



MARMARA UNIVERSITY
FACULTY OF ENGINEERING



AERODYNAMIC ANALYSIS OF ONERA M6 & RAE2822 AIRFOILS WITH OPENFOAM

MEHMET AYBARS SEYHAN, ONUR KOÇ

GRADUATION PROJECT REPORT
Department of Mechanical Engineering

Supervisor
Prof. Dr. Emre ALPMAN

ISTANBUL, 2021



MARMARA UNIVERSITY

FACULTY OF ENGINEERING



Aerodynamic Analysis of ONERA M6 & RAE2822 Airfoils with OpenFOAM

by

Mehmet Aybars SEYHAN, Onur KOÇ

26 June 2021, Istanbul

**SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF**

BACHELOR OF SCIENCE

AT

MARMARA UNIVERSITY

The author(s) hereby grant(s) to Marmara University permission to reproduce and to distribute publicly paper and electronic copies of this document in whole or in part and declare that the prepared document does not in anyway include copying of previous work on the subject or the use of ideas, concepts, words, or structures regarding the subject without appropriate acknowledgement of the source material.

Signature of Author(s) 

Department of Mechanical Engineering

Certified By Prof. Dr. Emre Alpman 

Project Supervisor, Department of Mechanical Engineering

Accepted By
Head of the Department of Mechanical Engineering

ACKNOWLEDGEMENT

First of all, I would like to thank our supervisor Prof. Dr. Emre ALPMAN for the valuable guidance and advice on preparing this thesis and giving me moral and material support.

June, 2021

Mehmet Aybars SEYHAN, Onur KOÇ

CONTENTS

Table of Contents

ACKNOWLEDGEMENT	i
CONTENTS	ii
ABSTRACT	iv
SYMBOLS	v
ABBREVIATIONS	vi
LIST OF FIGURES	vii
LIST OF TABLES	x
1. INTRODUCTION	2
1.1. What is Aerodynamics	2
1.2. What is Computational Fluid Dynamics (CFD)	3
1.3. What Does CFD Analysis Do	3
1.4. Importance of Open Source Softwares	4
1.5. ONERA M6 & RAE2822 Airfoils	5
2. MATERIALS AND METHODS	6
2.1 OpenFOAM	6
2.2 Meshing Tools	6
2.2.1 GMSH	6
2.2.2 BlockMesh	6
2.2.3 SnappyHexMesh	7
2.3 Finite Volume Method	9
2.4 Governing Equations	9
2.5 Main OpenFOAM Solvers Used in Project	10
2.5.1 rhoSimpleFoam	10
2.5.2 RhoPimpleFoam	11
2.5.3 rhoCentralFoam	11

2.6 Courant Number:	12
2.7 Adjustable Time Step:	12
2.8 Case Creation	13
2.8.1 Initial Conditions and Turbulence Model	13
2.8.2 Mesh	13
2.8.3 fvSchemes and fvSolution Folders	14
3. RESULTS AND DISCUSSION	15
3.1 Mach Number Effects	15
3.1.1. Case 1 Result	16
3.1.2. Case 2 Result	16
3.1.3. Case 3 Result	17
3.1.4. Case 4 Result	17
3.2 RhoPimpleFoam versus RhoCentralFoam	18
3.2.1 RhoPimpleFoam results	18
3.2.2 RhoCentralFoam results	19
3.3 RAE2822 Airfoil with RhoCentralFoam	20
3.3.1 Case 1 Results	21
3.3.2 Case 2 Results	22
3.3.3 Case 3 Results	24
3.3.4 Case 4 Results	25
3.4 ONERA M6 Results	27
3.4.1 RhoSimpleFoam Results	27
3.4.2 RhoPimpleFoam Results	29
3.4.3 RhoCentralFoam Results	32
3.5 Consideration of Constraints	34
4. CONCLUSION	35
REFERENCES	36
APPENDICES	36

ABSTRACT

Aerodynamic Analysis of ONERA M6 & RAE2822 Airfoils with Open-FOAM

In this study, we have investigated flows with different Mach numbers around RAE2822 2-dimensional airfoil. And also, we have investigated in specific angle of attack. We observed different shock waves on transonic and supersonic flows. To do this, firstly we have done some analysis on NACA0012 airfoil. By doing that we have determined our main solver as a RhoCentralFoam. After that we have generated our airfoil mesh on GMSH. Then we have created our modified solver for RAE2822 airfoil. Afterwards we have simulated in different Mach numbers.

After completing 2-dimensional analysis, we have studied ONERA M6 wing in transonic conditions. We have picked rhoSimpleFoam, RhoPimpleFoam and rhoCentralFoam solvers and compared the results with NASA's experimental datas.

At the end, our main aim is to get experimental values in with these solvers in Open-FOAM and compared the solvers.

Keywords: OpenFOAM, NACA0012, RAE2822, RhoSimpleFoam, RhoCentralFoam, RhoPimpleFoam, GMSH, ONERA M6

SYMBOLS

a : Sound of Speed (m/s)

C_d : Drag Coefficient

C_L : Lift Coefficient

V : Velocity (m/s)

ρ : Density (kg/m³)

γ : Specific Heat Ratio

R : Specific Gas Constant (J/kg*K)

T : Atmospheric Temperature (K)

ABBREVIATIONS

- CFD** : Computational Fluid Dynamics
- FOAM** : Field Operation and Manipulation
- GAMG** : Geometric Agglomerated Algebraic Multigrid Solver
- PBiCGStab** : Stabilized preconditioned (bi-)conjugate gradient

LIST OF FIGURES

	PAGE
Figure 1.1. Drag and Lift Forces	2
Figure 1.2. ONERA M6 Airfoil	5
Figure 1.3. RAE2822 Airfoil Section	5
Figure 3.1. Case 1 Velocity Result.....	9
Figure 3.2. Case 2 Velocity Result.....	10
Figure 3.3. Case 3 Velocity Result.....	10
Figure 3.4. Case 5 Velocity Result.....	11
Figure 3.5. RhoPimpleFoam Velocity Result	12
Figure 3.6. RhoPimpleFoam Pressure Result	12
Figure 3.7. RhoCentralFoam Velocity Result.....	13
Figure 3.8. RhoCentralFoam Pressure Result	13
Figure 3.9. Case 1 Velocity Result.....	14
Figure 3.10. Case 1 Pressure Result.....	15
Figure 3.11. Case 1 Temperature Result.....	15
Figure 3.12. Case 2 Lift/Drag Coefficients	16
Figure 3.13. Case 2 Velocity Result.....	16
Figure 3.14. Case 2 Pressure Result.....	17
Figure 3.15. Case 2 Temperature Result	17

Figure 3.16. Case 3 Velocity Result.....	17
Figure 3.17. Case 3 Pressure Result.....	18
Figure 3.18. Case 3 Temperature Result.....	18
Figure 3.19. Case 4 Lift/Drag Coefficients	18
Figure 3.20. Case 4 Velocity Result.....	19
Figure 3.21. Case 4 Pressure Result.....	19
Figure 3.22. Case 4 Temperature Result.....	19
Figure 3.23 -Cp/x Graph of RAE2822 Result Comparison with NASA Experimental Values	20
Figure 3.24. Onera M6 rhoSimpleFoam Pressure Distribution	27
Figure 3.25. rhoSimpleFoam Cp Distribution at 0.2.....	27
Figure 3.26. rhoSimpleFoam Cp Distribution at 0.44.....	28
Figure 3.27. rhoSimpleFoam Cp Distribution at 0.80.....	28
Figure 3.28. rhoSimpleFoam Cp Distribution at 0.95.....	29
Figure 3.29. Onera M6 rhoPimpleFoam Pressure Distribution	29
Figure 3.30. rhoPimpleFoam Cp Distribution at 0.2.....	30
Figure 3.31. rhoPimpleFoam Cp Distribution at 0.44.....	30
Figure 3.32. rhoPimpleFoam Cp Distribution at 0.80.....	31
Figure 3.33. rhoPimpleFoam Cp Distribution at 0.95.....	31

Figure 3.34. Onera M6 rhoCentralFoam Pressure Distribution.....	32
Figure 3.35. rhoCentralFoam Cp Distribution at 0.2	32
Figure 3.36. rhoCentralFoam Cp Distribution at 0.44	33
Figure 3.37. rhoCentralFoam Cp Distribution at 0.80	33
Figure 3.38. rhoCentralFoam Cp Distribution at 0.95	34

LIST OF TABLES

	PAGE
Table 2.1. Initial Case Values for Onera M6 Wing.....	24
Table 3.1. Mach Number Effect Case Properties.....	26
Table 3.2. Mach Number Values for Case Properties.....	27
Table 3.3. RAE2822 NASA Experimental Conditions	32

1. INTRODUCTION

In this project, we aim to realize aerodynamic analysis of ONERA M6 and RAE2822 airfoils by using OpenFOAM software. But we are not familiar with the OpenFOAM. For this reason, we go step by step. In first semester, we will realize two-dimensional analysis of the airfoil. In second semester, we will realize three-dimensional analysis of airfoil and at the end, we will compare results with the experimental results. This report for the first semester. So, it includes only two-dimensional analysis.

By doing this project, we will show and verify the OpenFOAM in terms of availability and reliability for the industry.

1.1. What is Aerodynamics

Aerodynamic is a branch of fluid dynamics which investigate of the fluid or air movements and also, forces resulting from these movements. In our case, we will investigate the airfoil and as examples of created forces are:

- Drag
- Lift

Here, drag is force which fluid flowing exerts on the body in flow direction. It depends on the surface roughness, normal area to the flow etc. Lift is force which is created by the pressure difference between top and bottom of the airfoil and also by wall shear forces.

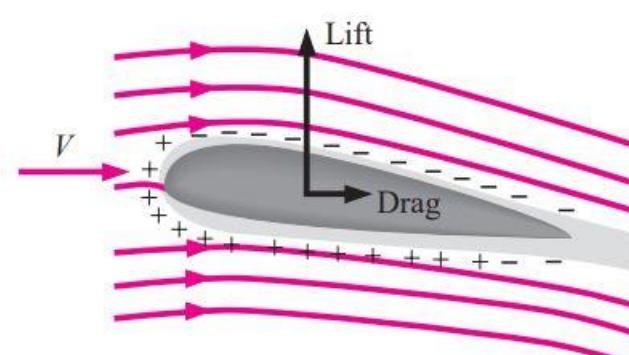


Figure 1.1. Drag and Lift Forces

1.2. What is Computational Fluid Dynamics (CFD)

With developing technology and computer processors, computers have been very powerful machines to model physical problems and to solve mathematical equations. In this way computational fluid dynamics (CFD) branch was emerged. CFD is subbranch of fluid dynamics. In CFD, data structures and numerical analysis is used to solve fluid flow problems. At this point, computers are used to visualize of the fluid flows and solve complex equations also used to analyze results. To get correct solutions, user should combine the suitable physical laws and suitable mathematical equations. Also, should define boundary conditions correctly.

To get better solution and to get fast results, more powerful computers should be used. Otherwise, we will not get the solution, or it will take days.

1.3. What Does CFD Analysis Do

To investigate of any fluid flow, two type of method is used. One of them is experimental method, other one is theoretical method. Both of them have some advantages and disadvantages. We can get more reliable results in the experimental method, but it takes more time to perform the experiment and it is also expensive. Especially while doing strength test. Because we have to use new part for each broken part. On the other hand, in theoretical experiments, we get approximate solutions, but we can perform it faster than experimental method and also it is cheaper than.

In CFD, we are performing theoretical method. It provides us to being fast and cheaper. Also, it gives us to preliminary information and provides to decrease experimental tests. Before doing experimental tests, we can see the possible results and we can change our design or conditions. So, we can see our deficient sides and we can improve the design. At the end it allows us to perform experimental test more accurately and less. By doing this, we can gain time and money.

CFD is not only used for fluid flows but also used for heat transfer, chemical reactions, mass transfer etc. Considering the area of use CFD, it is used in many different areas such as:

- Automotive Industry
- Metallurgy
- Aviation and Space
- Sport
- Biomedical
- Chemical and Mineral Processing
- Power Production and Renewable Energy

As a computer software of CFD, there are two types of software: One of them is commercial softwares such as Fluent, CFX, StarCD, Polyflow, Flow3D, SCRYU etc. Other one is open source softwares such as SU2, OpenFOAM.

In our project, we will use OpenFOAM.

1.4. Importance of Open Source Softwares

Using open source software has both advantages and disadvantages. Here, we speak about the OpenFOAM software. The advantages using it are more important than its disadvantages. Firstly, it is totally free, whatever you make with it, you will pay nothing. Especially while commercial software fees are too high. This saves a large amount of money in your pocket at the end of year. Other advantage is that you can manipulate the code and you get better it for your cases. Also, in big project especially military or territorial project, in commercial ones can restrict your project. But OpenFOAM, you don't encounter with that, you can easily do your confidential project.

As disadvantages, you have no interface to use program easily. For this reason, it is not user friendly. On the other hand, it is not well-known program. You may not easily find a solution for your errors or problems.

1.5. ONERA M6 & RAE2822 Airfoils

These airfoils are widely used in turbulence modelling. Both are in transonic airfoils.

ONERA M6 is a classic airfoil used to validate CFD results in transonic flows. It is also used for validating numeric and turbulent models.



Figure 1.2.ONERA M6 Airfoil

Likewise, RAE 2822 airfoil is used to validate and compare CFD results for transonic flows. Experimental tests of this airfoil were done. So, experimental results are achievable from internet and usable for comparing numerical results.

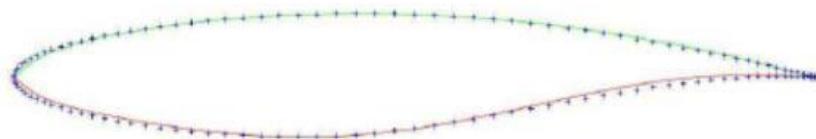


Figure 1.3. RAE 2822 Airfoil Section

2. MATERIALS AND METHODS

2.1 OpenFOAM

In today's world, firms are focusing on creating their own softwares with their custom specializations and tend to leave commercial software firms. Therefore, open source softwares are becoming more and more important in our times. In our project, we will OpenFOAM due to its rise across the industry.

OpenFoam is an open source software for computational fluid dynamics (CFD) calculations. OpenFoam name stands for Open Source Field Operation and Manipulation. It was created in 1989 and released open source as 2004. Due to its open source nature, OpenFOAM gives its users ability to modify the software to their own usage. With that, we can say OpenFOAM is a framework based on C++ libraries for developing CFD applications. In program there are solvers and utilities. Solvers are designed to solve various of fluid mechanics problems and utilities are designed to perform data manipulations.

2.2 Meshing Tools

2.2.1 GMSH

Gmsh is an open source 3D finite element mesh generator. It also has built in CAD engine and post-processor. It is a fast and user-friendly meshing tool with visualization capabilities. Gmsh has four main modules: geometry, mesh, solver and post-processing. In this project, Gmsh will be our main meshing tool.

2.2.2 BlockMesh

In this section, we will explain the blockMesh command. BlockMesh is a mesh generation tool which is supplied with OpenFOAM. By using blockMesh we can create parametric meshes with grading and curved edges. The blockMesh command reads *blockMeshDict* dictionary from the *system* folder.

BlockMesh basically decompose the domain geometry to groups of 1 or more three dimensional, hexahedral blocks. Hexahedral blocks edges can be lines, splines or arcs. The mesh is categorized as a number of cells in each direction of the block.

Every block is defined by its 8 vertices, one for every corner of the hexahedron. The

vertices are written in a list so that each vertex can be accessed using its label, remembering that OpenFOAM always uses the C++ convention that the first element of the list has label ‘0’. An example hexahedron can be seen from the figure below.

The local coordinate system is defined by the order in which the vertices are presented in the block definition according to:

- the axis origin is the first entry in the block definition, vertex 0 in our example;
- the x_1 direction is described by moving from vertex 0 to vertex 1;
- the x_2 direction is described by moving from vertex 1 to vertex 2;
- vertices 0, 1, 2, 3 define the plane $x_3 = 0$;
- vertex 4 is found by moving from vertex 0 in the x_3 direction;
- vertices 5,6 and 7 are similarly found by moving in the x_3 direction from vertices 1,2 and 3 respectively.

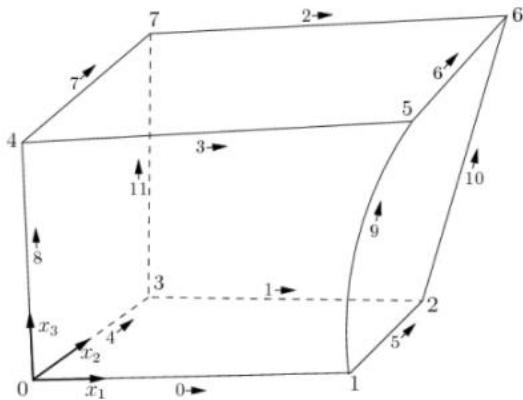


Figure 2.1 BlockMesh Mesh Creating Principles

Our blockMeshDict script is written in Appendix of this paper.

2.2.3 SnappyHexMesh

SnappyHexMesh is a mesh generation utility supplied with OpenFOAM. Its purpose is creating mesh where the hexahedrons are insufficient. It works by clipping the object that user created with a CAD program from the blockMesh. SnappyHexMesh utility needs blockMesh to work.

At first, user has to create his/her object which is going to be analysed. It should be saved as Stereolithography (STL) or Wavefront Object (OBJ) format. User needs to create a blockMeshDict which contains the desired object. After that snappyHexMesh tool confirms the objects place by iteratively refining the starting mesh and combining with the current

mesh. The sensitivity and refining degrees given by the user. The user should use overwrite command when utilizing snappyHexMesh due to after mesh refinement it changes the time step.

SnappyHexMesh utility working states that shown in the figure below from left to right.

1. Desired mesh
2. Initial mesh generation
3. Cell splitting by objects edge
4. Cell splitting by surface of the object
5. Cell removal in snappyHexMesh meshing process
6. Cell splitting by region
7. Surface snapping
8. Layer addition in snappyHexMesh

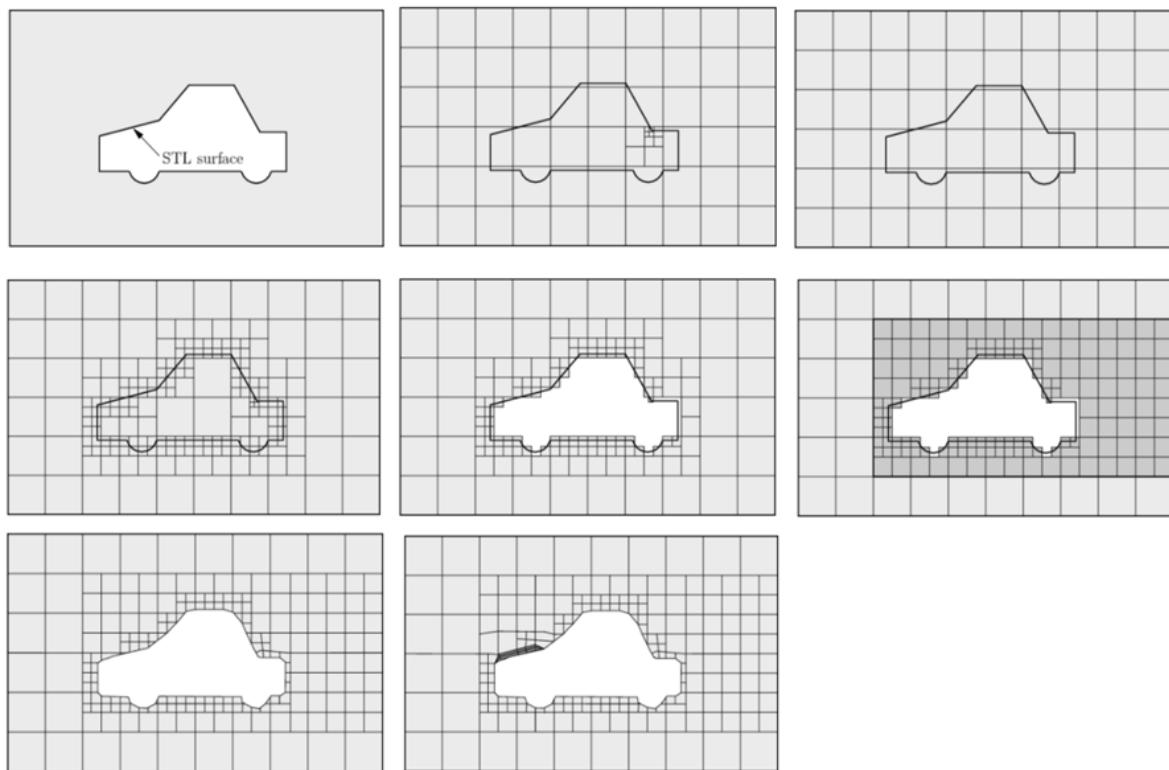


Figure 2.2 SnappyHexMesh Working Principle Steps

2.3 Finite Volume Method

Finite volume method transforms a series of partial differential equations into linear algebraic equations like other numerical methods created for fluid flow simulations. Nonetheless, discretization procedure applied in the finite volume method is differs from others and proceeds with two steps. Primarily, the first step is partial differential equation integration and turning them into a balance equations over an element. This step includes changing the surface and volume integrals into discrete algebraic relations over elements and their surfaces using an integration quadrature of a specified order accuracy. The result is a set of semi-discretized equations. In the second step, interpolation profiles are chosen to approximate the variation of the variables within the element and relate the surface values of the variables to their cell values and thus transform the algebraic relations into algebraic equations.

In OpenFOAM user guide it states that;

“OpenFOAM uses finite volume method to solve systems of partial differential equations ascribed on any 3D unstructured mesh of polyhedral cells. The fluid flow solvers are developed within a robust, implicit, pressure-velocity, iterative solution framework, although alternative techniques are applied to other continuum mechanics solvers. Domain decomposition parallelism is fundamental to the design of OpenFOAM and integrated at a low level so that solvers can generally be developed without the need for any parallel-specific coding.”

2.4 Governing Equations

In this project, we will focus on compressible flows due to high Mach Numbers of the aircrafts. The general governing equations depends on conservation of mass and conservation of momentum and conservation of energy.

	$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) = 0$	(1)
--	--	-----

or

	$\frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) = 0$	(2)
--	---	-----

Equation 1 and 2 are known as conservative and non-conservative forms of mass equations respectively.

Conversation of linear momentum is known as Navier-Stokes equation (In CFD literature the term Navier-Stokes is usually used to include both momentum and continuity equations, and even energy equation sometimes).

	$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \nabla \cdot \bar{\tau} + \rho \vec{f}$	(3)
--	---	-----

Where \vec{f} is the body force unit mass and $\bar{\tau}$ is the viscous stress tensor.

2.5 Main OpenFOAM Solvers Used in Project

2.5.1 rhoSimpleFoam

rhoSimpleFoam is a solver for compressible and turbulent flows. It is steady-state solver and pressure based. It uses simple (Semi Implicit Method for Pressure Linked Equations) algorithm. And also, this solver is pressure-based solver which firstly pressure equation is solved and density is related with the pressure. This relation is provided with equation of state.

This solver solves the equations for each variable (u, p, or internal energy) sequentially. Firstly, momentum equation is solved, after that energy equation is solved. Then continuity and momentum equation is used to create an equation for pressure. After getting pressure field, it is inserted in momentum equation and velocity field is corrected. Then equations are solved for turbulence. These steps are repeated until converging.

As input requirements for this solver, user should define velocity, pressure and temperature.

	$\int_{CS} \rho * \vec{V}^* (\vec{V} * \vec{n}) dS = - \int_{CS} p * \vec{n} dS$	(1)
--	--	-----

	$\nabla * (\rho * \vec{V}) = 0$	(2)
--	---------------------------------	-----

	$\int_{CS} \rho^* (e + \frac{V^2}{2})^* (\vec{V} * \vec{n}) dS = \dot{Q} - \int_{CS} p^* (\vec{V} * \vec{n}) dS$	(3)
--	--	-----

First equation is momentum equation, second one is continuity and last one is energy equation.

2.5.2 RhoPimpleFoam

RhoPimpleFoam is also a solver for compressible and transient flows. But it is transient solver. Generally, it is used for HVAC or similar cases. It uses pimple (Piso-Simple) algorithm. That means our case is transient, but we use simple algorithm to find steady state solution for each time step. This allows us to transient solutions for higher courant numbers. If users don't write anything in pimple directory, the solver is converted automatically piso algorithm. In piso algorithm courant number must be smaller than 1 for stability.

Input requirements are same with the rhoSimpleFoam.

$$\frac{d}{dt} * \int_{CV} \rho * \vec{V} d\nu + \int_{CS} \rho * \vec{V} * (\vec{V} * \vec{n}) dS = - \int_{CS} p * \vec{n} dS \quad (1)$$

$$\frac{\partial p}{\partial t} + \nabla * (\rho * \vec{V}) = 0 \quad (2)$$

$$\frac{d}{dt} * \int_{CV} \rho * (e + \frac{V^2}{2}) d\nu + \int_{CS} \rho * (e + \frac{V^2}{2}) * (\vec{V} * \vec{n}) dS = \dot{Q} - \int_{CS} p * (\vec{V} * \vec{n}) dS \quad (3)$$

This time governing equations have time derivative terms because this solver is transient.

2.5.3 rhoCentralFoam

rhoCentralFoam is solver for transient compressible flows based on central-upwind schemes of Kurganov and Tadmor which is combination of central difference and upwind schemes. Central-upwind schemes discretize convective terms of governing equations. rhoCentralFoam is density based solver.

Each solver solves momentum equation to get velocity field. But while getting pressure field, pressure-based solvers set up pressure correction equation which is obtained by manipulating momentum and continuity equations. On the other hand density-based solver (rhoCentralFoam) firstly obtains density field from continuity equation then get pressure field by using equation of state.

$$\frac{\partial}{\partial t} \rho + \nabla(\rho * \vec{V}) = 0 \quad (1)$$

$$\frac{\partial(\rho*\vec{V})}{\partial t} + \nabla * (\rho * \vec{V}^2 + p) = 0 \quad (2)$$

$$\frac{\partial}{\partial t}E + \nabla(\vec{V}(E+p)) = 0 \quad (3)$$

These are the Euler equations where

$$E = \rho e + \frac{1}{2} \rho * V^2$$

2.6 Courant Number:

Courant number is a number which is nondimensional. It is used in CFD simulations to calculate time step requirements of transient simulations. And also, it uses for stability. Its formula is

$$C = \frac{U*dt}{dx} \quad (1)$$

where U is the velocity, dt is the time step and dx is the characteristic size of the mesh cell. In simulations, for stability courant number generally is kept smaller than 1.

2.7 Adjustable Time Step:

Adjustable time step is a good feature for transient simulations. It adjusts the time step (deltaT) according to maximum courant number which is specified by the user. Until the reaching maximum courant number time step can grow unless solution diverging. By this way, it accelerates the solution.

2.8 Case Creation

2.8.1 Initial Conditions and Turbulence Model

Since we wanted to compare our results with NASA's experimental values, the initial conditions created with the information on their web page. Our initial conditions are shown in the table below.

Mach	Pressure	Temperature	Angle of Attack	Velocity
0.8395	101000 Pa	298 K	3.06 degree	290.32 m/s

Table 2.1 Initial Case Values for Onera M6 Wing

For this case we use Spalart-Allmaras turbulence model. Spalart-Allmaras model design for aerospace related purposes which include wall boundaries. Our wing placed onto a symmetric patch boundary so we decided that Spalart-Allmaras would be appropriate. The other reason is, our mesh size and mesh refinement was so fine that our numerical solutions were so slow due to our computational power. Spalart-Allmaras model only solve one equation around turbulent viscosity unlike k-epsilon. So due to save time and computational power we picked this turbulence model.

2.8.2 Mesh

From Micheal Aletto Onera M6 case [10], we have used his prepared mesh. By the help of snappyHexMesh, we have created our mesh file from stl file. This mesh contains level 9 mesh refinement for the surface. At the end we have approximately 1 million cells.

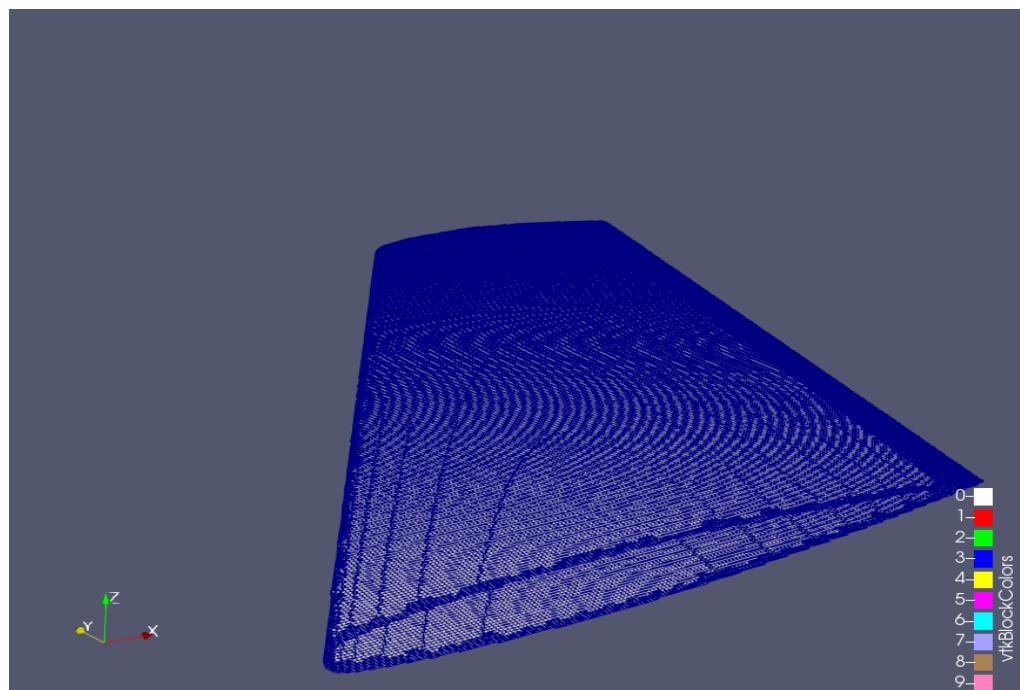


Figure 2.3 Onera M6 Wing Mesh

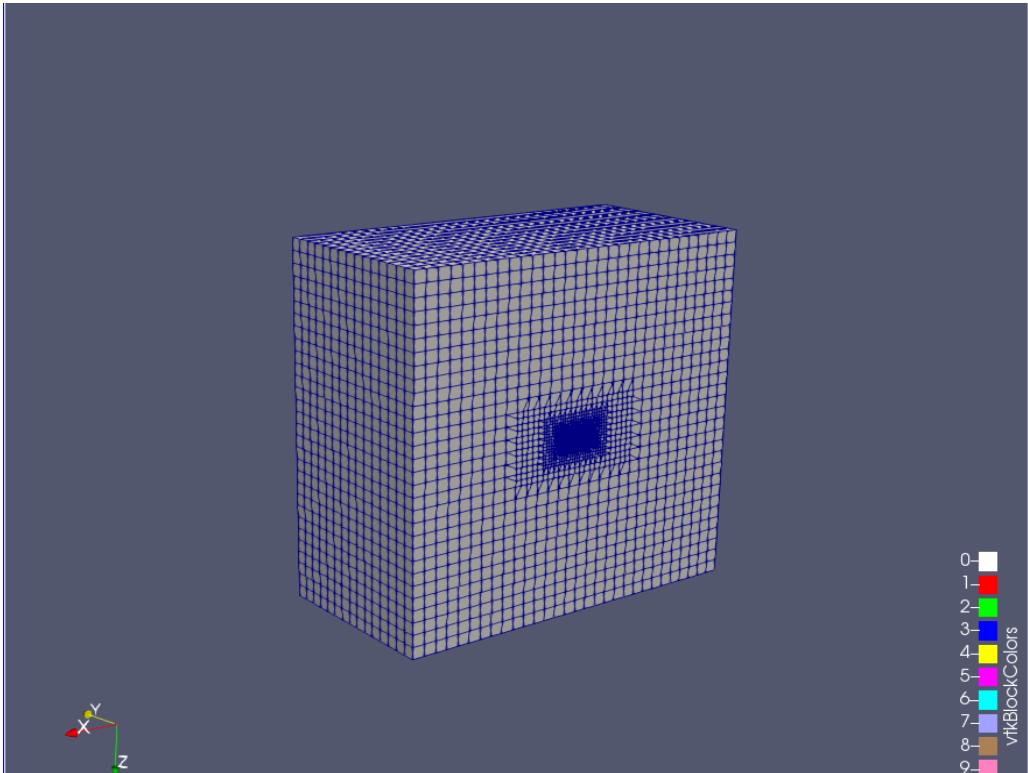


Figure 2.4 Onera M6 Mesh Box

2.8.3 fvSchemes and fvSolution Folders

In general, we have used default fvSolution and fvSchemes folders. For rhoSimpleFoam, in fvSchemes, Gauss linear is used for the gradSchemes and Gauss linerUpwind limited is used. There is no ddt scheme because it is steady state solver. In fvSolution, as a solver GAMG is used and Gauss Seidel is used as a smoother. And also only simple algorithm is used.

For rhoPimpleFoam, in fvSchemes again Gauss linear is used for gradSchemes and Gauss linearUpwind limited is used. Euler is used for ddtSchemes. In fvSolution, GAMG and PBICGstab are used as a solver. And both simple and pimple algorithm is used.

For rhoCentralFoam, flux scheme is Kurganov, ddtSchemes is LocalEuler. For gradSchemes and divSchemes Gauss linear is used. For the interpolationSchemes vanLeer is used for rho, U, T. In fvSolution, solvers are GAMG and diagonal. GAMG is used to solve for k, omega, e and nuTilda and diagonal is used to solve for rho, rhoU, rhoE.

3. RESULTS AND DISCUSSION

3.1 Mach Number Effects

Firstly, we have started with Mach number (Ma) effect to the flows. Mach number is a ratio of flow speed to the sound speed. It specifies of the flow type. If;

$Ma < 0.8$ Subsonic flow

$0.8 < Ma < 1.2$ Transonic flow

$1.2 < Ma$ Supersonic flow

$$Ma = \frac{\text{Object Speed}}{\text{Sound of Speed}} = \frac{V}{a} = \frac{V}{\sqrt{\gamma * R * T}} \quad (1)$$

where;

V = Object Velocity (m/s)

a = Sound of Speed (m/s)

γ = Specific Heat Ratio, generally 1.4

R = Specific Gas Constant (287 J/kg*K)

T = Atmospheric Temperature (K)

This is for Standard Atmospheric Model which assumes a temperature at sea level.
More accurately we can calculate the speed of sound (a);

$$a = 331 + T(^{\circ}\text{C}) \quad (2)$$

We will change the Mach number, by changing object velocity or air temperature, we will see the effects of the different Mach numbers to the airfoil. Our solver is rhoPimpleFoam and our case airfoil is NACA0012. Besides, we have defined the required thermophysical properties according to air temperature in thermophysicalproperties folder. So, our cases and results are:

Case	V(m/s)	A(m/s)	Ma
1	200	356	0.56
2	290	356	0.81
3	290	290	1
4	335	278	1.20

Table 3.1. Mach Number Effect Case Properties

3.1.1. Case 1 Result

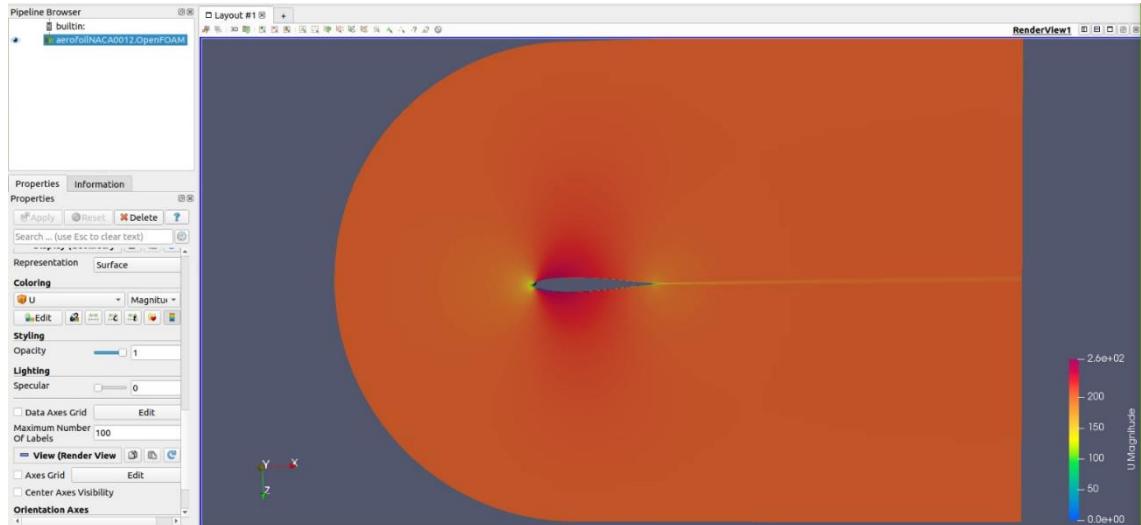


Figure 3.1. Case 1 Velocity(m/s)

3.1.2. Case 2 Result

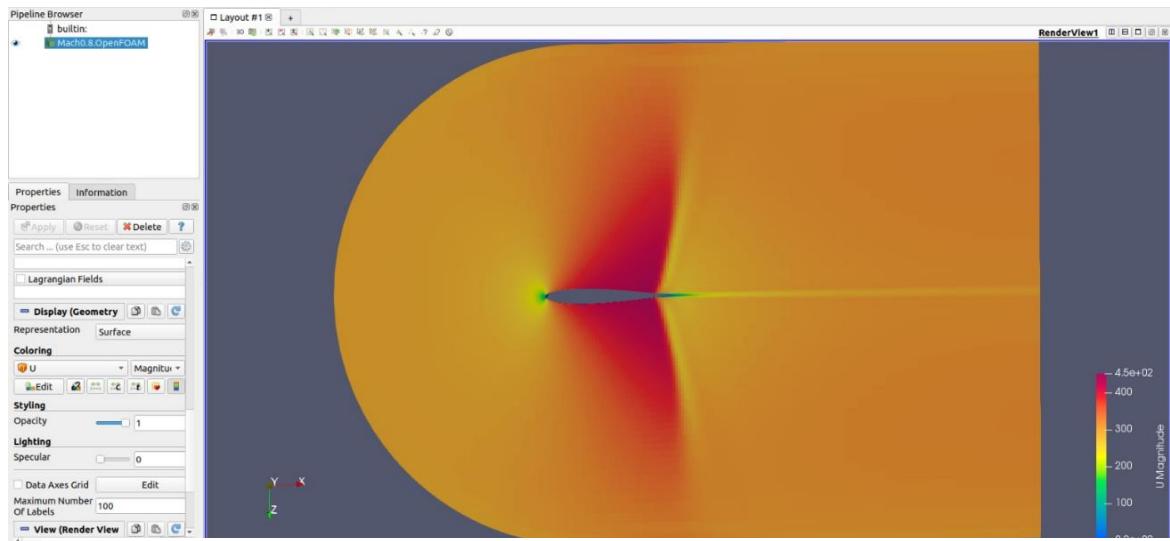


Figure 3.2. Case 2 Velocity (m/s)

3.1.3. Case 3 Result

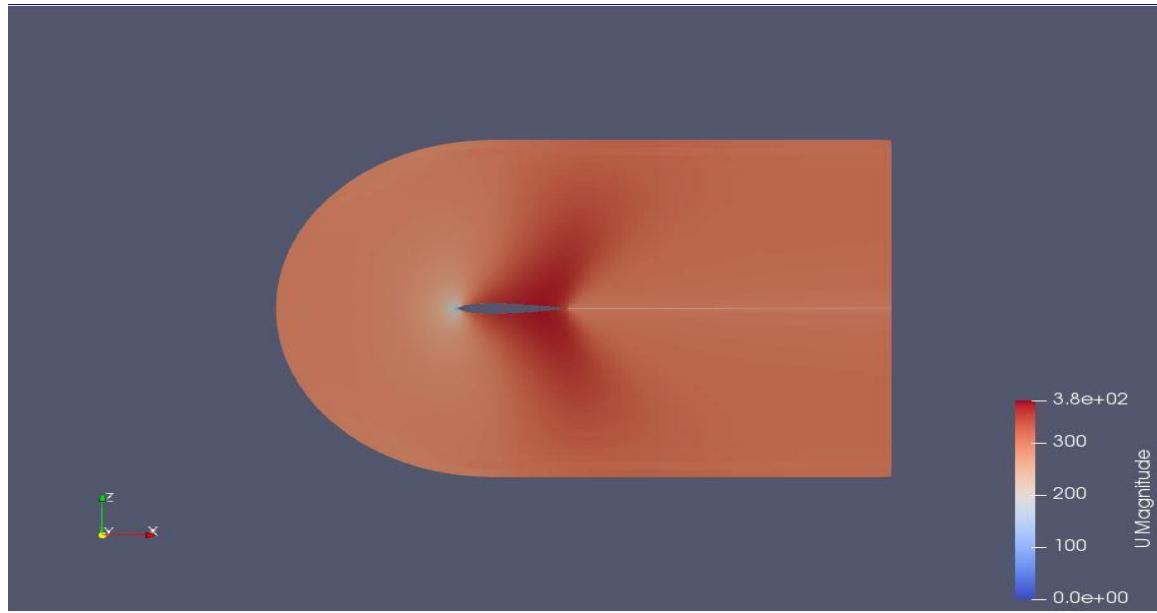


Figure 3.3. Case 3 Velocity(m/s)

3.1.4. Case 4 Result

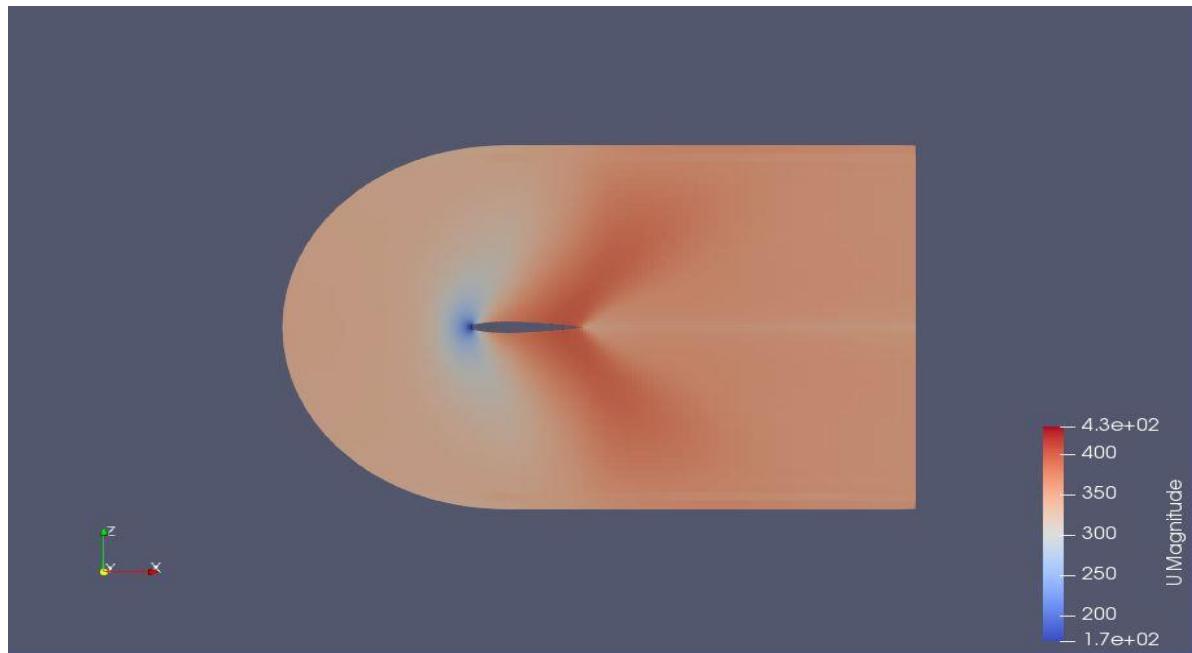


Figure 3.4. Case 4 Velocity(m/s)

From the results, we have seen that there is no shock wave in the first case as we expect, because it is subsonic flow. With second case, shock waves start clearly, and they go to trailing edge of the airfoil from second case up to fourth case.

This preliminary case study provides us to what we will see in our main analysis results. Because our main airfoils are transonic. So, we can expect results similar to case 2 and 3. There will be shock waves on the airfoils. But the locations can be different because of the airfoil geometry, angle of attack or velocity.

3.2 RhoPimpleFoam versus RhoCentralFoam

In this time, we will try to investigate RhoPimpleFoam and RhoCentralFoam. These are both using OpenFOAM solver for compressible flows. RhoCentralFoam density-based solver based on central-upwind schemes of Kurganov and Tadmor. On the other hand, RhoPimpleFoam transient solver for turbulent flow of compressible fluids. We will run them on again NACA0012 airfoil, and then we will compare the results.

3.2.1 RhoPimpleFoam results

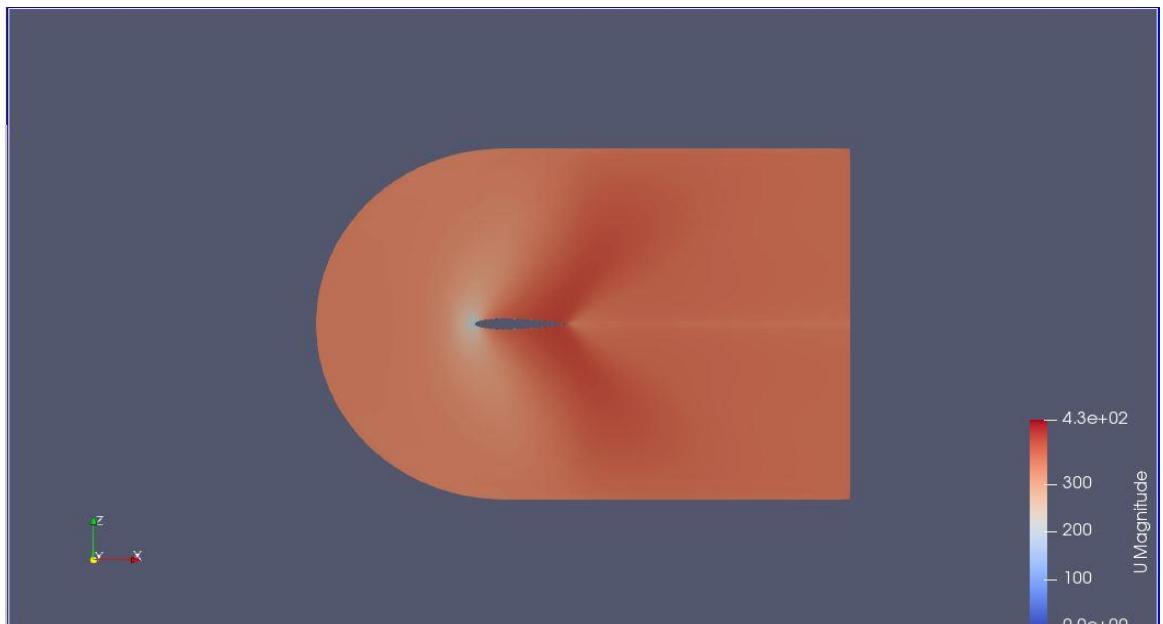


Figure 3.5. RhoPimpleFoam Velocity(m/s)

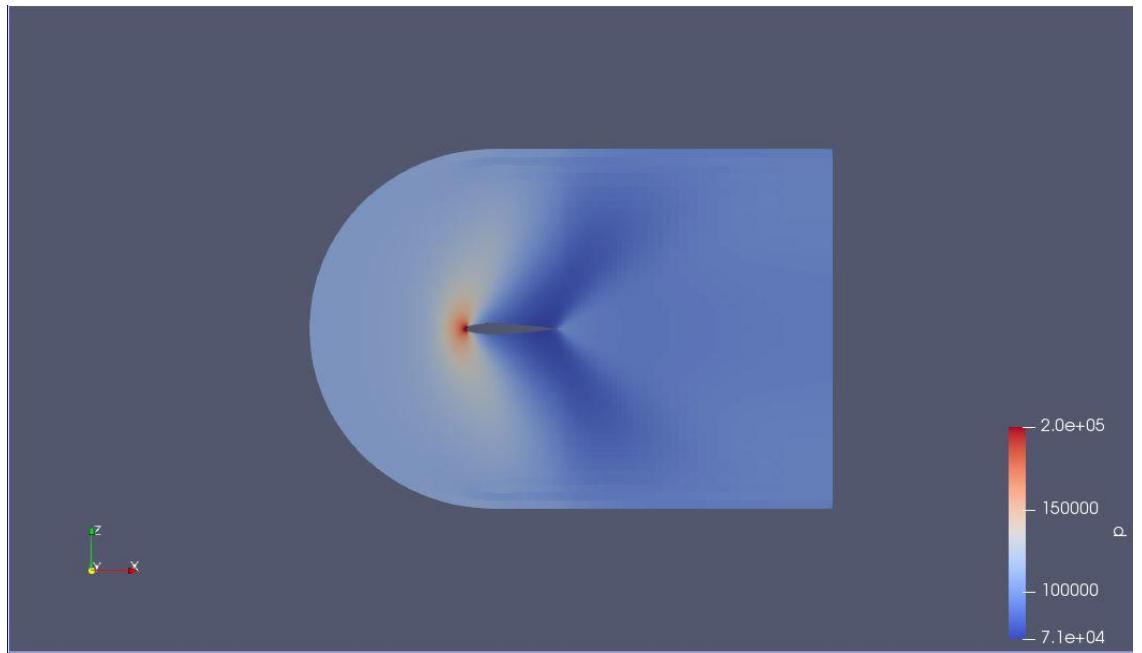


Figure 3.6. RhoPimpleFoam Pressure (Pa)

3.2.2 RhoCentralFoam results

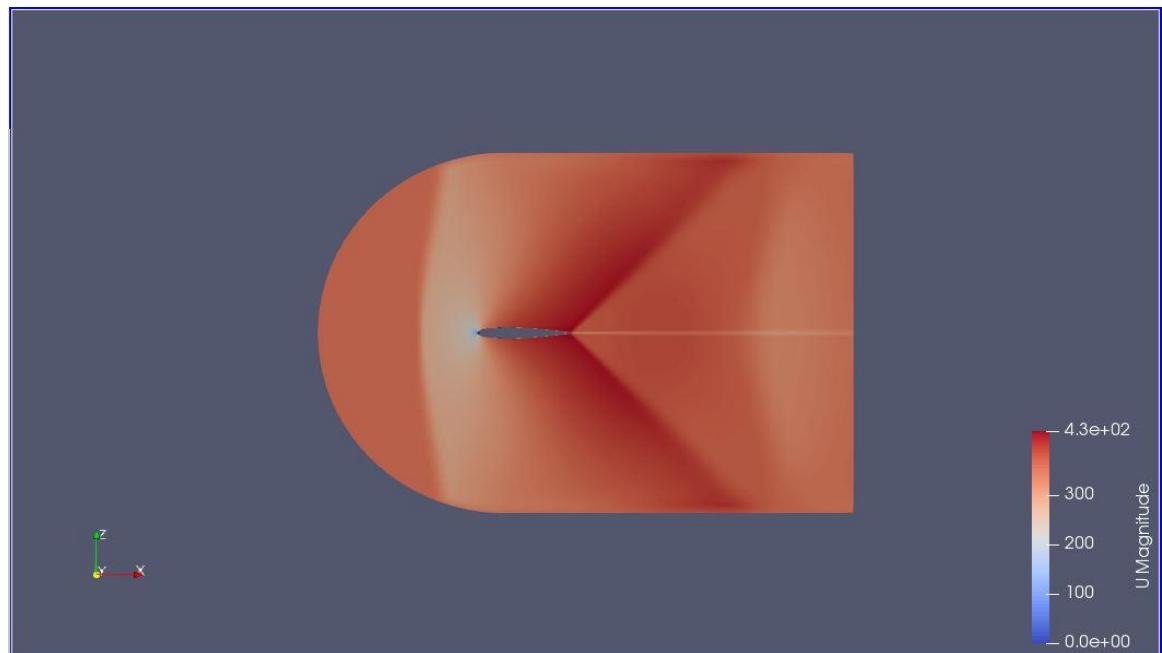


Figure 3.7. RhoCentralFoam Velocity(m/s)

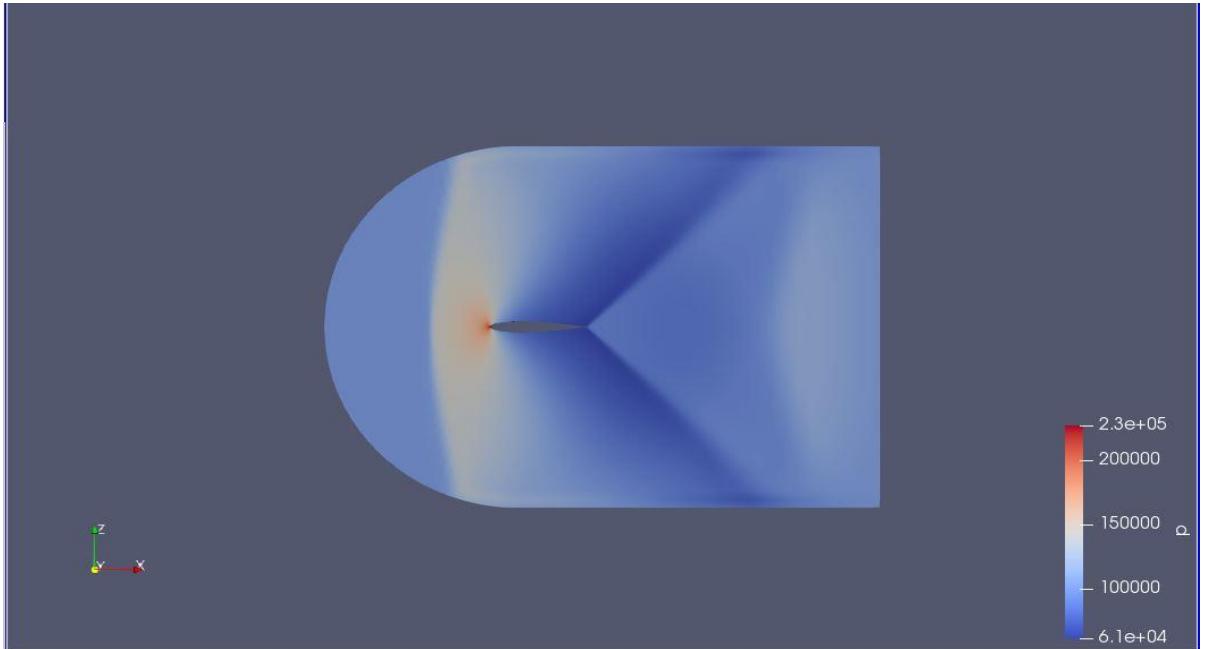


Figure 3.8. RhoCentralFoam Pressure (Pa)

After these results, in RhoCentralFoam we have get sharp results, we can see transitions more easily. It is so important for our main case because we need to find location of the shock waves on transonic airfoil. For this reason, with using RhoCentralFoam we believe that we will get better results. So, after this point of the project, we will use RhoCentralFoam as solver in cases.

3.3 RAE2822 Airfoil with RhoCentralFoam

With the decision we made as choosing RhoCentralFoam as our solver, we proceed to RAE2822 Airfoil simulation. We get the airfoil coordinates from airfoiltools.com and created a “.geo” file as we start. After creating the boundaries of the model and create a mesh. For this foil we added a command that calculates forces and lift/drag coefficients on airfoil.

We did 4 simulations as:

Case	V(m/s)	A(m/s)	Ma
1	242.2	346	0.7
2	311.4	346	0.9
3	380.6	346	1.1
4	445.89	297.26	1.5

Table 3.2. Mach Number Values for Case Properties

3.3.1 Case 1 Results

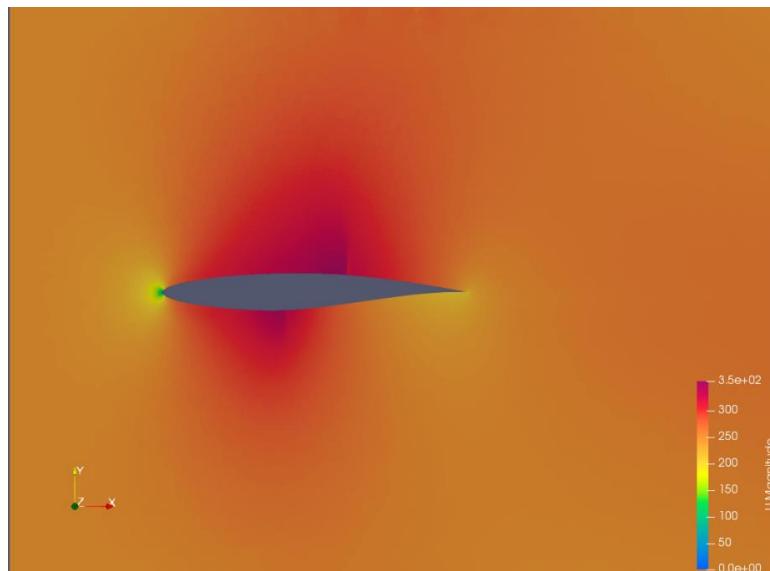


Figure 3.9. Case 1 Velocity(m/s)

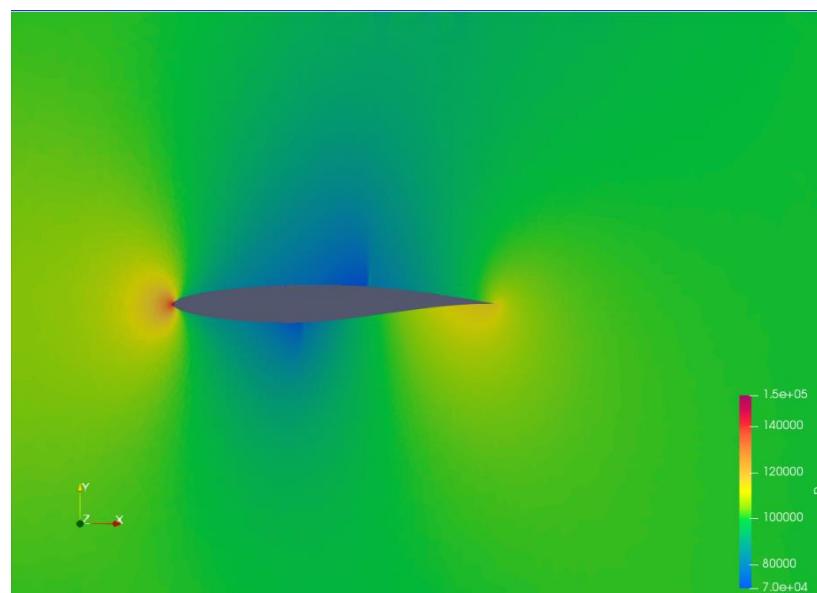


Figure 3.10. Case 1 Pressure (Pa)

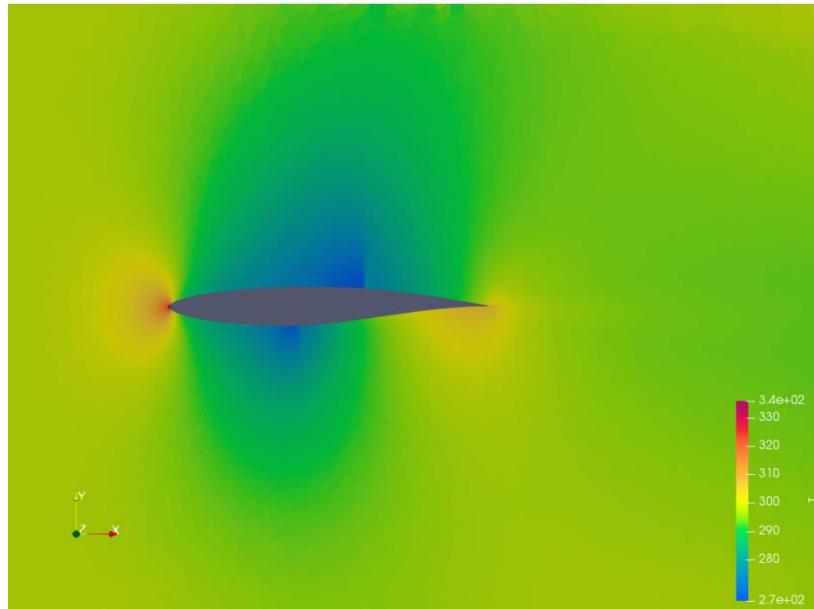


Figure 3.11. Case 1 Temperature (°K)

Due to its Mach number this airfoil is in transonic speeds. And a result of that the shockwave starts on the airfoil not at the front side of the wing. In these figures, we can clearly see the shockwave on the airfoil where sudden pressure, temperature and velocity changes happens.

3.3.2 Case 2 Results

With the command that we added to the ControlDict file we can plot the coefficient of drag (C_d) and coefficient of lift (C_l) values. Due its zero degree attack angle, this airfoil creates no lift.

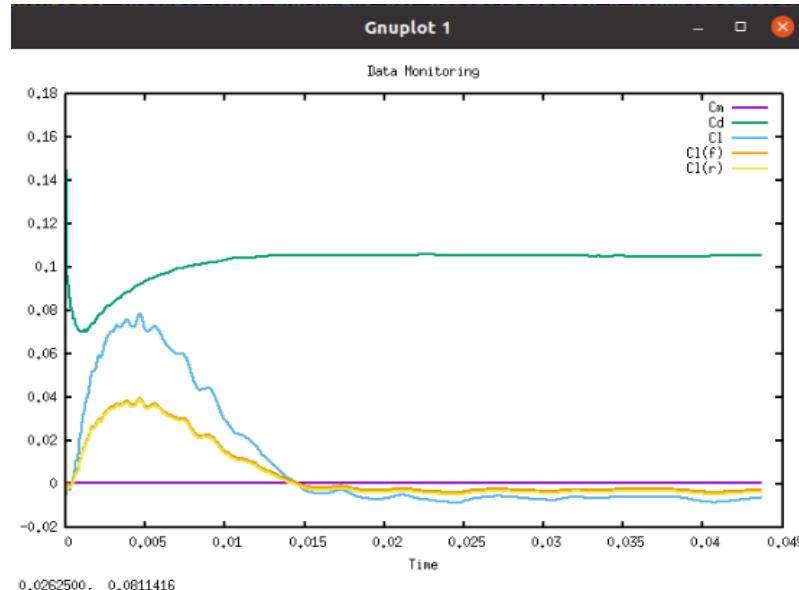


Figure 3.12. Case 2 Lift/Drag Coefficients

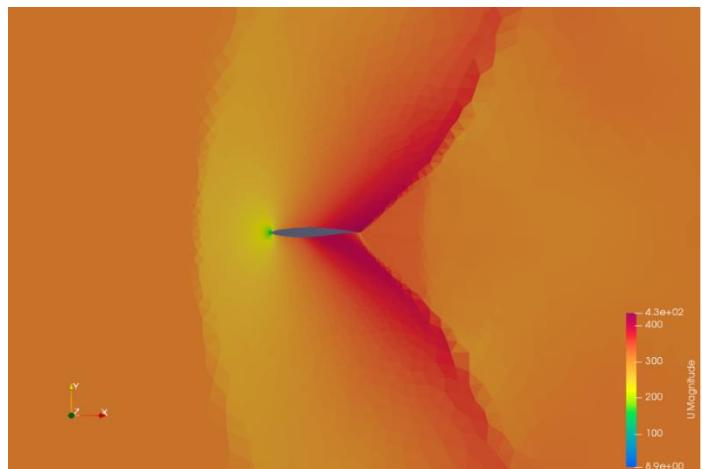


Figure 3.13. Case 2 Velocity(m/s)

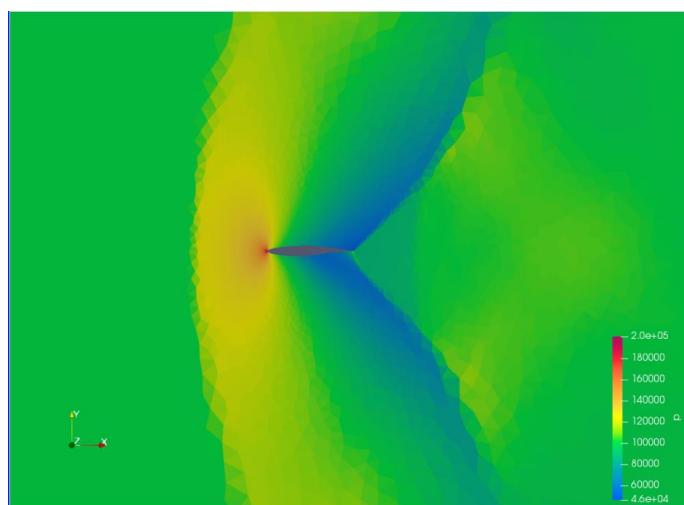


Figure 3.14. Case 2 Pressure (Pa)

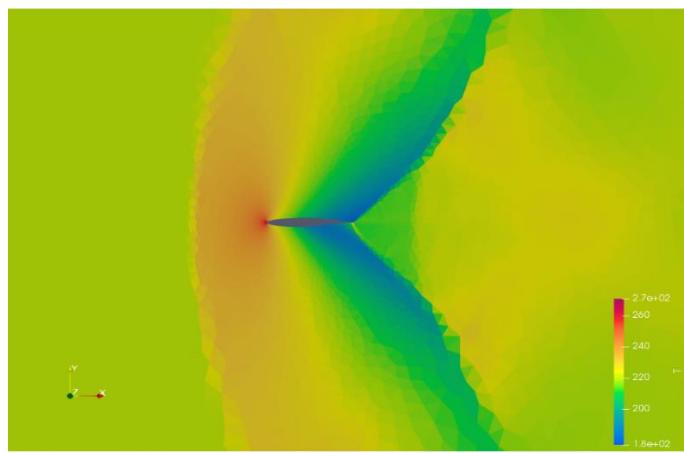


Figure 3.15. Case 2 Temperature (°K)

3.3.3 Case 3 Results

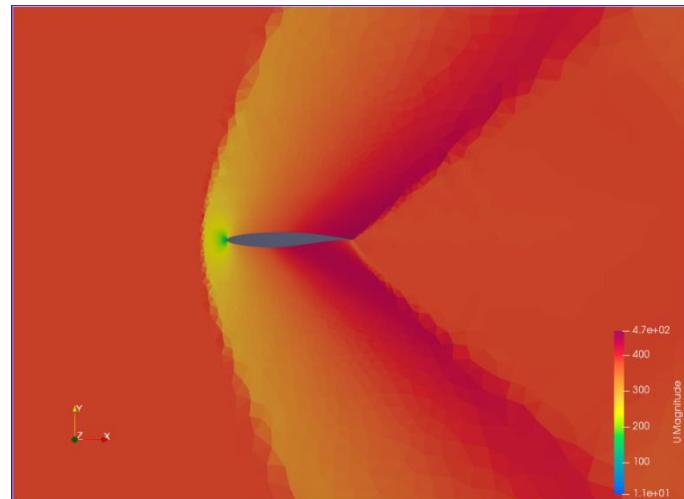


Figure 3.16. Case 3 Velocity(m/s)

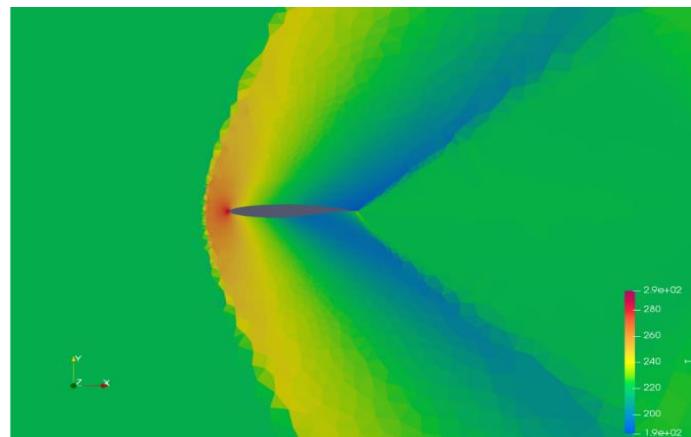


Figure 3.17. Case 3 Pressure (Pa)

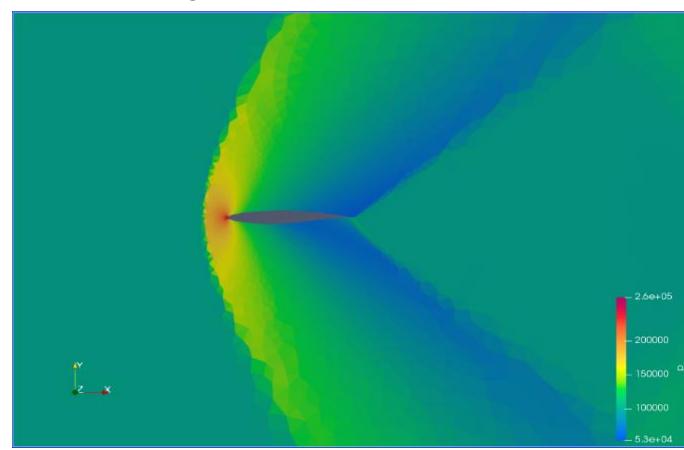


Figure 3.18. Case 3 Temperature (°K)

3.3.4 Case 4 Results

For this case, we have given an attack angle of 2 degrees to see lift creation of the wing. Therefore, initial velocity along the x axis and y axis are:

- $U_x = 445.618 \text{ m/s}$
- $U_y = 15.613 \text{ m/s}$

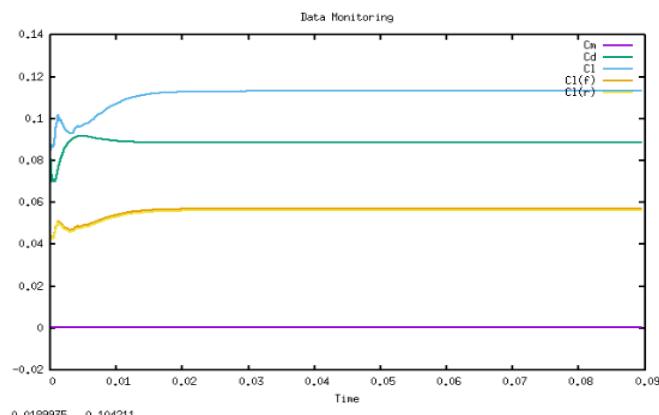


Figure 3.19. Case 4 Lift/Drag Coefficients

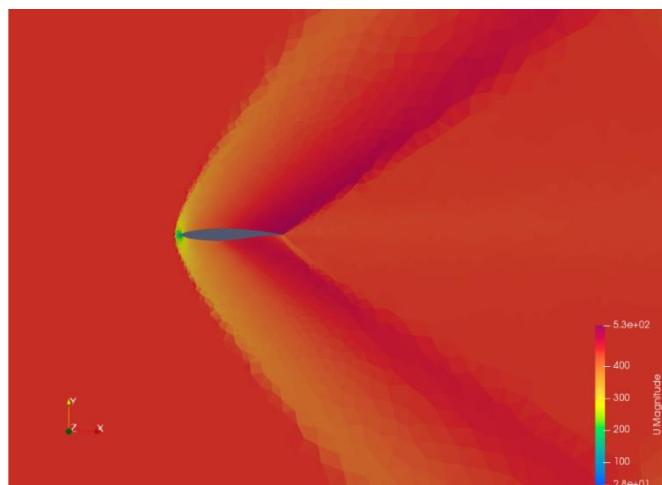


Figure 3.20. Case 4 Velocity(m/s)

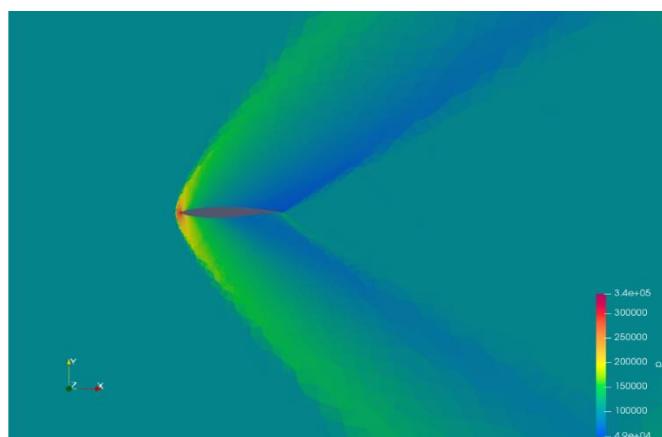


Figure 3.21. Case 4 Velocity(m/s)

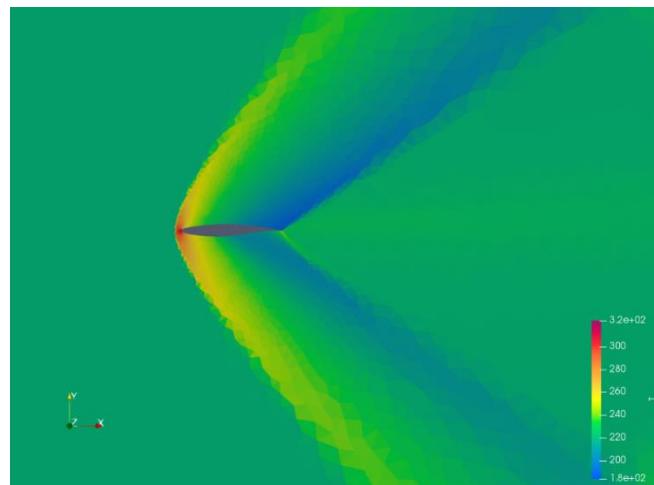


Figure 3.22. Case 4 Temperature (°K)

Table 3.3 RAE2822 NASA Experimental Conditions

Mach Number	Static Pressure (Pa)	Temperature (K)	Angle-of-Attack (Deg)
0.729	108987.7725	255.5556	2.31

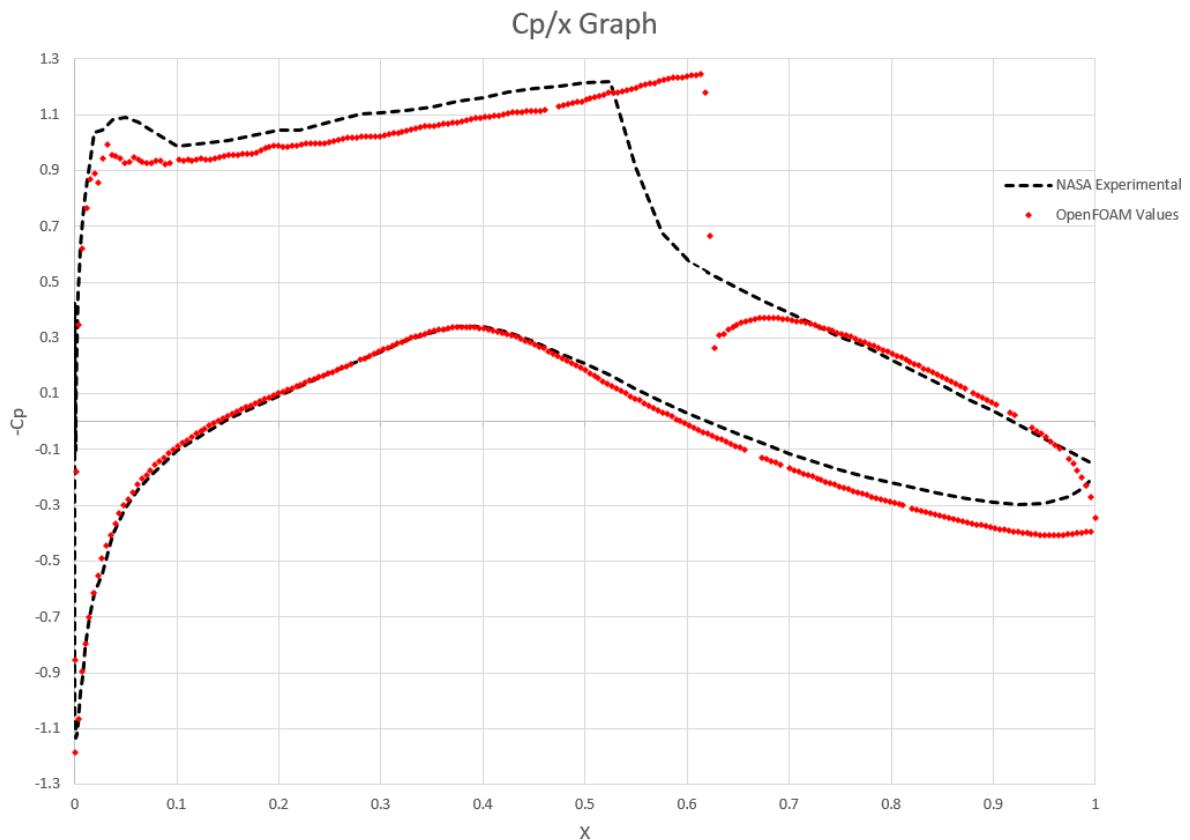


Figure 3.23. -Cp/x Graph of RAE2822 Result Comparison with NASA Experimental Values

3.4 ONERA M6 Results

3.4.1 RhoSimpleFoam Results

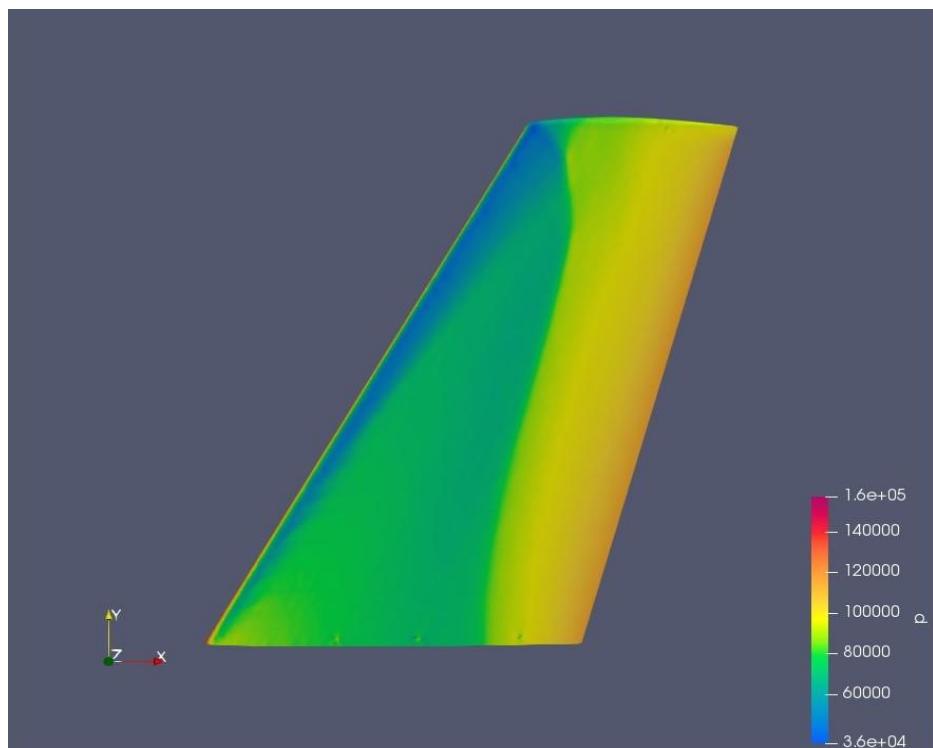


Figure 3.24 Onera M6 rhoSimpleFoam Pressure Distribution

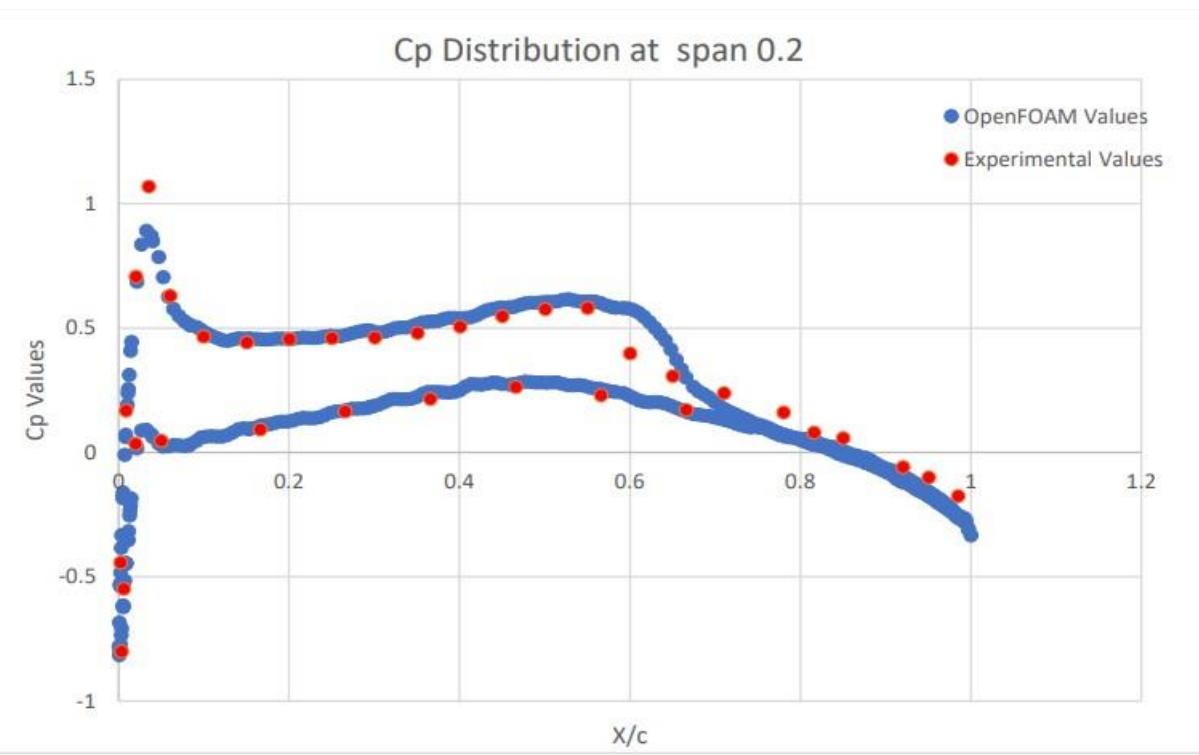


Figure 3.25 rhoSimpleFoam Cp Distribution at 0.2

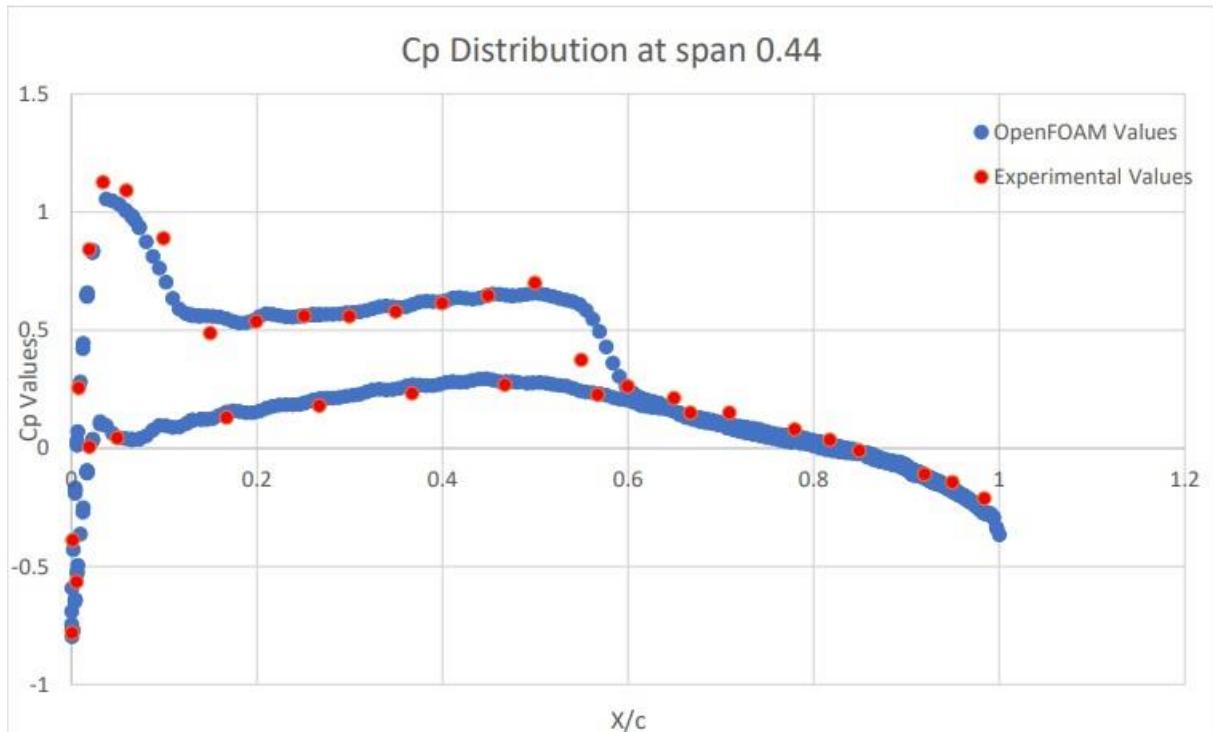


Figure 3.26 rhoSimpleFoam Cp Distribution at 0.44

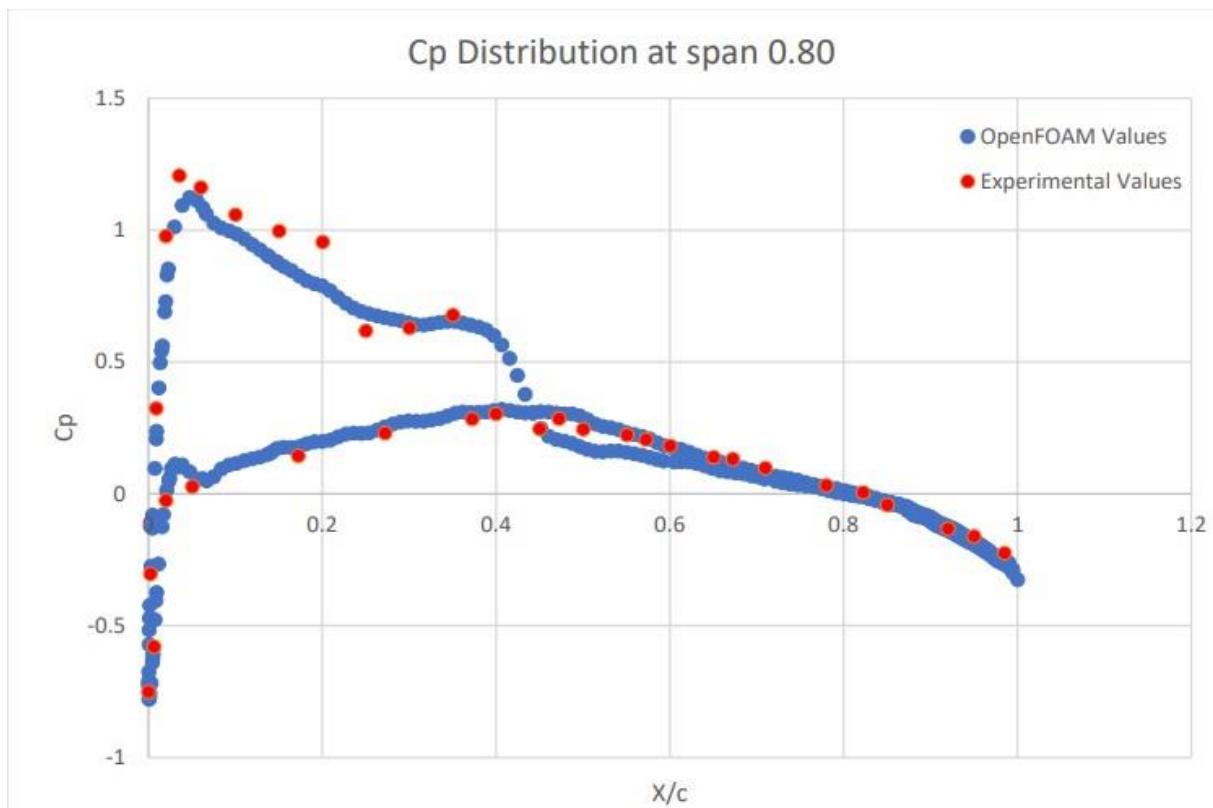


Figure 3.27 rhoSimpleFoam Cp Distribution at 0.80

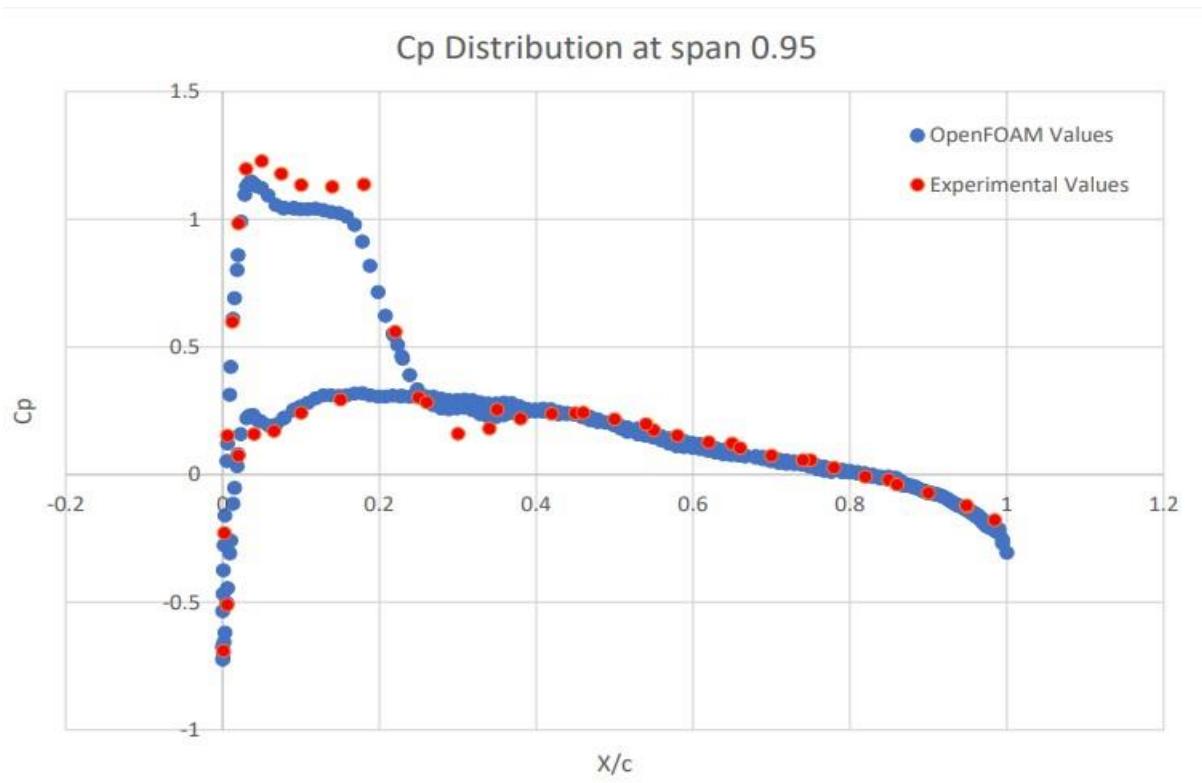


Figure 3.28 rhoSimpleFoam Cp Distribution at 0.95

3.4.2 RhoPimpleFoam Results

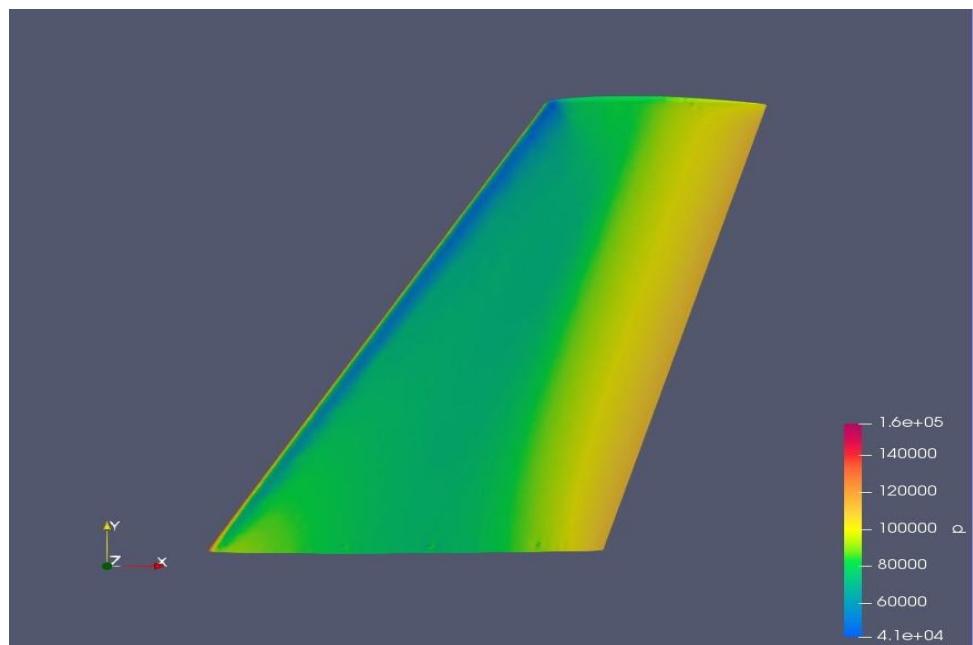


Figure 3.29 Onera M6 rhoPimpleFoam Pressure Distribution

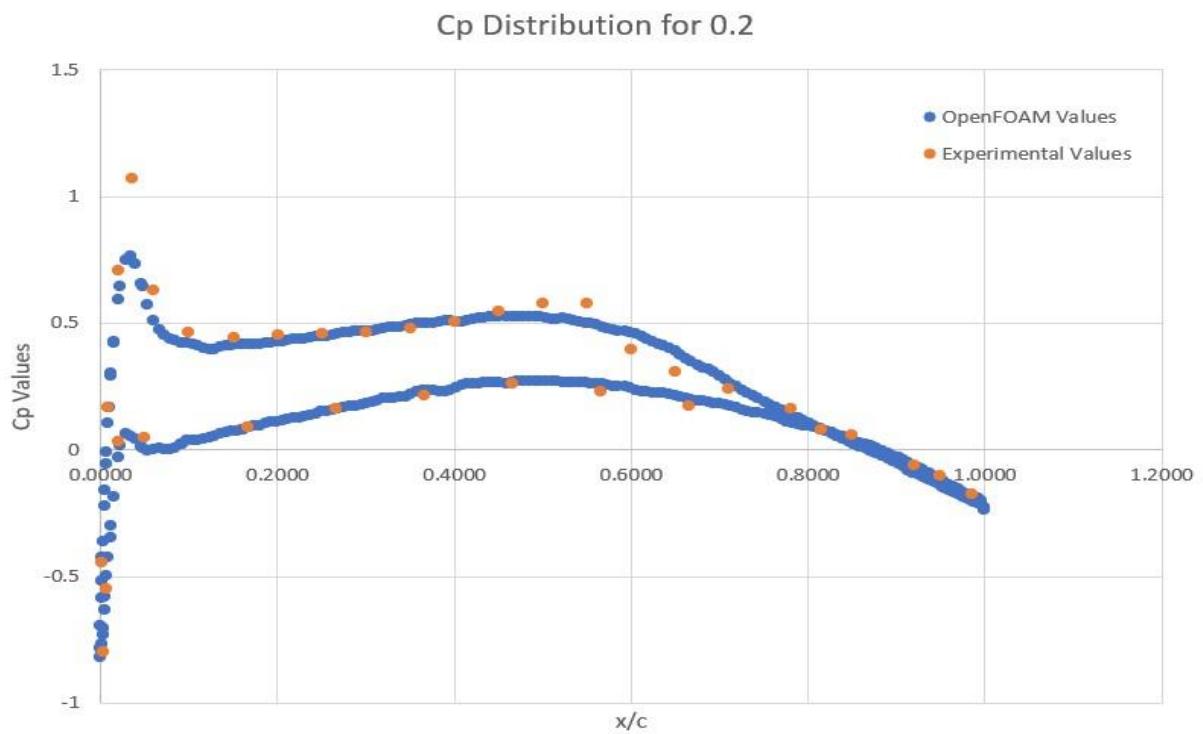


Figure 3.30 rhoPimpleFoam Cp Distribution at 0.2

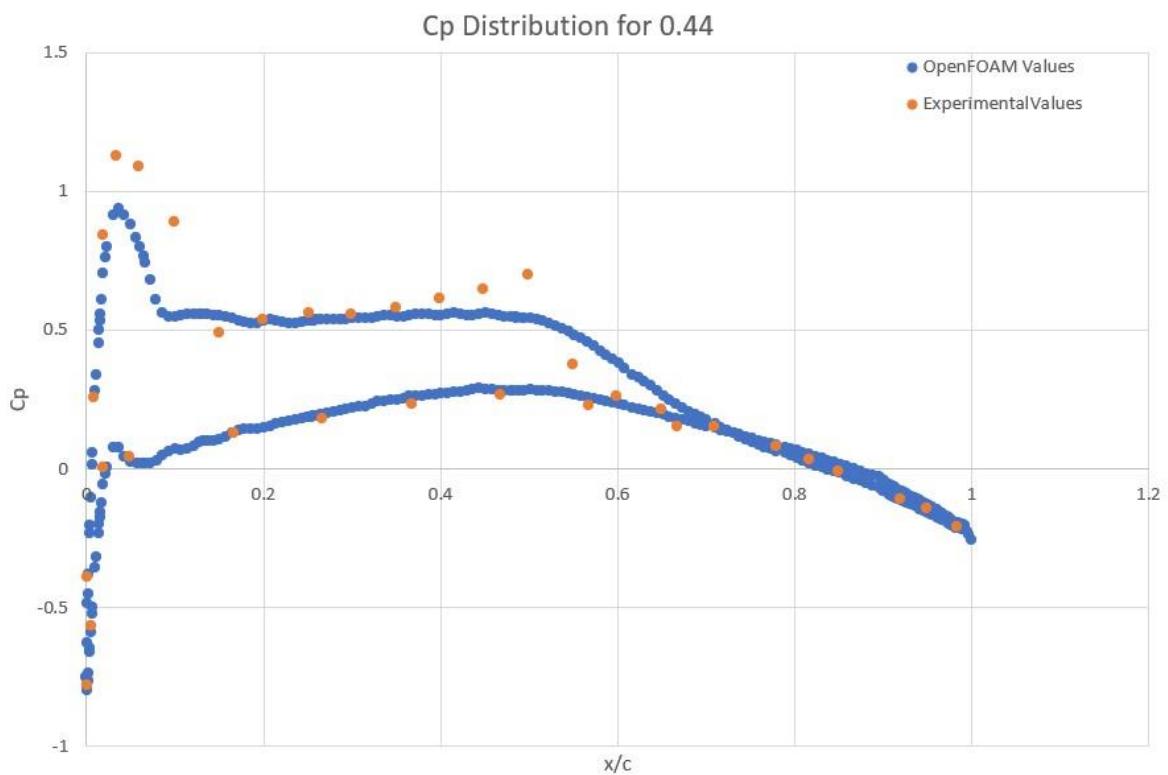


Figure 3.31 rhoPimpleFoam Cp Distribution at 0.44

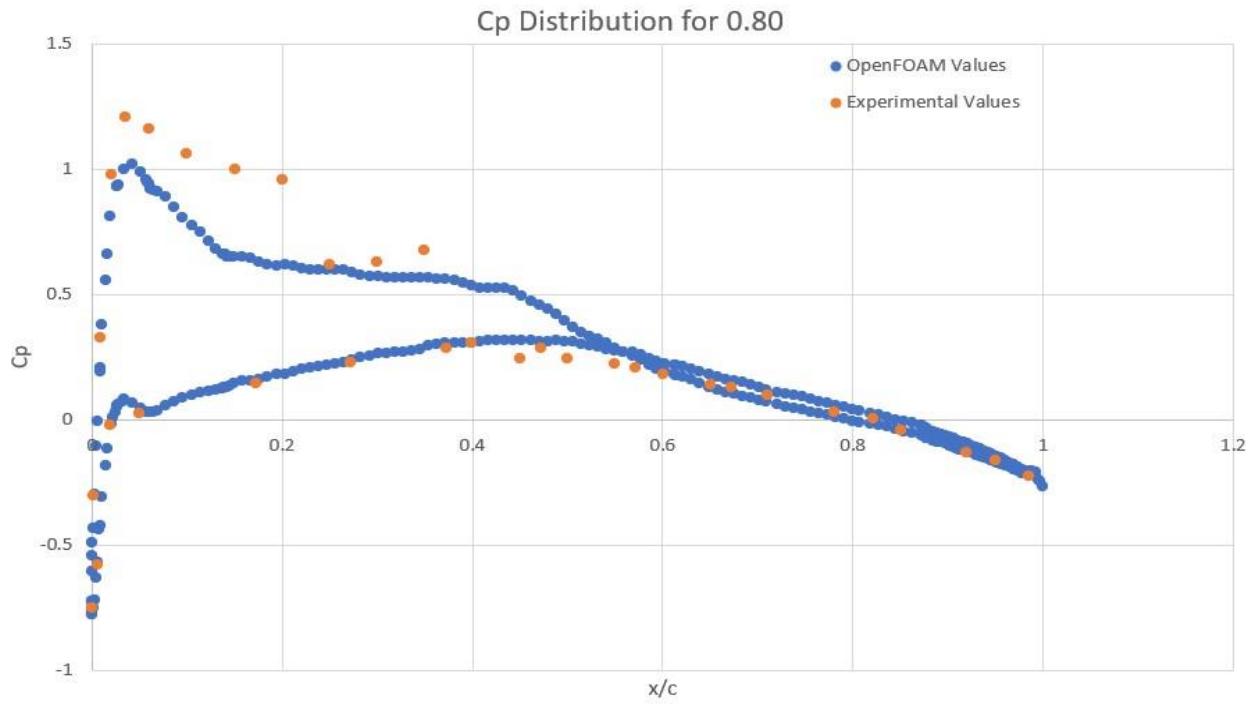


Figure 3.32 rhoPimpleFoam Cp Distribution at 0.80

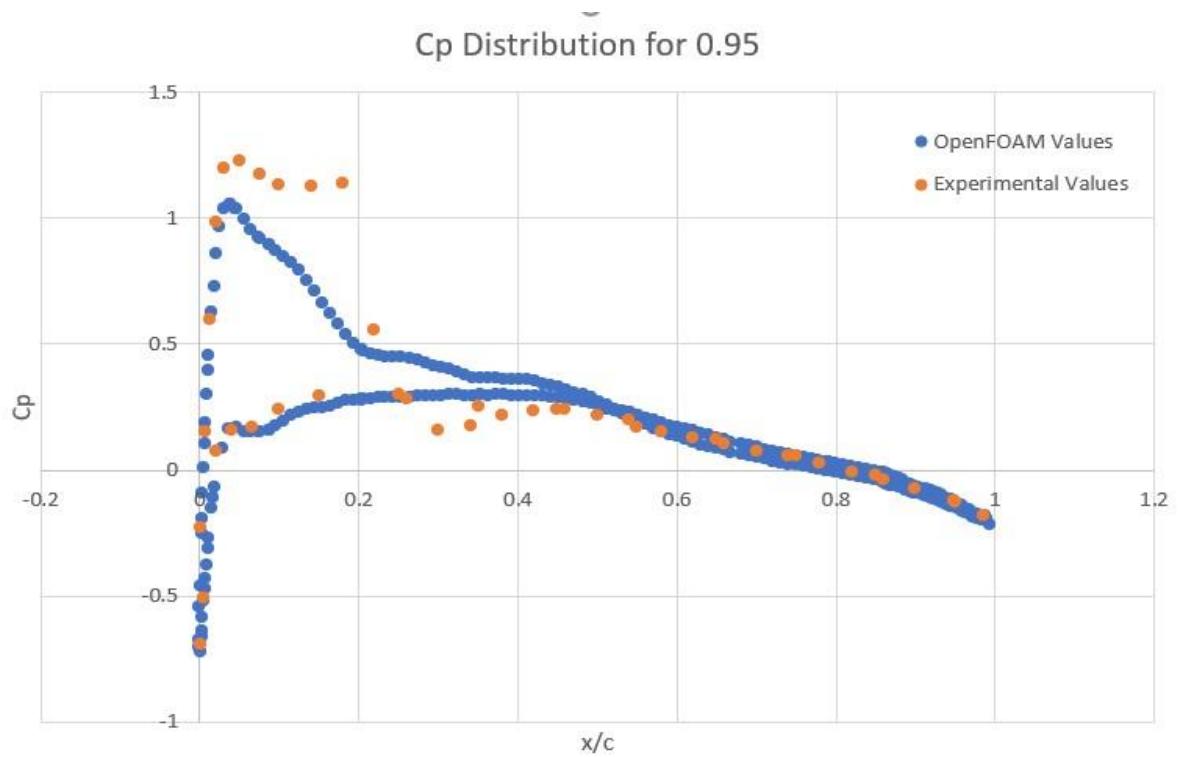


Figure 3.33 rhoPimpleFoam Cp Distribution at 0.95

3.4.3 RhoCentralFoam Results

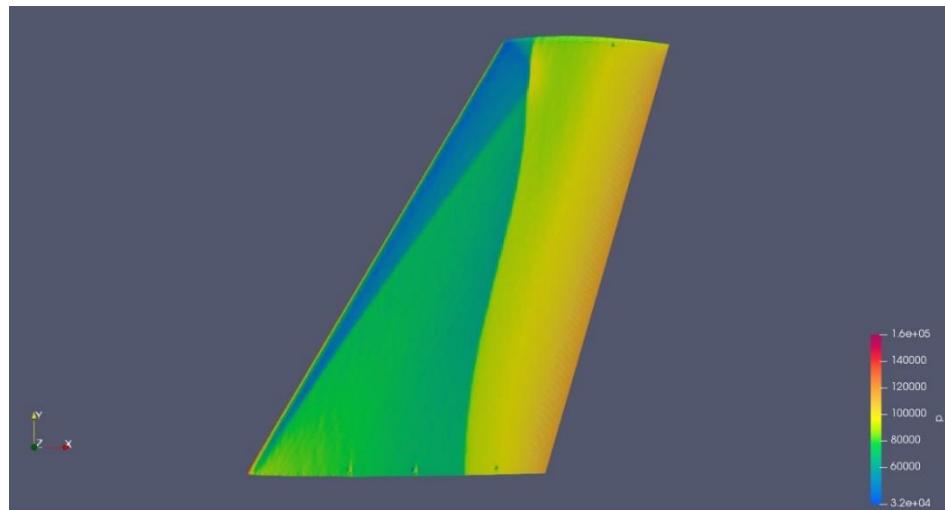


Figure 3.34 Onera M6 rhoCentralFoam Pressure Distribution

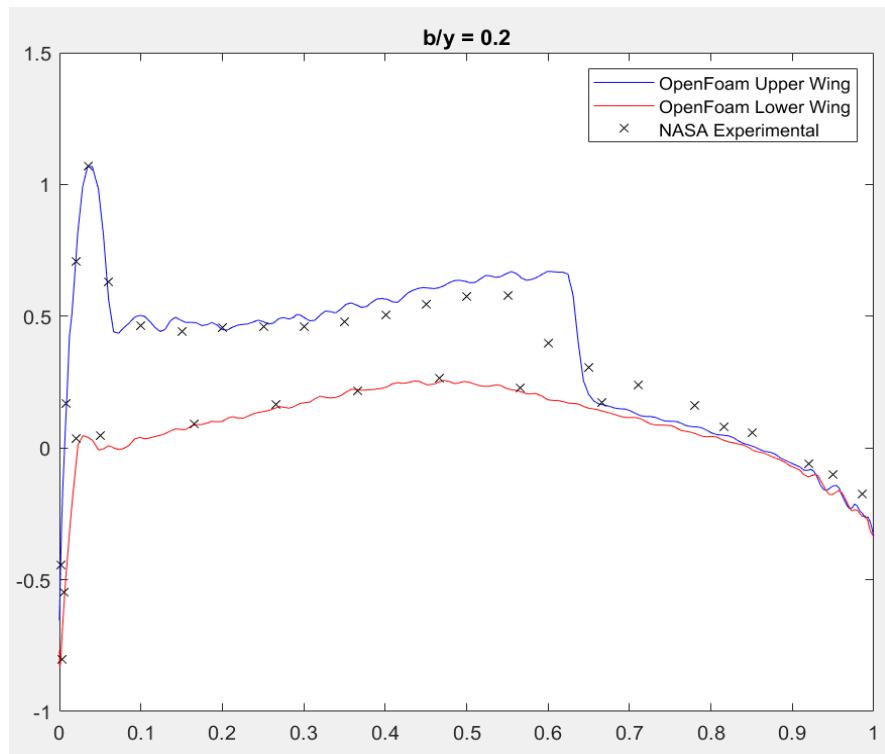


Figure 3.35 rhoCentralFoam C_p Distribution at 0.2

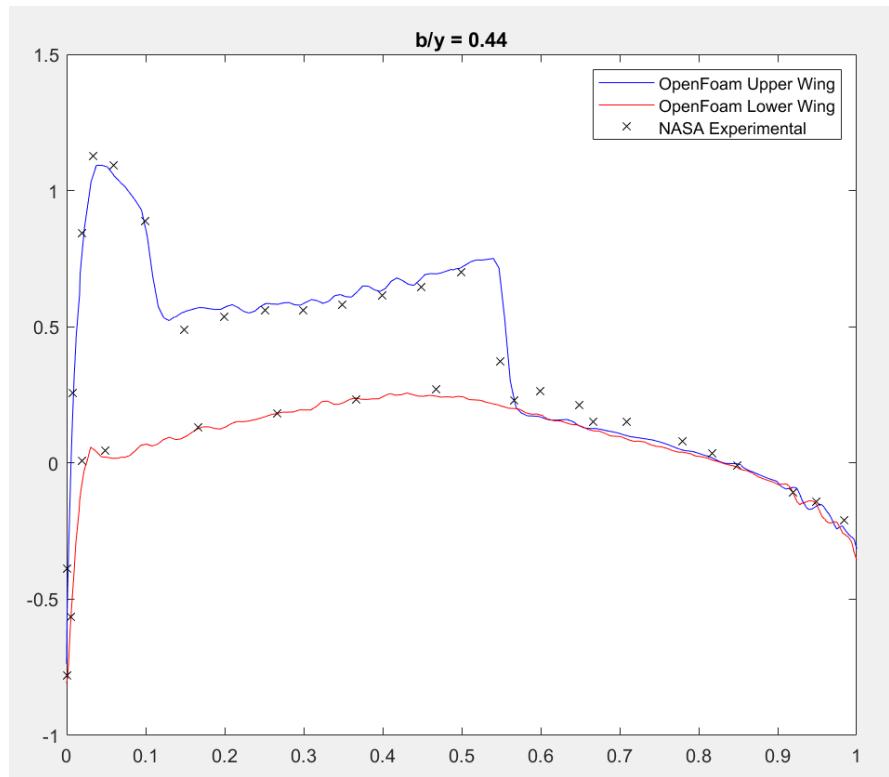


Figure 3.36 rhoCentralFoam Cp Distribution at 0.44

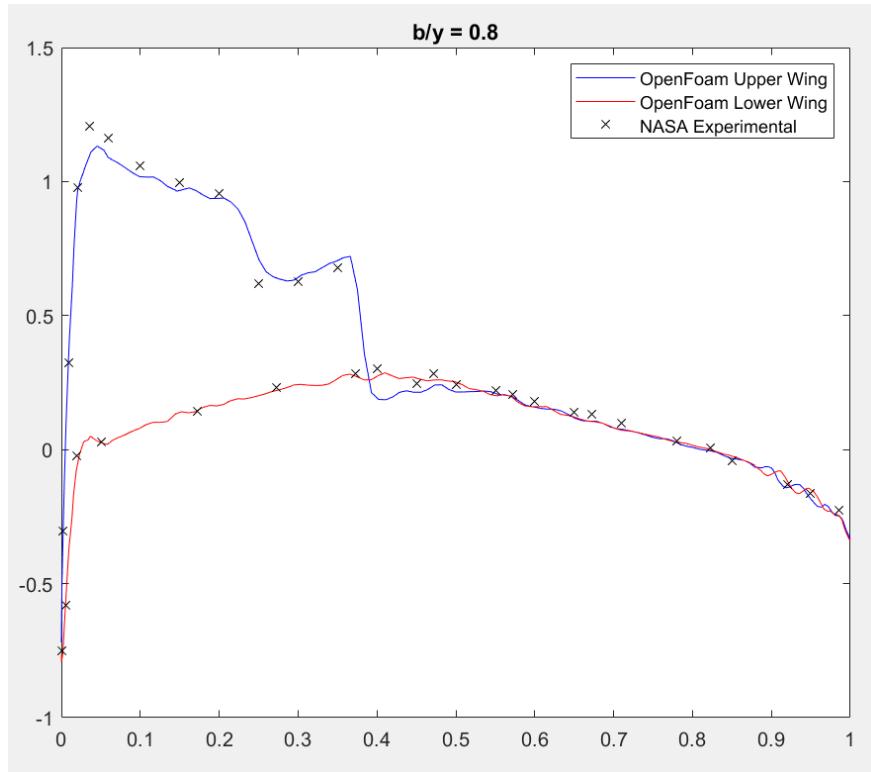


Figure 3.37 rhoCentralFoam Cp Distribution at 0.8

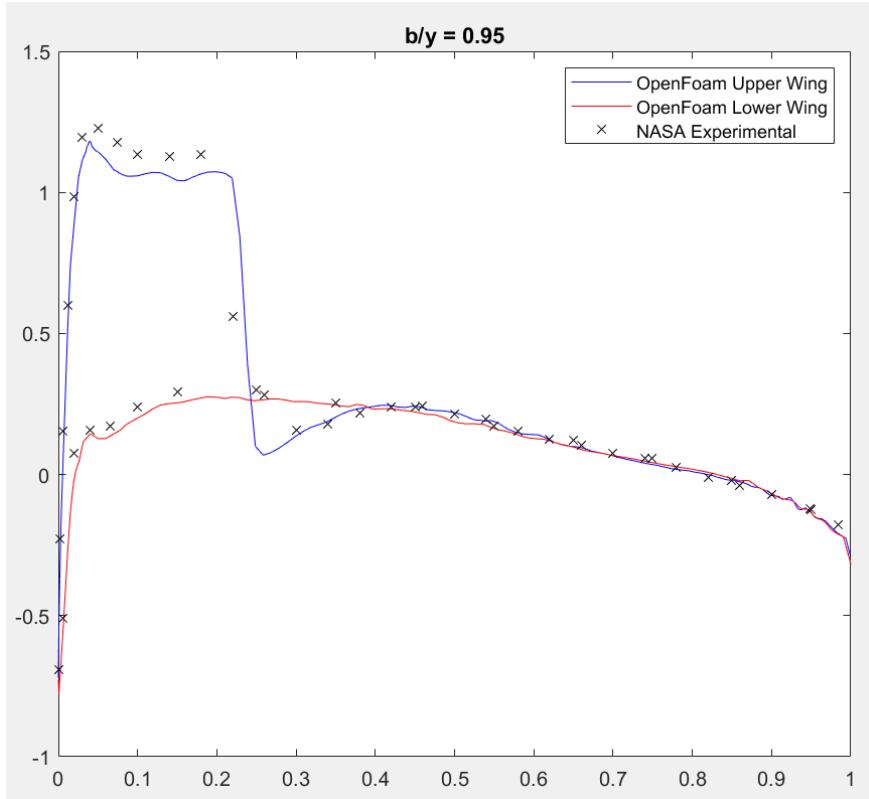


Figure 3.38 rhoCentralFoam Cp Distribution at 0.95

3.5 Consideration of Constraints

We see that rhoCentralFoam is the best solvers as the results wise but this computation took 80+ hours with Intel i5 7400K chipset. RhoCentralFoam is a transient state solver so, if we had a time constraint we could use rhoSimpleFoam or maybe rhoPimpleFoam.

OpenFOAM is an open source software so if any CFD project has a budget cap it is great choice. And it is equal and even better than commercial softwares for different cases.

3.6 Improvement of Design

As the air flows through the normal shock wave it undergoes a rapid compression. The compression decreases the kinetic energy of the airstream and converts it into a pressure and temperature increase behind the shock wave. The heat rise behind the shock wave is either radiated to the atmosphere or absorbed by the wing surface, but in either case it is lost, and this lost energy must be continuously supplied by the engines. This energy loss represents a type of drag known as wave drag. Wave drag cause the entropy increase and the loss of efficiency. We want the losses that occur at the shock as small as possible. The key factor

is what is the mach number. If the mach number normal to the shock wave is small it will decrease the strength of the shock.

Shock wave causes vibration and this affects the structural safety. Therefore knowing the shock wave strength and location, we can make improvement on design of the wing.

4. CONCLUSION

In this project, our aim was validation of the OpenFOAM software and its solvers. In this study, we have gained experiences about OpenFOAM, OpenFOAM solvers, GMSH and CFD procedures. We have seen Mach number effects on the flows and also, we have compared the different solvers which are RhoPimpleFoam and RhoCentralFoam on NACA0012 airfoil. After that we have decided to our solver as RhoCentralFoam. Then, we have done RAE2822 airfoil analysis in RhoCentralFoam. To get familiar and learn OpenFOAM, we have done these analysis's as 2-dimensional.

Then, we compared our cases with NASA's experimental values of RAE2822. After plotting the graph, we can see that our results are similar to real-life experimental results. There are some errors due to solver type, mesh quality and differences between experiments and numerical results.

After that we moved on the three dimensional geometries. At this part our wing is ONERA M6. By the help of OpenFOAM standard rho solvers (rhoSimpleFoam, rhoPimpleFoam and rhoCentralFoam), we have computed our simulations and compared them with the NASA'S experimental values.

At the end, after the comparison we saw that rhoPimpleFoam's lack of capturing sudden changes. Also, it shows low performance when it is getting close to the wing tip. For the rhoSimpleFoam, it has better performance except capturing the first shock wave and Cp values are generally smaller than the experimental values. At last, we get our best results with rhoCentralFoam. It showed great performance both capturing shock waves and sudden changes. But its execution time was the greatest.

REFERENCES

1. Cimbala J. , Çengel Y. (2006), Fluid Mechanics Fundamentals and Applications 1. Edition, McGrawhill-Companies, New York, America
2. Moukalled F. , Mangani L. , Darwish M. (2016), The Finite Volume Method in Computational Fluid Dynamics, Vol. 113, Springer International, Berlin, Germany
3. Z. DOĞAN, (2008) Ses Altı Hızlarda Kanat Profili Etrafında Akışın İncelenmesi, Yüksek Lisans Tezi, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, Kayseri, Türkiye
4. M. WINTER, (2013) Benchmark and Validation Of Open Source CFD Codes, With Focus On Compressible And Rotating Capabilities, For Integration On The Simscale Platform, Master's Thesis, Department of Applied Mechanics Chalmers University of Technology, Gothenburg, Sweden
5. K. Harish Kumar, CH. Kiran Kumar, T. Naveen Kumar, [CFD Analysis of RAE2822 Supercritical Airfoil at Transonic Mach Speed](#) (2015)
6. [Computational fluid dynamics - Wikipedia](#)
7. Cüneyt Sert, Finite Element Analysis in Thermofluids, [Governing Equations of Fluid Flow and Heat Transfer](#)
8. Brian J. Cantwell, [Fundamentals of Compressible Flow Course Reader](#), Department of Aeronautics and Astronautics Stanford University, Stanford, California
9. Christopher J. Greenshields, [OpenFOAM User Guide Version 8](#) (2020)
10. [OneraM6 by Michael Alletto - OpenFOAM Wiki](#)

APPENDICES

For the *system/fvSchemes* the following file specified in RAE2822 case.

```
fluxScheme    Kurganov;  
  
ddtSchemes  
{  
    default    Euler;  
}  
  
gradSchemes  
{  
    default    Gauss linear;
```

```

}

divSchemes
{
    default      none;
    div(tauMC)   Gauss linear;
    div(phi,omega) Gauss upwind;
    div(phi,k)   Gauss upwind;
}

laplacianSchemes
{
    default      Gauss linear corrected;
}

interpolationSchemes
{
    default      linear;
    reconstruct(rho) vanAlbada;
    reconstruct(U) vanAlbadaV;
    reconstruct(T) vanAlbada;
}

snGradSchemes
{
    default      corrected;
}

wallDist
{
    method meshWave;
}

```

For the *system/fvSolution* the following file specified in RAE2822 case.

```

solvers
{
    p
    {
        solver      GAMG;
        smoother    GaussSeidel;
        tolerance   1e-6;
        relTol     0.01;
    }

    pFinal
    {
        $p;
        relTol     0;
    }
}

```

```

}

"(rho|U|k|omega|e)"
{
    solver      PBiCGStab;
    preconditioner diagonal;
    tolerance   1e-6;
    relTol      0.1;
}

"(rho|rhoU|rhoE)"
{
    solver      diagonal;
}

"(rho|U|k|omega|e)Final"
{
    $U;
    relTol      0;
}
}

SIMPLE
{
    residualControl
    {
        p          1e-4;
        U          1e-4;
        "(k|omega|e)" 1e-4;
    }

    nNonOrthogonalCorrectors 0;
    pMinFactor   0.1;
    pMaxFactor   2;
}

PIMPLE
{
    nCorrectors      2;
    nNonOrthogonalCorrectors 1;
    nOuterCorrectors 1;
    pMinFactor     0.1;
    pMaxFactor     2;
}

relaxationFactors
{
    equations
    {
        ".*"    1;
    }
}

```

And at last for the *system/controlDict* the following file specified in RAE2822 case.

```

/*-----* C++ -----*/
=====
 \ / F ield      | OpenFOAM: The Open Source CFD Toolbox
 \ / O peration   | Website: https://openfoam.org
 \ / A nd        | Version: 8
 \W M anipulation |
/*-----*/
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location "system";
    object controlDict;
}
// ****
application rhoCentralFoam;

startFrom latestTime;

startTime 0;

stopAt endTime;

endTime 0.8;

deltaT 8e-6;

writeControl adjustableRunTime;

writeInterval 0.005;

cycleWrite 0;

writeFormat ascii;

writePrecision 6;

writeCompression off;

timeFormat general;

timePrecision 6;

runTimeModifiable true;

adjustTimeStep yes;

maxCo 0.2;

maxDeltaT 1;

functions
{

```

```

#includeFunc MachNo
#includeFunc residuals

forces1
{
    type forces;
    libs ("libforces.so");
    patches (wall);
    p p;
    U U;
    rho rho;
    pref 0;
    porosity no;
    writeFields yes;
    CofR (0 0 0);
    liftDir (0 1 0);
    dragDir (1 0 0);
    magUInf 233.55702;
}

forceCoeffs1
{
    // Mandatory entries
    type      forceCoeffs;
    libs     ("libforces.so");
    patches   (wall);

    // Optional entries

    // Field names
    p          p;
    U          U;
    rhoInf    rhoInf;

    // Reference pressure [Pa]
    pRef      101325;

    // Include porosity effects?
    porosity   no;

    // Store and write volume field representations of forces and moments
    writeFields yes;

    // Centre of rotation for moment calculations
    CofR      (0 0 0);

    // Lift direction
    liftDir   (0 1 0);

    // Drag direction
    dragDir   (1 0 0);
}

```

```
// Pitch axis
pitchAxis    (0 1 0);

// Freestream velocity magnitude [m/s]
magUInf     233.155702;
rhoInf      1.381;

// Reference length [m]
lRef        1;

// Reference area [m2]
Aref        0;
```

