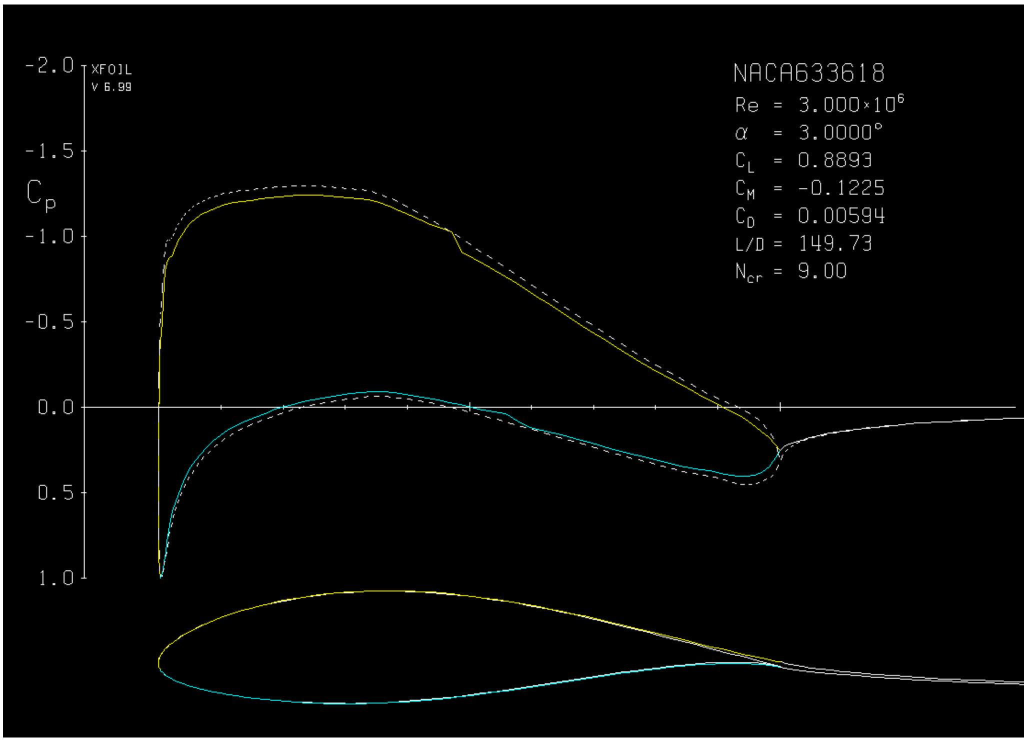
Siemens Gamesa Renewable Energy

Offshore Blade Design
Engineering Programming Challenge

Aerodynamic Performance of The NACA63(3)-618 Airfoil

Patrick M. Bodine-Ellison



May 8, 2020

Contents

List of Figures

1 Introduction and Assumptions

In order to more efficiently analyze the aerodynamic properties of the NACA63(3)-618 airfoil profile MATLAB code was developed to interface with the software xFoil to pull the polars, pressure profiles, and boundary layer conditions at the trailing edge at varying angles of attack and Reynolds numbers. In all analysis herein the following assumptions are made:

1. For the purposes of boundary layer transition the N_{crit} value is fixed at 9.00
2. Flow mach number is fixed at 0.00
3. All span-wise effects are ignored and all coefficients are 2-dimensional
4. Chord is equal to 1

2 The NACA 63(3)-618 airfoil

An airfoil is a shape that, when moving through a fluid, produces an aerodynamic lift force. There are an innumerable number of possible airfoil profiles and each one of these profiles will have different properties at different angles of attack and different air speeds. In order to bring some standardization to these airfoil shapes the National Advisory Committee for Aeronautics (NACA) developed a list of well tested airfoil shapes. This list has since expanded and now many airfoils are designated with "NACA" followed by a 4-6 digit code that designates the specific airfoil profile.

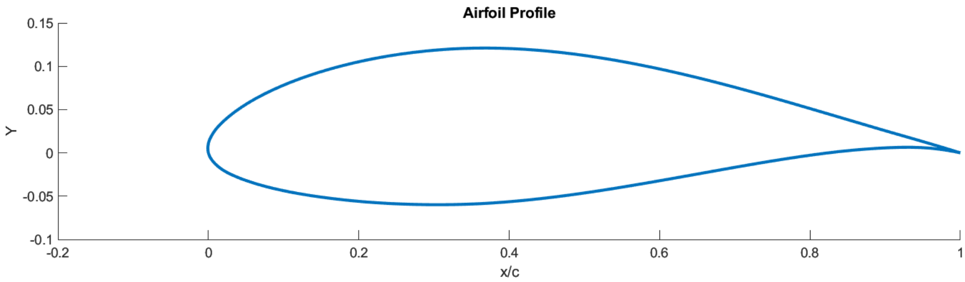


Figure 1: NACA 63(3)-618 airfoil profile

The specific airfoil profile discussed herein is the NACA 63(3)-618, shown in **Figure: 1**. This airfoil has a maximum thickness of 18% of chord length at 34.7% of chord and a max camber of 3.3% at 50% chord length. The data file used to calculate the aerodynamic properties of this airfoil was pulled from the UIUC Airfoil Coordinates Database on March 20th, 2020.

3 Airfoil Polars

Airfoil polar show how the section moment coefficient and section lift coefficient change with angle of attack and how the section list coefficient changes with the section drag coefficient. These coefficients are important as they tell us thing about the properties of the lift on the airfoil under different conditions. The section moment coefficient helps to determine the pitching moment direction on the airfoil and the section lift coefficient helps to determine the amount of lift gained. Similarly, we can look at the drag polar, a comparison between the lift coefficient and the drag coefficient, to get an approximation for the aerodynamic efficiency of the airfoil profile.

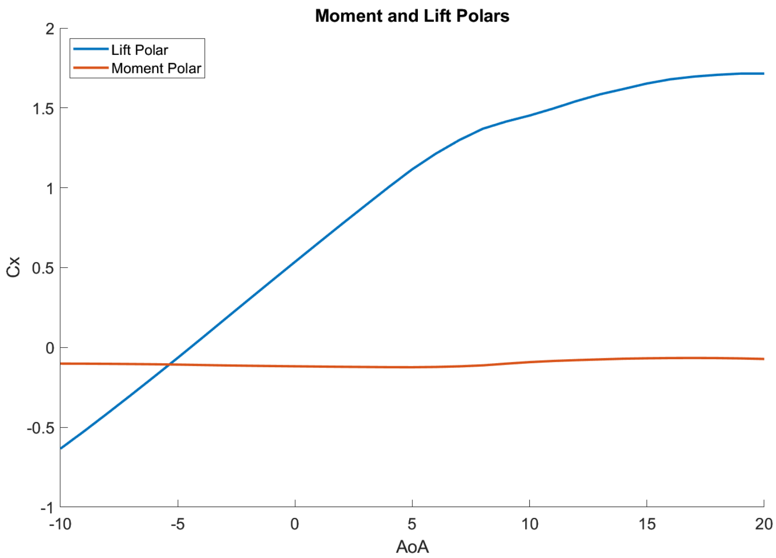


Figure 2: Moment and lift polars at Re: 3e6

For the NACA 63(3)-618 specifically, the moment and lift polars show little change as Reynolds number ranges from 3e6 to 15e6. The most noticeable differences are apparent in the higher angles of attack. At an angle of attack of 20 the coefficient of lift is 1.7 at a Reynolds number of 3e6 and 1.9 at a Reynolds number of 15e6. The moment coefficients have even less of a change, changing from -0.072 to -0.068 in the same range.

The trends of these polars, while relatively unchanging with Reynolds number, still give a significant amount of information about how this airfoil functions at varying angles of attack. The lift polar shows a steady increase in coefficient of lift until an angle of attack of 8-9 where it starts to level out. This shows that the greatest gains in lift per degree of AoA are to be had in the -10 to 8 degree AoA range. Additionally, coefficient of lift is negative for any angles of attack below -4 degrees, representing negative lift. It then becomes apparent that, to maintain significant and efficient lift, angles of attack above -4 and below 8 should be maintained.

The moment tells a different story, during all angles of attack it remains relatively flat and consistently negative. This represents a constant and slight downward pitching moment on the airfoil.

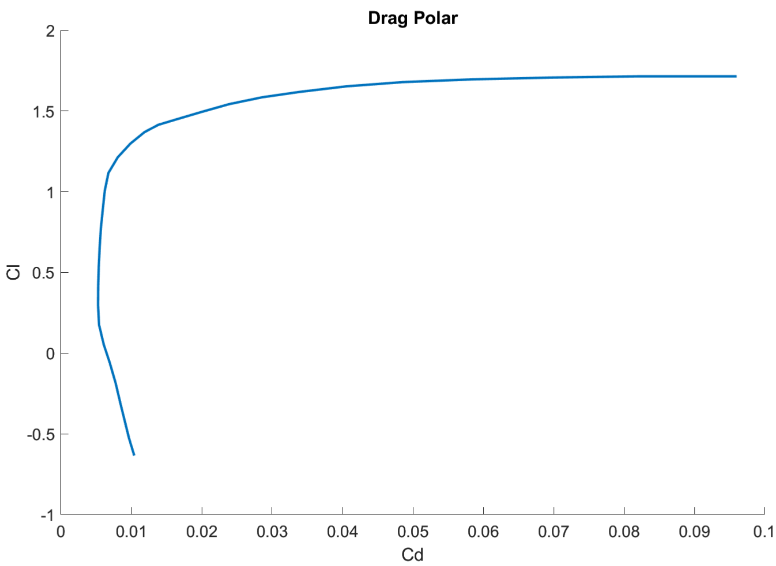


Figure 3: Drag polar at Re: 3e6

The drag polar is graphed differently to that of the moment and lift polars. This polar graphs the coefficient of drag against the coefficient of lift with the primary purpose of determining aerodynamic efficiency. High ratios of lift coefficient to drag coefficient represent highly efficient airfoils suited well for gliders.

For the NACA 63(3)-618 the drag polar shows a sharp increase in coefficient of lift at a coefficient of drag around 0.01 and then a flat coefficient of lift, staying around 1.5-2.0 with higher Reynolds numbers yielding higher lift coefficients. It is apparent from this drag polar the aerodynamic efficient decreases at lift in increased past its initial high point around a drag coefficient of 0.015. The returns on lift after this point correspond with much more significant increases in coefficient of drag and result in diminishing returns with regards to aerodynamic efficiency.

Using these two polar graphs together it can be seen that angles of attack between 4-9 degrees are optimal for high amounts of lift and high aerodynamic efficiency as these angles of attack correspond to high lift and low coefficient of drag.

4 Pressure Distributions

Airfoil pressure profiles are vital for understanding where along the chord of an airfoil lift is being generated. They show a distribution of pressure along the top and the bottom of the blade and, ideally, pressure is higher on the bottom than the top, thus generating the sought after lift.

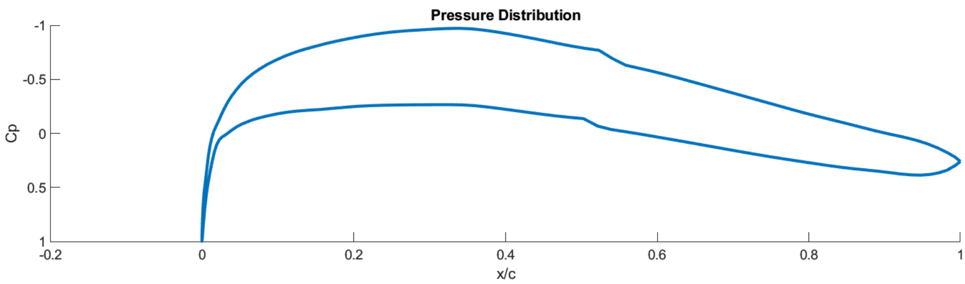


Figure 4: Pressure distribution at AoA: 0 and Re: 3e6

For the NACA 63(3)-618 foil pressure profiles change very little with changes in Reynolds number but show drastic changes with a change in angle of attack. At an angle of attack of 0°and a Reynolds number of 3e6 the pressure distribution appears quite narrow, a sign that the pressures above and below the airfoil are similar and are not generating a much lift. Conversely, at an angle of attack of 12°the difference between the coefficient of pressure and the pressure and suction side of the airfoil is much more significant and will result in significantly more lift than lower angles of attack.

The consequences of these pressure distributions are shown in the lift polar of this airfoil, angles of attack with a higher coefficient of lift in the lift polar showing a greater difference between coefficient of pressure on the pressure and suction sides of the airfoil in the pressure distribution.

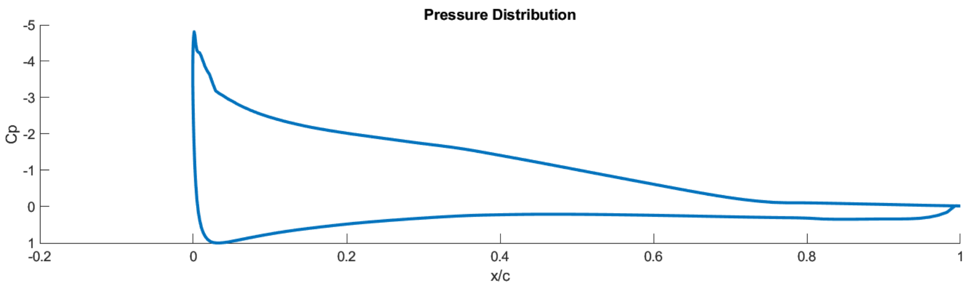


Figure 5: Pressure distribution at AoA: 12 and Re: 3e6

5 Boundary Layer Properties at Trailing Edge

In aerodynamics, it doesn't just matter what air is doing when it is running along the surface of the airfoil, it matter what happens when is leaves too. Depending on the conditions of the air and the geometry of the airfoil the air leaving the blade surface can have detrimental effects on aerodynamic performance, create unwanted noise, or damage the aerodynamic stability of the airfoil profile. In order to prevent these detrimental effects it is important to understand the properties of the air at the trailing edge of the airfoil.

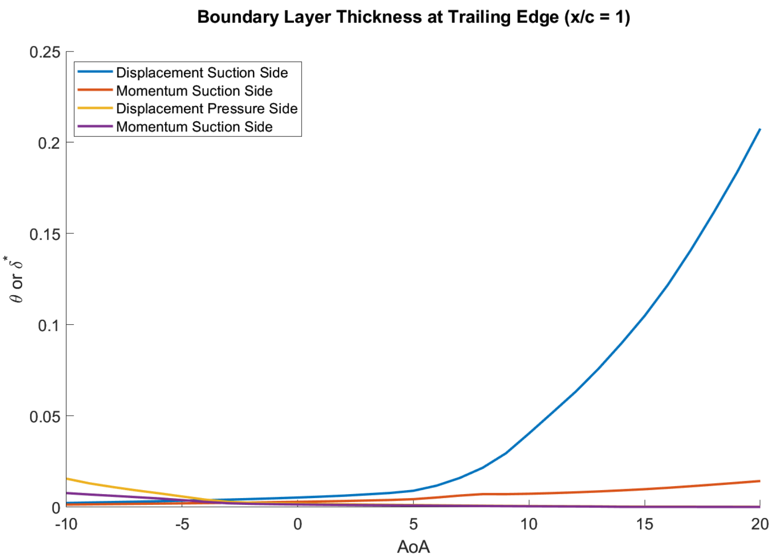


Figure 6: Displacement and Momentum Thickness at Re: 3e6

The displacement thickness, which represents an effective of thickening of the airfoil caused by frictional forces at the airfoil surface, and the momentum thickness, a quantification of the amount of viscous activity within a flow, both give a significant amount of information on the aerodynamic efficiency of the airfoil and the nature of the flow around it.

In the NACA 63(3)-618 airfoil the displacement and momentum thicknesses are both greatest on the suction side rather than the pressure side and the displacement thickness has a much higher value, which is to be expected. As displacement thickness increases a decrease in aerodynamic efficiency and increase in the frictional forces of the airfoil surface is seen. This is logically consistent with the ideal angles of attack seen in the previous section.

Additionally, Reynolds number has a more significant effect on this quantities than seen in earlier sections. At a Reynolds number of 3e6 and AoA of 20°displacement thickness is at 0.21 while at the same AoA but a Reynolds number of 15e6 displacement thickness is 0.16.

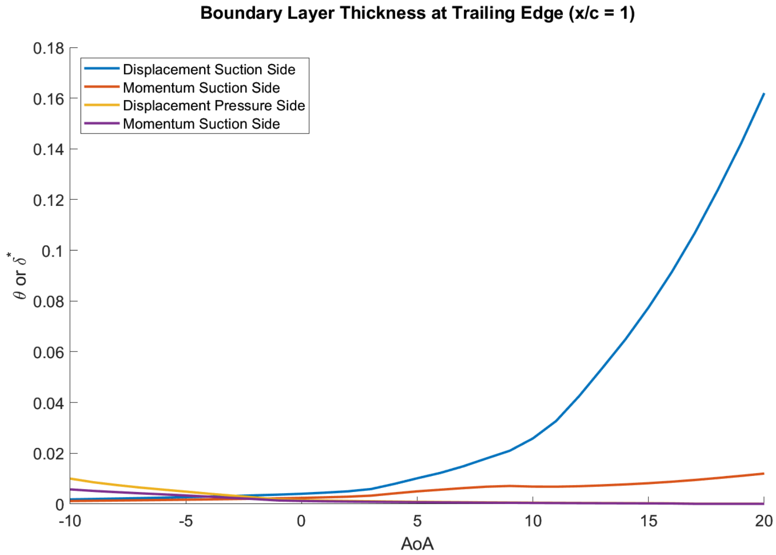


Figure 7: Displacement and Momentum Thickness at Re: 15e6

Shape factor is a ratio between momentum and displacement thickness and can be used to determine the presence and strength of an adverse pressure gradient. This information can, in turn, be used to determine the laminar or turbulent nature of flow. Looking at this factor at the trailing edge can inform about the nature of the flow leaving the surface of the airfoil.

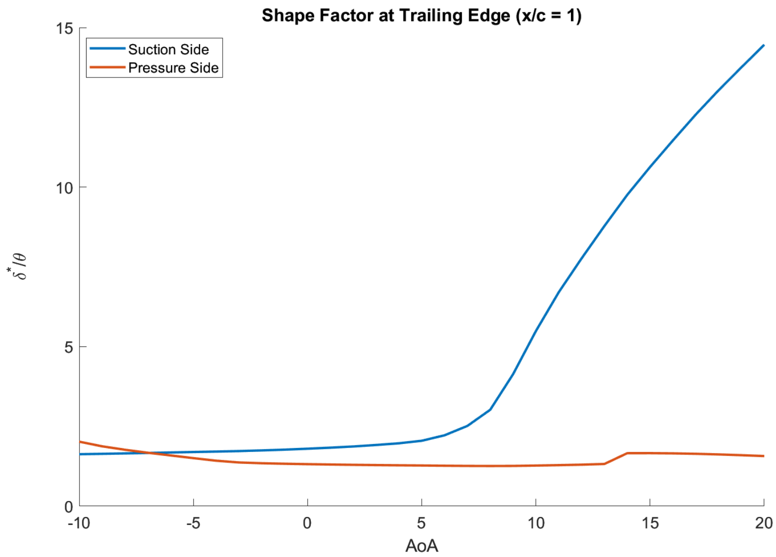


Figure 8: Shape factor at Re: 3e6

For the NACA 63(3)-618 airfoil the shape factor at the trailing edge both decreases slightly with an increase in Reynolds number and sees a sharp increase at angles of attack around 6°-10°. This follows logically from the pressure distributions in earlier sections as pressure recovery must occur and with higher angles of attack there is a greater amount of suction on the suction side of the airfoil. The higher values of shape factor at angles of attack about 10°create strong adverse pressure gradients which may result in flow separation and stalling of the airfoil, resulting in no or very little lift.

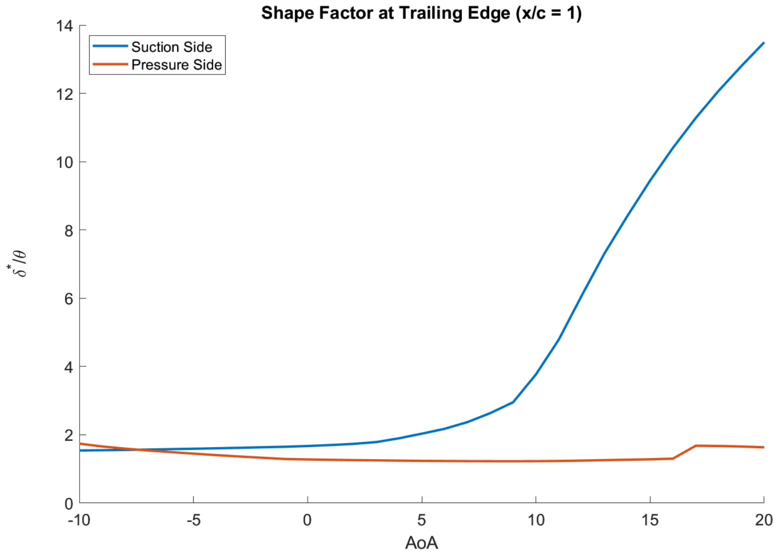


Figure 9: Shape factor at Re: 15e6

6 Conclusion

Ultimately, the NACA 63(3)-618 airfoil is a relatively stable airfoil profile that performs more efficient at angles of attack between 4°to 9°. Additionally, as flow Reynolds number increases over this airfoil there is little change in lift with a slight increase in the probability of stall conditions. Thus, this airfoil should be used in cases where fluid flow Reynolds number is likely to be low and angle of attack is not expected to increase above 9°.

Appendix A: Supplemental Graphs

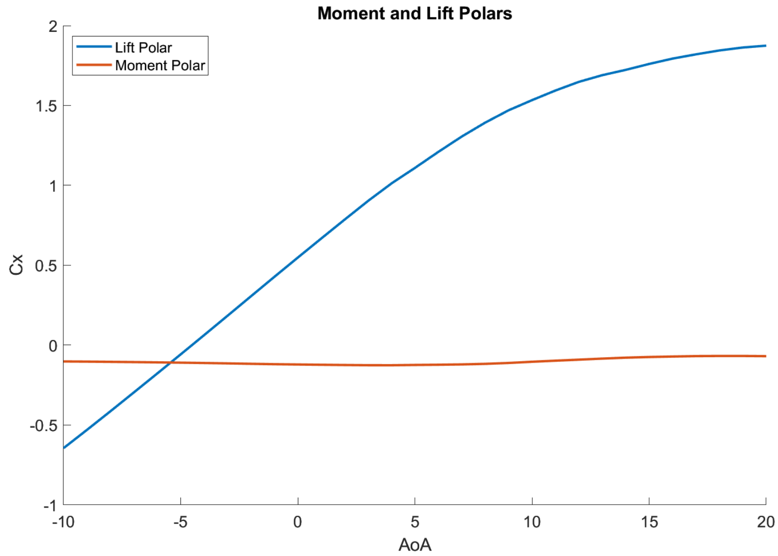


Figure 10: Moment and lift polars at Re: 10e6

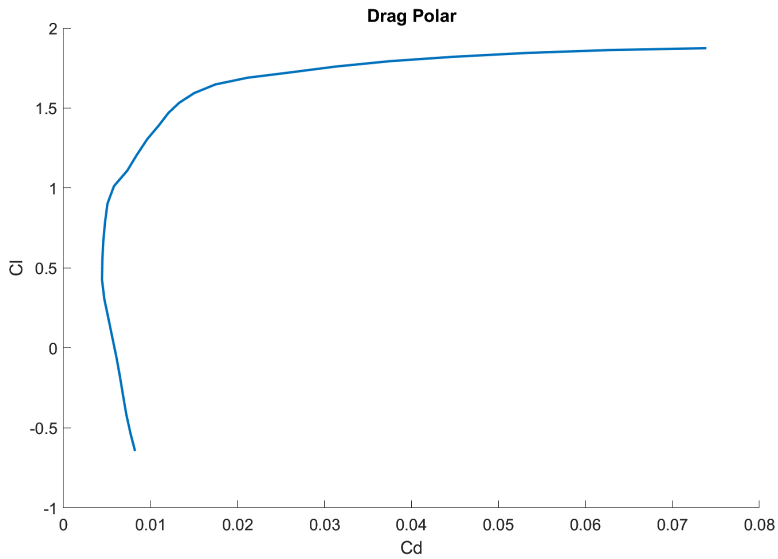


Figure 11: Drag polar at Re: 10e6

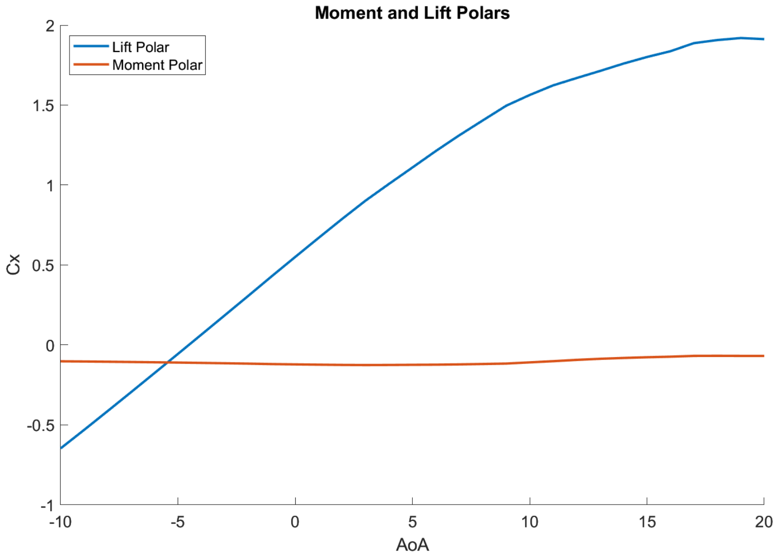


Figure 12: Moment and lift polars at Re: 15e6

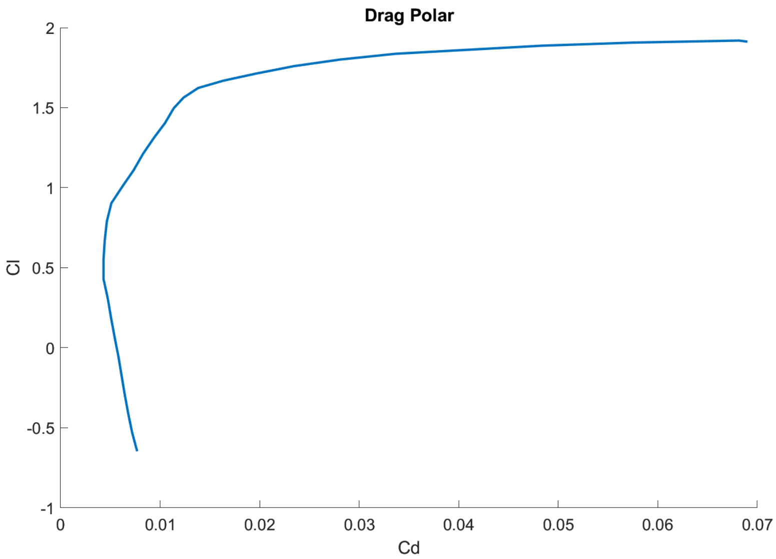


Figure 13: Drag polar at Re: 15e6

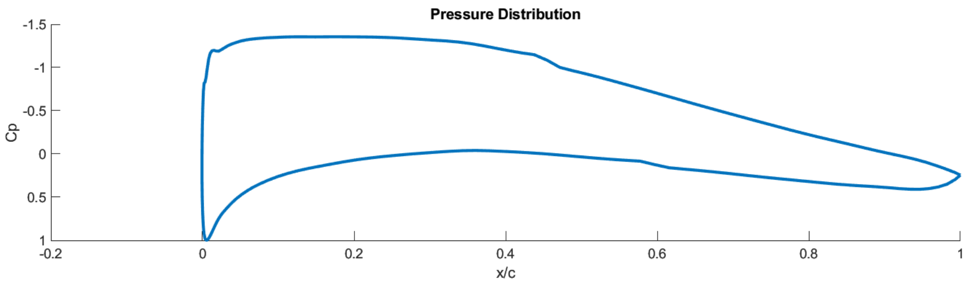


Figure 14: Pressure distribution at AoA: 4 and Re: 3e6

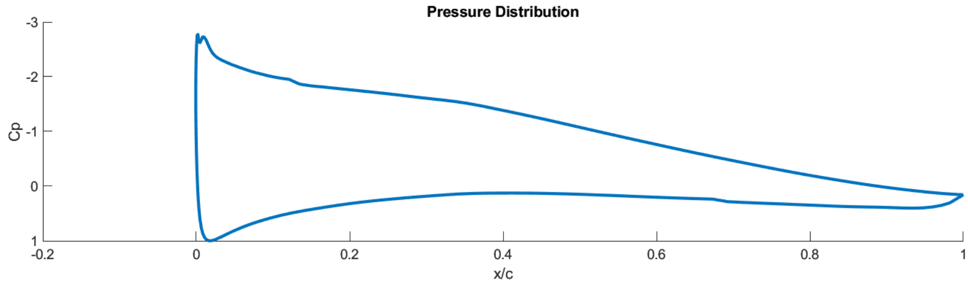


Figure 15: Pressure distribution at AoA: 8 and Re: 3e6

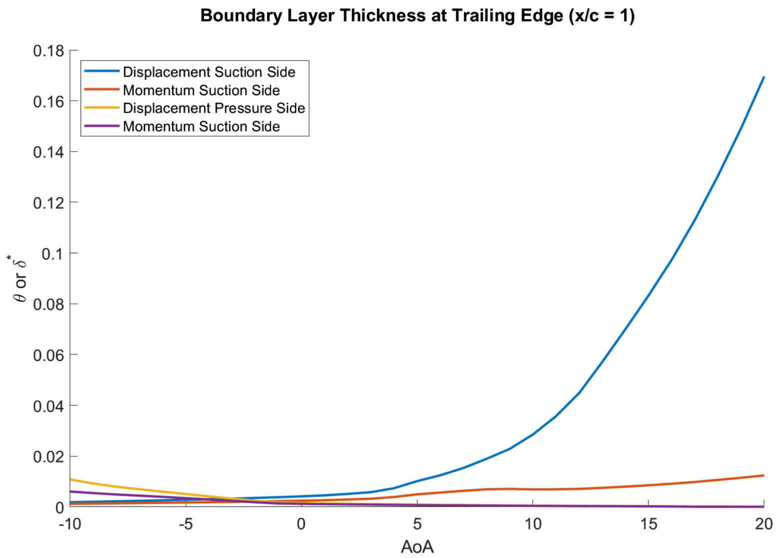


Figure 16: Displacement and Momentum Thickness at Re: 11e6

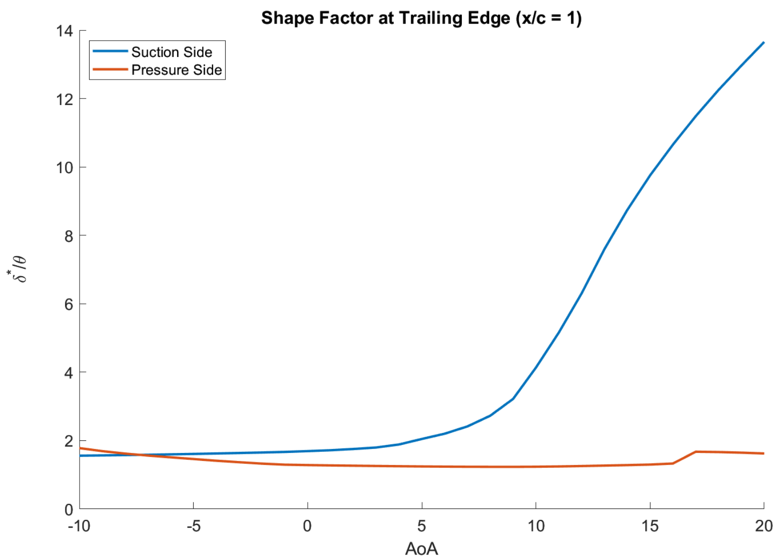


Figure 17: Shape factor at Re: 11e6

Appendix B: MATLAB Code

```
classdef XFoil_Run_Engineering_Programming_Challenge < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        XFoilRunEngineeringProgrammingChallengeUIFigure  matlab.ui.Figure
        TabGroup                     matlab.ui.container.TabGroup
        MomentandLiftPolarsTab       matlab.ui.container.Tab
        MnLPolars                    matlab.ui.control.UIAxes
        DragPolarsTab                matlab.ui.container.Tab
        DragPolars                   matlab.ui.control.UIAxes
        PressureTab                  matlab.ui.container.Tab
        PressureDist                 matlab.ui.control.UIAxes
        AirfoilProfile               matlab.ui.control.UIAxes
        AoASliderLabel               matlab.ui.control.Label
        AoASlider                    matlab.ui.control.Slider
        BoundaryLayerThicknessTab    matlab.ui.container.Tab
        BLThick                      matlab.ui.control.UIAxes
        ShapeFactorTab               matlab.ui.container.Tab
        ShapeFactor                  matlab.ui.control.UIAxes
        RunXFoilMayTakeTimeButton    matlab.ui.control.Button
        RynoldsNumber106SliderLabel  matlab.ui.control.Label
        RynoldsNumber106Slider       matlab.ui.control.Slider
    end


    properties (Access = private)
        Rey % Reynolds number value
        ReyInd %Reynolds number index
        DPol %Polar data
        dataAoA %Angle of attack data for polars
        dataCL %Lift coefficient data for polars
        dataCD %Drag coefficient data for polars
        dataCM %Moment coefficient data for polars
        dataTop_Xtr %
        dataBot_Xtr %
        dataX %
        dataY %
        dataCp %
        xfoilCommands %Polars xfoil commands file
        xfoilCommandsP %Pressure distribution xfoil commands file
        xfoilCommandsB %Boundary layers Xfoil commands file
        dataPressure %
        aeroPath %
        aeroFile %
        foilName %
        AoAValue %

        dataHTop %Shape factor data, trailing edge, top
        dataThetaTop %Moment thickness data, trailing edge top
        dataDStarTop %Displacement thickness data, trailing edge, top
        dataHBot %Shape factor data, trailing edge, bot
        dataThetaBot %Moment thickness data, trailing edge bot
        dataDStarBot %Displacement thickness data, trailing edge, bot
        dataXBound %Xpos data for Boundary layer parameters
    end


    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: RunXFoilMayTakeTimeButton
        function RunXFoilMayTakeTimeButtonPushed(app, event)
            %run XFoil and pull data
            %Get all that stuff out of here



clearvars -except app;clc;close all;
```

```matlab
app.aeroFile = 'NACA633618.txt';
app.aeroPath = '.\';



for k = 3:15



delete '.\dataPolars.txt'
delete '.\dataPressure.txt'
delete '.\boundParams.txt'
delete '.\xfoilCommands'
delete '.\xfoilCommandsP.txt'
delete '.\xfoilCommandsB.txt'
delete '.\xfoilout.txt'
delete '.\xfoiloutP.txt'



%Open the text file for writing commands to xFoil
app.xfoilCommands = fopen('.\xfoilCommands.txt','w')


%Write command to open PLOP menu to xfoil command text
    fprintf(app.xfoilCommands,'PLOP\n')

%Write command to turn off graphic plotting to xfoil command text
    fprintf(app.xfoilCommands,'G F\n\n\n')

%Write command to load aerofoil profile into xFoil to xFoil command text
fprintf(app.xfoilCommands,'LOAD %s%s\n',app.aeroPath,app.aeroFile)

%Write command to name aerofoil profile to xFoil command text
    fprintf(app.xfoilCommands,'NACA%s\n',app.foilName)

%Write command to regnerate paneling to xFoil command text
fprintf(app.xfoilCommands,'PANE\n')

%Write command to open oper menu to xFoil command text
fprintf(app.xfoilCommands,'OPER\n')

%Write command to change to visc mode to xFoil command text
fprintf(app.xfoilCommands,'VISC %i\n',k*1e6)

%Write command to increase iteration count to xFoil command text
fprintf(app.xfoilCommands,'ITER 250\n')

%Write command to open point accumulation to xFoil command text
fprintf(app.xfoilCommands,'PACC\n')

%Write command to define plar data file name to xFoil command text
fprintf(app.xfoilCommands,'dataPolars.txt\n\n')

%Write command to run aoa sequence from -10 to 30 to xFoil command text
fprintf(app.xfoilCommands,'aseq -10 -9 1\n')
fprintf(app.xfoilCommands,'aseq -8 -7 1\n')
fprintf(app.xfoilCommands,'aseq -6 -5 1\n')
fprintf(app.xfoilCommands,'aseq -4 -3 1\n')
fprintf(app.xfoilCommands,'aseq -2 -1 1\n')
fprintf(app.xfoilCommands,'aseq 0 1 1\n')
fprintf(app.xfoilCommands,'aseq 2 3 1\n')
fprintf(app.xfoilCommands,'aseq 4 5 1\n')
fprintf(app.xfoilCommands,'aseq 6 7 1\n')
fprintf(app.xfoilCommands,'aseq 8 9 1\n')
fprintf(app.xfoilCommands,'aseq 10 11 1\n')
fprintf(app.xfoilCommands,'aseq 12 13 1\n')
fprintf(app.xfoilCommands,'aseq 14 15 1\n')
fprintf(app.xfoilCommands,'aseq 16 17 1\n')
fprintf(app.xfoilCommands,'aseq 18 20 1\n')
```

```matlab
%Write command to close point accumulation to xFoil command text
fprintf(app.xfoilCommands,'PACC\n')

%Close xFoil command text
fclose(app.xfoilCommands)

%Tell cmd to run xfoil commands and write output to file
fRunText = sprintf('D: && cd %s && xfoil.exe < xfoilCommands.txt > xfoilout.txt && ...
    exit',pwd,pwd,pwd)
system(fRunText,'-echo')

%Intake and formatting for polar data file into MATLAB
dataPolarsText = tdfread('.\dataPolars.txt');
dataPolarsText = dataPolarsText.XFOIL_Version_60x2E99(11:end,:);
A = size(dataPolarsText,1);
for i = 1:A
    dataPolars(i,:) = cell2mat(textscan(dataPolarsText(i,:),'%f'));
end
%Submit to function output
app.DPol = dataPolars;


%Pulling in pressure data
%This runs a pressure profile and saves it for 40 different AoA values
%Pressure profiles are saved to dataPressure which is a 3 dimensional array
%dataPressure(:,:,1) is AoA -10 while dataPressure(:,:,41) is 30
dataPressure = zeros(160,3,40);
for j = -10:20

%Open xfoil command file
    app.xfoilCommandsP = fopen('.\xfoilCommandsP.txt','w')

%Write command to open PLOP menu to xfoil command text
    fprintf(app.xfoilCommandsP,'PLOP\n')

%Write command to turn off graphic plotting to xfoil command text
    fprintf(app.xfoilCommandsP,'G F\n\n\n')

%Write command to load airfoil file to xfoil command text
    fprintf(app.xfoilCommandsP,'LOAD %s%s\n',app.aeroPath,app.aeroFile)

%Write command to name aerofoil profile to xFoil command text
    fprintf(app.xfoilCommandsP,'NACA%s\n',app.foilName)

%Write command to regnerate paneling to xFoil command text
    fprintf(app.xfoilCommandsP,'PANE\n')

%Write command to open oper menu to xFoil command text
    fprintf(app.xfoilCommandsP,'OPER\n')

%Write command to change to visc mode to xFoil command text
    fprintf(app.xfoilCommandsP,'VISC %i\n',k*1e6)

%Write command to increase iteration count to xFoil command text
    fprintf(app.xfoilCommandsP,'ITER 250\n')

%Write command to create pressure distribution at AoA to xFoil command text
    fprintf(app.xfoilCommandsP,'alfa %f\n',j)

%Write command to save pressure distribution data to xFoil command text
    fprintf(app.xfoilCommandsP,'CPWR\n')

%Write command to speicfy file name to xFoil command text
    fprintf(app.xfoilCommandsP,'dataPressure.txt\n')

%Close xFoil command text
    fclose(app.xfoilCommandsP);

%Run xFoil command text and write output to xfoiloutP.txt
    fRunText = sprintf('D: && cd %s && xfoil.exe < xfoilCommandsP.txt > ...
        xfoiloutP.txt && exit',pwd,pwd,pwd);
    system(fRunText,'-echo')
```

```
%Intake pressure profile info into matlab array
    dataPressureText = tdfread('.\dataPressure.txt');
    dataPressureText = dataPressureText.('NACA')(3:end,:);
    A = size(dataPressureText,1);
%     temp = zeros();
    for i = 1:A
        temp(i,:) = cell2mat(textscan(dataPressureText(i,:),'%f'));
    end
    app.dataPressure(:,:,j+11) = temp;
end


%begin boundary layer properties
for u = -10:20

%Open boundary layers XFoil Commands file
app.xfoilCommandsB = fopen('.\xfoilCommandsB.txt','w')

%Write command to open PLOP menu to xfoil command text
fprintf(app.xfoilCommandsB,'PLOP\n')

%Write command to turn off graphic plotting to xfoil command text
fprintf(app.xfoilCommandsB,'G F\n\n\n')

%Load aerofoil profile into xfoil and name it
fprintf(app.xfoilCommandsB,'LOAD %s%s\n',app.aeroPath,app.aeroFile)
fprintf(app.xfoilCommandsB,'NACA%s\n',app.foilName)

%Rebuild panneling for the aerofoil profile
fprintf(app.xfoilCommandsB,'PANE\n')

%Open OPER menu and convert to visc mode with reynolds number
fprintf(app.xfoilCommandsB,'OPER\n')
fprintf(app.xfoilCommandsB,'VISC %i\n',k*1e6)


%Set number of iterations to 250
fprintf(app.xfoilCommandsB,'ITER 250\n')

%Run fluid flow simulation at given angle of attack
fprintf(app.xfoilCommandsB,'alfa %f\n',u)

%Run shape parameter at current angle of attack and dump data to text file
fprintf(app.xfoilCommandsB,'H\n')
fprintf(app.xfoilCommandsB,'DUMP boundParams.txt\n')

%Close xfoil command file
fclose(app.xfoilCommandsB)

%Run xfoil command file using CMD
fRunText = sprintf('D: && cd %s && xfoil.exe < xfoilCommandsB.txt > xfoiloutB.txt && ...
    exit',pwd,pwd,pwd);
system(fRunText,'-echo')

%Load dumped text files into matlab and format them
dataBoundText = tdfread('.\boundParams.txt')
dataBoundText = dataBoundText.x0x23_s_x_y_Ue0x2FVinf_Dstar_Theta_Cf_H(:,:);

%Pull values at trailing edge
app.dataXBound = dataBoundText(:,2);
[xind, ¬] = find(app.dataXBound==1);
topxind = xind(1);
botxind = xind(2);
app.dataHTop(u+11,k) = dataBoundText(topxind,8);
app.dataDStarTop(u+11,k) = dataBoundText(topxind,5);
app.dataThetaTop(u+11,k) = dataBoundText(topxind,6);
app.dataHBot(u+11,k) = dataBoundText(botxind,8);
app.dataDStarBot(u+11,k) = dataBoundText(botxind,5);
app.dataThetaBot(u+11,k) = dataBoundText(botxind,6);

end
```

```
%Sepparate inidividual columns of dataPolars
app.dataAoA(:,k) = app.DPol(:,1);
app.dataCL(:,k) = app.DPol(:,2);
app.dataCD(:,k) = app.DPol(:,3);
app.dataCM(:,k) = app.DPol(:,5);
app.dataTop_Xtr(:,k) = app.DPol(:,6);
app.dataBot_Xtr(:,k) = app.DPol(:,7);

%Sepparate individual columns of dataPressure
app.dataX(:,:,k) = app.dataPressure(:,1,:);
app.dataY(:,:,k) = app.dataPressure(:,2,:);
app.dataCp(:,:,k) = app.dataPressure(:,3,:);


end

app.Rey = app.RynoldsNumber106Slider.Value;
app.ReyInd = round(app.Rey);
plot(app.MnLPolars,app.dataAoA(:,3),app.dataCL(:,app.ReyInd),'LineWidth',2)
hold(app.MnLPolars,'on');
plot(app.MnLPolars,app.dataAoA(:,3),app.dataCM(:,app.ReyInd),'LineWidth',2)
hold(app.MnLPolars,'off');
legend(app.MnLPolars,'Lift Polar','Moment Polar','Location','northwest')


plot(app.DragPolars,app.dataCD(:,app.ReyInd),app.dataCL(:,app.ReyInd),'LineWidth',2)

app.AoAValue = round(app.AoASlider.Value) + 11;
plot(app.PressureDist,app.dataX(:,app.AoAValue,app.ReyInd),app.dataCp(:,...
app.AoAValue,app.ReyInd),'LineWidth',2)
set(app.PressureDist,'YDir','Reverse')
plot(app.AirfoilProfile,app.dataX(:,app.AoAValue,app.ReyInd),app.dataY(:,...
app.AoAValue,app.ReyInd),'LineWidth',2)

plot(app.ShapeFactor,-10:1:20,app.dataHTop(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'on')
plot(app.ShapeFactor,-10:1:20,app.dataHBot(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'off')
legend(app.ShapeFactor,'Suction Side','Pressure Side','Location','northwest')

plot(app.BLThick,-10:1:20,app.dataDStarTop(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'on')
plot(app.BLThick,-10:1:20,app.dataThetaTop(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataDStarBot(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataThetaBot(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'off')
legend(app.BLThick,'Displacement Suction Side','Momentum Suction Side','Displacement ...
    Pressure Side','Momentum Suction Side','Location','northwest')


        end

        % Callback function
        function NACADropDownValueChanged(app, event)
            app.foilName = app.NACADropDown.Value;
        end

        % Callback function
        function AoASliderValueChanging(app, event)
            app.sliderAoA = event.Value;
        end

        % Callback function
        function AoASliderValueChanged(app, event)
            app.AoAValue = round(app.AoASlider.Value) + 11;
        end

        % Value changing function: AoASlider
        function AoASliderValueChanging2(app, event)
            app.AoAValue = round(event.Value) + 11;
            plot(app.PressureDist,app.dataX(:,round(app.AoAValue),app.ReyInd),...
            app.dataCp(:,round(app.AoAValue),app.ReyInd),'LineWidth',2)
```

```
        end

        % Value changing function: RynoldsNumber106Slider
        function RynoldsNumber106SliderValueChanging(app, event)
            app.Rey = event.Value;
            app.ReyInd = round(app.Rey);
            plot(app.MnLPolars,app.dataAoA(:,3),app.dataCL(:,app.ReyInd),'LineWidth',2)
            hold(app.MnLPolars,'on');
            plot(app.MnLPolars,app.dataAoA(:,3),app.dataCM(:,app.ReyInd),'LineWidth',2)
            hold(app.MnLPolars,'off');
            legend(app.MnLPolars,'Lift Polar','Moment Polar','Location','northwest')


            plot(app.DragPolars,app.dataCD(:,app.ReyInd),app.dataCL(:,app.ReyInd),...
            'LineWidth',2)

            app.AoAValue = round(app.AoASlider.Value) + 11;
            plot(app.PressureDist,app.dataX(:,app.AoAValue,app.ReyInd),app.dataCp...
            (:,app.AoAValue,app.ReyInd),'LineWidth',2)
            set(app.PressureDist,'YDir','Reverse')
            plot(app.AirfoilProfile,app.dataX(:,app.AoAValue,app.ReyInd),app.dataY...
            (:,app.AoAValue,app.ReyInd),'LineWidth',2)

            plot(app.ShapeFactor,-10:1:20,app.dataHTop(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'on')
plot(app.ShapeFactor,-10:1:20,app.dataHBot(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'off')
legend(app.ShapeFactor,'Suction Side','Pressure Side','Location','northwest')

plot(app.BLThick,-10:1:20,app.dataDStarTop(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'on')
plot(app.BLThick,-10:1:20,app.dataThetaTop(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataDStarBot(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataThetaBot(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'off')
legend(app.BLThick,'Displacement Suction Side','Momentum Suction Side','Displacement ...
    Pressure Side','Momentum Suction Side','Location','northwest')
        end

        % Value changed function: RynoldsNumber106Slider
        function RynoldsNumber106SliderValueChanged(app, event)
            app.Rey = app.RynoldsNumber106Slider.Value;
            app.ReyInd = round(app.Rey);
            plot(app.MnLPolars,app.dataAoA(:,3),app.dataCL(:,app.ReyInd),'LineWidth',2)
            hold(app.MnLPolars,'on');
            plot(app.MnLPolars,app.dataAoA(:,3),app.dataCM(:,app.ReyInd),'LineWidth',2)
            hold(app.MnLPolars,'off');
            legend(app.MnLPolars,'Lift Polar','Moment Polar','Location','northwest')


            plot(app.DragPolars,app.dataCD(:,app.ReyInd),app.dataCL(:,app.ReyInd),...
            'LineWidth',2)

            app.AoAValue = round(app.AoASlider.Value) + 11;
            plot(app.PressureDist,app.dataX(:,app.AoAValue,app.ReyInd),app.dataCp...
            (:,app.AoAValue,app.ReyInd),'LineWidth',2)
            set(app.PressureDist,'YDir','Reverse')
            plot(app.AirfoilProfile,app.dataX(:,app.AoAValue,app.ReyInd),app.dataY...
            (:,app.AoAValue,app.ReyInd),'LineWidth',2)

            plot(app.ShapeFactor,-10:1:20,app.dataHTop(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'on')
plot(app.ShapeFactor,-10:1:20,app.dataHBot(:,app.ReyInd),'LineWidth',2)
hold(app.ShapeFactor,'off')
legend(app.ShapeFactor,'Suction Side','Pressure Side','Location','northwest')

plot(app.BLThick,-10:1:20,app.dataDStarTop(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'on')
plot(app.BLThick,-10:1:20,app.dataThetaTop(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataDStarBot(:,app.ReyInd),'LineWidth',2)
plot(app.BLThick,-10:1:20,app.dataThetaBot(:,app.ReyInd),'LineWidth',2)
hold(app.BLThick,'off')
```

```
legend(app.BLThick,'Displacement Suction Side','Momentum Suction Side','Displacement ...
    Pressure Side','Momentum Suction Side','Location','northwest')
        end
    end

    % Component initialization
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create XFoilRunEngineeringProgrammingChallengeUIFigure and hide until ...
                all components are created
            app.XFoilRunEngineeringProgrammingChallengeUIFigure = ...
                uifigure('Visible', 'off');
            app.XFoilRunEngineeringProgrammingChallengeUIFigure.Position = [100 100 ...
                940 754];
            app.XFoilRunEngineeringProgrammingChallengeUIFigure.Name = 'XFoil Run ...
                (Engineering Programming Challenge)';

            % Create TabGroup
            app.TabGroup = ...
                uitabgroup(app.XFoilRunEngineeringProgrammingChallengeUIFigure);
            app.TabGroup.Position = [27 21 888 669];

            % Create MomentandLiftPolarsTab
            app.MomentandLiftPolarsTab = uitab(app.TabGroup);
            app.MomentandLiftPolarsTab.Title = 'Moment and Lift Polars';

            % Create MnLPolars
            app.MnLPolars = uiaxes(app.MomentandLiftPolarsTab);
            title(app.MnLPolars, 'Moment and Lift Polars')
            xlabel(app.MnLPolars, 'AoA')
            ylabel(app.MnLPolars, 'Cx')
            app.MnLPolars.FontSize = 18;
            app.MnLPolars.Position = [13 13 865 623];

            % Create DragPolarsTab
            app.DragPolarsTab = uitab(app.TabGroup);
            app.DragPolarsTab.Title = 'Drag Polars';

            % Create DragPolars
            app.DragPolars = uiaxes(app.DragPolarsTab);
            title(app.DragPolars, 'Drag Polar')
            xlabel(app.DragPolars, 'Cd')
            ylabel(app.DragPolars, 'Cl')
            app.DragPolars.FontSize = 18;
            app.DragPolars.Position = [13 13 865 623];

            % Create PressureTab
            app.PressureTab = uitab(app.TabGroup);
            app.PressureTab.Title = 'Pressure';

            % Create PressureDist
            app.PressureDist = uiaxes(app.PressureTab);
            title(app.PressureDist, 'Pressure Distribution')
            xlabel(app.PressureDist, {'x/c'; ''})
            ylabel(app.PressureDist, 'Cp')
            app.PressureDist.Position = [13 361 865 266];

            % Create AirfoilProfile
            app.AirfoilProfile = uiaxes(app.PressureTab);
            title(app.AirfoilProfile, 'Airfoil Profile')
            xlabel(app.AirfoilProfile, {'x/c'; ''})
            ylabel(app.AirfoilProfile, 'Y')
            app.AirfoilProfile.Position = [13 73 865 266];

            % Create AoASliderLabel
            app.AoASliderLabel = uilabel(app.PressureTab);
            app.AoASliderLabel.HorizontalAlignment = 'right';
            app.AoASliderLabel.Position = [21 36 28 22];
            app.AoASliderLabel.Text = 'AoA';
```

```
        % Create AoASlider
        app.AoASlider = uislider(app.PressureTab);
        app.AoASlider.Limits = [-10 20];
        app.AoASlider.ValueChangingFcn = createCallbackFcn(app, ...
            @AoASliderValueChanging2, true);
        app.AoASlider.MinorTicks = [];
        app.AoASlider.Position = [70 45 796 3];

        % Create BoundaryLayerThicknessTab
        app.BoundaryLayerThicknessTab = uitab(app.TabGroup);
        app.BoundaryLayerThicknessTab.Title = 'Boundary Layer Thickness';

        % Create BLThick
        app.BLThick = uiaxes(app.BoundaryLayerThicknessTab);
        title(app.BLThick, {'Boundary Layer Thickness at Trailing Edge (x/c = ...
            1)'; ''})
        xlabel(app.BLThick, 'AoA')
        ylabel(app.BLThick, '\theta or \Delta^*')
        app.BLThick.FontSize = 18;
        app.BLThick.Position = [13 13 865 623];

        % Create ShapeFactorTab
        app.ShapeFactorTab = uitab(app.TabGroup);
        app.ShapeFactorTab.Title = 'Shape Factor';

        % Create ShapeFactor
        app.ShapeFactor = uiaxes(app.ShapeFactorTab);
        title(app.ShapeFactor, 'Shape Factor at Trailing Edge (x/c = 1)')
        xlabel(app.ShapeFactor, 'AoA')
        ylabel(app.ShapeFactor, {'\Delta^*/\theta'; ''})
        app.ShapeFactor.FontSize = 18;
        app.ShapeFactor.Position = [13 13 865 623];

        % Create RunXFoilMayTakeTimeButton
        app.RunXFoilMayTakeTimeButton = ...
            uibutton(app.XFoilRunEngineeringProgrammingChallengeUIFigure, 'push');
        app.RunXFoilMayTakeTimeButton.ButtonPushedFcn = createCallbackFcn(app, ...
            @RunXFoilMayTakeTimeButtonPushed, true);
        app.RunXFoilMayTakeTimeButton.FontSize = 18;
        app.RunXFoilMayTakeTimeButton.FontWeight = 'bold';
        app.RunXFoilMayTakeTimeButton.Position = [666.5 707 245 32];
        app.RunXFoilMayTakeTimeButton.Text = {'Run XFoil (May Take Time)'; ''};

        % Create RynoldsNumber106SliderLabel
        app.RynoldsNumber106SliderLabel = ...
            uilabel(app.XFoilRunEngineeringProgrammingChallengeUIFigure);
        app.RynoldsNumber106SliderLabel.HorizontalAlignment = 'right';
        app.RynoldsNumber106SliderLabel.Position = [58 711 124 22];
        app.RynoldsNumber106SliderLabel.Text = 'Rynolds Number 10^6';

        % Create RynoldsNumber106Slider
        app.RynoldsNumber106Slider = ...
            uislider(app.XFoilRunEngineeringProgrammingChallengeUIFigure);
        app.RynoldsNumber106Slider.Limits = [3 15];
        app.RynoldsNumber106Slider.ValueChangedFcn = createCallbackFcn(app, ...
            @RynoldsNumber106SliderValueChanged, true);
        app.RynoldsNumber106Slider.ValueChangingFcn = createCallbackFcn(app, ...
            @RynoldsNumber106SliderValueChanging, true);
        app.RynoldsNumber106Slider.MinorTicks = [];
        app.RynoldsNumber106Slider.Position = [203 730 417 3];
        app.RynoldsNumber106Slider.Value = 3;

        % Show the figure after all components are created
        app.XFoilRunEngineeringProgrammingChallengeUIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = XFoil_Run_Engineering_Programming_Challenge
```

```
            % Create UIFigure and components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.XFoilRunEngineeringProgrammingChallengeUIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.XFoilRunEngineeringProgrammingChallengeUIFigure)
        end
    end
end
```