A Numerical Simulation of Flow over a NACA0025 Airfoil using Large Eddy Simulation Turbulence Models

Babak Babaee Owlam

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Mechanical Engineering) at
Concordia University
Montreal, Quebec, Canada

January 2008

Your file Votre référence
ISBN: 978-0-494-40908-4
Our file Notre référence
ISBN: 978-0-494-40908-4

ABSTARCT

A NUMERICAL SIMULATION OF FLOW OVER A NACA0025 AIRFOIL USING LARGE EDDY SIMULATION TURBULENCE MODELS

Babak Babaee Owlam

The ability of computational fluid dynamic (CFD) to predict critical flow characteristics has always been questionable. Flow separation over lifting surfaces such as airfoils are one of the critical features which can significantly deteriorate their aerodynamic performances. The purpose of this research is to evaluate the accuracy of two CFD methods in predicting the flow separation over a NACA0025 airfoil at low Reynolds numbers. The first code is an in-house code which is based on a 3D compressible Navier-Stokes solver with preconditioning and self-adaptive upwinding methods. The second code is the commercial FLUENT software. In order to accurately simulate the laminar boundary layer separation, the Large Eddy Simulation (LES) method is used for turbulence modeling of both codes. Results comparison show that Fluent is not able to capture this feature. In addition the results are also compared with another similar numerical simulation and validated with available experimental data.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Dr. Marius Paraschivoiu for all his support, patience and valuable advices during my studies; many thanks to my colleagues and friends in the CFD lab especially Kaveh Mohamed and Mohammad Reza Kameshki for their very important suggestions throughout this project.

Finally, words fail me to express my appreciation from my father, my mother and my sister who each deserve special mention for their inseparable support and prayers. I would like to dedicate this thesis to my family and my love Maral.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

c	chord length of an airfoil
c_p	Constant pressure specific heat capacity
c_v	Constant volume specific heat capacity
C_s	Smagorinsky constant
e_I	Internal specific energy
e	Total specific energy
${}^{c}F$	Convective flux
${}^{v}F$	Diffusive flux
H	Specific total enthalpy
h	Total specific enthalpy
k	Thermal conductivity constant
m	meter
M	Mach number
p	Pressure
Pr	Prandtl number
Q	Conservative variables
R_g	Global ideal gas constant
Re	Reynolds number
Re_c	Chord Reynolds number
s	second
S	Strain tensor
t	Time
Δt	Time step
u	X-component of velocity

u_τ	Friction velocity
u_∞	Free stream velocity
v	Y-component of velocity
w	Z-component of velocity
x	X-direction of Cartesian coordinate
y	Y-direction of Cartesian coordinate
y^+	Non-dimensional distance from the wall
z	Z-direction of Cartesian coordinate
α	Angle of attack
ρ	Density
$\vec{\nabla}$	Gradient vector operator
δ_{ij}	Kronecker delta
∂_j	Partial derivative operator
μ	Dynamic viscosity coefficient
ψ	Test function
ν	Kinematic viscosity
Δ	Filter width
Δx	X-direction mesh size
Δy	Y-direction mesh size
Δz	Z-direction mesh size

ABBREVIATIONS

CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy
DNS	Direct Numerical Simulation
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
LES	Large Eddy Simulations
LSP	Laminar separation Bubble
MUSCL	Monotone Upwind Scheme for Conservation Laws
RANS	Reynolds Averaged Navier-Stokes
SGS	Sub-Grid Scale

1 Introduction

1.1 CFD Review

The capabilities of Computational Fluid Dynamics (CFD) are expanding rapidly as computer power increases. CFD is now used in partnership with experimental methods to address very complex engineering problems. Numerical simulations and experiments have each their strengths and limitations. On one hand, CFD can provide a very detailed view of the flow field, generating velocities, pressures and densities at every point in the field- something that would be very expensive to measure experimentally. However, calculations always approximate the flow in some way, either by solving a simplified equation or by introducing approximations through the numerical method itself. On the other hand, the wind tunnel test has the advantage of dealing with a real fluid and measuring the correct physics, though usually not at perfect real conditions (Reynolds number differences) or the right geometry (because of model support interference or wall effects). It often provides good measures of integrated flow properties such as total forces and moments acting on a body. It is best suited for validation and database building within acceptable limits of a development program's cost and schedule [1].

Since 1960s CFD has been used in design and R&D applications in aerospace industry [2]. The improvement in computation facilities including processing time and storage capacity is accelerating the growth in the use of CFD for industrial projects. In 2008, we can run models requiring three order of magnitudes greater computational effort comparing to those that were used in 1995. Today's typical models consist of one to five million cells and some models have tens of millions of cells. With the current rate of growth in computer resources we can expect billion-cell models within a few years [3].

The application of CFD today has revolutionized the process of aerodynamic design. The development of a fully unstructured mesh is a great advancement towards better CFD results. This type of mesh allows the CFD user to have a finer grid resolution for complex geometry details during the meshing generation process.

1.2 Turbulence models

Turbulence modeling in CFD is essential due to the presence of turbulenc in many engineering problems. Turbulent flows are characterized by eddies with a wide range of length and time scales. The largest eddies are typically comparable in size to the characteristic length of the mean flow. The smallest scales are responsible for the dissipation of turbulence kinetic energy.

It is possible, in theory, to directly resolve all bands of scales of motion from the largest scale to the smallest viscous dissipation scale using an

approach known as direct numerical simulation (DNS). No modeling is required in DNS. The Navier-Stokes equations can be solved directly without any filtration or averaging. However, DNS is not feasible for practical engineering problems involving high Reynolds number flows due to the high computational cost. DNS can only be done for very low Reynolds numbers and simple geometries with huge computing power.

Reynolds Averaged Navier-Stokes (RANS) models are the most commonly used turbulence model but they model all the scales of the flow in a time-averaged sense. This approximation causes some deficiency in the cases where the flows are time-dependant. RANS are not very accurate for unsteady flows and complex geometries simulation [4].

Large Eddy Simulations (LES) is the method that resolve large eddies directly, while small eddies are modeled. LES thus falls between DNS and RANS in terms of the fraction of the resolved scales. In LES only the small scaled fluctuations are modeled and the larger scaled fluctuations are computed directly [5].

Large eddies are more problem-dependent. Their behavior is dictated by the geometries and boundary conditions of the flow involved. Small eddies are less dependent on the geometry, tend to be more isotropic, and are consequently more universal. Resolving only the large eddies allows one to use much coarser mesh and larger times-step sizes in LES than in DNS. However, LES still requires substantially finer meshes than those typically used for RANS calculations. In addition, LES has to be run for a sufficiently long flow-time to

obtain stable statistics of the flow being modeled. As a result, the computational cost involved with LES is normally orders of magnitudes higher than that for steady RANS calculations in terms of memory (RAM) and CPU time. Therefore, high-performance computing (e.g., parallel computing) is a necessity for LES, especially for industrial applications.

There are many problems that RANS methods are not able to resolve the flow details accurately. However, LES has proved that it is better capable of modeling the flow than RANS in those flows conditions. In general simulations done by RANS have not obtained satisfactory results for unsteady flows and flows in complex geometries. Therefore problems including separated flows, vortex-boundary interactions, and three dimensional boundary layer flows should be modeled by LES rather than RANS.

1.3 Low Reynolds number flows and laminar separation

Flows in the range of Reynolds numbers between 10,000 to 1,000,000 are usually called Low-Reynolds number flows [6]. Wind turbines, turbo machinery, micro air vehicles, hydrofoils, and low-speed/high-altitude aircraft are some of the engineering systems dealing with airfoil operation at low Reynolds numbers and laminar boundary layers. Recall that a laminar boundary layer leads a lower skin friction due to the characteristic velocity profile of laminar flow. However, its low skin friction in the laminar boundary layer also causes it to be badly affected by adverse pressure gradients. As the pressure begins to recover over the rear part

of the airfoil, the flow in the neighborhood of a solid wall becomes reversed and a laminar boundary layer will tend to separate from the surface.

According to linear stability theory, the transition onset is associated with small-amplitude wave disturbances growth. These disturbances start to interact with each other and cause the transition begins [7]. Kelvin-Helmholtz instability mechanism determines when the disturbances reach sufficient amplitude, the shear layer roll up occurs and forming the free shear layer that amplify the incoming disturbances. This phenomenon finally leads to boundary layer separation [8].

When the cord Reynolds number ($\mathrm{Re}_c = CU_0/\nu$ where U_0 is the free-stream velocity, C is the cord length and ν is the kinematic viscosity) decreases below about 500,000 [9] the laminar boundary layer on the upper surface of the airfoil becomes subjected to an adverse pressure gradient even at low angles of attack, often resulting in laminar boundary-layer separation and formation of a shear layer (Figure 1.1).

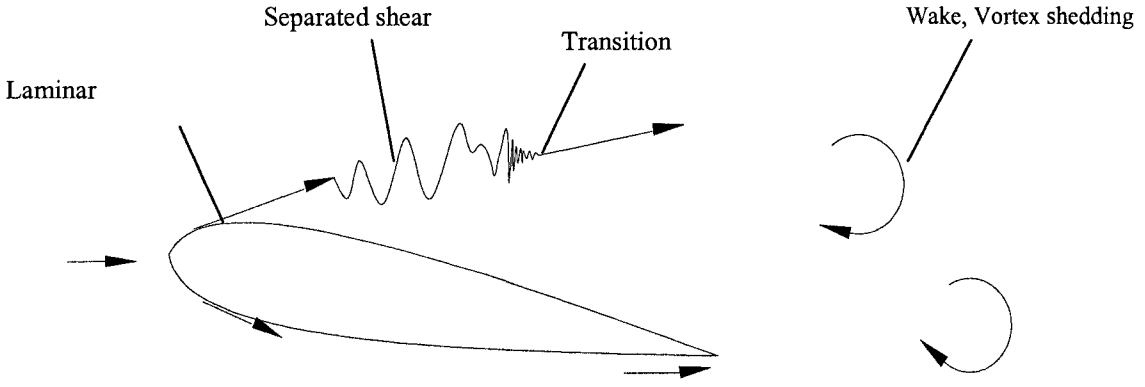


Figure 1.1: Laminar boundary separation without reattachment

This phenomenon was also shown by Mueller & DeLaurier (2003) that in Reynolds numbers of about 50,000, the shear layer separated over the airfoil surface and a large wake is formed. By increasing the Reynolds number, separated shear layer may reattach to the airfoil surface. In both cases, within the bubble a small region of constant pressure exists and the airfoil performance would be deteriorated effecting airfoil drag and lift due to the laminar boundary layer separation [10]. Klanfer (1953) classified the separation bubbles in two categories, small and long bubbles. A Laminar Separation Bubble (LSB) may cover only a small portion of cord of the airfoil. This small bubble is preferable because it does not affect lift and drag very much. On the other hand a long bubble which has a large surface of the airfoil covered may lead to high drag and even stall in particular if there is no reattachment.

Therefore, accurate evaluations of the characteristics and the extent of the separated flow region are crucial for designing effective airfoils and assessing airfoil performance [11]. Such flow separation causes a large increase in the pressure drag and decrease in lift. In general an airfoil with shorter bubble has higher lift to drag ratio (C_L/C_D).

The first studies about LSB were experiments of the airfoil stall conducted by Ward in 1963. Then Horton (1967) showed the laminar boundary layer goes to transition and then turbulent boundary layer reattached to the surface. Some similar works are numerical studies of bubble formation by Watmuff [12] for the flat plate geometry which agrees well with experimental data. Lin et al. Showed

that [13] vortices in the separation bubble are dependent of the surface pressure distribution on the airfoil. Pressure distribution varies by Reynolds number, surface curvature and surface roughness.

Yang et al. [14] showed that LSB is placed in the location of transition to turbulence. Lutz et al. showed for Reynolds number $Re \approx 100000$ that it becomes more and more difficult to obtain a turbulent boundary layer without extended laminar separation which may cause significant additional drag [15].

In 2006, NACA0025 airfoil boundary layer development at low Reynolds number were investigated experimentally in wind tunnel by Yarusevych et al. [16] and Tao Xu [17] compared these results numerically in 2007. Simulation results obtained in this thesis will be compared with the above mentioned works.

1.4 Preconditioning

The CFD code used in this research is a compressible code although the application for the validations is for incompressible flow regime at very low Mach number. This may cause some problems in terms of solver convergence rate. In recent years various methods have been proposed to solve low Mach numbers flows with compressible codes [18]. Preconditioning methods are used for solving incompressible flow problems which numerical algorithms designed for the compressible flows. When the Mach number goes to zero ($M \longrightarrow 0$) the preconditioners can help to converge to the solution of the incompressible equations. Moreover, for problems with low speed flows, these preconditioners

accelerate the convergence to a steady state for which convergence without preconditioning would have been extremely slow.

The large disparity of the acoustic wave speed, u+a, and the waves convected at the fluid speed, u, make it difficult to solve compressible equations for low Mach numbers. In other words the ratio of the convective speed to the speed of sound is quite small [19]. So preconditioners remove this large disparity of wave speeds by changing the eigenvalues of the system, accelerating the convergence to steady state and changing the solution [20].

There are many preconditioning methods available. For this work Turkel preconditioning method has been used.

There are many engineering problems with both subsonic and super sonic regions. A flow in a duct can be highly subsonic but it become supersonic in a portion of the domain due to the changes in the geometry. So CFD users usually prefer to use existing compressible codes regardless of the value of the Mach number in order to avoid dealing with multiple flow codes [21].

1.5 Numerical methods

Numerical methods are at the heart of the CFD process. Accuracy and speed are two important features of numerical methods. The solution of partial differential equations (PDEs) used to describe the fluid flow is arrived with a compromise between accuracy and speed.

There are many ways for solving PDEs numerically. Finite Element Method (FEM), Finite Difference Method (FDM) and Finite Volume Method (FVM) are the main three methods applied to solve governing equation in CFD. Each has their own advantages according to the application. The CFD code used in this research is based on FEM-FVM schemes.

The finite-element method originated from the needs for solving complex elasticity, structural analysis problems in civil engineering and aeronautical engineering. The most important feature of FEM is its feasibility for complicated geometries. The whole domain is divided into small cells by either unstructured or structured mesh without any restriction. Turbulent flows were solved by FEM mostly based on RANS models but since mid 90's LES is also applied in FEM [22].

Commercial software are usually the immediate choice people make. Fluent is one of the most popular choices holding approximately 40% of the market share. It offers a convenient way to model fluid dynamics problems. The software code is based on the finite volume method to solve the governing equations. FLUENT has a wide array of physical models that can be applied to a wide array of industries. It also has a large number of turbulence models (In its latest version LES was added which was not available in FLUENT 6). But there are some disadvantages associated with Fluent. For example since it solves non-conservative forms of the governing equations, it may not yield the correct location of shocks in the flow field.

In this research three different numerical methods including our in-house FEM-FVM code, another in-house FEM code done by [17], and FLUENT simulation are compared together.

1.6 Thesis Objective and Outline

In the range of low Reynolds numbers many complicated phenomena take place within the boundary layer. Separation, laminar, transition and turbulent boundary layers, and reattachment could all occur within a short distance on the upper surface of airfoil at incidence. The LSB that commonly forms in this range of Reynolds numbers plays an important role in lift to drag ratio which is one of the important factors in external flow understanding. The formation of laminar separation bubble may have dominant effect on the flow field.

In this work the flow over NACA0025 airfoil at low Reynolds number is investigated. For this purpose LES, self-adaptive upwinding, and preconditioning methods which developed over the years by other members (K. Mohamed, M. Karimi and N. Tajallipour) of the CFD group at Concordia University are applied to the existing in-house code. The aim is to compare the results with experimental data conducted by Yarusevych et al., numerical results done by Toa Xu with incompressible CFD code, and with FLUENT's results as a user-friendly commercial CFD software.

The in-house CFD code used is a parallel version of a Finite Volume-Finite Element compressible Navier-Stokes solver. The necessary modifications

which include LES, preconditioning and self-adaptive upwinding are applied to this existing in-house code in order to make it more accurate and reliable for the purpose of this low Reynolds number flow over a NACA0025 airfoil. The parallel code runs at Mammouth Parallel Computer of Sherbrooke University and Cirrus server at Concordia University

The outline of this thesis is as follows: Chapter two is devoted to the governing equations, spatial and temporal discretizations, adaptive upwinding method, preconditioning and Large Eddy Simulation. In chapter three, Results are presented. Conclusions and future works and are discussed in chapter four.

2 Numerical Techniques

2.1 Governing Equations

Fluid flows are governed by Navier–Stokes (NS) equations that represent the conservation of mass, momentum and energy. The conservative form of NS equations in the absence of any source terms for three dimensional compressible flows can be written as,

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x_j}\left({}^{c}F_j+{}^{v}F_j\right)=0 \qquad 2.1$$

Where Q is the vector of conservative variables, ${}^{c}F_j$ is convective flux, and ${}^{v}F_j$ is viscous flux.

$$Q=\begin{bmatrix}\rho\\ \rho u_1\\ \rho u_2\\ \rho u3\\ \rho e\end{bmatrix},{}^{c}F_j=\begin{bmatrix}\rho u_j\\ \rho u_j u_1+p\delta_{j1}\\ \rho u_j u_2+p\delta_{j2}\\ \rho u_j u_3+p\delta_{j3}\\ \rho u_j h\end{bmatrix},{}^{v}F_j=\begin{bmatrix}0\\ -\sigma_{j1}\\ -\sigma_{j2}\\ -\sigma_{j3}\\ -u_k\sigma_{kj}+q_j\end{bmatrix}$$

e and h are the total specific energy and total specific enthalpy respectively.

$$e=e_I+\frac{1}{2}u_i^{\,2}\ ,\ e_I=c_vT \qquad 2.2$$

$$h=e+\frac{P}{\rho}\ ,\ h=c_pT \qquad 2.3$$

and pressure is calculated by ideal gas law knowing that for sea level air $R_g = 287 m^2.s^2.t^{-1}$,

$$p = \rho R_g T \tag{2.4}$$

Assuming Newtonian fluid, the tensor of viscous forces is written as,

$$\sigma_{ij} = \mu(t)\left(\left(u_{ij} + u_{ji}\right) - \frac{2}{3} u_{kk} \delta_{ij} \right) \tag{2.5}$$

Where μ is the dynamic viscosity of the gas. It varies as a function of temperature according to the Sutherland equation when $T < 2000K$,

$$\mu(t) = 1.71 \times 10^{-5} \left(\frac{T}{273.1} \right)^{\frac{3}{2}} \left(\frac{383.5}{T + 110.4} \right) \tag{2.6}$$

Conduction heat flux by Fourier law is,

$$q_i = -k(t) \frac{\partial T}{\partial x_i} \tag{2.7}$$

Where k is thermal conductivity and Prandtl number assumed constant and it is equal to 0.72 for the air.

$$k = \frac{\mu c_p}{\mathrm{Pr}} \tag{2.8}$$

2.2 Numerical Method

The numerical methods which are used in this research are based on a mixed finite element-finite volume scheme on an unstructured three dimensional finite element mesh [23]. According to this scheme, the finite element technique is used for diffusive fluxes, while for convective fluxes, a finite volume integration is applied. The discretization is followed according to references [24] and [25].

2.2.1 Spatial discretization

By multiplying a test function Ψ to the equation 2.1 and integration over the total domain Ω, weak formulation is achieved as,

$$\int \frac{\partial Q}{\partial t} \Psi dv + \int \nabla . \left({}^{c}F + {}^{v}F \right) \Psi dv = 0 \qquad 2.9$$

The whole domain is discretized with tetrahedral elements $E(I)$ (Figure 2.1). The control volumes which are formed by tetrahedral elements which are sharing a single node at the vertex, are used as the finite volume cells. The interpolation function for FEM discretization is defined as follows which $\phi(I)$ is a variable on this element such that,

$$\phi(I) = \sum_{I=1}^{N_s} \phi_I N_I(\vec{x}) \qquad 2.10$$

Where N_I is equal to one at node I and linearly goes to zero at the other nodes of the element. We also have a finite volume cell constructed by the contributions

of adjacent elements noted by $C(I)$ (Figure 2.2). For the finite volume integration the interpolation function is defined (M_I) such that it is equal to one over the entire control volume and zero everywhere else in the computational domain.

$$\phi(I) = \sum_{I=1}^{N_s} \phi_I M_I(\vec{x}) \qquad 2.11$$

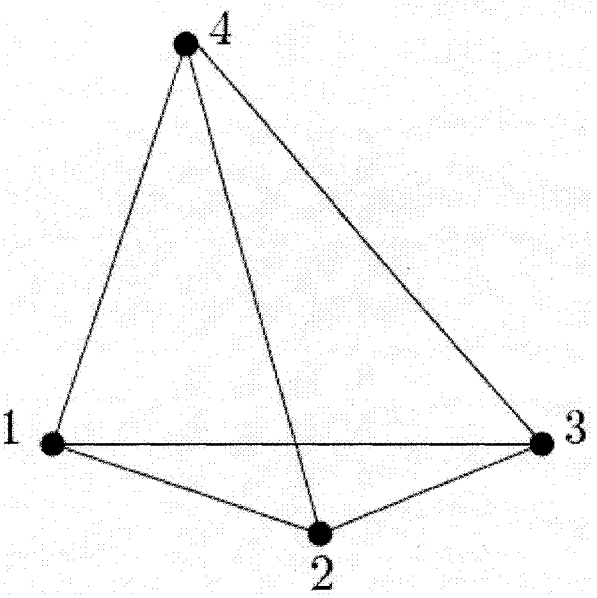


Figure 2.1: 3D element

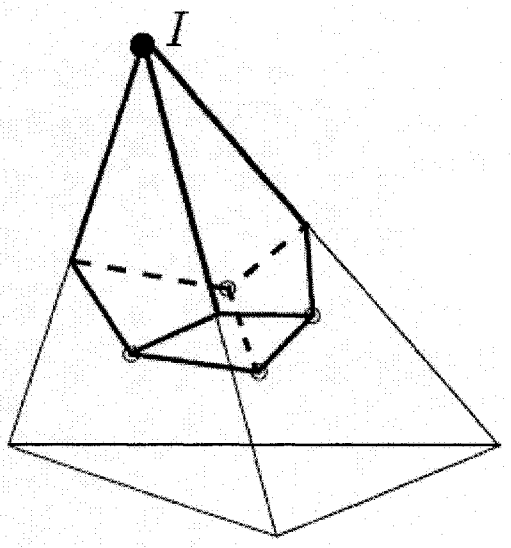


Figure 2.2: Contribution of an element to a cell

As mentioned earlier the convective terms are treated by FVM and diffusive terms are considered by the FEM as,

$$\int_{C(I)} \frac{\partial Q}{\partial t} dv + \int_{C(I)} \nabla .^{c} F(Q).dv = - \int_{E(I)} \nabla .^{v} F(Q) N_I dv \qquad 2.12$$

Here by applying the divergence theorem to the above equation, the final discretized weak formulation is:

$$\int_{C(I)} \frac{\partial Q}{\partial t} dv + \int_{\partial C(I)} {}^{c} F(Q).\vec{n}_{\partial C(I)} ds = \int_{E(I)} {}^{v} F(Q).\nabla N_I dv - \int_{\partial E(I)} {}^{v} F(Q).\vec{n}_{\partial \Omega} N_I ds \qquad 2.13$$

Where $\partial C(I)$ and $\partial E(I)$ are boundaries of the cells $C(I)$ and $E(I)$ respectively and $\vec{n}_{\partial C(I)}$ is the outward unit vector to the $\partial C(I)$.

2.2.2 Convective Flux Calculation

There are two options for the accuracy level for the convective calculation. For the first the flow field variables are considered constant over the entire control volume cells, while for the second order accurate convective flux calculation the flow field variables change linearly over the cells. Here the second order Roe-MUSCL method is used to calculate convective flux across cell boundary. As shown in Figure 2.3 the solution is approximated by linear function between two neighboring cells to achieve second order accuracy in space. It consists of an averaging term between two nodes plus an upwinding component. We can write this scheme as,

$$F_{IJ} = \frac{1}{2}\left[{}^{c}F(Q_{IJ}) + {}^{c}F(Q_{JI})\right] - \frac{1}{2}\partial^{c}F_{IJ} \qquad 2.14$$

Where Q_{IJ} and Q_{JI} are interface quantities that measured by extrapolation at boundaries of the cell as follows,

$$Q_{IJ} = Q_I + \frac{1}{2}(\nabla Q)_I . \vec{n}_{IJ} \qquad 2.15$$

$$Q_{JI} = Q_J + \frac{1}{2}(\nabla Q)_J . \vec{n}_{JI}$$

$$\partial^{c}F_{IJ} = |A_{IJ}|(Q_{JI} - Q_{IJ})$$

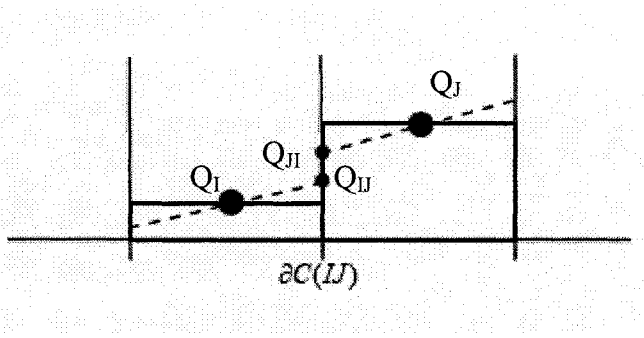


Figure 2.3: second order Roe method

2.2.3 Self Adaptive Upwinding Method

The material in this section is presented according to the method developed in reference [26]. The scheme used for convective flux calculation is found too dissipative for LES [27] although it is suitable for Euler and laminar flows. So the second term on the right hand side of equation 2.14 is examined here. In order to control the dissipation amount, a coefficient B is added to the Roe upwinding term, which can vary between 0 and 1. So we can rewrite this equation as,

$$\partial^c F_{IJ} = B \times |A_{IJ}|(Q_{JI} - Q_{IJ}) \qquad 2.16$$

Where:

B=0	pure central difference
0<B<1	upwinding model
B=1	full Roe-MUSCL

Table 1: value of upwinding coefficient

Note that the solution is unstable for pure central difference and the best turbulence solution is possible for the smallest possible B that can still lead to a stable solution.

A wiggle detector, [26], is used to adjust the upwinding coefficient dynamically. Based on this wiggle detection scheme, the wiggle exists along an edge, if the direction factor changes its sign at least twice (see the wiggle along the edge, connecting nodes i and i+1 in Figure 2.4:). In contrast, if it does not change its direction it means there is no wiggle present (see the value changing between nodes i-1 and i in Figure 2.4:.)

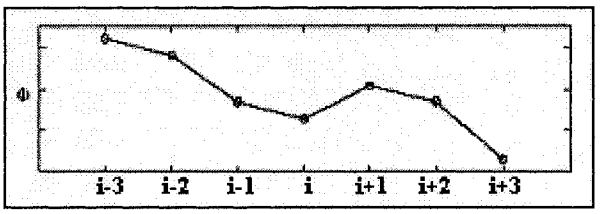


Figure 2.4: definition of a wiggle in the present computations.

Mathematically the wiggle existing criteria can be written as follows:

$$(\Phi_{i+1} - \Phi_i)(\Phi_i - \Phi_{i-1}) < 0 \qquad 2.17$$

$$(\Phi_{i+2} - \Phi_{i+1})(\Phi_{i+1} - \Phi_i) < 0$$

Where $\Phi \in [\rho, u, v, w, p]$ can be any of the flow variables.

Therefore, if this wiggle detector captures a wiggle, then the upwinding coefficient will be increased towards the full Roe-MUSCL using a linear function. Otherwise, the coefficient is decreased and the scheme is more as centered difference scheme.

For DNS methods, a very fine grid is used and the highest mode of energy spectrum of the flow is lower than the highest mode which numerical method is capable to capture. So even the smallest eddies diameter are bigger than the size of the local grids in the flows and all eddies can be captured and there is no wiggle in the flow. But in LES, the simulation should capture some energy in the highest scales. So a new modification is implemented by [26] which matches better with LES. Assume a tetrahedral's edge where $\vec{X}_I$ and $\vec{X}_J$ are the position of nodes I and J. Along the edge IJ we can calculate $(\nabla Q)^C$ ("C" stands for centered) as follows,

$$[(\nabla Q)^C . \vec{n}_{IJ}] = (Q_J - Q_I)/(X_J - X_I) \qquad 2.18$$

Replacing the above equations with equations 2.17 leads to:

$$[(\nabla Q)_I . \vec{n}_{IJ}][(\nabla Q)^C . \vec{n}_{IJ}] < \theta \qquad 2.19$$

$$[(\nabla Q)_J . \vec{n}_{IJ}][(\nabla Q)^C . \vec{n}_{IJ}] < \theta$$

So, because of the existence of this energy, a very small negative value for θ should be chosen. This small value helps to adjust the amount of energy in the smallest scales of our grid. Here θ is set to be 0.0001.

2.2.4 Temporal discretization

A backward second order implicit method for discretization of the unsteady term is applied for the temporal term. This scheme is always numerically stable and convergent but usually computationally expensive as it requires storage of two previous solution vectors and solving a system of linear equations at each time step [28]. In order to do this, the temporal term in the NS equations is kept on the left hand side of equation 2.9 while all the non-temporal components are taken to the right hand side as shown in equation 2.20,

$$\int_{C(I)} \frac{\partial Q}{\partial t} dv = K(Q) \qquad 2.20$$

Because of the undeformable mesh type, the time derivative Q is constant over cell $C(I)$ and can be brought out of integrand part. ($vol(C(I))$ is the control volume)

$$vol(C(I))\frac{\partial Q^{n}}{\partial t} = K(Q) \tag{2.21}$$

For implicit time integration, the right hand side is written at time step n+1, as

$$vol(C(I))\frac{\partial Q^{n}}{\partial t} = K(Q^{n+1}) \tag{2.22}$$

The above equation is linearized by Taylor series expansion.

$$K(Q^{n+1}) = K(Q^{n}) + \frac{\partial K(Q^{n+1})}{\partial t}\Delta t + \frac{\partial^2 K(Q^{n+1})}{\partial t^2}\frac{\Delta t^2}{2} + O(\Delta t)^3 \tag{2.23}$$

Using chain rule leads to,

$$\frac{\partial K(Q^{n+1})}{\partial t} = \frac{\partial K(Q^{n+1})}{\partial Q^{n}}\frac{\partial (Q^{n})}{\partial t} \tag{2.24}$$

$$\frac{\partial^2 K(Q^{n+1})}{\partial t^2} = \frac{\partial K(Q^{n+1})}{\partial Q^{2n}}\frac{\partial^2 (Q^{n})}{\partial t^2}$$

Second order backward finite difference time discretization is applied as follows,

$$\frac{\partial Q^{n}}{\partial t} = \frac{(1+2\tau)/(1+\tau)Q^{n+1} - (1+\tau)Q^{n} + (\tau^2)/(1+\tau)Q^{n-1}}{\Delta t} \tag{2.25}$$

$$\frac{\partial^2 Q^{n}}{\partial t^2} = \frac{(2\tau)/(1+\tau)Q^{n+1} - (2\tau)Q^{n} + (2\tau^2)/(1+\tau)Q^{n-1}}{\Delta t^2}$$

Substituting all above terms into equation 2.20 yields,

$$\left[a\frac{vol(C(I))}{\Delta t} - (a+c)\frac{\partial K(Q^{n+1})}{\partial Q_h}\right]\Delta Q^{n} = K(Q^{n}) + \left[b\frac{vol(C(I))}{\Delta t} - 2b\frac{\partial K(Q^{n+1})}{\partial Q}\right]\Delta Q^{n} \tag{2.26}$$

Where :

$$\tau = \frac{\Delta t^{n}}{\Delta t^{n-1}}$$

$$a = \frac{1+2\tau}{1+\tau}\ ,\ b = \frac{\tau^2}{1+\tau}\ ,\ c = \frac{\tau}{1+\tau}$$

The equation 2.26 can now be iteratively solved by the help of GMRES iterative solver.

2.2.5 Preconditioning

When the Mach number decreases, the precision of compressible solvers falls [29]. To avoid stability problems within the limits of low Mach numbers, a classical approach is preconditioning the dissipative terms of a compressible scheme [30]. One of the easiest preconditioning methods proposed by Turkel is used in this work. One can write the dissipative term as,

$$\delta^c F = |A|\delta Q \qquad 2.27$$

The preconditioner is realized by a matrix P and becomes:

$$\delta^c F = P_c^{-1}|P_c A|\delta Q \qquad 2.28$$

Where:

$$P_c = MP_eM^{-1} \qquad 2.29$$

$$M = \frac{\partial Q}{\partial U}$$

Assuming $\beta = 1/M_\infty$ where M_∞ is the Mach number, P_e is defined as,

$$P_e = \begin{bmatrix} \beta^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad 2.30$$

This method decreases the truncation error of the scheme within the limits of low Mach numbers. So it is a better representative of the behavior of acoustic waves [22]. Also the convergence of numerical simulations is more delicate while using the compressible codes in very low Mach numbers. In order to solve this problem, the matrix P_c^{-1} should be multiplied to the unsteady term as follows,

$$P_c^{-1} \frac{\partial Q}{\partial t} \qquad 2.31$$

This can increase the convergence rate that will be discussed later.

2.2.6 LES

As mentioned in the introduction, the modeling of turbulence is based on Large Eddy Simulations. Two important steps towards the LES method are filtering and Sub-Grid-Scale (SGS) modeling. A spatial filter is applied to the domain to simulate large scales of the flows. Thus, each variable can be broken into a resolved part as a large scale $\bar{\phi}$, and an unresolved part of a small scale (sub grid) ϕ' where ϕ can be any variable. This filter can be a function such as,

$$\bar{f} = \int_{\Omega} G_{\Delta}(x-\xi) f(\xi) d\zeta \qquad 2.32$$

Where $\bar{f}$ is the large scale component of a variable, G is the filter function, and Δ is the filtering size.

For compressible flows, it is convenient to define Favre filtering operation of the variable ϕ as:

$$\tilde{\phi} = \frac{\overline{\rho\phi}}{\bar{\rho}} \qquad 2.33$$

where the variable is decomposed to the filtered and fluctuating components as follows,

$$\phi = \tilde{\phi} + \phi' \qquad 2.34$$

Applying the Favre filtering operation to the pre-discussed Navier-Stokes equations leads to the following equation:

$$\frac{\partial \bar{Q}}{\partial t} + \frac{\partial}{\partial x_j}\left({}^{c}\bar{F}_j + {}^{v}\bar{F}_j\right) = 0 \qquad 2.35$$

Where Q is the vector of conservative variables, ${}^{c}F_j$ and ${}^{v}F_j$ are convective and viscous fluxes respectively.

$$\bar{Q} = \begin{bmatrix} \bar{\rho} \\ \overline{\rho\tilde{u}_1} \\ \overline{\rho\tilde{u}_2} \\ \overline{\rho\tilde{u}_3} \\ \overline{\rho\tilde{e}} \end{bmatrix}, {}^{c}\bar{F}_j = \begin{bmatrix} \overline{\rho\tilde{u}_j} \\ \overline{\rho\tilde{u}_j\tilde{u}_1} + \bar{p}\delta_{j1} \\ \overline{\rho\tilde{u}_j\tilde{u}_2} + \bar{p}\delta_{j2} \\ \overline{\rho\tilde{u}_j\tilde{u}_3} + \bar{p}\delta_{j3} \\ \overline{\rho\tilde{u}_j}(\overline{\rho\tilde{e}} + \bar{p}) \end{bmatrix}, {}^{v}\bar{F}_j = \begin{bmatrix} 0 \\ -\tilde{\sigma}_{j1} - \tilde{\tau}_{j1} \\ -\tilde{\sigma}_{j2} - \tilde{\tau}_{j2} \\ -\tilde{\sigma}_{j3} - \tilde{\tau}_{j3} \\ -\tilde{u}_k(\tilde{\sigma}_{kj} + \tilde{\tau}_{kj}) + \tilde{q}_j + \tilde{q}_j^t \end{bmatrix}$$

In the above vectors, the filtered viscous constraints tensor and heat flux conduction are:

$$\tilde{\sigma}_{ij} = \mu(t)\left(\left(\tilde{u}_{i,j} + \tilde{u}_{j,i}\right) - \frac{2}{3}\tilde{u}_{k,k}\delta_{i,j}\right) \quad 2.36$$

$$\tilde{q}_i^{\,t} = -k(t)\frac{\partial \tilde{T}}{\partial x_i} \quad 2.37$$

The sub-grid-scale stress tensor and heat flux are:

$$\tilde{\tau}_{ij} = -\left(\overline{\rho u_i u_j} - \overline{\rho}\tilde{u}_i\tilde{u}_j\right) = \mu_t\left(\tilde{u}_{ij} + \tilde{u}_{ji} - \frac{2}{3}\tilde{u}_{kk}\delta_{ij}\right) + \frac{1}{3}\tilde{\tau}_{kk}\delta_{ij} \quad 2.38$$

$$\tilde{q}_i = c_p\left(\overline{\rho u_i T} - \overline{\rho}\tilde{u}_i\tilde{T}\right) \quad 2.39$$

2.2.6.1 Smagorinsky-Lilly Model

For LES, the Smagorinsky subgrid scale model is the most commonly accepted SGS model. This model uses subgrid viscosity based on the scales resolved [31] with a constant factor defined as,

$$\mu_t = 2\overline{\rho}C_s^2\Delta^2|S| \quad 2.40$$

The Favre averaged strain tensor is:

$$|S| = \sqrt{2S_{ij}(\tilde{u})S_{ij}(\tilde{u})} \quad \text{and} \quad S_{ij}(\tilde{u}) = \frac{1}{2}\left(\tilde{u}_{ij} + \tilde{u}_{ji}\right) \quad 2.41$$

And the filter width is defined as,

$$\Delta = \left(Vol(C_I)\right)^{\frac{1}{3}} = \left(dxdydz\right)^{\frac{1}{3}} \quad 2.42$$

Where C_s is the Smagorinsky constant. This constant is not a universal constant and finding the appropriate value for it is the most challenging decision for this model. A value around 0.1 usually has been used for a wide range of flows which leads to satisfactory results [32]. However, this value was found to cause excessive damping of large-scale fluctuations near solid boundaries and has to be reduced in such regions.

2.2.7 FLUENT

In this section a brief summary of the numerical formulation that is used in FLUENT software is discussed. FLUENT uses a coupled solver with control-volume-based technique to solve the governing equations of continuity, momentum, and energy (where appropriate) simultaneously. The FVM used in FLUENT convert the governing equations to algebraic equations that can be solved numerically by integrating the governing equations about each control volume. A two dimensional triangular cell is shown in Figure 2.5 as an example of such a control volume [33].

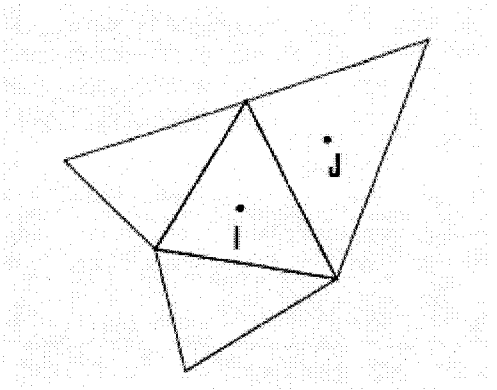


Figure 2.5: Control Volume used in FLUENT

Discretization of non-temporal components of the governing equations written in integral form for a scalar quantity ϕ and an arbitrary control volume is as follows:

$$\oint \rho\phi\vec{v}.d\vec{A} = \oint \Gamma_\phi \nabla\phi.d\vec{A} \qquad 2.43$$

Where $\vec{A}$ is the surface normal vector, Γ_ϕ is diffusion coefficient for quantity ϕ and $\nabla\phi$ is the gradient of ϕ. This equation is applied to each control volume of the computational domain. Discretization of Equation 2.43 on a given cell yields.

$$\sum_{f}^{N_{faces}} \rho_f \vec{v}_f \phi_f .\vec{A}_f = \sum_{f}^{N_{faces}} \Gamma_\phi (\nabla\phi)_n .\vec{A}_f \qquad 2.44$$

Where N_{faces} is number of faces enclosing cell, ϕ_f is value of ϕ convected through face f, $\vec{A}_f$ is normal vector of face f and $(\nabla\phi)_n$ is magnitude of $\nabla\phi$ across the face f.

The diffusion terms in equation 2.44 are always central differenced and second order accurate in space. The face value ϕ_f in convective terms are

calculated by an upwind method via interpolation from the cell center values (I and J in Figure 2.5). Therefore, for second-order accuracy quantities at cell faces are computed using a multidimensional linear function. Thus when second-order upwinding is activated, ϕ_f is computed as,

$$\phi_f = \phi + \nabla\phi.\Delta\vec{s} \qquad 2.45$$

Here ϕ and $\nabla\phi$ are the cell-centered value and its gradient in the upstream cell respectively, and $\Delta\vec{s}$ is the displacement vector from the upstream cell centroid to the face centroid. This gradient is computed using the divergence theorem, which in discrete form is written as

$$\nabla\phi = \frac{1}{V}\sum_{f}^{N_{faces}} \tilde{\phi}_f.\vec{A} \qquad 2.46$$

Where the face values of $\tilde{\phi}_f$ are computed by averaging ϕ from the two cells adjacent to the face.

A second order backward implicit scheme is applied for discretization of the temporal term. As discussed before the spatial components of Navier-Stokes equations discretization, $F(\phi)$, are taken to the left hand side as follows,

$$\frac{\partial\phi}{\partial t} = F(\phi) \qquad 2.47$$

The second order implicit formulation is given by

$$\phi^{n+1} = \frac{4}{3}\phi^{n} - \frac{1}{3}\phi^{n-1} + \frac{2}{3}\Delta t F(\phi^{n+1}) \qquad 2.48$$

2.2.8 Incompressible FEM code

This section presents the numerical methods that Tao Xu [17] used in his simulation. The code is a three dimensional incompressible LES solver. The governing equations are unsteady filtered Navier-Stokes equations using Smagorinsky model as:

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}\left(\overline{u}_i \overline{u}_j\right) = -\frac{\partial \overline{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\upsilon_t\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\right] \qquad 2.49$$

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \qquad 2.50$$

Where υ_t (the total viscosity) and υ_{sgs} (the eddy viscosity of the Smagorinsky model) are:

$$\upsilon_t = \upsilon + \upsilon_{sgs} \qquad 2.51$$

And the modified pressure term with the SGS stress tensor is:

$$\overline{P} = \overline{p} + \frac{1}{3}\tau_{kk}\delta_{ij} \qquad 2.52$$

The code uses a dynamic Smagorinsky model by applying a second filter to the equations. This means instead of the constant C_s, the coefficient can be adapted locally to the characteristics of the flow as a function of time and the location, $C_s = C_s(x,t)$.

Finite Element Method is used for discretizating the above equations. For spatial discretization a second order Galerkin FEM is used. An Implicit second

order backward Euler scheme is applied for the discretization of temporal terms. Also a multigrid technique is used as the linear solver at this simulation to solve the problem using coarser grids in order to accelerate the convergence rate.

3 Comparison of numerical results and experimental validation

Before reporting the results obtained numerically, we first present the experimental work that will be used for validation.

3.1 Experiment of Yarusevych

Yarusevych et al. [11] analyzed a laminar separation bubble on the upper surface of the NACA0025 airfoil at low Reynolds numbers flows. The experiment is conducted in a low-turbulence recirculating wind tunnel at the University of Toronto. As shown in Figure 3.1 the wind tunnel has 5-m-long octagonal test section with a span of 0.91m, and height of 1.215 m. The airfoil is mounted horizontally inside the wind tunnel, 0.4m downstream of the contraction section. The airfoil has a 0.3m chord and a span of 0.88m. Smoke wire technique is employed for boundary layer and wake formation visualization. A digital protractor changes the angle of attack with uncertainty of 0.1 degrees. The airfoil's surface is equipped by 65 pressure taps to measure the pressure distribution along the upper and lower surfaces in the mid-span. Interested readers can find the detailed information about the experiment in [11] and [16].

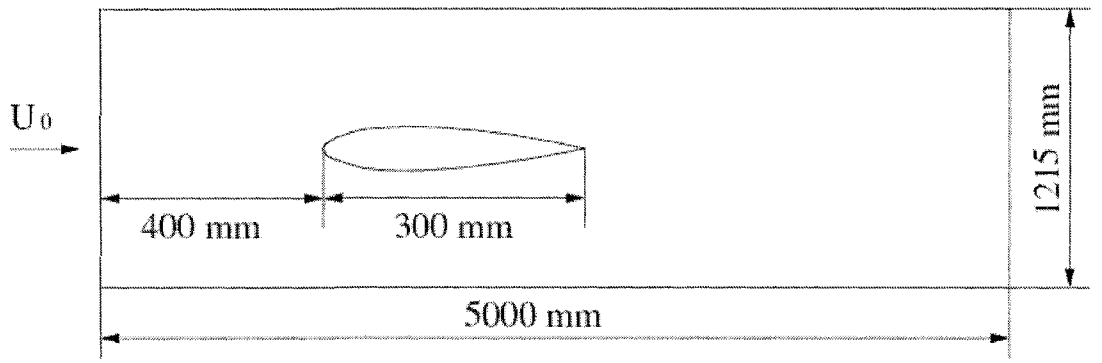


Figure 3.1: experimental test section

3.2 The scope of present research

The flow around a symmetrical airfoil (NACA0025) is used for validation in this research. The prediction of surface pressure distribution and the location of separation points are the main objectives of the simulation. The flows conditions are chord Reynolds number of 100000 at attack angles of 0 and 5 degrees. Simulations are done using the second-order accurate convective flux calculations and the self-adaptive upwinding method. The results are compared with the experimental data by Yarusevych et al. [11] to see if the numerical simulation can predict the flow separation at the same location as the experimental studies. The results are then compared against FLUENT (commercial software) and to the in-house incompressible LES code by [17] as well.

3.3 Computational domain geometry and meshing

The geometry used in this simulation is a NACA0025 airfoil same as in the experiment. The chord length(c) is 0.3m. The span length is chosen one chord (0.3m) in order to avoid the walls effects at mid-span although it is 0.15c in the experiment.

The computational domain has 0.5c upstream of the leading edge and one chord downstream of the trailing edge. The height of the test section is 4c equal to the height of the wind tunnel.

The entire domain is meshed with the Gambit mesh generator software to construct 4-node unstructured tetrahedral elements. In order to accurately capture the separation and vortices in the flow field near the wall, a very fine boundary layer mesh is required at that region. To illustrate mesh grid that is used, a slice view of the mesh is shown in Figure 3.2.

There is a problem generating the mesh inside the boundary layer. The code used for this simulation can only work with the tetrahedral elements; however, Gambit can not provide tetrahedral cells inside the boundary layer. Hence, in order to solve this problem, first the whole domain is meshed by tetrahedral elements except the boundary layer which is meshed by prism elements. Then another code is used to break the prism to tetrahedral elements. The final mesh compromised of 50 layers of structured mesh with the ratio of 1.05 which smoothly mix with the grids outside of boundary layer.

The finest mesh that is possible to construct according to the available computational resources is composed of around 3 million nodes generated on Cirrus server using 64GB of memory. The produced mesh is then decomposed into 64 sub-domains, as shown in Figure 3.3, to be run in parallel on either supercomputer clusters: Cirrus or Sherbrooke servers.

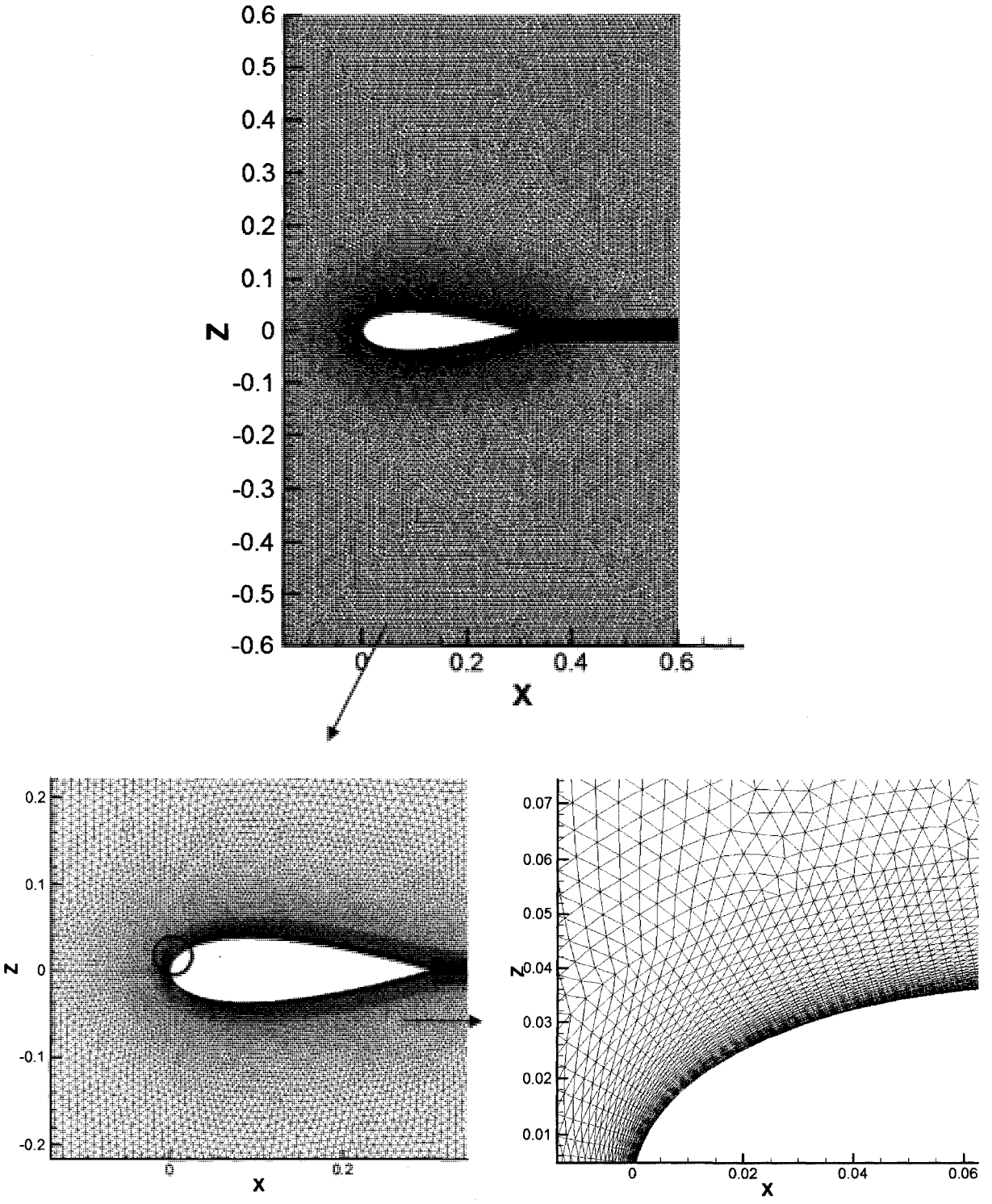


Figure 3.2: A slice view of the mesh

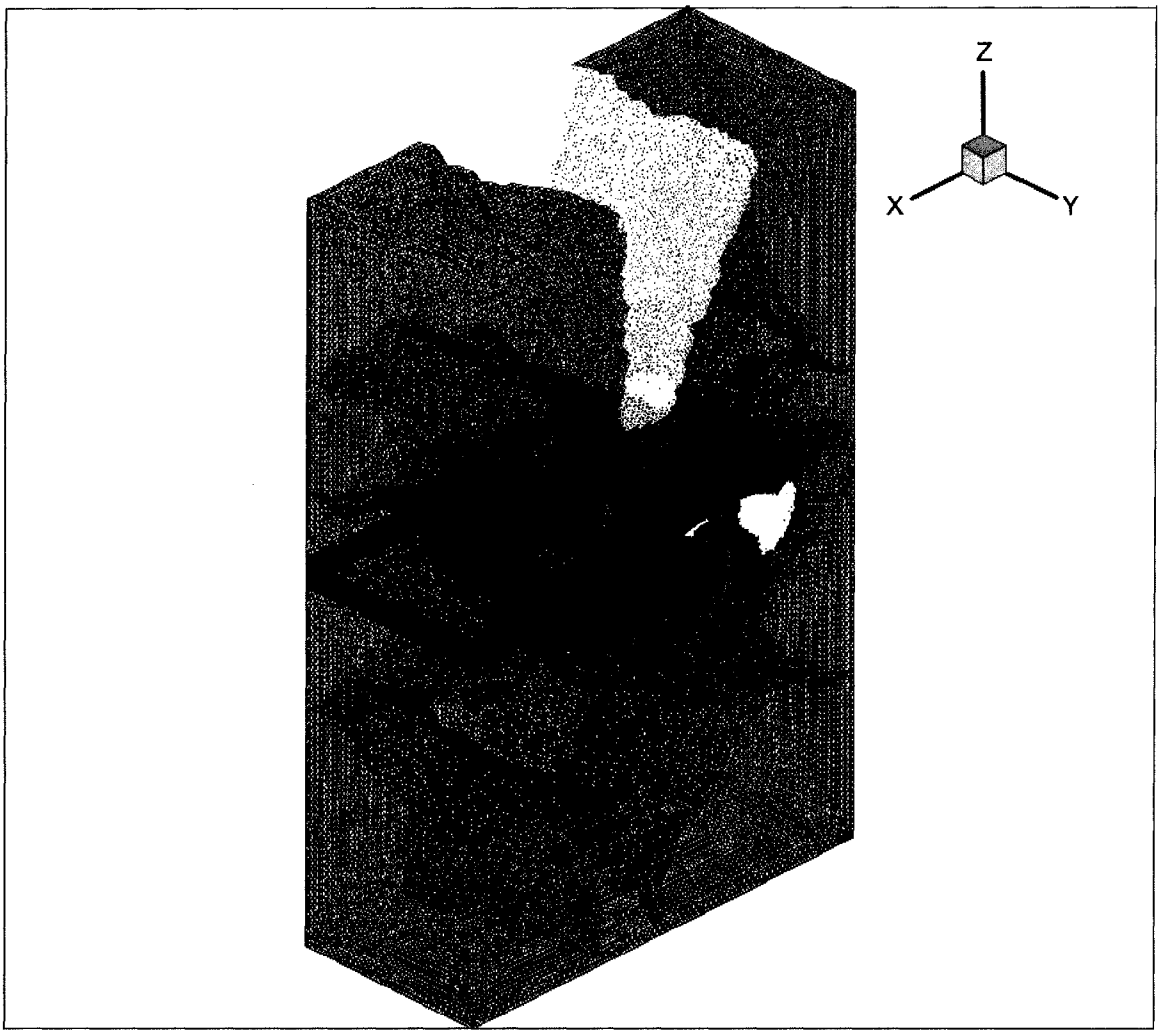


Figure 3.3: Partitioned mesh

Mesh resolution in LES method is a very important criteria which can significantly affect the results. Mesh size is usually specified in wall units.

A wall unit is defined as,

$$\Delta = \frac{yu_\tau}{\nu} \tag{3.1}$$

Where the friction velocity is,

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \tag{3.2}$$

and the wall shear stress is

$$\tau_w = \mu\left(\frac{\partial u}{\partial y}\right)_{y=0} \cong \frac{0.0288\rho u^2}{\mathrm{Re}^{1/5}} \tag{3.3}$$

The non-dimensional normal distance from the wall is:

$$\Delta y^+ = \frac{yu_\tau}{\nu} \tag{3.4}$$

Where y is the normal distance to the wall.

For LES method the resolution of Δy^+ has to be in the order of unity, [34], in order to accurately resolve the velocity gradient at the wall vicinity. For the $\mathrm{Re}_c = 100000$ and $\Delta y^+ = 1$ the first node after the wall should be located at most at $y = 2\times10^{-4}$ (m) from the wall.

In our simulation the resolution of the mesh on the wall is $\Delta y^+ \approx 1$ and $\Delta x^+ = \Delta z^+ \approx 20$ (streamwise and spanwise spacing) in wall units for the finest mesh used. The largest edge length is $\Delta x^+ = \Delta y^+ = \Delta z^+ \approx 100$ in the far field regions. This helps to have enough near wall resolution. This mesh is composed

of about 3 million nodes and 9.5 millions elements. There is also a coarser mesh used with $\Delta y^+ \approx 10$ and one million nodes, however it cannot capture the separation at all as shown in Figure 3.12.

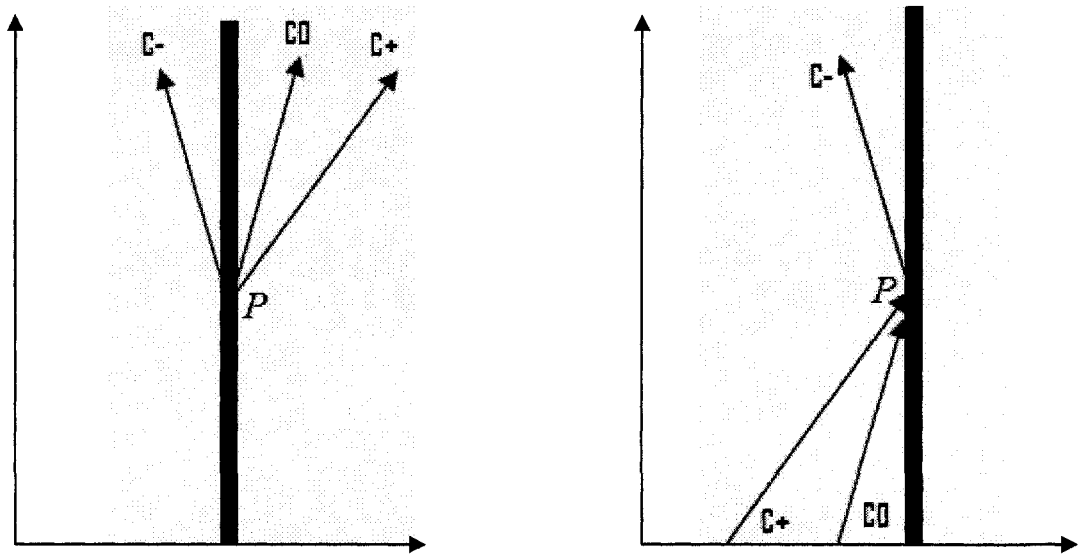
For the sake of numerical stability of the solution the time step for time marching has to be set according the smallest size of the grids in the mesh. Therefore, by increasing the number of nodes (finer mesh), we should choose smaller time steps. However, there is no problem of stability because of the implicit method for the temporal integration used. Nevertheless, in order to accurately capture the rapid changes in the flow variables, the time step has to be set reasonably small. In our simulation the steps are controlled by CFL which linearly increase from 1 to 5. So the maximum CFL is limited to 5 in this work and the maximum time step is chosen to be $\Delta t = 0.0001$ (s). The simulation continues running until the flow becomes statistically steady. The best way to make sure if the flow is fully developed and steady is to watch the residuals and the forces on the airfoil. Afterwards, the sampling data statistics should be extracted. Therefore, the simulation should be continued for a period of time and the quantities are averaged. This time has to be much longer than the period of flow oscillations and for this work 200 iterations are chosen.

3.4 Boundary conditions

Three types of boundary conditions are applied in this simulation including inflow, pressure outflow and non-slip wall. The non-slip boundary conditions are used for the airfoil surface and all the channel walls while inflow and outflow boundary conditions are used for the inlet and outlet planes of the channel.

3.4.1 Inlet and outlet

Boundary conditions depend on the characteristic properties of the governing equations. The fluxes at the boundaries are determined by the wave propagation directions. These waves are the eigenvalues ($\vec{u}.\vec{n}$,$\vec{u}.\vec{n}$,$\vec{u}.\vec{n}$, $\vec{u}.\vec{n}+c$, and $\vec{u}.\vec{n}-c$) of Jacobian matrix of the flux vectors [25]. Flux calculations at the boundaries are done by the superposition of stemming waves from interior domain and incoming waves based on the boundaries variables. Figure 3.4 shows incoming and outgoing waves at the boundaries and their influence in one dimensional computational domain. For the incoming waves, the physical conditions should be imposed and for outgoing waves the variables are set using information coming from the interior domain. Also non-reflecting boundary condition is used for the outgoing waves by changing the characteristic variable of the wave to zero in order to prevent the reflection at the boundary [35].



(a) (b)

Figure 3.4: Waves characteristics for a subsonic boundary condition:

(a) Inflow (b) Outflow

As shown in Table 2 the velocities and density are imposed at inflow plane and for outflow plane static pressure is set. So, there are four variables set from the physical domain and one comes from numerical solution at the inlet. For the outlet one variable is set according to the physical domain and the remaining four variables come from the numerical solution.

Inflow	Outflow
ρ, u, v, w	p

Table 2: Fixed variables at subsonic boundary condition

For the case studies herein the incoming velocities are set as follows,

$u = U_\infty \cos\alpha$, $v = 0$, and $w = U_\infty \sin\alpha$

Where U_∞ is the free stream velocity and α is the attack angle.

3.4.2 Solid walls

For the solid walls (airfoil surface and channel walls), zero velocity condition is applied both in parallel and normal directions to the surface. This is to ensure that the solid walls are non-slip and non-porous walls to be compatible with the viscous flow and non-permeable walls assumptions respectively. In addition, to avoid the influence of heat transfer through the walls, the adiabatic boundary condition is also applied to the solid walls by means of setting the temperature gradient equal to zero across these surfaces. In summary these conditions can be expressed as follow,

$$\vec{u} = 0 \qquad 3.5$$

$$\nabla T.\vec{n} = 0 \qquad 3.6$$

3.5 Numerical Results

3.5.1 Compressible LES Code

In order to introduce the flow that is studied herein, non-dimensional time averaged pressure and velocity contours for $\mathrm{Re}_c = 100000$ at $\alpha = 5^\circ$ simulation are shown. In Figure 3.5 pressure contours show a stagnation region near the nose and a low pressure region on the upper surface. Also velocity contours in the vicinity of the airfoil show the high speed region above the airfoil as shown in Figure 3.6.

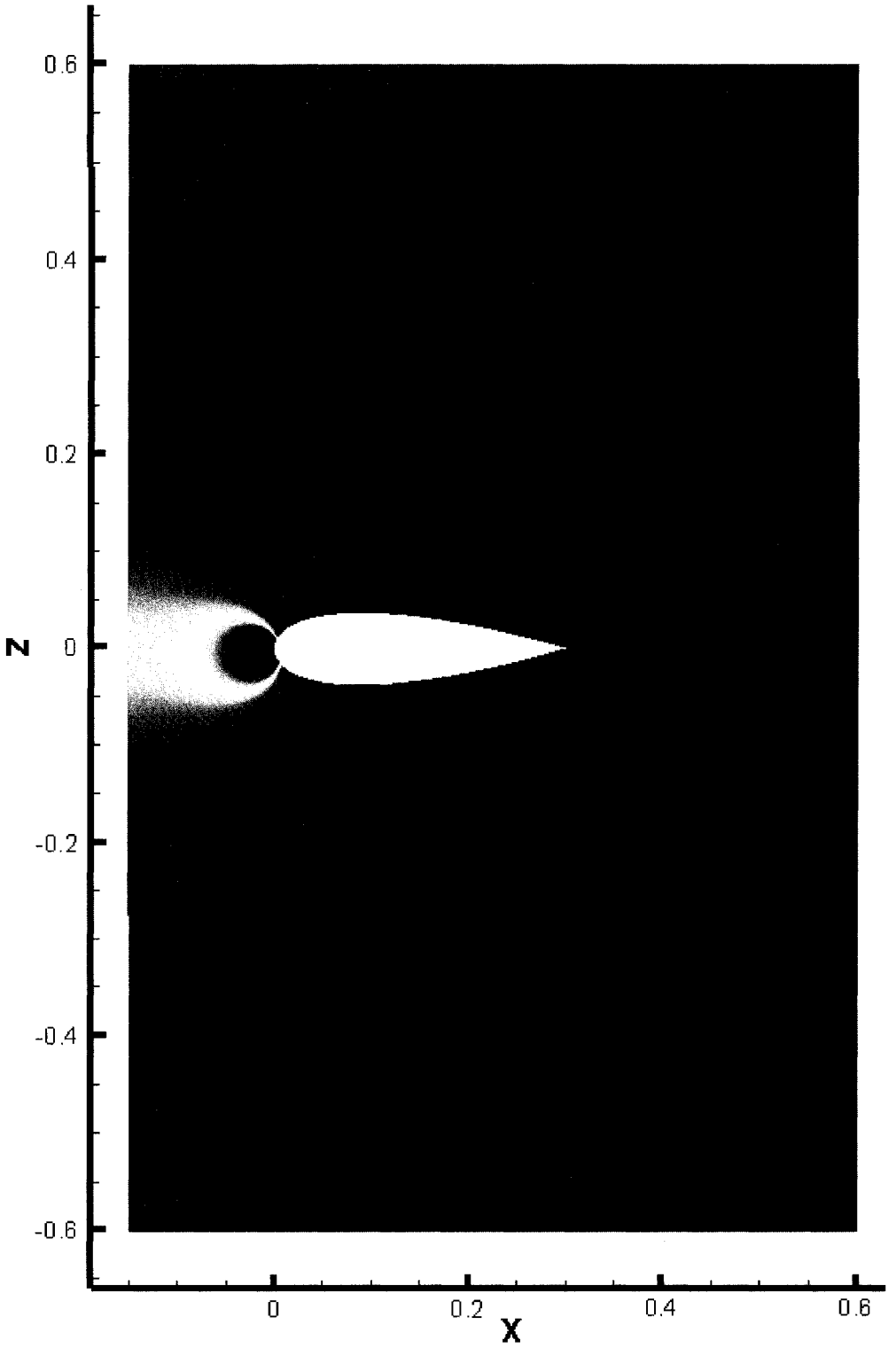


Figure 3.5: pressure contours at $\alpha = 5^{\circ}$

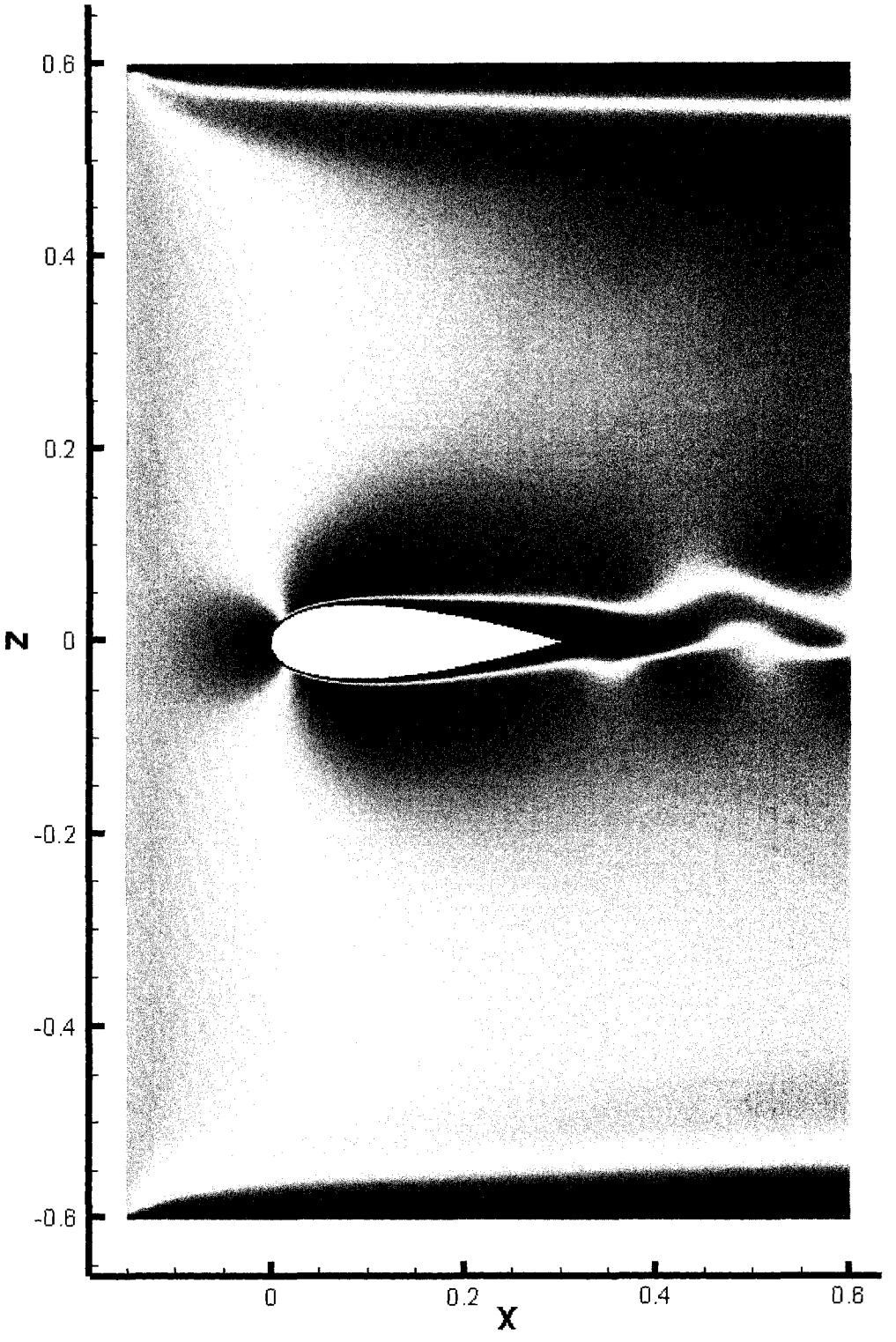


Figure 3.6: velocity contours at $\alpha = 5^{\circ}$

Figure 3.8 reveals that the compressible LES code with self-adaptive upwinding method used at this work is able to capture the separation and recirculation zone which are the most important features for the boundary layer in this case. As it can be seen the flow separates on the upper surface of the airfoil and fails to reattach. It is good to note that the code without self-adaptive upwinding method can not capture the separation in this simulation. As the streamlines show in Figure 3.9 the flow completely attaches to the airfoil's surface and no separation occurres.

Boundary layers tend to separate from a solid body when there is an increasing fluid pressure in the direction of the flow known as an adverse pressure gradient. The separation occurs where the velocity gradient reaches zero, $\left(\frac{\partial u}{\partial z}\right)_0 = 0$ and then the reverse flow begins where this term becomes negative, $\left(\frac{\partial u}{\partial z}\right)_0 < 0$. Figure 3.7 depicts a schematic view of the flow separation process over the upper surface of a typical airfoil.

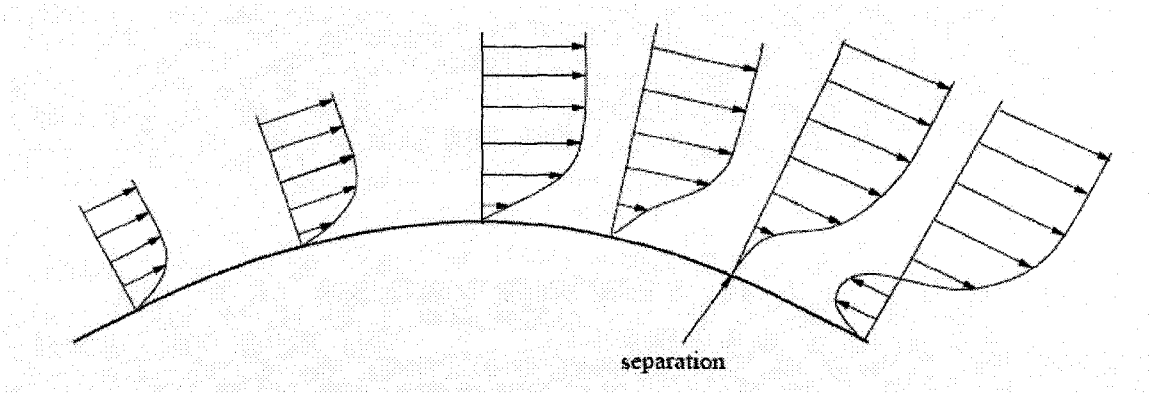


Figure 3.7: separation of boundary layer

Figure 3.10 presents the corresponding situation from the current numerical simulation where the velocity vectors inside the boundary layer are shown. The arrow indicates the point where the separation takes place which is approximately at x/c=0.31.

Figure 3.11 shows a reverse flow in the separated region and formation of a big wake over the upper surface of the airfoil.

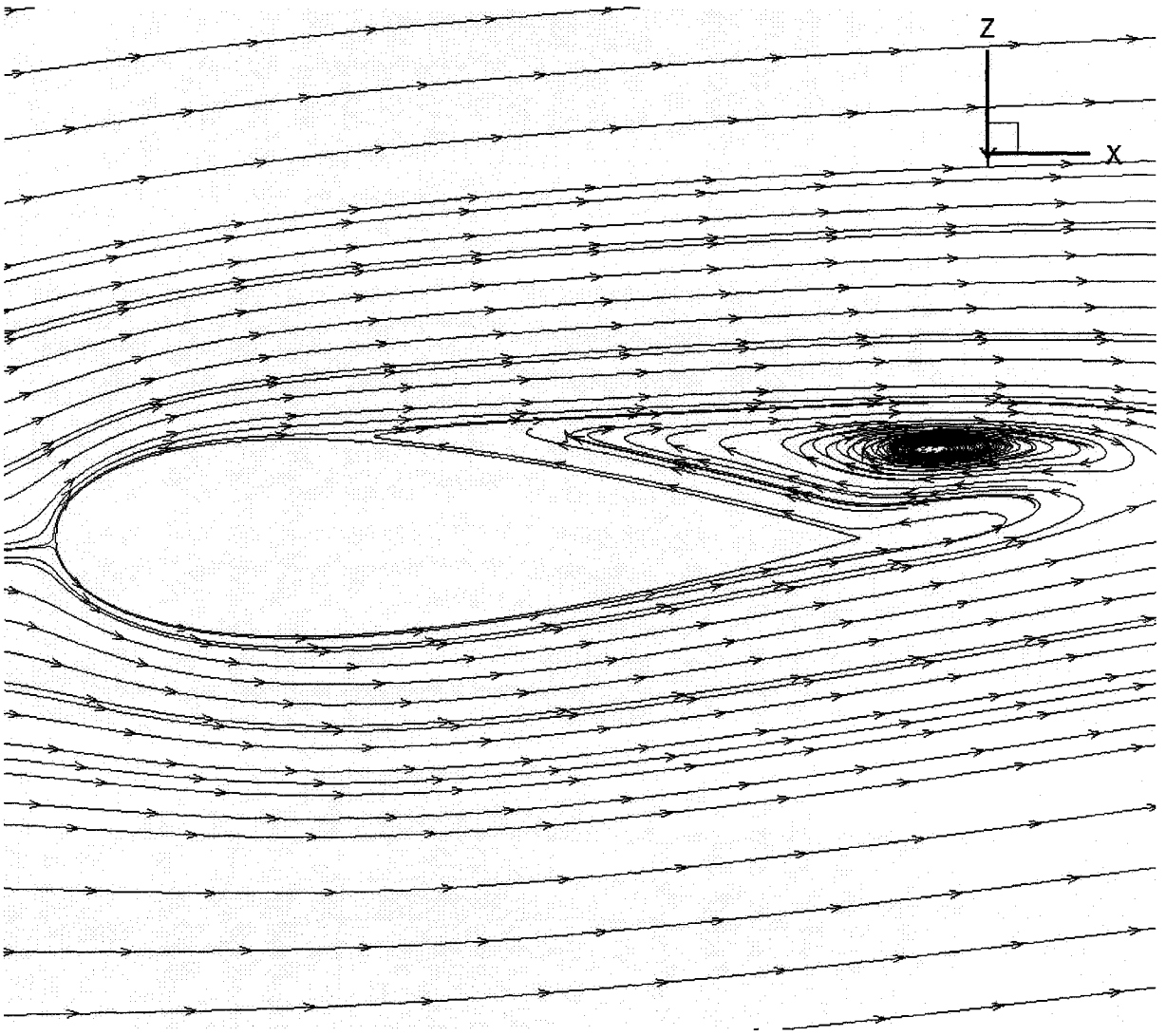


Figure 3.8: streamlines of velocity at $\alpha = 5^{\circ}$

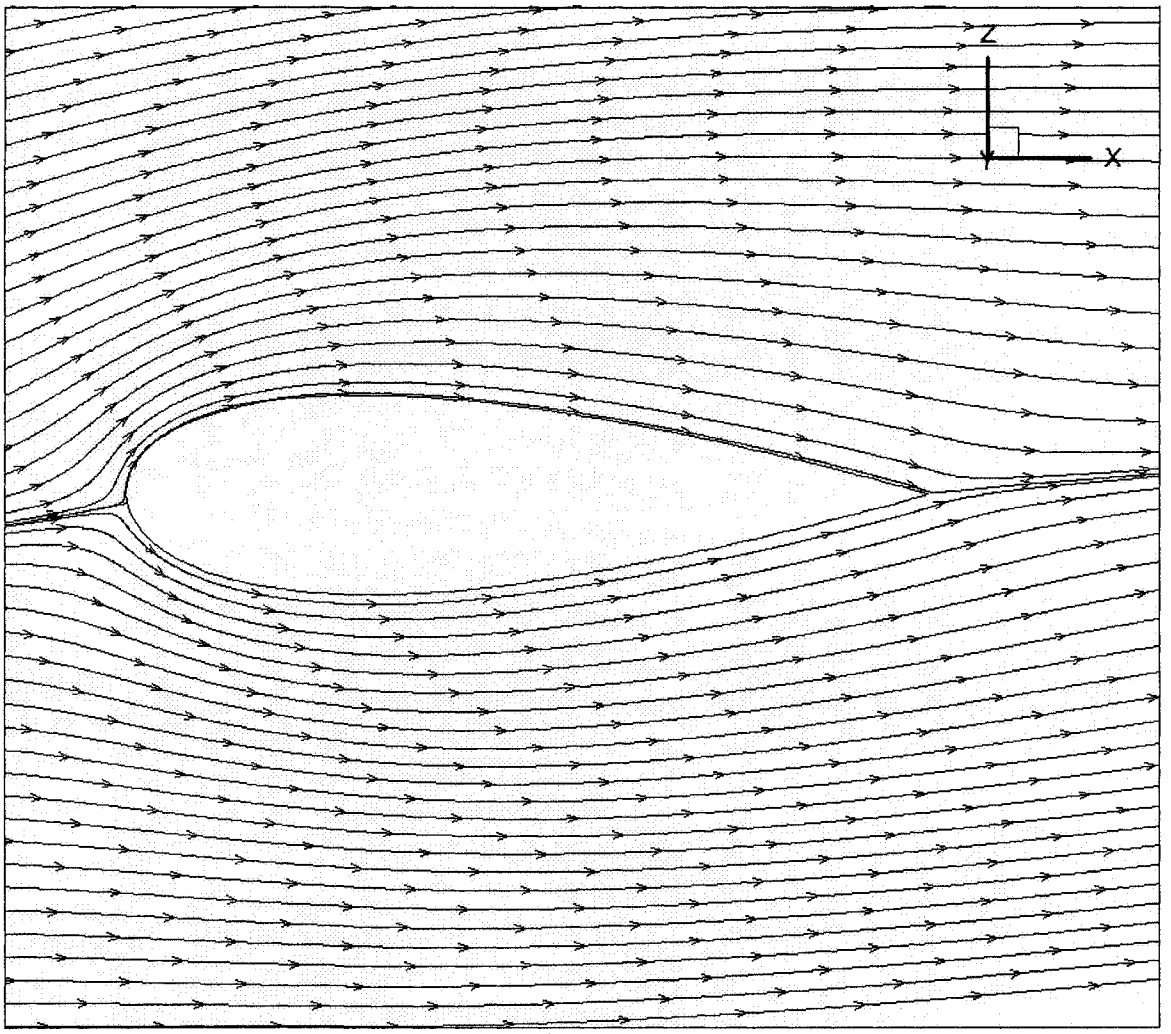


Figure 3.9: streamlines of velocity at $\alpha = 5^{\circ}$

(without self-adaptive upwinding method)

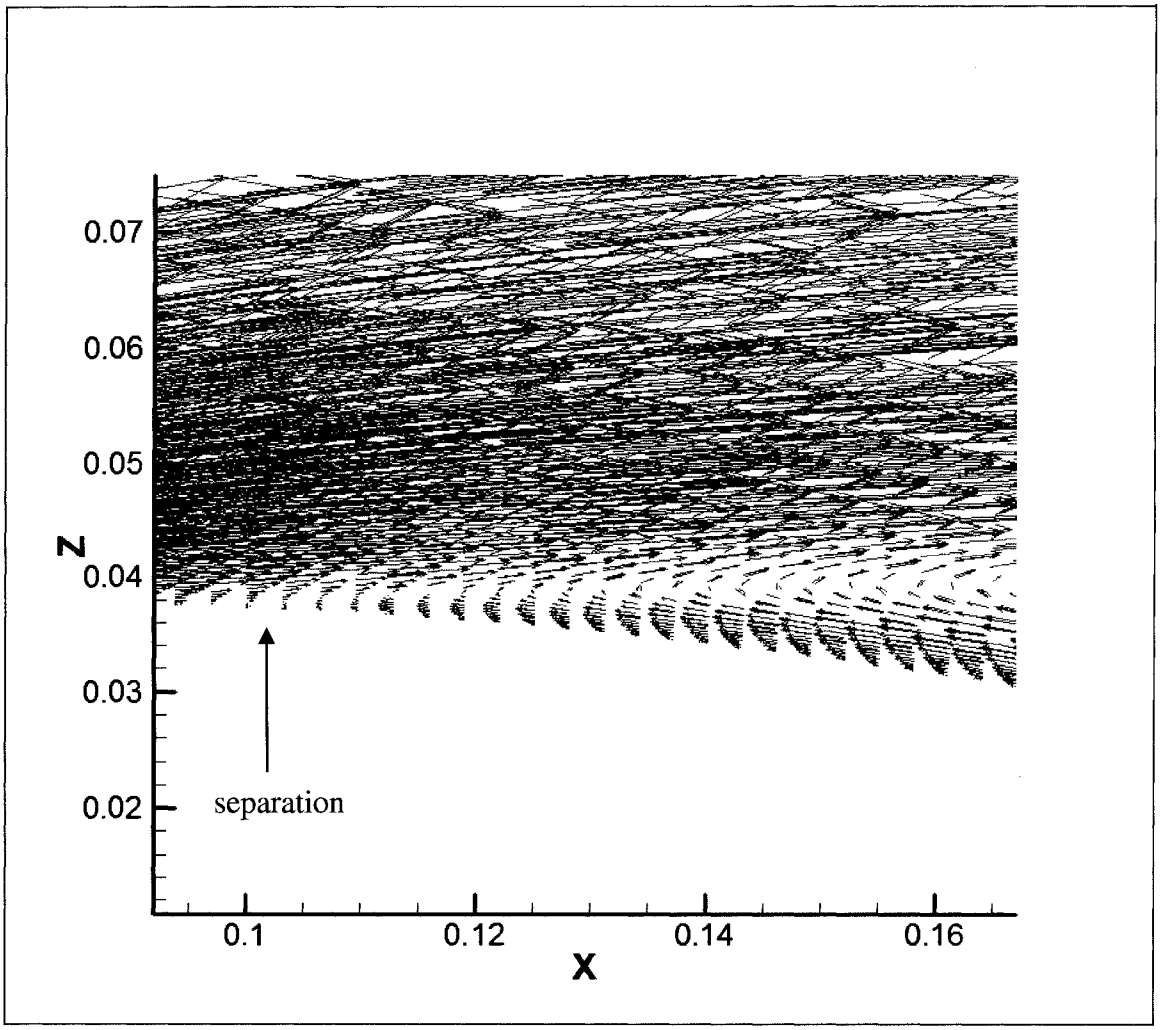


Figure 3.10: separation point at $\alpha = 5^{\circ}$

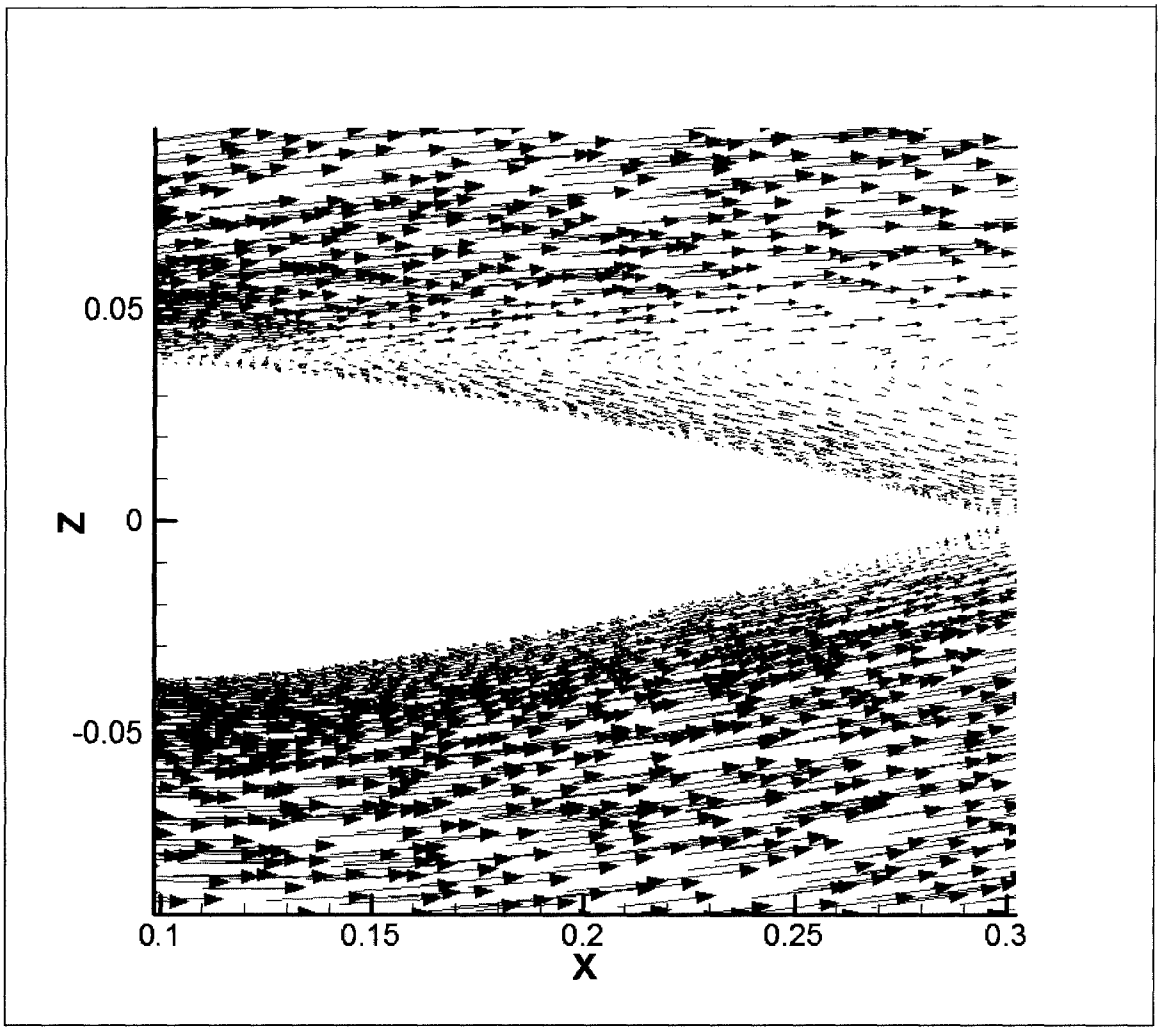


Figure 3.11: velocity vectors near the leading edge at $\alpha = 5^{\circ}$

In Figure 3.13 the pressure coefficient (C_p) is plotted against the distance from the leading to trailing edge of the airfoil at Reynolds number 100000 and $\alpha = 0^\circ$. As expected due to the symmetrical profile of the airfoil and zero incidence angle the upper and lower pressure distributions are exactly the same. The boundary layer separates at approximately x/c=0.48 while the experiment get it at x/c=0.37. As can be observed in this graph, the region downstream of the separation point has almost constant pressure. Since the boundary layer does not reattach to the airfoil's surface downstream of the separation point, constant pressure region extends up to the trailing edge. The separation point moves towards the leading edge as the attack angle is increasing. In our simulation, it reaches the x/c=0.31 at $\alpha = 5^\circ$ while the experiment data reports x/c=0.30 for the separation location at the same angle of attack. (Figure 3.14 and Figure 3.16)

As it was mentioned before the flow does not separate without self-adaptive upwinding method. This can be observed by comparing the pressure distributions at Figure 3.15. From experimental data, the pressure reaches a plateau after the separation point. This is also confirmed through the numerical simulation by Compressible LES code using self-adaptive upwinding method. But for the simulation without self-adaptive upwinding method, there is no separation captured and the pressure decreases almost linearly towards the trailing edge.

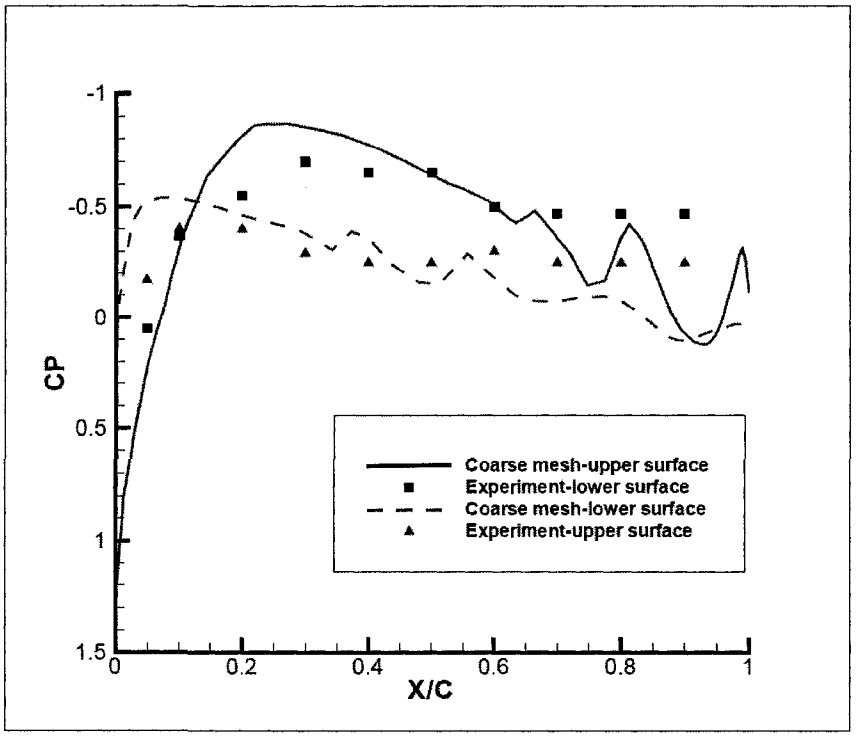


Figure 3.12: upper and lower surface pressure distributions at $\alpha = 5^{\circ}$

(coarse mesh)

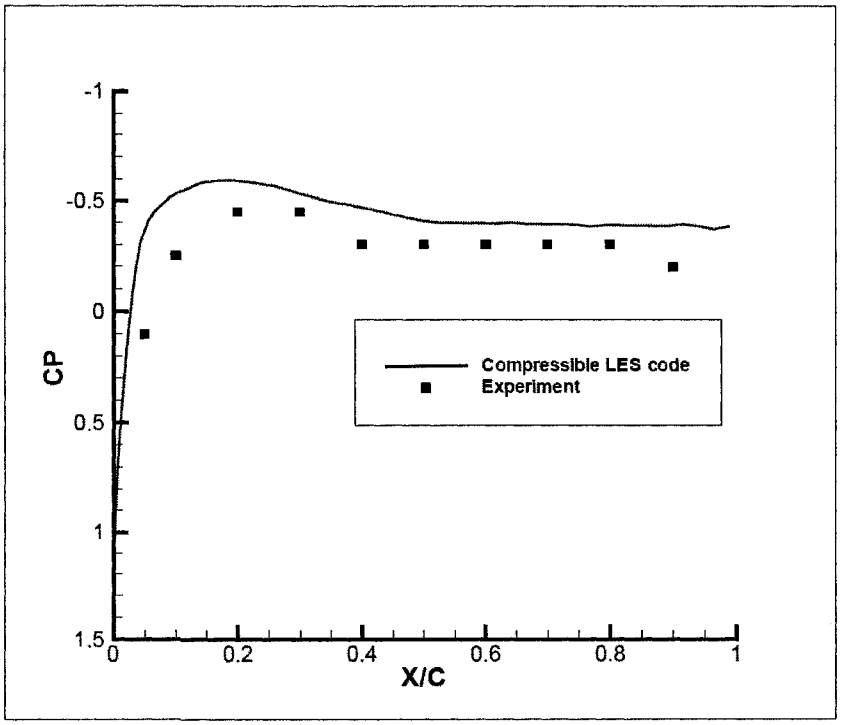


Figure 3.13: surface pressure distribution at $\alpha = 0^{\circ}$

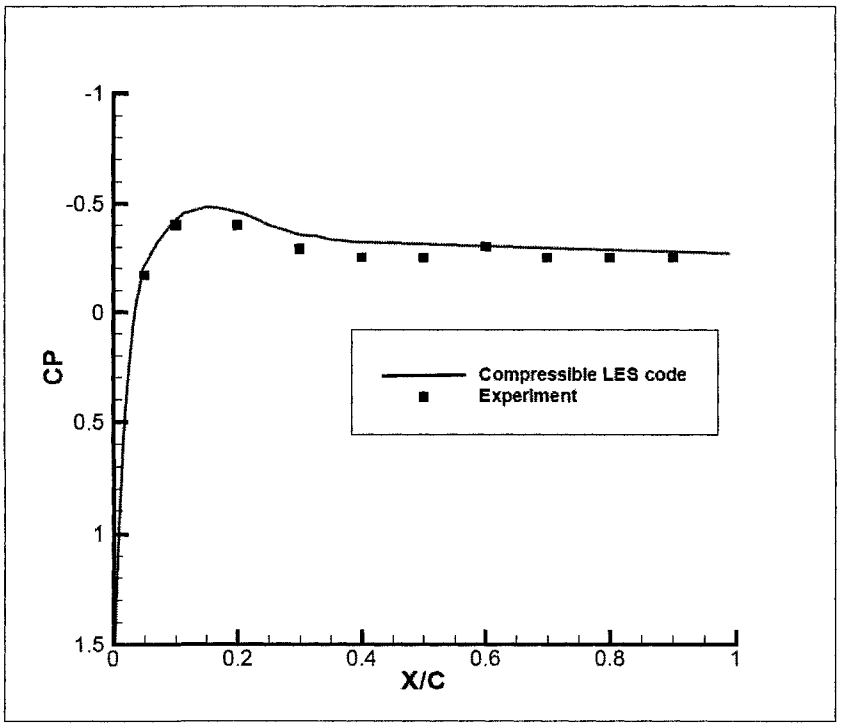


Figure 3.14: upper surface pressure distribution at $\alpha = 5^{\circ}$

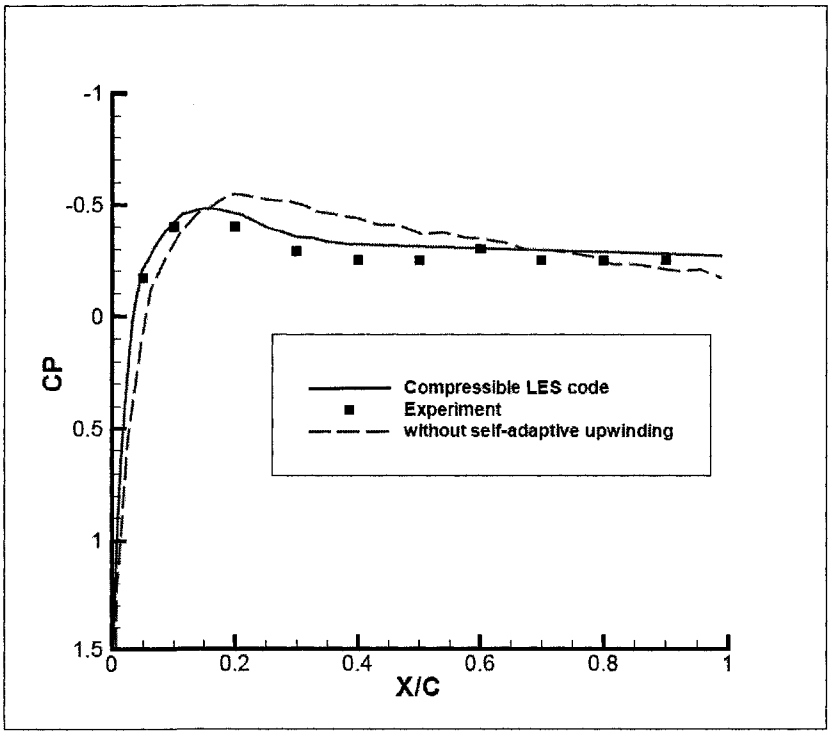


Figure 3.15: comparison of upper surface pressure distribution at $\alpha = 5^{\circ}$

(with and without self-adaptive upwinding methods)

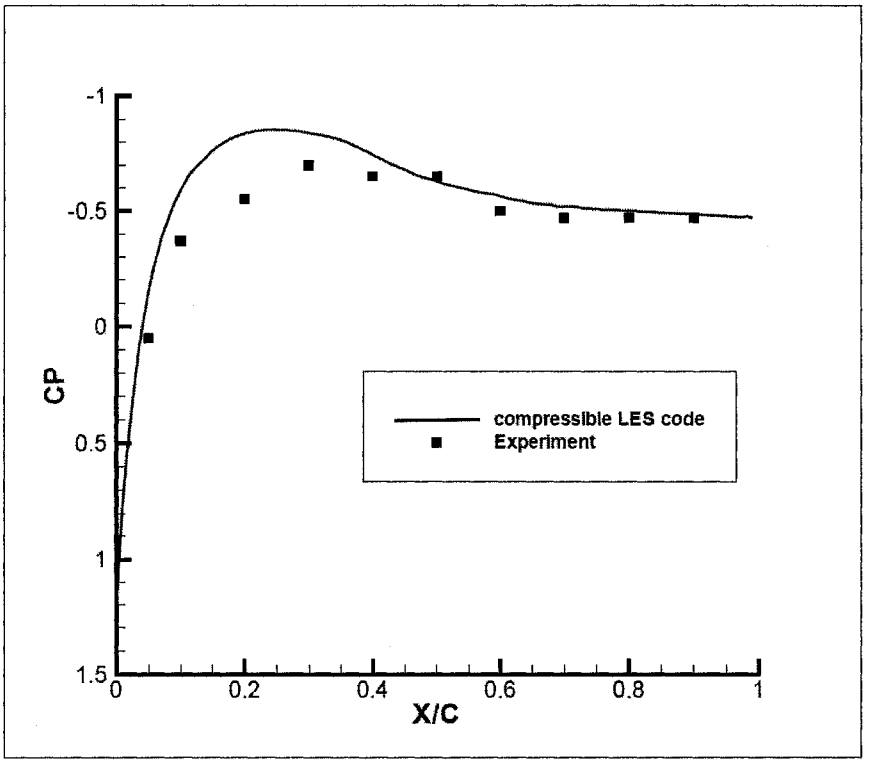


Figure 3.16: lower surface pressure distribution at $\alpha = 5^{\circ}$

3.5.1.1 Unsteady Results

One of the best features of LES method (rather than RANS) is its unsteady capability which can be seen in our simulations. The unsteady development of the separated shear layer and the periodic vortex shedding is important criteria [36]. Therefore, for better understanding of the unsteady flow in the separated region, instantaneous streamlines and velocity contours in different times are shown for one shedding cycle in Figures 3.17-25. The time step is fixed at 0.0001s. Since the flow is unsteady, the streamlines are changing with time and the vortex shedding can be observed at the trailing edge region. As can be seen through these figures, the cycle starts in Figure 3.17 at 1.0s and terminated in Figure 3.25 at 1.067s. This shows that a complete vortex shedding period takes almost 0.067 seconds for our simulation.

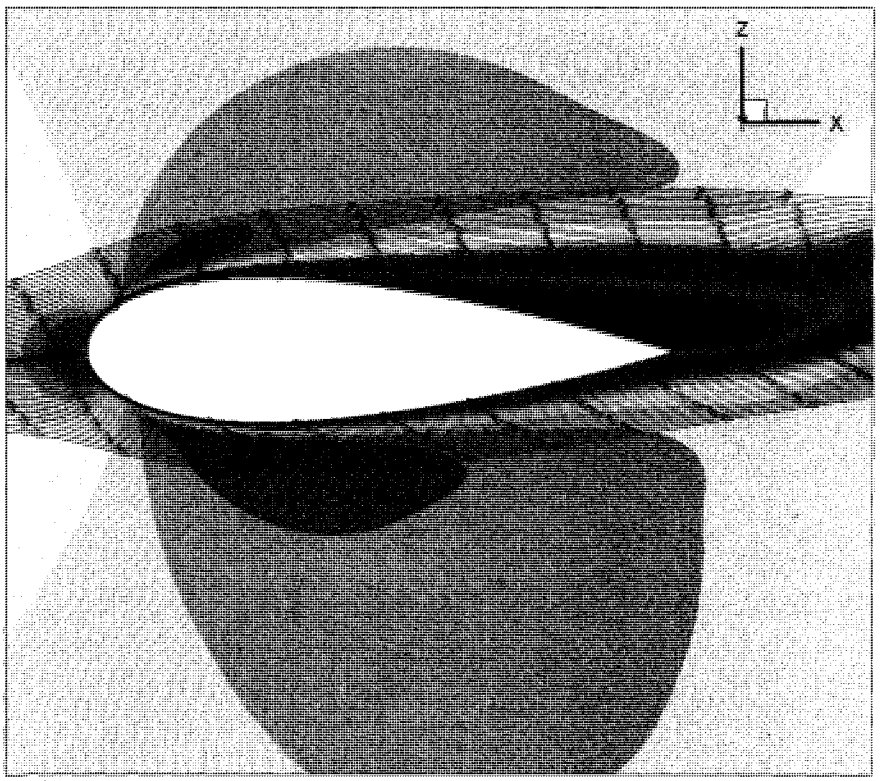


Figure 3.17: streamlines and velocity contours at time=1.0s

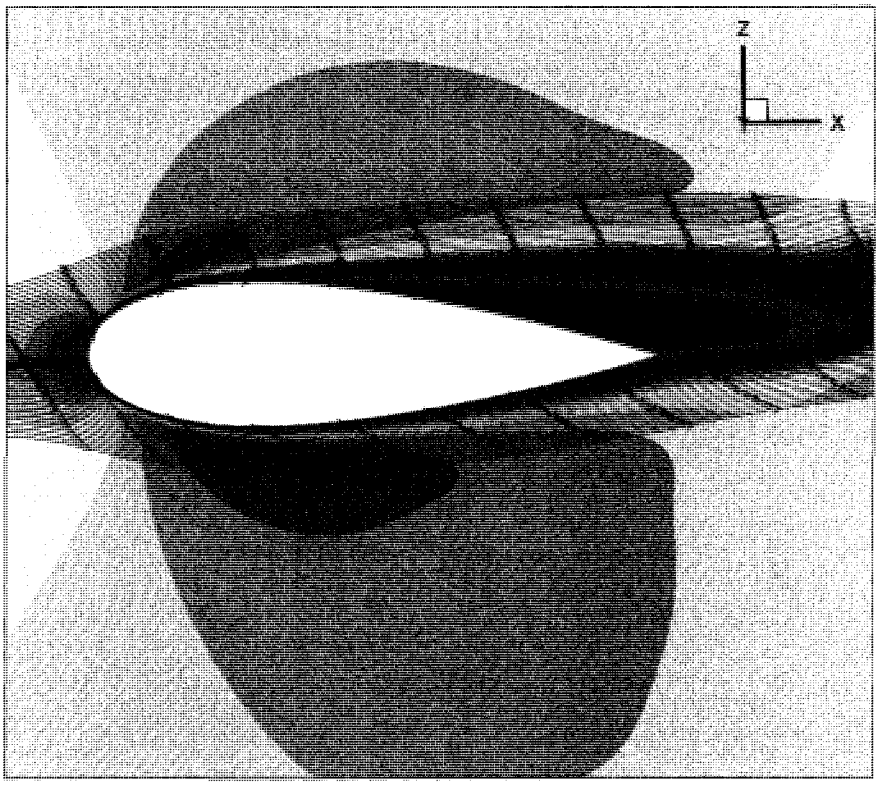


Figure 3.18: streamlines and velocity contours at time=1.007s

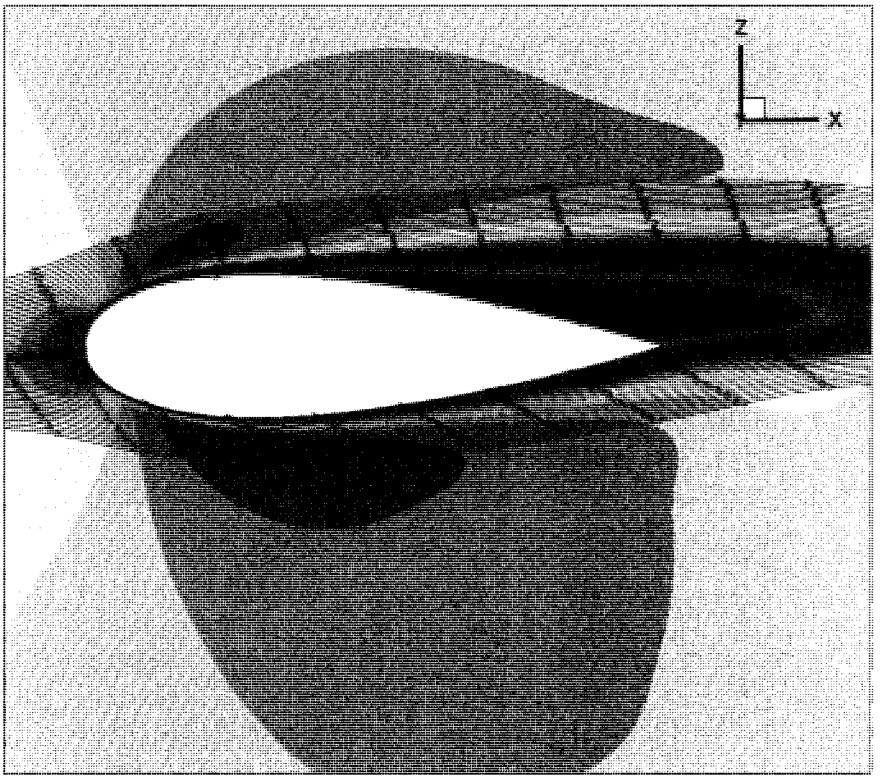


Figure 3.19: streamlines and velocity contours at time=1.015s

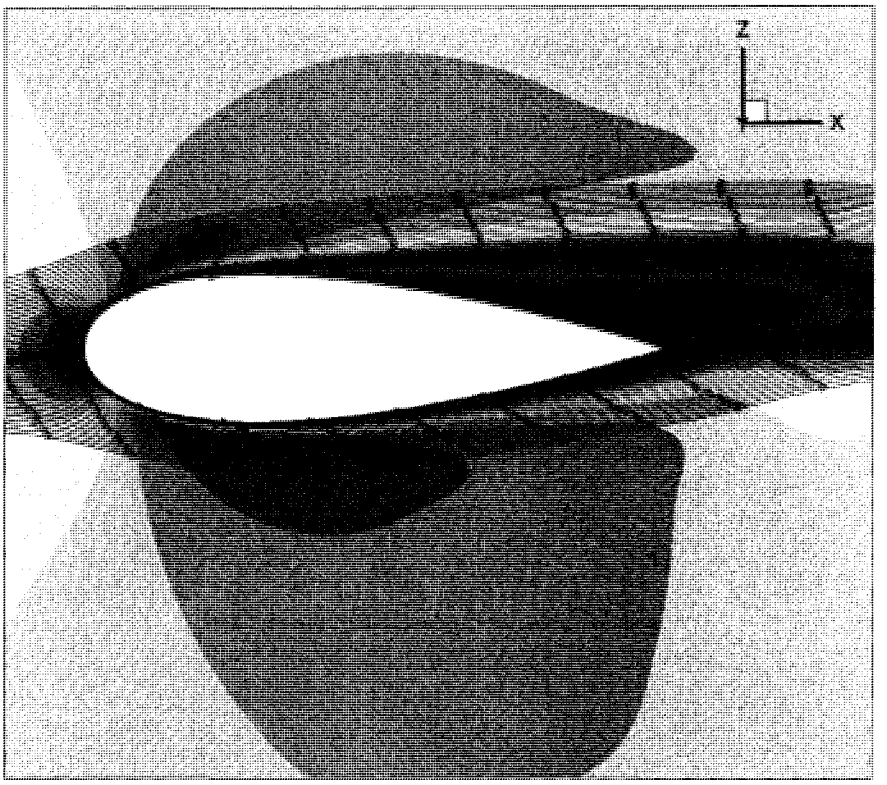


Figure 3.20: streamlines and velocity contours at time=1.023s

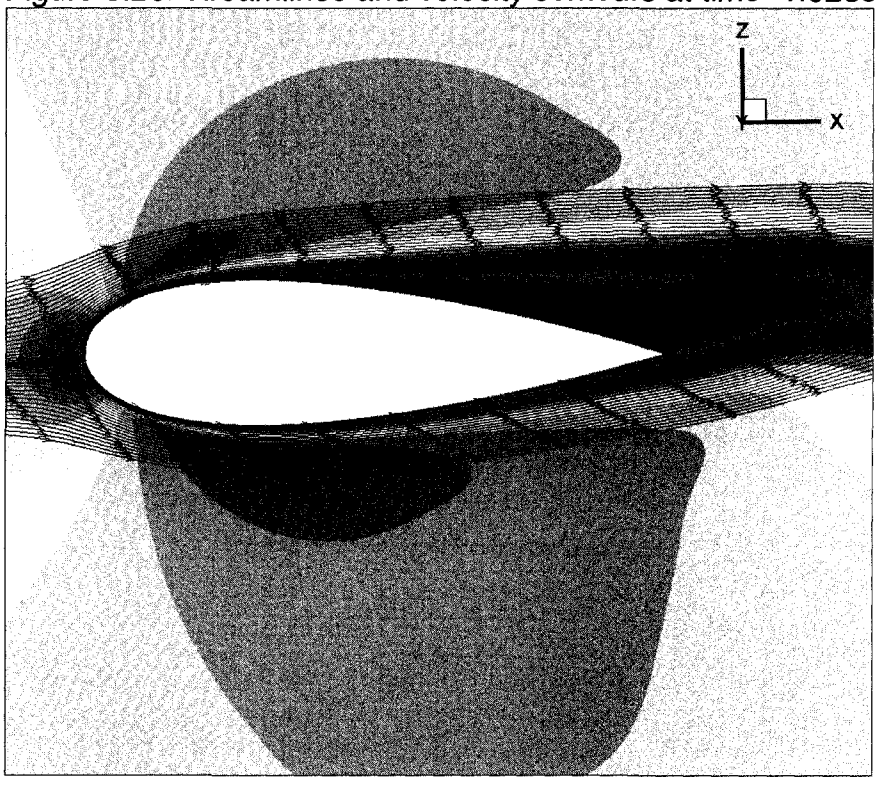


Figure 3.21: streamlines and velocity contours at time=1.033s

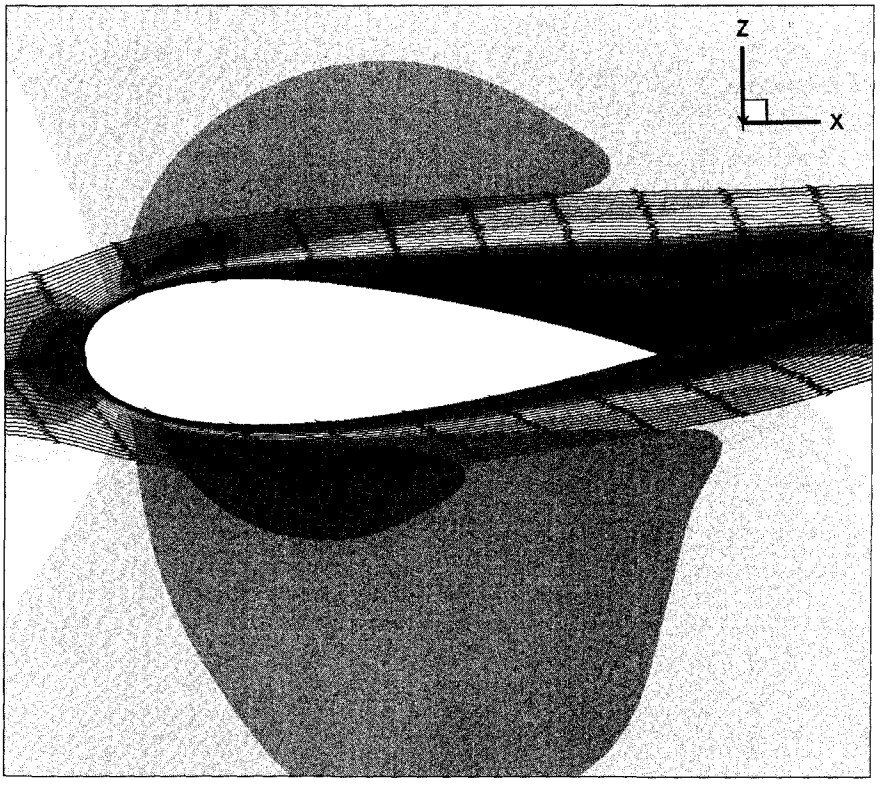


Figure 3.22: streamlines and velocity contours at time=1.041s

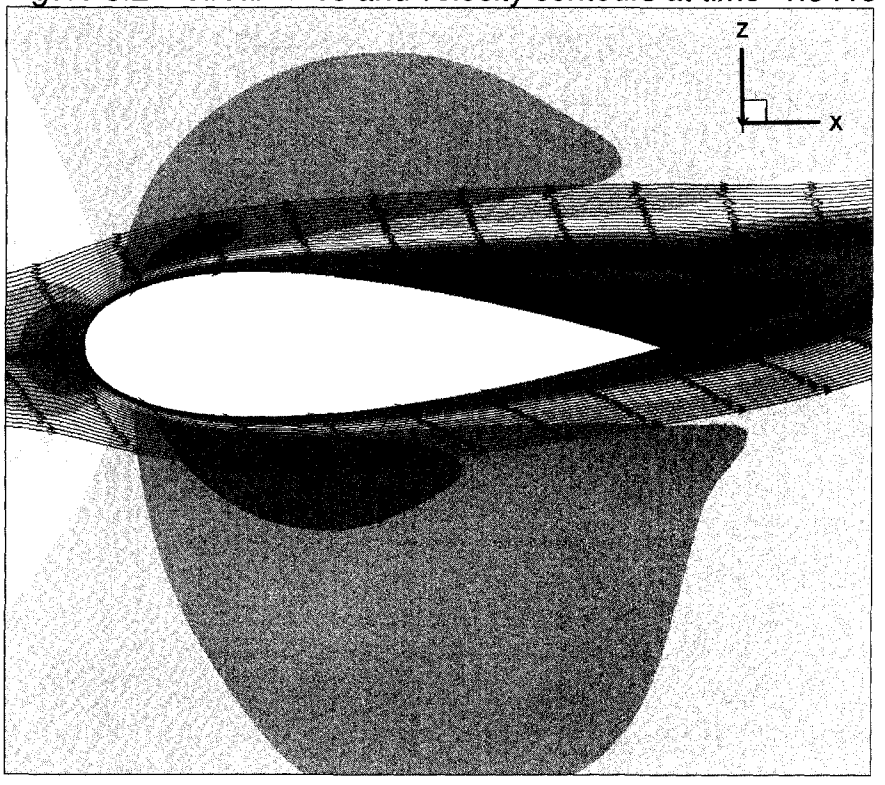


Figure 3.23: streamlines and velocity contours at time=1.049s

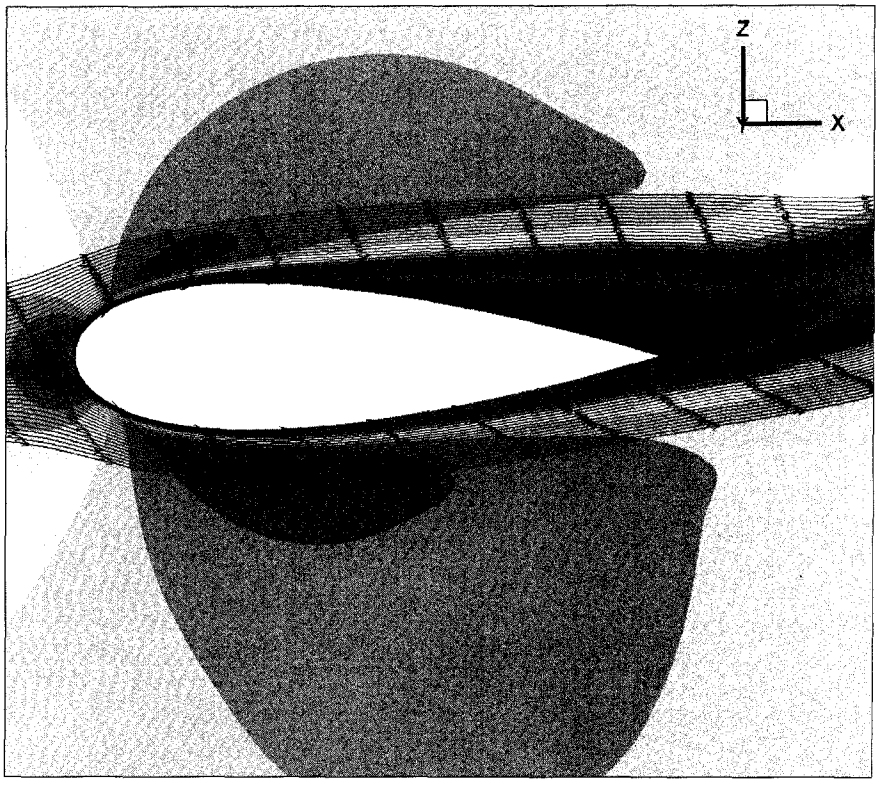


Figure 3.24: streamlines and velocity contours at time=1.059s

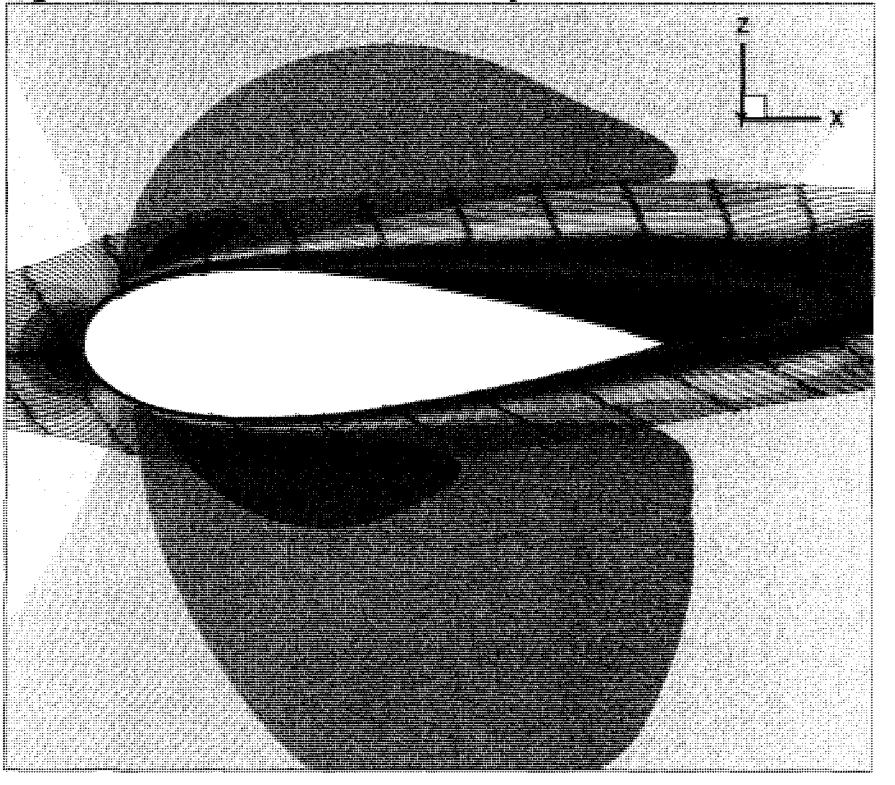


Figure 3.25: streamlines and velocity contours at time=1.067s

3.5.2 FLUENT

Results of the in-house code presented in chapter 3.5.1 are compared with a commercial CFD software to highlight the need for the present code development. FLUENT is chosen because its LES capability was upgraded in version 6.2 with several enhancements in SGS modeling and also in its numerics, such as Bounded Central Differencing (BCD) and non iterative time advancement (NITA). BCD scheme replaces the second order upwind method for discretizing the convective terms to suppress the existence of unphysical numerical wiggles by the full central differencing method and NITA algorithm to speed up LES calculations [37].

For the purpose of comparison, the same flow condition is simulated with FLUENT. For boundary conditions, outflow boundary condition is used because the details of the flow velocity and pressure are not known prior to the solution of the flow problem. Velocity inlet boundary condition is applied to define the flow velocity, along with all relevant scalar properties of the flow, at inlet.

LES involves running a transient solution from some initial condition. So as FLUENT's user guide recommended we start by running a steady state flow simulation using a RANS turbulence model (a standard $k-\varepsilon$) with a small Courant number, and increase it gradually as the iterations proceed. Then we continue running until the flow field is reasonably converged. In the next step an appropriate time step size and all the required solution parameters are set and the LES option is enabled. Afterwards, LES runs until the flow becomes

statistically steady. After LES computations, the initial statistics are zeroed out and data sampling is enabled to get statistically stable data.

The same mesh is used for the compressible in-house code was used for FLUENT simulation first; however, it could not capture the separation. Hence, another mesh with finer grids in the boundary layer is used for fluent. The mesh is fined about $\Delta y^+ \approx 0.2$. In this finer mesh the flow separates at x/c=0.42 for $\mathrm{Re}_c = 100000$ and $\alpha = 5°$. According to FLUENT's results, as can be observed in Figure 3.26, there are a few numbers of small separation bubbles (blue colors) at both upper and lower boundary layer surfaces. On the other hand, the in-house code predicts a large laminar separation bubble. Figure 3.28 and Figure 3.29 show that the surface pressure distributions are acceptable around the leading edge; but for x/c larger than 0.6 although we expect a constant static pressure as predicted by the in-house code, the pressure decreases in both upper and lower surfaces due to the small vortices in near-wall region. This is far from the experiment results.

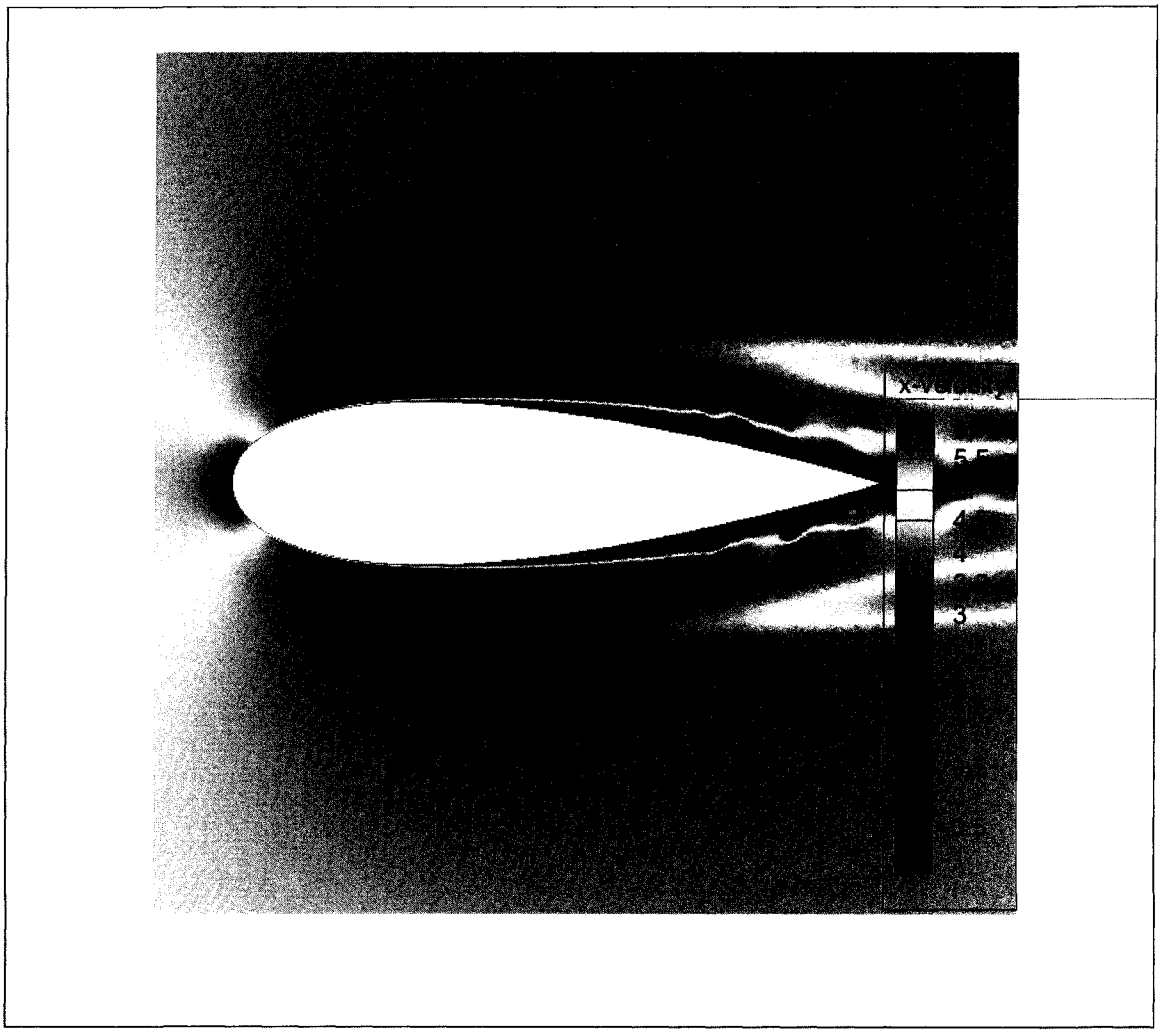


Figure 3.26: Velocity contours at $\alpha = 5^{\circ}$ (FLUENT)

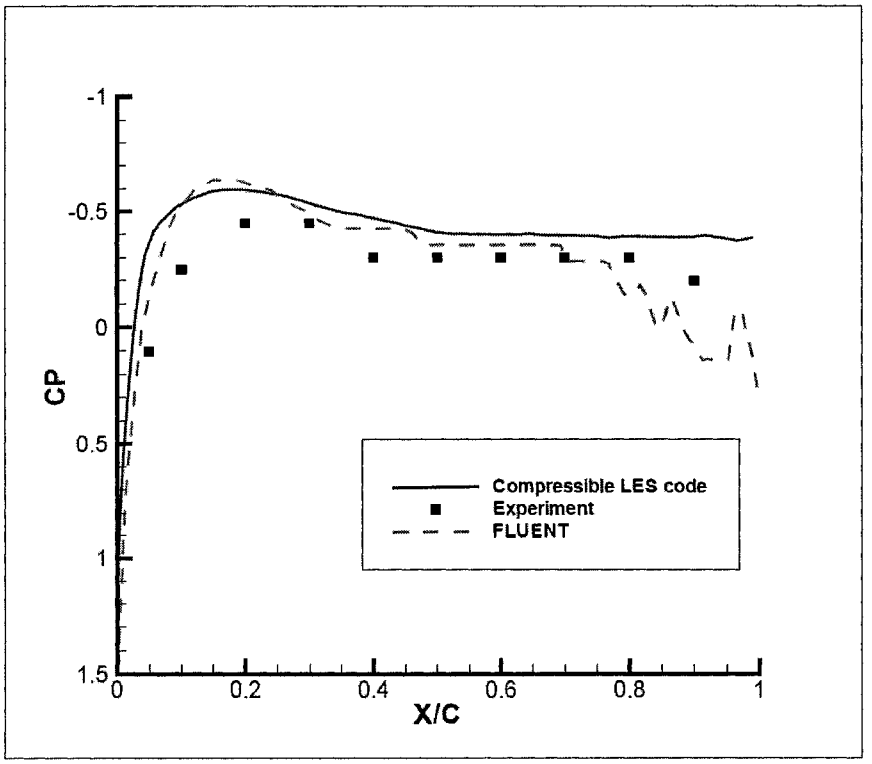


Figure 3.27: surface pressure distribution at $\alpha = 0^{\circ}$

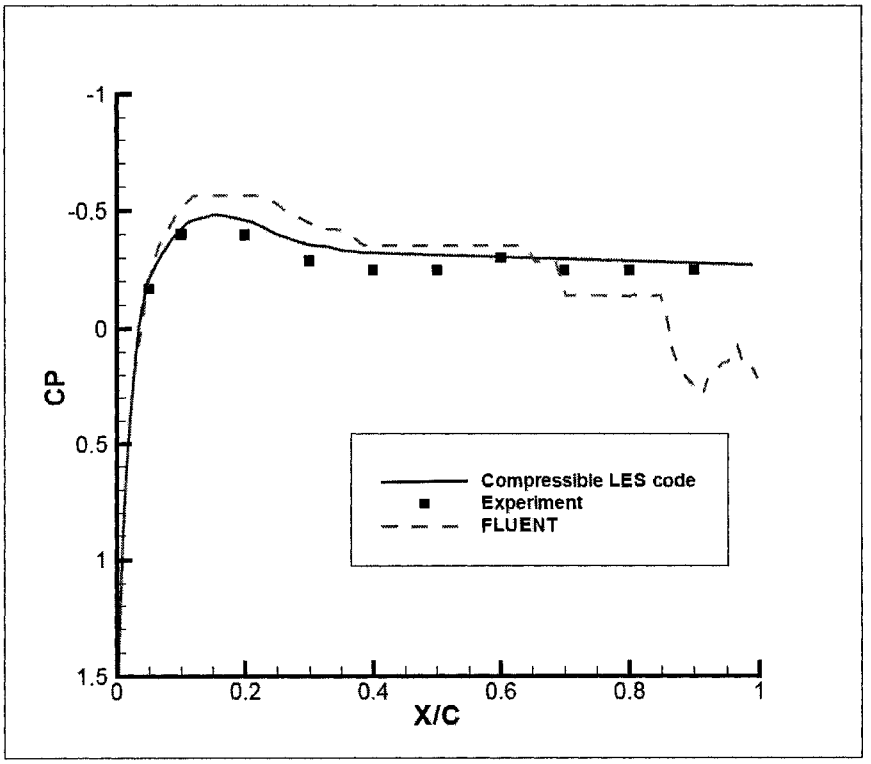


Figure 3.28: upper surface pressure distribution at $\alpha = 5^{\circ}$

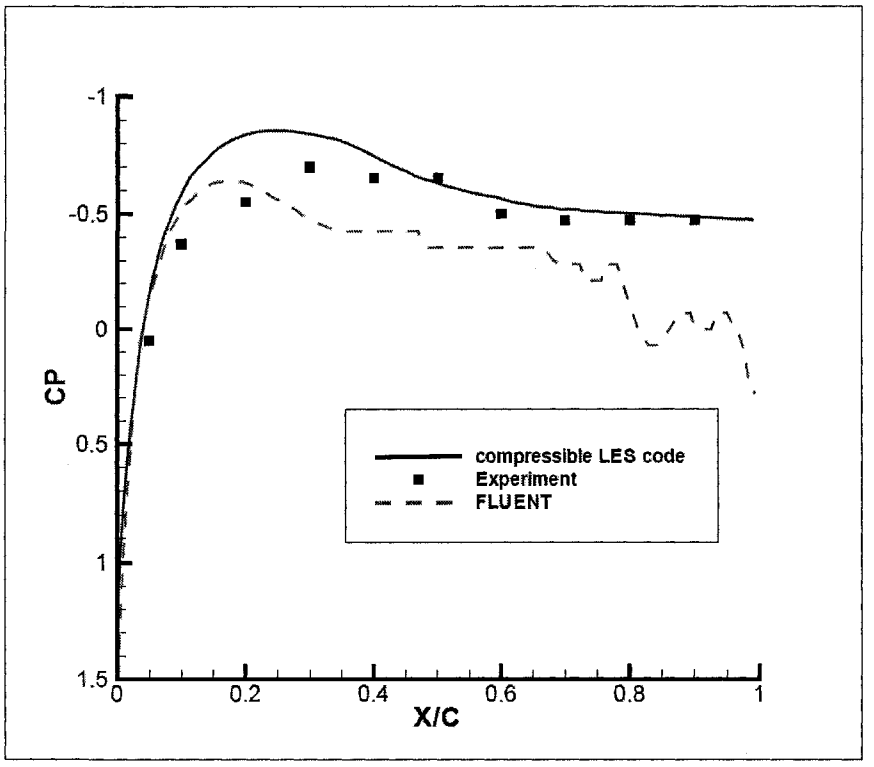


Figure 3.29: lower surface pressure distribution at $\alpha = 5^{\circ}$

3.5.3 Incompressible FEM Code

Furthermore a comparison with results from the work by [17] based on an incompressible LES Code are reported. This code uses FEM solver with a dynamic Smagorinsky model for LES. The computational domain and geometry is the same as what is used for the simulation by our code; however, it only consists of 514,800 nodes. Using multigrid solver helps to increase its convergence rate to be run on serial machine.

As shown in following figures this code is not able to predict the bump near leading edge but it yields the separation points approximately at their correct locations. Moreover, the pressure reaches a plateau, as it was expected due to the constant static pressure after separation.

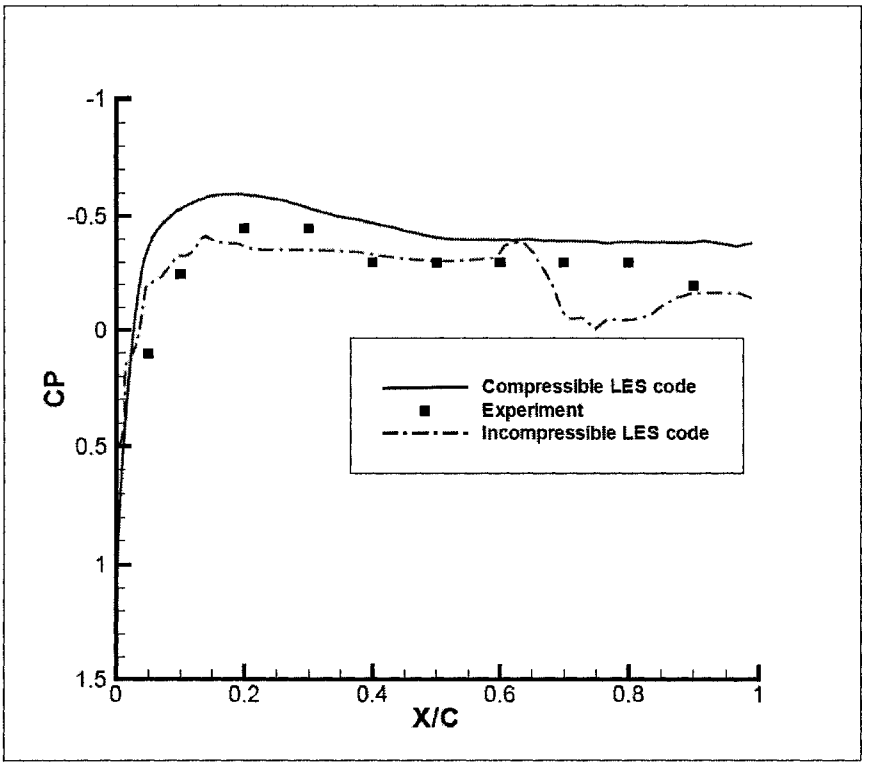


Figure 3.30: surface pressure distribution at $\alpha = 0^{\circ}$

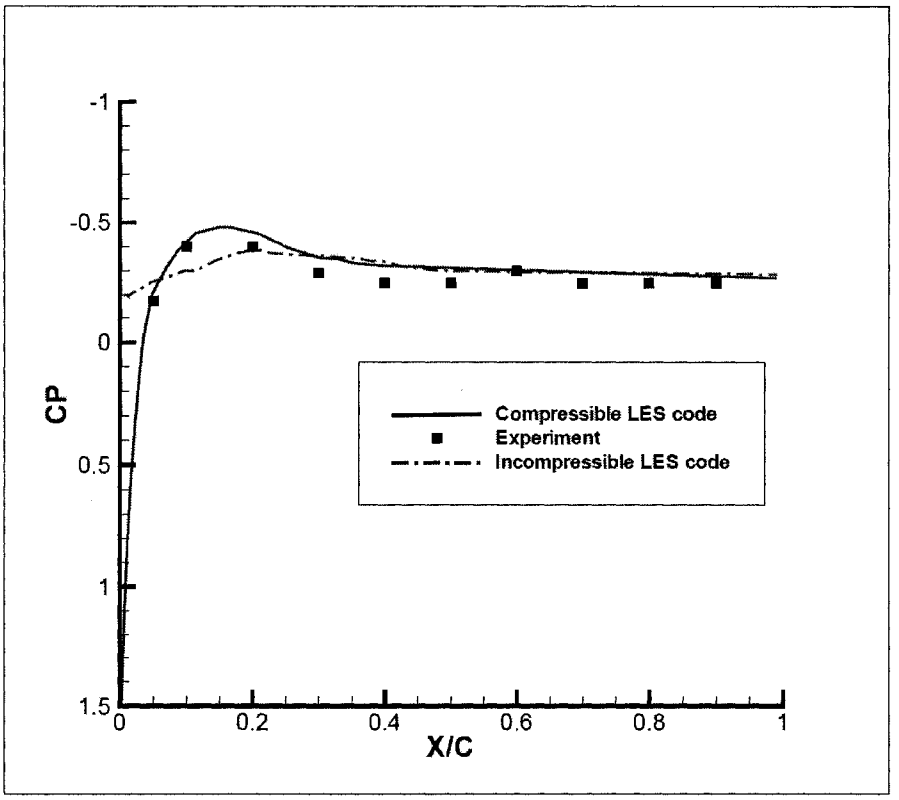


Figure 3.31: upper surface pressure distribution at $\alpha = 5^{\circ}$

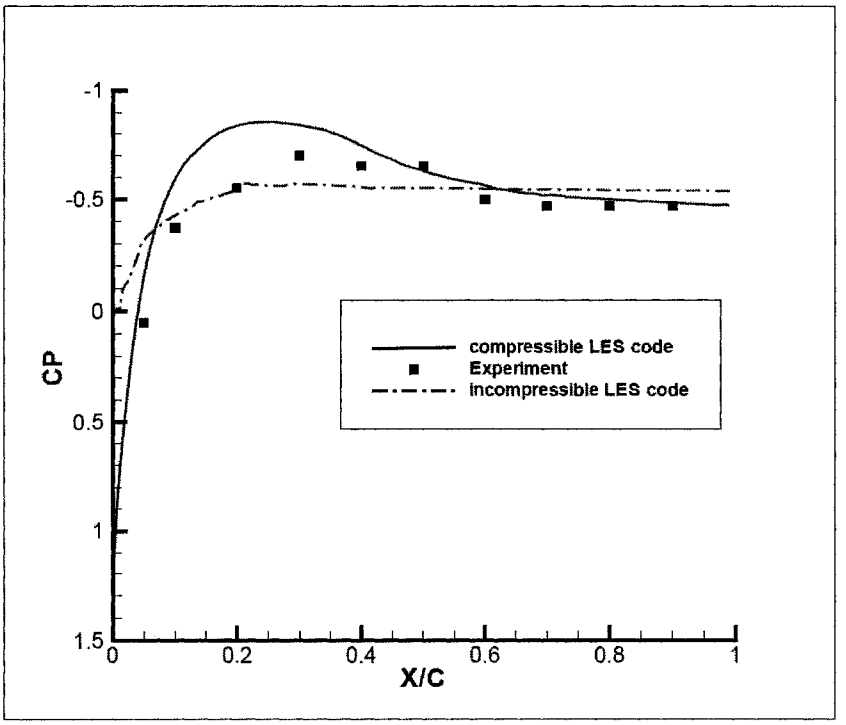


Figure 3.32: lower surface pressure distribution at $\alpha = 5^{\circ}$

3.6 Summary

Computations time, mesh size, and accuracy of CFD codes are the main factors determining the performance of a flow simulation. Therefore, in this section the three pre-discussed CFD codes (our in-house Compressible code, commercial FLUENT code, and Incompressible FEM in-house code) are compared against each others in terms of the mesh size, computational time and accuracy to capture the location of separation points.

As mentioned before, for FLUENT simulations the mesh has to be very fine in the near-wall regions in order to capture the separation bubble. So as shown in Table 3 this mesh is almost two times larger in terms of the number of nodes than our in-house code used. On the other hand the incompressible code used by Tao Xu has only half a million nodes (1/6 of compressible code and 1/11 of FLUENT) with $\Delta y^+ \approx 15$ (15 and 30 times coarser than compressible code and FLUENT in terms of the first node of the boundary layer).

	Compressible Code	FLUENT	Incompressible Code
nodes	3,000,000	5,800,000	514,800
elements	9,500,000	18,000,000	486,912
D.o.F	5x3M	5x18M	4x0.5M
Δy^+	≈ 1	≈ 0.2	≈ 15

Table 3: comparison of mesh size in different codes

In Table 4 the computational time in different codes are compared. The simulations for Compressible Code and FLUENT are performed on a 64 and 16 CPUs cluster server respectively. The Compressible Code is run for an elapsed time of 48 hours while FLUENT is run in 45 hours. The Incompressible Code is run in a Sun Microsystem Sparc workstation for a couple of days.

	Compressible Code	FLUENT	Incompressible Code
No. of CPUs	64	16	1
Elapsed time	48 hours	45 hours	4 days

Table 4: comparison of computational time in different codes

Airfoil performance depends upon the points where the flow detach from the airfoil's surface. Therefore, in order to validate the results of any numerical simulation at flow over an airfoil, they have to be compared with experimental results. Table 5 presents comparison of separation point predictions by the three pre-mentioned codes with the results obtained from the experiment on the same flow conditions at two attack angles. Our code can predict this point at $\alpha = 5^{\circ}$ very close to the experimental data with less than 3% of error while FLUENT and Incompressible codes are a bit far from the experiment. By decreasing the angle of attack to zero degree, the location of the separation points do not change in FLUENT and Incompressible Codes. While for our code this point goes to the

Trailing edge as was expected. However, there is a delayed compared to the experimental data.

	compressible code	FLUENT	Incompressible code	Exp.
$\alpha = 0^\circ$	0.48	0.43	0.35	0.37
$\alpha = 5^\circ$	0.31	0.42	0.35	0.30

Table 5: comparison of the location of separation point with different methods

In Table 6 the difference in predicting the location of separation points with experimental data are shown in percentage of error. It should be noted that the pressure transducer has an uncertainty of about 2%. Therefore, the experimental results are measured with $\pm 2\%$ of uncertainty.

	compressible code	FLUENT	Incompressible code
$\alpha = 0^\circ$	29%	16%	5%
$\alpha = 5^\circ$	3%	40%	16%

Table 6: error comparison in predicting separation point in different methods

4 Closure

4.1 Conclusion

In this research the boundary layer behavior of a NACA0025 airfoil at low Reynolds number has been simulated numerically. The most popular turbulence models to simulate such flows are RANS models that are not accurate enough to predict the flow behavior in the highly separated regions successfully. Therefore, LES must be used and this work has confirmed that LES is able to capture separated flows better [38]. Hence, LES solver is used. It is based on an existing compressible code with a 3D unstructured tetrahedral Navier-Stokes parallel solver. In addition, the flux calculation method is improved to reduce the artificial diffusion by self-adaptive upwinding technique. The artificial viscosity is reduced up to the level of flow instability. This is done because the original method is over dissipative and preventing the flow to separate over the airfoil even in LES method.

Laminar boundary layer separation occurs on the upper surface of this airfoil at $\mathrm{Re}_c = 100000$ at two angles of attack $\alpha = 0^\circ$ and $\alpha = 5^\circ$ due to the surface curvature of the airfoil. The flow fails to reattach to the airfoil surface and makes the airfoil performance deteriorated by decreasing the lift and increasing the drag.

The obtained results have been compared with wind tunnel observations by [11]. They show good agreement with the experimental data especially in terms of the surface pressure distributions.

According to the literature, prediction of separation point is not an easy task. So agreement between our numerical results and the experiment confirms that the LES code with self-adaptive upwinding method can correctly compute the flow separation. However, there are still some modifications essential to get more accurate results.

Considering pressure distribution results compared to the FLUENT software results, the Compressible in-house code is superior especially in the after separation regions despite the fact that the mesh used for FLUENT simulation is almost two times finer. This demonstrates the need to continue to develop in-house codes at least for this specific case of simulation. The license limit is another disadvantage of using FLUENT compared to in-house codes. It also good to note here that initially for the same number of the nodes the in-house code convergence rate is almost the same as FLUENT. However the Turkel, [21], precondition accelerate the simulation process and with this preconditioning the in-house LES code is approximately two times faster than the commercial FLUENT software.

Compressible LES results of this work are more accurate that the Incompressible LES results by [17] especially in the regions close to the leading edge which contains a pressure decrease on the surface of the airfoil that the latter code can not predict. This can be because of not having enough mesh

resolution in that region. However the Incompressible FEM code gives very good results considering its computational cost than two other methods used at this work. This might be due to introducing the dynamic Smagorinsky for LES model.

4.2 Future Works

The following subjects are suggested as the future works:

- Dynamic Smagorinsky model (DSM)

  DSM models are better representative of physics of the flow especially near-wall effects. In this model the constant is adjusted automatically during the LES calculation based on the resolved fields. The C_s obtained in this method varies in time and space.

- Hybrid RANS and LES

  LES is computationally expensive due to the fine mesh required. Therefore, the main disadvantage of using LES is its requirement for very fine meshes near walls. Hybrid LES-RANS was invented to solve this limitation. The idea is that the near-wall's effect should be predicted by the RANS turbulence model rather than being resolved by LES. As shown in Figure 4.1, in this method RANS (such as Spalart-Allmaras model) is used near walls and LES is applied away from walls [39]. They should match in the inner part of the logarithmic region. Grid generation is more complicated than for a simple RANS or LES case due to the RANS-LES switch [40].

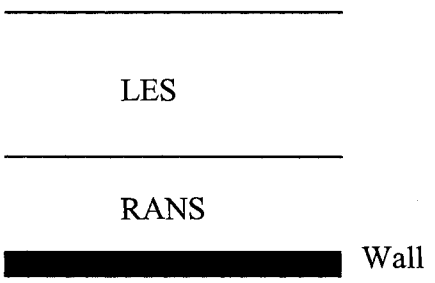


Figure 4.1: Hybrid LES-RANS

Therefore the grid resolution is not as demanding as pure LES, thereby considerably cutting down the cost of the computation.

- Improving the incoming boundary condition

  In this research the turbulence in the flow regime is only due to the transition flow occurring on the airfoil. However, the simulation can be improved considerably by adding fluctuations to the incoming flow. [41] Shows that also the results obtain with RANS methods may not be sensitive to these fluctuations at the incoming flow but for LES they can improve the results particularly in pressure distribution. Therefore, our simulation can be more accurate by introducing random fluctuations to the inlet flow.

- Near wall model

  A friction velocity is increasing proportional to Reynolds number. So as discussed in chapter 3.3 in order to keep Δy^+ around one, a finer mesh is essential. For example solving the same case for $\mathrm{Re}_c = 1\times10^6$ needs the

first node in the wall region approximately four times closer to the airfoil surface than the mesh used for $\mathrm{Re}_c = 1\times10^5$ here. So using LES method for higher Reynolds number can be impossible based on the same computer resources and it is recommended to use near wall models such as Wall functions. These models enable the user to generate a mesh with Δy^+ around 15.

References

1 Forrester T. Johnson, Edward N., Tinoco, N. Jong Yu "Thirty years of development and application of CFD at Boeing Commercial Airplanes," Computers and Fluids J. Vol 34, Issue 10, pp 1115-1151, 2005.

2 Anil W., "An introduction to computational fluid dynamics," Cambridge university press, 2005.

3 Elder, R.L., "Advanced of CFD in Fluid Machinery design," Bury St. Edmunds, England, Professional Engineering, 2003.

4 Piomelli, U., "Large Eddy Simulation: Achievements and Challenges," Progress in Aerospace Sciences, 35, 335-362,1999.

5 S.Dahlstrom and L.Daavidson, "Large eddy simulation applied to a high Reynolds flow around an airfoil close to stall," Chalmers University of Technology, Swede, 41st AIAA Aerospace sciences Meeting and Exhibit 6-9, Reno, NV, January 2003.

6 Mohamed Gad-El-Hak, "Control of low-Reynolds-number airfoils," low-Reynolds-number Aerodynamics conference Notre Dame , Indiana, USA 1989.

7 Schlichting, H., Boundary-Layer Theory," 6th edn. McGraw-Hill, 1968.

8 Carmichael, B., H., "Low Reynolds number airfoil survey," NASA Contractor Report, No. 165803, Vol. I, 1981.

9 Zifeng Yang, Fred L. Haan, Hu Hui,"An Experimental Investigation on the Flow Separation on a Low-Reynolds-Number Airfoil," AIAA-2007-0275, 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007.

10 P. B .S. Lissaman, "Low-Reynolds-number airfoils," Annu. Rev. Fluid Mech. 15, 223, 1982.

11 Yarusevych, S., "Investigation of airfoil boundary layer turbulent wake development at low Reynolds numbers," PhD thesis, University of Toronto, 2006.

12 Watmuff J. H., "Evolution of a wave packet into vortex loops in a laminar separation bubble," *J. Fluid Mech.*, Vol. 397, pp. 119-169, 1999.

13 J. C. M. Lin and L. L. Pauley, "Low-Reynolds-number separation on an airfoil," AIAA J. 34, 1570, 1996.

14 Yang, Z., and Voke, P.R., "Large eddy simulation of boundary layer separation and transition at a change of surface curvature," J. Fluid Mech. pp. 305-333, 2001.

15 Th. Lutz ,W. Wurz , and S. Wagner, Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Stuttgart, Germany.

16 Yarusevych, S., Sullivan, P. E., and Kawall, J. G., "Coherent structures in an airfoil boundary layer and wake at low Reynolds numbers," Physics of Fluids, 18, 044101, 2006.

17 Tao Xu, "A numerical investigation of airfoil boundary layer using large eddy simulations," PhD thesis, University of Toronto, 2007.

18 Volpe G. Performance of compressible flow codes at low Mach numbers, AIAA J. 31(1):49, 1993.

19 Turkel. E., Fiterman, A. and van Leer, B. "Preconditioning methods for low-speed flows," NASA Contractor Report No. 201605, 1996.

20 Turkel. E., Fiterman, A. and van Leer, B. "Preconditioning and the limit to the incompressible flow equations," Computing the future: Frontiers of computational

fluid dynamics 1994, D.A.Caughey and M.M.Hafez. eds., Wiley Publishing , pp. 215-234, 1994.

21 Turkel, E., "Preconditioning techniques in computational fluid dynamics," Annu. Rev. Fluid Mech., 31:385-416, 1999.

22 Bienkiewicz, B., "Toward the practical use of CFD technology for numerical prediction of wind loading," IWEF Workshop on CFD for Pred. Wind Load. on Buildings and Structures, Tokyo, Japan, Sep., 1995.

23 L. Hallo, C. Le Ribault, M. Buffat, An Implicit Mixed Finite-Volume-Finite-Element Method for Solving 3D Turbulent Compressible Flows, Intl. J. for Numerical Methods in Fluids, vol.25, 1241-1261, 1997.

24 Cadiou, A., *NadiaLES : Manuel Theorique*, Note Technique CODICIEL-LMFA N :2003-01.

25 Karimi M., "LARGE EDDY SIMULATION (LES) OF TURBULENT FLOWS IN GAS TURBINE," MASc thesis, Concordia University, 2007.

26 Tajallipour N., "Large eddy simulations of turbulent flows," PhD proposal, Concordia University, 2007.

27 Bui, T. T., "A Parallel, Finite-Volume Algorithm for Large Eddy Simulation of Turbulent Flows", Dryden Flight Research Center, California, NASA TM-206570, Jan. 1999.

28 K.W. Morton and D.F. Mayers, *Numerical Solution of Partial Differential Equations, An Introduction*. Cambridge University Press, 2005.

29 G. Volpe. Performance of compressible ow codes at low Mach number. AIAA Journal, 31:49 56,1993.

30 E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. J. Comp. Phys., 72:77 298, 1987.

31 J. Smagorinsky. General circulation experiments with the primitive equations, the basic experiment. Mon. Weather Rev., 92, 1963.

32 D. K. Lilly. A Proposed Modification of the Germano Subgrid-Scale Closure Model. Physics of Fluids, 4:633-635, 1992.

33 FLUENT 6.2 user's guide, January 2005.

34 Piomelli, U., and Chasnov, J. R., Large-eddy simulation: Theory and application," Transition and turbulence modeling, pp. 269-336, Dordrecht, Kluwer Academic Publishers, 1996.

35 Hirsch, C., "Numerical computation of internal and external flows," Vol.2. A Wiley-Intersciense Publication, 1990.

36 Roberts, K. S., and Yaras, M. I., "Large-Eddy simulation of transition in a separation bubble," ASME, 232, Vol. 128, 2006.

37 Kim., S.E, "LES is more", FLUENT news, spring 2005.

38 Kim,S.E.:Large Eddy Simulation using unstructured meshes and synamic subgrid-scale turbulence models. AIAA paper no. 2004-2548, 2004.

39 L.Davidson and S.-H.Peng "A Hybrid LES-RANS Model Based on a One-Equation SGS Model and a Two-Equation k-omega Model", *The Second International Symp. on Turbulence and Shear Flow Phenomena*, Eds: E. Lindborg, A. Johansson, J. Eaton, J. Humphrey, N. Kasagi, M. Leschziner, M. Sommerfeld, Vol. 2, pp. 175-180, Stockholm, 2001.

40 Spalart, P.R. "Comments on the feasibility of LES for wing and on a hybrid RANS/LES approach". *1st ASOSR CONERFENCE on DNS/LES. Arlington, TX.* August 1997.

41 Matsuura, K., Kato, C., "*Large Eddy Simulation of Compressible Transitional Flowsin a Low-Pressure Turbine Cascade*, "AIAA Journal, 45(2): 442–457, 2007.