

# NACA 0012 AIRFOIL TURBULENCE VALIDATION USING OPENFOAM & SALOME

Dinesh Poluru Baskar

## Contents

Abstract:.....	2
Introduction: .....	2
Zero-Degree AOA .....	2
Salome Mesh Creation:.....	2
Far Field:.....	2
Mesh: .....	3
Case Setup:.....	5
Results:.....	6
10-degree AOA.....	8
Mesh: .....	8
Case Setup:.....	9
Results:.....	9
Results Comparison between ANSYS and OpenFOAM:.....	10
Velocity Comparison: .....	11
Pressure Comparison: .....	12
Conclusion:.....	13
Figure 1: Far Field.....	3
Figure 2: Mesh. ....	3
Figure 3: Number of segments_100 .....	4
Figure 4: Error while compiling.....	4
Figure 5: Number of segments_150. ....	5
Figure 6: Patches. ....	5
Figure 7: Run time.....	6
Figure 8: Velocity magnitude.....	7
Figure 9: Pressure magnitude. ....	7
Figure 10: Mesh 10-degree AOA. ....	8
Figure 11: Converged solution. ....	9
Figure 12: Velocity magnitude.....	10
Figure 13: Pressure magnitude. ....	10
Figure 14: Velocity from OpenFOAM.....	11
Figure 15: Velocity from ANSYS.....	11
Figure 16: Pressure from OpenFOAM.....	12
Figure 17: Pressure from ANSYS. ....	12

## Abstract:

The following report is a detailed explanation of NACA 0012 airfoil turbulence validation. OpenFOAM and paraview software's are used for initializing the solvers and post processing respectively. This gives the detailed comparison of NASA validation data over the simulated data. This report also provides a detailed comparison between ANSYS and OpenFOAM simulation results.

## Introduction:

The project is based on validation of pressure and velocity magnitude on the airfoil using OpenFOAM software. The airfoil geometry and Mesh are created using **Salome Meca** software which is third party application for creating user defined mesh for the geometry. The pre-defined mesh has been deleted for better results and new mesh has been created. The simulation is done for both 0-degree angle of attack (AOA) and 10-degree angle of attack. The simulation is run by using **SimpleFoam**.

## Zero-Degree AOA

### Salome Mesh Creation:

Salome is a third-party application for creation of better mesh over a geometry for OpenFOAM. The pre-defined mesh has been deleted and new mesh with specified data has been created and the simulation is set to run. The creation of mesh and their failures have been discussed below.

### Far Field:

The Far Field is an important parameter one should consider before doing the simulation. The Far Field is similar to the wind tunnel one can find in NASA AMES research center. The simulation here in this problem is done outside the geometry, as the flow properties are studied on the body of the object. The mesh is also represented outside the body and inside the Far Field.

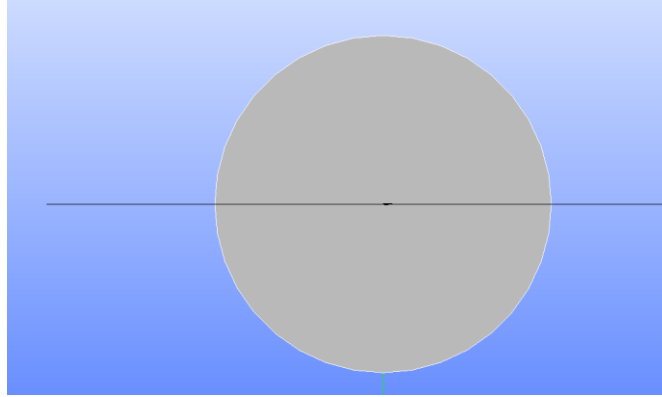


Figure 1: Far Field.

The Far Field here created is two disks placing the geometry on the center and the disks placed perpendicular to each other so that the geometry can be separated into two faces. This is done because to separate the faces of the airfoil. The Far Field is created in disk shape because the use of mesh type as **Quadrangle Mapping** is pretty much better in circle shapes.

### Mesh:

The mesh creation is quite a challenging process since the mesh should be finer near the airfoil and coarser on the outer Far Field. In order to create the desired type of mesh, sub-meshing has been implemented on the right and left side of the airfoil.

This sub-mesh is created with **Wire Discretization** algorithm with the number of segments to be 100 and scaling factor to be 100. But this results in an error while uploading it to the OpenFOAM display.

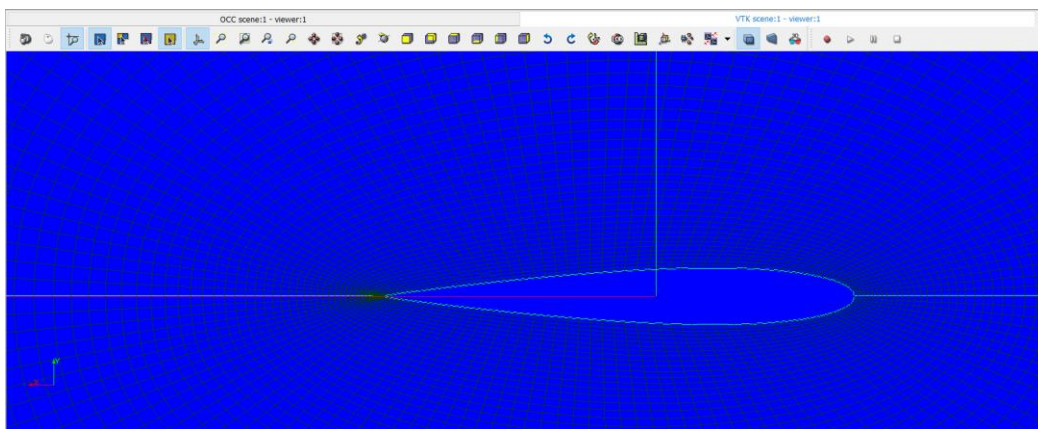


Figure 2: Mesh.

From the above figure, we can see that the discretization near the airfoil is high when compared to the outer region.

A number of attempts has been made for the mesh to be accepted by the OpenFOAM solver, since the discretization in the has been changed according to it. At first the number of segments and discretization is kept as 100 and 100 respectively, which is shown below in the figure.

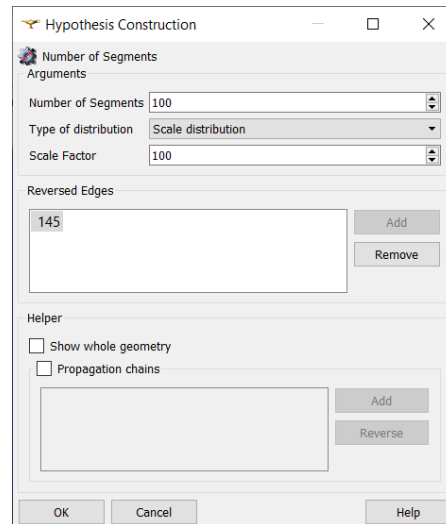


Figure 3: Number of segments\_100

```
Create polyMesh for time = 0
time = 0
Mesh stats
  points: 28860
  internal points: 0
  faces: 57306
  internal faces: 27954
  cells: 14292
  faces per cell: 5.96558
  boundary patches: 1
  point zones: 0
  face zones: 0
  cell zones: 0
Overall number of cells of each type:
  hexahedra: 13800
  prisms: 492
  wedges: 0
  pyramids: 0
  tet wedges: 0
  tetrahedra: 0
  polyhedra: 0
Checking topology...
  Boundary definition OK.
  Cell to face addressing OK.
  Point usage OK.
  Upper triangular ordering OK.
  Face vertices OK.
  Number of regions: 1 (OK).
Checking patch topology for multiply connected surfaces...
  Patch      Faces    Points  Surface topology
  defaultFaces 29352  28860  ok (closed singly connected)
Checking geometry...
  Overall domain bounding box (-10 -9.99988 0) (10 9.99988 0.1)
  Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
  Mesh has 3 solution (non-empty) directions (1 1 1)
  Boundary openness (-9.06912e-20 -2.54367e-18 -3.936e-19) OK.
  Max cell openness = 2.15702e-16 OK.
  Max aspect ratio = 578.811 OK.
  Minimum face area = 7.87476e-07. Maximum face area = 0.185769. Face area magnitudes OK.
  Min volume = 7.87476e-08. Max volume = 6.0185769. Total volume = 31.4124. Cell volumes OK.
  Mesh non-orthogonality Max: 66.1681 average: 12.4121
  Non-orthogonality check OK.
  Face pyramids OK.
  ***Max skewness = 6.80488, 6 highly skew faces detected which may impair the quality of the results
  <<skirting 0 skew faces to set skewFaces
  Coupled point location match (average 0) OK.
Failed 1 mesh checks.
end
```

Figure 4: Error while compiling.

This error message has arrived since the skewness of the mesh is higher while providing the values. This has been changed and provided with 150 and 600 number of segments and discretization respectively.

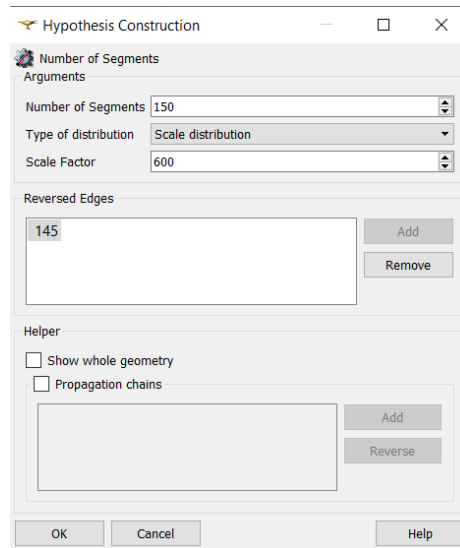


Figure 5: Number of segments\_150.

The change of number of segments and scale factor has been used and the OpenFOAM accepted the mesh. Since OpenFOAM uses only 3D mesh geometry an **extrusion** of **0.05** is given to the geometry.

## Case Setup:

The case setup for the OpenFOAM is done in 0 and constant folder of the airfoil case. The initial velocity, pressure case and boundary settings have been changed. The geometry is set up with patches which is carried over to the airfoil case setup which are shown below.

```

10 version      2.0;
11 format       ascii;
12 class        polyBoundaryMesh;
13 location     "constant/polyMesh";
14 object       boundary;
15 }
16 //
17
18 4
19 (
20   Face1
21   {
22     type      empty;
23     nFaces    21192;
24     startFace 41754;
25   }
26   FarField
27   {
28     type      patch;
29     nFaces    630;
30     startFace 62946;
31   }
32   AirFoil
33   {
34     type      wall;
35     nFaces    138;
36     startFace 63570;
37   }
38   Face2
39   {
40     type      empty;
41     nFaces    21192;
42     startFace 63714;
43   }
44 )

```

Figure 6: Patches.

## Boundary Conditions:

- Incompressible flow ( $Ma < 0.3$ )
- Reynolds Number 6 million.
- Velocity ( $u$ ) = 88.65 m/s.
- Density ( $\rho$ ) = 1.225 kg/m<sup>3</sup>.
- Length ( $L$ ) = 1m.
- $\mu = 1.81 \times 10^{-5}$  kg/m-s.

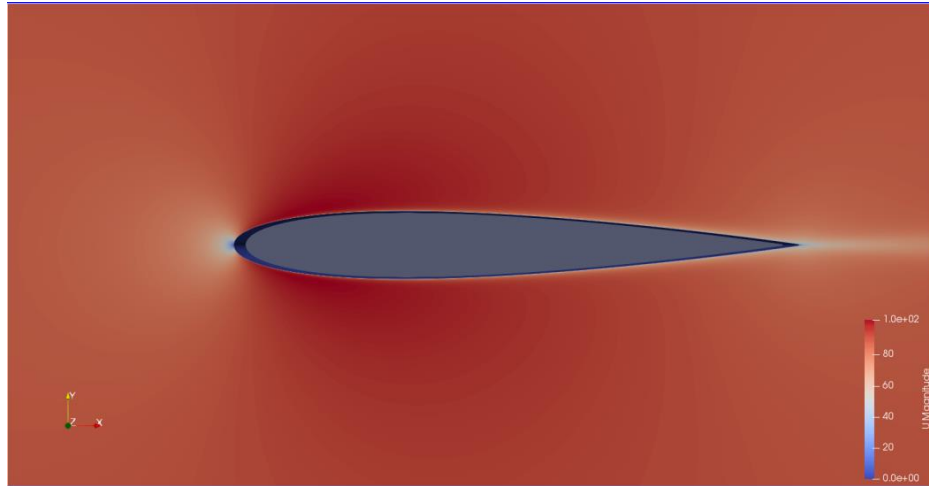
The Boundary conditions have been set up in the 0 folder and the Run time and iterations has been changed ControlDict which is shown below.

```
12 class dictionary;
13 location "system";
14 object controlDict;
15 }
16 // *****
17
18 application simpleFoam;
19
20 startFrom startTime;
21
22 startTime 0;
23
24 stopAt endTime;
25
26 endTime 1000;
27
28 deltaT 1;
29
30 writeControl timeStep;
31
32 writeInterval 100;
33
34 purgeWrite 1;
35
36 writeFormat ascii;
37
38 writePrecision 6;
39
40 writeCompression off;
41
42 timeFormat general;
43
44 timePrecision 6;
45
46 runTimeModifiable true;
47
```

Figure 7: Run time.

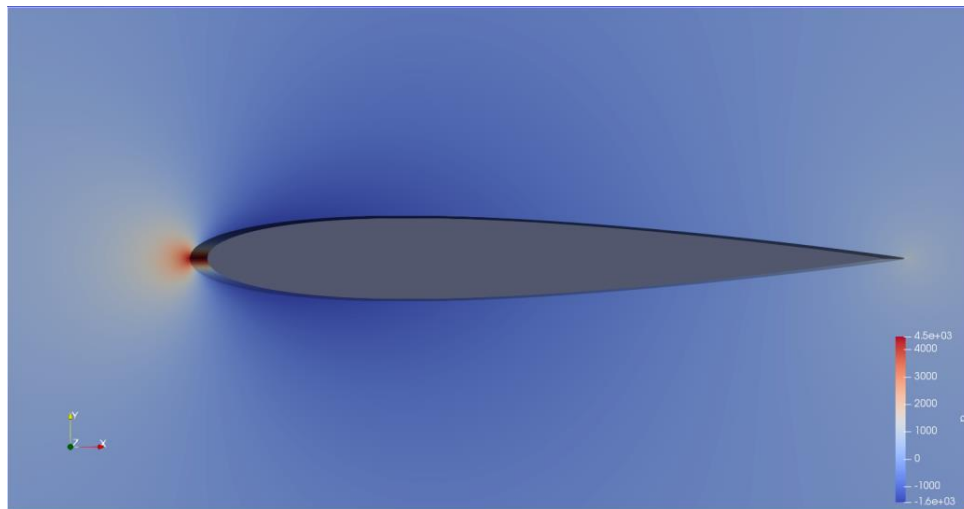
## Results:

The simulation is done by simple typing SimpleFoam in terminal window. SimpleFoam is a steady-state solver for incompressible and turbulent flows. The iteration has been done for 1000 steps and the results are observed and demonstrated below.



*Figure 8: Velocity magnitude.*

From the picture above we can see that the velocity on the front nose of the airfoil is minimum as the pressure is maximum (Bernoulli's principle). The outside wing is applied with maximum pressure on both sides.



*Figure 9: Pressure magnitude.*

The pressure magnitude is maximum at the front side of the nose as the simulation is done from left to right, the flow is from left to right side.

This velocity and pressure magnitude can be compared with the ANSYS solution in which the solution is similar to the OpenFOAM solution which has been show cased here.

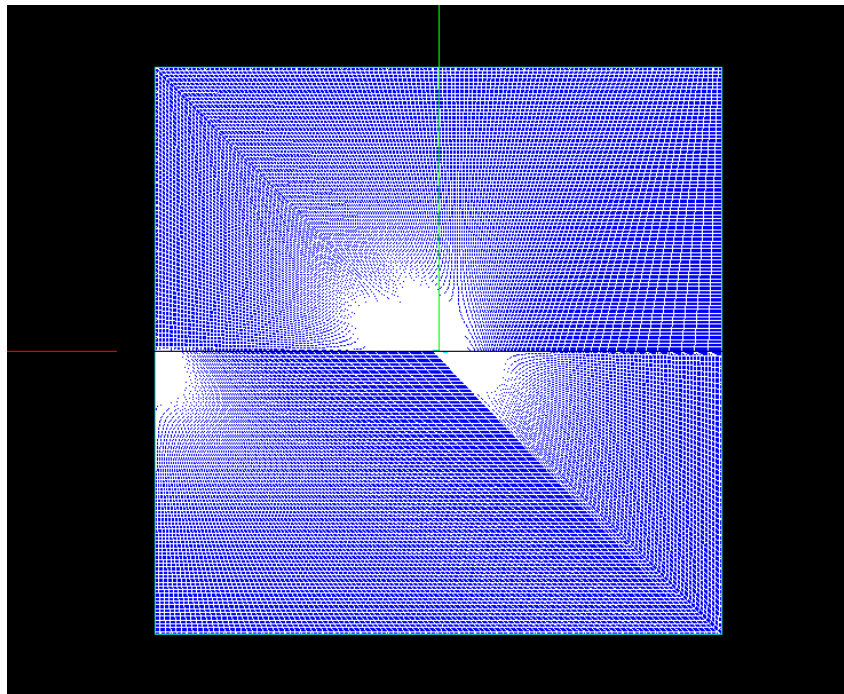


## 10-degree AOA

Now that we have seen the simulation results of airfoil with angle of attack zero. The angle of attack has been changed from the axis point of the airfoil in Salome software where because of its angular position the Far Field has been changed to square for a better meshing property.

### Mesh:

The geometry has been modified with respect to the AOA using Salome software. The angle is given from the axis point of the airfoil to the Z-axis. The mesh has been tried with many attempts to create a proper mesh using a circular Far Field, but as the geometry is not in symmetry with the axis the Far Field has been changed to square shape. The mesh used here is **Quadrangle mapping** and the sub-mesh has been created with same 100 number of segments and 600 discretization as zero-degree angle problem.



*Figure 10: Mesh 10-degree AOA.*

From the above picture we can see that the mesh is not properly arranged this is because of the non-symmetric geometry of the profile. But in case of the ANSYS Fluent which can handle complex geometry has done some pretty good job in meshing and simulation. But the results which came from both the software are almost similar.

## Case Setup:

The case setup for the 10-degree AOA is almost similar to the one which we had seen before. The velocity, boundary conditions are all the same and the run time for the simulation is the same too. The run time is set for 1000 iterations but the solution converged in 487 iteration.

```
time step continuity errors : sum local = 1.55007e-07, global = 1.69704e-08, cumulative = -0.000297074
smoothSolver: Solving for nuTilda, Initial residual = 3.85889e-07, Final residual = 2.02165e-08, No Iterations 4
ExecutionTime = 4.41 s ClockTime = 4 s

Time = 483
smoothSolver: Solving for Ux, Initial residual = 7.08526e-07, Final residual = 4.87936e-08, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1.34856e-06, Final residual = 1.89115e-07, No Iterations 4
GAMG: Solving for p, Initial residual = 1.82021e-05, Final residual = 1.85883e-06, No Iterations 1
time step continuity errors : sum local = 3.44897e-07, global = -1.32113e-08, cumulative = -0.000297887
smoothSolver: Solving for nuTilda, Initial residual = 3.80568e-07, Final residual = 1.99254e-08, No Iterations 4
ExecutionTime = 4.42 s ClockTime = 4 s

Time = 484
smoothSolver: Solving for Ux, Initial residual = 7.49131e-07, Final residual = 4.88666e-08, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1.32144e-06, Final residual = 1.87269e-07, No Iterations 4
GAMG: Solving for p, Initial residual = 1.82952e-05, Final residual = 4.88127e-07, No Iterations 2
time step continuity errors : sum local = 1.59959e-07, global = 1.59048e-08, cumulative = -0.000297871
smoothSolver: Solving for nuTilda, Initial residual = 3.75594e-07, Final residual = 1.96551e-08, No Iterations 4
ExecutionTime = 4.43 s ClockTime = 4 s

Time = 485
smoothSolver: Solving for Ux, Initial residual = 7.39698e-07, Final residual = 4.737e-08, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1.36021e-06, Final residual = 1.85755e-07, No Iterations 4
GAMG: Solving for p, Initial residual = 1.82333e-05, Final residual = 4.51892e-07, No Iterations 2
time step continuity errors : sum local = 1.47831e-07, global = 1.96338e-08, cumulative = -0.000297856
smoothSolver: Solving for nuTilda, Initial residual = 3.70409e-07, Final residual = 1.93763e-08, No Iterations 4
ExecutionTime = 4.44 s ClockTime = 4 s

Time = 486
smoothSolver: Solving for Ux, Initial residual = 7.38541e-07, Final residual = 4.67251e-08, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1.2955e-06, Final residual = 1.84889e-07, No Iterations 4
GAMG: Solving for p, Initial residual = 1.81444e-05, Final residual = 9.42944e-07, No Iterations 1
time step continuity errors : sum local = 3.22119e-07, global = -1.23797e-08, cumulative = -0.000297868
smoothSolver: Solving for nuTilda, Initial residual = 3.65495e-07, Final residual = 1.9185e-08, No Iterations 4
ExecutionTime = 4.45 s ClockTime = 4 s

Time = 487
smoothSolver: Solving for Ux, Initial residual = 7.19131e-07, Final residual = 4.5991e-08, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1.26941e-06, Final residual = 1.83039e-07, No Iterations 4
GAMG: Solving for p, Initial residual = 9.91877e-06, Final residual = 4.50699e-07, No Iterations 2
time step continuity errors : sum local = 1.47691e-07, global = 1.4312e-08, cumulative = -0.000297854
smoothSolver: Solving for nuTilda, Initial residual = 3.60737e-07, Final residual = 1.88479e-08, No Iterations 4
ExecutionTime = 4.45 s ClockTime = 4 s

SIMPLE solution converged in 487 iterations
End
dinesh@Dinesh-HP-Laptop-15g-br301:~/OpenFOAM/dinesh-8/run/atrFoil205
```

Figure 11: Converged solution.

This is because of the unmatched meshing of geometry, which can be possible rectify by replacing the geometry and the mesh type in a better way. But the solution when compared with the ANSYS solution it came almost similar to the solution with the OpenFOAM.

## Results:

Now that we have looked at the meshing and case setup for the 10-degree of AOA, the solution which arrived for 10-degree AOA is similar to the solution arrived with the ANSYS.

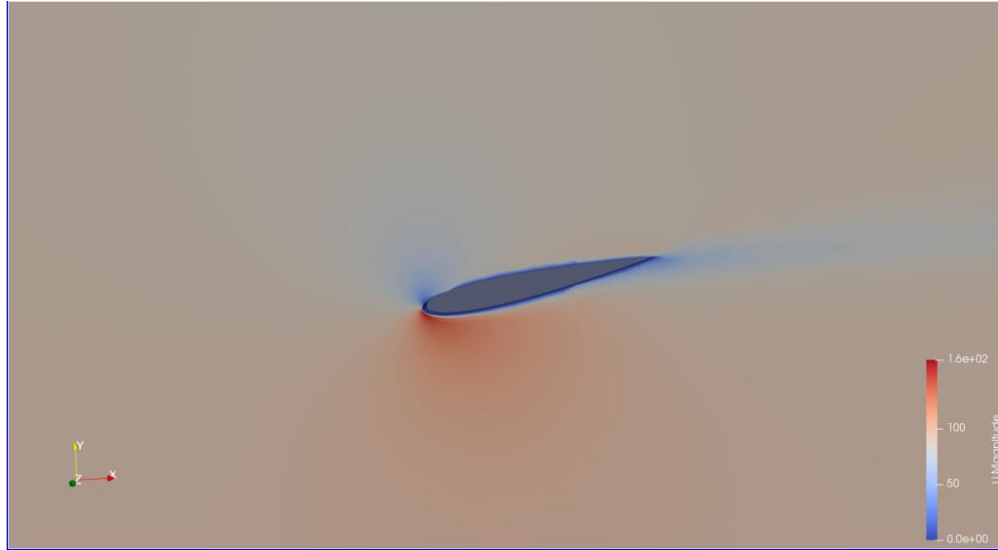


Figure 12: Velocity magnitude.

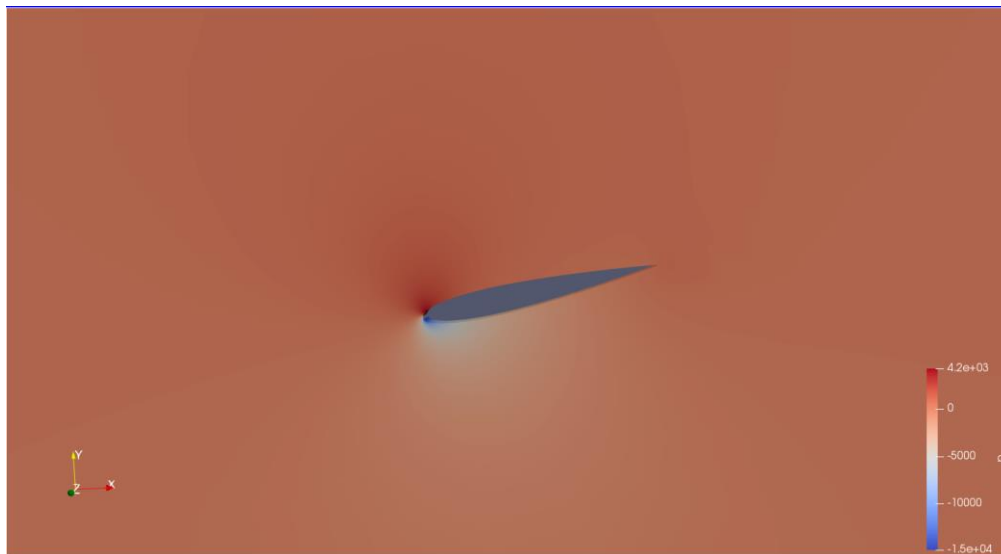


Figure 13: Pressure magnitude.

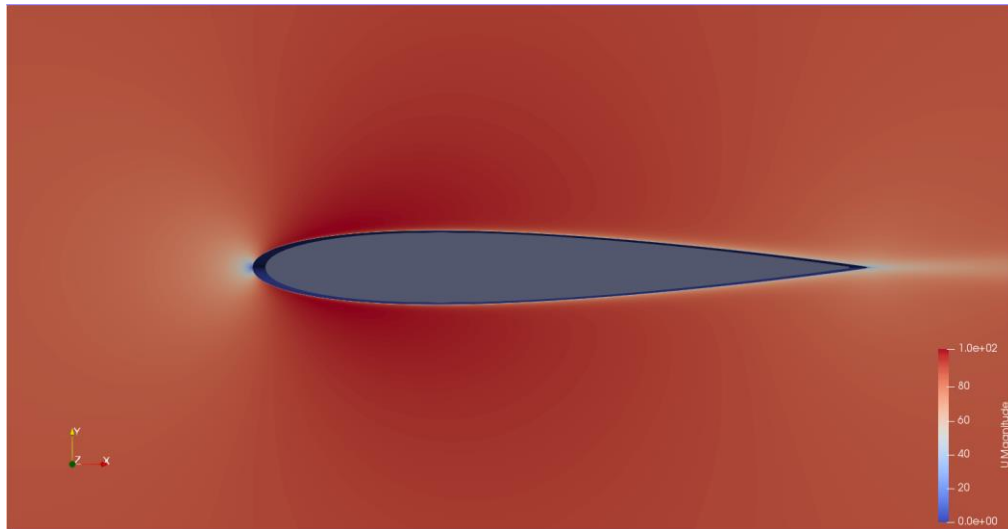
From the above two pictures we can see that both pressure and velocity are acting on both the cases, this is because of the angle between them. As the flow is exactly perpendicular to the y and z axis, the angle which the geometry makes gives that property.

Now that we have seen the 10-degree of AOA on the airfoil. As we mentioned the upcoming will be an results comparison between the ANSYS and OpenFOAM solutions.

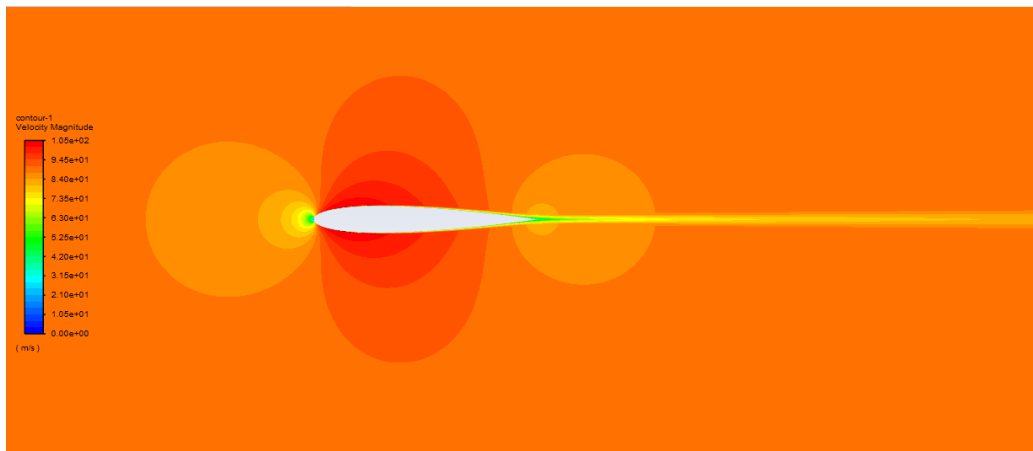
## Results Comparison between ANSYS and OpenFOAM:

This comparison of results between two software will make clear that which is more efficient and more time consuming or vice-versa.

### Velocity Comparison:



*Figure 14: Velocity from OpenFOAM.*



*Figure 15: Velocity from ANSYS.*

From the above pictures we can see that the velocity profile for both the cases are similar and the maximum values which are shown are almost similar in case. OpenFOAM : 1.0e02, whereas ANSYS : 1.05e2

## Pressure Comparison:

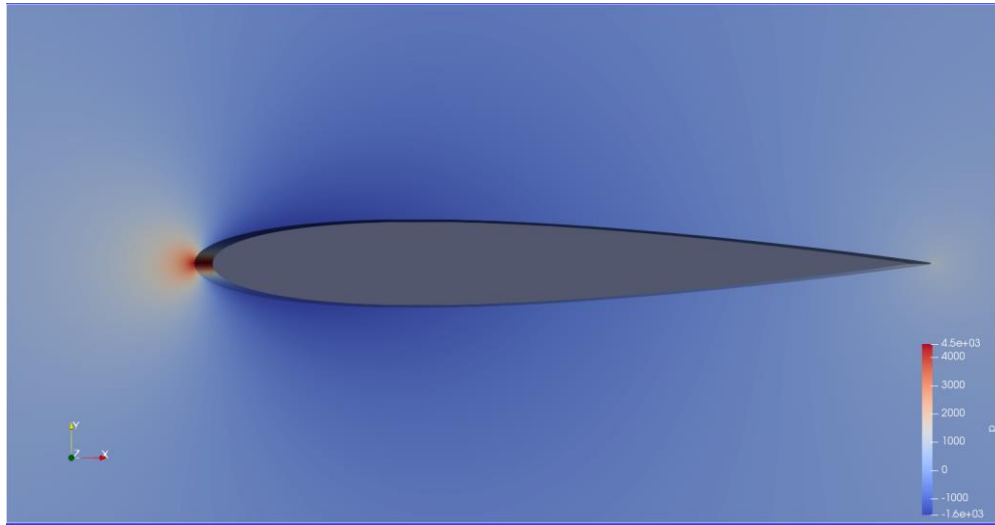


Figure 16: Pressure from OpenFOAM.

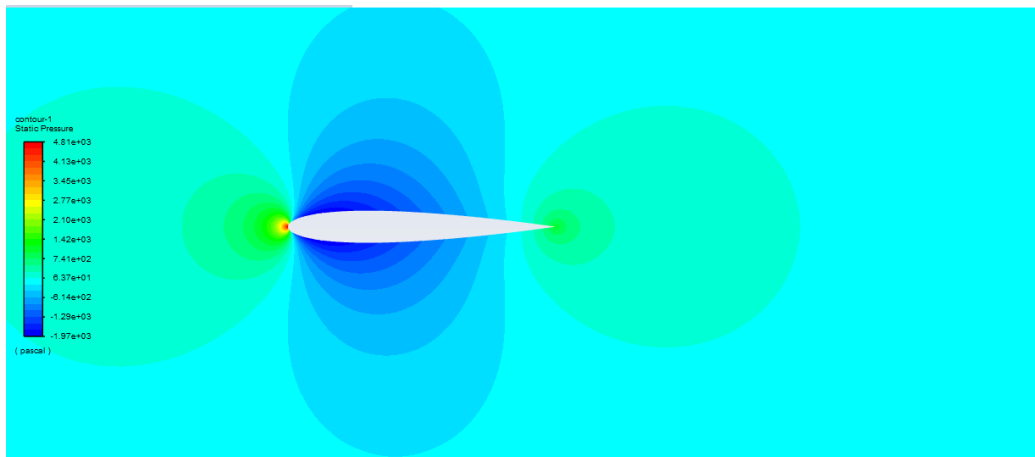


Figure 17: Pressure from ANSYS.

From the above pressure pictures, we can see that the results from both ANSYS and OpenFOAM are similar in values. This is the case for 10-degree of AOA, the same values arrived for both the cases.

Even though both the cases have same results but simulated in different software, there is a lot differences in the simulation process. While OpenFOAM is considered to be the software which can be used for small geometry and not so difficult problem solving, on the other hand ANSYS seems to be controlling the complex geometry easily.

But with the openFOAM while running the simulation it takes hardly 30-40 seconds to complete the simulation to run for 1000 iteration steps, while in case of ANSYS it took me more than 30 minutes to complete one simulation.

On the whole each software's has their own advantages and disadvantages and can be used as per the geometry provided for the problem.

### Conclusion:

From this project the procedure, results and comparison of NACA 0012 airfoil of different angle of attack and different software has been compared and provided with detailed report. From this conclusion ANSYS software is much more convenient in the case setup as well as post processing condition, but on the other hand OpenFOAM is an open-source software where everyone can use it for free. From this report I can prove that both the software does the problem solving in a manner that one can satisfy by their work.