DESIGN, ANALYSIS AND EXPERIMENTATION OF FRONT AND REAR WINGS OF FORMULA STUDENT RACE-CAR

Authors

Ahsan Javed Reg # 10-NUST-BE-ME-16

Ammar Mehmood Malik Reg # 10-NUST-BE-ME-22

Aftab Jan Reg # 10-NUST-BE-ME-07

M.Umer Jalees Alvi Reg # 10-NUST-BE-ME-68

Supervisor

Dr. Muhammad Sajid

DEPARTMENT OF MECHANICAL ENGINEERING

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

JUNE, 2014

Design, analysis and experimentation of front and rear wings of Formula Student race-car

Authors

Ahsan Javed - Reg # 10-NUST-BE-ME-16

Ammar Mehmood Malik - Reg # 10-NUST-BE-ME-22

Aftab Jan - Reg # 10-NUST-BE-ME-07

M.Umer Jalees Alvi - Reg # 10-NUST-BE-ME-68

A thesis submitted in partial fulfillment of the requirements for the degree of

BE Mechanical Engineering

Thesis Supervisor's Signature

Dr. Muhammad Sajid

Co-Supervisor's Signature

Dr. Samiur Rahman Shah

Committee member's Signature

Dr. Nabeel Anwar

Principle SMME Signature

Declaration

I certify that this research work titled "*Design, analysis and experimentation of front and rear wings of formula student race-car"* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources, it has been properly acknowledged / referred.

Signature of Students

Ahsan Javed
10-NUST-SMME-BE-ME-16

Ammar Mehmood Malik
10-NUST-SMME-BE-ME-22

Aftab Jan
10-NUST-SMME-BE-ME-09

M.Umer Jalees Alvi
10-NUST-SMME-BE-ME-68

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Students

Ahsan Javed
10-NUST-SMME-BE-ME-16

Ammar Mehmood Malik
10-NUST-SMME-BE-ME-22

Aftab Jan
10-NUST-SMME-BE-ME-09

M.Umer Jalees Alvi
10-NUST-SMME-BE-ME-68

Signature of Supervisor

Copyright Statement

Acknowledgements

I am thankful to my Creator Allah Subhana-Watala to have guided me throughout this work at every step and for every new thought which You setup in my mind to improve it. Indeed I could have done nothing without Your priceless help and guidance. Whosoever helped me throughout the course of my thesis, whether my parents or any other individual was Your will, so indeed none be worthy of praise but You.

I am profusely thankful to my beloved parents who raised me when I was not capable of walking and continued to support me throughout in every department of my life.

I would also like to express special thanks to my supervisor Dr. Muhammad Sajid for his help throughout my thesis and also for Fluid Mechanics I, II and Computational Fluid Dynamics courses which he has taught me. I can safely say that I haven't learned any other engineering subject in such depth than the ones which he has taught.

I would also like to pay special thanks to members of NOC and HPC Lab, OIC College of Aeronautical Engineering, Risalpur, Dean Mechanical Engineering GIKI for their tremendous support and cooperation in this project.

IT SMME helped us along the way. Through their support we completed our project on time. We specially thank IT SMME for their tremendous support

Finally, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Abstract

Increasing down-force is of main concern for race car aerodynamic engineers in motor sports. To increase the cornering stability and mechanical grip of formula race-cars, front, rear wings and under-body diffusers are mounted. These aerodynamic devices provide high down-force which keeps F-1 cars stick to the road surface and increase cornering speed. A wing has a profile of an airfoil which is mounted with a negative angle of attack with respect to free stream air to provide down-force (opposite to airplane wings which provide lift). Front wings are mounted on front at nose and rear wings are mounted at a height attached to the rear chassis. As speed increases, down force increases providing grip to race-cars. Wings can be single or multiple. Multiple wings can provide high level of down-force. In this project we surveyed 1700 airfoils out of which we selected top 170 airfoils. Out of these airfoils, 3 best airfoils were selected according to their lift to drag ratio. OpenFOAM (freeware) was used to for analysis of wings. 3D Mesh grid was made around airfoils with 2 million cells. These airfoils were plotted in Solid works and imported to OpenFOAM and analysis was done at multiple speeds and angle of attacks for single and multiple airfoils. As simulations took a lot of time therefore cluster was made in Experimentation and Computational Mechanics Lab of SMME. A 96 core cluster was made which performed rapid simulations and decreased total time to complete laminar and turbulent cases. Results were compared to actual results using wind tunnels at Aero Lab College of Aeronautical Engineering, Risalpur and Wind Tunnel Lab GIKI, Topi Swabi. C_l/C_d was plotted at different speeds to provide the actual amount of down force obtained using wings. Laminar and turbulent cases were compared and results were visualized in post processing using ParaView.

Key Words: *BlockMesh, SnappyHexMesh, Cluster processing, Turbulence Modeling, Parametric Modeling*

Table of contents

SECTION A

INTRODUCTION

BLOCKMESH ANALYSIS

Wings in race cars provide down force for cornering stability. While in aircrafts, wings provide lift. In this project, Open-foam analysis was done for Selig 1223, Selig 1223 RTL (Richard T. LaSalle modification of the S1223) and NACA 6412 (NACA series air foil). Data for these airfoils was taken from [1]. These air-foils were simulated at different speeds with varying angle of attacks. C_l, C_d coefficients were calculated. Horizontal, vertical velocities (Ux,Uy) and Pressure plots were plotted in Open-foam's post processor 'Para-view' . For simulation of airfoil, a 3D C-shaped dense mesh-grid was made around the S-1223 with 2.2 million hexahedral cells (See fig. 2, 3, 4 and 5). Airfoil surface points were taken from www.airfoiltools.com and these points were connected through splines to model the profile of airfoil. The blockMesh file is made in such a way that any airfoil can be modeled in Open-foam by replacing the points making up the profile with the points of a desired airfoil taken from above mentioned website. Control volume (blockMesh file) was defined for S-1223 with boundaries such as inlet, outlet and walls (airfoil profile). Initial boundary conditions were given by giving the free-stream air velocity, angle of attack of air, free-stream air pressure at inlet and outlet, zero velocity at surface of airfoil (No-slip condition). Mesh density is high near airfoil (Fig. 4 and 5). Mesh size is 12C, (chord length of airfoil) towards left, right, up and down from the origin (origin taken at the leading edge of S-1223) so that the effects of boundary layer created by airfoil lies inside the mesh and doesn't reach the boundaries of mesh.

DOMAIN OF MESH:

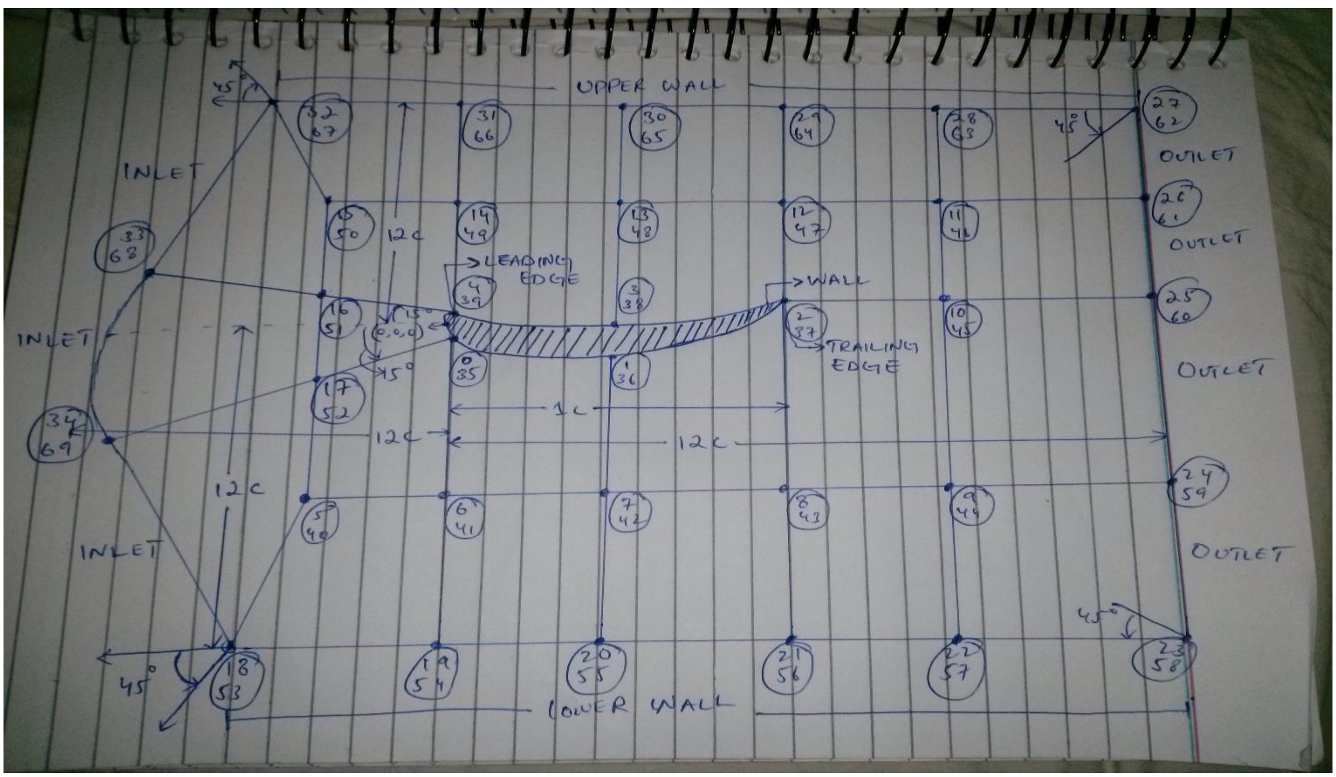


Fig. 1 Domain Box

Origin is taken at the leading edge of S-1223. Surface of airfoil consist of 5 splines connected to make the complete profile.

MESH OPTIMIZATION:

Most difficult part of this project was to remove mesh errors like orthognality, skewness, aspect ratio adjustment etc. The leading edge part of mesh had most skewness and high aspect ratio. Skewness was adjusted by changing cordinates of vertices “0, 35, 4, 39” by hit and trial method. Aspect ratio was adjusted by reducing the number of cells in y-direction in the leading edge part of the mesh. 15 degree angle was set for leading edge part of the mesh. Increasing or reducing this angle increases mesh skewness. Boundary vertices “27, 62, 32 67, 18, 53, 23, 58” were taken at 45 degrees from the origin.

MESH VISUALIZATION:

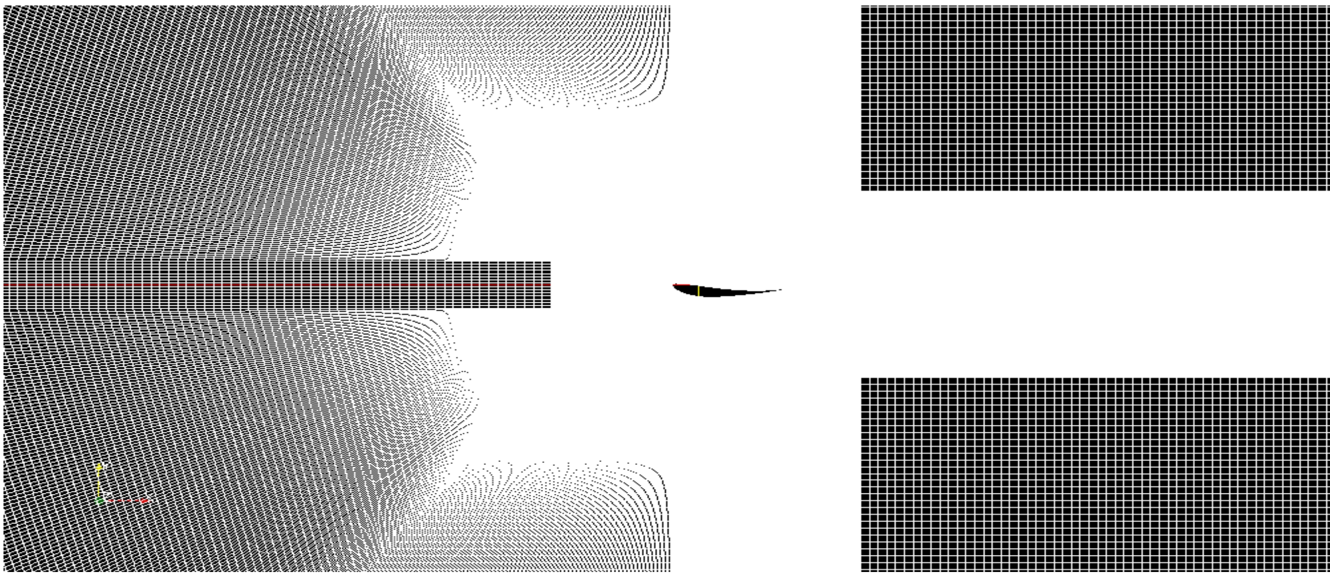


Fig. 2 far view S-1223

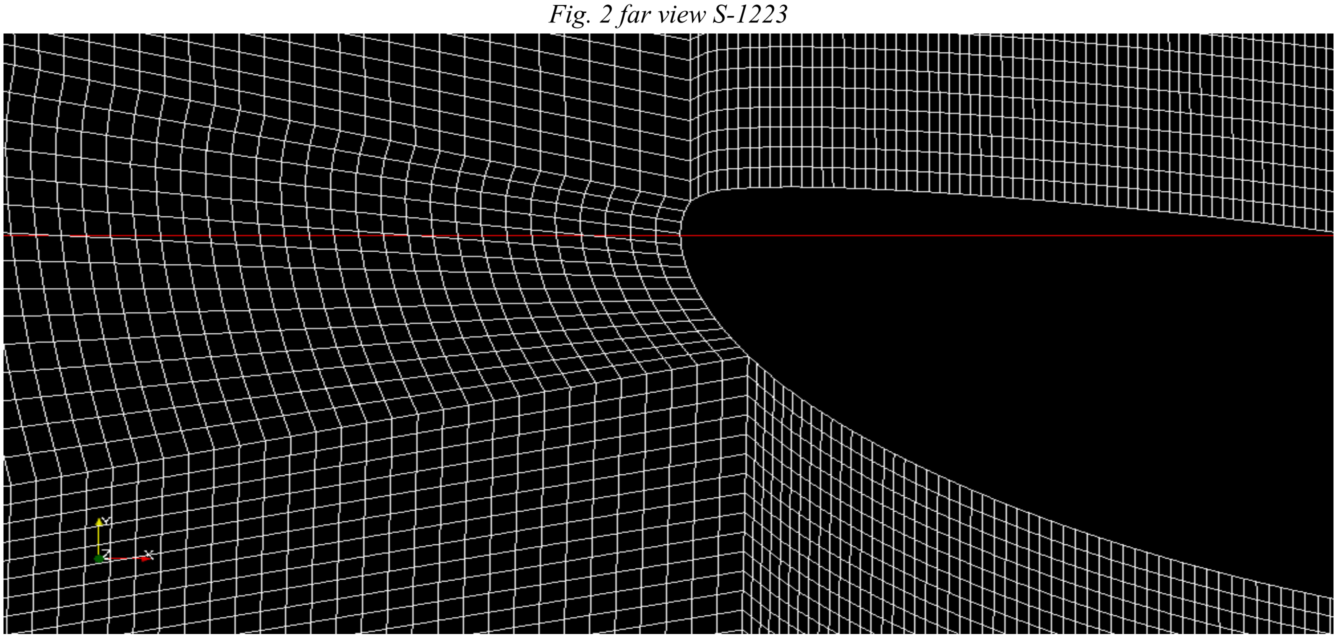


Fig. 3 Leading edge S-1223

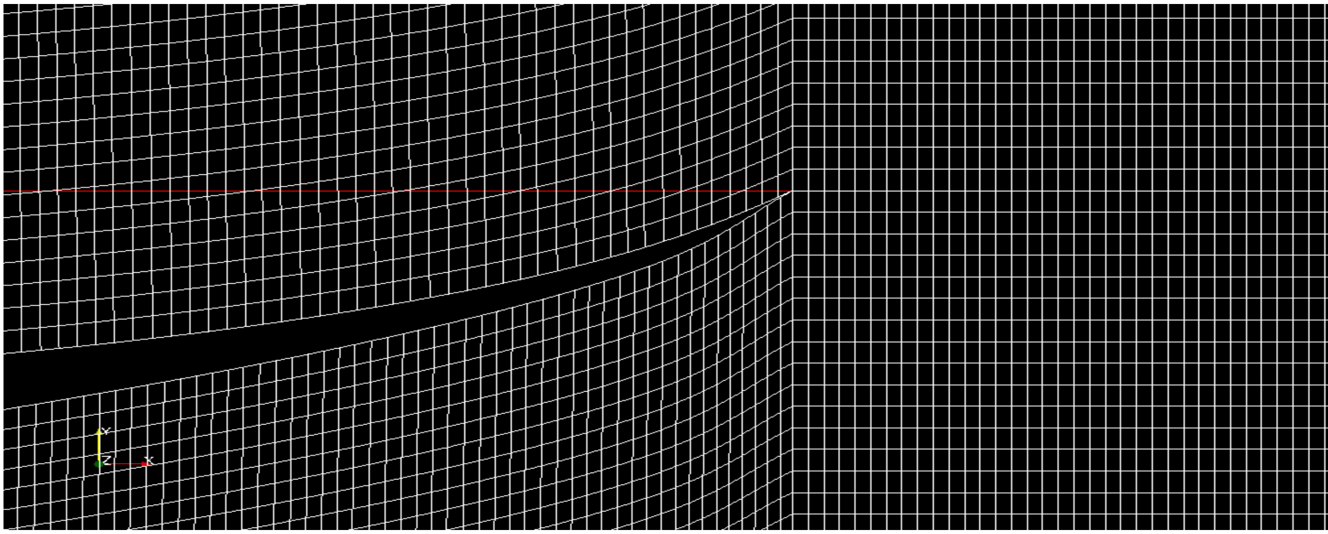


Fig. 4 Trailing edge S-1223

POST-PROCESSING:

SimpleFOAM solver was run by using parallel processing technique. My laptop is Core-i7 and has 4 main cores. Openfoam supports MPI (Message Pass Interface) parallel processing. When solver starts following graphs depicts the behavior of all 4 processors (Fig. 6)

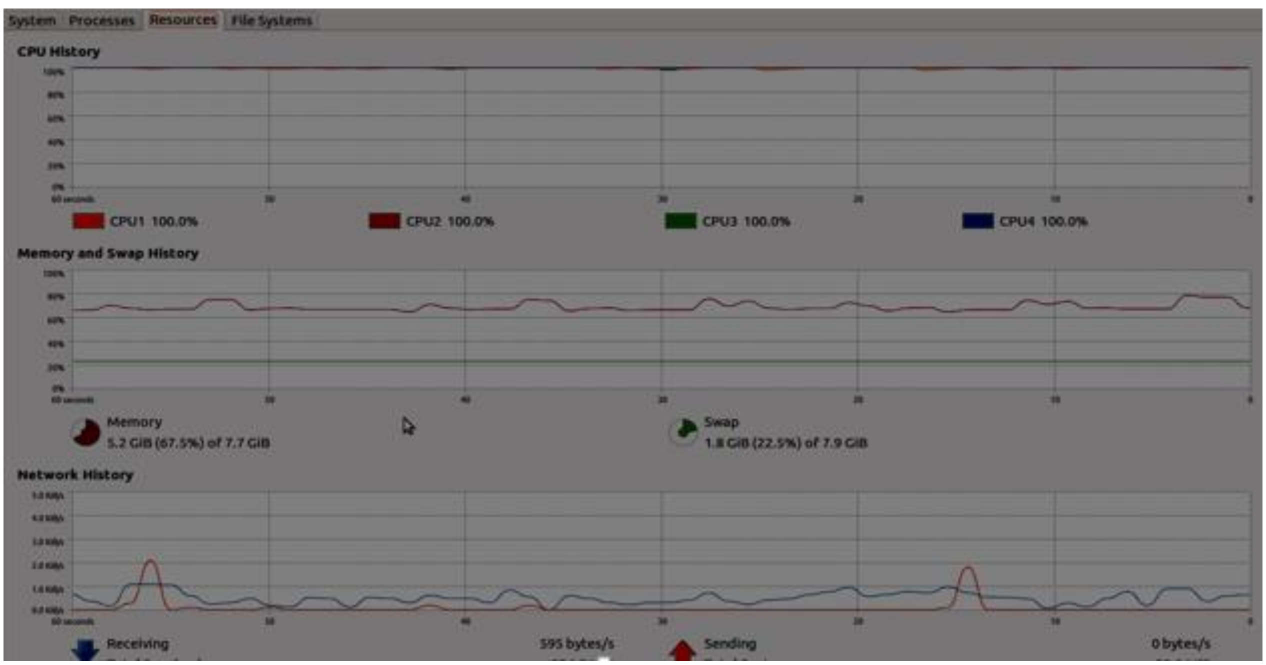


Fig. 5 parallel processing

For running a case in parallel, following command line in terminal is used to start the solver in parallel taken from [2]:

```
**************************************************
mpirun --hostfile hosts -np 4 simpleFoam -parallel > log &
**************************************************
```

All the processing is shown in a log file created in the case folder. A host file should be made in the case folder. It should include the name of the host pc across which parallel processing is done. In my case my hostname in “hosts” was:

ahsan@ahsan-Inspiron-5521
It can be found in /etc/hostname file.

ANALYSIS METHODOLOGY [3]

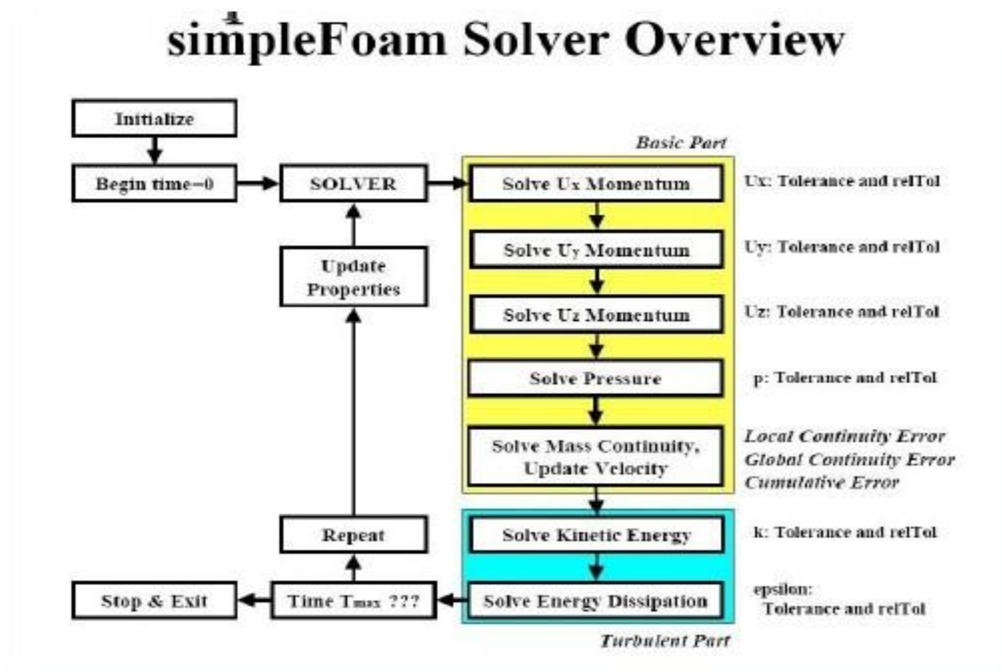


Fig. 6

RESULTS:

After mesh creation and given boundary conditions, simpleFoam solver was run to simulate the case. In each case either velocity was changed or angle of attack or an airfoil. Simulation time for each case took about 35 – 40 hours till solution reached a steady state and converged. "Write Interval" was kept 500 – 1000 as each case produced post processing results ranging in size from 70 – 80 Gbs. Simulations were done for some specific angle of attacks to reduce time and effort. These angle of attacks provided maximum Cl/ Cd values. They were taken from www.airfoiltools.com and our simulation results were also compared to the above mentioned website. As clear from figure 7,8,9,10,11 and 12, Pressure above airfoil is high as shown by red regions, while below, low pressure shown by blue and green regions.

DOWNFORCE CALCULATIONS:

$$L = \frac{1}{2}\rho v^2 A C_L$$

L=Lift force for our case its down force
A=frontal area of airfoil
Cl=Coefficient of lift
Rho= density of air
V= velocity of car

Following are OpenFOAM's results:

Sr. No	Table I Selig – 1223 No. of cells: 1.1 million Laminar flow Overall domain box volume = 4.2x4.2x1.2 m^3 upstream air velocity = 36.11 m/s = 130 km/hr Density of air = 1.204 kg/m ^{3} Area of wing = 0.25 x 1.2 = 0.3 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
1	-20	2.52603	N/A	0.229897	54.14	594.85
2	- 16	2.51032	2.3598	0.160219	37.73	591.15
3	-13	2.44132	2.3902	0.118652	27.94	574.90
	Table II Selig – 1223 Overall domain box volume = 4.2x4.2x1.2 m^3 No of cells: 2.5 million upstream air velocity = 36.11 m/s = 130 km/hr Density of air = 1.204 kg/m ^{3} Area of wing = 0.35 x 1.2 = 0.42 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
Laminar flow	-13	2.326	N/A	0.23	75.82	766.94
Turbulent flow	-13	1.6	2.3902	0.208	68.57	527.497
Laminar flow	-16	2.465	N/A	0.3	98.9	812.6
Turbulent flow	-16	1.705	2.3598	0.354	116.708	562.11
Laminar flow	-20	2.53	N/A	0.493	162.56	834.1
Turbulent flow	-20	0.2582	N/A	0.55	181.33	85.12

PARAVIEW RESULTS:

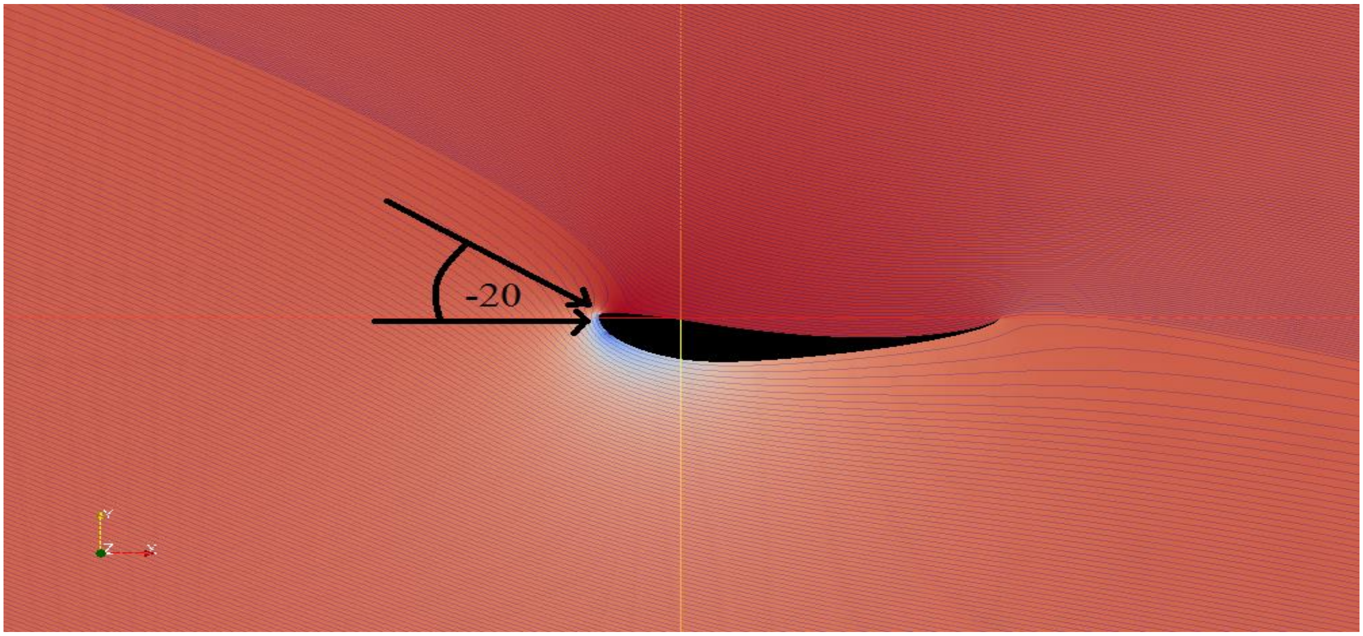


Fig. 7 Streamlines across S-1223 at -20 aoa

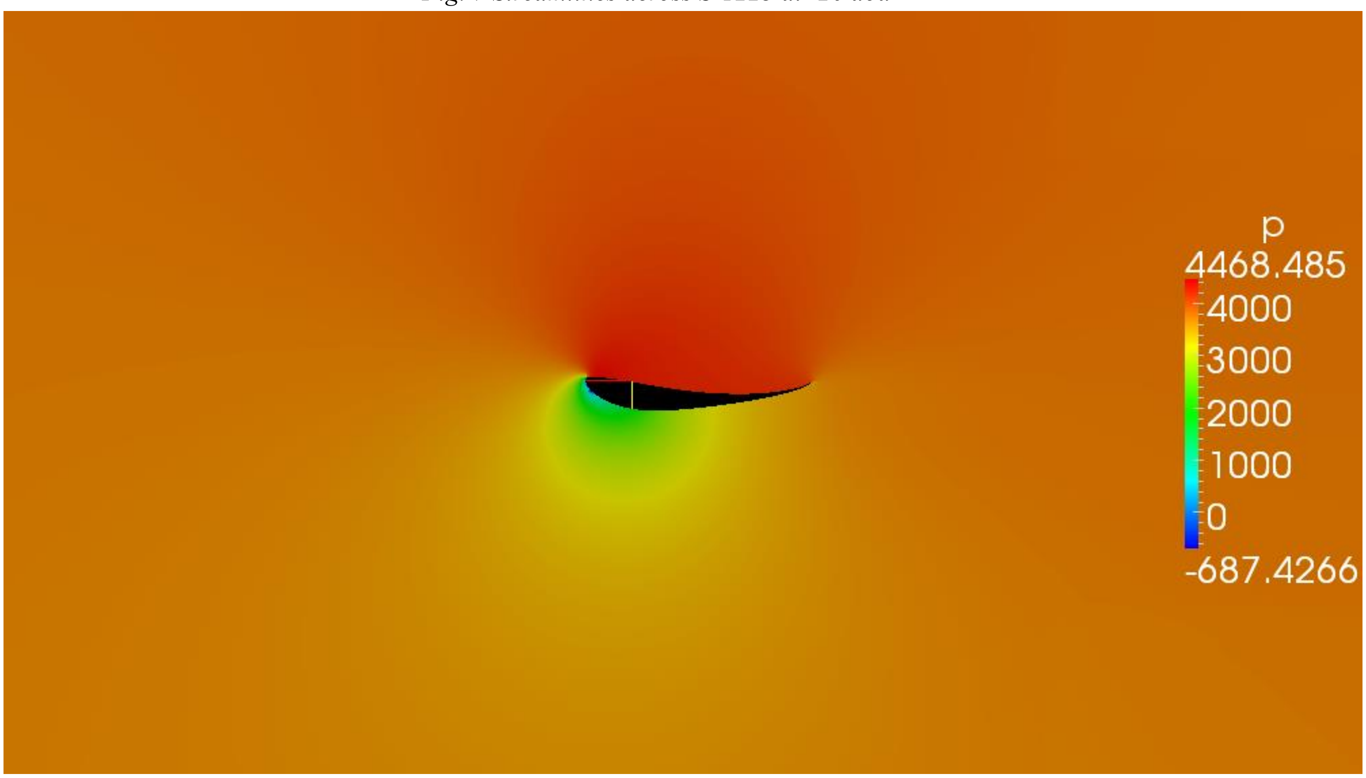


Fig. 8 Pressure plot around S-1223 at -20 aoa

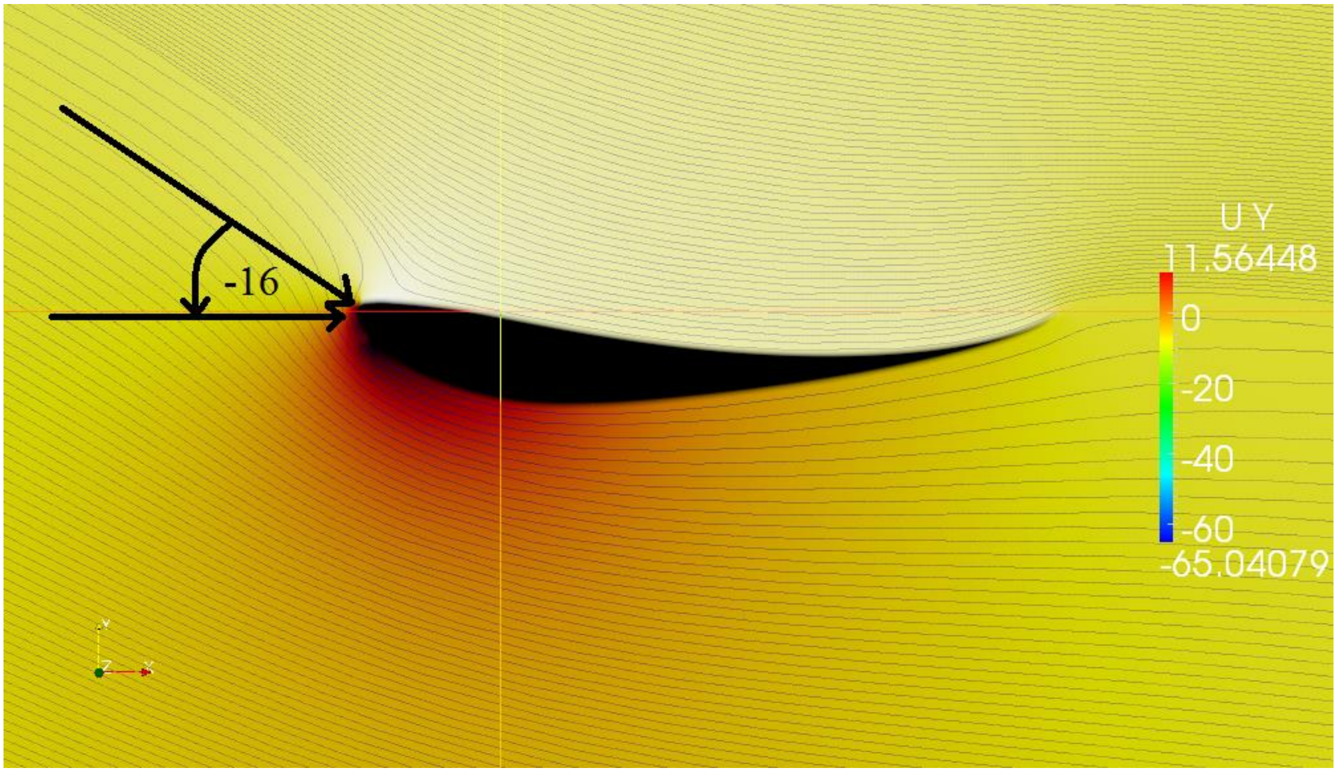


Fig. 9 Streamlines

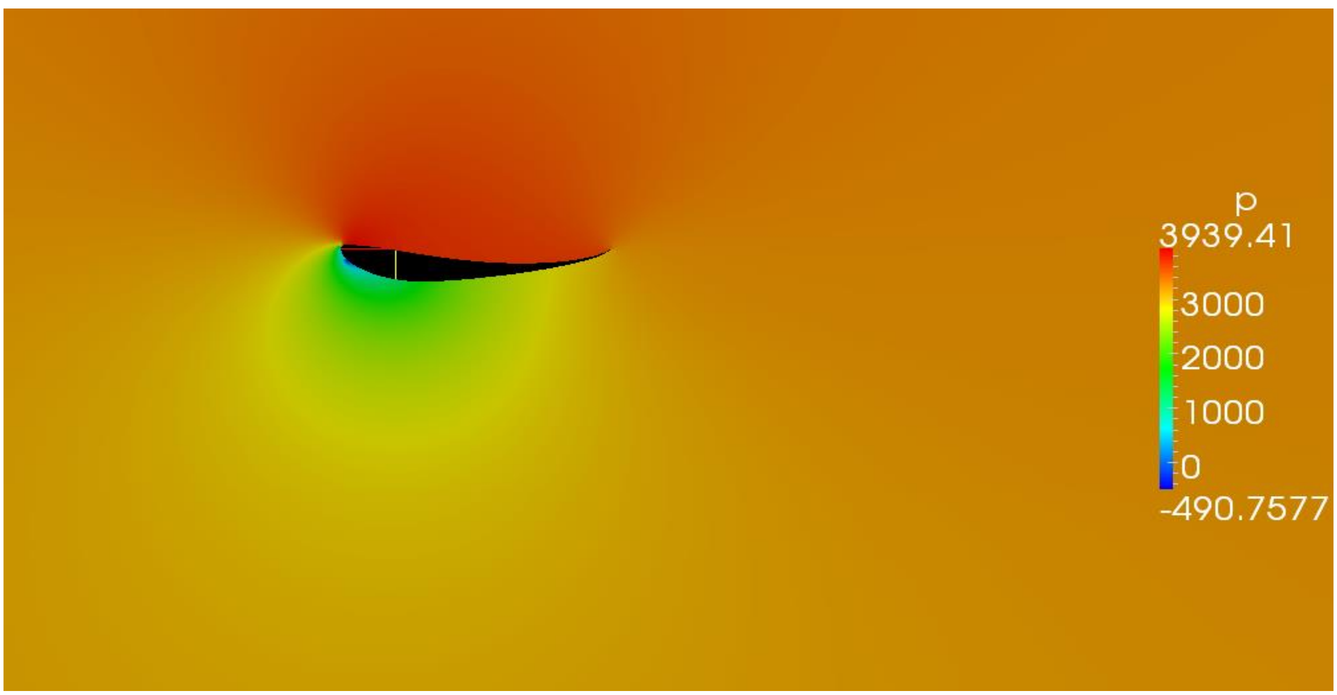


Fig. 10 Pressure plot around S-1223 at -16 aoa

Sr. No	Table II S-1223 No. of cells = 1.1 million Overall domain box volume = 4.2x4.2x1.2 m^3 upstream air velocity = 25 m/s = 90 km /hr Density of air = 1.204 kg/m ^{3} Area of wing = 0.25 x 1.2 = 0.3 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
1	-16	2.49898	2.3370	0.189559	21.40	282.10

Sr. No	Table III NACA- 6412 No. of cells = 1.1 million Overall domain box volume = 4.2x4.2x1.2 m^3 upstream air velocity = 36.11 m/s Density of air = 1.204 kg/m ^{3} Area of wing = 0.25 x 1.2 = 0.3 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
1	-15	1.9723	1.7063	0.414639	97.64	464.45

Sr. No	Table IV S-1223 RTL No. of cells = 1.1 million Overall domain box volume = 4.2x4.2x1.2 m^3 upstream air velocity = 36.11 m/s Density of air = 1.204 kg/m ^{3} Area of wing = 0.25 x 1.2 = 0.3 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
1	-13	2.06883	2.3228	0.374725	88.244	487.90

CONCLUSION:

As clear from tables, when angle of attack increases, Cl and Cd increases. But after certain angle of attack Cl is maximum and drops rapidly. It is called the stall angle of airfoil. Before stall angle, lift is maximum. At stall angle there is no lift and boundary layer separates from the airfoil profile. Also S-1223 has the highest lift coefficient and provide minimum drag. This airfoil can be used in formula racecars to provide down force at high speeds as it offers high levels of down force and minimum drag.

S1223 provides the maximum down force therefore we selected it for our wing.

TURBULENCE MODELING:

Turbulence modeling is one of the 3 key elements in computational fluid dynamics. Very precise mathematical theories have evolved for other key elements, via, grid generation and algorithm development. It creates a mathematical model that approximates the physical behavior of turbulent flows. Precision in turbulence modeling is not very good yet as approximations are done to capture nearly physical complex flows.

THE K-OMEGA MODEL:

This model was proposed by Kolmogorov. In this model omega was referred to as “rate of dissipation of energy in unit volume and time” and k was referred as “Turbulent Kinetic Energy”.

STANDARD WILCOX MODEL [4]

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial(\rho u_j k)}{\partial x_j} = P - \beta^* \rho \omega k + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_k \frac{\rho k}{\omega}\right)\frac{\partial k}{\partial x_j}\right]$$

$$\frac{\partial(\rho \omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = \frac{\gamma \omega}{k} P - \beta \rho \omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma_\omega \frac{\rho k}{\omega}\right)\frac{\partial \omega}{\partial x_j}\right] + \frac{\rho \sigma_d}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j}$$

SECTION-B

COMPARISON BETWEEN PARAMETRIC MODELLING AND SNAPPYHEXMESH

OpenFOAM comes with two mesh generators: blockmesh and snappyHexMesh. The first one can generate block structured grids from scratch, whereas the second mesh generator requires a hexahedral mesh to work properly.

PARAMETRIC MODELLING

Parametric modelling is comes under blockmesh. Parametric modelling is the ability to create geometries in the preprocessing phase based on a few fundamental variables called parameters.In parametric modelling we can generate any mesh geometry in OpenFoam from some independent parameters. The whole geometry depends on these parameters; if we want to change the dimensions of the geometry we will not change the code but only these parameters. It's a very lengthy and difficult process to generate a good parametric code but once it is generated then a variety of geometries can be made in a very short time.
Our task was to prepare a good parametric code to create an airfoil in OpenFoam.For this purpose we have done the following steps:

Coordinates of airfoils

Vertical and horizontal coordinates of about 1600 airfoils are available from [1]. We selected coordinates of symmetric airfoil NACA 0062 and NACA 2412 from [1].

NACA 0062

Table I

Y coordinate	X coordinates	Y coordinates	X coordinates	Y coordinates	X coordinates	Y coordinates
0.000630	0.300000	0.030010	0.012500	0.009470	0.200000	-0.028690
0.004030	0.250000	0.029710	0.000000	0.000000	0.250000	-0.029710
0.007240	0.200000	0.028690	0.012500	-0.009470	0.300000	-0.030010
0.013120	0.150000	0.026730	0.025000	-0.013070	0.400000	-0.029020
0.018320	0.100000	0.023410	0.050000	-0.017770	0.500000	-0.026470
0.022820	0.075000	0.021000	0.075000	-0.021000	0.600000	-0.022820
0.026470	0.050000	0.017770	0.100000	-0.023410	0.700000	-0.018320
0.029020	0.025000	0.013070	0.150000	-0.026730	0.800000	-0.013120
	1.000000	-0.000630	0.950000	-0.004030	0.900000	-0.007240

PLOTTING OF GRAPHS IN AND OBTAINED EQUATIONS

We used above coordinates to plot graphs in Microsoft Excel. NACA 2412 is a symmetric airfoil and that is why we plotted two graphs and derived two equations from that graphs. First graph represents the portion of airfoil above the chord line and second graph represents the lower portion.

UPPER CAMBER

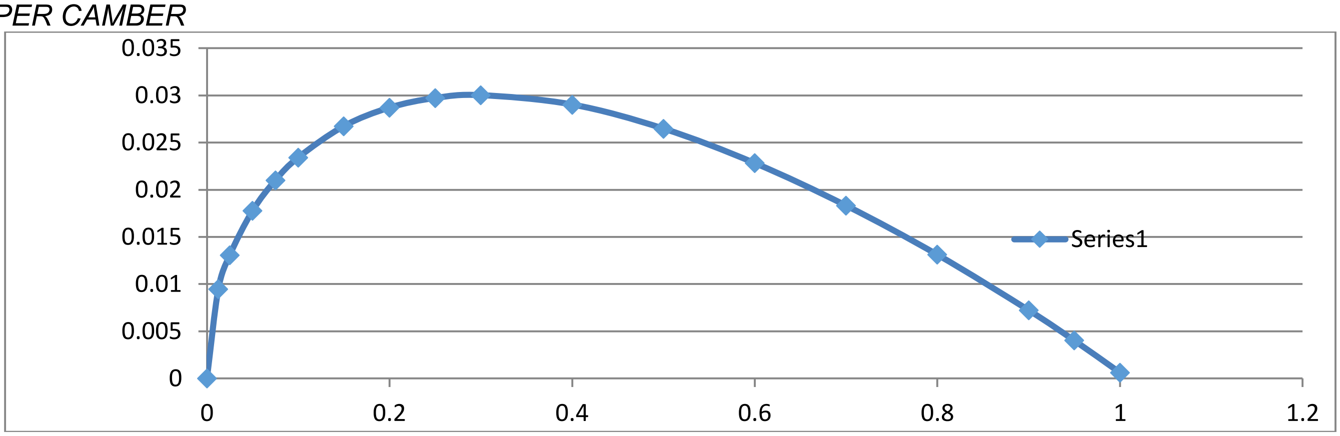


Fig.1

EQUATION:

$$y = -2.2056x^6 + 7.3255x^5 - 9.4918x^4 + 6.0975x^3 - 2.0986x^2 + 0.3702x + 0.0031$$

LOWER CAMBER

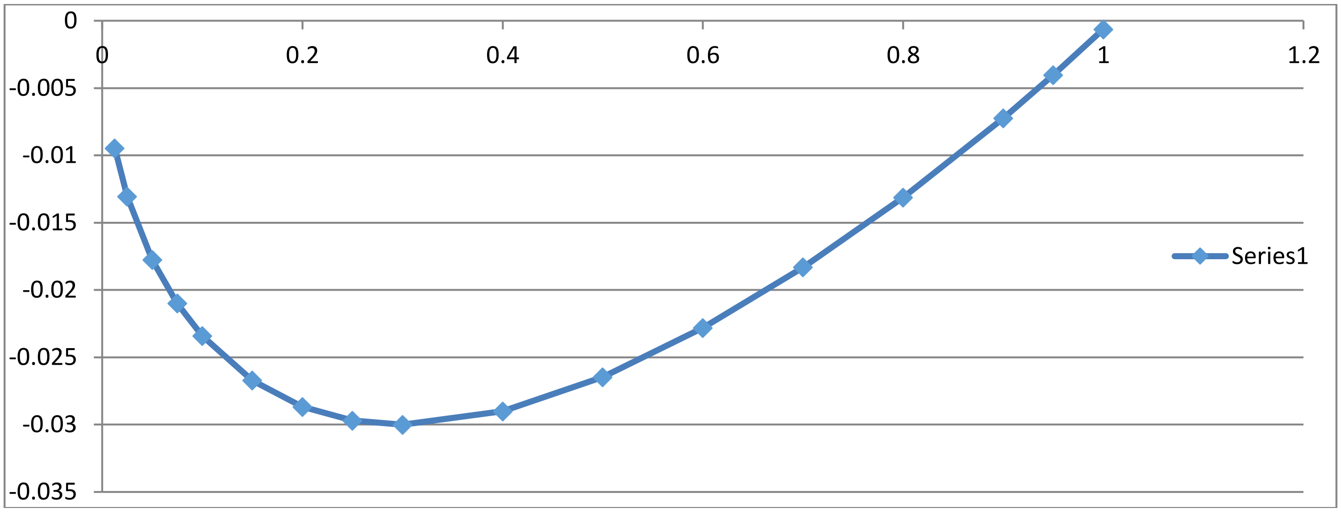


Fig.2

EQUATION:

$$y = 0.9188x^6 - 3.1595x^5 + 4.3167x^4 - 3.021x^3 + 1.2128x^2 - 0.2613x - 0.0069$$

The above two equations have used to calculate values of y in terms of x in OpenFOAM.

ACTIVATION OF PARAMETRIC MODELLING

OpenFoam 2.1.1 and 2.2.1 need to be activate before any simulation of parametric modelling. The following are the steps to activate parametric in OpenFoam:

1 - In Terminal type: **gedit ~/.OpenFOAM/2.1.1/controlDict**
if an empty file opens, try: **sudo gedit /opt/openfoam211/etc/controlDict**

2 - In the portion:

```
InfoSwitches
  {
     writePrecision  6;
     writeJobInfo    0;
     writeDictionaries 0;
 // Allow case-supplied C++ code (#codeStream, codedFixedValue)
     allowSystemOperations   0;
  }
```

Replace "**allowSystemOperations   0;**" with "**allowSystemOperations   1;**". Save and exit

BLOCKMESHDICT

We edit the blockMeshDict file from the cavity example of OpenFoam tutorials.

First we initialize all the independent variables. Which includes x coordinates, z coordinates and constants of both the equations. The next step was to write the equation in OpenFoam format to calculate values of y. There were total of 38 x coordinates and 2z coordinates.

EQUATIONS

```
y0 #calc "-$b0*pow($x0,6)+$b1*pow($x0,5)-$b2*pow($x0,4)+$b3*pow($x0,3)-
$b4*pow($x0,2)+$b5*pow($x0,1)+$b6";

y18 #calc "$b0*pow($x17,6)-$b1*pow($x17,5)+
$b2*pow($x17,4)-$b3*pow($x17,3)+$b4*pow($x17,2)-$b5*pow($x17,1)-$b6";
```

- y0 is value of y at x0
- b1, b2,b3, b4 ,b5,b6 are constants of the 6 degree polynomial
  We repeated the above equations 38 times each by varying x from x0 to x37 to calculate 76 values of y from y0 to y75.

VERTICES, BLOCKS AND EDGES

We defined 72 vertices required to generate airfoil as well as mesh domain. The vertices format in parametric modelling is like ($x36 $y39 $z1).
We defined 24 blocks for our mesh geometry. There are 8 inner and 16 outer blocks. The inner blockes are smaller and dense as compared to the outer blocks. In the edges we define only 2 arcs and four splines. We defined all the boundaries.

MESH OPTIMIZATION

Parametric meshing was not an easy task. The blockmesh code was very lengthy and difficult to debug, but we managed to remove all the errors and run the code.

FIRST MESH

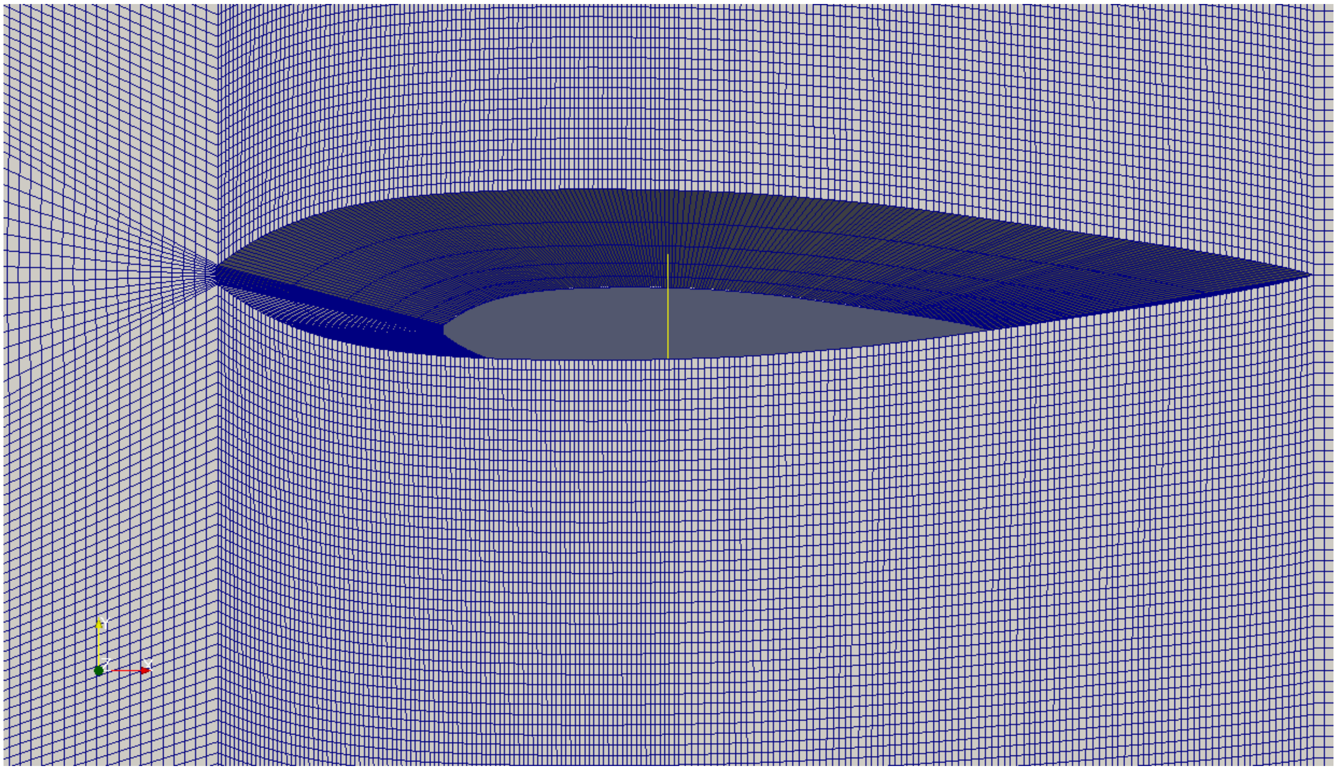


Fig.3

CHARACTERISTICS OF THE ABOVE MESH

Overall domain bounding box (-11 -11 0) (12 11 1.3)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (-8.52309e-16 8.55937e-16 9.08991e-16)
Max cell openness = 1.76227e-15
Max aspect ratio = 171.057
Minimum face area = 1.52631e-05. Maximum face area = 0.203493.
Face area magnitudes
Min volume = 3.9684e-06. Max volume = 0.0202569.
Total volume = 657.659. Cell volumes
Mesh non-orthogonality Max: 48.2966 average: 13.1831
Non-orthogonality check
Face pyramids
Max skewness = 2.47499 OK.
Coupled point location match (average 0)

Overall number of cells of each type:
hexahedra: 1115000

SNAPPYHEXMESH

SnappyHexMesh is a mesh generator that takes an already existing mesh created with blockMesh and chisels it into the mesh you want. But it requires:

- A very well defined dictionary, namely system SnappyHexMeshdict
- Good geometric definitions like STL object file with well-defined surfaces.
- SnappyHexMesh is more of a mesh sculptor than a mesh generator, because it requires an already existing base mesh to work with.

- Depending on the options given through the file “system/snappyHexMeshDict , it can:
- Refine the mesh
- Adjust the mesh to fit into provide geometries
- Add the boundary layer near requested patch

CAD FILE

We generate an airfoil NACA 2412 in SolidWorks by taking its coordinates from airfoil tools.
Following are the steps we used to generate airfoil.

- Start up the SolidWorks
- Select the units from the bottom(inches, meters or any other)
- Go to new and select part from the open window
- Go to insert and select curve and then select curve through xyz points
- Browse the xyz coordinate notepad text file and select ok
- A 2D airfoil will be formed
- In the 2D airfoil select sketch and then front plane
- Select the airfoil then convert entities
- Go to Future and Select Extrude Boss/Base
- Give the Extrude Length in any unit and then ok

We give the extrude length (span) of 1.3 meter

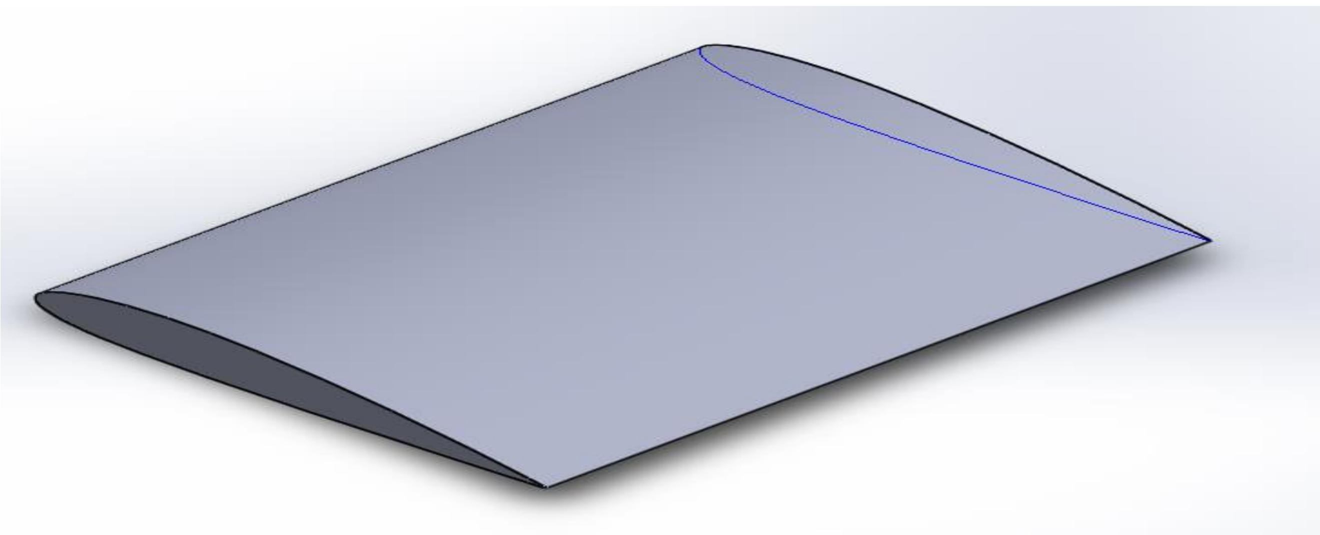


Fig. 4

STL (Stereo lithography) File

STL is a file format native to the stereo lithography CAD software created by 3D systems. This file format system is supported by many other software packages. STL files describe only the surface geometry of a three dimensional object without any representation of color, texture, or other common Cad model attributes. The STL format specifies both ASCII and binary representations. STL coordinates must be positive numbers, there is no scale information, and the units are arbitrary.
STL files are also supported by OpenFoam. That is why we saved the cad files of SolidWorks in STL format.
We did the following steps to save the file in STL format:

- Save As the part file of SolidWorks
- Give any name to the file and from save as type select STL (*.stl)
- From options Select ASII and select appropriate units and then ok

The file is ready to import in OpenFoam.

SnappyHexMesh in OpenFoam

We have generated well defined geometries in Solidwork and Now will import all thse files in OpenFoam and will generate meshes.

BlockMesh

We have used blockMesh to create simple block based fully structured hexahedral meshes.
The tutorial Motorbike is an example of snappyHexMesh. So we edited the blockMesh of motorbike. To access the BlockMeshDict open Motorbike, double click constat, then double click polyMesh and the blockMeshDict.
BlockMeshDict file consist of five main sections:

- vertices → Prescribe vertex locations for blocks
- blocks → Prescribe block topology and mesh settings
- edges → Prescribe curved edges
- patches → Prescribe surface patches for boundary conditions
- merge PatchPairs → Merge disconnected meshed blocks

We defined 8 vertices to defined a block, which is also our mesh doamin.
There were no any curves so we did not use eges and finaly we defined patches for boundary conditions. We created a mesh domain with the follwing dimensions:

Length = 19 meters
Thickness= 1.3 meters
Height = 20 meters

SnappyHexMeshDict

Utility snappyHexMesh is used to create high quality hex-dominant meshes based on arbitrary geometry.
This utility has the following key features:

- Fully parallel execution
- STL and Nastran files support for geometry data
- Quality guaranteed final mesh that will run in OpenFoam

The snappyHexMeshDict is a dictionary in Motorbike/system; we have edited this file as our requirement.

Requirements of SnappyHexMesh

- Dictionary file system/snappyHexMeshDict
- Geometry data (stl, nas, obj) in *constant/triSurface*
- Hexahedral base mesh (decomposed if running in parallel)
- Dictionary file *system/decomposeParDict* for parallel runs
- All system dictionaries (e.g controlDict, fvSchemes, fvSolutions)

SnappyHexMesh Domain

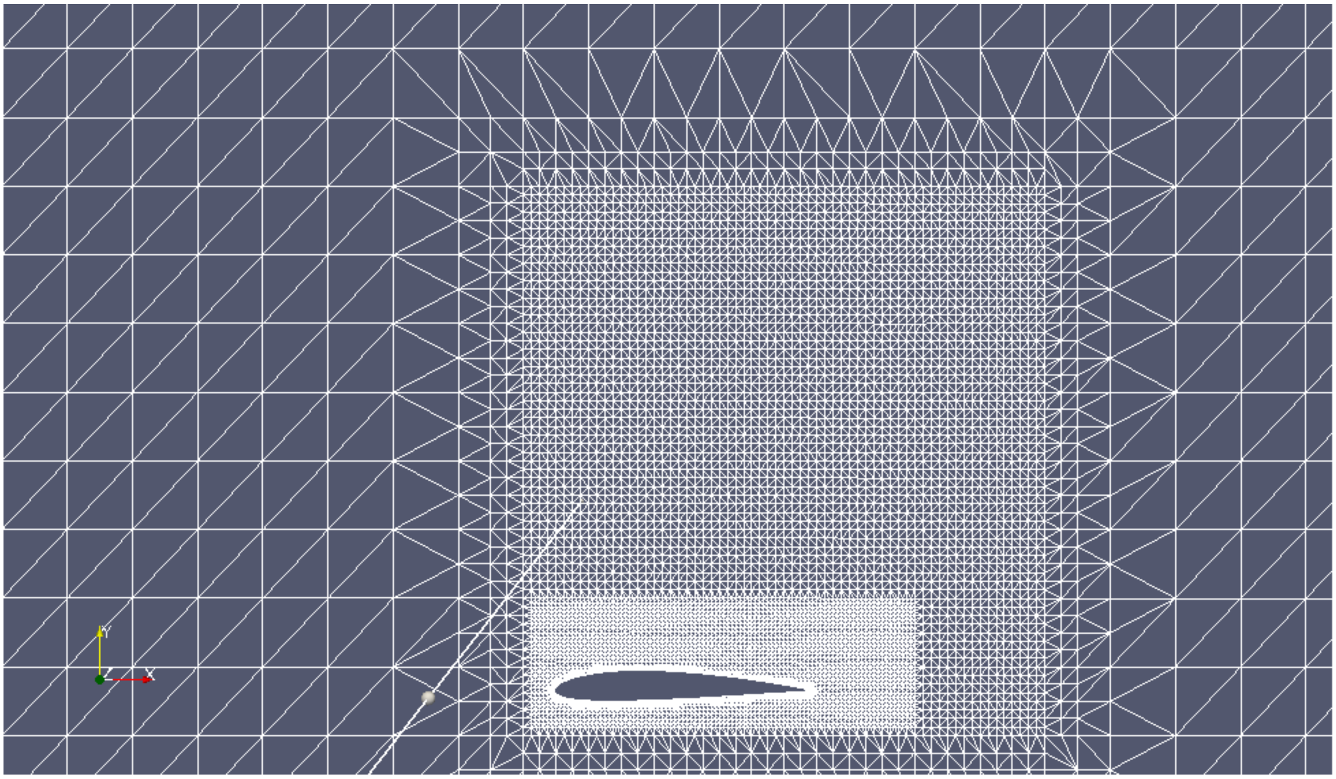


Fig. 5

Overall domain bounding box (-11 -11 -0.5) (12 11 2)
Mesh (non-empty, non-wedge) directions (1 1 1)
Mesh (non-empty) directions (1 1 1)
Boundary openness (1.07962e-15 1.8974e-16 2.22938e-14) OK.
Max cell openness = 3.34367e-16 OK.
Max aspect ratio = 21.881 OK.
Minimum face area = 1.51225e-07. Maximum face area = 0.0225827.

Face area magnitudes OK.
Min volume = 1.01839e-10. Max volume = 0.00313451.
Total volume = 1264.89. Cell volumes OK.
Mesh non-orthogonality Max: 54.861 average: 7.86709
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 1.50754 OK.
Coupled point location match (average 0) OK.

Overall number of cells of each type:
hexahedra: 1158697
prisms: 21260
wedges: 458
pyramids: 0
tet wedges: 608
tetrahedra: 0
polyhedra: 90216

SOLVER:

Simple Foam solver was chosen for analysis as it calculates pressure, velocity, Cl,Cd coefficients.

After mesh creation and given boundary conditions, simpleFoam solver was run to simulate the case. In each case either Velocity was changed or angle of attack or an airfoil.

PRESSURE PLOTS:

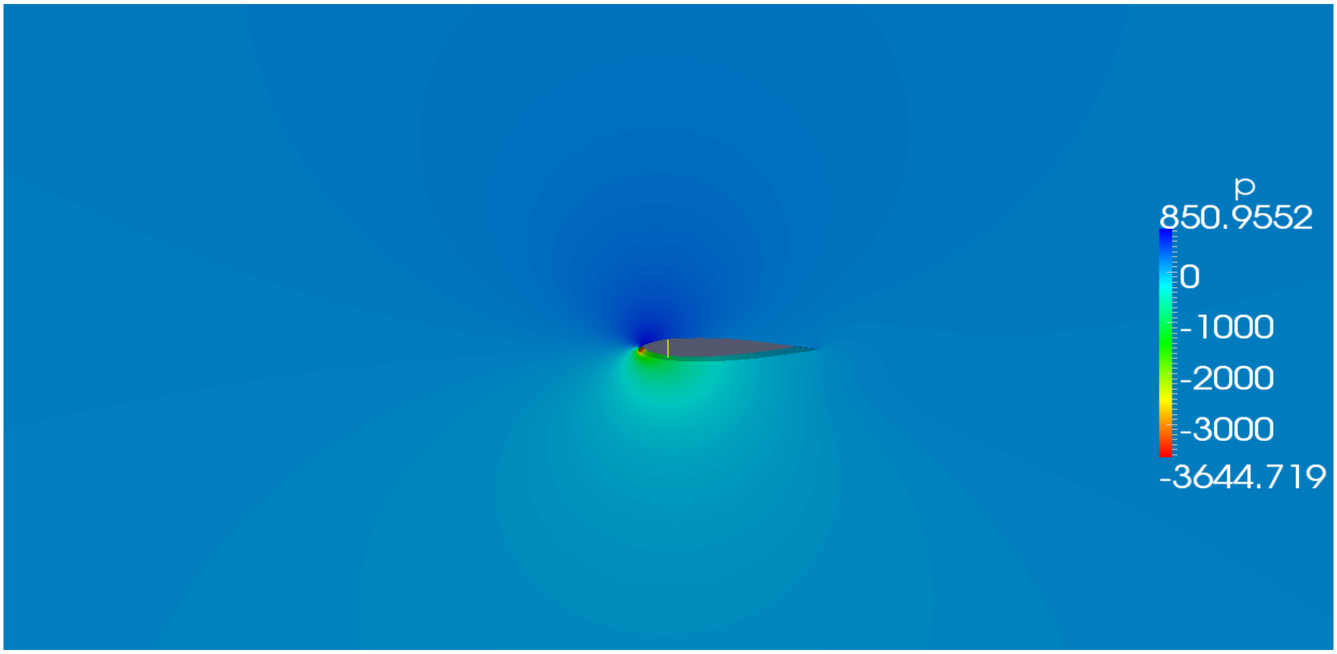


Fig.6 parametric pressure plot

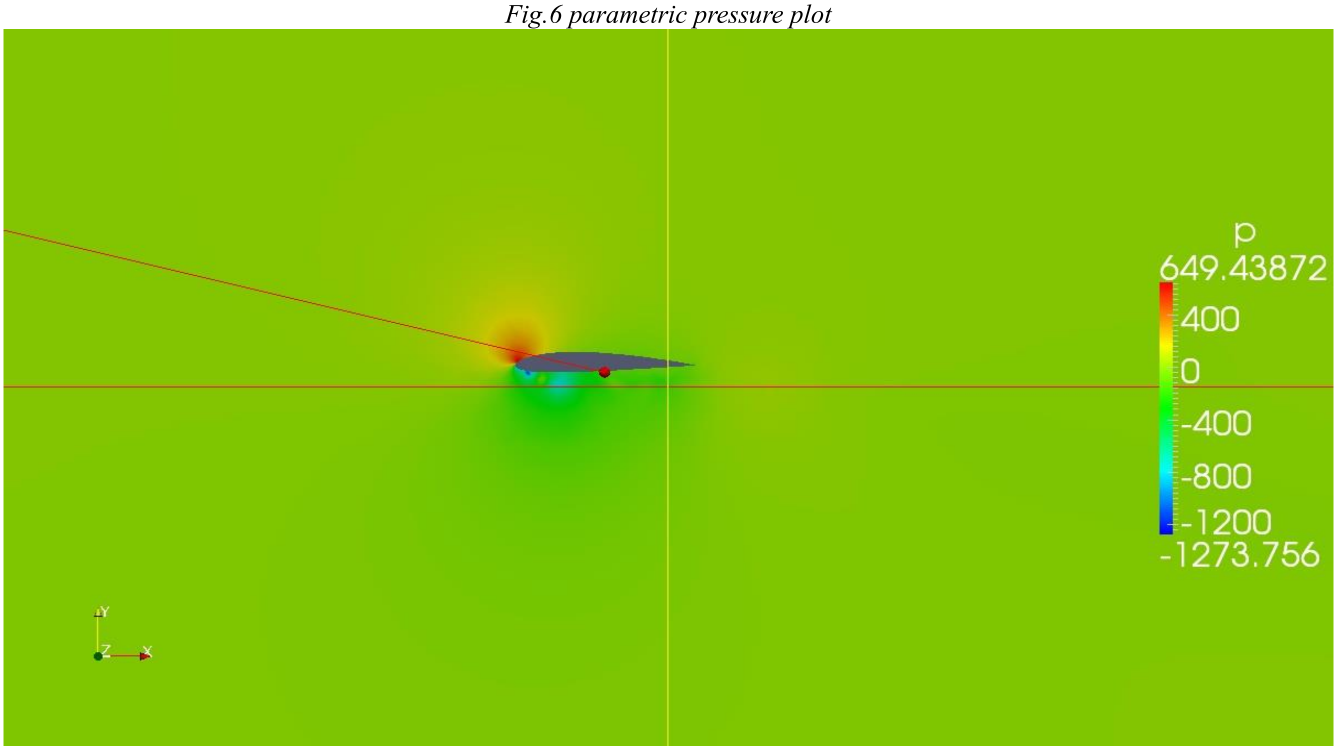


Fig.7 parametric pressure plot

COMPARISON OF COEFFICIENT OF LIFT AND DRAG

Mesh Type	Table Laminar flow NACA 2412 upstream air velocity = 36.11 m/s = 130 km/hr Density of air = 1.204 kg/m ^{3} Area of wing = 1 x 1.3 = 1.3 m^2					
	Angle of attack	Number of cells	OpenFOAM Cl	OpenFOAM Cd	Drag Force N	Down force N
Parametric	-16.5	1115000	1.19873	0.187126	190.95	1223.25
SnappyHexMesh	- 16.5	1271239	0.538467	0.152069	155.1799	549.48
Parametric	-16.5	5040000	1.25437	0.180995	184.69	1280.03
SnappyHexMesh	-16.5	5545724	1.47305	0.0896157	91.448	1503.184

Experimental Results

Angle of attack	Cl	Cd
-16.5	**1.67**	**0.25**

CONCLUSIONS:

- The results of parametric are very closer to the experimental results as compared to the snappyhexmesh.
- The computation time to solve same number of cells in snappyhexmesh is 5 times greater than parametric meshing.
- Results of snappyhexmesh become better when number of cells increased.
- It’s a challenging task to create parametric mesh but once it is created then it works for any type of airfoil.
- Snappyhexmesh requires software to create cad file and convert it to stl file.
- Snappyhexmesh is better option to mesh for very complex geometries.
- To optimize mesh characteristics like skewness and mesh orthogonality of parametric modelling require much time.

SECTION-C

MULTIWING CONFIGURATION

PROBLEM DEFINITION:

In the analysis of airfoils, analysis on single air foil was done but there was still a better option to get more effective results and that was using multiple air foils orientated at different angle of attacks. Also different airfoils have max cl/cd ratio at different angle of attacks so in spite of using a single air foil with one angle of attack, multiple airfoil with multiple angle of attack will eventually provide more down force.

Action plan

As mentioned earlier 3 best airfoils were chosen after the survey at different angle of attacks. All the needs to be plotted in the OPEN Foam, Two methods were there to achieve the goal:

1) Write the program in the block mesh dictionary
2) Plot it in CAD software and import to OPEN Foam

As programing in the block mesh dictionary was already done, so Solid Works software was used to plot the airfoils by importing its coordinates. These solid works files were than imported in OpenFOAM. Snappy hex was used to discretize and mesh the airfoils in the OPEN Foam. Boundary conditions were according to pressure and velocity conditions that will be prevailing on Silverstone track in the month of July. To get the solution of real time problem. Simple Foam solver was used to solve it.

PLOTTING AIRFOILS IN SOLID WORKS:

- Open Solid Works 2013
- Press ctrl+N
- Select part
- On toolbar click on curves
- Select curves through xyz points
- In the pop up window, click browse and enter the destination of notepad file containing the coordinates of airfoil (note that coordinates should be in xyz format)
- Press ok and the airfoil will be plotted in solid works.
- Repeat the same for the other two airfoil coordinates
- Select them all and in the sketch tab click on convert entities.(it will make it one single entity)
- Extrude it according to the desired dimensions.
- Press ctrl+S, in the pop up window select the save as type "STL (*.stl)", click options, check output as ASCII and unit meters.
- Press OK and save to save it to the desired location.

IMPORTING IN OPEN FOAM:

Copy the file bike from tutorial folder in simple foam solver
Copy the STL file in the trisurface folder
Run the example using terminal, press ctrl+O select that STL file and it will be imported to the bike example.
Adjust the dimensions of the box from the block mesh.
Now mesh it using snappy hex mesh command and run in simple foam solver to get the results

MULTIWING ANALYSIS (with and without gurney flap):

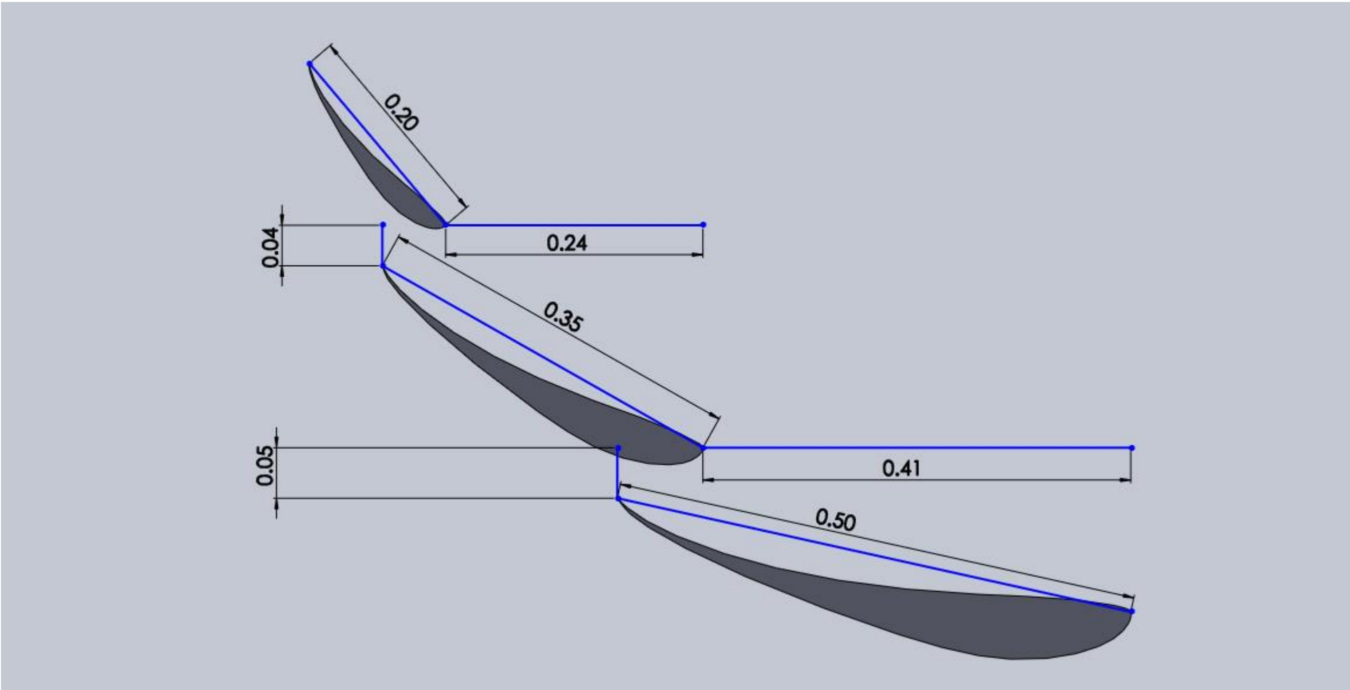


Fig. 1 leading edge to leading edge distance and trailing edge to leading edge distance with chord lengths

	Table I **Laminar flow** **3 S-1223** **No. of cells = 395232** **2^{nd} wing chord length=70%c of 1^{st} wing** **3^{rd} wing chord length=40%c of 1^{st} wing** **upstream air velocity = 36.11 m/s** **Density of air = 1.204 kg/m ^{3}** **Area of wing = 0.8712 m^2**					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
Without gurney flap	**0**	**3.4418**	**N/A**	**1.38688**	**948.436**	**2353.7**
With gurney flap Height = 10 mm (5 % of chord length) Thickness = 3 mm	**0**	**3.5308**	**N/A**	**1.50197**	**1027.142**	**2414.6**
% increase in drag and down force with gurney flap					**8.2 %**	**3 %**

Sr. No	Table II 3 S-1223 (without gurney flap) No. of cells = 395232 2nd wing chord length=70%c of 1st wing 3rd wing chord length=40%c of 1st wing upstream air velocity = 36.11 m/s Density of air = 1.204 kg/m ^{3} Area of wing = 0.8712 m^2					
	Angle of attack	**OpenFOAM Cl**	**Wind tunnel Cl**	**OpenFOAM Cd**	**Drag Force N**	**Down force N**
Turbulent flow	**0**	**2.831**	**N/A**	**1.110**	**759.08**	**1936.01**
Laminar flow	**0**	**3.4418**	**N/A**	**1.38688**	**948.436**	**2353.7**
% decrease in drag and down force w.r.t laminar flow					**20 %**	**17.7 %**

SECTION-D

FLUID STRUCTURE INTERACTION ANALYSIS

The solver used for Fluid Structure Interaction (FSI) is icoStructFoam. It is a combination of transient incompressible laminar solver icoFoam and the stress analysis solver solidDisplacementFoam, with conjugate base on an idea from conjugateFoam. The solver thus combine two pre-existing solvers of openFoam. Technically two meshes are produced which are overlapping at certain patches, as user wishes. These are then put in different polyMesh directories, one for the region 1 and one for region 2 respectively.
In a fluid flow system there is always some kind of FSI, but the effect might not be noticeable on the domain we wish to study. For instance, flow around a building is usually modelled as a one–phase system since the building usually is rigid and does not influence the flow other than statically. On the micro scale, however, there might be some wear on the walls of the building which again might slightly alter the boundary layer behavior. This is usually deemed negligible.
When a body deforms, as the blades of a wind turbine or an airplane wing for instance, the domain over which the fluid flows alters and the flow is affected by this deformation. The new flow pattern will again influence the flexible body and we have non-negligible interaction between fluid and solid.

Equation:

IcoFoam solves the Navier-Stroke’s equation and couples the momentum field to the continuity equation and pressure by the PISO formulations. The Navier-Strokes equation and the continuity equation for the incompressible flow are given by:

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} - \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i}$$
$$\frac{\partial u_i}{\partial x_i} = 0$$

solidDisplacement Foam solves the following form of Navier equations for the solid region.

$$\frac{\partial^2 D_i}{\partial t^2} = \frac{\partial}{\partial x_j}\left(2\mu + \lambda\right)\frac{\partial D_i}{\partial x_j} + \frac{\partial \sigma_{ij}}{\partial x_j}$$

where $\mu = \frac{E}{2(1+\nu)}$ and $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$.

After the solidDisplacement Foam solver has been run the pressure from the fluid solver is written to the solid side of the boundary, before the next time step is initiated.

The solver may be concluded as doing the following in every time step:

1. The fluid mesh is deformed according to the displacement of the solid boundary.
2. The flow is solved for in the deformed region.
3. The deformation of the solid region is solved for, based on the pressure distribution at the solid / fluid interface from previous time step.
4. The pressure is transferred from the fluid to the solid region.

INTEGRATION WITH OPENFOAM 2.2.1

First of all this solver is integrated with openFoam version as it is not present in these version. For the integration following steps are taken.

- Delete the dependency file (.dep) Example: icoFoam.dep. Also delete the 'linuxGccDPOpt' folder.
- In the 'Make' subdirectory open the 'files' file and change the names of the main files. E.g. replace "icoFoam" with "my_ico-Foam". Also change "EXE = $(FOAM_APPBIN)" by "EXE = $(FOAM_USER_APPBIN)".
- Compile the solver by going to the solver directory in a terminal window and running the 'wmake' command.
- If the solver compiled correctly it should be in the list of apps obtained by running 'ls $FOAM_USER_APPBIN' in the terminal.

MESH GENERATION:

Now in this there are two regions. One of the fluid and one for the solid, so there are two separate blockMeshes which will then be coupled together. So that one solution can be formed by combining two. There is an outer mesh of fluid and an inner mesh of solid.
An outer Mesh is formed and placed in region 1, this is the fluid mesh. Then an inner mesh is formed and is placed in the region 2, this is the solid mesh. Now these two meshes are to be solved together, as the fluid mesh is solved then its value is used to solve the solid mesh for each time step. Now the coupling parameters should be defined in which we had to define the boundaries which should couple. Inlet, Outlet and Front and Back are defined.

Mesh Domain:

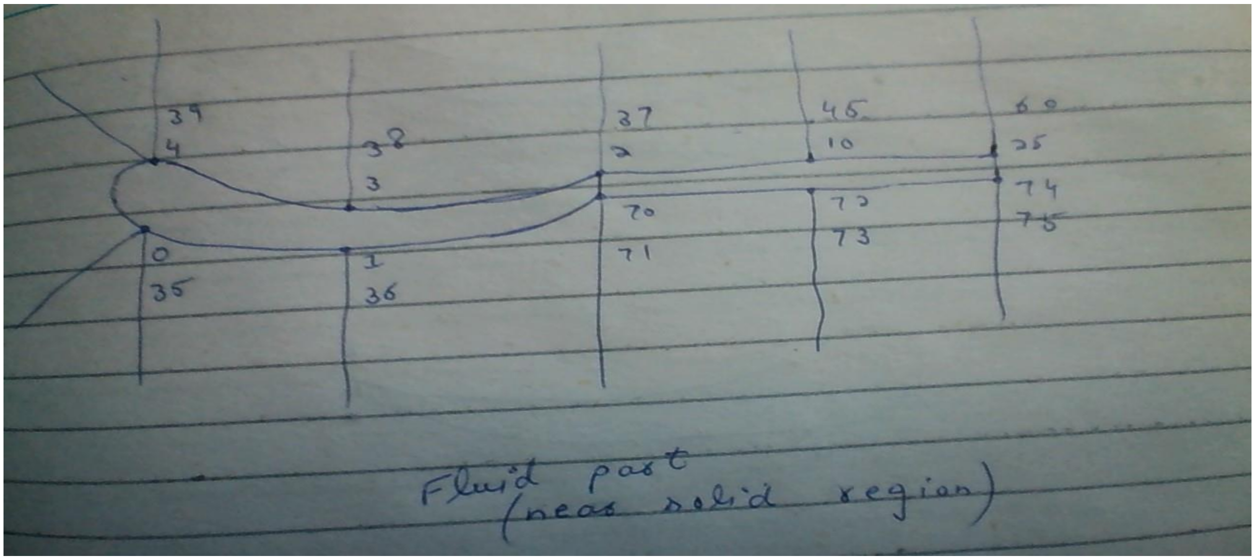


Fig. 1 Fluid Domain

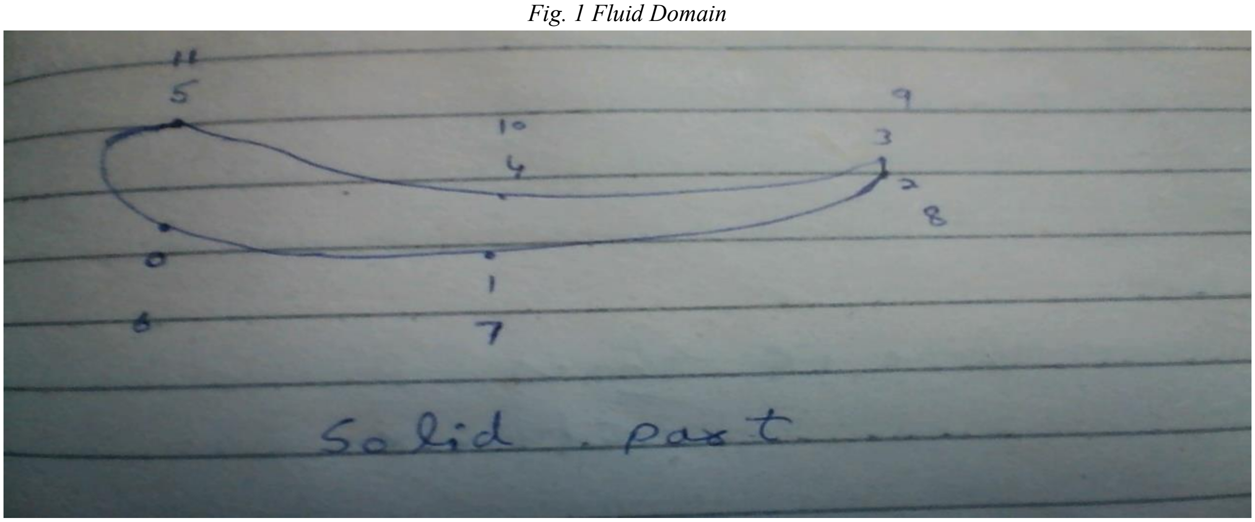


Fig. 2 Solid Domain

Mesh Optimization:

The difficult part of the process was to align the two meshes otherwise we receive an error that “size does not matches” for this on the coupling boundary the point should be same as in both the fluid and solid. When the points are same, the meshes will be aligned and same cell division is used. Then mesh errors are removed like orthognality, skewness, aspect ratio.

BlockMesh and Its Characteristics:

As this is problem with two region so its method of block Meshing is different and it depends on the following series of commands
/home/my_orig/IcoStructFoam_Rev561
> ln -s region1/polyMesh/ airFoil/constant/polyMesh
> blockMesh -case airFoil
> rm airFoil/constant/polyMesh
>
> ln -s region2/polyMesh/ airFoil/constant/polyMesh
> blockMesh -case airFoil
> rm airFoil/constant/polyMesh
>
> cd airFoil
> icoStructFoam (It will run successfully without any editing dictionaries)

In these commands airfoil is the case which we are running. It can named any all these commands should be typed in the terminal window outside the case folder. After the BlockMesh is done then we can enter in the case folder and run the solver or convert it to VTK files for visualization.
IcoStrcutFoam_Rev561 is just a folder in which my case lies. It can be renamed or the case can be placed anywhere in the home.

Fluid Mesh Characteristics:
```
----------------------
Mesh Information
----------------------
  Bounding Box: (-4.05685 -4.2 0) (4.2 4.2 1.2015)
  Number of points: 1567200
  Number of Cells: 1300000
  Number of Faces: 4166000
  Number of Internal Faces: 3634000
```
Solid Mesh Characteristics
```
----------------------
Mesh Information
----------------------
  Bounding Box: (-1.00429e-05 -0.0473598 0) (0.349388 0.00554355 1.20006)
  Number of points: 121806
  Number of Cells: 100000
  Number of Faces: 321500
  Number of Internal Faces: 278500
```

MESH VISUALIZATION:

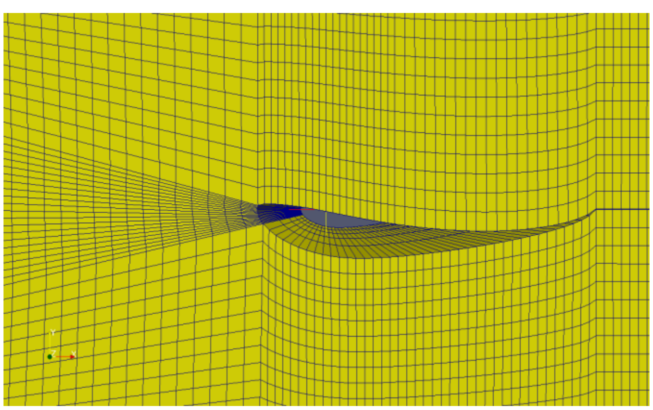
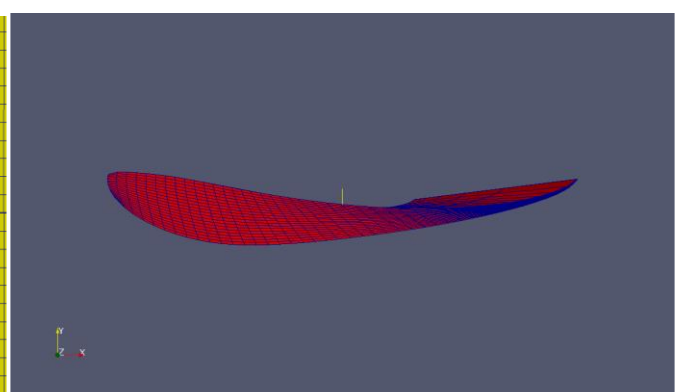



Fig. 3 Fluid Mesh *Fig. 4 Solid Mesh*

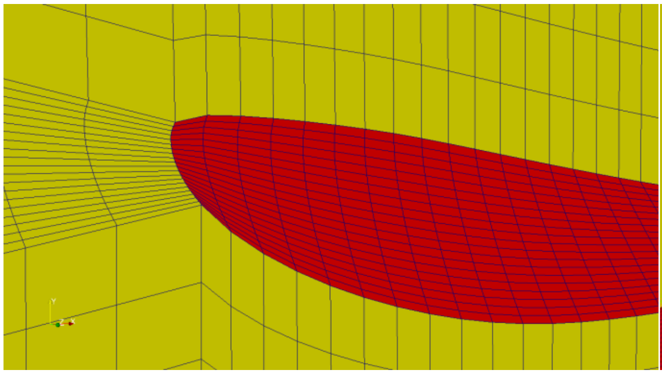


Fig.5 Leading edge

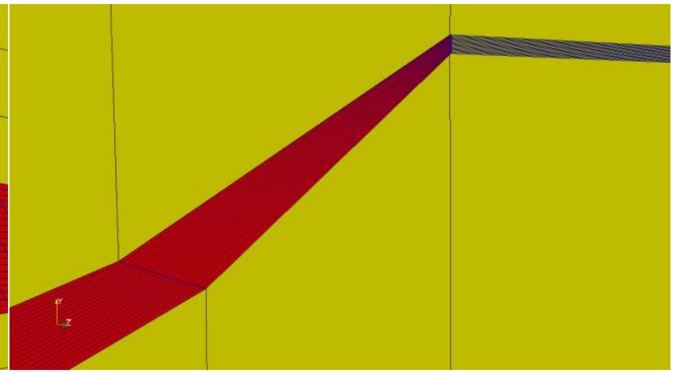


Fig. 6 Trailing edge

Physical Properties:

Density of Carbon Fiber = 1950 kg/m3
Poison ratio = 0.3
Bulk Modulus= 666.60 GPa

Density of Steel = 8095 kg/m3
Poison ration = 0.266
Bulk Modulus = 166 GPa

Density of Fiber Glass = 2560 kg/m3
Poison ration = 0.22
Bulk Modulus = 45 GPa

Initial and Boundary Conditions:

Basically there are four boundaries in our case

1. Inlet
2. Outlet
3. frontAndBack
4. wall

The coupling parameters are only defined at the wall boundary condition, as only the wall boundary is contact with each other of the two regions.

Post-Processing:

For post-processing we can't directly open the paraFoam. Instead of this we use the following commands inside the case directory to convert it to VTK file, which then can be opened separately for the both the region by opening them in paraview.

> foamToVTK -region region1

> foamToVTK -region region2

> paraview (paraFoam)

Please use Warp filter with vector D to see region2 deformation.

OpenFOAM Results:

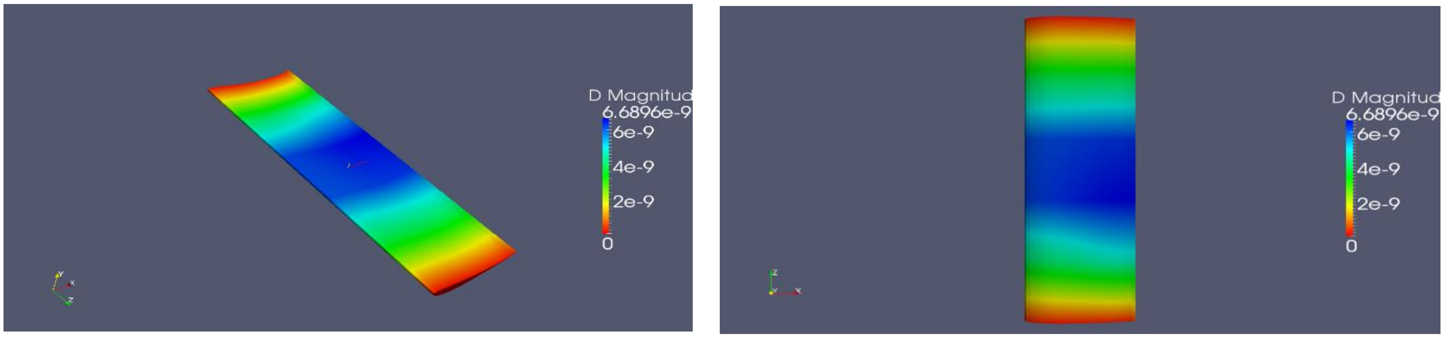


Fig. 7 Steel with maximum displacement of 6.6896e-9 m.

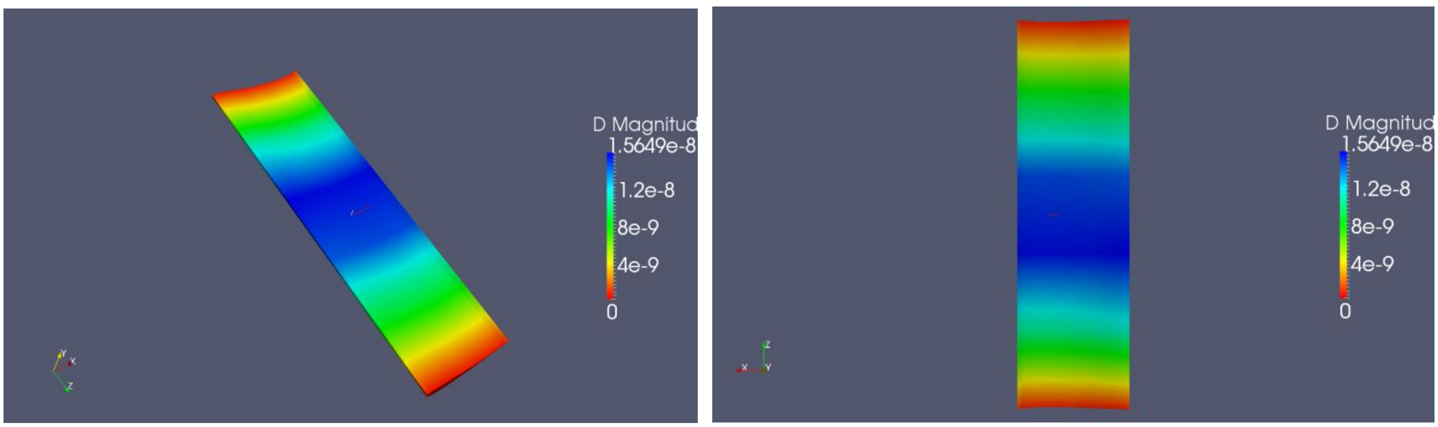


Fig. 8 Carbon fiber with maximum displacement of 1.5649 e-8

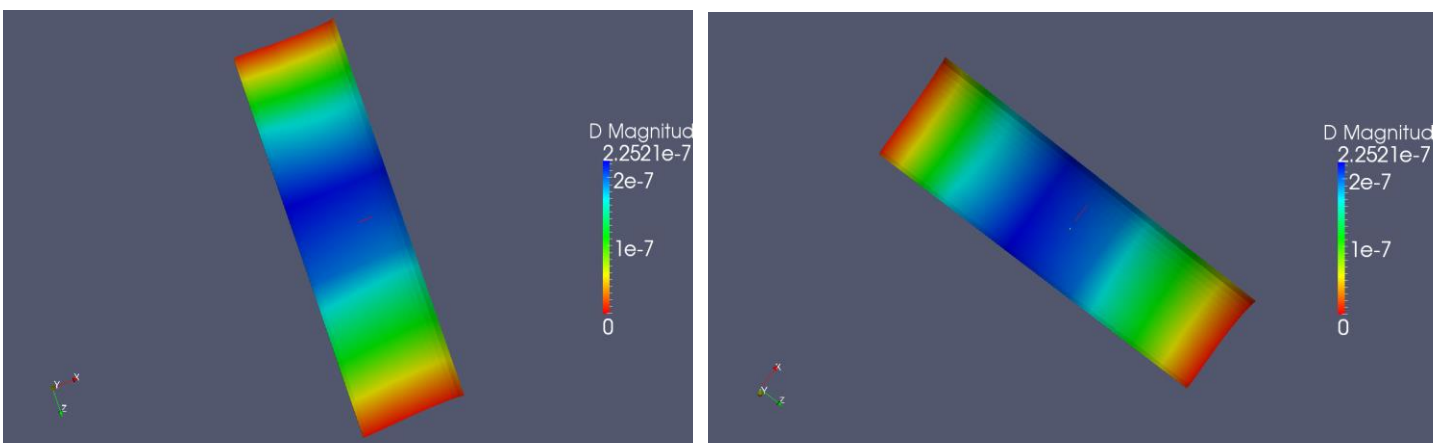


Fig. 9 Fiber glass with maximum displacement of 2.2521e-7

Static load analysis:

For Static loading testing ANSYS was used using mechanical APDL and Static Structural solver. The steps followed are as follow..

1. Import the IGS file of the part.
2. Define the Material.
3. Make the Mesh.
4. Define loads and Supports.

5. Solve

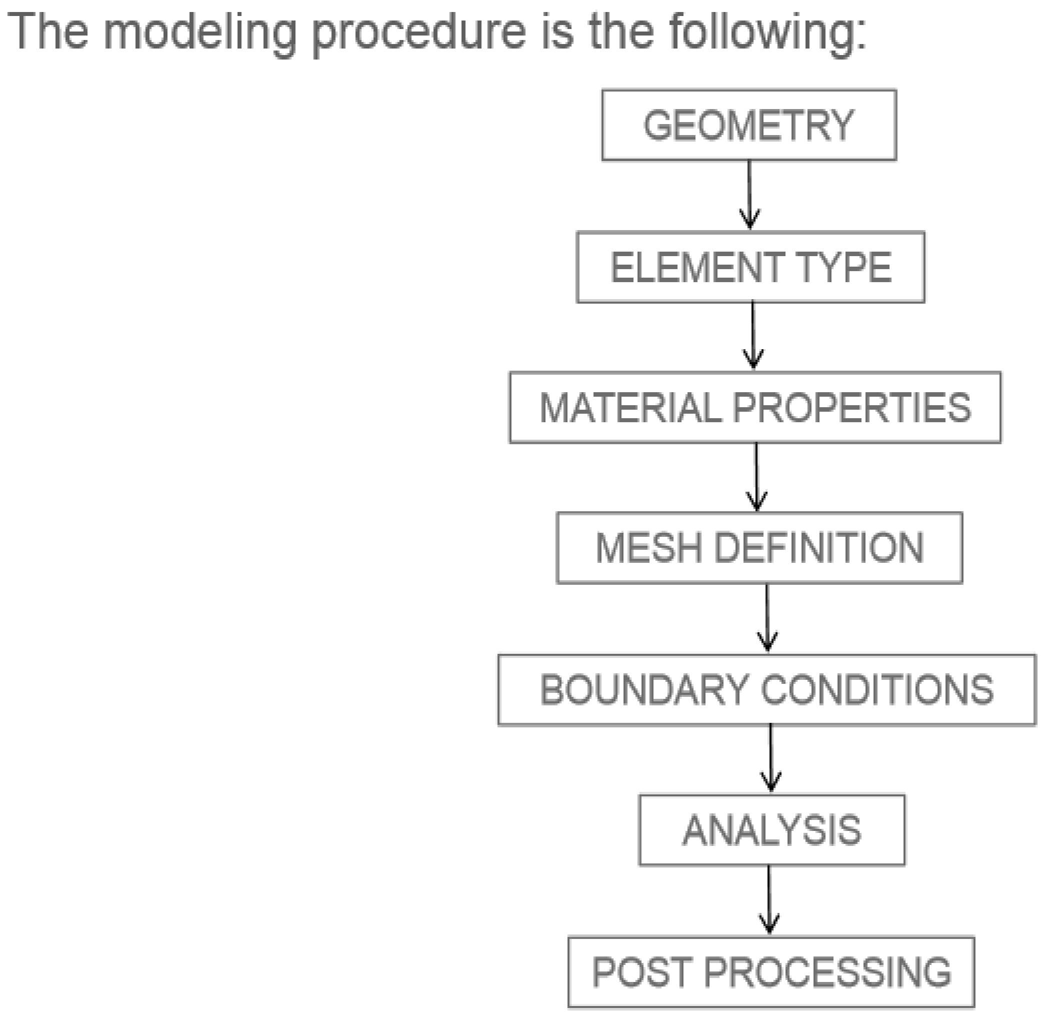
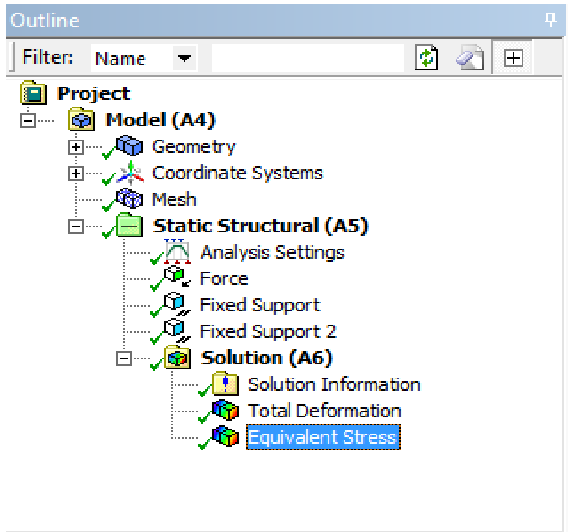


Fig. 11

The scheme which is used is as follows:



After the geometry and coordinate are selected. ANSYS uses automatic meshing technique to mesh all the object using

- Triangular Surface meshes
- Tetrahedral meshes

The mesh can be refined by using the refinement technique, then in the structural part all the loads and fixed support are defined. In our case the load is applied on the top surface of the airfoils and the two ends of the airfoil are the fixed support. Then once all the boundary conditions are defined, we can select type of solution and then ask the ANSYS to solve for the required stresses or deflections.

The force applied is in linear way.

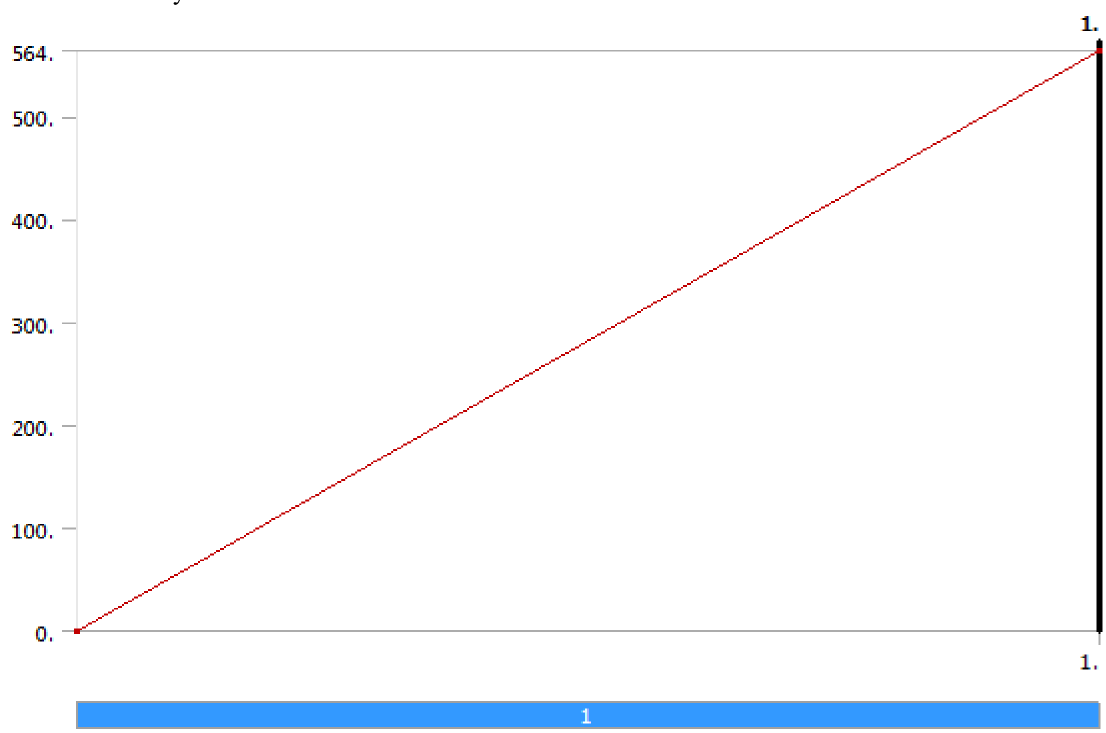
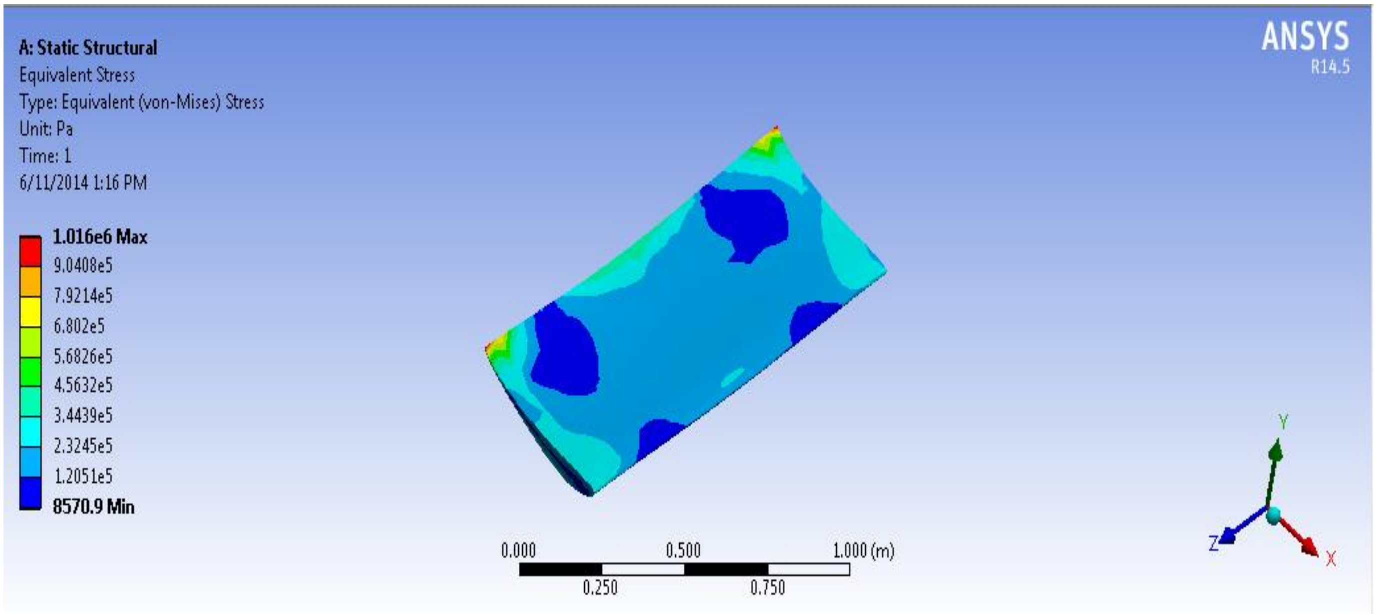


Fig. 12

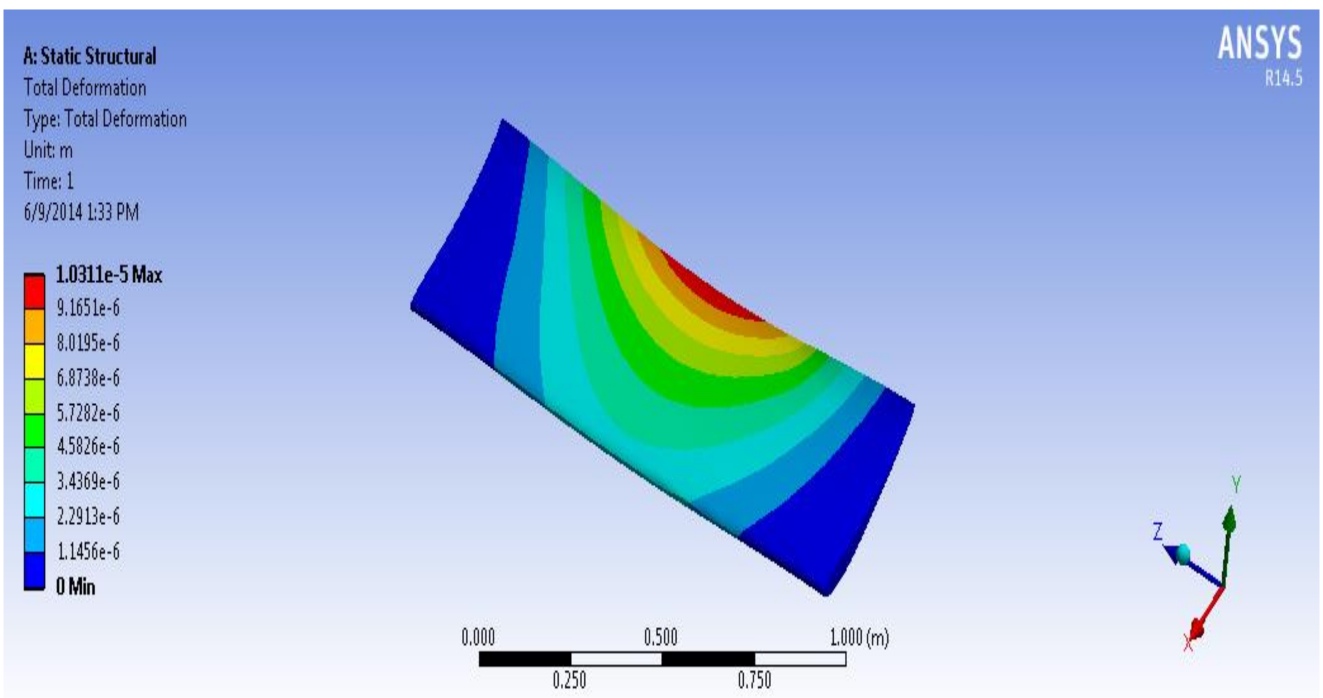
The factor of safety used is 1.2
The factor of safety was used to ensure the strength of the airfoil if the load exceed then the airfoil is still safe.

The result for the von misses stress is as

ANSYS Results



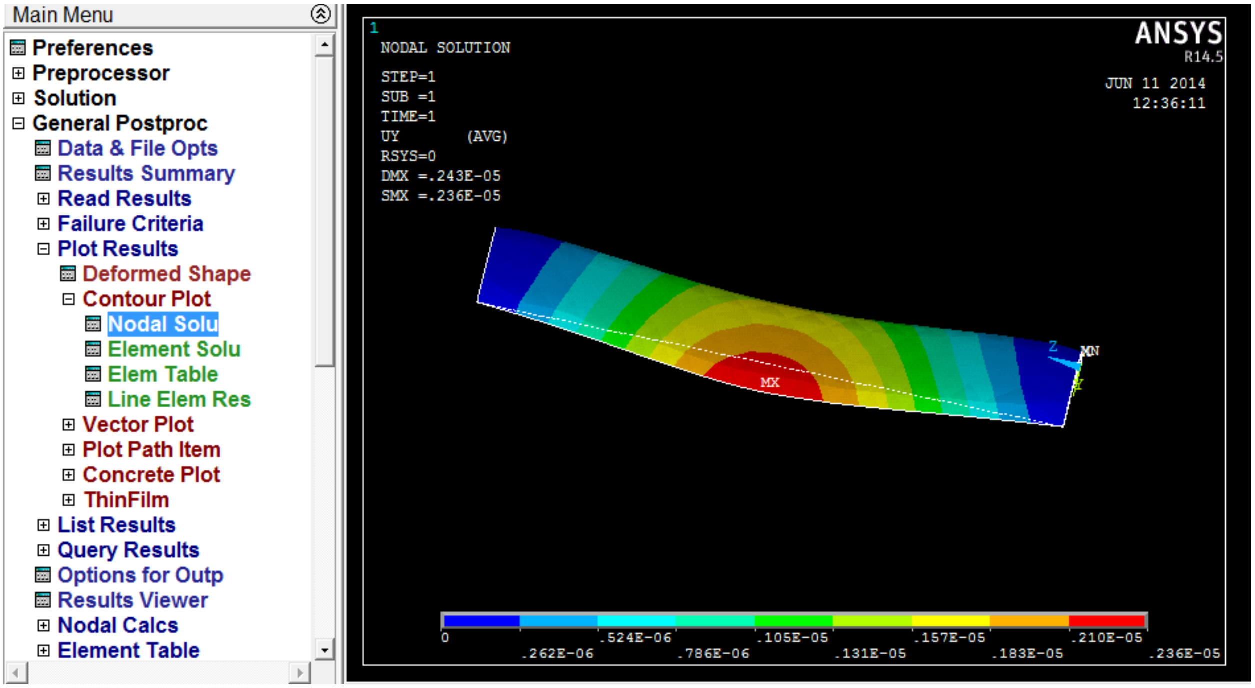
The result for the maximum displacement is



The trailing edge has maximum displacement, as it has the least thickness and the least area but the material is still safe.

Mechanical APDL is used to apply the pressure distribution in the airfoil and then find the maximum displacement. The result are almost the same. The procedure is the same as used in the workbench. It consists of preprocessing, solution, post processing.
First of all the part file is imported and then the material properties are selected and then mesh is generated using automatic meshing

technique and then the pressure distribution is applied. The pressure of openfoam is used and applied in the APDL, then the supports are defined at the ends of airfoils and the mesh is solved for the required solution.
The result from the ANSYS Mechanical APDL are as follow:



Here also the maximum displacement is 0.26e-05 and hence the material is safe and the maximum displacement is in the center of the airfoil due to least thickness and less area.

All these results conclude that the material Carbon Fiber and Fiber Glass are safe for he manufacturing of the airfoil and the deformation is very less in the airfoil which can be neglected and it has no effects on the airfoil as it is safe. This very much less deflection cannot effect the airfoil.
So it is good to go with Carbon fiber or fiber glass as they are very light material and are very hard, so they cannot be easily deformed. So with the two best options are

- Carbon Fiber
- Fiber Glass

Any grade of these material can be used because all are good/acceptable for this kind of manufacturing.

Detailed Survey of wing Manufacturing Techniques:

Manufacturing Techniques	Pros	Cons	Estimated Cost
Scale model by Rapid Prototype Machine and then use for wind tunnel testing.	• Results can be tested. • Light weight and small	• High cost for multiple airfoil wing	30,000-35,000 Rs
Scale model of nylon to be tested in wind tunnel	• Nylon’s cost is less then metal • Verification of results • Light and small • Easy machining • No wear and tear of tool	• High machining cost	Nylon = 7000 Rs Machining = 80,000 Rs
Mild steel scale model of multiple airfoil wings (to be tested in wind tunnel)	• Less manufacturing cost as compared to other techniques for multiple airfoils • One mold used for multiple airfoils		Mild steel = 12,000 Rs Machining = 1,25000 Rs
CNC Machining of an Aluminum Block to form a mold and then layup with a material like fiber glass.	• Very Accurate • Very Precise • Mold can be used for long term	• Very Costly • CNC machining of metal is a lengthy process. • Aluminum wont withstand stresses	Machining = > 100,000 Rs Layup = 10,000 Rs
CNC Machining of a Wood Block to form a mold and then layup with material like fiber glass.	• Very Accurate • Very precise • Mold can be re-used. • A light mold	• Very Costly • Wooden mold can be easily damaged.	Wood = 10,000 Rs Machining = 55,000 Rs Layup = 10,000 Rs
(For Actual wing) Mold of hard board: Hard board pieces can be cut and then can be joined in a pattern to form a mold by glue or The pieces can be aligned at a distance and then an aluminum sheet can be raped over it to form a mold.	• Less price. • Fast process • Light Weight	• Not very accurate • Hard board swells when comes in contact with water • In rapping of sheet the negative side tend to move upside due to tension thus non uniform thickness • Can’t be used for scale model • Not good surface finish	Hard board = 8,000 Rs Layup = 10,000 Rs

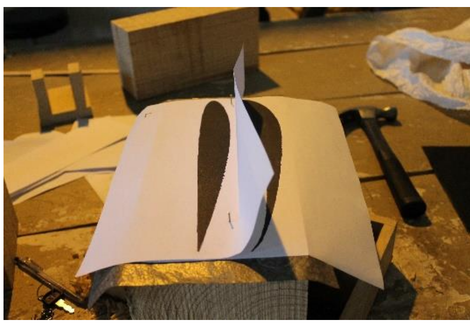
SECTION-E

MANUFACTURING AND EXPERIMENTAL TESTING OF SCALE MODEL WINGS

TEST MODEL:

Test model was manufactured from Balsa wood. Balsa is a soft wood. Its cutting can be easily done. Steps involved in making test model are:

a) CAD modeling in Solid works (Original dimensions)
b) Printing out the side view of airfoil on A4 sheet
c) Cutting out the airfoil part from paper and making it as a stencil to draw on wooden block using carbon paper
d) The required sketch can be cut out using bent saw
e) Airfoil is made in 3 pieces due to maximum cutting length of bent saw
f) All 3 pieces are joined together using Gym saw, a strong bonding agent
g) Finally surfaces are polished



Stencil of airfoil

Sketch of airfoil

Bent saw cutting

Surface smoothing using foil

Bent saw at MRC

GIKI Wind tunnel Testing:

Table I

Angle of attack (degrees)	Airspeed (m/s)	Lift coefficient	Drag coefficient	Side coefficient
-13	25	-1.8	0.8	-0.2
-13	29.1	-2.5	1.1	-0.2
-13	33.3	-3.3	1.4	-0.3
-13	36	-3.9	1.6	-0.4
-13	37.5	-4.3	1.9	-0.45
-20	25	-2.7	1.5	-0.3
-20	29.1	-3.8	2.0	-0.4
-20	33.3	-5	2.6	-0.5
-20	36	-5.9	2.9	-0.7

SECTION-E

PARALLEL PROCESSING ACROSS A NETWORK

Multiple processors can be used in the form of clusters to perform cluster computing for a given case in OpenFOAM consuming relatively much less time than using a single processor but it's not always the case. It depends on the case to be solved, number of cells, mesh type, number of cores used, operating system specifications (RAM, processor). It is possible that a certain case run faster on one PC using its all cores than running case in parallel because when running in parallel all PCs consume time in communicating with each other. OpenFOAM supports parallel processing using Message Pass Interface (MPI). MPI is a tool used for performing tasks at multiple processors simultaneously. Through MPI OpenFOAM can trigger all the processors connected in a network and use all cores of each PC in simulation. OpenFOAM stores simulation results in huge amount of data at-least 100 GB space should be allocated for Ubuntu before installing it. Keeping at-least 5-10 GB for swap memory. Swap memory uses hard disk space as RAM. For making a good cluster, high performance operating systems having good RAM should be used. Before starting to make cluster, each section of the document should be read carefully. This document shows how to setup a Beo-wolf type cluster using individual Pcs connected using LAN (hub). Improvement suggestions are given at the end section.

INSTALLATION OF UBUNTU AND OPENFOAM:

Ubuntu can be downloaded from http://www.ubuntu.com/download. The ISO image is burnt in usb to make a bootable usb. Ubuntu should be installed as a second operating system besides windows (dual boot). After successful Ubuntu installation, OpenFOAM can be downloaded from http://www.openfoam.org/download/ubuntu.php. It's a debian package which will run on Ubuntu only. This tutorial will show how to setup a cluster on Ubuntu 12.04LTS with OpenFOAM version 2.2.1 and Paraview 3.2.1.0 installed. An important consideration, that a particular version of Paraview is compatible with OpenFOAM version. Here in this tutorial Paraview 3.1.2.0 version is used with OpenFOAM 2.2.1. All systems should have same Ubuntu and OpenFOAM versions.

CHALLENGES IN SETTING UP A CLUSTER IN UBUNTU OS:

There are certain issues with Ubuntu due to which setting up a cluster is not an easy job. First of all syntax should be properly followed. A single space not given will not perform the desired task. If a command does not work try it again. There are some built-in issues with Ubuntu operating system due to which command should be tried a few times to make it work. Ubuntu is case-sensitive.

CONDITIONS FOR MPI JOB:

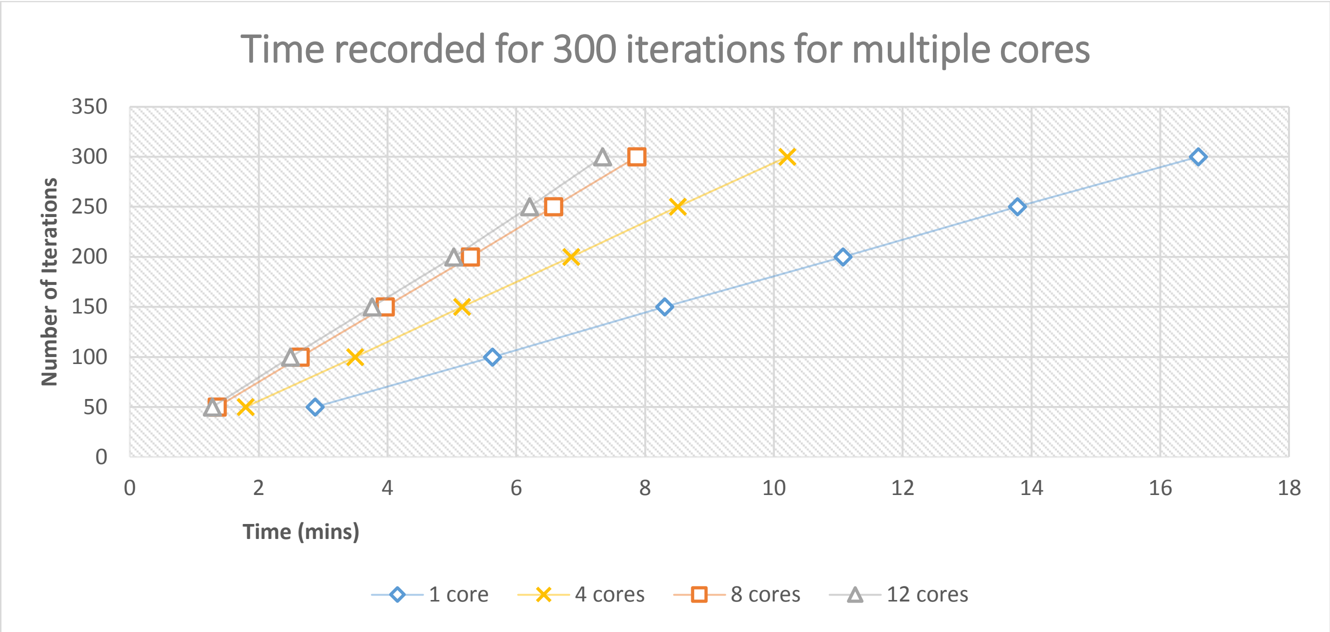
To perform an MPI job all the PCs should be connected through LAN (Local Area Network). They can be connected using a hub with LAN cables or a switch. There are certain conditions which should be followed before starting to perform parallel computing. These are listed below:

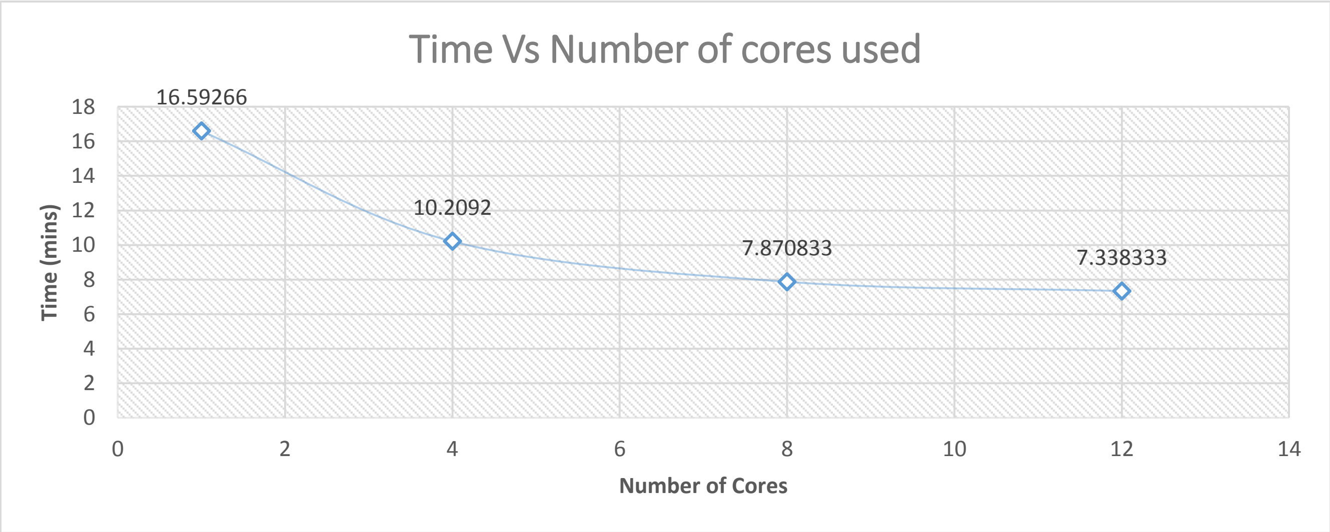
1. Every PC should have same specification. E.g. all PCs taking part in cluster computing should have same processor, same RAM Operating system should be 32 or 64 bit for all Pcs. 64 bit preferable
2. Same Ubuntu version installed
3. Same OpenFOAM and Paraview version installed
4. Same case folder should be copied in same directory for all Pcs (generally home folder)
5. Running "decomposePar" in all Pcs for a given case
6. Successful SSH "Secure shell" and ping done on all Pcs

If these are not followed then there are possibilities that MPI will not initiate. Also if it does initiate for a case having different specifications Pcs connected in a cluster, then the slower one will lag behind the others and MPI will be forced to quit its job.

GRAPHICAL COMPARISON OF CLUSTERS:

Graph is plotted for time against the number of iterations completed for above case to summarize the tables. Graph depicts that as the number of cores is increased, time reduction becomes smaller. This is due to slow LAN connection speed limited to only 100 Mbps, as more time is wasted in communicating among Pcs in network. Increasing LAN speed to 1 Gbps will improve time reduction and need for using more cores for simulation. The Parabolic curve shows rapid decrease in time when using 4 cores instead of one on 1 Pc. As more cores are added through network, time reduction is decreased because there is a percentage of time which is wasted in communication.





IMPROVEMENT SUGGESTIONS:

The more powerful systems making up the cluster, more time reductions in simulation times will occur. RAM should be 8 GB or more. 1 GB LAN connection speed will give best performance. A switch should be used rather a hub as the later one divides speed into Pcs connected across it. Above simulations were run using 100 Mbps connection speed connected using a hub. Time reductions were not too significant due to slow communication medium used to connect cores. At least 5-10 Gb space should be allocated for swap memory. A good LAN speed have minimum latency. Latency can be checked by pinging another Pc. Time = 0.1-0.3 ms will give good simulation time. More cores can be added to reduce simulation time at the cost of high connection speed i.e. 1 Gbps. OpenFOAM simulations generally lasts more than 30 hours if case is heavy. Using clusters can help a lot in reducing time.

TIME SPENT ON SIMULATIONS:

TASK	DURATION
LAMINAR SIMULATIONS	November 2013 – 1st week of January 2014 Approximate completion time for one simulation = 38 hours (without cluster) Total simulations done = 13
CLUSTER PREPARATION	January 2014 – March 2014 Cluster was used 5 days a week (Monday - Friday) for almost 12 hours per day
TURBULENT SIMULATIONS	15th April 2014 - May 2014 Approximate completion time for one simulation = 60 hours (with 26 core cluster) Total simulations done = 4

REFERENCES:

[1] www.airfoiltools.com

[2] http://www.openfoam.org/docs/user/running-applications-parallel.php

[3] www.slideshare.net

[4] Wilcox, D. C., *Formulation of the k-omega Turbulence Model Revisited*, AIAA Journal, Vol. 46, No. 11, 2008, pp. 2823-2838