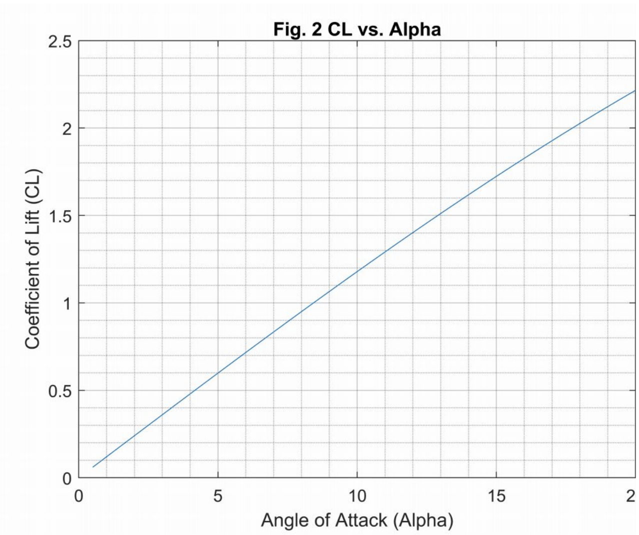
```
for i = 1:M-1
    V(i) = cos(theta(i)-alpha);
    for j = 1:M
        V(i) = V(i) + AT(i,j)*Gama(j);
        CP(i) = 1 - (V(i))^2;
    end
end
```

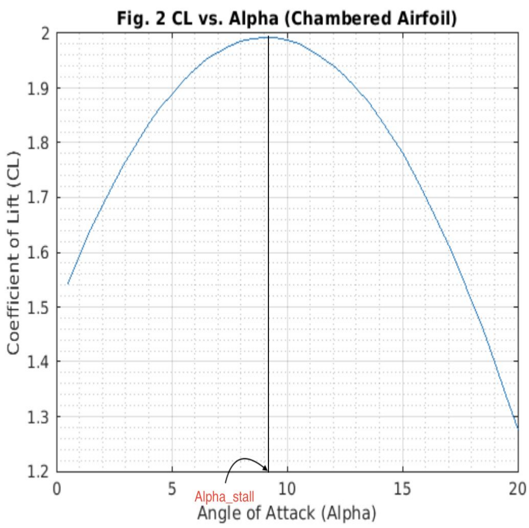
3.) In order to determine the actual force produced by a stress, we must sum all the pressure contributions over the entire surface, thus we must integrate the pressure over the surface.

$CL = \int_0^1 (C_{pl} - C_{pu})\, d(\frac{x}{c})$ where C_{pl} = CP on the lower surface of airfoil, c= chord,

C_{pu} = CP on the upper surface of airfoil,

x = distance from leading edge, along the chord





For some simple examples, the lift coefficient can be determined mathematically. For thin airfoils at subsonic speed, and small angle of attack, the lift coefficient **Cl** is given by: Cl = 2* α. For larger angles, the lift relation is complex[3].

From the above two graphs, we infer :

Symmetric Airfoil : CL = 0 when $\alpha = 0$. Also, $\alpha_{stall} > 20\,(deg)$.

Cambered Airfoil : CL = Finite Value (≈ 1.5) when $\alpha = 0$. Also, $\alpha_{stall} \approx 9\,(deg)$.

```
coeff = trapz(dx,dCP);
c_arr(p)=coeff;                          % > For making array of coefficient of lift
```

(here, coeff is the variable for storing values of CL for a given angle of attack)

4.) We have taken the average of the CP (here, CP is the coefficient of pressure for every panel) and plotted them against the values of CL obtained by changing the angle of attack.

We observe that for the symmetric airfoil, CL decreases with decreasing values of | Avg(CP) | in a nonlinear fashion. Whereas for the cambered airfoil, the value of CL increases with decreases values of | Avg(CP) | uptill a certain value of | Avg(CP) | $\simeq$ 0.45 after which it again decreases with | Avg(CP) | in a
nonlinear fashion.

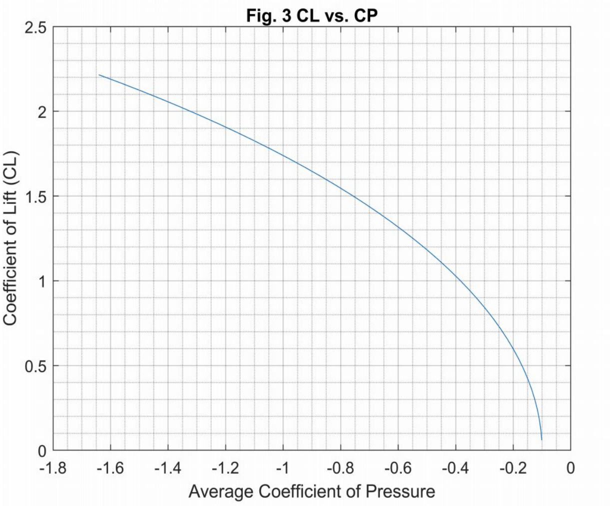


Fig. 3 CL vs. CP

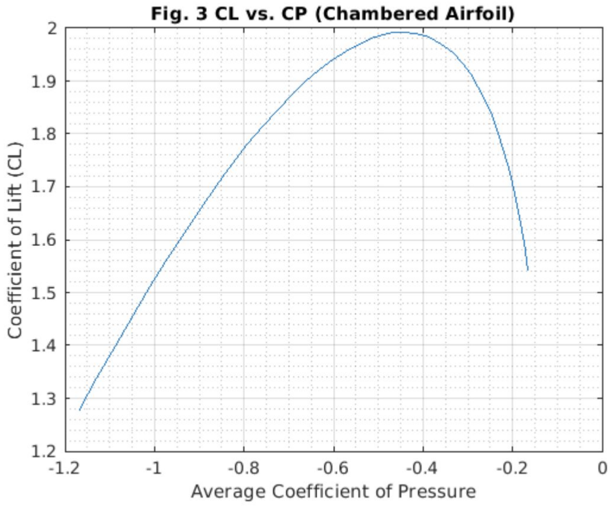


Fig. 3 CL vs. CP (Chambered Airfoil)

```
    % > For making array of Avg(coefficient of pressure)

    cavg_arr(p)=cavg;
    cavg=0;
    for j = 1:160
        cavg=cavg+CP(j);
    end

    cavg=cavg/160;                          % > Calculating average value of CP
    cp_arr(p)=cavg;                         % > array of Avg(CP)
end
```

5.) The graphs of Avg(CP) vs. α for symmetric airfoil and cambered airfoil are almost identical, with | Avg(CP) | increases with increasing α.

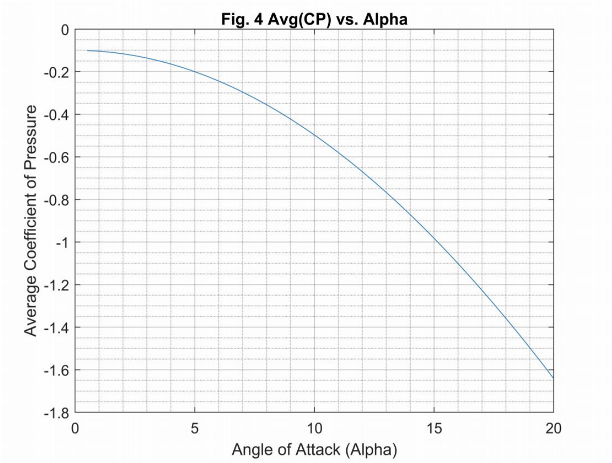


Fig. 4 Avg(CP) vs. Alpha

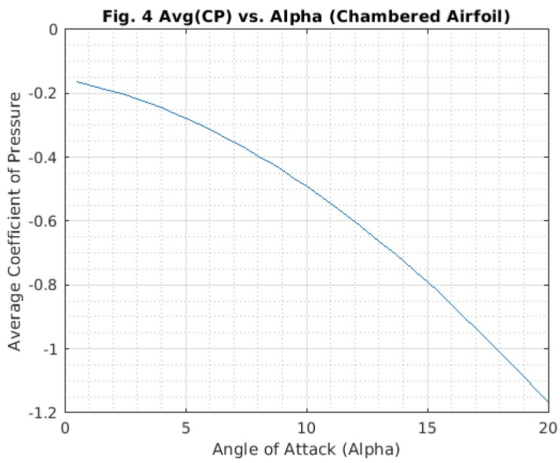
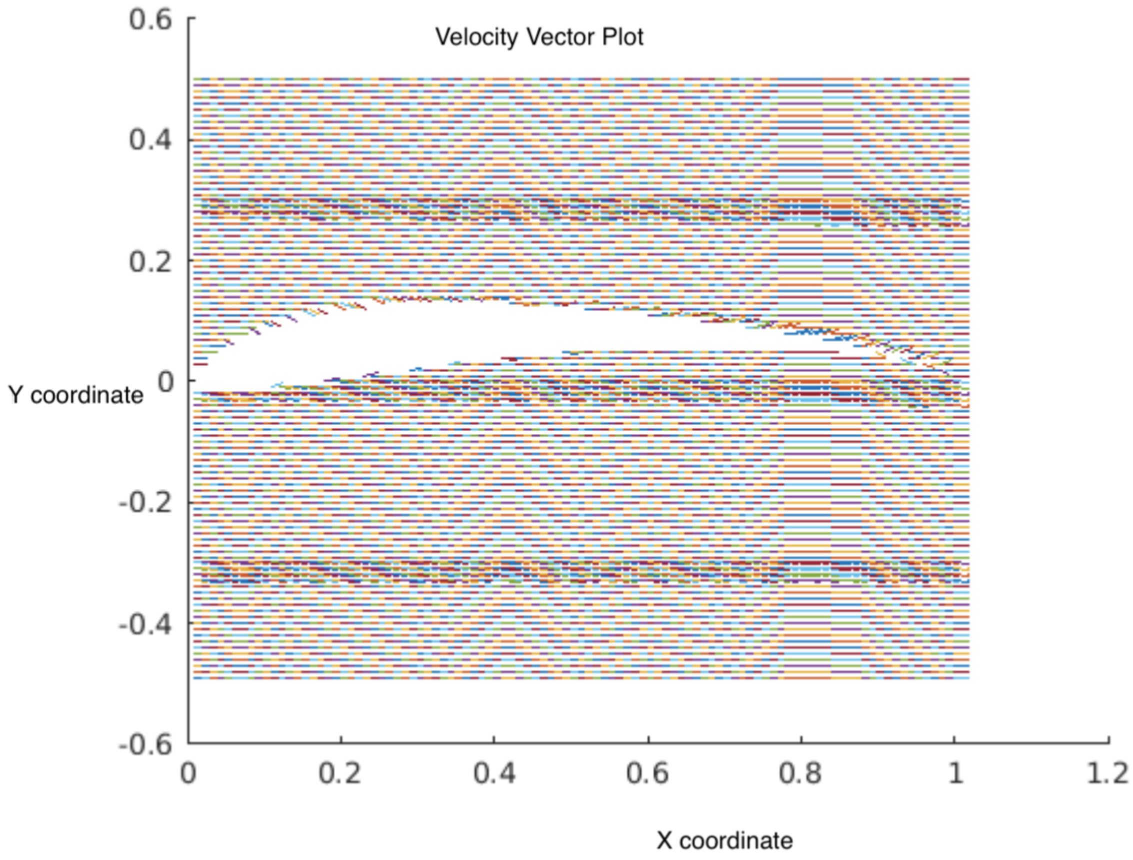


Fig. 4 Avg(CP) vs. Alpha (Chambered Airfoil)

6.) Plot of velocity over the cambered airfoil S1223 by finding the potential function.



This plot represents the direction of velocity vectors over the airfoil. The V_∞ is taken as 2m/s. The velocity plot is obtained by taking the divergence of the potential function.
The formula for finding the potential is as follows :

$$\phi = -\frac{1}{2\Pi}\Sigma\,\gamma_j \int_j \theta_{pj} ds$$

Where, γ_j = strength per unit length of jth panel

θ_{pj} = angle at which the point into consideration, makes with the jth panel

ds = differential panel length

The code given below represents how we obtained the potential function.

```
for i=0:100                      %-> This loop is used to find the potential over the surfa
    for j=-50:50
        potential = 0;
        if 0 == inpolygon(scale*i,scale*j,X,Y)
            for k = 1:n
               thet=atan((j*scale-Y(k)/(i*scale-X(k))));
               potential=potential-(1/(2*pi)).*Gama(k).*ds(k).*thet;  %-> Formula for p
            end
        end
        Q(i+1,j+51) = potential;
    end
end
```

The codes below show how we obtained the velocity by finding the gradient of the potential function.

```
            U(i,j+50) = V_inf + (Q(i+1, j+50) - Q(i, j+50))/scale;
            V(i,j+50) = (Q(i, j+51) - Q(i, j+50))/scale;
        else
            U(i,j+50) = 0;                                  %-> cancelling the velo
            V(i,j+50) = 0;
        end
    end
end
```

```
U = zeros(100, 100);                                        %-> defining the horizo
V = zeros(100, 100);

for i = 1:100
    for j = -49:50
        if 0 == inpolygon(scale*i,scale*j,X,Y)              %-> condition to obtain
```


REFERENCES

[a]Simple analytic equation for airfoil shape description by David Ziemkiewic
https://www.researchgate.net/publication/312222678_Simple_analytic_equation_for_airfoil_shape_description

[1] Source Panel Method applied to Flow around Cylinder
https://in.mathworks.com/matlabcentral/fileexchange/56156-source-panel-method-applied-to-flow-around-cylinder

[2] Vortex Panel Method
https://github.com/dpkprm/Vortex-Panel-Method

[3] Inclination effects on Lift (NASA)
https://www.grc.nasa.gov/www/k-12/airplane/incline.html