SELF STUDY REPORT

INCOMPRESSIBLE FLUID DYNAMICS

16ME5DEIFD

Department of Mechanical Engineering

Visveswaraya Technological University, Belagavi

Submitted by:

Shrihari Ravichandran	**1BM16ME152**
Rohan S.	**1BM16ME124**
Prathik M.P.	**1BM16ME113**

Under the faculty,

Mr. Devaraj K., Assistant Professor

Department of Mechanical Engineering

BMS College of Engineering, Bengaluru-19

IFD: Analysis of MH45 airfoil on FLUENT and OpenFOAM.

Rohan S, Prathik MP, and Shrihari Ravichandran

December 18, 2018

Contents

0.1 Problem Specifications

We decided to analyze the Airfoil- MH45, (Martin Hepperle series) on the CFD software OpenFOAM (Version- 2.1.0). It is widely used for small to medium sized flying wings as it is a reflex airfoil i.e. the trailing edge is slightly curved upwards, which acts as a built-in tail for the flying wing.

The tail, as we know, is used to stabilise an aircraft longitudinally. For an aircraft using a reflex airfoil, the sweep of the wing and its taper ratio is adjusted, along with the centre of gravity, so that optimum longitudinal stability is attained.

The analysis includes everything from checking the variation of parameters like pressure, velocity of the flow over the airfoil till the contour plots.

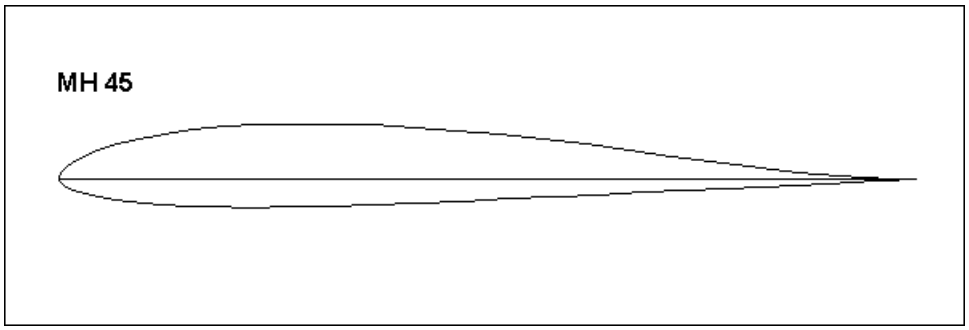
We also try to analyse the Lift coefficient and Drag coefficient (Cf and Cd) of the said Airfoil on the software called FLUENT (Version: 18.1). These values are then verified in the software XFLR5, plotting Cp vs X.

We then try varying the angle of attack of the flow and analyzing the said parameters in both laminar and turbulent models. The plots of velocity and pressure for both the angle of attack is compared between the two softwares. The results are later verified with the experimental data.

The stream-lines for different flows is analysed, and so is the flow seperation.

simpleFoam Solver is used in case of OpenFOAM.

The post processing is done using Paraview in case of Open-FOAM while the FLUENT has an in-built post processor.



The above figure shows the outline of the geometry formed just by connecting the splines.

0.2 Geometry

The airfoil was first imported into SOLIDWORKS^{TM}. A profile was then created around the airfoil, and a surface was created between the airfoil and the profile. This geometry was then imported into, and viewed in the modelling package offered by ANSYS Workbench 18.1, Spaceclaim.

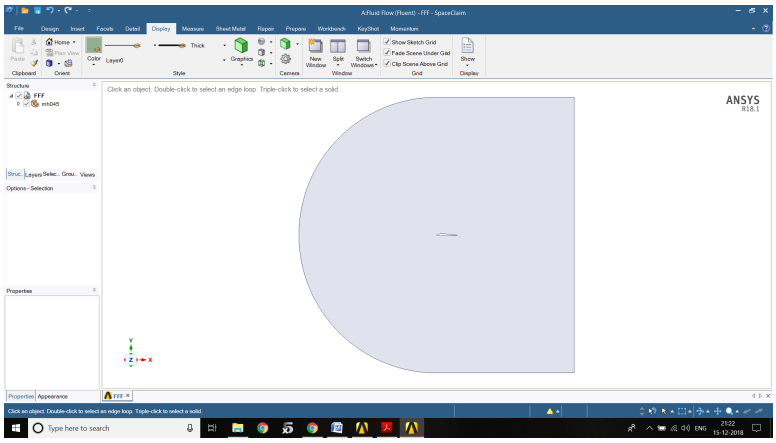
We chose a chord length of 150mm for the airfoil. The radius of the semi circular part of the domain was 1000mm and width of the rectangular part was 2000mm. The origin of the coordinate system was at the leading edge of the airfoil. The analysis type was set as 2D in the geometry properties in the Workbench window. This sufficed for the analysis on FLUENT but on OpenFOAM, the methodology was a bit different.

The geometry was imported into OpenFOAM using two methods. One was using the software called Gmsh- an open source meshing software. The coordinates are easily imported into Gmsh as OpenFOAM has in built libraries that facilitate this action.

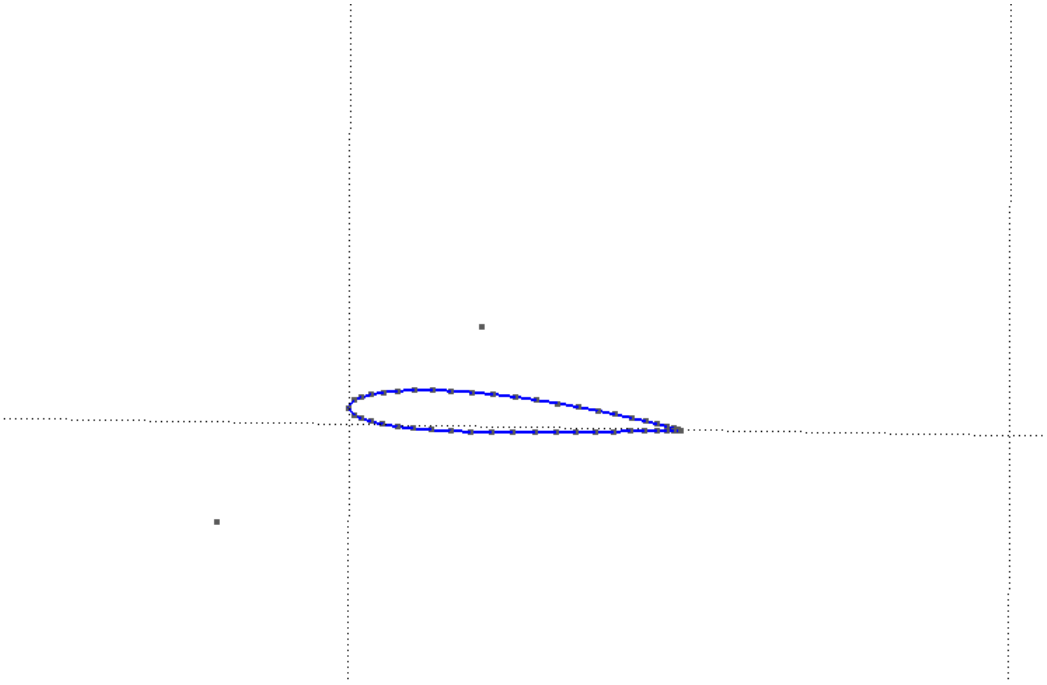
Using the code : gmshToFoam, the mesh is converted into an extension that can be accesible by OpenFOAM.

The saved file would be under the extension .geo which later using the software called ATOM, could be edited or read.

The second method is direct importing from FLUENT. OpenFOAM allows importing from fluent using the code : fluentMeshToFoam, where the file can be easily interpreted by OpenFOAM and the mesh is accessible, and finally, using the general BlockMeshdict code, the geometry is run.



The planar model is shown above.



The Gmsh file used to import onto OpenFOAM is shown in the above figure.

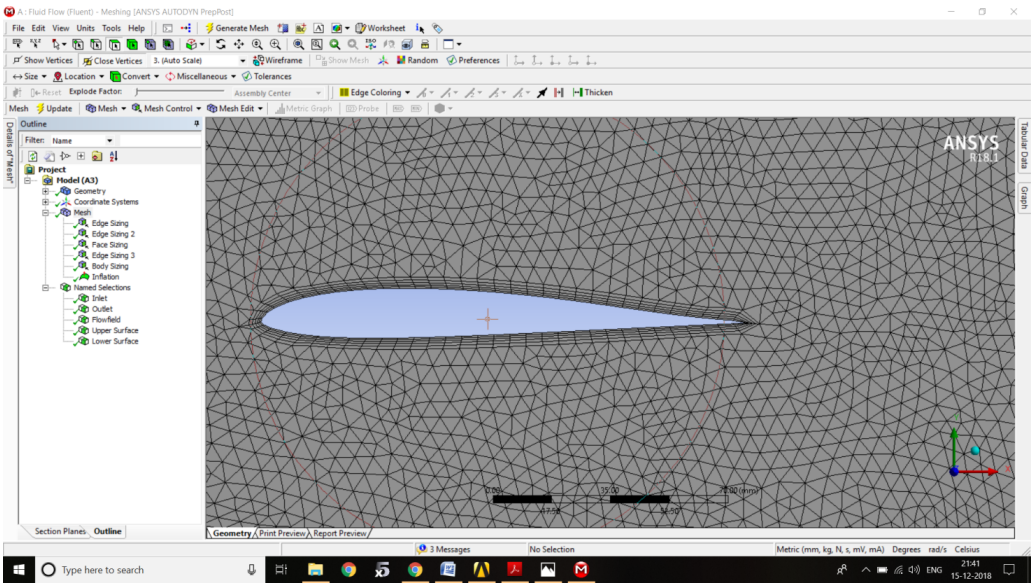
0.2.1 Meshing

On Ansys

Meshing was done in ANSYS Meshing, the meshing software provided by ANSYS. The first step was to create a default mesh. As expected, this default mesh was not fine enough to carry out an analysis of an accurate magnitude. Therefore, we applied the following sizings, and in the same order as shown below:-

1. Edge Sizing (Element size- 50mm, Soft Behaviour, Selected Edge- Semicircular)
2. Edge Sizing 2 (Element size- 50mm, Hard Behaviour, Selected edges- 2 horizontal, 1 vertical)
3. Face Sizing (Element size- 20mm, Soft Behaviour, Selected face- Surface)
4. Edge Sizing 3 (Element size- 2mm, Hard Behaviour, Selected Edges- Airfoil)
5. Body Sizing- (Sphere of Influence centred at leading edge, radius 500mm, element size- 5mm)

6. Inflation (First layer height 0.5mm, maximum number of layers- 5, growth rate-1.2, edges- airfoil) The resultant mesh was as shown below. Note that due to the soft nature of face sizing, the cells have been triangulated.



The resultant mesh was as shown above. Note that due to the soft nature of face sizing, the cells have been triangulated.

On OpenFOAM

Our initial idea was to extract the airfoil out of a 2-D box using blockMesh file. So the components in X and Y direction have been given sufficient value while the thickness is very low.

The analysis is 2D, hence the said reason.

The refinement along X and Y direction is higher than in Z direction. (Because of the flow direction)

The patches indicate the entry of the flow, wall (empty) condition is given in the directions that is devoid of flow and redundent for analysis.

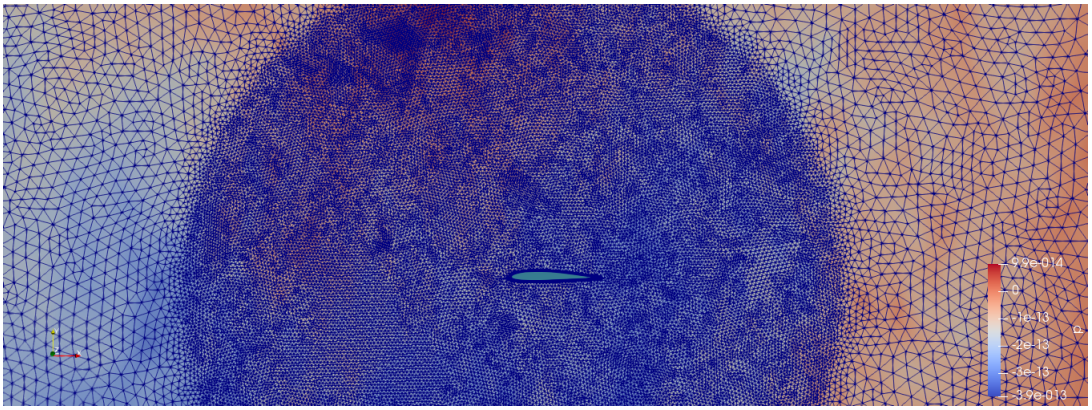
The code is given below:

FoamFile
version 2.0;
format ascii;
class dictionary;
object blockMeshDict;
// * //
convertToMeters 1;

```
vertices
(
(0 0 0)
(6 0 0)
(6 2 0)
(0 2 0)
(0 0 0.01)
(6 0 0.01)
(6 2 0.01)
(0 2 0.01)
);
blocks
(
hex (0 1 2 3 4 5 6 7) (20 8 8) simpleGrading (1 1 1)
 );
edges ( );
boundary
(
topAndBottom
type patch;
faces
 (
(2 3 7 6) (0 1 5 4)
);
inlet
type patch;
faces
( (3 0 4 7) );
outlet  type patch;
faces ( (1 2 6 5) );
front  type empty;
faces ( (4 5 6 7) );
back  type empty;
faces ( (0 3 2 1) );
);
mergePatchPairs ( );
// ********************************************************************//
```

Later we found out that the code fluentMeshToFoam runs through the mesh and creates the polyMesh folder of its own, containing the geometry inside the folder.

The geometry is later meshed in OpenFOAM which can be viewed using Paraview. (Using the code: paraFoam)



The meshed surface with the airfoil extracted can be clearly observed in the above figure.

The meshing is highly refined (more grading) near the airfoil region than the edges of the 2D- box.

0.3 Solver used: SimpleFoam

Equations:

The solver solves the basic equations of flow, the mass conservation, and the momentum conservation (Navier-Stokes Equation) as follows:

$$\nabla \cdot \mathbf{u} = 0$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} - \nabla p + \mathbf{g}$$

$$\nu \equiv \frac{\mu}{\rho}$$

About:

OpenFOAM does not have a generic solver applicable to all cases. The solvers with the OpenFOAM distribution are in directory, reached quickly by typing sol at the command line. This directory is further subdivided into several directories by category of continuum mechanics, e.g. incompressible flow, combustion and solid body stress analysis. Each solver is given a name that is reasonably descriptive, e.g.icoFoam solves incompressible, laminar model.

It depends on the conditions and kind of results required. The solver here is SimpleFoam, which takes in just pressure and velocity as the parameters to solve for the laminar condition.

(In case of rhoCentralFoam, it needs temperature condition too)

The simpleFoam solver, so-called as it is an implementation for steady incompressible flow using the SIMPLE algorithm.

The solver has full access to all the turbulence models in the incompressibleTurbulenceModels library and the non-Newtonian models incompressibleTransportModels library of the standard OpenFOAM release.

It can be found in:

WM-PROJECT-DIR/applications/solvers/incompressible/simpleFoam

0.4 Setup- Ansys

Mathematical model and the boundary conditions were set up using ANSYS FLUENT 18.1, the CFD Software provided by ANSYS.

FLUENT solves the momentum and mass conservation equations for each cell and evaluates the values of pressure, velocity, mass imbalance, etc., at the cell centres. For locations other than cell centres, interpolation of a suitable order is used to find the values of the said quantities.

The solver used was Pressure Based. 2D Planar Space option was used, keeping in mind the planar domain. Analysis was conducted in steady state conditions. Airspeed was taken as 15m/s. The values of drag and lift coefficients were monitored and printed to the console after each iteration. The maximum number of iterations was set at 1000 to allow for convergence. Convergence criteria for continuity, x-velocity, y-velocity, k and epsilon were set at 0.001.

Based on the above data, we generated four cases, which were as follows:-

1. Laminar Method at 0 degree angle of attack 2. K-epsilon Turbulence model at 0 degree angle of attack 3. Laminar model at 3 degree angle of attack 4. K-epsilon Turbulence model at 3 degree angle of attack

Case 1-

Model Laminar

Initial Gauge pressure - 0 Pa

Reference Temperature - 288K

Convergence - None

Drag Coefficient - 0.004

Lift Coefficient - 0.03

Case 2-

Model Realizable K-Epsilon Turbulence with Enhanced Wall Treatment

Initial Gauge pressure - 0 Pa

Reference Temperature - 288K

Turbulence

Convergence - 183rd iteration

Drag Coefficient - 0.0034

Lift Coefficient - 0.024

Case 3-

Model Laminar

Initial Gauge pressure 0 Pa

Reference Temperature 288K

Convergence None

Drag Coefficient 0.022

Lift Coefficient 0.432

Case 4- Model Realizable K-Epsilon Turbulence with Enhanced Wall Treatment

Initial Gauge pressure 0 Pa

Reference Temperature 288K

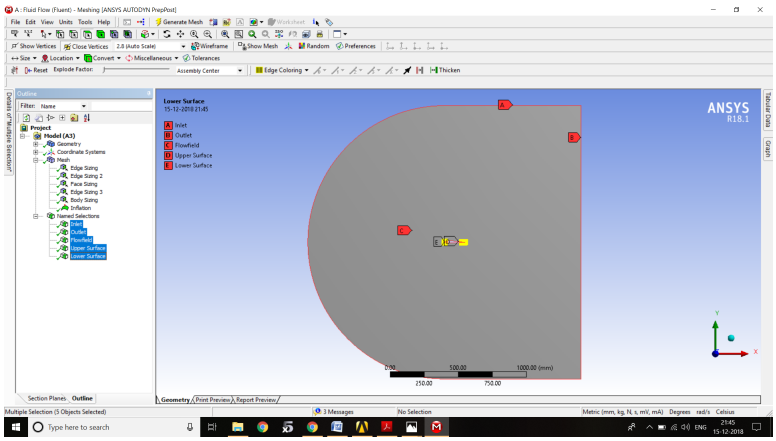
Turbulence Kinetic Energy 1.859*(10e-5)

Turbulence dissipation rate 9.3706*(10e-5)

Convergence 135th iteration

Drag Coefficient 0.024

Lift Coefficient 0.473



0.5 Setup- OpenFOAM

0.5.1 Laminar model

Pressure

The pressure values are typed in the p file of th folder '0' of the OpenFoam case directory.

The code is given below for the laminar case.

```
/*---------------------*- C++ -*----------*/ —
/ Field — OpenFOAM: The Open Source CFD Toolbox —
/ Operation — Version: 2.3.0 —
/ And — Web: www.OpenFOAM.org —
/ Manipulation — ————————————————————*/
 FoamFile
 version 2.0;
 format ascii;
 class volScalarField;
 object p;
 dimensions [1 -1 -2 0 0 0 0];
 internalField uniform 0;
 boundaryField
 Inlet
 type zeroGradient;
 Outlet
 type fixedValue;  value uniform 0;
Front
type zeroGradient;
Top
type zeroGradient;
Bottom
type zeroGradient;
Suction
type kqRWallFunction;
```

The dimensions are of pressure and as it can be seen, the internal field is uniform and kept at 0, all that matters is the inlet pressure conditions, which is kept at zero gradient. (Meaning the normal gradient of pressure is zero).

dimensions specifies the dimensions of the field, here of pressure.

internalField the internal field data which can be uniform, described by a single value; or nonuniform, where all the values of the field must be specified.

boundaryField the boundary field data that includes boundary conditions and data for all the boundary patches.

Velocity- Zero angle of attack

The velocity for laminar flow is low and the internal field is kept at a uniform 15m/s. The code is given below. It is saved in '0' folder as well, along with the pressure.

```
/*——————-*- C++ -*——* — —
/ Field — OpenFOAM: The Open Source CFD Toolbox —
/ Operation — Version: 2.1.0 —
/ And — Web: www.OpenFOAM.org — —
/ Manipulation — —
 FoamFile
 version 2.0;
 format ascii;
 class volVectorField;
 location "0";
 object U;
 dimensions [0 1 -1 0 0 0 0];
 internalField uniform (15.00 0 0);
 boundaryField
Inlet  type fixedValue; value uniform (15 0 0);
Outlet  type inletOutlet;
 inletValue uniform (0 0 0);
 value internalField;
Front  type fixedValue;
 value uniform (0 0 0);
Top  type zeroGradient;
Bottom  type zeroGradient;  Suction
type slip;
// ********************************* //
```

This represents the code for the velocity file at zero AOA. There is only one component of velocity in X-direction, and zero in the other two.

Velocity- 3 degree Angle of attack

The difference in this case is the increase in the component of velocity in Y-direction. Rest all the conditions present, remains the same.

The only change in the code is:

boundaryField

Inlet type fixedValue;

value uniform (14.484 0.78 0);

There is no need of coding Mach number (Ma) for such low velocity values, all our analysis takes place at subsonic levels.

0.5.2 Turbulent model

kEpsilon model was used to simulate the flow.

The values of k and Epsilon has to be coded in the '0' folder of the OpenFOAM case directory.

Epsilon code

The codes are given below:

```
Epsilon:
 FoamFile
 version 2.0;
 format ascii;
class volScalarField;
 object epsilon;
  // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
 dimensions [0 2 -3 0 0 0 0];
internalField uniform 14.855;
 boundaryField
inlet  type fixedValue; value internalField;
 outlet
type zeroGradient;
 upper surface
type slip;
 lower surface
type slip;
```

K code:

```
 dimensions [0 2 -2 0 0 0 0];
 internalField uniform 0.0096;
 boundaryField
 inlet
type fixedValue; value internalField;
 outlet
type zeroGradient;
upper surface  type slip;
lower surface  type slip;
```

0.6 Schemes Used:

The fvSchemes file in the Openfoam case directory consists of various schemes which should be used depending on the flow conditions and requirements.

The fvSchemes dictionary in the system directory sets the numerical schemes for terms, such as derivatives in equations, that are calculated during a simulation.

The terms that must typically be assigned a numerical scheme in fvSchemes range from derivatives, e.g. gradient Δ , to interpolations of values from one set of points to another.

The aim in OpenFOAM is to offer an unrestricted choice to the user, starting with the choice of discretisation practice which is generally standard Gaussian finite volume integration Gaussian integration is based on summing values on cell faces, which must be interpolated from cell centres.

The user has a wide range of options for interpolation scheme, with certain schemes being specifically designed for particular derivative terms, especially the advection divergence terms.

The set of terms, for which numerical schemes must be specified, are subdivided within the fvSchemes dictionary into the categories below.

- timeScheme: first and second time derivatives
- gradSchemes: gradient Δ • divSchemes: divergence Δ •
- laplacianSchemes: Laplacian
- interpolationSchemes: cell to face interpolation values.
- snGradSchemes: component of gradient normal to a cell face.
- wallDist: distance to wall calculation, where required.

Each keyword in represents the name of a sub-dictionary which contains terms of a particular type, e.g.gradSchemes contains all the gradient derivative terms such as grad(p)

0.7 ControlDict:

The OpenFOAM solvers begin all runs by setting up a database.

The database controls I/O and, since output of data is usually requested at intervals of time during the run, time is an inextricable part of the database.

The controlDict dictionary sets input parameters essential for the creation of the database. The keyword entries in controlDict are listed in the following sections.

Only the time control and writeInterval entries are mandatory, with the database using default values for any of the optional entries that are omitted.

The controlDict dictionary file for the given case is given below:

```
FoamFile
version 2.0;
format ascii;
class dictionary;
object controlDict;
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
application simpleFoam;
startFrom latestTime;
startTime 0;
stopAt endTime;
endTime 200;
deltaT 1;
writeControl timeStep;
writeInterval 100;
purgeWrite 0;
writeFormat ascii;
writePrecision 6;
writeCompression uncompressed;
timeFormat general;
timePrecision 6;
runTimeModifiable true;
// ********************************************** //
```

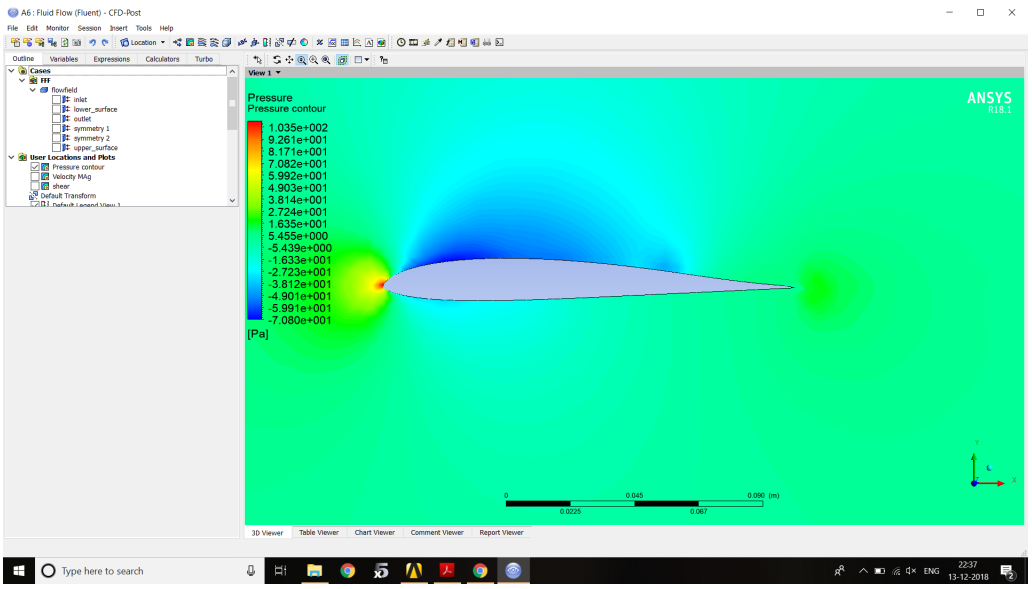
As it can be observed, the writeInterval, timestep etc, are set in this file, it gives software the details regarding when to start and end the analysis/simulation.

0.8 Post Processing

Pressure Plots: Laminar model at 0 and 3 degree angle of attacks in Ansys

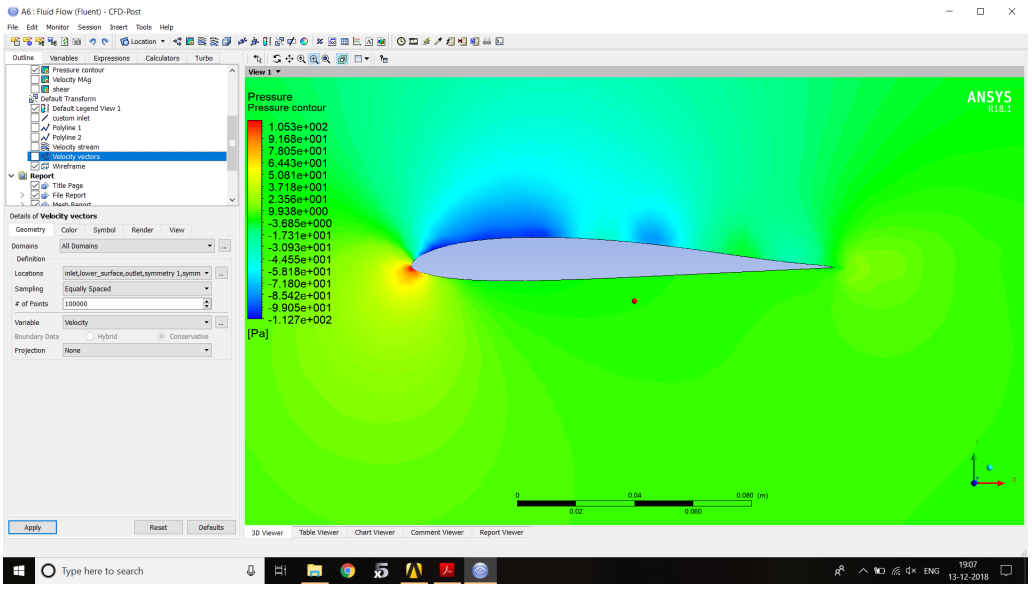
The MH45 airfoil is tested against two angle of attacks - 0 and 3 degree. The pressure contour and velocity vectors are plotted and shown in the figure.

The velocity vector gives the clear picture of the velocity distribution and the pressure contour gives clear picture of static pressure over an airfoil. The pressure coefficient over an airfoil is also plotted with help of Fluent and is compared with the experimentation data to give a clear indication of stalling, and various response of flow over an airfoil. The fluid speed was taken to be uniform and equal to 15m/s.



The figure shows the pressure contour for laminar flow at 0 degree angle of attack.

At three degree angle of attack, there's increase in component of velocity in the Y-direction and hence the flow over the leading edge differs from that of the latter case. It can be observed in the below figure.

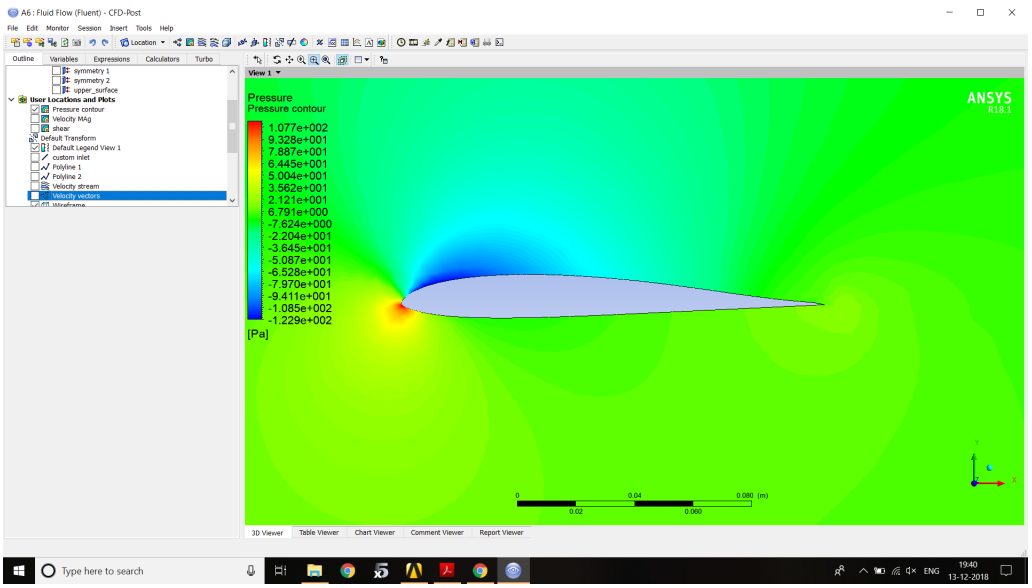


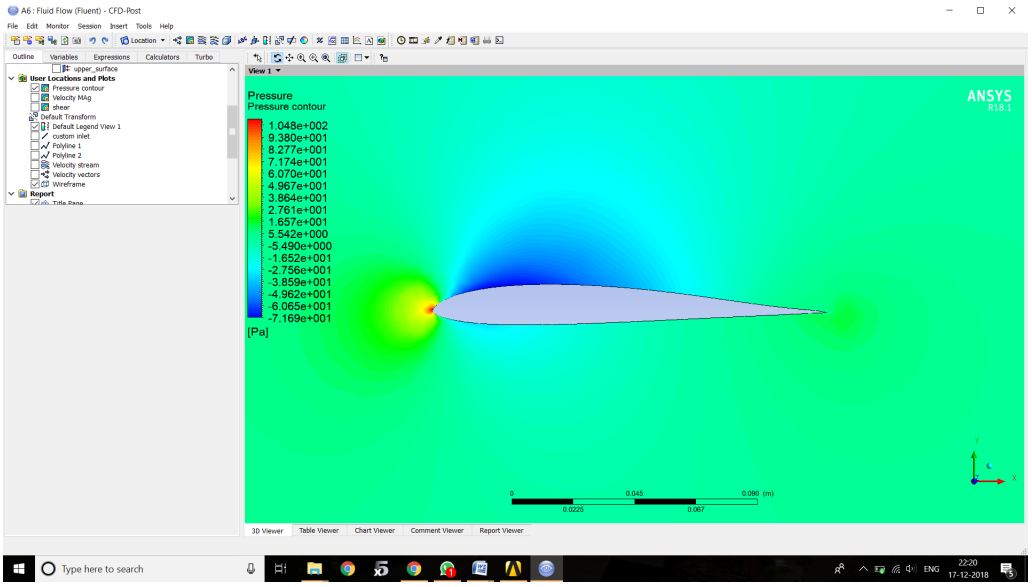
Pressure Plots: Turbulent model at 0 and 3 degree angle of attacks in Ansys

k-Epsilon model of turbulence was used. K-epsilon turbulence model is the most common model used in CFD to simulate mean flow characteristics for turbulent flow conditions. It is a two equation model that gives a general description of turbulence by means of two transport equations (PDEs). The original impetus for the K-epsilon model was to improve the mixing-length model, as well as to find an alternative to algebraically prescribing turbulent length scales in moderate to high complexity flows.

The first transported variable is the turbulence kinetic energy (k). The second transported variable is the rate of dissipation of turbulence energy (epsilon).

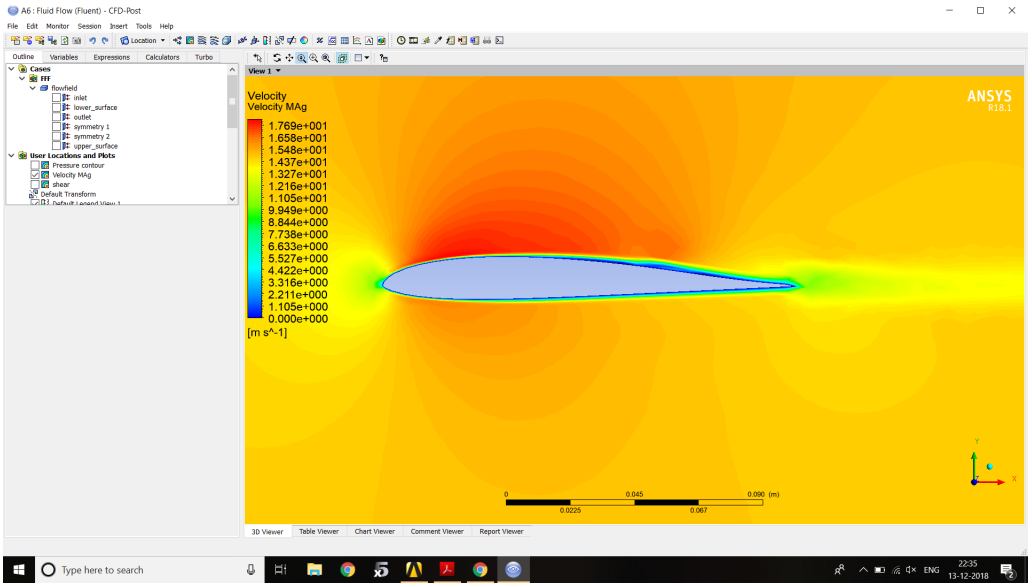
The below figure shows the pressure contour in case of turbulence model and 3 degree angle of attack. Observe that the contour seems *comparatively smooth* than that of laminar flow case.


A6 : Fluid Flow (Fluent) - CFD-Post
Pressure
Pressure contour
1.077e+002
9.328e+001
7.887e+001
6.445e+001
5.004e+001
3.562e+001
2.121e+001
6.791e+000
-7.624e+000
-2.204e+001
-3.645e+001
-5.087e+001
-6.528e+001
-7.970e+001
-9.411e+001
-1.085e+002
-1.229e+002
[Pa]
ANSYS
R18.1

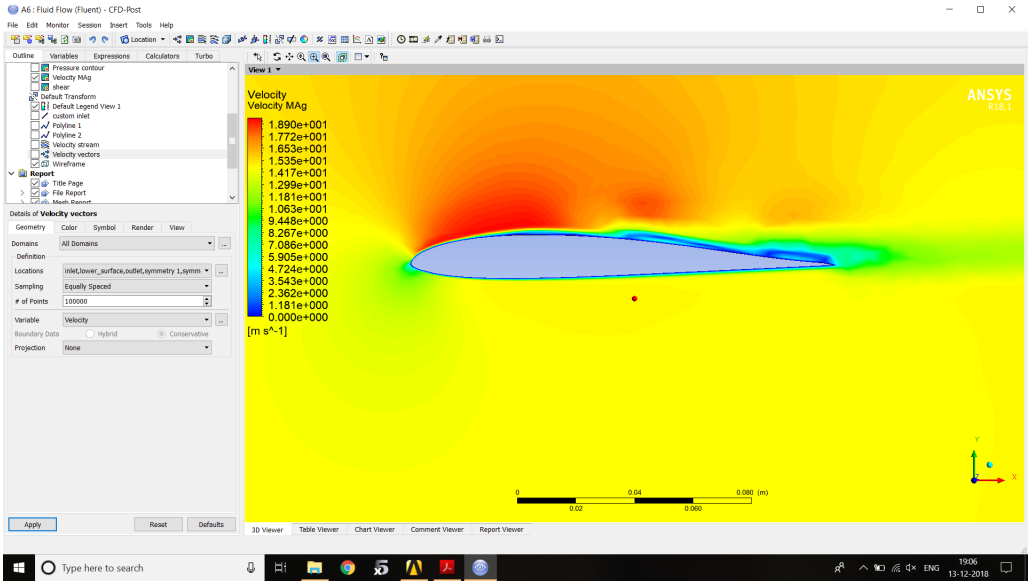

A6 : Fluid Flow (Fluent) - CFD-Post
Pressure
Pressure contour
1.048e+002
9.380e+001
8.277e+001
7.174e+001
6.070e+001
4.967e+001
3.864e+001
2.761e+001
1.657e+001
5.542e+000
-5.490e+000
-1.652e+001
-2.756e+001
-3.859e+001
-4.962e+001
-6.065e+001
-7.169e+001
[Pa]
ANSYS
R18.1

Velocity Plots: Laminar model at 0 and 3 degree angle of attacks in Ansys

The following images show the velocity contour for both the angle of attacks.

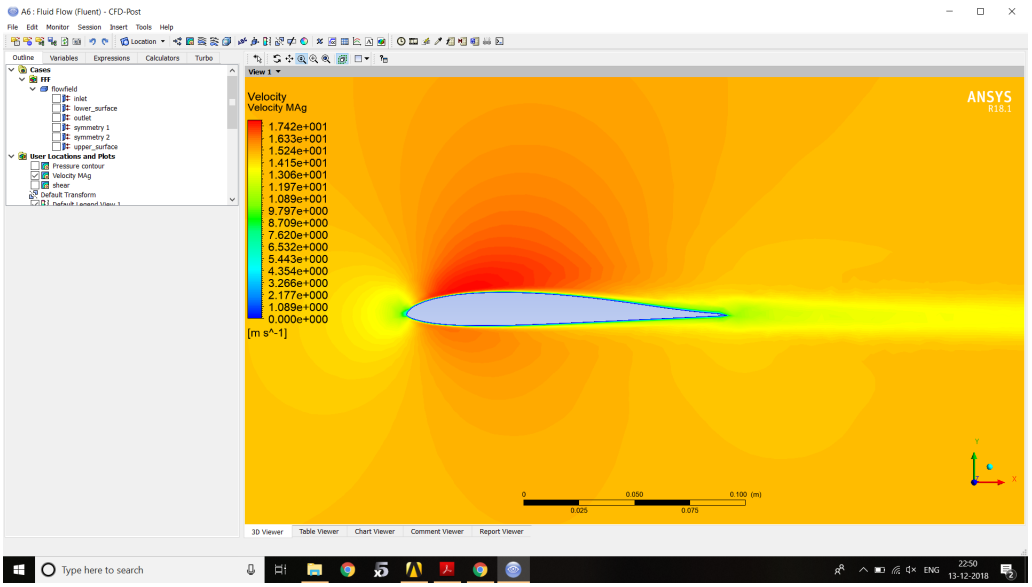


The above figure is the contour for 0 degree angle of attack in a laminar flow.

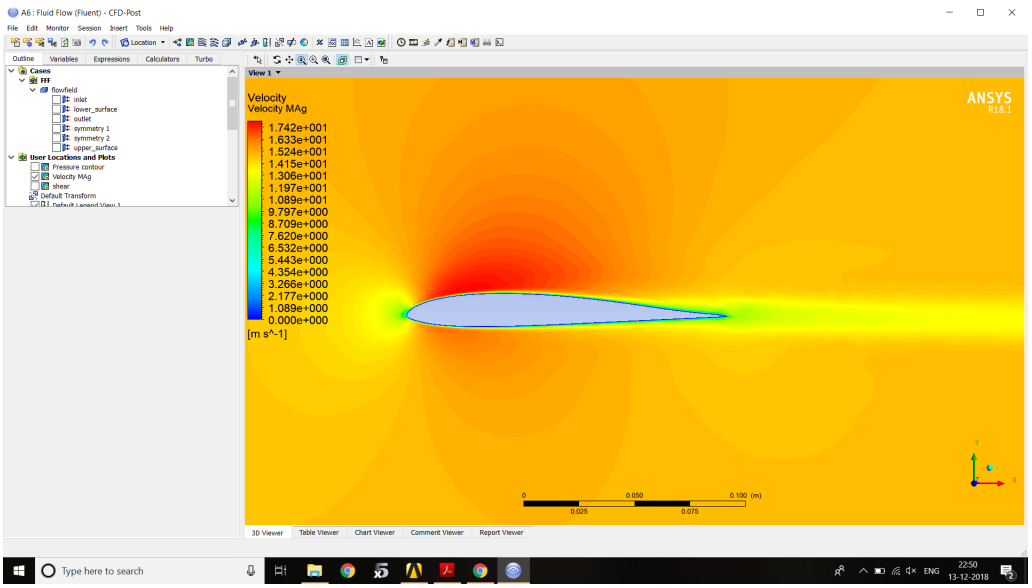


In the above figure it is observed at 3 degree angle of attack, the contour seems *not-so-smooth* and it's primarily because of the extra velocity component.

Velocity Plots: Turbulent model at 0 and 3 degree angle of attacks in Ansys

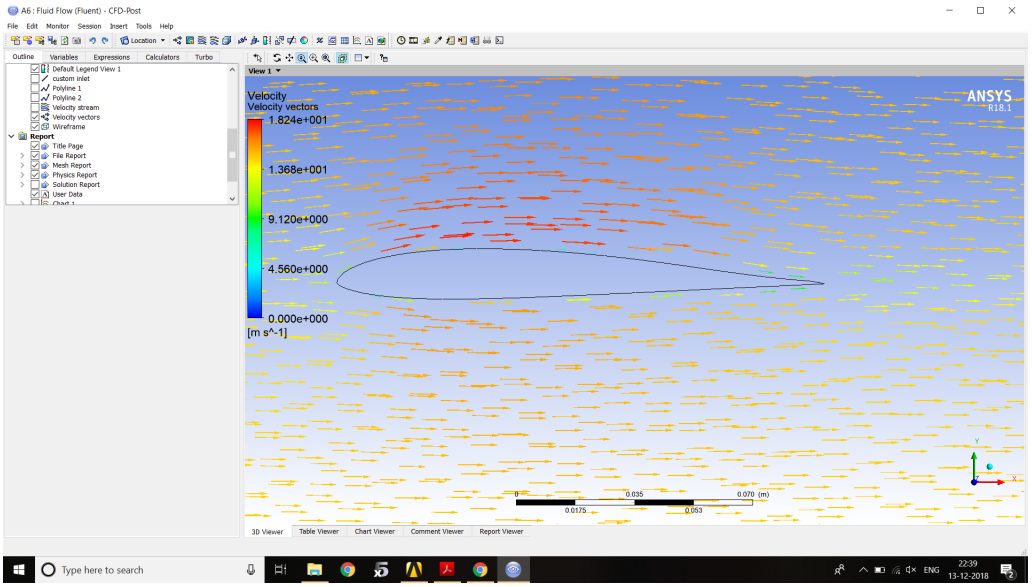


Surprisingly, the smoothness is established in turbulent flow well, compared to that of laminar. The above figure shows the turbulent flow velocity contour at zero degree angle of attack.

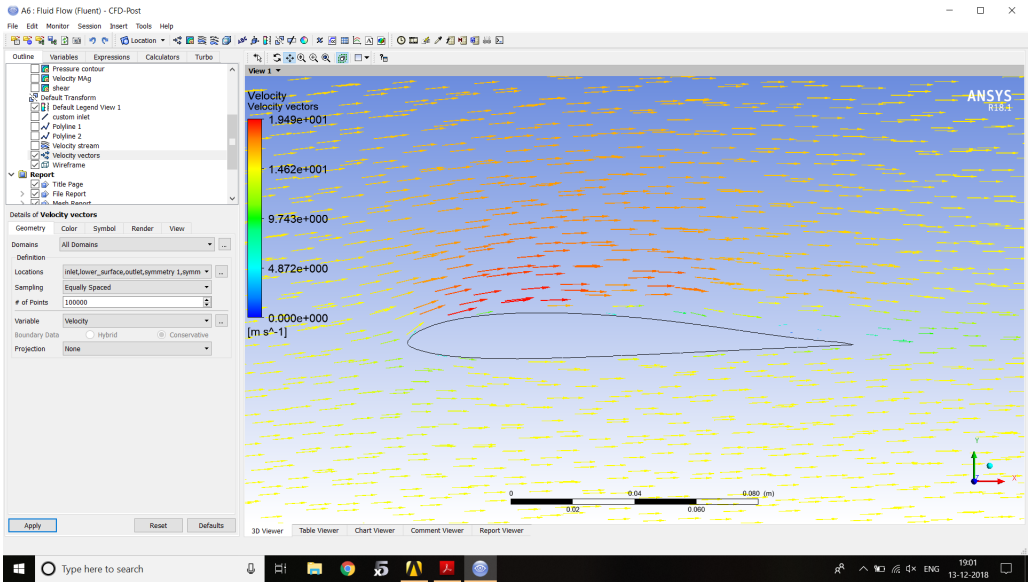


The above figure shows the contour at 3 degree angle of attack.

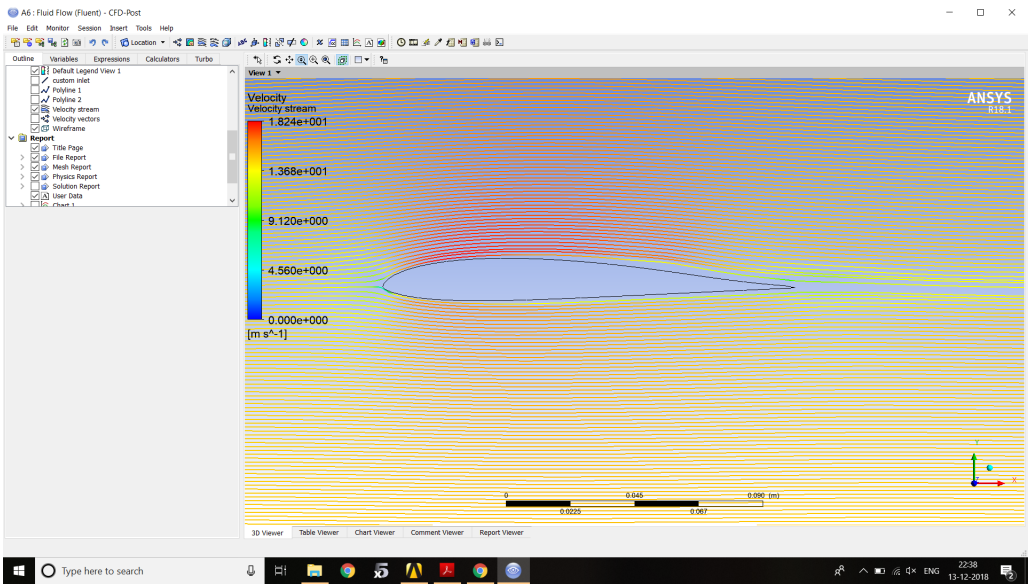
Velocity vectors and flow seperation

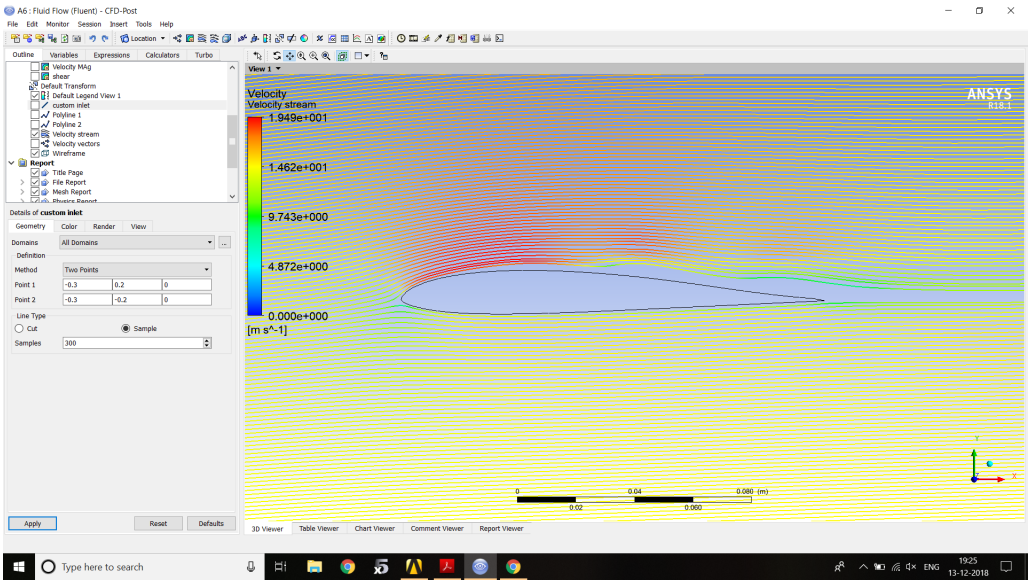


The above figure shows the velocity vector in case of laminar flow at 0 degree angle of attack.

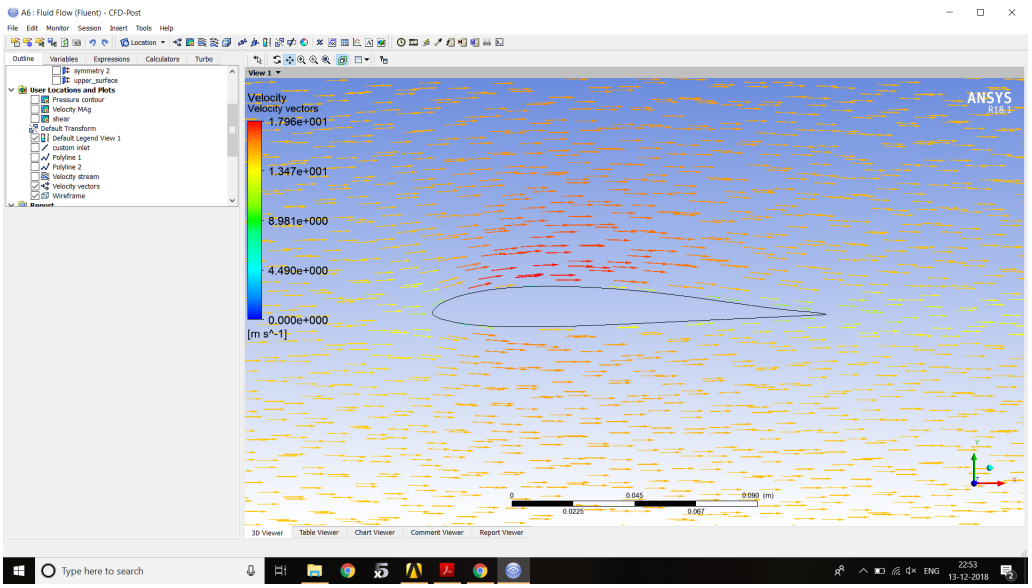


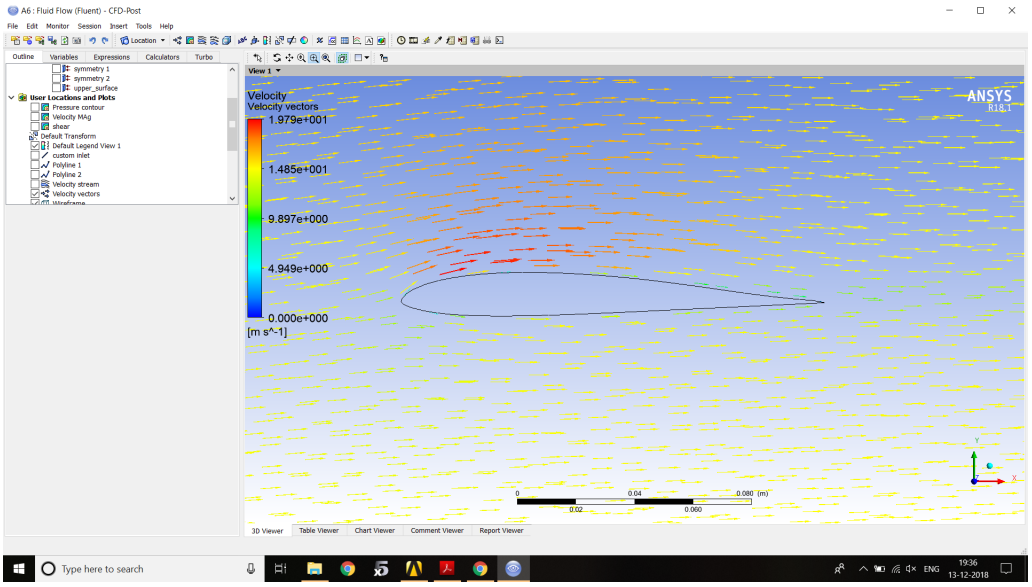
The above figure shows the velocity vector in case of laminar flow at 3 degree angle of attack.



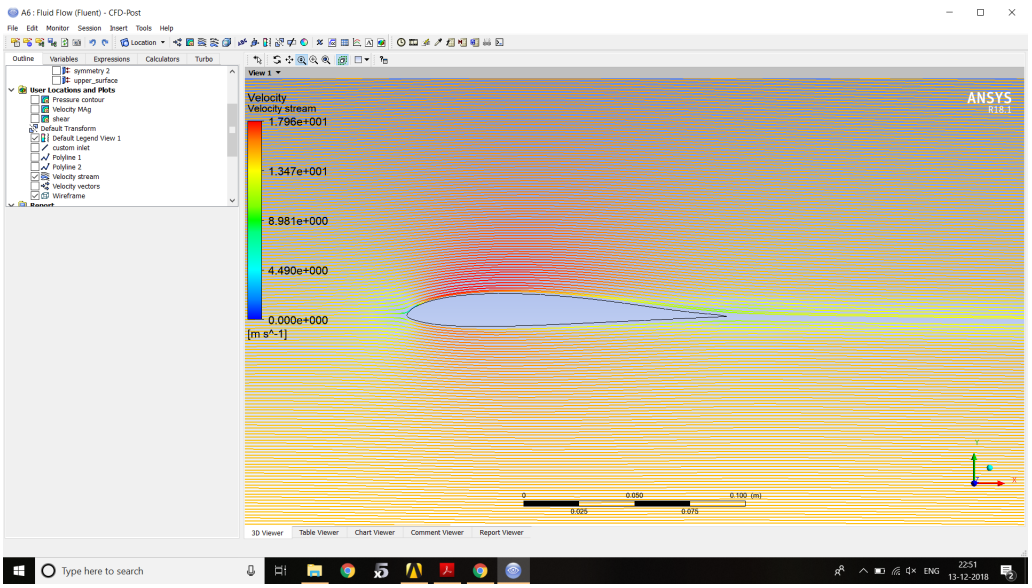


The above figure shows streamlines in case of laminar method of analysis for both the angle of attacks.

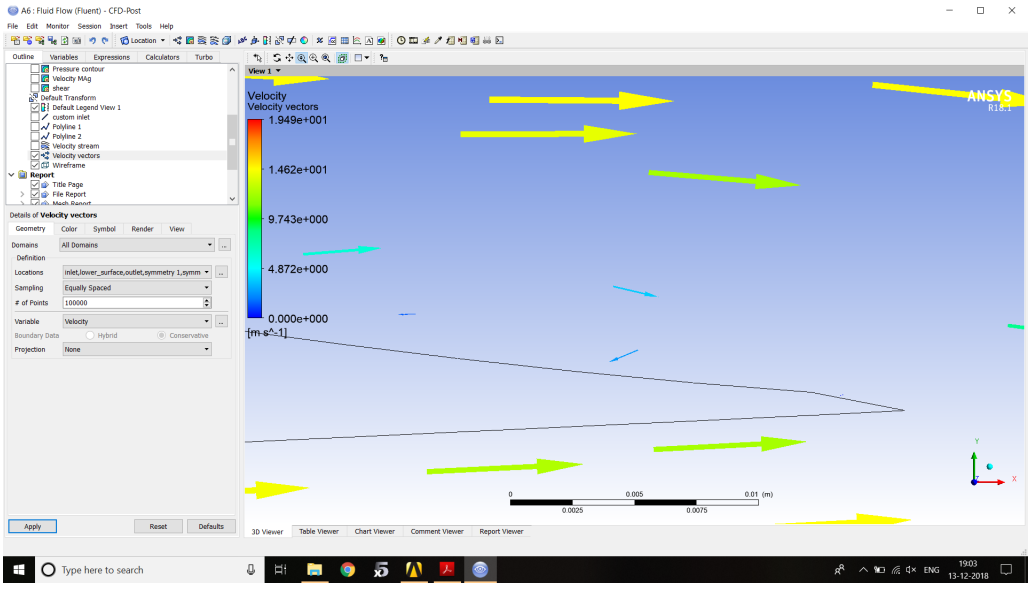




The above figure shows streamlines in case of turbulent method of analysis for both the angle of attacks.

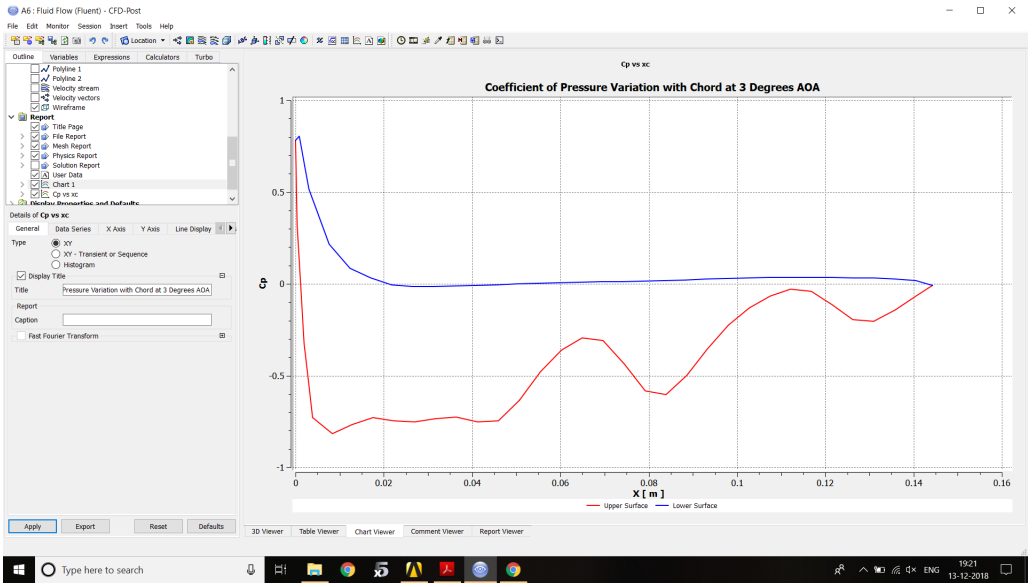


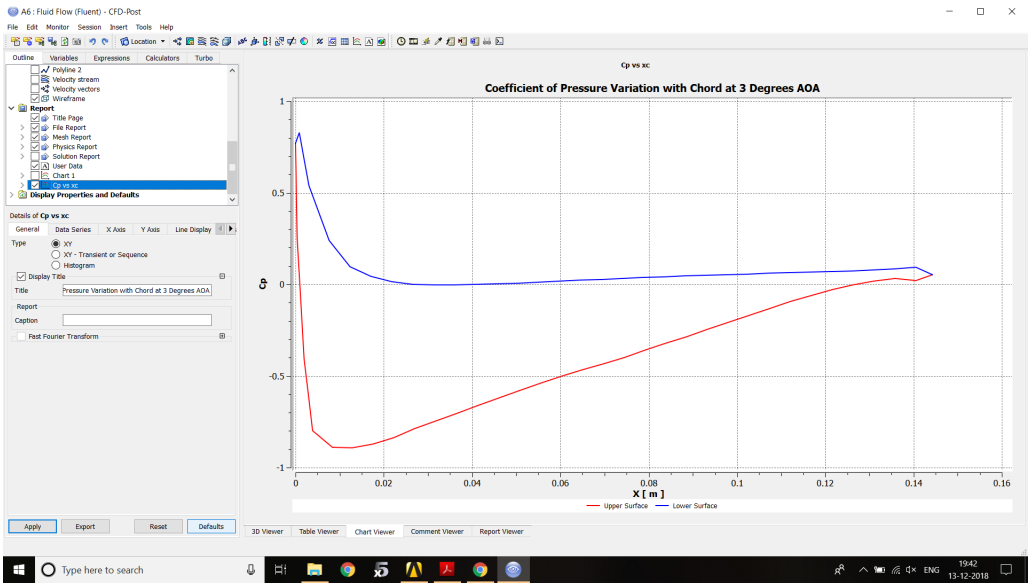
The above figure shows the velocity vector in case of turbulent flow at 0 degree angle of attack.



The above figure clearly illustrates the flow seperation in case of laminar flow analysis. In case of turbulent model, it is not clearly visible.

Cp vs x Plots on Ansys

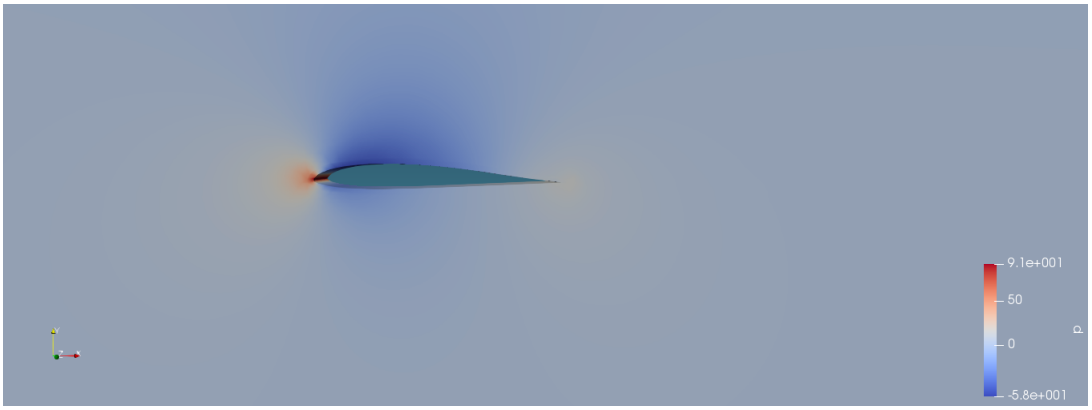




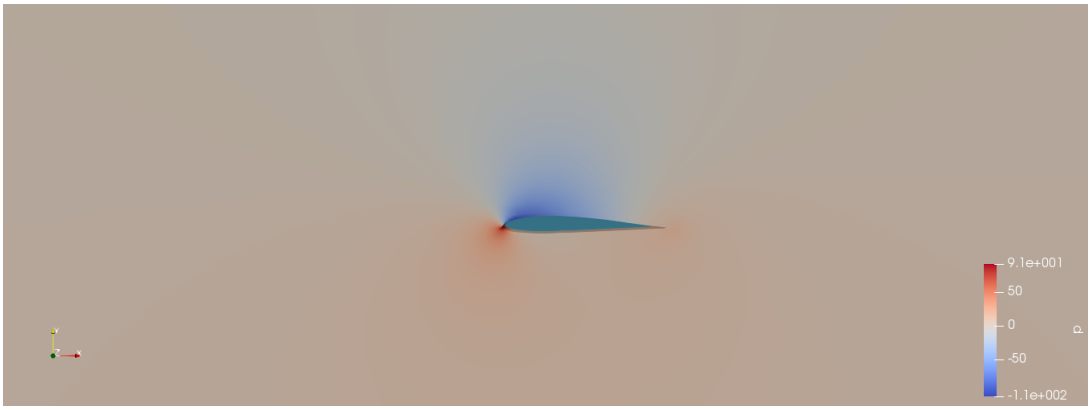
The above figures were obtained on ansys. The plots are for turbulent flow case for both the angle of attacks. The values obtained here are verified using XFLR5 which is discussed later.

Pressure Plots: Laminar flow at 0 and 3 degree angle of attacks in OpenFOAM

The same analysis and with same initial conditions were used in simulating the flow over the airfoil in OpenFOAM. The contour plots are shown below.



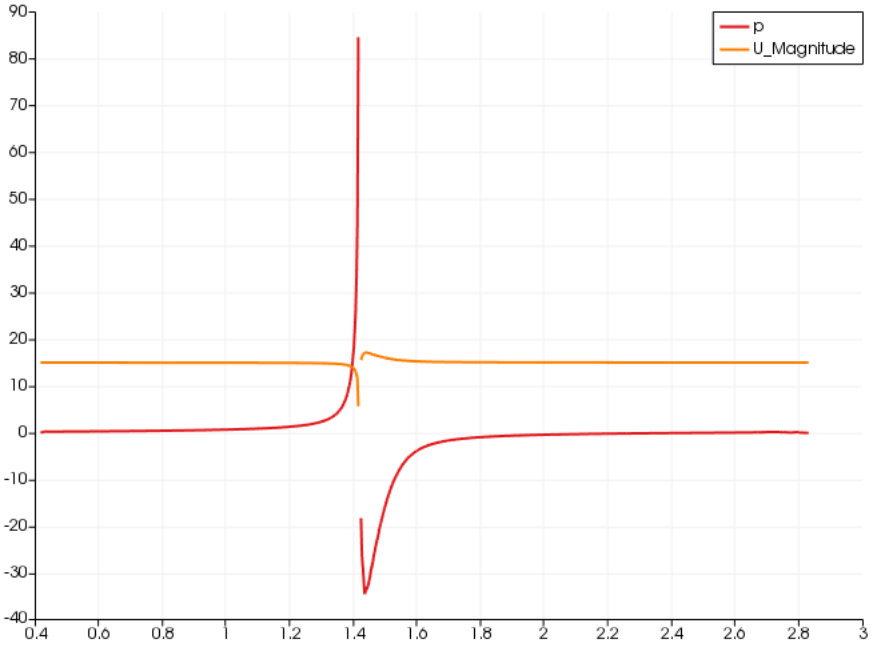
The figure shows the pressure contour for laminar flow at 0 degree angle of attack.

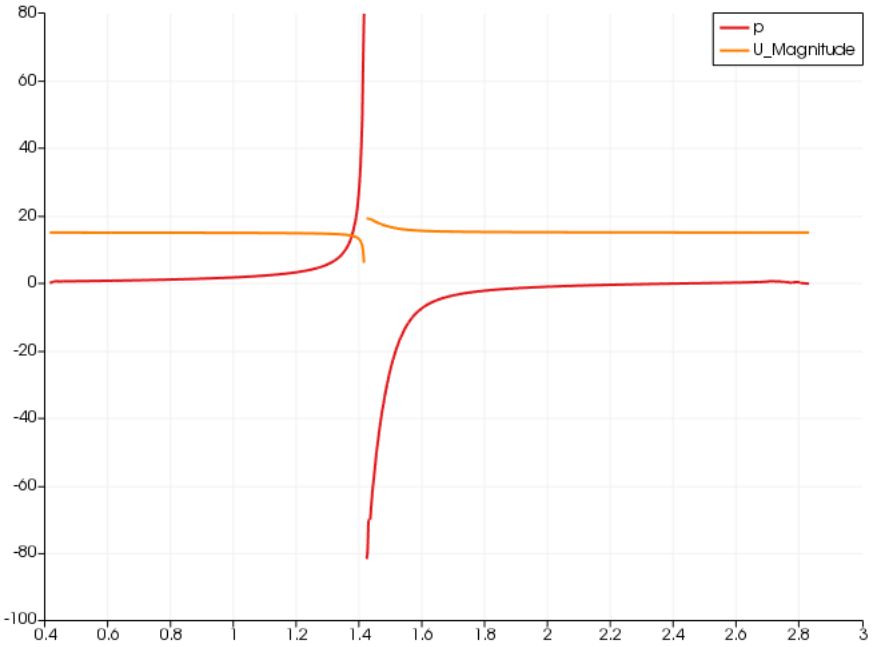


The figure shows the pressure contour for laminar flow at 3 degree angle of attack.

The figures can be compared with the contours obtained in Ansys. The pressure contour, especially of 3 degree angle of attack is similar to that obtained in Ansys. And so are the magnitudes.

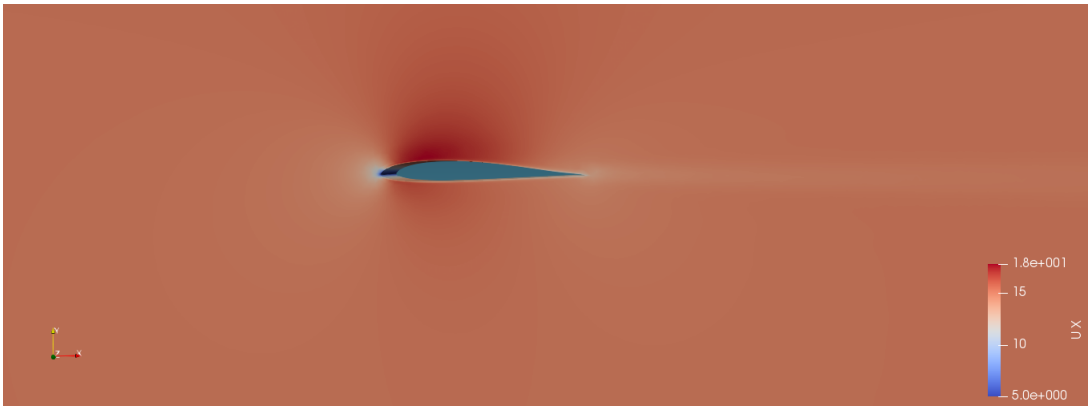
Next the variation of pressure for both the angle of attacks were obtained as a plot on openfoam, which bifurcates for both the surfaces of the airfoil.



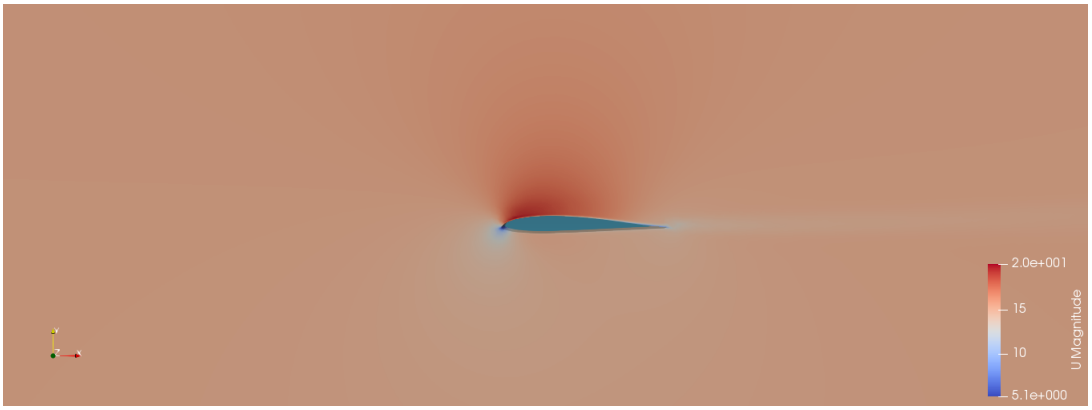


The first plot is for 0 degree angle of attack, while the second is for 3 degree. Notice that the pressure drop takes place in 3 degree angle of attack due to flow seperation.

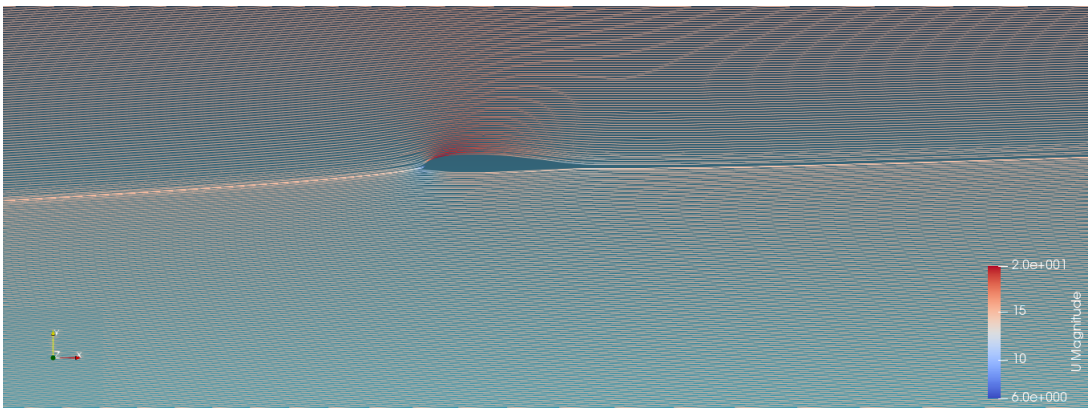
Velocity Plots: Laminar flow at 0 and 3 degree angle of attacks in OpenFOAM



The above figure is the contour for 0 degree angle of attack in a laminar flow.



In the above figure it is observed at 3 degree angle of attack, the contour seems *not-so-smooth again* and it's primarily because of the extra velocity component. This is highly comparable with the values and contour obtained on Ansys.



The above figure illustrates the stream-lines over the airfoil for 3 degree angle of attack in laminar flow case. The lines are finer than that obtained on Ansys.

Verification in XFLR5

To verify the results of lift and drag coefficients obtained by ANSYS FLUENT, we used a software called XFLR5, which is an airfoil and airplane analysis software based on the XFoil code.

XFLR5 uses vortex lattice methods to determine the various airfoil performance parameters for a given Reynolds Number or a list of Reynolds Numbers. Also, these results were then cross-checked with data available on the UIUC Airfoil Database. The version used for XFLR5 was 6.40.

Reynolds Number for the CFD analysis was obtained first.

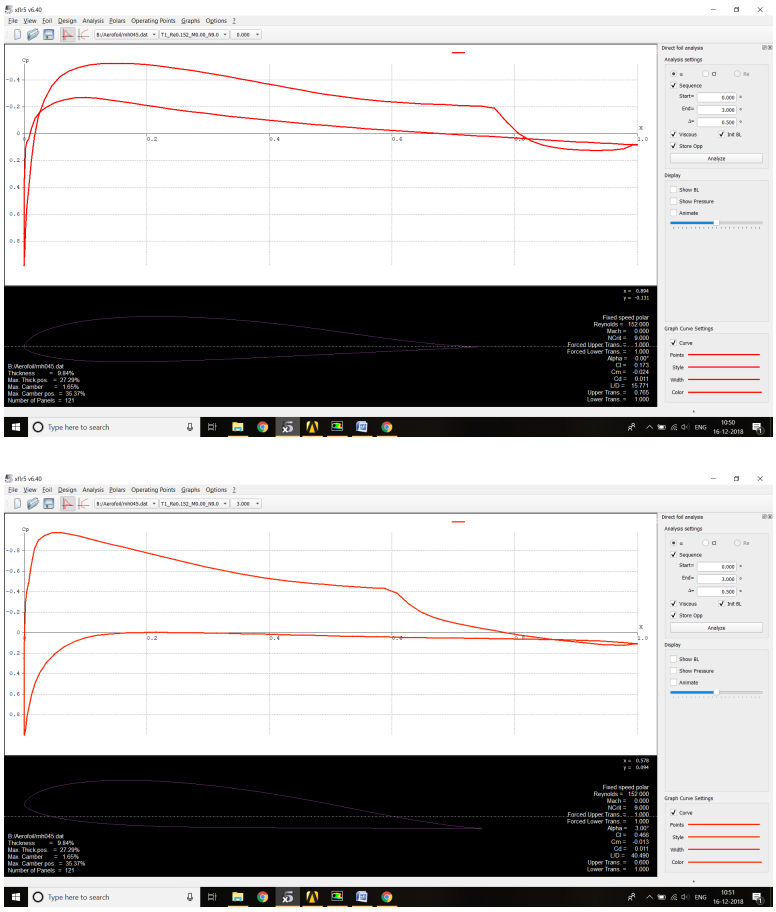
Where, C=Chord (0.15m), viscosity=1.81*(10e-5), Density=1.225 kg/m3.

Reynolds Number was found to be 152279. An analysis was carried out in XFLR5 for the said Reynolds Number for our angles of attack i.e. 0 and 3 degrees. The results were as shown below. Also, the variation of Cp vs x/c was plotted, similar to what was plotted on ANSYS CFD-Post.

Angle of Attack	Lift Coefficient	Drag Coefficient
0	0.173	0.0109
3	0.460	0.0115

It is evident that the lift coefficient results are quite close to those obtained from ANSYS FLUENT, but the results of drag coefficients are offset by a large amount, slightly lesser than half of what was obtained from FLUENT.

Therefore, we verified the results from the UIUC Airfoil Database and found them to be coincident with those of XFLR5.



The above figure shows the plots of Cp vs X obtained on XFLR5.

It was hence concluded that a straightforward methodology is enough to obtain an agreeable value of lift coefficient for an airfoil, but for the drag coefficient, many errors and variables need to be taken into account to obtain an accurate result.

Further, it was also found that although the analyses conducted under turbulence models converged, the results were substantially averaged out, and hence, reliable only to a certain extent.

0.9 Conclusion

The airfoil MH45 was analysed in FLUENT and on OpenFOAM for two different angle of attacks, namely 0 and 3 degrees. Both turbulent (kEpsilon) and laminar models were used.

The contours for pressure and velocity were obtained on both the softwares and compared. The contours were sufficiently similar, so were the streamlines. Flow seperation and velocity vector were later obtained on FLUENT.

Then Cp vs x plots for both the angle of attacks were obtained on FLUENT and the values of lift coefficent, and drag coefficient were compared with the software XFLR5, due to convergence in case of turbulence models, the reliability decreased.

It was also observed that the flow seperation could not be obtained for turbulent modeling, only laminar.

0.10 Acknowledgment

We would like to thank our Professor Mr.Devaraj.K for the course on incompressible fluid dynamics which was also an introductory course into this field of Aerodynamics.

The knowledge gained there greatly enhanced our understanding of airfoil flows and hence providing support towards entire process of this project.