Comparison of Turbulant RANS Model to Experimental Data and other Numerical Tools

Connor T. O'Reilly
University of Colorado Boulder, Boulder, CO, 80301
SID: 107054811

I. Nomenclature

C_L	=	Sectional coefficient of Lift
C_d	=	Sectional coefficient of Drag
AOA	=	Angle of Attack
deg	=	Degrees
rad	=	radians

II. Introduction

This lab utilized Ansys Fluent, a Computational Fluid Dynamics software to improve prediction capabilities for sectional lift and sectional drag coefficients for a NACA 0012 airfoil. Experimental Data and other numerical tools were used to predict these values in previous labs. Not all methods used accounted for viscous or turbulent affects. The purpose of this lab was to compare results from Thin Airfoil Theory, Vortex Panel Method and Experimental data to the predicted quantities utilizing the Ansys CFD software.

III. Methodology

Using Ansys Fluent CFD software, the provided NACA 0012 mesh was utilized to obtain predicted values for the sectional coefficients of lift and drag for multiple angles of attack. Initially the settings were set to analyze a two dimensional mesh grid with double precision turned on. Then the provided NACA 0012 case file was opened. In the solver a pressure based simulation was chosen to be run. All models except for the viscous model remained off. The viscous mode is changed from a laminar model to Splart-Allamares one equation model. which by definition $isaone-equationmodelthatsolvesamodelledtransportequationforthekinematiceddyturbulentviscosity.$. All model constants are not changed and are left as the default values. Under the material section, the fluid that we are using is air. Then density of air is then changed from constant to the density computed based on a ideal gas relationship. The viscosity of the fluid is set to 1.021e-5. After, the cell zone conditions are changed from a solid to a fluid. The material name remains as air but the operation pressure is set to 0 pascal so the simulation is not over pressured. Moving to the boundary conditions, the airfoil wall should be set to a wall condition, the interior should be set to an interior condition, and for the farfield conditions the type should be changed to a pressure farfield condition. The gauge pressure of the farfield should be changed to the atmospheric pressure at sealevel which is 101325.1 pascals. The mach number is changed to 0.15 for this lab, and the flow direction is changed to the respective angle of attack. For the x direction this is $\cos(\alpha)$ and for the y direction this will be $\sin(\alpha)$. No dynamic motion is imposed, and lastly the simulation is given a set of reference values. The reference conditions should be set to compute from farfield. For the solution, under methods a coupled approach is used and the default settings are not changed. Under controls the courant number is changed from 200 to 1. For these simulations, as the angle of attack was increased ans separation was noticed the relaxation factors for momentum and pressure were not changed. This was due to time restraints and lack of knowledge which produced minor inaccuracies in the computed values for sectional lift and drag coefficients. Under the monitor section, the convergence check criteria are lowered to 1e-6 to prevent the simulation from completing prematurely. New report plots are then created to monitor the convergence of C_L and C_d. A corresponding force report for C_L or C_d is then selected, and the coefficient is calculated along the airfoil wall. For the sectional drag coefficient, the force vector needs to be alligned with the free-stream direction. The x component is $\cos(alpha)$ and the y component is $\sin(alpha)$. For the sectional lift coefficient, x component and y component are swapped with the x-component being negated. All new plots are generated and printed to the console. Lastly the initialization is a standard initialization computed from farfield.

After all steps mentioned above are completed, 3,000 iterations are run and the last sectional lift and drag coefficients are selected for calculations.

IV. Results

After 3000 iterations of the simulation were completed for multiple angles of attack, the last sectional lift and drag coefficients were placed in an excel file along with their corresponding AOA. This file was then converted to a matrix in the $cfd_main.m$ matlab script to be used for calculations. Similarly, the Abbott and von Doenhoff experimental sectional lift and drag coefficient data was also imported from a text file and used for calculations. The matlab scatter function was used to create a scatter plot for both sub-figures in figure 1 and figure 2. The matlab curve fitting tool box was used to fit a quintic polynomial curve for Fig.1 (a), a smoothing spline for Fig.1 (b), a quadratic polynomial curve for Fig.2 (a) and lastly a linear polynomial curve for Fig.3 (a) and (b). After curve fitting, The Lift Slope, Zero-Lift Angle of Attack, Stall Angle, and Maximum C_L were calculated for the experimental Data, RANS model and the other numerical tools. For the RANS model and experimental data the zero lift angle of attack was calculated by parsing through both the data provided from NASA for the experiment and the simulation data for the RANS model to find the closest C_l element to zero. After these were compared to the closest C_l element from the curve fitted data for both and the closest value for each was chosen. The lift slope for both the RANS model and the Experimental data was found by fitting a linear polynomial curve to the linear section of each lift curve using the matlab function $polyfit$. $Polyfit$ computes the slope of each curve, which corresponded to the lift slope of each graph. The lift slope and Zero-Lift Angle of attack for Thin Airfoil Theory and Vortex Panel method was provided. The Maximum sectional coefficient of lift was found using the matlab $max()$ function, which returns the maximum value from an array. For the Experimental Data, the maximum C_l of the provided data and the curve fitted data was determined. For the RANS model the maximum C_l for the simulation data and the curve fitted data was used. And for both the Thin Airfoil Theory and Vortex Panel Method lift plots, the maximum sectional lift coefficient occurred at the largest Angle of Attack which was at 17 degrees for this instance. The Stall angle occurs at the Maximum sectional lift coefficient where after this point the lift coefficient starts to decrease rapidly. For the RANS model and Experimental Data the matlab $find()$ function was used to find the corresponding angle of attack for their maximum sectional coefficient of lift values. For Thin Airfoil Theory and the Vortex Panel the stall angle does not exist. This is because both methods do not account for viscous affects or stall simply, for an infinite amount of AOA the sectional lift coefficient never decreases.

Model	Lift Slope ($\frac{1}{rad}$)	Zero-Lift AOA (deg)	Stall Angle (deg)	Maximum C_L
Expermental	6.5834	0.000	16.3759	1.5959
RANS	6.015	0.000	13.500	1.270
Thin Airfoil Theory	2pi ≈ 6.283	0	DNE	1.864
Vortex Panel Method	6.890	0.004	DNE	2.0448

Table 1 Results from different methods

Following the above calculations the relative error for all quantities were found. It can be noticed that the RANS model does not have the most accurate predictions for Lift slope and Maximum sectional coefficient of lift. Because these values were determined from curve fitting data from simulations there are a few ways to obtain a more accurate calculation. The easiest way to obtain better predicted values would be to run more simulations at different angles of attack for the linear and stall regions to improve the accuracy of the curve fitting used in the matlab script. The only downside to that method would be the time for each simulation to complete which could take hours. For maximum sectional coefficient the accuracy of Thin Airfoil Theory can be attributed to the similarity to the experimental lift slope. Similar to the lift slope, the accuracy for computing the Stall angle using the RANS model can be improved by running more simulations for angles of attack in the stall region. In addition, different sectional lift coefficients were computed for the same angle of attack near the stall region. Due to time restraints the RANS model was not run in unsteady mode which could have improved the accuracy.

Method	Lift Slope	Zero-Lift AOA	Stall Angle	Max C_L
RANS	8.639 %	0.000 %	17.562 %	20.422 %
Thin Airfoil Theory	4.560 %	0.000 %	DNE	16.815 %
Vortex Panel Method	4.657%	0.4 %	DNE	28.127 %

Table 2 Relative error compared to Experimental

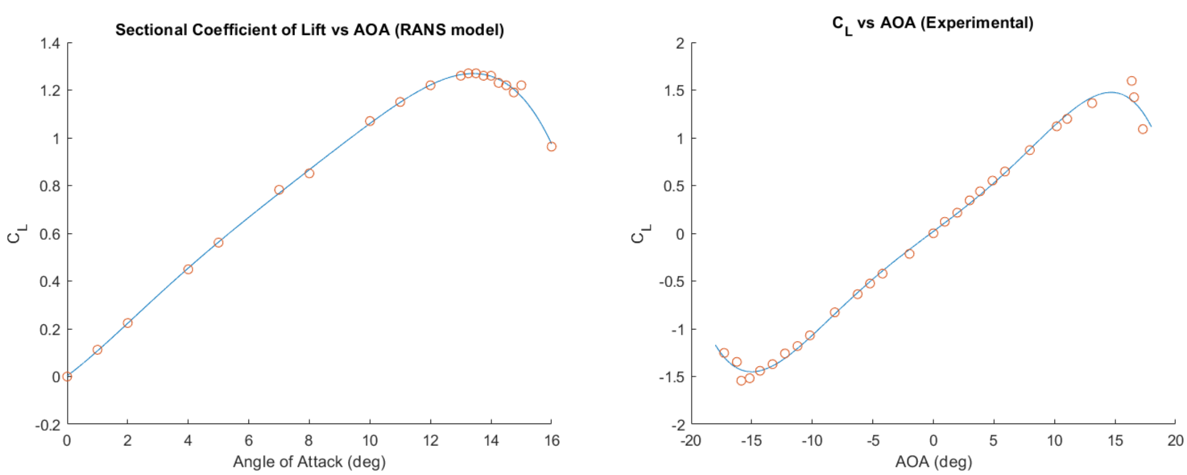
V. Conclusion

Compared to all the numerical tools taught to us in this course, using Ansys Fluent provides an easy way to estimate different values for an airfoil in in compressible, irrotatinal aerodynamic flow. The high relative errors in the predicted values stated in the results section are can be attributed to human error. To reduce these errors more simulations could have been run for angles of attack in the linear region and stall region of the lift curve. In addition, due to the computational time of each simulation the relaxation factors for momentum and pressure were not changed, causing sectional lift coefficient values in the stall region to be inaccurate. Lastly in for this case, obtaining values for each simulation from a poorly fitted line of best fit also affected the accuracy of the results. Besides the inaccuracies from the RANS model, it is the only model taught to us that can produce a lift slope similar to the lift slope of experimental data. Which provides us with important data that Thin airfoil theory and Vortex panel method cannot predict. These are the Stall Angle and the maximum coefficient of lift which were computed with 17.562 % and 20.422 % relative error respectively. If time is not a concern, this lab has shown that computational fluid dynamics software can provide solid predictions for airfoil characteristics.

VI. References

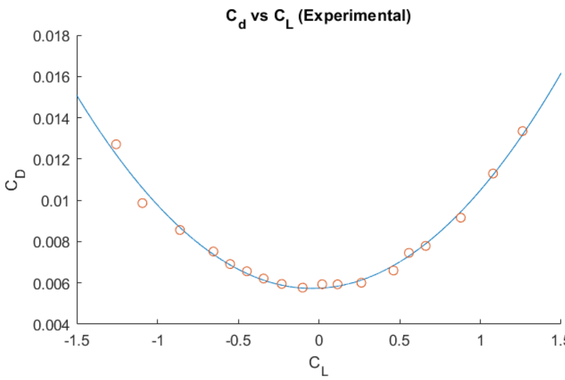
NASA Experimental Data for NACA 0012 Airfoil: $turbmodels.larc.nasa.gov/naca0012_{v}al.html$

VII. figures

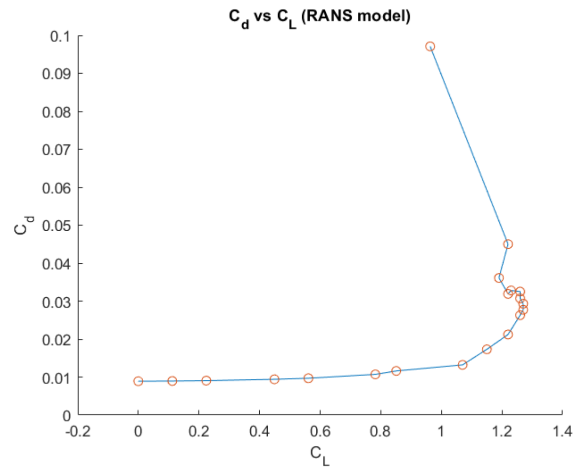


(a) **(b)**

Fig. 1

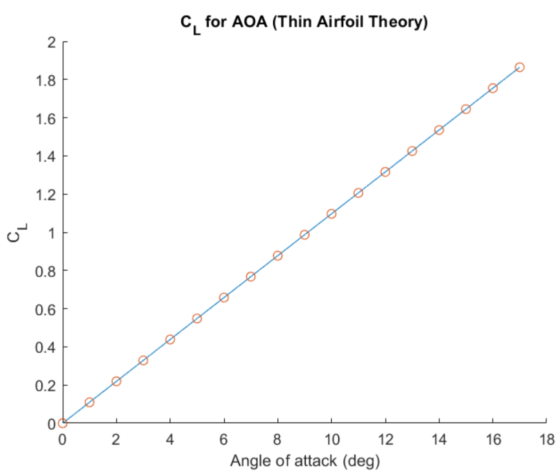


(a)

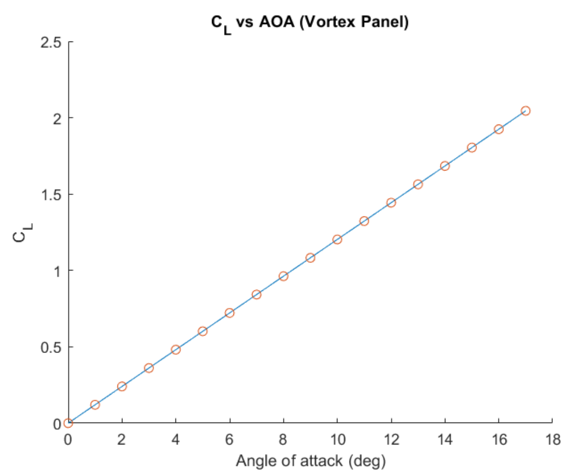


(b)

Fig. 2



(a)



(b)

Fig. 3

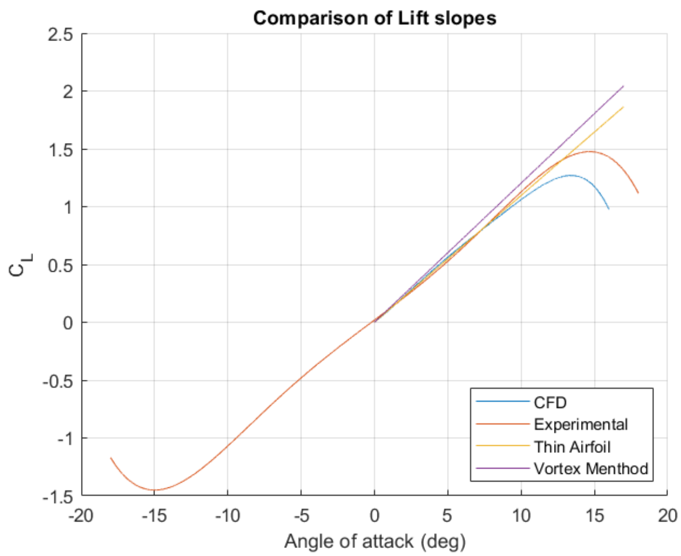


Fig. 4

VIII. matlab

```
%{
Author: Connor T. O'Reilly
Collaborators: kevin yevak
Last Revision: 12/12/2019
 %}


%% House Keeping
clc; clear all; close all;

%% Data

%obtained from ansys fluent at multiple AOA
cfd_data = readtable('CFD_Data_Real.csv');
%delete NAN values from empty rows
cfd_data = cfd_data(~any(ismissing(cfd_data),2),:);

%seperate dat
AOA = cfd_data{:,1};
cd_ansys = cfd_data{:,2};
cl_ansys = cfd_data{:,3};
AOA_rad = cfd_data{:,4};
cos_AOA = cfd_data{:,5};
sin_AOA = cfd_data{:,6};

%Import NASA Experimental Data for NACA 0012 Airfoil
    %Digitized data from Abbott & Von Doenhoff

%first column: AOA (deg) Second column: CL
 NASA_Exp_Cl = dlmread('NACA0012cl.dat');

%first column: CL Second column: CD
```

```
NASA_Exp_ClvCd = dlmread('NASA_Exp_ClCd.dat');

%% Problem 1 estimations

%plot cd and cl values estimated using ansys

%cl
figure('name', 'CL ansys')
hold on
%fit line to cfd
AOA_cfd_vals = linspace(0,16,10000);
p = fit(AOA,cl_ansys,'poly5');
cl_cfd_fit = feval(p,AOA_cfd_vals);
%actual plotting
plot(AOA_cfd_vals,cl_cfd_fit)
scatter(AOA,cl_ansys)
title('Sectional Coefficient of Lift vs AOA (RANS model)')
xlabel('Angle of Attack (deg)')
ylabel('C_L')
hold off

figure('name', 'Cd ansys')
hold on
%actual plotting
plot(cl_ansys,cd_ansys)
scatter(cl_ansys,cd_ansys)
title('C_d vs C_L (RANS model)')
xlabel('C_L')
ylabel('C_d')
hold off

%determine following quantities
    %Lift slope
    %Zero-lift angle of attack
    %Stall angle
    %Maximum sectional coefficient of lift

%Lift slope
    %determined from linear portion of lift slope
    %from AOA = 2 deg to AOA = 10 deg

p = polyfit(AOA(3:8),cl_ansys(3:8),1);
a_cfd = p(1); %lift slope
a_cfd_rad = p(1) * (180/pi);

%Zero-lift angle of attack
    %when CL is closest to zero
    %use both curve fitted line and cfd data

%take abs of array, grab minimum
val = min(abs(cl_ansys));
val2 = min(abs(cl_cfd_fit));

if val < val2
```

```
    %if sim values are closer to zero
    idx = find(abs(cl_ansys) == val);
    zero_L_cfd = AOA(idx);
else
    %if curve fitted values are closer
    idx = find(abs(cl_cfd_fit) == val2);
    zero_L_cfd = AOA_cfd_vals(idx);
end

%stall angle and maximum sectional lit coefficient
    %stall angle is at max c_l
    %find max of both fitted and actual values

val = max(cl_ansys);
val2 = max(cl_cfd_fit);
if val > val2
    max_cl_cfd = val;
    idx = find(cl_ansys == val);
    stall_cfd = AOA(idx);
else
    max_cl_cfd = val2;
    idx = find(cl_cfd_fit == val2);
    stall_cfd = AOA_cfd_vals(idx);
end

%weird error correction
if length(stall_cfd) ~= 1
    stall_cfd = max(stall_cfd);
end


%% Experimental Data and Numerical Methods

%lift slope for thin airfoil is 2pi

% Plot experimental data obtained from NASA 0012 exp

%cl vs AOA
figure('name', 'NASA Exp CL AOA' )
hold on
%fit line to data
AOA_fitted = linspace(-18,18,10000);
p = fit( NASA_Exp_Cl(:,1), NASA_Exp_Cl(:,2),'poly5');
a_exp = p(1);
a_exp_rad = p(1) * (180/pi);
%smoothing spline is used to find maximum cl values, other curves did not
%work too well
clAOA_exp_fit = feval(p,AOA_fitted);
%actual plotting
plot(AOA_fitted,clAOA_exp_fit)
scatter(NASA_Exp_Cl(:,1), NASA_Exp_Cl(:,2))
title('C_L vs AOA (Experimental)')
xlabel('AOA (deg)')
ylabel('C_L')
```

```
hold off

%characteristics
%zero lift AOA
zero_L_exp = 0;
%stall angle exp
val = max(NASA_Exp_Cl(:,2));
val2 = max(clAOA_exp_fit);
if val > val2
    max_cl_exp = val;
    idx = find(NASA_Exp_Cl(:,2) == val);
    stall_exp =  NASA_Exp_Cl(idx,1);
else
    max_cl_exp = val2;
    idx = find(clAOA_exp_fit == val2);
    stall_exp = AOA_fitted(idx);
end

%weird error correction
if length(stall_exp) ~= 1
    stall_exp = max(stall_exp);
end

%Cd vs CL
figure('name', 'NASA Exp CL CD' )
hold on
%fit line to data
cd_fitted = linspace(-1.5,1.5,10000);
p = fit( NASA_Exp_ClvCd(:,1), NASA_Exp_ClvCd(:,2),'poly2');
clcd_exp_fit = feval(p,cd_fitted);
%actual plotting
plot(cd_fitted,clcd_exp_fit)
scatter(NASA_Exp_ClvCd(:,1), NASA_Exp_ClvCd(:,2))
title('C_d vs C_L (Experimental)')
xlabel('C_L')
ylabel('C_D')
hold off

%plot for thinairfoil theory

%characteristics of
a_thin = 2*pi;
AOA_thin = rot90(0:17);
cl_thin = AOA_thin * 2 * (pi^2/180) ;
max_cl_thin = max(cl_thin);

%thin airfoil theory
figure('name', 'thin airfoil theory' )
hold on
%line fitting
AOA_thin_vals = linspace(0,17,10000);
p = fit(AOA_thin, cl_thin,'poly1');
cl_thin_fit = feval(p,AOA_thin_vals);
%actual plotting
```

```
plot(AOA_thin_vals,cl_thin_fit)
scatter(AOA_thin,cl_thin)
title('C_L for AOA (Thin Airfoil Theory)')
xlabel('Angle of attack (deg)')
ylabel('C_L')
hold off

%cahracteristics of vortex panel method
a_vortex = 6.89;
zero_L_vortex = 0.004;
AOA_vortex = rot90(0:17);
cl_vortex = (AOA_vortex + 0.004) * a_vortex * (pi/180);
max_cl_vortex = max(cl_vortex);

%vortex panel method
figure('name', 'vortex panel method' )
hold on
%line fitting
AOA_vor_vals = linspace(0,17,10000);
p = fit(AOA_vortex, cl_vortex,'poly1');
cl_vor_fit = feval(p,AOA_vor_vals);
%actual plotting
plot(AOA_vor_vals,cl_vor_fit)
scatter(AOA_vortex,cl_vortex)
title('C_L vs AOA (Vortex Panel)')
xlabel('Angle of attack (deg)')
ylabel('C_L')
hold off


%% comparison plots

%plot lines of best fit
figure('name', 'comparison' )
hold on
%line fitting
plot(AOA_cfd_vals,cl_cfd_fit)
plot(AOA_fitted,clAOA_exp_fit)
plot(AOA_thin_vals,cl_thin_fit)
plot(AOA_vor_vals,cl_vor_fit)
title('Comparison of Lift slopes')
xlabel('Angle of attack (deg)')
ylabel('C_L')
legend('CFD','Experimental','Thin Airfoil','Vortex Menthod','Location','southeast')
grid on
hold off

%% calculation outputs
fprintf('RANS Model: \n');
fprintf('\t    Lift Slope: %0.3f (/rad) \n', a_cfd_rad )
fprintf('\t Zero-Lift AOA: %0.3f (degrees)\n', zero_L_cfd)
fprintf('\t   Stall angle: %0.3f (degrees)\n', stall_cfd)
fprintf('\t   Maximum C_l: %0.3f \n', max_cl_cfd)
```

```
%% comparison outputs

%thin airfoil
fprintf('Thin Airfoil Theory: \n');
fprintf('\t    Lift Slope: 2pi (/rad) or %0.3f \n',a_thin)
fprintf('\t Zero-Lift AOA: 0 (degrees)\n')
fprintf('\t   Stall angle: DNE, thin airfoil doesnt\n \t \t \t \t \t \t account for stall \n')
fprintf('\t   Maximum C_l: %0.3f \n', max_cl_thin)

%vortex panel
fprintf('Vortex Panel Method: \n');
fprintf('\t    Lift Slope: %0.4f (/rad) \n', a_vortex )
fprintf('\t Zero-Lift AOA: %0.4f (degrees)\n', zero_L_vortex)
fprintf('\t   Stall angle: DNE \n')
fprintf('\t   Maximum C_l: %0.4f \n', max_cl_vortex)

%experimental panel
fprintf('Experimental: \n');
fprintf('\t    Lift Slope: %0.4f (/rad) \n', a_exp_rad )
fprintf('\t Zero-Lift AOA: %0.4f (degrees)\n', zero_L_exp)
fprintf('\t   Stall angle: %0.4f \n', stall_exp)
fprintf('\t   Maximum C_l: %0.4f \n', max_cl_exp)

%% Error analysis

%Lift slope
rel_a_rans = abs((a_exp_rad - a_cfd_rad)/a_exp_rad) * 100;
rel_a_thin = abs((a_exp_rad - a_thin)/a_exp_rad) * 100;
rel_a_vor = abs((a_exp_rad - a_vortex)/a_exp_rad) * 100;

rel_cl_rans = abs((max_cl_exp - max_cl_cfd)/max_cl_exp) * 100;
rel_cl_thin = abs((max_cl_exp - max_cl_thin)/max_cl_exp) * 100;
rel_cl_vor = abs((max_cl_exp -  max_cl_vortex)/max_cl_exp) * 100;

rel_st_rans = abs((stall_exp -  stall_cfd)/stall_exp) * 100;

fprintf('Error Analysis (values are precentage)')
fprintf('\n \n Lift Slope: \n')
fprintf('\t RANS: %0.3f \n',rel_a_rans)
fprintf('\t Thin Airfoil: %0.3f \n',rel_a_thin)
fprintf('\t Vortex: %0.3f \n',rel_a_vor)

fprintf('\n Zero Lift AOA: \n')
fprintf('\t RANS: %0.3f \n',0)
fprintf('\t Thin Airfoil: %0.3f \n',0)
fprintf('\t Vortex: %0.3f \n', zero_L_vortex * 100)


fprintf('\n Max CL: \n')
fprintf('\t RANS: %0.3f \n',rel_cl_rans)
fprintf('\t Thin Airfoil: %0.3f \n',rel_cl_thin)
fprintf('\t Vortex: %0.3f \n', rel_cl_vor)
fprintf('Thin airfoil and vortex do not account for stall so do not take zeriously')
```

```
fprintf('\n Stall Angle: \n')
fprintf('\t RANS: %0.3f \n',rel_st_rans)
```