12-2008

Modeling Fluid Structure Interaction over a Flexible Fin Attached to a NACA0012 Airfoil

Srinivasa Ravindra Pantula
Western Michigan University

MODELING FLUID STRUCTURE INTERACTION OVER A FLEXIBLE FIN ATTACHED TO A NACA0012 AIRFOIL

by

Srinivasa Ravindra Pantula

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Mechanical and Aeronautical Engineering
Advisor: Dr. William Liou

Western Michigan University
Kalamazoo, Michigan
December 2008

ACKNOWLEDGMENTS

I would like to thank my Professor Dr. William Liou for his for his brilliant guidance, great innovative ideas and constant help through out of my thesis. I would like to thank each of my committee members, Dr. Tianshu Liu and Dr. Iskender Sahin of Western Michigan University and Dr. Ray Hixon of University of Toledo for their interest, enthusiasm and wonderful guidance through out my thesis.

This research project was supported by the Air Force Office of Scientific Research (AFOSR). Thanks to AFOSR for their support.

I am very fortunate to be working at the computation engineering physics laboratory of the Mechanical and Aeronautical engineering department along side Dr. Meng Huang Lu, Dr. Yang Yang and soon to be doctor yongqing Peng. I am thankful to Dr. Meng Huang Lu for providing me with his fluid solver and helping me in understanding the code. I am also thankful to Dr. Yang Yang for proving me with a structural solver and helping me modify his code according to my needs. Yongqing Peng and I shared some graduate classes together and it was great to study and

work with him. I would like to thank him for the wonderful discussions we had day in and day out. I would also like to thank Dr. Javier Montefort for his help through out my thesis in providing me with the processed experimental data when ever needed. I am very thankful for the long discussions and those free invitations to view experiments. I would also like to thank Dr. Parviz Merathi and all other people at the Fluids lab. My thanks to all of my other friends at Western Michigan University and especially the volleyball gang for making my life at WMU very enjoyable.

Finally I would like to thank my mother, Rajeswari Pantula and my sister, Usha Pantula for their constant support, encouragement and care all these years. Without them any of my accomplishments are not possible. I would like to dedicate this work to the memory of my late father Chandra Sekhar Rao Pantula.

Srinivasa Ravindra Pantula

TABLE OF CONTENTS

Table of Contents—continued

CHAPTER

Table of Contents—continued

CHAPTER

LIST OF FIGURES

Table of Contents—continued

Table of Contents—continued

Table of Contents—continued

CHAPTER I

INTRODUCTION

1.1 Motivation

Natural flyers like birds and insects are speculated to utilize their wing flexibility, particularly thin flexible fins, for more efficient flight and effective flow control in different flight regimes. The flexible fins seem to play an important role in flapping flights where highly unsteady aerodynamics is nonlinearly coupled with the deforming wing. Bird flight inspired many early aviation pioneers like Lilienthal and Wright brothers who used flexible thin wings for flight control. In Lippisch's [1] first (and last) successful man-powered ornithopter test in 1929, the dramatic effect of quasi-flexible trailing edges (made up of bamboo pieces attached to the rigid wing near the tips) on improving flapping propulsion over that of a rigid flapping wing was observed. However, as remarkable advances were made in fixed-wing aircraft, the potential benefit of wing flexibility had been largely ignored, partially because flexibility has usually been considered a dangerous factor and the associated unsteady aerodynamics is too complicated to handle. Recently, the potential advantage

of wing flexibility has been re-discovered, and relevant research has been supported by NASA's Morphing Program (McGowan 2001) [2].

Separation flow control is of massive importance to the performance of air, land, sea vehicles, turbo machinery and diffusers. Generally, it is desired to postpone flow separation so that form drag is reduced, stall is delayed, lift is enhanced, and pressure recovery is improved. Therefore, considerable research efforts have been made over years for separation control (or stall control) by using various techniques like synthetic jets (Glezer & Amitay [3], Mittal et al. [4]), vortex generators (Gad-el-Hak [5]), passive and active blowing (Gad-el-Hak [5]), local suction (Atik et al. [6]), flapping wing (Jones et al. [7]), and oscillating camber (Munday & Jacob [8]). In this thesis the concept of separation control (or stall control) for a post-stall NACA0012 airfoil is done by using a flexible fin to passively manipulate the interactions of the organized vortical structures in the separation region.

1.2 Basic design

The basic design is illustrated below in the figure 1.1. A thin flexible fin is attached to the upper surface of a NACA0012 airfoil. The oscillation of the membrane,

induced by the separated wake from the post-stall airfoil interacts passively with the flow field to alter the global aerodynamic properties of the NACA0012 airfoil. The oscillations allied with the shape deformations change the overall pressure distribution on the fin, which in turn affect the fin dynamics. Thus this mutual effect of inertial forces and elastic forces can be considered through fluid-structure interactions (FSI). Hence in this thesis a computational fluid dynamic solver is combined with a computational structural dynamics solver in order to model these fluid structure interactions around the thin flexible fin attached to the upper surface of a NACA0012 airfoil that passively manipulates the flow field in fully separated flows.

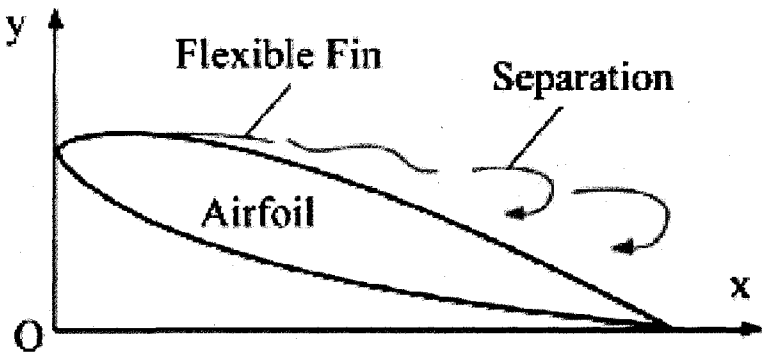


Figure 1.1: Flexible fin attached to a NACA0012 airfoil

1.3 Fluid structure interactions

There are two major numerical techniques to compute the solution of fluid structure interaction problems. They can be classified as monolithic methods and partition/segregated methods. In monolithic methods the complete system of non linear equations for the fluid and structure are coupled and integrated into one system and solved at their common interface [9, 10]. This procedure leads to a single matrix containing all equations and couplings [11]. This matrix might be large and ill-conditioned [11] and there is a chance for numerical difficulties in convergence of the solution. This could be a major problem when dealing with large geometries. On the other hand monolithic methods are considered to be more robust of the two numerical techniques. The second technique, segregated method, is the widely used method by commercial software packages where different software and different meshes are employed by the fluid and structural problems. In this method both the fluid as well as structural field are defined separately and solved and the interface conditions from the structure and the fluid are applied as boundary conditions at different times. These methods are very popular because the individual codes can be modified accordingly depending on the complexity of the problem. The

methodology presented in this thesis comes under the second category.

Coupling the CFD and CSD solvers

From a physical view point the fluid structure interaction problem is a combination of two problems, flow field and the structural field. These problems use different numerical procedures to compute the solution on different domains and meshes. In addition to using different meshes there is a need for data transfer (pressure from fluid to structure and displacement, velocity from structure to fluid) across the interface. The transfer of data which are the boundary conditions is a very important feature of fluid structure interactions. There are basically two different ways to transfer the data.

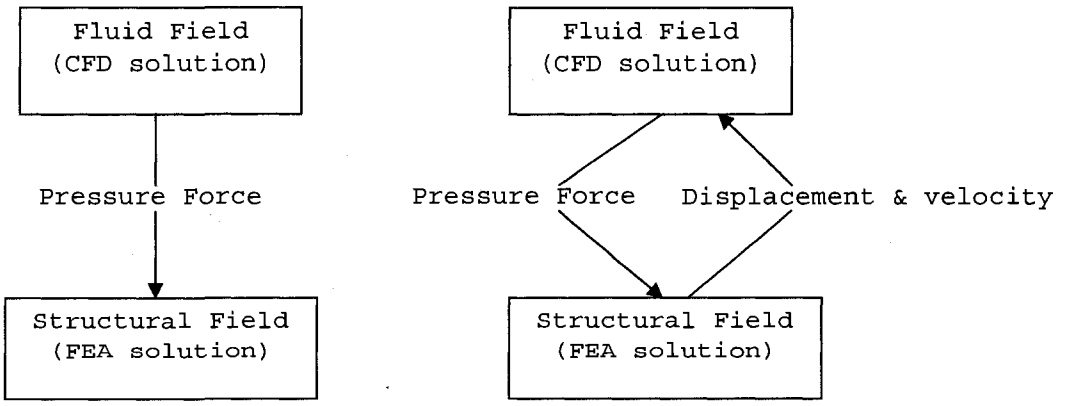


Figure 1.2: One-way coupling vs. two-way coupling

The easiest is one-way coupling. In this procedure the forces are transferred only one way from the fluid (CFD) solution to the structure (e.g. transferring static pressure loads from the fluid solution onto a structural model). The underlying assumption here is that the deformation of the solid is so small that it doesn't affect the overall fluid flow solution. The second coupling method is the widely used two-way coupling procedure for most problems involving large displacements. This is required when the fluid forces cause a significant oscillation of the structure. The results are mapped from the first solution to the second and back from the second to the first. Either the structural solution or the fluid solution takes the lead and at significant intervals the solution is mapped to and fro from fluid/structural to structural/fluid. For example when the total pressure force causes the structure to deflect, the solution is mapped from fluid to structure and in return the displacement (or position) and velocity of the structure is transferred back to the fluid domain.

Immersed boundary method

In the present thesis the term fluid structure interaction is considered as interaction of forces

(pressure) and the corresponding movement of fin (momentum interaction) rather than thermal interaction. Hence coupling the computational fluid dynamics (CFD) solver with the computational structural dynamics (CSD) solver provides an effective tool for calculating the fin dynamics. There are a number of factors that need to be handled in order to couple the CFD solver with the CSD solver. Other than handling the different spatial as well as temporal characteristics of each solver difficulty arises because of the moving interface (i.e. fin) present in the domain. These dynamically moving boundary problems are amongst the most demanding problems in contemporary computational fluid dynamics. The major complexity arises from the fact that generally all the fluid dynamic domains are described in Eulerian frame of reference. This method is suitable and works well when the boundary location doesn't change with respect to time. This becomes a problem when the boundary location changes with time. There are different techniques that have been proposed to account for this time dependent movement of the boundary, such as the overset grid method, dynamic meshing and coordinate transformations which can be applied to body conformal grids. In order to account for the motion of the boundary these grids need to be regenerated at every time step and also the old solution

needs to be projected onto the new grid. For problems involving large deformations and or large motions these grid regeneration methods are not only complex and time consuming but can have adverse effects on the simplicity, accuracy and efficiency of the solver. A study done by Liou & Pantula [13] where a dynamically moving flat plate was simulated using the commercial codes Fluent and Ansys CFX supports the above argument. Hence there is a need for developing a cost efficient numerical procedure that can deal with large boundary motions. An alternate to the boundary conforming methods which do not require the regeneration of grid at every time step is the widely used non conforming boundary fitted technique called immersed boundary method (IBM). In this method the boundary location need not be dependent on the mesh layout. The basic idea of the immersed boundary technique lies on the definition of the solid boundaries which may be static or dynamic. The immersed boundary technique mimics a solid body by means of suitably defined body forces applied to the discretized set of the momentum equations. These body forces impose a kinematic condition such that the velocity at each node point is coupled to the interpolated fluid velocity. The body force-field f is imposed so that a desired velocity distribution **V** can be specified on an immersed boundary [14]. This means

that we just add the body force **f** to the Navier-stokes equations and solve for **u** from the equation

$$\frac{\partial u}{\partial t}+\nabla(uu)=-\nabla P+\nu\nabla(\nabla u)+f\ ,\quad \nabla.u=0 \qquad 1.1$$

The main advantage of this approach is that f can be prescribed on a regular mesh so that the accuracy and efficiency of the solution procedure on simple grids are maintained. Another advantage of these formulations is the simplification of grid generation, especially in the case of moving boundaries where the need for regeneration or deformation of the grid is eliminated.

1.4 Numerical representation of FSI problem

In figure 1.3 the numerical representation of procedure that is adopted and used in the present thesis is represented. The first step in the numerical procedure is to calculate the fluid solution on a eulerian grid. Then the total pressure force acting is exported as a condition for the structural field to obtain the velocity and the displacement. Then the location of the body is traced in a lagrangian fashion and appropriate virtual forces at interface locations are formulated and smoothly transferred onto the eulerian grid nodes using the immersed boundary technique. Then the solution is

advanced to next time step. The most convenient way for transferring the data between the CFD mesh and the FEA mesh would be if the nodes are concurrent. The pressure forces as well as the velocities and displacements need to be interpolated to the nearest node points if they are not coincident.

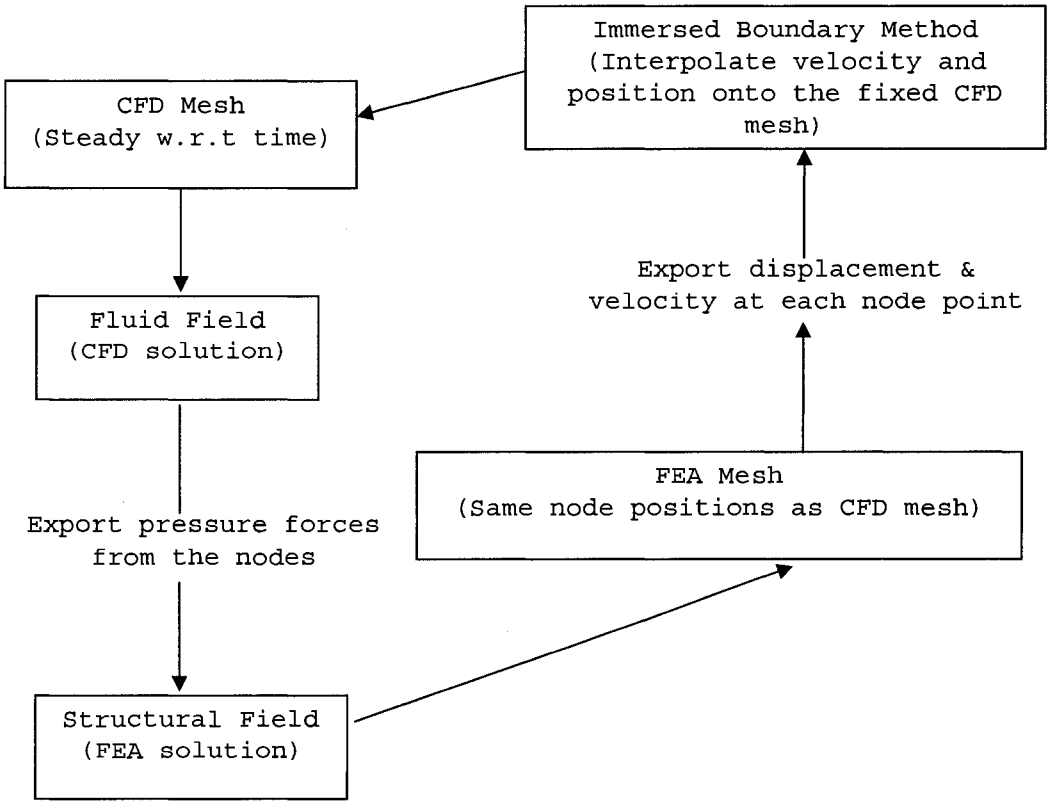


Figure 1.3: Numerical representation of the FSI problem

The low Reynolds number K-ε model of Launder and Sharma [15] is also incorporated into the CFD code to compute the turbulent scales in the flow. One major factor that needs to be addressed here when using the

immersed boundary technique is that the thickness of the fin is very small. The thickness of fin in the experiments is considered to be 0.0001 m but in the numerical procedure we consider the fin to be infinitesimally thin (i.e. thickness=0). Different interpolation techniques have been formulated to take care of this problem. These techniques are discussed in detail in the chapter IV of the thesis.

1.5 Literature review of immersed boundary methods

Fluid structure interaction modeling has a wide range of industrial applications ranging from blood vessels, functioning heart value, elastic arteries to flexible tubes to parachutes to tents bridges to flapping flag to swimming fish. The first non-conformal boundary method to conduct fluid structure interactions was proposed by Peskin [16] where a fluid structure interaction over a cardiovascular circulation was studied assuming a low Reynolds number (Re) flow. In this calculation the boundary was modeled as a set of elements linked by springs. The body forces were easily computed using Hooke's law. Numerical difficulties arose when this method was applied to solid/rigid boundaries because the assumption of elements being elastic becomes unacceptable. Goldstein [17] applied it to solid

boundaries by introducing a feedback forcing approach that asymptotically enforces the desired boundary conditions on the solid. These come under the category of continuous forcing functions according to immersed boundary methods review by Mittal and Iaccarino [18]. The forcing is incorporated into continuous equations even before discretization. This feedback mechanism of Goldstein combined with the spectral method was used to simulate two-dimensional flow around a circular cylinder, as well as three-dimensional plane [17] and ribbed-turbulent channel flow [18]. These results were in good agreement with the reference data. The major drawback of this procedure is in order to calculate the feedback forcing two empirical constants related to flow frequencies were introduced. These two free constants need to be tuned according to the frequency of the flow. The equations became stiffer when the magnitude of these constants was high. This induced spurious oscillations and numerical instability which restricted the computational time step size. Saiki and Biringen [20] used the same forcing to compute the flow around fixed and rotating circular cylinders using a fourth-order central finite-difference scheme. Their results showed that the use of finite difference scheme eliminated the occurrence of spurious oscillations of flow at the

boundary. Even though this approach was successful at very low Reynolds number flows they cannot be used for high/moderate Reynolds number flows.

Recently, Mohd-Yusof [21] and Fadlun [14] proposed a direct forcing embedded boundary formulation and showed that discrete time forcing is much more accurate and efficient compared to the feedback forcing. There are no empirical methods in this approach and the derivation of the forcing is explicit making the derivation of **f** flow independent. The methodology of Fadlun [14] was to introduce the forcing at the first grid point external to the immersed boundary using a velocity that, in a linear approximation, this point would have if the boundary had a desired velocity. Fadlun [14] reconstructed the solution at the fluid zones closest to the solid zone where as Kim [22], Majumdar [23] reconstructed the solution at ghost cells, which are solid zones closest to the fluid zone. Both the direct and feedback forcing procedures are discussed in detail in the later chapters.

1.6 Manuscript organization

The second chapter of the manuscript/thesis begins by describing the Navier-Stokes fluid dynamic solver. The governing equations, spatial and temporal discretizations of the momentum and continuity, pressure velocity

couplings are discussed.

Chapter three describes the subdivision finite element structural dynamics solver. The kinematics of deformation, finite element discretization and unstructured mesh generation techniques are presented in this chapter.

Chapter four describes the immersed boundary method in detail. The application to moving boundaries as well as coupling between the CFD and the CSD solvers is discussed for the couple of FSI simulations conducted in this thesis. The velocity forcing applied on the fin is also discussed.

In Chapter five the CFD solver and the CSD solver are validated. The turbulent flow over a NACA0012 airfoil is simulated at Re=170,000 and compared with the published results. Different grid independence tests were done to validate the accuracy of the CFD solver. Then the CSD solver is validated by comparing unsteady simply supported beam solution with analytical results

In Chapter six, different validations for the immersed boundary technique are presented. The major aim is to model the fluid structure interaction of a flexible fin attached to the upper surface of a post stall NACA0012 airfoil. The fin is considered infinitesimally thin. The Reynolds number of the flow is fixed at 63000.

In order for the immersed boundary technique to be applicable to the present design the following validations are imperative.

- Steady state immersed boundary (laminar flow)
    a) Laminar flow over a NACA0012 airfoil at Reynolds number of 500.
    b) Laminar flow over a circular cylinder at Reynolds number of 40.
- Unsteady immersed boundary (laminar flow)
    a) Unsteady laminar flow over a circular cylinder at Reynolds number of 200.
- Unsteady flow over a infinitesimally thin flat plate.
    a) Unsteady flow over a flat plate at AOA of 30.
- Steady state turbulent immersed boundary
    a) Turbulent flow over a NACA0012 airfoil on a rectangular grid using immersed boundary technique.
- Moving boundary solution using immersed boundary method.
    a) Unsteady laminar flow around a flexible flat plate validated against a boundary fitted data of Pantula and Liou [13].

In Chapter seven combined fluid structure interaction modeling around a passively flapping flat plate at angle of attack of five is described.

In Chapter eight the fluid structure interaction around a flexible fin attached to a NACA0012 airfoil is described. Firstly the experimental set up is described followed by the numerical procedure used to model the fluid structure interactions around the fin.

Finally the conclusions and future directions are presented in chapter nine.

CHAPTER II

THE NAVIER STOKES FINITE DIFFERENCE FLUID DYNAMIC SOLVER

In this chapter the details of the fluid dynamic part of the coupled fluid structure interaction solver is discussed. The basic solver in curvilinear coordinates is discussed in detail. The chapter starts with transforming the coordinate system from Cartesian to curvilinear coordinates followed by spatial and temporal discretizations.

The numerical method is based on partial transformation approach, where the appropriate forms of the incompressible governing Navier-Stokes equations expressed in curvilinear coordinates, with velocity components expressed in Cartesian coordinates. The basic advantage of utilizing general curvilinear coordinates comes from the fact that the numerical fluxes can easily be estimated for non-orthogonal grids. The Navier-Stokes equations 2.1 to 2.4 are discretized in space, on a non-staggered mesh, using second order finite difference scheme for the pressure gradient and viscous terms, and second order upwind finite differencing for the convective terms. The upwind differencing of the convective terms eliminates the need for adding

artificial dissipation terms, to the right hand side of the momentum equations, to stabilize the numerical algorithm. This is due to the fact that a fixed amount of dissipation is inherent in the upwind differencing. The finite difference schemes employed for the pressure gradient, which pressure located on mesh points, and viscous terms, which calculating velocity components on half mesh points, are two-point central finite differencing, and for the convective terms are three-point one-sided finite differencing. The pressure-velocity equation is solved using the alternate-direction-implicit (ADI) approximate factorization method. A four-stage Runge-Kutta method is also used to advance the discrete equations in time.

2.1 Transforming Cartesian to curvilinear coordinates

The Navier-Stokes equations in Cartesian coordinates can be written as

$$\frac{\partial \rho}{\partial t}+\frac{\partial}{\partial x}\left(\rho U_x\right)+\frac{\partial}{\partial y}\left(\rho U_y\right)+\frac{\partial}{\partial z}\left(\rho U_z\right)=0 \qquad 2.1$$

$$\rho\left(\frac{\partial U_x}{\partial t}+U_x\frac{\partial U_x}{\partial x}+U_y\frac{\partial U_x}{\partial y}+U_z\frac{\partial U_x}{\partial z}\right)=\left[\frac{\partial}{\partial x}\tau_{xx}+\frac{\partial}{\partial y}\tau_{yx}+\frac{\partial}{\partial z}\tau_{zx}\right]-\frac{\partial p}{\partial x} \qquad 2.2$$
$$+\rho g_x$$

$$\rho\left(\frac{\partial U_y}{\partial t}+U_x\frac{\partial U_y}{\partial x}+U_y\frac{\partial U_y}{\partial y}+U_z\frac{\partial U_y}{\partial z}\right)=\left[\frac{\partial}{\partial x}\tau_{xy}+\frac{\partial}{\partial y}\tau_{yy}+\frac{\partial}{\partial z}\tau_{zy}\right]-\frac{\partial p}{\partial y}+\rho g_y \qquad 2.3$$

$$\rho\left(\frac{\partial U_z}{\partial t}+U_x\frac{\partial U_z}{\partial x}+U_y\frac{\partial U_z}{\partial y}+U_z\frac{\partial U_z}{\partial z}\right)=\left[\frac{\partial}{\partial x}\tau_{xz}+\frac{\partial}{\partial y}\tau_{yz}+\frac{\partial}{\partial z}\tau_{zz}\right]-\frac{\partial p}{\partial z}+\rho g_z \qquad 2.4$$

In equation 2.1 if the flow is incompressible the derivative of the density following the fluid material [the term in brackets] is zero. In many flows of interest the fluid behaves as a Newtonian fluid in which viscous stress can be related to the fluid motion by constitutive relation of the form

$$T_{ij}=2\mu\left[s_{ij}-\frac{1}{3}s_{kk}\delta_{ij}\right] \qquad 2.5$$

Where s_{ij} is the instantaneous strain tensor defined by

$$s_{ij}=\frac{1}{2}\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right] \qquad 2.6$$

From its definition, $s_{kk}=\frac{\partial u_k}{\partial x_k}$. If the flow is incompressible $s_{kk}=0$ and the Newtonian constitutive equation reduces to $T_{ij}=2\mu s_{ij}$. Incorporating these changes into equation 2.1 to 2.4 the equations are transformed as

$$\frac{\partial U_x}{\partial x}+\frac{\partial U_y}{\partial y}+\frac{\partial U_z}{\partial z}=0 \qquad 2.7$$

$$\rho\left(\frac{\partial U_x}{\partial t}+U_x\frac{\partial U_x}{\partial x}+U_y\frac{\partial U_x}{\partial y}+U_z\frac{\partial U_x}{\partial z}\right)=\mu\left[\frac{\partial^2 U_x}{\partial x^2}+\frac{\partial^2 U_x}{\partial y^2}+\frac{\partial^2 U_x}{\partial z^2}\right]-\frac{\partial p}{\partial x} \qquad 2.8$$

$$\rho\left(\frac{\partial U_y}{\partial t}+U_x\frac{\partial U_y}{\partial x}+U_y\frac{\partial U_y}{\partial y}+U_z\frac{\partial U_y}{\partial z}\right)=\mu\left[\frac{\partial^2 U_y}{\partial x^2}+\frac{\partial^2 U_y}{\partial y^2}+\frac{\partial^2 U_y}{\partial z^2}\right]-\frac{\partial p}{\partial y} \quad 2.9$$

$$\rho\left(\frac{\partial U_z}{\partial t}+U_x\frac{\partial U_z}{\partial x}+U_y\frac{\partial U_z}{\partial y}+U_z\frac{\partial U_z}{\partial z}\right)=\mu\left[\frac{\partial^2 U_z}{\partial x^2}+\frac{\partial^2 U_z}{\partial y^2}+\frac{\partial^2 U_z}{\partial z^2}\right]-\frac{\partial p}{\partial z} \quad 2.10$$

Transforming the continuity equation

Let's assume that there is a unique, single-valued relationship between the generalized coordinates and the physical coordinates, which can be written as

$$\xi=\xi(x,y,z),\quad \eta=\eta(x,y,z),\quad \zeta=\zeta(x,y,z) \quad 2.11$$

Introducing

$$\frac{\partial}{\partial x}=\frac{\partial\xi}{\partial x}\frac{\partial}{\partial\xi}+\frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta}+\frac{\partial\zeta}{\partial x}\frac{\partial}{\partial\zeta} \quad 2.12$$

$$\frac{\partial}{\partial y}=\frac{\partial\xi}{\partial y}\frac{\partial}{\partial\xi}+\frac{\partial\eta}{\partial y}\frac{\partial}{\partial\eta}+\frac{\partial\zeta}{\partial y}\frac{\partial}{\partial\zeta} \quad 2.13$$

$$\frac{\partial}{\partial z}=\frac{\partial\xi}{\partial z}\frac{\partial}{\partial\xi}+\frac{\partial\eta}{\partial z}\frac{\partial}{\partial\eta}+\frac{\partial\zeta}{\partial z}\frac{\partial}{\partial\zeta} \quad 2.14$$

And the contravariant velocity,

$$V_\xi=\frac{\partial\xi}{\partial x}U_x+\frac{\partial\xi}{\partial y}U_y+\frac{\partial\xi}{\partial z}U_z \quad 2.15$$

$$V_\eta=\frac{\partial\eta}{\partial x}U_x+\frac{\partial\eta}{\partial y}U_y+\frac{\partial\eta}{\partial z}U_z \quad 2.16$$

$$V_\zeta = \frac{\partial \zeta}{\partial x} U_x + \frac{\partial \zeta}{\partial y} U_y + \frac{\partial \zeta}{\partial z} U_z \qquad 2.17$$

Substituting equations 2.12 to 2.14 into equation 2.7, we have,

$$\frac{\partial \xi}{\partial x}\frac{\partial U_x}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial U_x}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial U_x}{\partial \zeta} + \frac{\partial \xi}{\partial y}\frac{\partial U_y}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial U_y}{\partial \eta} + \frac{\partial \zeta}{\partial y}\frac{\partial U_y}{\partial \zeta} + \frac{\partial \xi}{\partial z}\frac{\partial U_z}{\partial \xi} + \frac{\partial \eta}{\partial z}\frac{\partial U_z}{\partial \eta} + \frac{\partial \zeta}{\partial z}\frac{\partial U_z}{\partial \zeta} = 0 \qquad 2.18$$

Where,

$$\frac{\frac{\partial \xi}{\partial x}\frac{\partial U_x}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial x}U_x}{J}\right) - U_x \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial x}}{J}\right) \qquad 2.19$$

$$\rightarrow$$

$$\frac{\frac{\partial \xi}{\partial x}\frac{\partial U_x}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial x}U_x}{J}\right) - U_x \frac{\partial}{\partial \xi}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right)$$

$$\frac{\frac{\partial \eta}{\partial x}\frac{\partial U_x}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial x}U_x}{J}\right) - U_x \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial x}}{J}\right) \qquad 2.20$$

$$\rightarrow$$

$$\frac{\frac{\partial \eta}{\partial x}\frac{\partial U_x}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial x}U_x}{J}\right) - U_x \frac{\partial}{\partial \eta}\left(\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta}\right)$$

$$\frac{\frac{\partial \zeta}{\partial x}\frac{\partial U_x}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial x}U_x}{J}\right) - U_x\frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial x}}{J}\right) \qquad 2.21$$

$$\rightarrow$$

$$\frac{\frac{\partial \zeta}{\partial x}\frac{\partial U_x}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial x}U_x}{J}\right) - U_x\frac{\partial}{\partial \zeta}\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right)$$

$$\frac{\frac{\partial \xi}{\partial y}\frac{\partial U_y}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial y}}{J}\right) \qquad 2.22$$

$$\rightarrow$$

$$\frac{\frac{\partial \xi}{\partial y}\frac{\partial U_y}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \xi}\left(\frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta}\right)$$

$$\frac{\frac{\partial \eta}{\partial y}\frac{\partial U_y}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial y}}{J}\right) \qquad 2.23$$

$$\rightarrow$$

$$\frac{\frac{\partial \eta}{\partial y}\frac{\partial U_y}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi}\right)$$

$$\frac{\frac{\partial \zeta}{\partial y}\frac{\partial U_y}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial y}}{J}\right) \qquad 2.24$$

$$\rightarrow$$

$$\frac{\frac{\partial \zeta}{\partial y}\frac{\partial U_y}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial y}U_y}{J}\right) - U_y\frac{\partial}{\partial \zeta}\left(\frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta}\right)$$

$$\frac{\frac{\partial \xi}{\partial z}\frac{\partial U_z}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial z}}{J}\right) \qquad 2.25$$

$\rightarrow$

$$\frac{\frac{\partial \xi}{\partial z}\frac{\partial U_z}{\partial \xi}}{J} = \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \xi}\left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta}\right)$$

$$\frac{\frac{\partial \eta}{\partial z}\frac{\partial U_z}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial z}}{J}\right) \qquad 2.26$$

$\rightarrow$

$$\frac{\frac{\partial \eta}{\partial z}\frac{\partial U_z}{\partial \eta}}{J} = \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \eta}\left(\frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \zeta}\right)$$

$$\frac{\frac{\partial \zeta}{\partial z}\frac{\partial U_z}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial z}}{J}\right) \qquad 2.27$$

$\rightarrow$

$$\frac{\frac{\partial \zeta}{\partial z}\frac{\partial U_z}{\partial \zeta}}{J} = \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial z}U_z}{J}\right) - U_z\frac{\partial}{\partial \zeta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)$$

Substitute equations 2.19 to 2.27 into equation 2.18,

$$J\left[\frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial x}U_x}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial x}U_x}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial x}U_x}{J}\right) + \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial y}U_y}{J}\right)\right. \qquad 2.2$$

$$\left. + \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial y}U_y}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial y}U_y}{J}\right) + \frac{\partial}{\partial \xi}\left(\frac{\frac{\partial \xi}{\partial z}U_z}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\frac{\partial \eta}{\partial z}U_z}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{\frac{\partial \zeta}{\partial z}U_z}{J}\right)\right] = 0$$

which can be written as

$$J\left[\frac{\partial}{\partial\xi}\left(\frac{\frac{\partial\xi}{\partial x}U_x+\frac{\partial\xi}{\partial y}U_y+\frac{\partial\xi}{\partial z}U_z}{J}\right)+\frac{\partial}{\partial\eta}\left(\frac{\frac{\partial\eta}{\partial x}U_x+\frac{\partial\eta}{\partial y}U_y+\frac{\partial\eta}{\partial z}U_z}{J}\right)\right.$$
$$\left.+\frac{\partial}{\partial\zeta}\left(\frac{\frac{\partial\zeta}{\partial x}U_x+\frac{\partial\zeta}{\partial y}U_y+\frac{\partial\zeta}{\partial z}U_z}{J}\right)\right]=0 \qquad 2.29$$

Substituting equations 2.15 to 2.17 into equation 2.29, we have:

$$J\left[\frac{\partial}{\partial\xi}\left(\frac{V_\xi}{J}\right)+\frac{\partial}{\partial\eta}\left(\frac{V_\eta}{J}\right)+\frac{\partial}{\partial\zeta}\left(\frac{V_\zeta}{J}\right)\right]=0 \qquad 2.30$$

Equation 2.30 is the continuity equation in curvilinear coordinates.

Transforming the momentum equation

The LHS of the momentum equations can be written as

$$LHS(eq(2.10))=\rho\left[\frac{\partial U_x}{\partial t}+U_x\left(\frac{\partial\xi}{\partial x}\frac{\partial U_x}{\partial\xi}+\frac{\partial\eta}{\partial x}\frac{\partial U_x}{\partial\eta}+\frac{\partial\zeta}{\partial x}\frac{\partial U_x}{\partial\zeta}\right)\right. \qquad 2.31$$
$$\left.+U_y\left(\frac{\partial\xi}{\partial y}\frac{\partial U_x}{\partial\xi}+\frac{\partial\eta}{\partial y}\frac{\partial U_x}{\partial\eta}+\frac{\partial\zeta}{\partial y}\frac{\partial U_x}{\partial\zeta}\right)+U_z\left(\frac{\partial\xi}{\partial z}\frac{\partial U_x}{\partial\xi}+\frac{\partial\eta}{\partial z}\frac{\partial U_x}{\partial\eta}+\frac{\partial\zeta}{\partial z}\frac{\partial U_x}{\partial\zeta}\right)\right]$$

→

$$LHS(eq(2.10))=\rho\left[\frac{\partial U_x}{\partial t}+\left(\frac{\partial\xi}{\partial x}U_x+\frac{\partial\xi}{\partial y}U_y+\frac{\partial\xi}{\partial z}U_z\right)\frac{\partial U_x}{\partial\xi}\right. \qquad 2.32$$
$$\left.+\left(\frac{\partial\eta}{\partial x}U_x+\frac{\partial\eta}{\partial y}U_y+\frac{\partial\eta}{\partial z}U_z\right)\frac{\partial U_x}{\partial\eta}+\left(\frac{\partial\zeta}{\partial x}U_x+\frac{\partial\zeta}{\partial y}U_y+\frac{\partial\zeta}{\partial z}U_z\right)\frac{\partial U_x}{\partial\zeta}\right]$$

Substituting equations 2.15 to 2.17 into the above

equation we have

$$LHS(eq(2.10)) = \rho\left[\frac{\partial U_x}{\partial t} + V_\xi \frac{\partial U_x}{\partial \xi} + V_\eta \frac{\partial U_x}{\partial \eta} + V_\zeta \frac{\partial U_x}{\partial \zeta}\right] \quad 2.33$$

From the coordinate transformation formula we have

$$\nabla^2 \phi = J \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial}{\partial \xi^i}\left[\frac{g^{ij}}{J}\frac{\partial \phi}{\partial \xi^j}\right] \quad 2.34$$

Where

$$g^{ij} = \vec{a}^i \bullet \vec{a}^j \quad 2.35$$

$$\vec{a}^1 = \nabla \xi = \frac{\partial \xi}{\partial x}\vec{i} + \frac{\partial \xi}{\partial y}\vec{j} + \frac{\partial \xi}{\partial z}\vec{k} \quad 2.36$$

$$\vec{a}^2 = \nabla \eta = \frac{\partial \eta}{\partial x}\vec{i} + \frac{\partial \eta}{\partial y}\vec{j} + \frac{\partial \eta}{\partial z}\vec{k} \quad 2.37$$

$$\vec{a}^3 = \nabla \zeta = \frac{\partial \zeta}{\partial x}\vec{i} + \frac{\partial \zeta}{\partial y}\vec{j} + \frac{\partial \zeta}{\partial z}\vec{k} \quad 2.38$$

Substituting equation 2.35 into the RHS of equation 2.10

$$RHS(eq(2.10)) = \mu J \sum_{i=1}^{3} \sum_{j=1}^{3} \frac{\partial}{\partial \xi^i}\left[\frac{g^{ij}}{J}\frac{\partial U_x}{\partial \xi^i}\right] - \left(\frac{\partial \xi}{\partial x}\frac{\partial p}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial p}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial p}{\partial \zeta}\right) \quad 2.39$$

$$= \mu J \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}\left[\frac{g^{i1}}{J}\frac{\partial U_x}{\partial \xi} + \frac{g^{i2}}{J}\frac{\partial U_x}{\partial \eta} + \frac{g^{i3}}{J}\frac{\partial U_x}{\partial \zeta}\right] - \left(\frac{\partial \xi}{\partial x}\frac{\partial p}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial p}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial p}{\partial \zeta}\right)$$

$$= \mu J \left[\frac{\partial}{\partial \xi}\left(\frac{g^{11}}{J}\frac{\partial U_x}{\partial \xi} + \frac{g^{12}}{J}\frac{\partial U_x}{\partial \eta} + \frac{g^{13}}{J}\frac{\partial U_x}{\partial \zeta}\right) + \frac{\partial}{\partial \eta}\left(\frac{g^{21}}{J}\frac{\partial U_x}{\partial \xi} + \frac{g^{22}}{J}\frac{\partial U_x}{\partial \eta} + \frac{g^{23}}{J}\frac{\partial U_x}{\partial \zeta}\right)\right.$$

$$\left. + \frac{\partial}{\partial \zeta}\left(\frac{g^{31}}{J}\frac{\partial U_x}{\partial \xi} + \frac{g^{32}}{J}\frac{\partial U_x}{\partial \eta} + \frac{g^{33}}{J}\frac{\partial U_x}{\partial \zeta}\right)\right] - \left(\frac{\partial \xi}{\partial x}\frac{\partial p}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial p}{\partial \eta} + \frac{\partial \zeta}{\partial x}\frac{\partial p}{\partial \zeta}\right)$$

Finally equation 2.10 can be written as:

$$\rho\left[\frac{\partial U_x}{\partial t}+V_\xi\frac{\partial U_x}{\partial \xi}+V_\eta\frac{\partial U_x}{\partial \eta}+V_\zeta\frac{\partial U_x}{\partial \zeta}\right] \qquad 2.40$$
$$=\mu J\left[\frac{\partial}{\partial \xi}\left(\frac{g^{11}}{J}\frac{\partial U_x}{\partial \xi}+\frac{g^{12}}{J}\frac{\partial U_x}{\partial \eta}+\frac{g^{13}}{J}\frac{\partial U_x}{\partial \zeta}\right)+\frac{\partial}{\partial \eta}\left(\frac{g^{21}}{J}\frac{\partial U_x}{\partial \xi}+\frac{g^{22}}{J}\frac{\partial U_x}{\partial \eta}+\frac{g^{23}}{J}\frac{\partial U_x}{\partial \zeta}\right.\right.$$
$$\left.\left.+\frac{\partial}{\partial \zeta}\left(\frac{g^{31}}{J}\frac{\partial U_x}{\partial \xi}+\frac{g^{32}}{J}\frac{\partial U_x}{\partial \eta}+\frac{g^{33}}{J}\frac{\partial U_x}{\partial \zeta}\right)\right]-\left(\frac{\partial \xi}{\partial x}\frac{\partial p}{\partial \xi}+\frac{\partial \eta}{\partial x}\frac{\partial p}{\partial \eta}+\frac{\partial \zeta}{\partial x}\frac{\partial p}{\partial \zeta}\right)\right.$$

Similarly the momentum equations 2.8 and 2.9 can be written as:

$$\rho\left[\frac{\partial U_y}{\partial t}+V_\xi\frac{\partial U_y}{\partial \xi}+V_\eta\frac{\partial U_y}{\partial \eta}+V_\zeta\frac{\partial U_y}{\partial \zeta}\right] \qquad 2.41$$
$$=\mu J\left[\frac{\partial}{\partial \xi}\left(\frac{g^{11}}{J}\frac{\partial U_y}{\partial \xi}+\frac{g^{12}}{J}\frac{\partial U_y}{\partial \eta}+\frac{g^{13}}{J}\frac{\partial U_y}{\partial \zeta}\right)+\frac{\partial}{\partial \eta}\left(\frac{g^{21}}{J}\frac{\partial U_y}{\partial \xi}+\frac{g^{22}}{J}\frac{\partial U_y}{\partial \eta}+\frac{g^{23}}{J}\frac{\partial U_y}{\partial \zeta}\right.\right.$$
$$\left.\left.+\frac{\partial}{\partial \zeta}\left(\frac{g^{31}}{J}\frac{\partial U_y}{\partial \xi}+\frac{g^{32}}{J}\frac{\partial U_y}{\partial \eta}+\frac{g^{33}}{J}\frac{\partial U_y}{\partial \zeta}\right)\right]-\left(\frac{\partial \xi}{\partial y}\frac{\partial p}{\partial \xi}+\frac{\partial \eta}{\partial y}\frac{\partial p}{\partial \eta}+\frac{\partial \zeta}{\partial y}\frac{\partial p}{\partial \zeta}\right)\right.$$

$$\rho\left[\frac{\partial U_z}{\partial t}+V_\xi\frac{\partial U_z}{\partial \xi}+V_\eta\frac{\partial U_z}{\partial \eta}+V_\zeta\frac{\partial U_z}{\partial \zeta}\right] \qquad 2.42$$
$$=\mu J\left[\frac{\partial}{\partial \xi}\left(\frac{g^{11}}{J}\frac{\partial U_z}{\partial \xi}+\frac{g^{12}}{J}\frac{\partial U_z}{\partial \eta}+\frac{g^{13}}{J}\frac{\partial U_z}{\partial \zeta}\right)+\frac{\partial}{\partial \eta}\left(\frac{g^{21}}{J}\frac{\partial U_z}{\partial \xi}+\frac{g^{22}}{J}\frac{\partial U_z}{\partial \eta}+\frac{g^{23}}{J}\frac{\partial U_z}{\partial \zeta}\right.\right.$$
$$\left.\left.+\frac{\partial}{\partial \zeta}\left(\frac{g^{31}}{J}\frac{\partial U_z}{\partial \xi}+\frac{g^{32}}{J}\frac{\partial U_z}{\partial \eta}+\frac{g^{33}}{J}\frac{\partial U_z}{\partial \zeta}\right)\right]-\left(\frac{\partial \xi}{\partial z}\frac{\partial p}{\partial \xi}+\frac{\partial \eta}{\partial z}\frac{\partial p}{\partial \eta}+\frac{\partial \zeta}{\partial z}\frac{\partial p}{\partial \zeta}\right)\right.$$

Equations 2.40 to 2.42 are the momentum equations in curvilinear coordinates.

2.2 Spatial discretization of continuity and momentum equations

The tensor representation of incompressible Navier-Stokes equations in generalized curvilinear co-ordinates is

$$J\frac{\partial}{\partial\zeta_i}\left(\frac{1}{J}V^i\right)=0 \qquad 2.43$$

$$\frac{\partial U_i}{\partial t}+V^m\frac{\partial U_i}{\partial\zeta_m}=\frac{\partial\zeta_m}{\partial x_i}\frac{\partial p}{\partial\zeta_m}+\frac{J}{\mathrm{Re}}\frac{\partial}{\partial\zeta_m}\left(g^{mn}\frac{\partial U_i}{\partial\zeta_n}\right) \qquad 2.44$$

where U_i is the mean velocity component in Cartesian space, x_i is the component of Cartesian coordinates, p is the pressure, and Re is the Reynolds number. J is the Jacobian of the geometric transformation, and g^{mn} represents the contravariant metric tensor of the geometric transformation defined as

$$g^{mn}=\frac{1}{J}\xi^m_{x_i}\xi^n_{x_i} \qquad 2.45$$

$$\xi^k_{x_i}=\frac{\partial\xi^k}{\partial x_i} \qquad 2.46$$

V_m are the contra variant components of the mean Cartesian velocity components U_i, defined as

$$V^m=U_i\xi^m_{x_i} \qquad 2.47$$

Convective terms:

$$C(u_i)=V^m\frac{\partial u_i}{\partial\zeta_m}=V^\xi\frac{\partial u_i}{\partial\xi}+V^\eta\frac{\partial u_i}{\partial\eta}+V^\zeta\frac{\partial u_i}{\partial\zeta} \qquad 2.48$$

Introducing

$$V^{i\pm}_{i,j,k} = \frac{1}{2}\left[V^{i}_{i,j,k} \pm \left|V^{i}_{i,j,k}\right|\right] \tag{2.49}$$

$$\delta^{\pm}_{\zeta_1}(\,)_{i,j,k} = \pm\frac{1}{2\Delta\zeta_1}\left[-3(\,)_{i,j,k} + 4(\,)_{i\pm1,j,k} - (\,)_{i\pm2,j,k}\right] \tag{2.50}$$

and

$$V^{\xi}\frac{\partial u_i}{\partial \xi} = V^{\xi+}_{i,j,k}\delta^{-}_{\xi}(u_i)_{i,j,k} + V^{\xi-}_{i,j,k}\delta^{+}_{\xi}(u_i)_{i,j,k} \tag{2.51}$$

$$V^{\eta}\frac{\partial u_i}{\partial \eta} = V^{\eta+}_{i,j,k}\delta^{-}_{\eta}(u_i)_{i,j,k} + V^{\eta-}_{i,j,k}\delta^{+}_{\eta}(u_i)_{i,j,k} \tag{2.52}$$

$$V^{\zeta}\frac{\partial u_i}{\partial \zeta} = V^{\zeta+}_{i,j,k}\delta^{-}_{\zeta}(u_i)_{i,j,k} + V^{\zeta-}_{i,j,k}\delta^{+}_{\zeta}(u_i)_{i,j,k} \tag{2.53}$$

Substituting equations 2.51 to 2.53 into equations 2.49 and 2.50 we have

$$V^{\xi}\frac{\partial u_i}{\partial \xi} = \frac{\left(V^{\xi}_{i,j,k} - \left|V^{\xi}_{i,j,k}\right|\right)}{4\Delta\xi}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i+1,j,k} - (u_i)_{i+2,j,k}\right] - \frac{\left(V^{\xi}_{i,j,k} + \left|V^{\xi}_{i,j,k}\right|\right)}{4\Delta\xi}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i-1,j,k} - (u_i)_{i-2,j,k}\right] \tag{2.54}$$

$$V^{\eta}\frac{\partial u_i}{\partial \eta} = \frac{\left(V^{\eta}_{i,j,k} - \left|V^{\eta}_{i,j,k}\right|\right)}{4\Delta\eta}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i+1,j,k} - (u_i)_{i+2,j,k}\right] - \frac{\left(V^{\eta}_{i,j,k} + \left|V^{\eta}_{i,j,k}\right|\right)}{4\Delta\eta}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i-1,j,k} - (u_i)_{i-2,j,k}\right] \tag{2.55}$$

$$V^{\zeta}\frac{\partial u_i}{\partial \zeta} = \frac{\left(V^{\zeta}_{i,j,k} - \left|V^{\zeta}_{i,j,k}\right|\right)}{4\Delta\zeta}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i+1,j,k} - (u_i)_{i+2,j,k}\right] - \frac{\left(V^{\zeta}_{i,j,k} + \left|V^{\zeta}_{i,j,k}\right|\right)}{4\Delta\zeta}\left[-3(u_i)_{i,j,k} + 4(u_i)_{i-1,j,k} - (u_i)_{i-2,j,k}\right] \tag{2.56}$$

Then, the discretized convective terms can be written as:

$$C(u_i)=\frac{\left(V^{\xi}_{i,j,k}-\left|V^{\xi}_{i,j,k}\right|\right)}{4\Delta\xi}\left[-3(u_i)_{i,j,k}+4(u_i)_{i+1,j,k}-(u_i)_{i+2,j,k}\right] \qquad 2.57$$
$$-\frac{\left(V^{\xi}_{i,j,k}+\left|V^{\xi}_{i,j,k}\right|\right)}{4\Delta\xi}\left[-3(u_i)_{i,j,k}+4(u_i)_{i-1,j,k}-(u_i)_{i-2,j,k}\right]$$
$$+\frac{\left(V^{\eta}_{i,j,k}-\left|V^{\eta}_{i,j,k}\right|\right)}{4\Delta\eta}\left[-3(u_i)_{i,j,k}+4(u_i)_{i+1,j,k}-(u_i)_{i+2,j,k}\right]$$
$$-\frac{\left(V^{\eta}_{i,j,k}+\left|V^{\eta}_{i,j,k}\right|\right)}{4\Delta\eta}\left[-3(u_i)_{i,j,k}+4(u_i)_{i-1,j,k}-(u_i)_{i-2,j,k}\right]$$
$$+\frac{\left(V^{\zeta}_{i,j,k}-\left|V^{\zeta}_{i,j,k}\right|\right)}{4\Delta\zeta}\left[-3(u_i)_{i,j,k}+4(u_i)_{i+1,j,k}-(u_i)_{i+2,j,k}\right]$$
$$-\frac{\left(V^{\zeta}_{i,j,k}+\left|V^{\zeta}_{i,j,k}\right|\right)}{4\Delta\zeta}\left[-3(u_i)_{i,j,k}+4(u_i)_{i-1,j,k}-(u_i)_{i-2,j,k}\right]$$

Viscous terms:

$$D(u_i)=\frac{\partial}{\partial\zeta_m}\left(g^{mn}\frac{\partial u_i}{\partial\zeta_n}\right) \qquad 2.58$$
$$=\frac{\partial}{\partial\xi}\left(g^{11}\frac{\partial u_i}{\partial\xi}+g^{12}\frac{\partial u_i}{\partial\eta}+g^{13}\frac{\partial u_i}{\partial\zeta}\right)$$
$$+\frac{\partial}{\partial\eta}\left(g^{21}\frac{\partial u_i}{\partial\xi}+g^{22}\frac{\partial u_i}{\partial\eta}+g^{23}\frac{\partial u_i}{\partial\zeta}\right)$$
$$+\frac{\partial}{\partial\zeta}\left(g^{31}\frac{\partial u_i}{\partial\xi}+g^{32}\frac{\partial u_i}{\partial\eta}+g^{33}\frac{\partial u_i}{\partial\zeta}\right)$$

Central differencing gives,

2.59

$$\left[\frac{\partial}{\partial\xi}\left(g^{11}\frac{\partial u_i}{\partial\xi}+g^{12}\frac{\partial u_i}{\partial\eta}+g^{13}\frac{\partial u_i}{\partial\zeta}\right)\right]_{i,j,k}$$

$$=\frac{1}{\Delta\xi}\left[\left(g^{11}\frac{\partial u_i}{\partial\xi}+g^{12}\frac{\partial u_i}{\partial\eta}+g^{13}\frac{\partial u_i}{\partial\zeta}\right)_{i+1/2,j,k}-\left(g^{11}\frac{\partial u_i}{\partial\xi}+g^{12}\frac{\partial u_i}{\partial\eta}+g^{13}\frac{\partial u_i}{\partial\zeta}\right)_{i-1/2,j,k}\right]$$

$$=\frac{1}{\Delta\xi}\left[\left(g^{11}\right)_{i+1/2,j,k}\frac{\left(u_i\right)_{i+1,j,k}-\left(u_i\right)_{i,j,k}}{\Delta\xi}\right.$$

$$+\left(g^{12}\right)_{i+1/2,j,k}\frac{\left(u_i\right)_{i,j+1,k}-\left(u_i\right)_{i,j-1,k}+\left(u_i\right)_{i+1,j+1,k}-\left(u_i\right)_{i+1,j-1,k}}{4\Delta\eta}$$

$$+\left(g^{13}\right)_{i+1/2,j,k}\frac{\left(u_i\right)_{i,j,k+1}-\left(u_i\right)_{i,j,k-1}+\left(u_i\right)_{i+1,j,k+1}-\left(u_i\right)_{i+1,j,k-1}}{4\Delta\zeta}$$

$$-\left(g^{11}\right)_{i-1/2,j,k}\frac{\left(u_i\right)_{i,j,k}-\left(u_i\right)_{i-1,j,k}}{\Delta\xi}$$

$$-\left(g^{12}\right)_{i-1/2,j,k}\frac{\left(u_i\right)_{i,j+1,k}-\left(u_i\right)_{i,j-1,k}+\left(u_i\right)_{i-1,j+1,k}-\left(u_i\right)_{i-1,j-1,k}}{4\Delta\eta}$$

$$\left.-\left(g^{13}\right)_{i-1/2,j,k}\frac{\left(u_i\right)_{i,j,k+1}-\left(u_i\right)_{i,j,k-1}+\left(u_i\right)_{i-1,j,k+1}-\left(u_i\right)_{i-1,j,k-1}}{4\Delta\zeta}\right]$$

Similarly,

2.6(

$$\left[\frac{\partial}{\partial\eta}\left(g^{21}\frac{\partial u_i}{\partial\xi}+g^{22}\frac{\partial u_i}{\partial\eta}+g^{23}\frac{\partial u_i}{\partial\zeta}\right)\right]_{i,j,k}$$

$$=\frac{1}{\Delta\eta}\left[\left(g^{21}\frac{\partial u_i}{\partial\xi}+g^{22}\frac{\partial u_i}{\partial\eta}+g^{23}\frac{\partial u_i}{\partial\zeta}\right)_{i,j+1/2,k}-\left(g^{21}\frac{\partial u_i}{\partial\xi}+g^{22}\frac{\partial u_i}{\partial\eta}+g^{23}\frac{\partial u_i}{\partial\zeta}\right)_{i,j-1/2,k}\right]$$

$$=\frac{1}{\Delta\eta}\left[\left(g^{21}\right)_{i,j+1/2,k}\frac{(u_i)_{i+1,j+1,k}-(u_i)_{i-1,j+1,k}+(u_i)_{i+1,j,k}-(u_i)_{i-1,j,k}}{4\Delta\xi}\right.$$

$$+\left(g^{22}\right)_{i,j+1/2,k}\frac{(u_i)_{i,j+1,k}-(u_i)_{i,j,k}}{\Delta\eta}$$

$$+\left(g^{23}\right)_{i,j+1/2,k}\frac{(u_i)_{i,j,k+1}-(u_i)_{i,j,k-1}+(u_i)_{i,j+1,k+1}-(u_i)_{i,j+1,k-1}}{4\Delta\zeta}$$

$$-\left(g^{21}\right)_{i,j-1/2,k}\frac{(u_i)_{i+1,j-1,k}-(u_i)_{i-1,j-1,k}+(u_i)_{i+1,j,k}-(u_i)_{i-1,j,k}}{4\Delta\xi}$$

$$-\left(g^{22}\right)_{i,j-1/2,k}\frac{(u_i)_{i,j,k}-(u_i)_{i,j-1,k}}{\Delta\eta}$$

$$\left.-\left(g^{23}\right)_{i,j-1/2,k}\frac{(u_i)_{i,j,k+1}-(u_i)_{i,j,k-1}+(u_i)_{i,j-1,k+1}-(u_i)_{i,j-1,k-1}}{4\Delta\zeta}\right]$$

2.6:

$$\left[\frac{\partial}{\partial\zeta}\left(g^{31}\frac{\partial u_i}{\partial\xi}+g^{32}\frac{\partial u_i}{\partial\eta}+g^{33}\frac{\partial u_i}{\partial\zeta}\right)\right]_{i,j,k}$$

$$=\frac{1}{\Delta\zeta}\left[\left(g^{31}\frac{\partial u_i}{\partial\xi}+g^{32}\frac{\partial u_i}{\partial\eta}+g^{33}\frac{\partial u_i}{\partial\zeta}\right)_{i,j,k+1/2}-\left(g^{31}\frac{\partial u_i}{\partial\xi}+g^{32}\frac{\partial u_i}{\partial\eta}+g^{33}\frac{\partial u_i}{\partial\zeta}\right)_{i,j,k-1/2}\right]$$

$$=\frac{1}{\Delta\zeta}\left[\left(g^{31}\right)_{i,j,k+1/2}\frac{(u_i)_{i+1,j,k+1}-(u_i)_{i-1,j,k+1}+(u_i)_{i+1,j,k}-(u_i)_{i-1,j,k}}{4\Delta\xi}\right.$$

$$+\left(g^{32}\right)_{i,j+1/2,k}\frac{(u_i)_{i,j+1,k+1}-(u_i)_{i,j-1,k+1}+(u_i)_{i,j+1,k}-(u_i)_{i,j-1,k}}{4\Delta\eta}$$

$$+\left(g^{33}\right)_{i,j+1/2,k}\frac{(u_i)_{i,j,k+1}-(u_i)_{i,j,k}}{\Delta\zeta}$$

$$-\left(g^{31}\right)_{i,j-1/2,k}\frac{(u_i)_{i+1,j,k-1}-(u_i)_{i-1,j,k-1}+(u_i)_{i+1,j,k}-(u_i)_{i-1,j,k}}{4\Delta\xi}$$

$$-\left(g^{32}\right)_{i,j-1/2,k}\frac{(u_i)_{i,j+1,k-1}-(u_i)_{i,j-1,k-1}+(u_i)_{i,j+1,k}-(u_i)_{i,j-1,k}}{4\Delta\eta}$$

$$\left.-\left(g^{33}\right)_{i,j-1/2,k}\frac{(u_i)_{i,j,k}-(u_i)_{i,j,k-1}}{\Delta\zeta}\right]$$

To compute the metrics and the Jacobian at the half nodes ((i+1/2,j,k) and etc.), where they are needed for the discretization of the viscous terms, a simple averaging procedure is employed. Finally

Pressure Gradient:

$$H_i(p)=\frac{\partial \zeta_m}{\partial x_i}\frac{\partial p}{\partial \zeta_m}=\frac{\partial \xi}{\partial x_i}\frac{\partial p}{\partial \xi}+\frac{\partial \eta}{\partial x_i}\frac{\partial p}{\partial \eta}+\frac{\partial \zeta}{\partial x_i}\frac{\partial p}{\partial \zeta} \qquad 2.62$$

Central differencing the above equation we get 2.63

$$\left(\frac{\partial \xi}{\partial x_i}\frac{\partial p}{\partial \xi}\right)_{i,j,k}=\left(\frac{\partial \xi}{\partial x_i}\right)_{i,j,k}\cdot\frac{p_{i+1,j,k}-p_{i-1,j,k}}{2\Delta\xi}$$

$$\left(\frac{\partial \mu}{\partial x_i}\frac{\partial p}{\partial \eta}\right)_{i,j,k}=\left(\frac{\partial \eta}{\partial x_i}\right)_{i,j,k}\cdot\frac{p_{i,j+1,k}-p_{i,j-1,k}}{2\Delta\eta} \qquad 2.64$$

$$\left(\frac{\partial \zeta}{\partial x_i}\frac{\partial p}{\partial \zeta}\right)_{i,j,k}=\left(\frac{\partial \zeta}{\partial x_i}\right)_{i,j,k}\cdot\frac{p_{i,j,k+1}-p_{i,j,k-1}}{2\Delta\zeta} \qquad 2.65$$

2.3 Reynolds stress modeling

The instantaneous momentum equations for a Newtonian fluid in tensor form can be represented as

$$\left[\frac{\partial \tilde{u}_i}{dt}+\tilde{u}_j\frac{\partial \tilde{u}_i}{dx_j}\right]=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i}+\upsilon\frac{\partial^2 \tilde{u}_i}{\partial x_j^2} \qquad 2.66$$

Generally the incompressibility in the stress term is referred to the incompressible momentum equation in the

following form

$$\left[\frac{\partial \tilde{u}_i}{dt}+\tilde{u}_j\frac{\partial \tilde{u}_i}{dx_j}\right]=-\frac{1}{\rho}\frac{\partial \tilde{p}}{\partial x_i}+\frac{\partial T_{ij}^{(v)}}{\partial x_j} \qquad 2.67$$

Laminar solutions to Navier-Stokes equations exist that are consistent with the boundary conditions but perturbations can lead to these solutions to become turbulent. It is convenient to analyze the flow into two parts, a mean (or average) component and fluctuating component. Thus the instantaneous velocity and stress can be written using Reynolds decomposition as :

$$\begin{aligned}\tilde{u}_i &= U_i + u_i \\ \tilde{p} &= P + p \\ \tilde{\tau}_{ij} &= T_{ij} + \tau_{ij}\end{aligned} \qquad 2.68$$

where capital letters represent mean motion and the small letters represent fluctuating motions. This technique for decomposing is referred to as Reynolds decomposition. Substituting 2.68 into 2.67 and considering the equation of averaged motion reduces to

$$\rho\left[\frac{\partial U_i}{\partial t}+U_j\frac{\partial U_i}{\partial x_j}\right]=-\frac{\partial P}{\partial x_i}+\frac{\partial T_{ij}^{(v)}}{\partial x_j}-\rho\left\langle u_j\frac{\partial u_i}{\partial x_j}\right\rangle \qquad 2.69$$

Subtracting 2.67 from 2.69 we get the equation for

fluctuation as:

$$\rho\left[\frac{\partial u_i}{\partial t}+U_j\frac{\partial u_i}{\partial x_j}\right]=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}^{(v)}}{\partial x_j}-\rho\left[u_j\frac{\partial U_i}{\partial x_j}\right]-\left\{u_j\frac{\partial u_i}{\partial x_j}-\rho\left\langle u_j\frac{\partial u_i}{\partial x_j}\right\rangle\right\} \quad 2.70$$

The first two terms of the equation 2.70 are like that for the mean flow but the third term unlike the others , and will be seen later to represent the primary means by which fluctuations extract energy from the production terms. The last term in the equation is quadratic in fluctuating velocity unlike other terms which are linear.

Deriving the u_iu_j equation

The steps for deriving the Reynolds stress equation is described below

1) Multiply equation 2.70 with u_j
2) Time average the obtained equation
3) Subtract the obtained from 2 with 2.70
4) Repeat the same procedure for u_j
5) Add them together to obtain the Reynolds stress.

The equation obtained after completing the above procedure is displayed below in equation 2.71

$$\frac{\partial\langle u_i u_j\rangle}{\partial t}+U_j\frac{\partial\langle u_i u_j\rangle}{\partial x_j}=-\frac{1}{\rho}\left[\left\langle u_i\frac{\partial p}{\partial x_j}\right\rangle+\left\langle u_j\frac{\partial p}{\partial x_i}\right\rangle\right]-\left[\left\langle u_i u_k\frac{\partial U_j}{\partial x_k}\right\rangle+\left\langle u_j u_k\frac{\partial U_i}{\partial x_k}\right\rangle\right] \quad 2.71$$
$$+\frac{1}{\rho}\left[\left\langle u_i\frac{\partial \tau_{kj}^{(v)}}{\partial x_k}\right\rangle+\left\langle u_j\frac{\partial \tau_{ik}^{(v)}}{\partial x_k}\right\rangle\right]$$
$$-\left[\langle u_i u_k\rangle\frac{\partial u_j}{\partial x_k}+\langle u_k u_j\rangle\frac{\partial u_i}{\partial x_k}\right]$$

Rearranging the terms and by expanding and after a large number of simplifications the equation becomes

$$\frac{\partial}{\partial t}\langle u_i u_j\rangle+U_k\frac{\partial}{\partial x_j}\langle u_i u_j\rangle=-\left\langle\frac{P}{\rho}\left[\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right]\right\rangle-\left[\langle u_i u_k\rangle\frac{\partial U_j}{\partial x_k}+\langle u_k u_j\rangle\frac{\partial U_i}{\partial x_k}\right] \quad 2.72$$
$$+\frac{\partial}{\partial x_j}\left\{-\left[\langle pu_j\rangle\delta_{ik}+\langle pu_i\rangle\delta_{jk}\right]-\langle u_i u_j u_k\rangle+2\nu\left[\langle s_{ik}u_j\rangle+\right.\right.$$
$$\left.\left.-2\nu\left[\left\langle s_{ik}\frac{\partial u_j}{\partial x_k}\right\rangle+\left\langle s_{kj}\frac{\partial u_i}{\partial x_k}\right\rangle\right]\right.$$

This is the Reynolds stress equation that has been a primary vehicle for turbulence modeling over past few decades. The terms on the right hand side are referred to respectively as

1) the pressure-strain rate term
2) production term
3) turbulence transport term
4) dissipation term

The number of unknown quantities in the above

equation is much more than the number of equations. The absence of additional equations is often referred to as the Turbulence closure problem. There are different approximations in closing the equation. Lars Davidson [24] has a excellent description on the closure methods in this class notes.

Algebraic Models: An algebraic equation is used to compute a turbulent viscosity, often called *eddy* viscosity. The Reynolds stress tensor is then computed using an assumption which relates the Reynolds stress tensor to the velocity gradients and the turbulent viscosity. This assumption is called the *Boussinesq assumption*. Models which are based on a turbulent (eddy) viscosity are called *eddy viscosity* models

One equation models: In these models a transport equation is solved for a turbulent quantity (usually the turbulent kinetic energy) and a second turbulent quantity (usually a turbulent length scale) is obtained from an algebraic expression. The turbulent viscosity is calculated from Boussinesq assumption.

Two equation models: These models fall into the class of eddy viscosity models. Two transport equations are derived which describe transport of two scalars, for example the turbulent kinetic energy k and its dissipation ε. The Reynolds stress tensor is then

computed using an assumption which relates the Reynolds stress tensor to the velocity gradients and an eddy viscosity. The latter is obtained from the two transported scalars.

Reynolds stress modeling: Here a transport equation is derived for the Reynolds tensor $\langle u_i u_j \rangle$. One transport equation has to be added for determining the length scale of turbulence. Usually an equation for the dissipation is used.

The turbulent kinetic energy can be obtained by setting indices i=j in the Reynolds stress equation.

$$\left[\frac{\partial}{\partial t}+U_k\frac{\partial}{\partial x_j}\right]k=\frac{\partial}{\partial x_j}\left\{-\frac{1}{\rho}\langle pu_i\rangle\delta_{ij}-\frac{1}{2}\langle q^2u_k\rangle+2\nu\langle s_{ik}u_i\rangle\right\}-\langle u_iu_k\rangle\frac{\partial U_i}{\partial x_k} \qquad 2.73$$
$$-2\nu\langle s_{ik}s_{ik}\rangle$$

Where q is the average fluctuating kinetic energy per unit mass.

Launder and Sharma closure model

The second term on the LHS of the equation 2.73 is the production term. This is modeled from Boussinesq assumption.

$$\langle u_iu_k\rangle\frac{\partial U_i}{\partial x_k}=\nu_t\left(\frac{\partial U_i}{\partial x_k}+\frac{\partial U_k}{\partial x_i}\right)\frac{\partial U_i}{\partial x_k}-\frac{2}{3}\rho k\frac{\partial U_i}{\partial x_i} \qquad 2.74$$

The last term in the above equation becomes zero.

The triple correlations term or the diffusion term

which is the first term in the equation 2.73 can be modeled using a gradient law where we assume k is diffused down the gradient from higher k regions to lower k regions. We get

$$\left\langle u_j u_j u_i \right\rangle = -\frac{\nu_t}{\sigma_k}\frac{\partial}{\partial x_j}k \qquad 2.75$$

Finally the dissipation term which is the last term becomes

$$\varepsilon \equiv \mu\left\langle \frac{\partial u_i}{\partial x_j}\frac{\partial u_i}{\partial x_j} \right\rangle = \frac{k^{\frac{3}{2}}}{l} \qquad 2.76$$

The transport equation for ε is

$$\frac{\partial \varepsilon}{\partial t} + \frac{1}{\varepsilon_\beta}\langle \overline{u}_j \rangle \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\nu_{t_{eff}}}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j}\right] + C_1 f_1 \frac{\varepsilon}{k} G - C_2 f_2 \frac{\varepsilon^2}{k} + E - C_3 \frac{\varepsilon}{k}\overline{f}_i \langle \overline{u}_i \rangle \qquad 2.77$$

$$E = 2\nu\nu_{t_{eff}}\left(\frac{\partial^2 \langle U_2 \rangle}{\partial x_2^2}\right)^2 \qquad 2.78$$

Similarly, the forcing term in equation 2.77 has been obtained by a simple scaling of the corresponding term in equation 2.73. As a result, the coefficient C_3 is a model constant. It was set at 0.11. The value of the model constants are σ_k = 1.0, σ_ε = 1.3, C_1 = 1.44, C_2 = 1.92, and C_μ = 0.09.

The damping functions f_μ, f_1, and f_2 proposed by

Launder and Sharma [25] are,

$$f_\mu = exp\left[-3.4/\left(1 + R_t/50\right)^2\right] \qquad (2.46)$$

$$f_1 = 1 \qquad (2.47)$$

$$f_2 = 1 - 0.3\,exp\left(-R_t^2\right) \qquad (2.48)$$

2.4 Temporal discretization of continuity and momentum equations

The system of the discrete continuity and momentum equations is integrated in time using the four-stage, explicit Runge-Kutta scheme. The Runge-kutta first used by Jameson *et al.* [26] to solve the compressible Euler equations although is explicit in time, is known to have very good error damping properties. The Runge-Kutta scheme is applied to the system of the governing equations as follows (for ℓ = 1,2,3,4):

$$DIV\left(Q^{\ell}_{i,j,k}\right) = 0 \qquad 2.79$$

$$Q^{\ell}_{i,j,k} = Q^{n}_{i,j,k} - \alpha_{\ell}\Delta t_{i,j,k} RHS^{\ell-1}_{i,j,k} \qquad 2.80$$

In the above equations, the superscript "*n*" denotes the time step at which the solution is known, while the

superscript "ℓ" denotes an intermediate time level (or iteration level) used to advance the solution from time step "n" to time step "n+1" (we designate $Q^{\ell} = Q^{n}$ for $\ell = 0$ and $Q^{\ell} = Q^{n+1}$ for $\ell = 4$). For the four-stage scheme, the coefficient of ℓ is: 1/4,1/3,1/2 and 1 for $\ell = 1,2,3,4$, in sequence. The *RHS* in Equation 2.80 denotes the discrete approximation of the right-hand side of the momentum equations (2.8-2.10) at the node (i,j,k):

$$RHS = C(u_i) - D(u_i) + H_i(p) \qquad 2.81$$

$t_{i,j,k}$ in equation 2.80 is the time increment which varies in space (local time stepping).

CHAPTER III

THE SUBDIVISION FINITE ELEMENT STRCUTURAL DYNAMIC SOLVER

3.1 Basic formulation

In this chapter the details of the structural dynamic part of the coupled fluid structure interaction solver developed by Dr. Yang Yang as part of his PhD work at the computational engineering physics group of western Michigan University is discussed in detail. The fin is made of Mylar and only 0.0001 m in thickness, which ensures the validity of thin shell assumptions. The thin shell can be represented by the Cosserat surface i.e. the middle Surface and its normal director, denoted in figure 3.1 by the dotted lines and the blue arrows respectively. As shown in figure 3.1 , a shell structure that undergoes a deformation $\boldsymbol{F}$ from the reference configuration to the deformed one[1]. This process can be represented by

$$\bar{r}(\theta^1,\theta^2,\theta^3)=\bar{x}(\theta^1,\theta^2)+\theta^3\bar{a}_3(\theta^1,\theta^2),\quad -\frac{h}{2}\le\theta^3\le\frac{h}{2} \tag{3.1}$$

---

[1] In the text that follows the quantities belonging to the reference configuration are accentuated by an over bar while the quantities of the deformed configuration are not decorated.

$$r(\theta^1,\theta^2,\theta^3) = x(\theta^1,\theta^2) + \theta^3 \bar{a}_3(\theta^1,\theta^2), \quad -\frac{h}{2} \le \theta^3 \le \frac{h}{2} \tag{3.2}$$

Courtesy of Dr.Yang

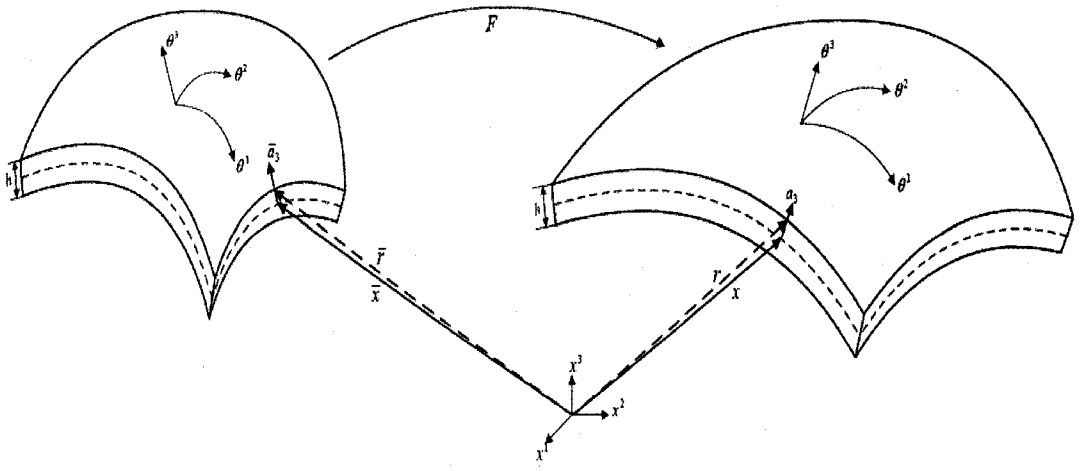


Figure 3.1: Shell geometry in the reference (left) and deformed (right) configuration

where the function $\bar{x}(\theta^1,\theta^2)$ and $x(\theta^1,\theta^2)$ furnish the parametric representations of the middle surface in the reference and deformed configurations, respectively. The basis vectors of middle surface are

$$\bar{a}_\alpha = \bar{x}_{,\alpha}{}^2, a_\alpha = x_{,\alpha} \tag{3.3}$$

where α runs from 1 to 2. The covariant components of the surface metric tensors are

$$\bar{a}_{\alpha\beta} = \bar{a}_\alpha \cdot \bar{a}_\beta, a_{\alpha\beta} = a_\alpha \cdot a_\beta \tag{3.4}$$

The corresponding contravariant components are the

inverse of Equation 3.4, i.e. $\bar{a}^{\alpha\gamma}\bar{a}_{\gamma\beta}=\delta^{\alpha}_{\beta}$, and $a^{\alpha\gamma}a_{\gamma\beta}=\delta^{\alpha}_{\beta}$. The Jacobian for the referential embedded (θ^1,θ^2) plane relative to the Euclidean space (x^1,x^2,x^3) is $d\bar{\Omega}=\sqrt{\bar{a}}d\theta^1d\theta^2=\|\bar{a}_1\times\bar{a}_2\|d\theta^1d\theta^2$.. The shell director $\bar{a}_3$ in the reference configuration should be the unit vector normal to the middle surface as shown in figure 3.1, and is given by

$$\bar{a}_3=\frac{\bar{a}_1\times\bar{a}_2}{\|\bar{a}_1\times\bar{a}_2\|} \qquad 3.5$$

While the shell director in the deformed configuration, according to the Kirchhoff-Love thin shell assumptions, remains straight and normal to the middle surface. Hence we have

$$a_3=\frac{a_1\times a_2}{\|a_1\times a_2\|} \qquad 3.6$$

Now based on Equations 3.1 and 3.2, the covariant base vectors for the body of shell in the reference and the deformed configurations are

$$\bar{g}_\alpha=\frac{\partial\bar{r}}{\partial\theta^\alpha}=\bar{a}_\alpha+\theta^3\bar{a}_{3,\alpha},\ \bar{g}_3=\frac{\partial\bar{r}}{\partial\theta^3}=\bar{a}_3 \qquad 3.7$$

$$g_\alpha=\frac{\partial r}{\partial\theta^\alpha}=a_\alpha+\theta^3a_{3,\alpha},\ g_3=\frac{\partial r}{\partial\theta^3}=a_3 \qquad 3.8$$

where α goes from 1 to 2. The corresponding covariant metrics of the shell body are simply the dot products of base vectors, $\bar{g}_{ij}=\bar{g}_i\cdot\bar{g}_j$ and $g_{ij}=g_i\cdot g_j$, where i

and j run from 1 to 3. By definition the Green-Lagrange strain tensor is $E_{ij} = \frac{1}{2}\left(g_{ij} - \bar{g}_{ij}\right)$, to the first order in the shell thickness h, it can be shown that the Green-Lagrange strain of the shell is of the following form

$$E_{ij} = \alpha_{ij} + \theta^3 \beta_{\alpha\beta} \tag{3.9}$$

where $\alpha_{ij} = \frac{1}{2}\left(a_i \cdot a_j - \bar{a}_i \cdot \bar{a}_j\right)$ and $\beta_{\alpha\beta} = a_\alpha \cdot a_{3,\beta} - \bar{a}_\alpha \cdot \bar{a}_{3,\beta}$ and i, j from 1 to 3; α, β from 1 to 2. The in-plane components $\alpha_{\alpha\beta}$ with α, β running from 1 to 2, or the membrane strains, measures the straining of the surface; the components $\alpha_{\alpha 3}$ measures the shearing of the shell director; the component α_{33} measures the stretching of the director; and the component $\beta_{\alpha\beta}$ or the bending strains, measures the bending or change in the curvature of the shell.

Since only the linear small deformation is considered, a simple relation between the reference and deformed configurations can be assumed

$$x\left(\theta^1, \theta^2\right) = \bar{x}\left(\theta^1, \theta^2\right) + u\left(\theta^1, \theta^2\right) \tag{3.10}$$

where u is the displacement vector of the middle surface. After substituting Equation 3.10 into Equation 3.3, the strain tensors $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$ can be linearized as

$$\alpha_{\alpha\beta} = \frac{1}{2}\left(\bar{a}_\alpha \cdot u_{,\beta} + u_{,\alpha} \cdot \bar{a}_\beta\right) \tag{3.11}$$

$$\beta_{\alpha\beta} = -u_{,\alpha\beta}\cdot\bar{a}_3 + \frac{1}{\sqrt{\bar{a}}}\left[u_{,1}\cdot\left(\bar{a}_{\alpha,\beta}\times\bar{a}_2\right)+u_{,2}\cdot\left(\bar{a}_1\times\bar{a}_{\alpha,\beta}\right)\right]+ \frac{\bar{a}_3\cdot\bar{a}_{\alpha,\beta}}{\sqrt{\bar{a}}}\left[u_{,1}\cdot\left(\bar{a}_2\times\bar{a}_3\right)+u_{,2}\cdot\left(\bar{a}_3\times\bar{a}_1\right)\right] \quad 3.12$$

The comma in the subscript of a variable in the above equations, similar to Equation 3.3 represents the partial derivatives with respect to θ^{α}. Equation 3.11 and 3.12 show that the only unknowns are the displacement vectors u of the middle surface while the rest terms are all already known from the reference configuration.

The simplest form of the strain energy per unit surface area for thin shell structures is due to Simo and Fox [27]

$$W = \frac{1}{2}\frac{Eh}{1-\nu^2}H^{\alpha\beta\gamma\delta}\alpha_{\alpha\beta}\alpha_{\gamma\delta} + \frac{1}{2}\frac{Eh^3}{12(1-\nu^2)}H^{\alpha\beta\gamma\delta}\beta_{\alpha\beta}\beta_{\gamma\delta} \quad 3.13$$

where E is Young's modulus, ν is Poisson's ratio,

$$H^{\alpha\beta\gamma\delta} = \nu\bar{a}^{\alpha\beta}\bar{a}^{\gamma\delta} + \frac{1}{2}(1-\nu)(\bar{a}^{\alpha\gamma}\bar{a}^{\beta\delta} + \bar{a}^{\alpha\delta}\bar{a}^{\beta\gamma}) \quad 3.14$$

and $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$ are defined by Equations 3.11 and 3.12 while $\bar{a}^{\alpha\beta}$ is the inverse of $\bar{a}_{\alpha\beta}$ defined in Equation 3.4. If the hyper elastic material is assumed then the

constitutive relations between the membrane/bending stress and the strain can be derived as

$$n^{\alpha\beta} = \frac{\partial W}{\partial \alpha_{\alpha\beta}} = \frac{Eh}{1-\nu^2} H^{\alpha\beta\gamma\delta} \alpha_{\gamma\delta} \tag{3.15}$$

$$m^{\alpha\beta} = \frac{\partial W}{\partial \beta_{\alpha\beta}} = \frac{Eh^3}{12(1-\nu^2)} H^{\alpha\beta\gamma\delta} \beta_{\gamma\delta} \tag{3.16}$$

If the shell structure is subject to some kind of external loads, such as pressure p, then the total potential energy of the structure can be expressed as

$$\begin{aligned} E_{total} &= E_{\text{internal}} - E_{external} \\ &= \int_{\Omega} W d\Omega - \int_{\Omega} p \cdot u d\Omega \end{aligned} \tag{3.17}$$

The integration in Equation 3.17 should be understood to be carried out on the reference configuration. By the minimum potential energy principal and the variational method, a static equilibrium configuration could be found by solving the following problem,

$$\delta E_{total} = \int_{\Omega} \left[n^{\alpha\beta} \delta \alpha_{\alpha\beta} + m^{\alpha\beta} \delta \beta_{\alpha\beta} \right] d\Omega - \int_{\Omega} p \cdot \delta u d\Omega = 0 \tag{3.18}$$

where $n^{\alpha\beta}$ and $m^{\alpha\beta}$ are defined in Equations 3.15 and 3.16 respectively. Equation 3.18 serves as the basis for the finite element approximation.

3.2 Finite element discretization

Since the Kirchhoff-Love thin shell assumption was used in deriving the linear elastic formulation, it requires square integrable finite element space to maintain the C^1continuity. The subdivision finite element method proposed by Cirak [28] was adopted for this purpose. In this method, a thin shell surface is recursively triangulated by Loop's rule [29], which is a set of very simple linear relations. It can be proven that the repetitive use of Loop's rule on the thin shell will converge to a smooth surface with at least C^1 continuity property. And it can be conveniently represented by a closed-form mathematical formula. The hallmark of the subdivision finite element method is that the interpolation of a function involves not only the local nodal values but the ones in the first neighbor defined by a topological mask. The interpolants or the shape functions are twelve fourth order polynomials, $N^I\left(t^1,t^2\right)$ where I runs from 1 to 12, $t^1$and $t^2$are the triangular coordinates. Hence a continuous function such as the displacement vector u can be represented as

$$u\left(t^1,t^2\right)=\sum_{I=1}^{12} N^I\left(t^1,t^2\right)u_I \qquad 3.19$$

For the elements that are on the boundary or

irregular in topology, there may not exactly be twelve nodes in the first neighbor. For these cases, special treatments such as building the ghost nodes and subpatch scheme need to be used. For the detailed implementation procedure, readers are referred to the paper by Cirak [29]. Here we only list the final working formula. After substituting Equations 3.15 and 3.16 into Equation 3.18, the minimum potential energy principal can be written as

$$\delta E_{total} = \int_{\Omega} \left[\frac{Eh}{1-\nu^2} H^{\alpha\beta\gamma\delta} \alpha_{\gamma\delta} \delta\alpha_{\alpha\beta} + \frac{Eh^3}{12\left(1-\nu^2\right)} H^{\alpha\beta\gamma\delta} \beta_{\gamma\delta} \delta\beta_{\alpha\beta} \right] d\Omega = 0 \qquad 3.20$$
$$- \int_{\Omega} p \cdot \delta u d\Omega$$

The strain tensors $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$ are interpolated using Equation 3.19. After some mathematical operations, the membrane and bending strain tensors for one element take the form

$$\alpha_h = \sum_{I=1}^{12} M^I u_I \qquad 3.21$$

$$\beta_h = \sum_{I=1}^{12} B^I u_I \qquad 3.22$$

where M and B are the elemental strain matrices, the components of which can be found by a series of

substitutions of the related equations as shown above. The integration in Equation 3.20 under the framework of finite element method, can be broken down to the summation of individual elemental integrations, as follows, where *NEL* represents the number of finite elements

$$\sum_{K=1}^{NEL}\left\{\int_{\Omega_K}\delta u_I\left[\frac{Eh}{1-\nu^2}(M^I)^T HM^J+\frac{Eh^3}{12\left(1-\nu^2\right)}(B^I)^T HB^J\right]u_J d\Omega_k - \int_{\Omega_K}\delta u_I p\cdot N^I d\Omega_k\right\}=0 \qquad 3.23$$

Since δu_I is an arbitrary small quantity, the above equation is equivalent to

$$\sum_{K=1}^{NEL}\left\{\int_{\Omega_K}\left[\frac{Eh}{1-\nu^2}(M^I)^T HM^J+\frac{Eh^3}{12\left(1-\nu^2\right)}(B^I)^T HB^J\right]u_J d\Omega_k - \int_{\Omega_K} p\cdot N^I d\Omega_k\right\}=0 \qquad 3.24$$

Equation 3.23 can be put into a compact format, by recognizing the stiffness matrix *K* as

$$K^{IJ}=\sum_{K=1}^{NEL}\left\{\int_{\Omega_K}\left[\frac{Eh}{1-\nu^2}(M^I)^T HM^J+\frac{Eh^3}{12\left(1-\nu^2\right)}(B^I)^T HB^J\right]d\Omega_k\right\} = \sum_{K=1}^{NEL}K_K{}^{IJ} \qquad 3.25$$

and the force vector *f* as

$$f^I = \sum_{K=1}^{NEL} \int_{\Omega_k} p \cdot N^I d\Omega_k \qquad 3.26$$

Hence Equation 3.23 can be expressed as

$$M\ddot{u} + Ku = f \qquad 3.27$$

where the double dots represent the second time derivative, and M is the mass matrix defined as

$$M^{IJ} = \sum_{K=1}^{NEL} \int_{\Omega_K} \rho N^I N^J d\Omega_K \qquad 3.28$$

in which ρ is the material density.

The method for integrating Equation 25 in time adopted in this work is the explicit Newmark central difference scheme for its advantages in easy algorithm organization and low computer memory requirement as long as the lumped mass matrix is used. A pseudo code for the Newmark method is provided in Listing 1. First the lumped mass matrix stored in *mass_lump(:)* is computed by calling subroutine *Massassem* on line 100, which is of the diagonal form so that the inversion is trivial. Then the stable time step is determined on line 102 followed by the main Newmark integration loop starting from line 104 to line 122. In the loop, the displacement vector and velocity vector are first calculated at the predictor step on line 105 and line 106, and then the boundary conditions are applied explicitly. The internal and

external forces of the system are calculated in the subroutine *Getforce* on line 111 and their sums are stored in *resi_global(:)*. The velocity vector is updated on line 116 which concludes one Newmark integration iteration. The transient results are written to files in the followings lines at user-specified output frequency.

Listing 1: Algorithm for explicit finite element method

```
100     call Massassem(...) !calculate mass_lump(:)
        ...
102     call stabletimestep(...)
        ...
104     do  step = 1, numberOfSteps
          disp = disp + stable_time_step*vel + coef1*acc
106       vel  = vel + coef2*acc

108       !boundary conditions treatments
          !clamped B. C. and simply-support B. C.
110       ...
          call Getforce(...) !calculate resi_global(:)
112       do i = 1, 3*allnodenumGho
            acc(i) = (resi_global(i) - damping*vel(i)*mass_lump(i))/ &
114                  mass_lump(i)
          enddo !i
116       vel = vel + coef3*acc
          if (mod((step-1), printAtEach) .eq. 0) then
118         ...
            !output interested variables at pre-set frequency
120         ...
          endif
122     enddo !step
        ...
```

CHAPTER IV

IMMERSED BOUNDARY METHODS

Non-boundary conforming methods are mainly developed to deal with complex moving boundaries. Many modern applications, the surfaces are in motion, either actively operated or passively responding to the fluid force. There is a significant advantage in using this non-boundary conforming methods as compared to the boundary fitted techniques based on the arbitrary Lagrangian Eulerian (ALE) approach [29]. Although boundary fitted methods are very good for high Reynolds number simulations they are limited to boundaries with small deformations. This can be overcome by solving equations on fixed meshes and accounting for the body, either moving or stationary, which no longer coincides with the grid points by forcing the solution variables at grid points close to the boundary.

4.1 Non-boundary conforming methods

According to Gilmanov et al [31] the non-boundary conforming methods can be divided into two major categories: Cartesian methods and immersed boundary methods. In Cartesian grid methods the boundary is tracked as a sharp interface and computational domain is

modified at the cells which interact with the boundary.

Figure 4.1 shows the different categories of non-boundary conforming grids.

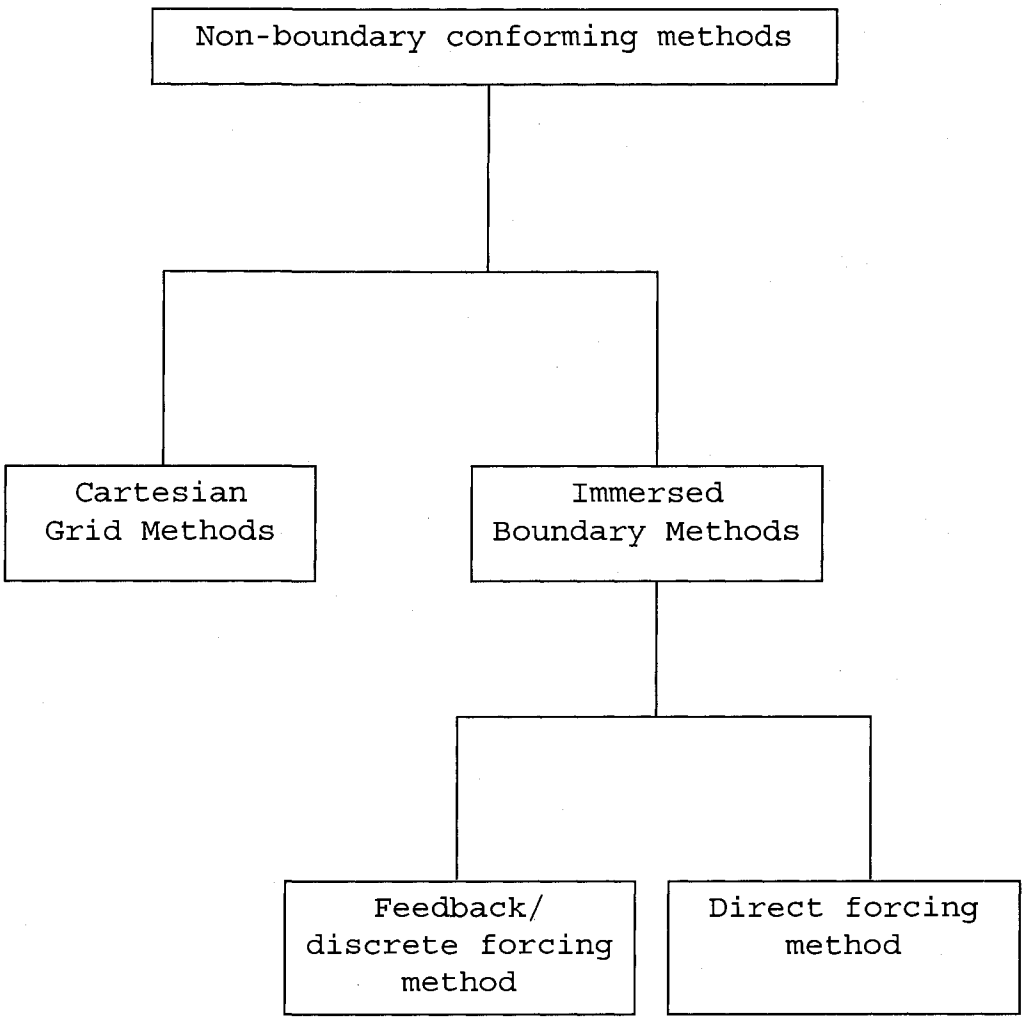


Figure 4.1: Classification of non-boundary conforming methods

This method is used for both and inviscid and viscous flows [32, 33, 34 and 35]. The major disadvantage of this method is that when there are a large number of

interactions between the boundary and the fixed computational grid it leads to a wide variety of cut-cells. In order to apply to 3D configurations complex numerical algorithms need to be devised to treat these cells efficiently.

A Contrary to the above class of non-boundary conforming methods are immersed boundary methods where either a stationary or a moving boundary can be modeled by introducing external force into the momentum equation to satisfy the no-slip boundary condition. The major challenge for this method is the precise imposition of boundary condition on immersed boundaries. There are two methods for the imposing the boundary condition at the immersed boundary. They are

1. Feedback Forcing

2. Direct Forcing

Feedback Forcing:

The external force imposed at discrete surface points x_s according to Goldstein [19] and Saiki and Biringen [20] the forcing $f(x_s, t)$ can be formulated as

$$f(x_s,t) = \alpha \int_0^t (U(x_s,t) - \bar{v}(x_s,t))dt + \beta(U(x_s,t) - \bar{v}(x_s,t)) \qquad 4.1$$

where α and β are constants. $\bar{v}$ is the velocity on the boundary (zero for stationary no-slip boundary). U is the desired fluid velocity at the immersed boundary location. The major drawback of this forcing, however, is those large values of α and β render equation 4.1 stiff and its time integration requires very small time steps according to [14]. To take care of this Goldstein [17] performed simulations at a *CFL* of 0.001 and 0.01. Hence this approach for three dimensional flows will be very costly. A modification to equation 4.1 can be done by calculating the second term in equation 4.1 implicitly in time. This allows to relax the values of α and β in the early stages of flow evolution but there is no unique criterion for this and *ad hoc* judgments are needed.

Direct Forcing

The direct forcing method is another way of representing immersed boundaries. It is a very straight forward and doesn't have any stability issues and therefore used in the current numerical implementation. The direct forcing function is derived by Mohd-Yusof [21]. According to Mohd-Yusof [21] when the Navier-Stokes equation 4.1 is discretized in time we have

$$\frac{u^{n+1}-u^n}{\Delta t} = -\nabla P - \nabla(uu) + \nu\nabla(\nabla u) + f \qquad 4.2$$

A simpler representation of equation 4.2 is

$$\frac{u^{n+1}-u^n}{\Delta t} = -RHS + f \qquad 4.3$$

$$\text{Where } RHS = \nabla P + \nabla(uu) - \nu\nabla(\nabla u)$$

To know the forcing **f** which will yield **u=V** on the immersed boundary, the solution can be obtained from equation (4.3) as

$$f = \frac{V^{n+1}-u^n}{\Delta t} + RHS \qquad 4.4$$

This forcing in equation 4.4 would be true for all points on the immersed boundary but would be zero elsewhere.

That is

$$f = \begin{cases} \dfrac{V^{n+1}-u^n}{\Delta t} + RHS & \text{on} \quad \Omega \\ 0 & \text{elsewhere} \end{cases} \qquad 4.5$$

For a No-slip stationary boundary V is zero and hence the equation 4.5 becomes

$$f = \begin{cases} \dfrac{-u^n}{\Delta t} + RHS & \text{on} \quad \Omega \\ 0 & \text{elsewhere} \end{cases} \qquad 4.6$$

There are no additional terms that are needed to compute **f** in equation 4.6 and hence there is no extra computational time required for computing **f** which is of a distinct advantage.

4.2 Treatment of the immersed boundary (boundary reconstruction)

The previous direct forcing procedure is based on the assumption that the immersed boundary coordinates coincide with the grid points of the computational domain. This assumption becomes invalid for most geometry such as the circular cylinder and a NACA0012 airfoil. In order to overcome this difficulty an interpolation procedure needs to be developed and implemented into the code and tested. There are two different schemes developed by different authors to overcome this difficulty. They are

1. Smearing the forces in the vicinity of the

boundary.

2. Local reconstruction of the solution near the boundary.

Peskin [16] proposed the earlier method but the main drawback of the method is that it has extra dissipation close to the immersed boundary thereby corrupting boundary layer development.

Linear interpolation in one direction

The procedure employed here is to compute the velocity value, in a linear approximation with the point closest to the boundary if the boundary had the particular velocity. We extend this concept by also computing the velocity value V_{inside} (inside the boundary) by linear approximation and set it as a negative value so that the velocity on the immersed surface would have an exact required value. This means just reversing the velocity at the 1^{st} grid point inside the immersed boundary. The method and the expressions are discussed in the next section. The linear interpolations scheme can also be used for turbulent flows because for very fine grids used in turbulent simulations, the nearest grid point from the immersed surface would be in the viscous sub-layer.

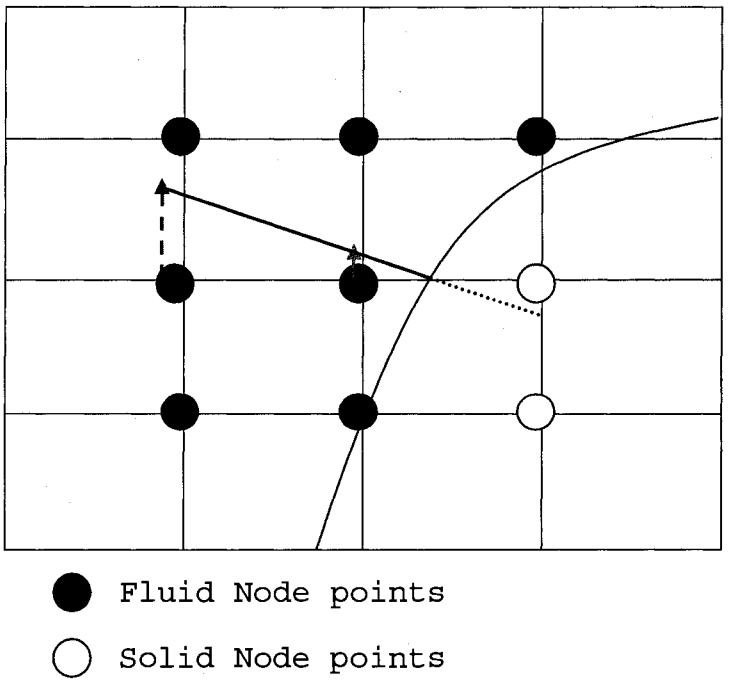


Figure 4.2: Schematic interpretation of one-direction linear interpolation for solid boundary with finite thickness

The use of linear interpolation for tangential velocities can therefore be justified by the law of wall [53].

In figure 4.2 the red dotted line denotes the computed velocity at the grid point and the green line represents the velocity value that needs to be imposed that will render the velocity at the immersed boundary exactly equal to V (zero for stationary boundaries or the boundary point velocity for moving boundaries). Needless to say but the governing equations are all solved inside

at the solid nodes but all the unknowns are set to zero. The grid resolution must be very fine in order to for the linear interpolation procedure to be applicable for high Reynolds number flows. Therefore there are is a need, however, of using high order interpolation schemes for high Reynolds number flows.

The procedure for imposed velocity for one grid point is shown below.

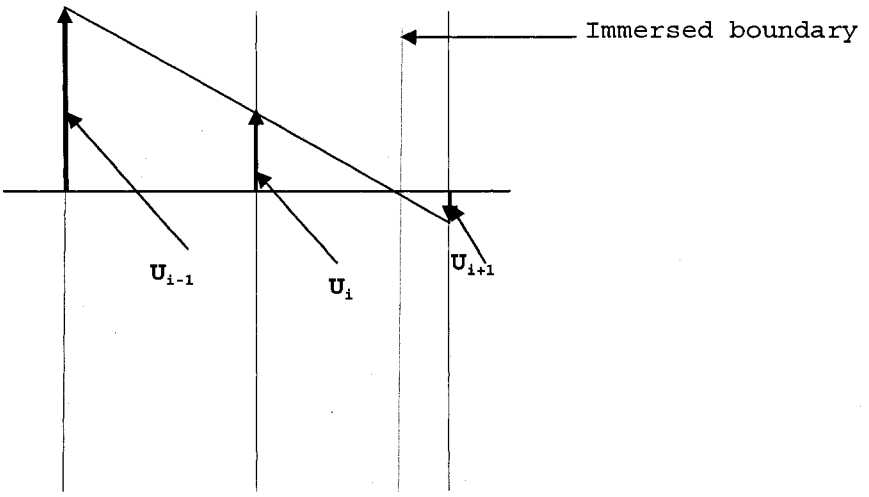


Figure 4.3: Linear interpolation technique for forcing at one grid point

Here we impose the velocities at nodes U_i and U_{i+1} by linear interpolation from the velocity value at U_{i-1} .Let h be the grid spacing in the equidistant zone and h_1 be the distance between the immersed point and the solid node. The linear interpolated velocity values at nodes i and i+1 are

$\underline{U_{i+1}}$

$$\frac{U_{i-1} - V_s}{2 * h - h_1} = \frac{V_s - U_{i+1}}{h_1} \qquad 4.7$$

$$\rightarrow \quad U_{i+1} = V_s(\frac{2h}{2h - h_1}) - U_{i-1}(\frac{h_1}{2h - h_1})$$

$\underline{U_i}$

$$\frac{U_{i-1} - V_s}{2 * h - h_1} = \frac{U_i - V_s}{h - h_1} \qquad 4.8$$

$$\rightarrow \quad U_i = V_s(\frac{h}{2h - h_1}) - U_{i-1}(\frac{h - h_1}{2h - h_1})$$

The immersed boundary method formulation described above is applicable for geometries that have a small finite thickness associated with them. Therefore when applying the appropriate forcing on the grid points to model the no-slip condition, the use of negative forcing on the solid grid points is applicable, as the flow

pattern inside the body will be different from the previous case but the external flow computed will be unaffected. For the infinitesimally thin flat plate, however, this technique of forcing inside the solid body becomes invalid. Hence we discuss two techniques below to model the forcing on the fluid points so that the flow computed is accurate.

Two-sided one direction Interpolation

Let Δ be the mesh spacing and h be the distance of the forced point from the immersed surface. A linear extrapolation utilizing the point on the immersed surface (with specified boundary velocity) and the point just outside the solid (with computed velocity) is then used to obtain the velocity at the forced point inside the solid. Let V_{im} be the desired velocity at the point on the immersed surface and V_u be the computed flow velocity outside the solid surface. Therefore, the velocity at the forcing point V_{uf} is given by

$$\frac{V_u - V_{uf}}{\Delta} = \frac{V_{uf} - V_{im}}{h} \qquad 4.9$$

if $V_{im}=0$ (for stationary bodies)

$$\Rightarrow \frac{V_u - V_{uf}}{\Delta} = \frac{V_{uf}}{h} \Rightarrow V_{uf} = V_u\left(\frac{h}{\Delta + h}\right) \qquad 4.10$$

Similarly an expression for forcing in the downward direction can be computed easily. The expression for V_{df} can be calculated as

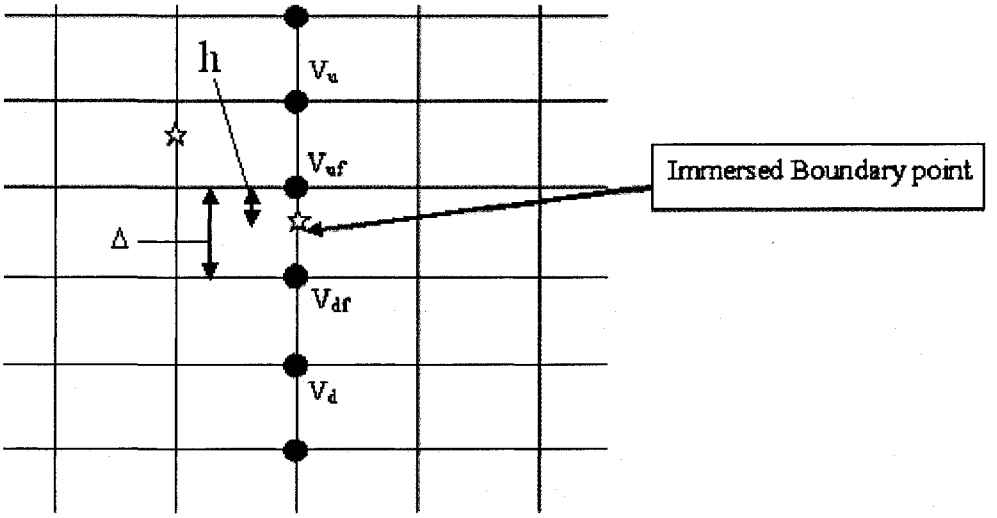


Figure 4.4: Two-sided one-direction interpolation technique

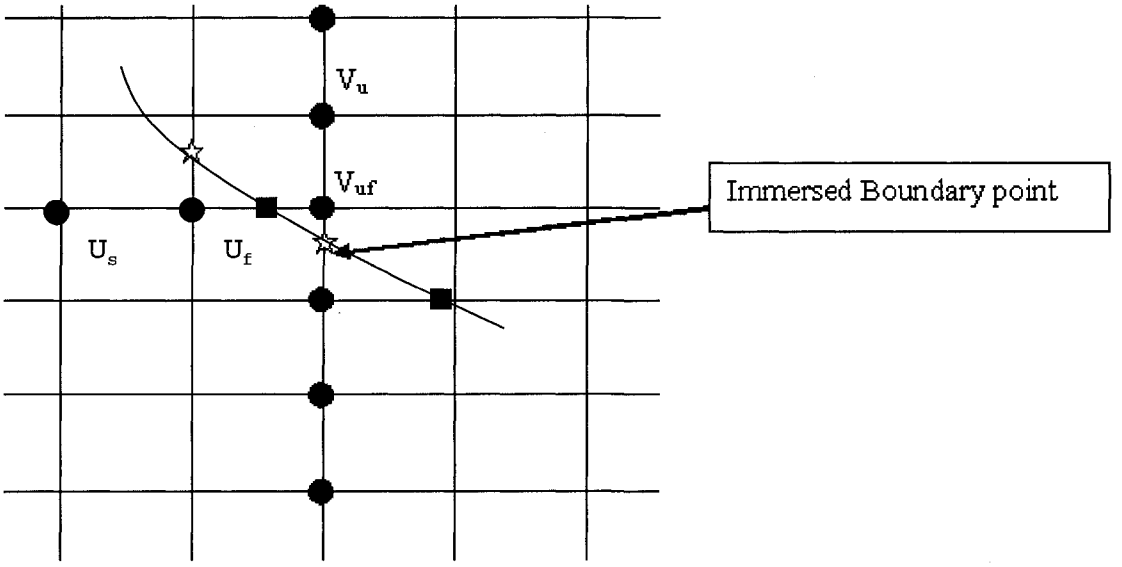
$$V_{df} = V_d\left(\frac{\Delta - h}{2\Delta - h}\right) \qquad 4.11$$

Two Direction one-sided interpolations

Let Δx be the mesh spacing and δx be the distance of the forced point from the immersed surface in the x direction, Δy and δy be the mesh spacing and distance

from the immersed surface in y-direction. The expressions for the velocity forcing can easily be obtained. The expression for x velocity forcing at point U_f is

$$U_f = U_s\left(\frac{\delta x}{\Delta x + \delta x}\right) \qquad 4.12$$



☆ Y direction immersed boundary point position

■ X direction immersed boundary point position

Figure 4.5: Two-direction one-sided interpolation technique

Y velocity forcing at point V_{uf} is

$$V_{uf} = V_u \left(\frac{\delta y}{\Delta y + \delta y} \right) \qquad 4.13$$

Hence for the infinitesimally thin plate case a combination of the above two mentioned techniques are used, depending on the position of the immersed boundary.

CHAPTER V

VALIDATION OF THE FLUID DYNAMIC AND STRUCTURAL DYNAMIC SOLVERS

In this chapter the Navier-Stokes fluid dynamic solver is validated. The major part of the FSI solver being developed is to model fluid structure interactions of a flexible fin attached to the upper surface of a NACA0012 airfoil. The procedure would be to use a structured C-grid for computing flow around the unsteady NACA0012 airfoil with an attached flexible fin. The total pressure force on the fin is calculated at every time step. Then the structural solver takes as an input, the total pressure force acting on the fin calculated from the CFD solver, and the position and velocity are outputted. Then the flexible fin movement due to the pressure loading is modeled using the immersed boundary method described in the previous chapter. The first part of the chapter validates the CFD solver by comparing results around a turbulent NACA0012 airfoil with the published data. The second part of this chapter contains the validation of the structural solver followed by immersed boundary method (IBM) validation in the next chapter.

5.1 Turbulent flow over a NACA0012 airfoil (body-fitted coordinates)

In this section turbulent flow around a NACA0012 at an angle of attack (AoA) of zero and five degrees are computed and validated against the published data [36]. The lift and the drag coefficient computed at various angles of attack are also validated against published results. Different grid densities are used to show that the turbulent flow computed is grid independent.

Numerical details

The three-dimensional incompressible dimensionless RANS equations in the generalized curvilinear coordinates are discretized in space, on a non-staggered mesh using second-order finite difference approximations, and advanced in time using a four-stage Runge-Kutta scheme following a similar procedure described in Chapter III.

The CFL number used in these computations is 1.5 for both the k and ε equations. The numerical solution process was regarded as converged with four to five orders-of-magnitude decrease of residual.

Computational domain and grid

The c-type computational grid used in the turbulent flow over a NACA0012 is generated using Meshpilot [39] software is shown in the figure 5.1. The computational

domain is x/c=30 in length, the airfoil located in the x-axis between 0 and 1, z/c=30 in height, and y/c=0.2 in width.

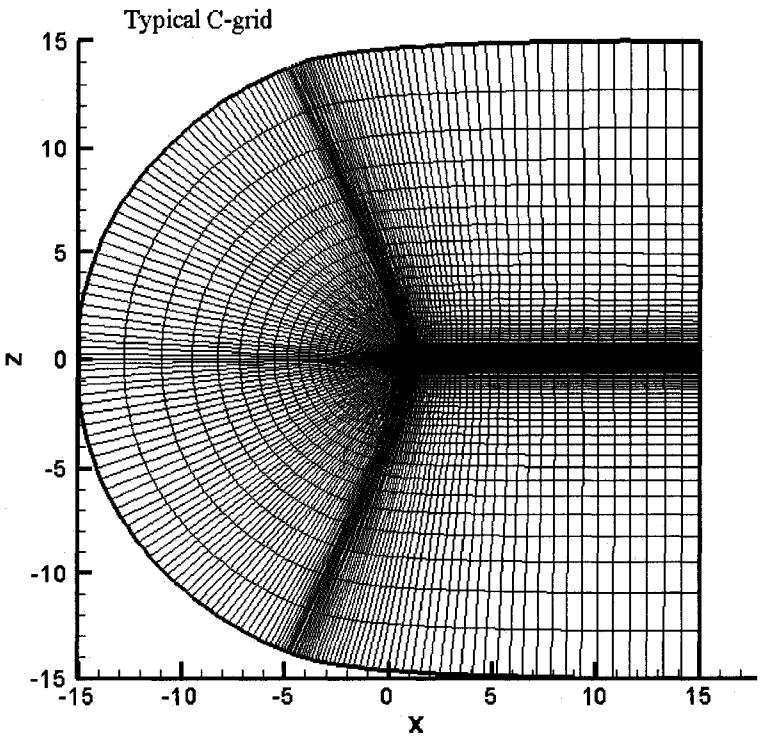


Figure 5.1: C-type computational grid with nodes of 200*95

This means the computational domain extends from 15 chord lengths upstream of the airfoil to 15 chord lengths downstream of the airfoil in the stream wise direction and 15 chord lengths in either direction normal to the airfoil in cross stream direction and 0.2 chord lengths in the span wise direction. The zoomed up plot of the

grid is shown in figure 5.2. It is evident that the grids are clustered near the NACA0012 airfoil and along the centerline of the wake.

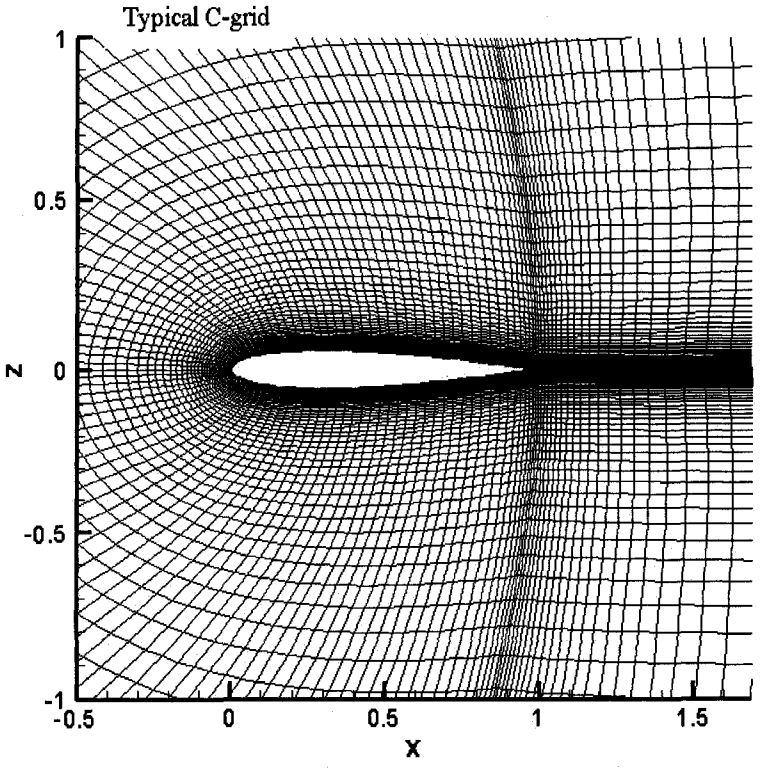


Figure 5.2: Close up of the body-fitted C-grid with a NACA0012 airfoil

Boundary conditions

The inlet boundary conditions for the turbulent flow over a NACA0012 airfoil are assumed uniform for all variables, where U = cos(α), V = sin(α), and $k=\varepsilon=10^{-6}$. On the symmetry boundaries in the span wise direction, the mirror-image reflections for the grid and the flow

variables are used. The exit boundary condition is imposed by assuming zero stream wise diffusion. At the outer boundary, the corresponding free stream recovery is assumed. The wall boundary condition is zero value for all variables, that is $u = v = k = \varepsilon = 0$.

Results and Discussion

In order to first validate the in-house Navier-Stokes finite difference solver grid independence tests are done. With the numerical setup as described in the previous section, turbulent calculations on a NACA0012 airfoil were performed with a number of grids. The results for the time-averaged surface pressure coefficient distributions with four most dense grids used (160x95, 200x95, 200*125 and 240x95) is shown in figure 5.3 for Re=170,000 and angle of attack of five and compared with data.[36]

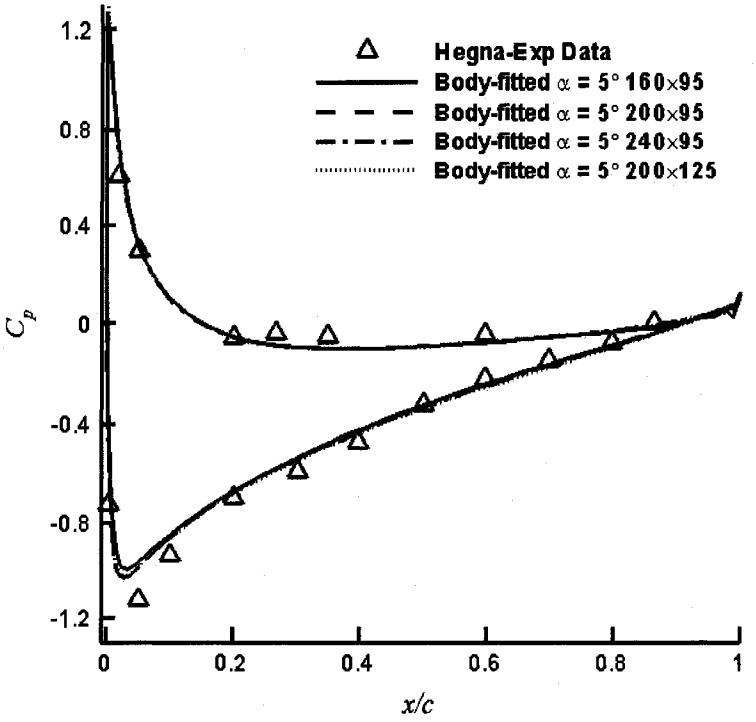
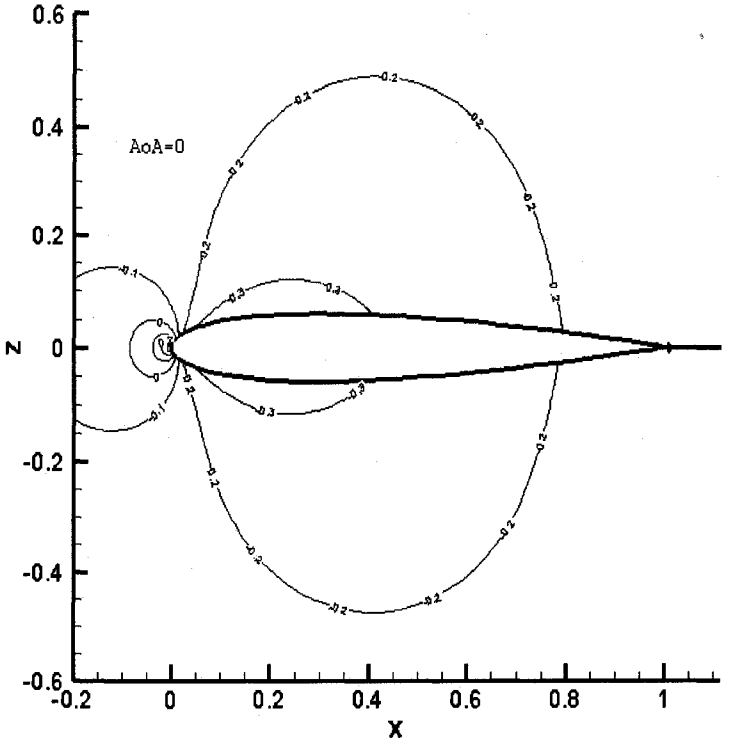


Figure 5.3: Grid independence study by comparing surface pressure distributions

The numerical results show little variance between the solutions and can be regarded as grid-independent. The computational distributions agree well the data along most part of the airfoil surface. The results presented hereafter were obtained by using the fine 200x125 grid.

The pressure contours plots at three different angels of attack using the 200*125 grid are also plotted below in figure 5.4. The calculated lift and drag coefficients for the flow over a NACA0012 airfoil at various angles of attack are calculated and also compared

with the experimental data of Hegna [36] and Jacobs and Sherman [37]. The comparisons can be found in figure 5.5 and 5.6 respectively. The lift and drag force components are computed using trapezoidal integration to sum the total surface stresses obtained from the flow field solution.



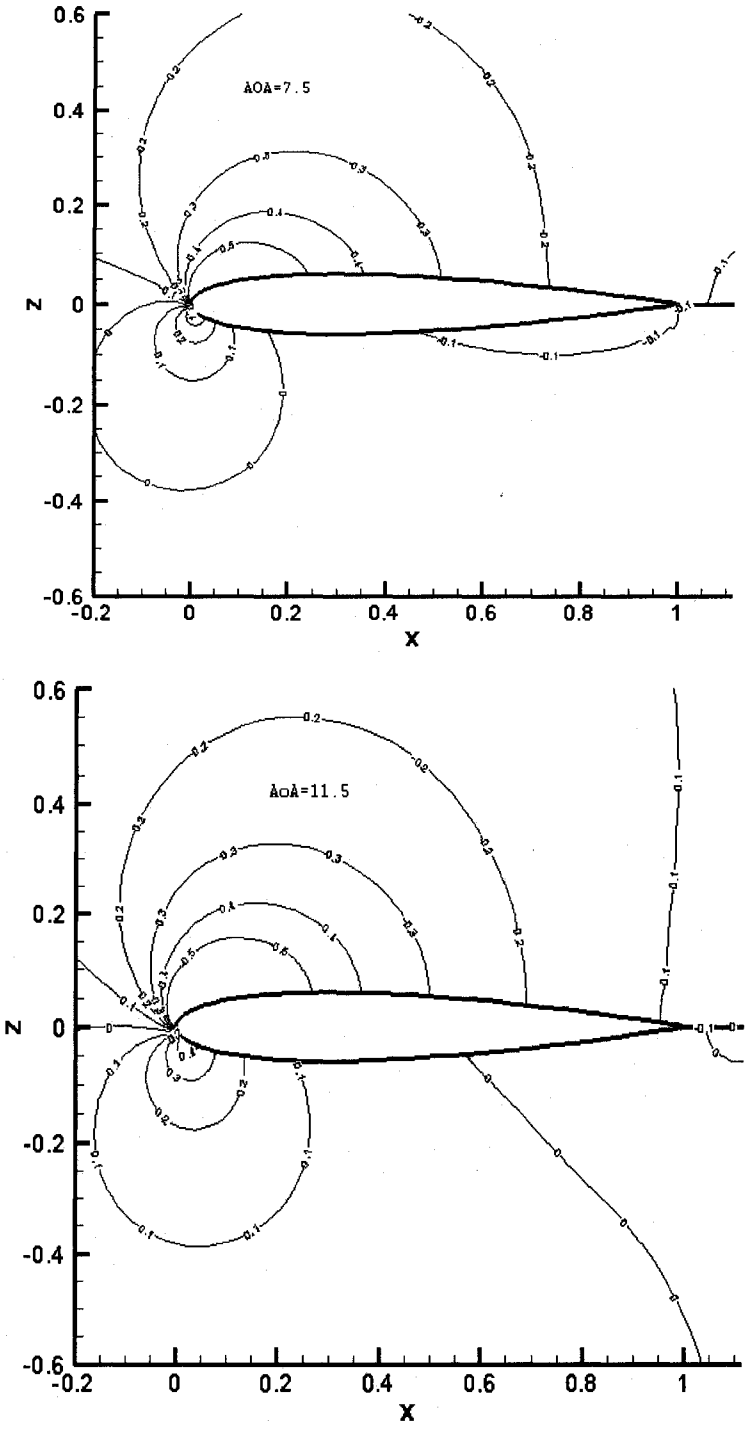


Figure 5.4: Pressure contours for three different angles of attack

It can be observed that the predicted lift coefficient curve from the present model is in good agreement with the experimental measurements [36].

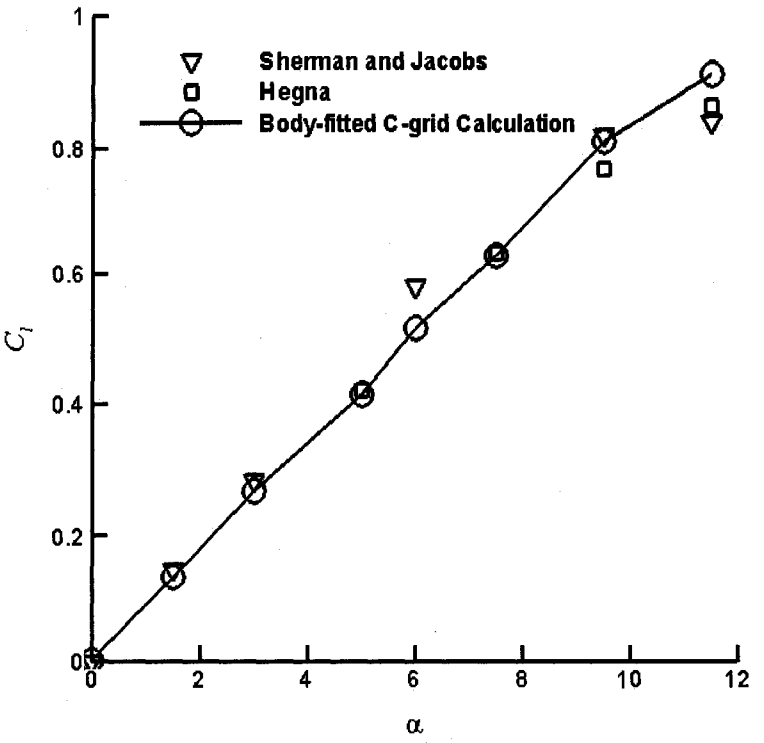


Figure 5.5: Lift coefficient curve for different angels of attack for a NACA0012 airfoil

From figure 5.6 it is clearly evident that the drag coefficient computed from this in-house solver compares well with the published data at different angels of attack. The important fact that needs to be noticed, from figures 5.6, is the accuracy in the drag measurement close to the stall. The lift coefficient computed even

though under predicted compared to the published data close to stall, the error is within a reasonable limit.

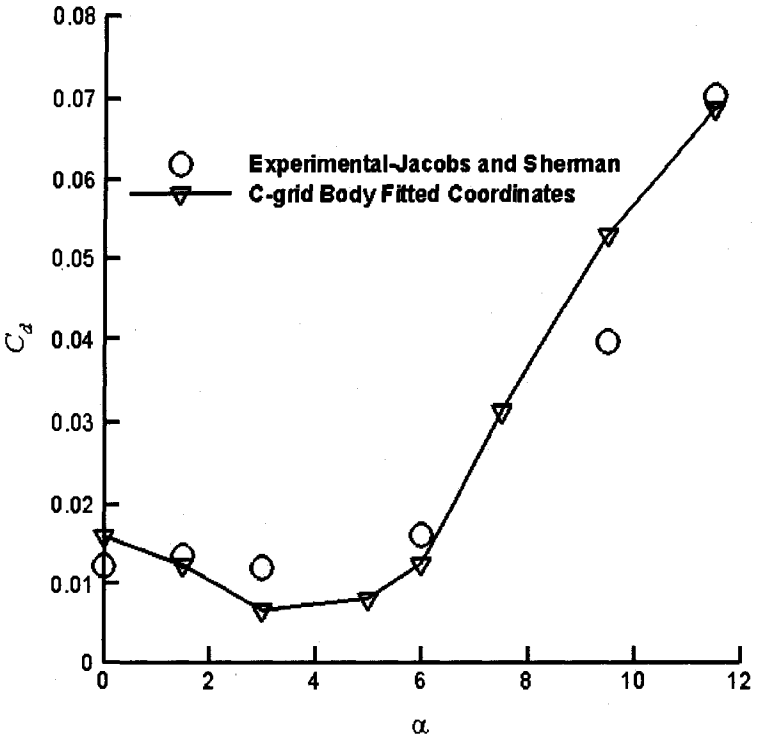


Figure 5.6: Drag coefficient computed at different angles of attack

5.2 Unsteady deflection of a simply supported square plate

The next step in the process is to validate the structural dynamics solver developed by Dr. Yang at the computational engineering physics group of Western Michigan University. In the following case a simple dynamic case is computed using the explicit Newark time

integration scheme. Here a simply supported plate is impacted with a load of certain magnitude. Once the plate is loaded the plate bends due the structural loading. The maximum deflection which is close to the center of the plate is computed at every time step using the subdivision finite element method, as explained in chapter III. This transient variation of the deflection at the center is compared with the analytical solution of Yang and Bhatti [40]

Results and discussion

The simply-supported square plate has the following dimensions. The edge length is 2.438m and thickness of 0.00635 m. Material properties is: density 2500 kg/m^3, Young's modulus $6.90X10^{10}$ N/m^2, Poisson ratio 0.25. The impact loading with the magnitude of 47.89 N/m^2 is considered. In figure 5.7 the time varying central deflection of simply supported beam is plotted and compared with published data of Yang and Bhatti [40].

The results show that both the computed solution the analytical solution of Yang and Bhatti [40] collapse into a single curve over time.

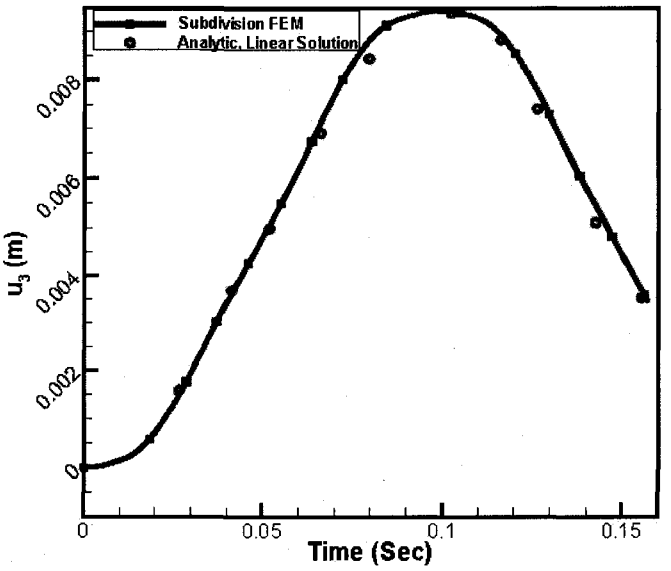


Figure 5.7: Time variation of central deflection of a simply supported plate

CHAPTER VI

VALIDATION OF THE FINITE DIFFERENCE IMMERSED BOUNDARY NAVIER-STOKES SOLVER

The immersed boundary method of Fadlun [14], Mohd-yusof [21] is applied in this chapter. In order to evaluate the accuracy and the applicability of immersed boundary method a number of simulations are performed on different geometries and validated. This analysis shows the accuracy of the immersed boundary method developed here and also gives confidence that results to new problems will be valid. The following steps are followed while validation the immersed boundary method used in combination with the Navier-Stokes solver.

- Step 1: Validate the steady state solver in combination with the Navier-Stokes solver. In order to accomplish this task the steady flow over a circular cylinder is validated against a published numerical result [41]. In addition to this the steady flow over a NACA0012 airfoil is computed and validated against a published result[43]
- Step 2: Validate the unsteady solver in combination with the immersed boundary method. In order to accomplish this task the unsteady flow over a circular cylinder is computed and validated against other

published results. [42]

- Step 3: Validate the turbulent solver in combination with the immersed boundary method. Turbulent flow over a NACA0012 airfoil is computed using the immersed boundary method and validated against published results [36].

The major reason behind applying the immersed boundary method for modeling FSI over a flexible fin attached to a NACA0012 airfoil is to account for the fin movement just by using body forces rather than tradition grid movement algorithms. The fin used in the numerical simulation is considered infinitesimally thin (thickness=0). In both the previous steps the boundary that needs to be forced in the momentum equation has non-zero thickness. This means there were solid nodes inside the boundary, which facilitates the use of negative forcing while interpolation as explained in chapter IV on the solid boundary. This cannot be done for an infinitesimally thin object and hence combined one-sided two-direction forcing or two-sided one-direction forcing explained earlier are used.

Step 4: Validate the immersed boundary method for infinitesimally thin plates. The flow over an infinitesimally thin plate at high angle of attack is validated with a published result [48].

- Step 5: Validate the solver for moving boundaries. This means that the immersed boundary method in conjunction with the finite difference Navier-Stokes solver needs to be validated for moving boundaries. According to the knowledge of the author there weren't any published data that used zero thickness moving boundaries in their numerical computations. Hence the zero thickness moving immersed boundary cases were validated with the work of Liou and Pantula [13], where a commercial code Fluent was used to model flow around an unsteady flexible flat plate with a prescribed boundary motion.

Step 1: Steady state solver combined with immersed boundary method

6.1 Flow over a circular cylinder

The flow over a circular cylinder has been studied quite extensively and a number of numerical and experimental data exist. This flow is very attractive because it varies with Reynolds number and it's not very easy to simulate on non-boundary fitted Cartesian grids. The flow is stationary at around Reynolds number of 40 where two symmetrical standing vortices are formed but remain attached to the cylinder. As the Reynolds number is increases a wavy nature of the tail can be observed and gradually the vortices stretch and eventually at high

Reynolds number alternating vortex shedding called the Karman vortex street can be observed.

Computational Domain Details

In this section a description about the domain and mesh used in the case I of simulations is given. A simple Cartesian grid is selected for this case. X and Y are the stream wise and cross stream directions respectively. A symmetry boundary condition is used in the Z-direction. A circular cylinder of 0.4 m diameter is deployed in a channel of 20m X 20m X 0.05m. The circular cylinder is placed at a center of the channel. The figure 6.1 below gives a schematic idea of the domain. The computational domain is x/L = 20 in length, z/L = 20 in height, and y/L = 1 in width, and the number of grid nodes are 181x181x5. In zone 1 from x=-10 to -0.5 grid stretching factor of 1.15 is used and 40 grids are constituted in that region. In zone 2 From x=-0.5 to 0.5 where the cylinder is located an equidistant grid consisting of 100 nodes is used giving a grid spacing of 0.01 and again in zone 3 from x=-0.5 to 10 like in zone 1 a grid stretching factor of 1.15 is used and 40 grids are constituted in that region. Similar in y direction zone 1 consists of 40 grids from -10 to -0.5 with grid stretching factor of 1.15 and zone 2 consists from -0.5 to 0.5 consisted of

100 nodes with an equidistant grid spacing of 0.01 in this zone. Similar to zone 1 in y-direction the zone 3 consists of a grid form 0.5 to 10 with 40 nodes and a grid stretching factor of 1.15. The CFL number is set to 1.5. All the residuals have reduced at least five orders of magnitude, and the values are all smaller than 10^{-6}.

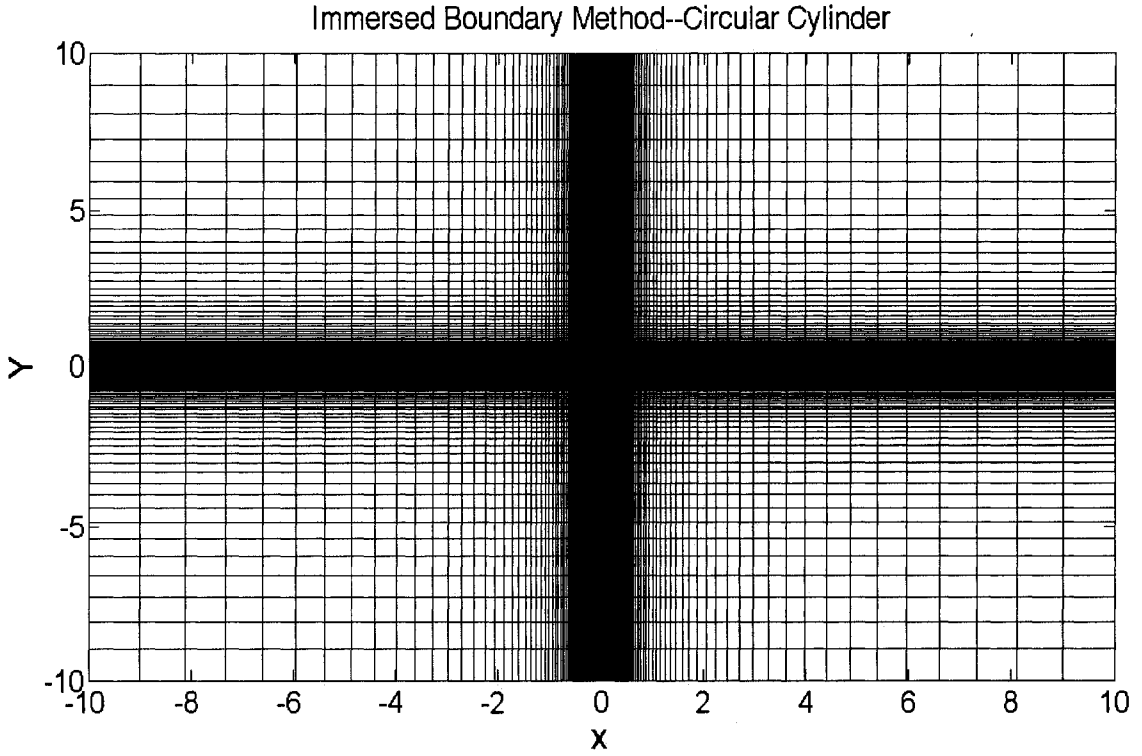


Figure 6.1: Circular cylinder domain for 181*5*181 mesh

The pressure distribution curves in figure 6.2 are compared the published result of Fornberg [41]. Grid independence studies done using three different meshes showed that a reasonable number of forcing points are

required for accurate prediction of the boundary layer.

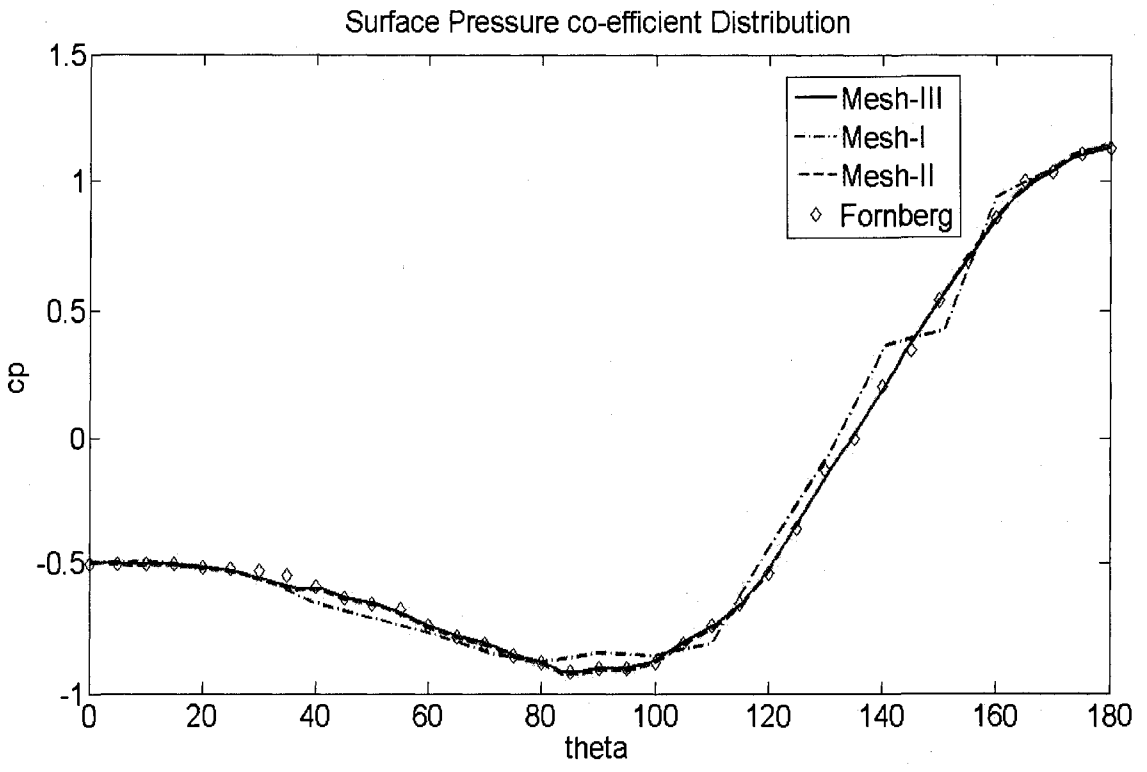


Figure 6.2: Surface pressure distribution over a Circular Cylinder at Re=40

From figure 6.2 we can observe that with Mesh I which has the coarsest resolution of all and there are wiggles or oscillations on the surface pressure curves. This is because of the fact that only 40 immersed boundary points were used to define the surface. Mesh II and III exhibit no wiggles because sufficient number of immersed points were used to define the surface geometry.

Mesh II uses 80 and Mesh III uses 160 immersed boundary points respectively to model the cylinder.

Figures 6.3 and 6.4 show the surface pressure coefficient (Cp) and velocity contours along the cylinder surface at *Reynolds number 40* using the immersed boundary technique. From the pressure contour and the x velocity contours in figures 6.3 and 6.4 one can see the flow is steady with separation bubble behind the cylinder.

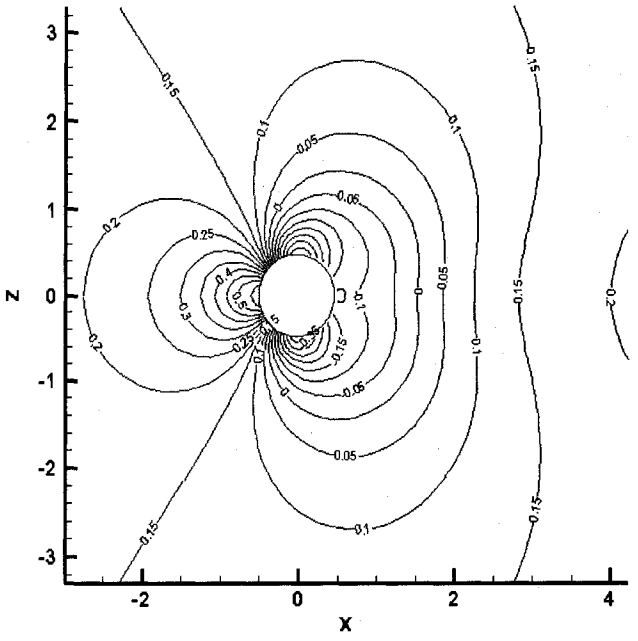


Figure 6.3: Pressure contours over a circular cylinder at Re=40

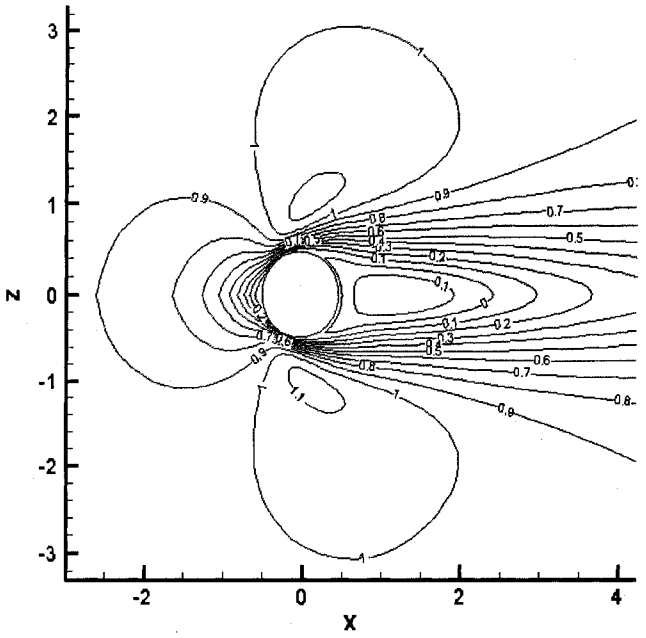


Figure 6.4: Velocity contours over a circular cylinder at Re=40

6.2 Laminar flow over a NACA0102 airfoil

The laminar flow past a NACA0012 airfoil with zero degree angle of attack is simulated at Reynolds number of 500 based on the chord length. The computational domain is x/L = 21 in length, the airfoil is located on x-axis between 0 and 1, z/L = 20 in height, and y/L = 1 in width, and the number of grid nodes are 281×281×3. Figure 6.5 shows the computation domain and mesh used in the simulations. In zone 1 from x=-10 to -0.5 grid stretching factor of 1.15 is used and 40 grids are constituted in that region. In zone 2 From x=-0.5 to1.5 where the airfoil is located from x=0 to 1 an equidistant grid

consisting of 200 nodes is used giving a grid spacing of 0.01 and again in zone 3 from x=-1.5 to 11 like in zone 1 a grid stretching factor of 1.15 is used and 40 grids are constituted in that region. Similar in y direction zone 1 consists of 40 grids from -10 to -1 with grid stretching factor of 1.15 and zone 2 consists from -1 to 1 consisted of 200 nodes with an equidistant grid spacing of 0.01 in this zone. Similar to zone 1 in y-direction the zone 3 consists of a grid form 1 to 10 with 40 nodes and a grid stretching factor of 1.15. The CFL number is set to 1.5. The numerical solution process was regarded as converged with four to five orders-of-magnitude decrease of residual.

A grid independence study was done to check the order of accuracy of the code. The pressure distributions on the airfoil surface are computed and compared with the documented results from Ypeng et.al [43] as well as a boundary-fitted solution computed using commercial code Fluent. Then the obtained values of lift and drag coefficient are compared with documented values of D. Yu et al [49]. Figure 6.6 shows the pressure coefficient (Cp) along the airfoil surface at Re of 500 using linear extrapolation.

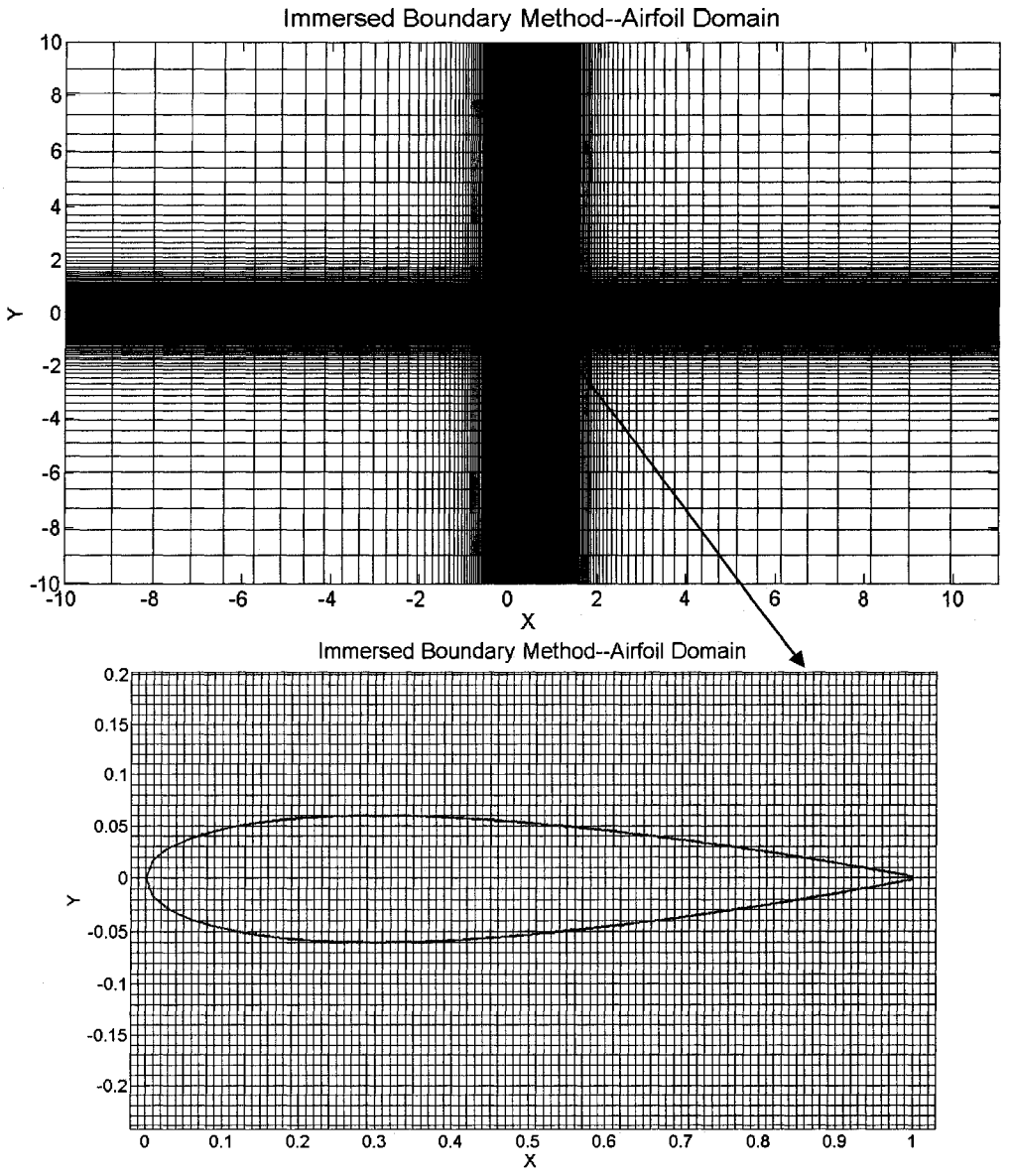


Figure 6.5: Airfoil domain for 281*3*281 mesh

In the present study, both the pressure and velocity at the surface are linearly interpolated from the nearest cells. The results are very close to those obtained from

the boundary-fitted grid solution. The grid resolution study performed to analyze the accuracy of the linear interpolation scheme showed that the solution is grid independent. The pressure and velocity contours from the semi finest 281*3*281 grid using the immersed boundary technique are also shown in figures 6.7 and 6.8 respectively.

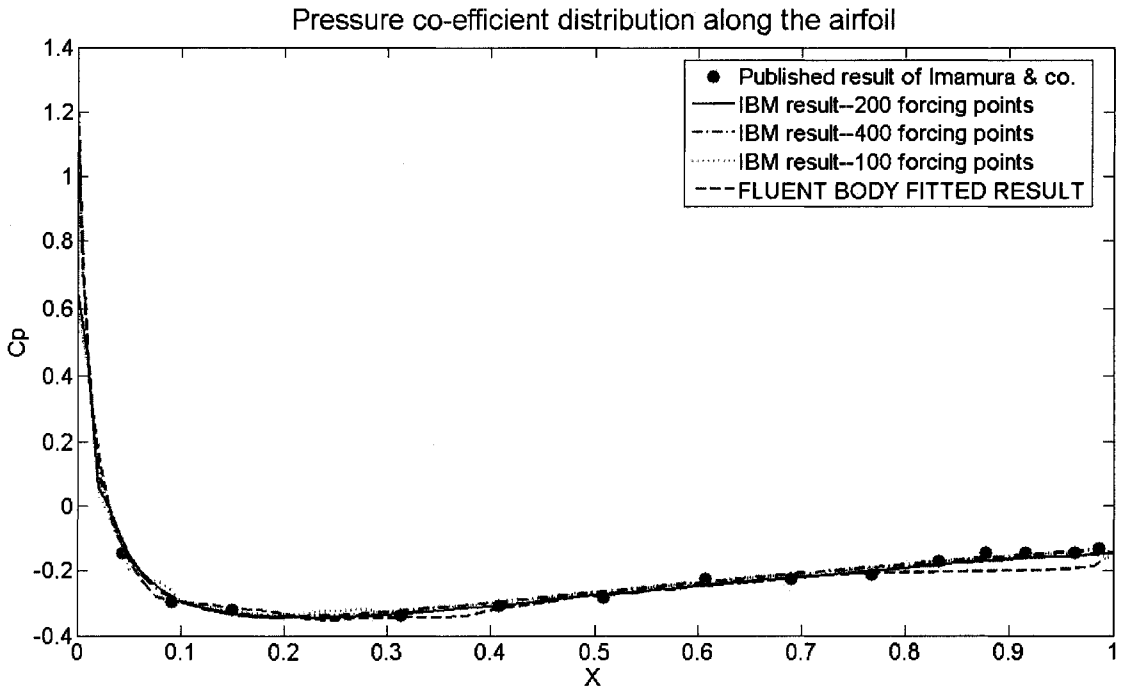


Figure 6.6: Pressure distributions along the airfoil

Drag Co-efficient

The computed drag co-efficient is compared with the documented results of Dr. Yu [49]. The present value drag

coefficient of 0:1739 compares very well with the results reported in reference [44] where a drag coefficient of 0:1762 was obtained using the Navier-Stokes equation-based finite diference method, and the drag coefficient of 0:1717 was obtained when using Power flow code developed by EXA Corporation, which is based on the lattice Boltzmann equation method.

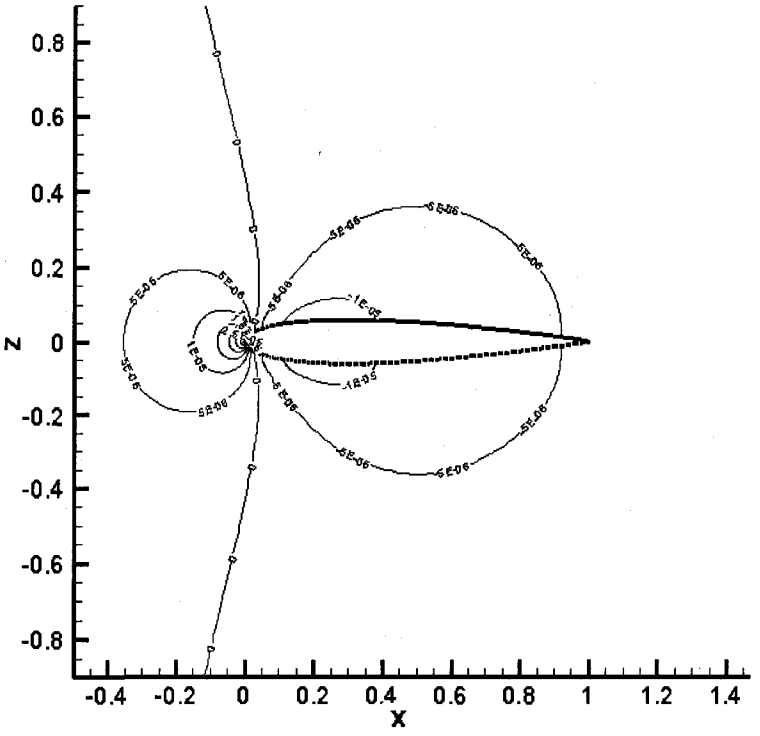


Figure 6.7: Pressure contours over a NACA0012 airfoil using immersed boundary method at Re=500

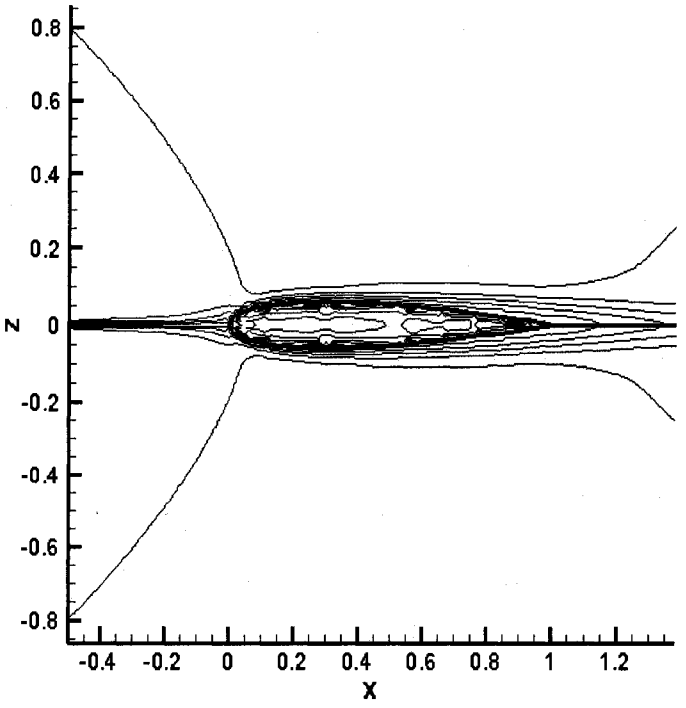


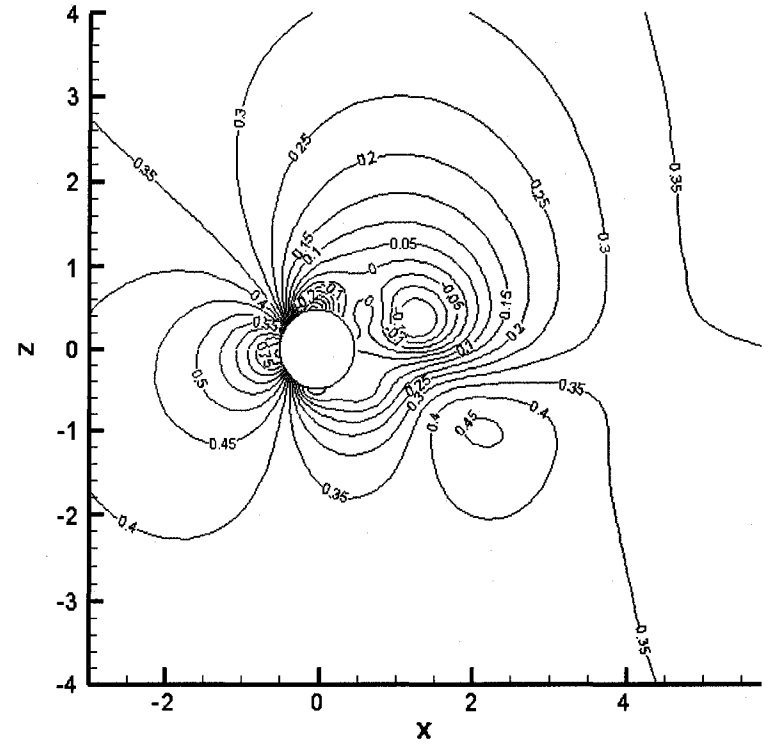
Figure 6.8: Velocity contours over a NACA0012 airfoil using immersed boundary method at Re=500

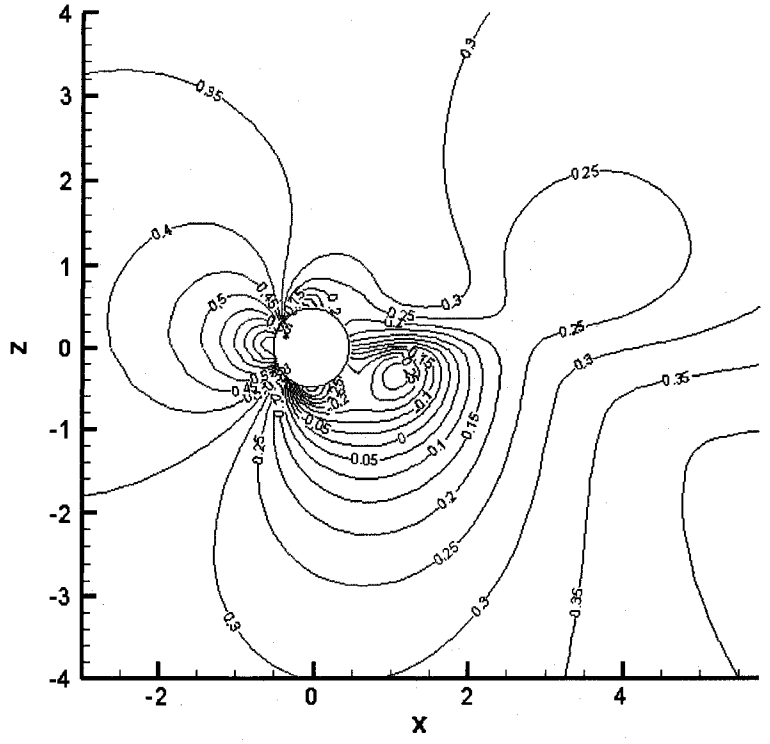
Step 2: Unsteady solver coupled with the immersed boundary method

6.3 Unsteady flow over a circular cylinder at Reynolds number of 200

From the pressure and velocity contours from figures 6.3 and 6.4, it is clear that the flow around a circular cylinder is steady with separation bubble behind the cylinder. As the Reynolds number is increase the wake becomes unstable due to perturbations. The cylinder wake becomes unstable and in stabilizes at $Re \geq 47$. The velocity and pressure contours are displayed below in

figures 6.9 and 6.10.





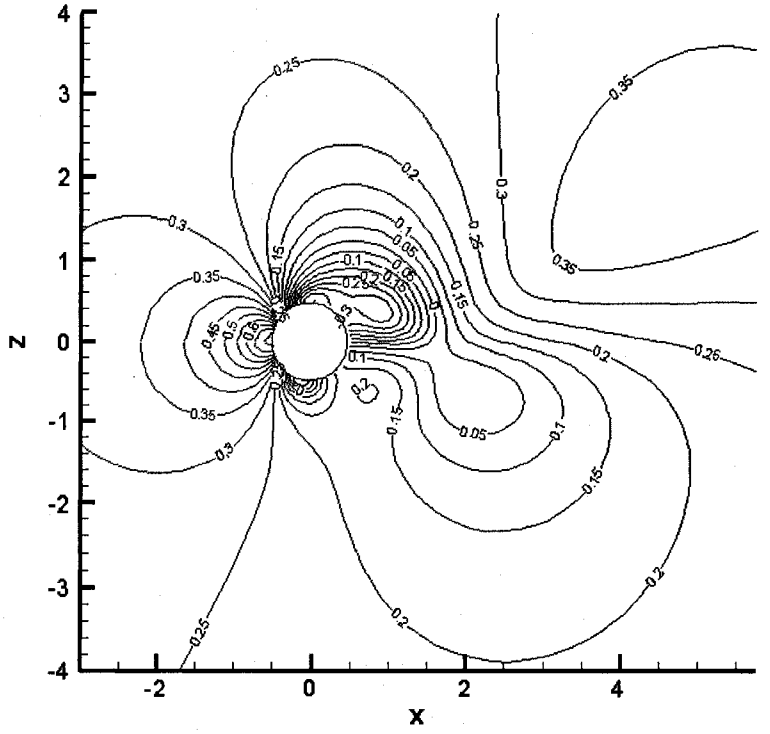
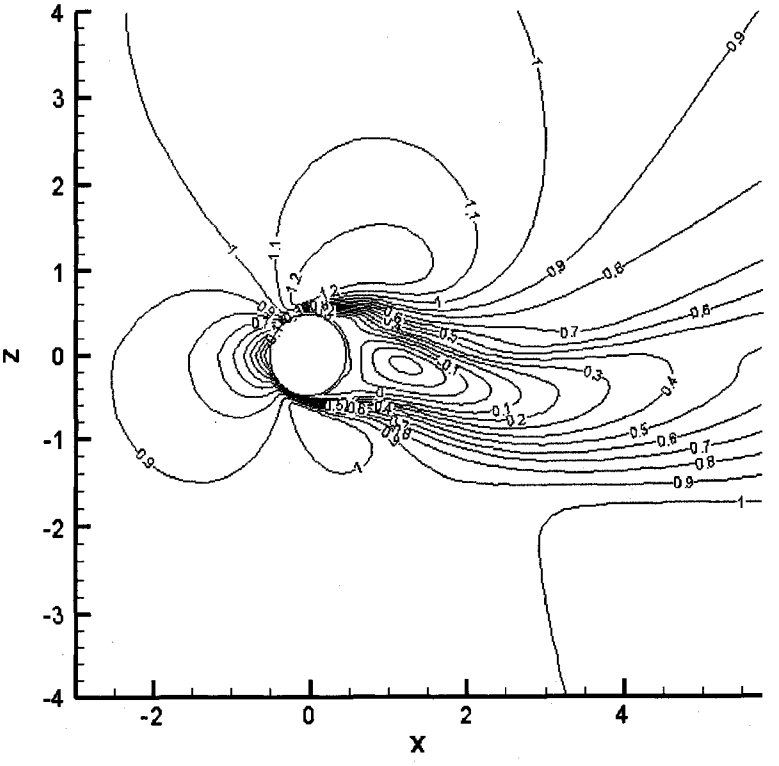
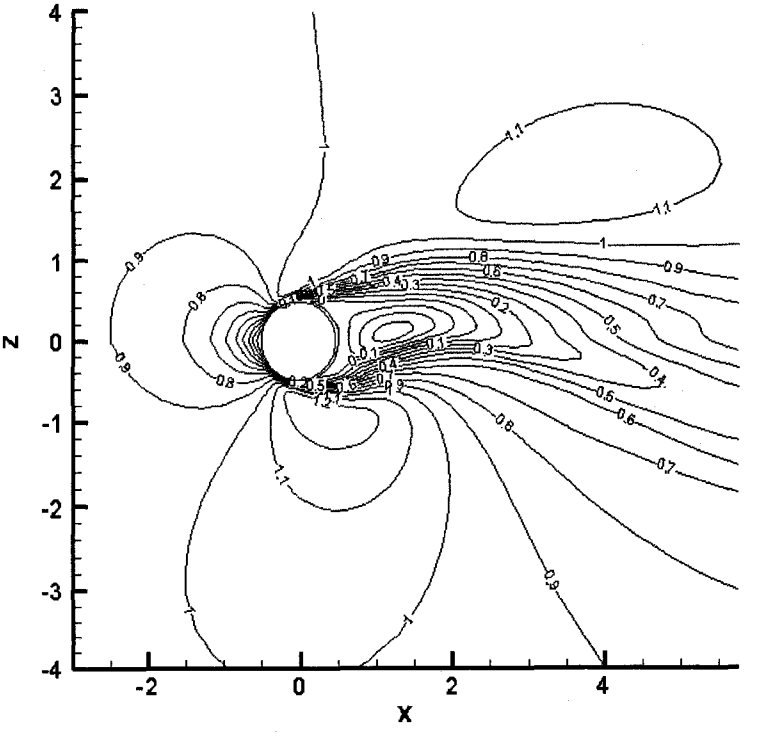


Figure 6.9: Contours of pressure at three different times around a circular cylinder at Re=200 using immersed boundary technique (t=30, 40 and 50 sec)

In figures 6.9 and 6.10 one can see the velocity contours and pressure contours in the near wake of a cylinder at three different time steps. We can clearly see that the present immersed boundary finite difference Navier-Stokes method accurately captures the wake deformation around a complex bluff body like the circular cylinder. Figure 6.11 shows the time evolution of the lift co-efficient measured at Reynolds number of 200. From this lift history a very important quantity for unsteady bluff bodies called strouhal number can be computed.


4
3
2
1
0
-1
-2
-3
-4
Z
-2
0
2
4
X


4
3
2
1
0
-1
-2
-3
-4
Z
-2
0
2
4
X

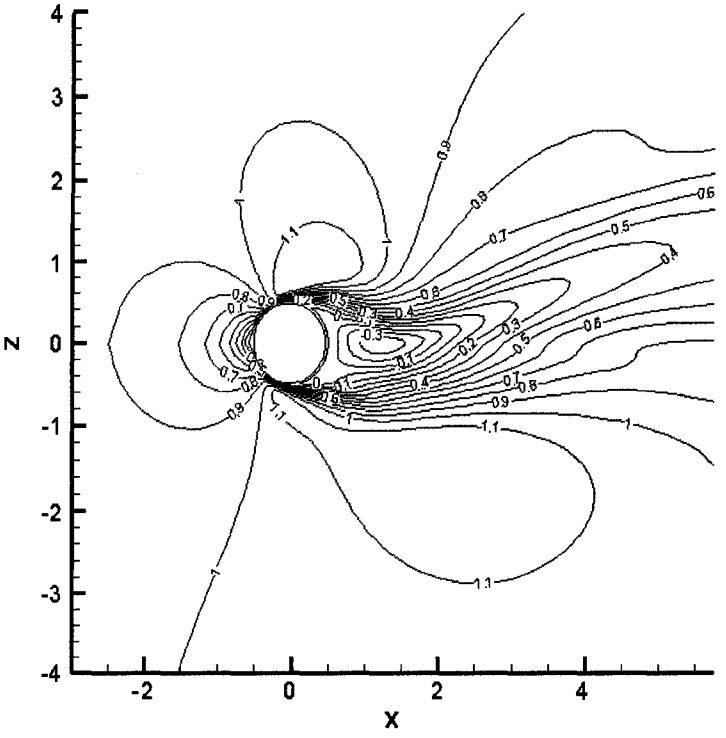


Figure 6.10: Velocity contours at three different time steps of 30, 40 and 50 sec

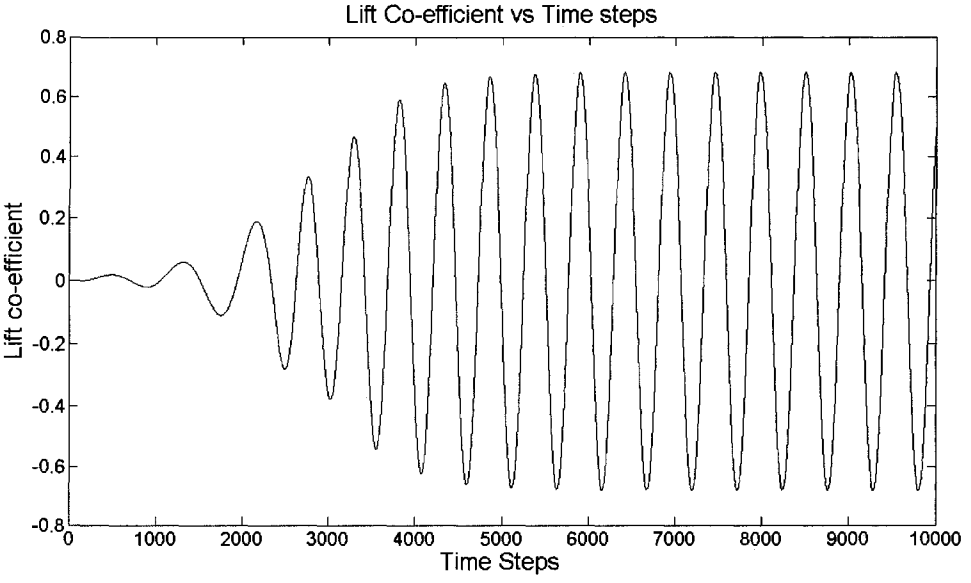


Figure 6.11: Lift coefficient variation with time

Strouhal number: The Strouhal number is defined as the dimensionless frequency with which the vortices are shed behind the body

$$st = \frac{f.d}{U_\infty} \qquad 6.1$$

where f is the vortex shedding frequency. St tabulated in table 1 shows a good agreement with the published result from Kiris and Kwak [42].

Method	Reynolds Number	Parameter Compared	Comments
Unsteady Solver-current method	200	Strouhal number, Lift variation	0.19, ±0.68
Kiris and Kwak [42]	200	Strouhal number, Lift variation	0.184, ±0.67

Table 1: Strouhal number and lift variation

Step 3: Turbulent model coupled with the immersed boundary method

6.4 Turbulent flow over a NACA0012 airfoil at Reynolds number of 170000

In this part of the chapter, turbulent flow over a NACA0012 airfoil is studied using the immersed boundary method. This part of the work validates the immersed boundary method application for turbulent flows. The Reynolds number of the flow based on the chord length of the flow is fixed at 170,000 with the angle of attack of 7.5^0. The two-equation low Reynolds number K-ε model of Launder and Sharma [38] is used to resolve the turbulent properties in the flow region. The k and ε transport equation as well the damping functions used by Launder and Sharma [38] are as indicated in chapter II.

Computational domain details

The computational domain is x/L = 21 in length, the airfoil is located on x-axis between 0 and 1, z/L = 20 in height, and y/L = 1 in width, and the number of grid nodes are 281×281×3. Figure 6.12 shows the computational domain and grid used in these simulations. In zone 1 from x=-10 to -0.5 grid stretching factor of 1.15 is used and 40 grids are constituted in that region. In zone 2 from x=-0.5 to1.5 where the airfoil is located from x=0 to 1, an

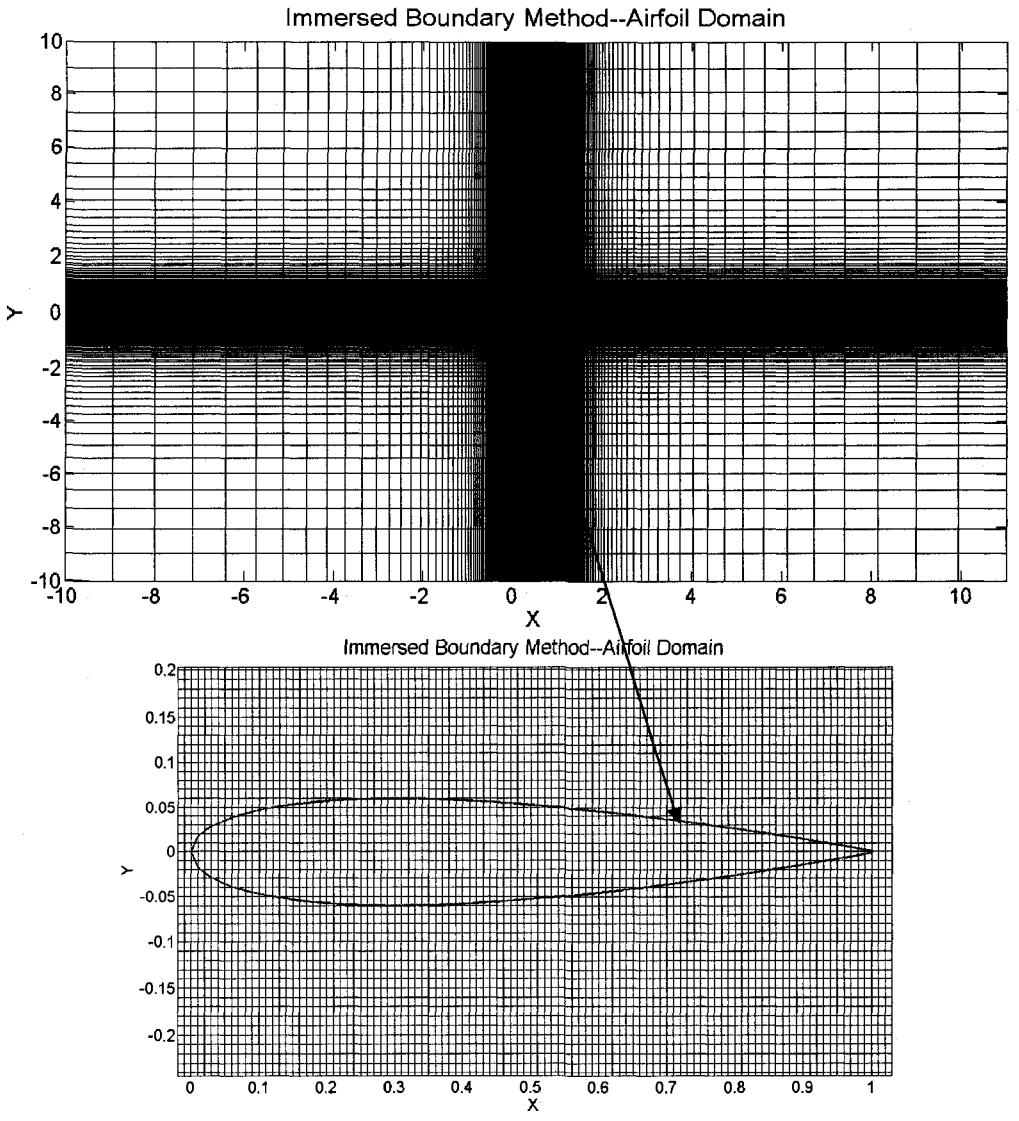


Figure 6.12: Cartesian airfoil domain for 281*3*281 mesh

equidistant grid consisting of 200 nodes is used giving a grid spacing of 0.01 and again in zone 3 from x=-1.5 to 11 like in zone 1 a grid stretching factor of 1.15 is used and 40 grids are constituted in that region. Similar in y direction zone 1 consists of 40 grids from -10 to -1

with grid stretching factor of 1.15 and zone 2 consists from -1 to 1 consisted of 200 nodes with an equidistant grid spacing of 0.01 in this zone. Similar to zone 1 in y-direction the zone 3 consists of a grid form 1 to 10 with 40 nodes and a grid stretching factor of 1.15. The CFL number is set to 1.5. All the residuals have reduced at least five orders of magnitude, and the values are all smaller than 10^{-6}.

Boundary conditions

The inlet boundary conditions for the turbulent flow over a NACA0012 airfoil are assumed uniform for all variables, where U = cos(α), V = sin(α), and k=ε=10^{-6}. On the symmetry boundaries in the span wise direction, the mirror-image reflections for the grid and the flow variables are used. The exit boundary condition is imposed by assuming zero stream wise diffusion. At the outer boundary, the corresponding free stream recovery is assumed.

Results and discussion

In order to validate the turbulent flow over a NACA0012 airfoil using the immersed boundary method the time-averaged surface pressure coefficient distribution over the airfoil for Reynolds number of 170,000 and angle

of attack of 7.5^0 is compared with the experimental data of Hegna [36]. From figure 6.15 the numerical results show little variance between the solutions and can be regarded as grid-independent. The computational distributions agree well with the data along most part of the airfoil surface as shown in the figure 6.15. The pressure and velocity contours are displayed in the figures 6.13 and 6.14 respectively.

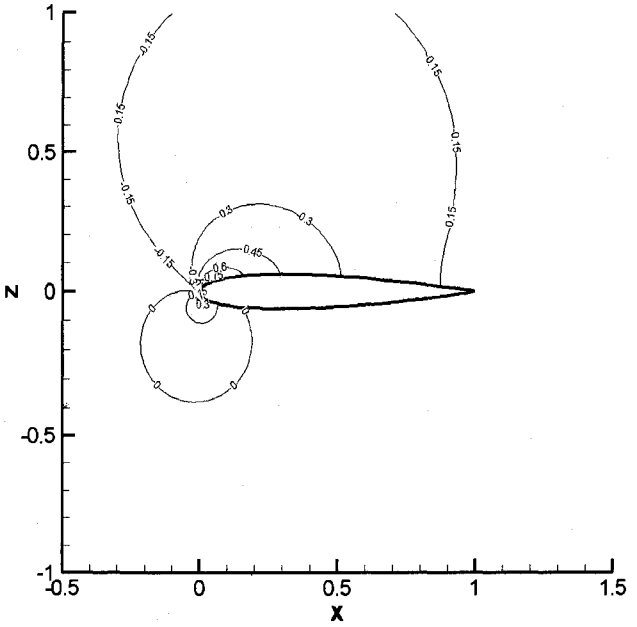


Figure 6.13: Pressure contours over a turbulent NACA0012 airfoil at Re=170,000 using immersed boundary technique

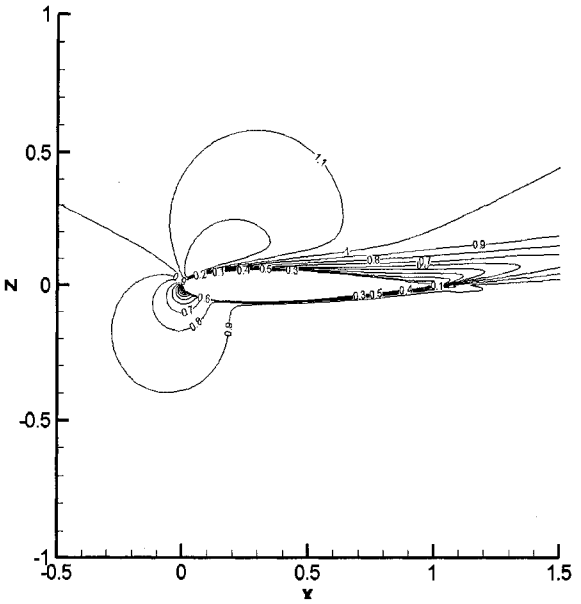


Figure 6.14: Velocity contours over a turbulent NACA0012 airfoil at Re=170,000 using immersed boundary technique

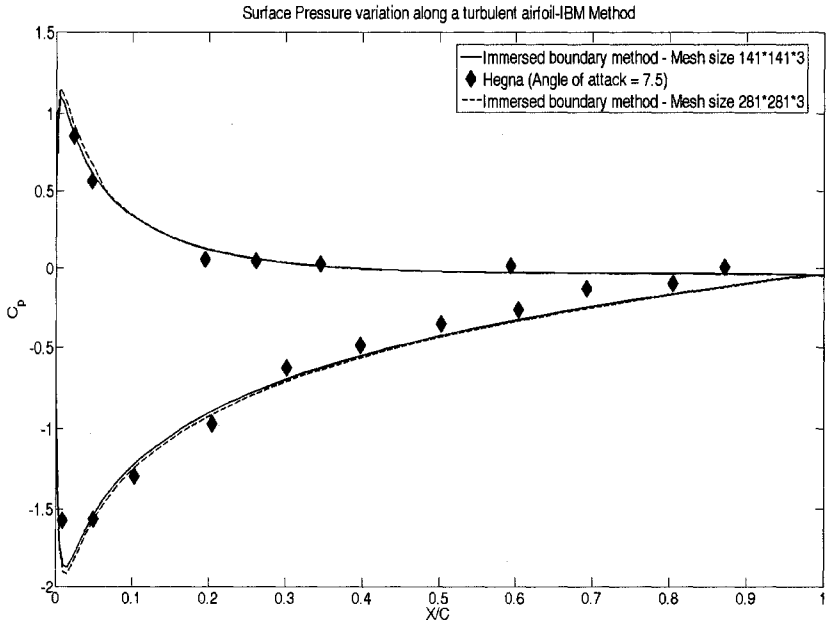


Figure 6.15: Grid independence test on a NACA0012 computed using immersed boundary method

Step 4: Steady infinitesimally thin flat plate

6.5. Laminar flow over an infinitesimally thin flat plate

In this part of the chapter flow over a infinitesimally thin flat plate at angle of attack of 30 is investigated for Reynolds number of 100. The two dimensional flow over the flat plate is solved using the immersed boundary method and validated against the flow over a rectangular flat plate with experimental measurements from a companion tow-tank equipped with stereo digital particle image velocimetry (DPIV) and a six-axis force tensor.

Computational domain and boundary conditions

The laminar flow past a static flat plate with angle of attack $\alpha = 30°$ is simulated using the immersed boundary method on a Cartesian grid at Reynolds number, based on the chord length c, of 100. The CFL number is set to 1.5. The 2-D solutions are sought, and the computational domain is $x/c = 20$ in length, the plate is located on x-axis between 0 and 1, and $z/c = 20$ in height. The computational domain and grid used in this simulation is shown in the figure 6.16. The inlet boundary conditions for this high angle of attack flow over a infinitesimally flat plate are assumed uniform for all variables, where $U = \cos(\alpha)$, $V = \sin(\alpha)$.

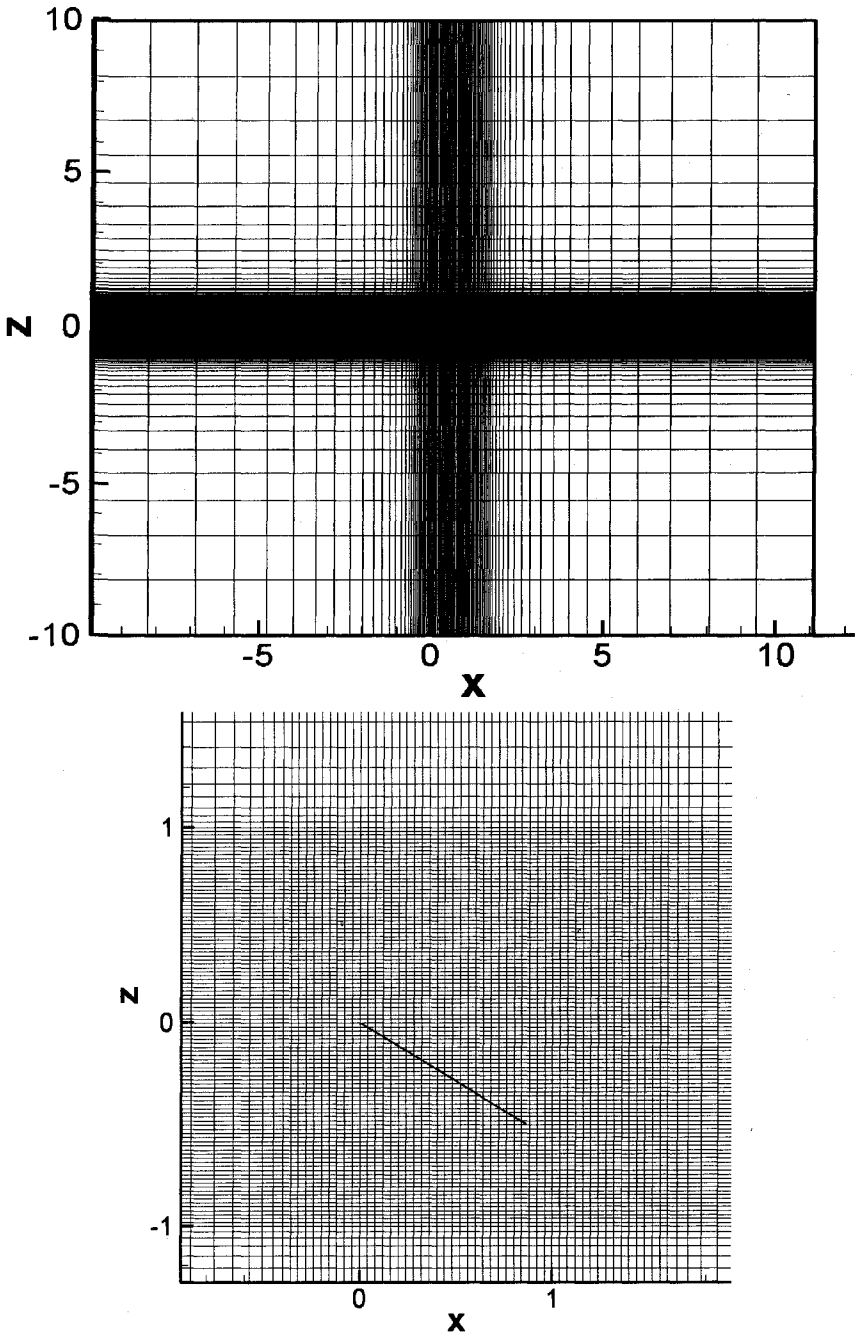


Figure 6.16: 2D grid with the infinitesimally thin flat plate

The exit boundary condition is imposed by assuming

zero stream wise diffusion. At the outer boundary, the corresponding free stream recovery is assumed.

Grid independence study

The results for the time-averaged surface pressure coefficient distributions with two grids (91x181, 141x231) is shown in figure 6.17 for Re=100 and angle of attack of 30^0. The numerical results show little variance between the solutions and can be regarded as grid-independent.

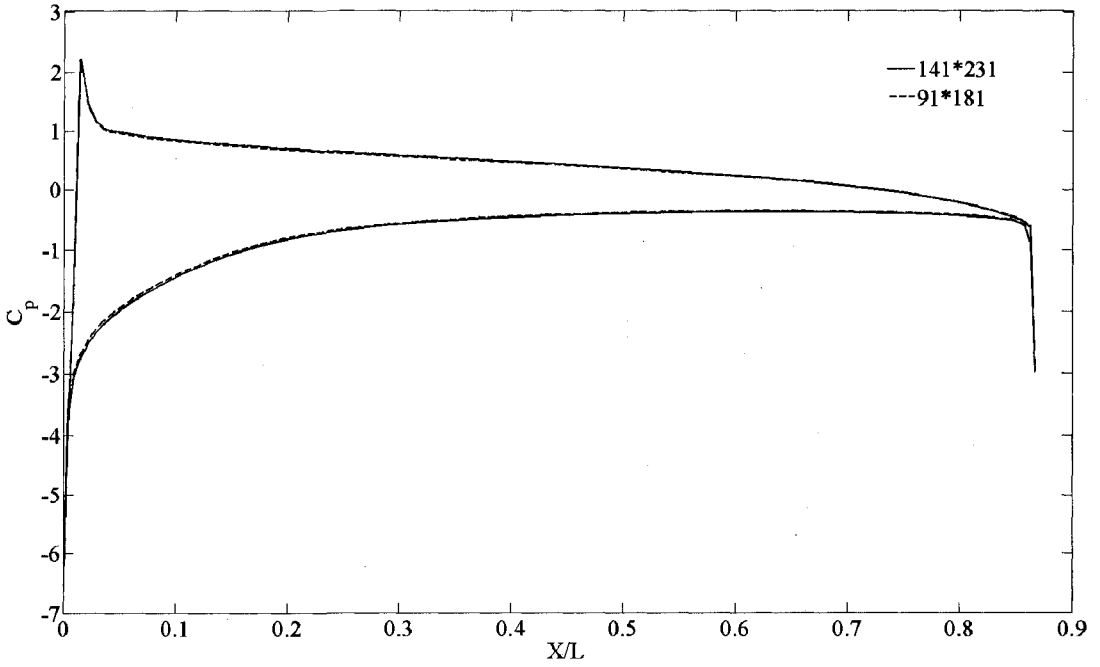


Figure 6.17: Grid independence test for a infinitesimally thin plate computed using immersed boundary method

In figure 6.18 the y-vorticty contour after t=20 is plotted using the finest grid.

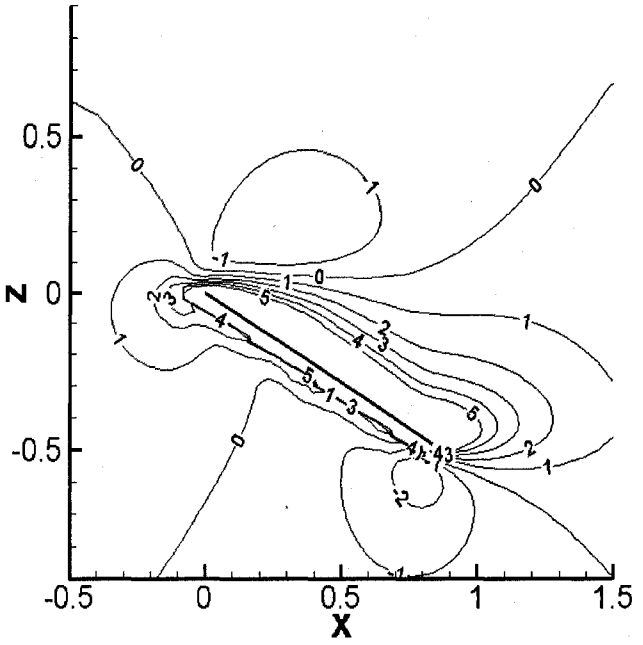


Figure 6.18: Y-vorticty contours after t=20

Lift Co-efficient from the current simulation 0.9123

Experimental Lift Co-efficient of Ahuja and co 0.9678

This is

$$0.9678-0.9123/(0.9678)\approx 5.73\%$$

The lift coefficient computed is compared with the experimental publication [48] shows the computation agrees reasonably well with the published result. The infinitesimally thin flat plate used in the computations compared to a finite thickness plate used in the experiments could a reason behind this error.

Step 5: Moving boundary model coupled with the immersed boundary method

6.6 Flow over a flapping flat plate using immersed boundary method

In this part of the chapter the flow around an unsteady flapping flat plate is studied using the immersed boundary technique and compare with the commercial code FLUENT. The Reynolds number is set as 1000, based on the length of the flat plate. The same immersed boundary technique is applied as suggested in the earlier section. Since the plate moves with a known frequency the position of the plate is known as a priori, the forcing points are found using a search algorithm with respect to the plate position. Once this is done, appropriate forcing at these points is applied so that the velocity at the boundary is equal to the velocity of the flapping plate.

In Fluent, the dynamic mesh method updates the volume mesh in the deforming regions by using spring based smoothing and local remeshing. These methods were employed to update the mesh at each time step to simulate the plate's new position. Interior nodes behave as if they have a series of springs attached to them. This enables the nodes which define the cells to be squished or pulled, but the number of nodes and cells remain

unchanged. Thus connectivity remains the same before remeshing. After each time step, these methods check the cells near the plate's centerline to ensure that they are within the specified minimum and maximum cell size limits and also under the maximum skewness. Cells that did not meet this criterion are remeshed to improve skewness [54]. This method works well, but only when the displacement of the boundaries is relatively small compared to the distance between the nodes on the same boundary.

Kinematic models

A kinematic model used for the flapping thin plate. The deformation $y(x,t)$ is described as $y(x,t) = -0.05x\sin(2\pi t)$. In this model, the plate is solid and the angular frequency of the flapping is 2π. Figure 6.19 mimics the motion of the flat plate. The position of the leading edge (x=0) of the plate remains unchanged for all flapping motions. In Fluent, the kinematic models are created in the software using user defined function (UDF).

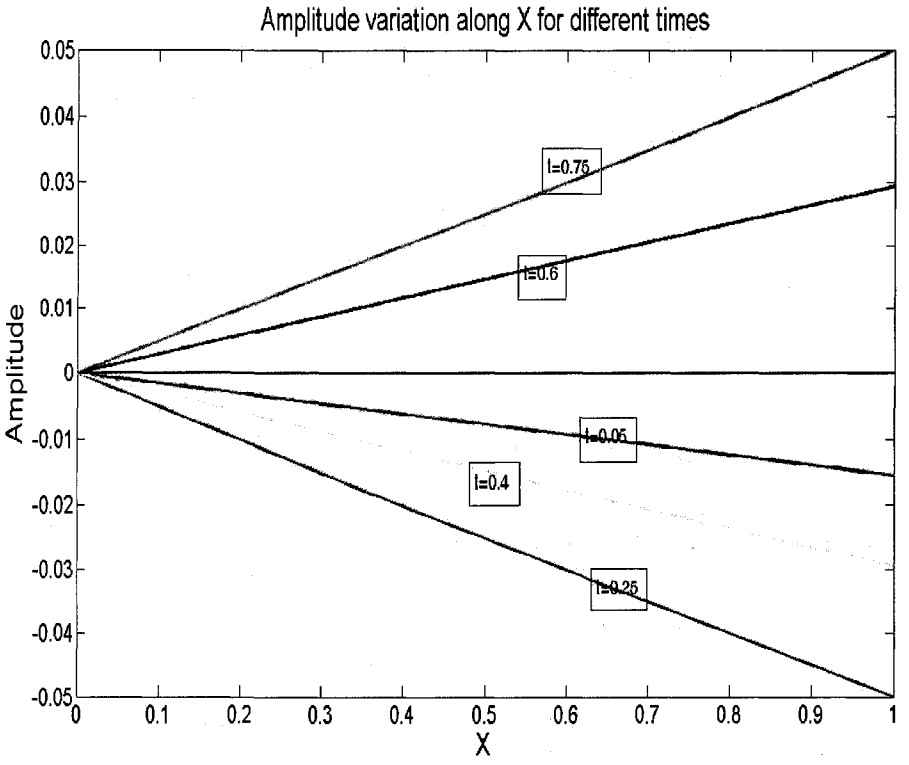


Figure 6.19: Kinematic model showing the object position at different times

Solver characteristics

Fluent

The commercial computational fluid dynamics code Fluent version 6.2 has been used to model the flow field by solving the two-dimensional, incompressible, unsteady Navier-Stokes equations and the continuity equation given as:

$$\rho \frac{DV}{Dt} = -\nabla p + (\mu + \mu_t)\nabla^2 V \qquad 6.2$$

$$\nabla \bullet V = 0$$

where *D/Dt* denotes the total derivative with respect to time. The segregated solver has been used. The computational domain is two-dimensional and the calculations are considered time-accurate with the first order discretization. The discretization schemes used for the pressure and the momentum equations are standard second-order upwinding. For pressure velocity coupling the SIMPLE algorithm was used.

Combined Navier-Stokes Immersed boundary method

The two-dimensional incompressible dimensionless RANS equations in the generalized curvilinear coordinates are discretized in space, on a non-staggered mesh using second-order finite difference approximations, and advanced in time using a four-stage Runge-Kutta scheme following a similar procedure described in Chapter II.

The *CFL* number used in these computations is 1.5 for the k and ε equations. For all the results shown here, the residuals defined by the summation of differences between the current and the previous iterations were reduced by at least four orders of magnitude.

Computational domain and grid details

The same computational domain used is used for both Fluent as well the combined immersed boundary Navier-stokes method. The domain spans from x/L = -10 to x/L = 11 in length, with the plate located on x-axis between 0 and 1, z/L = -10 to z/L = 10 in height, and the number of grid nodes for the compared case fixed at 201x201. Figures 6.20 and 6.21 show the domain and the grid used for both Fluent and combined Navier-stokes immersed boundary method. The Reynolds number based on the plate length and the free stream velocity is 1000. The flow is assumed laminar at all time during the unsteady calculations and no turbulence model has been used.

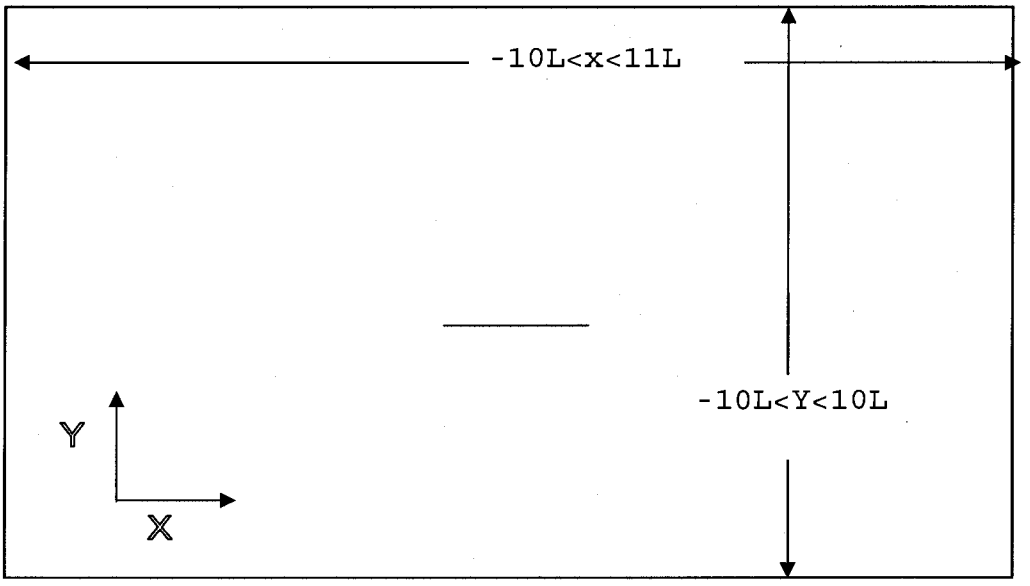


Figure 6.20: Kinematic model showing the object position at different times

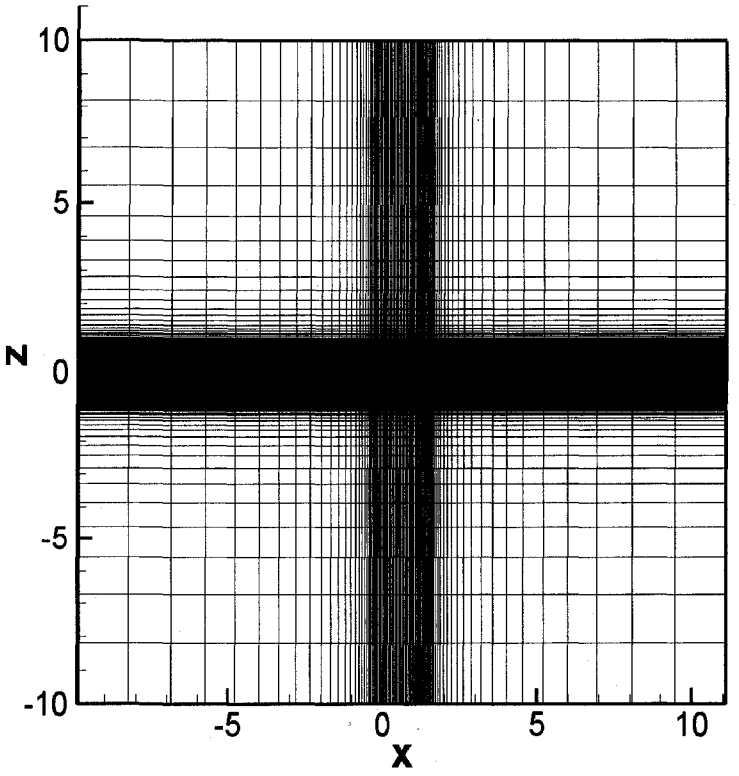


Figure 6.21: Computational grid employed by both codes

Results and discussion

The performance of the moving immersed boundary method is first validated by comparing the surface pressure coefficients from both Fluent and Immersed boundary Navier-stokes method at two different locations of the plate. From figure 6.19, since the starting motion is a down-stroke and the flapping period is 1 sec, the plate is at its lowest (trough), level, and highest (crest) positions every one second. The pressure distributions on the upper and the lower surfaces are

symmetrical with respect to the plate positions. The surface pressures distributions at two different positions of the plate i.e. crest and level positions respectively are compared with the commercial code Fluent. The non dimensional times at which surface pressures are compared in both the codes are at the crest position (6.75) and at the base level position (7) respectively. The lift variation history (plotted in figure 6.22) is calculated for both codes and compared. From the surface pressure distributions and the lift histograms one can say that there is a difference in the variation of amplitude of the lift coefficient. In brief, when running dynamic grid motion algorithms the commercial code Fluent reduces its temporal order of convergence to 1^{st} order where as the order of accuracy is close to 1.75 for the non-boundary stationary grid used in the immersed boundary case. This we predict could be one reason behind the difference of the surface pressures computed by the commercial code Fluent in comparison to the immersed boundary Navier-Stokes method. The spatial order of accuracy of both the codes remains close to 1.7.

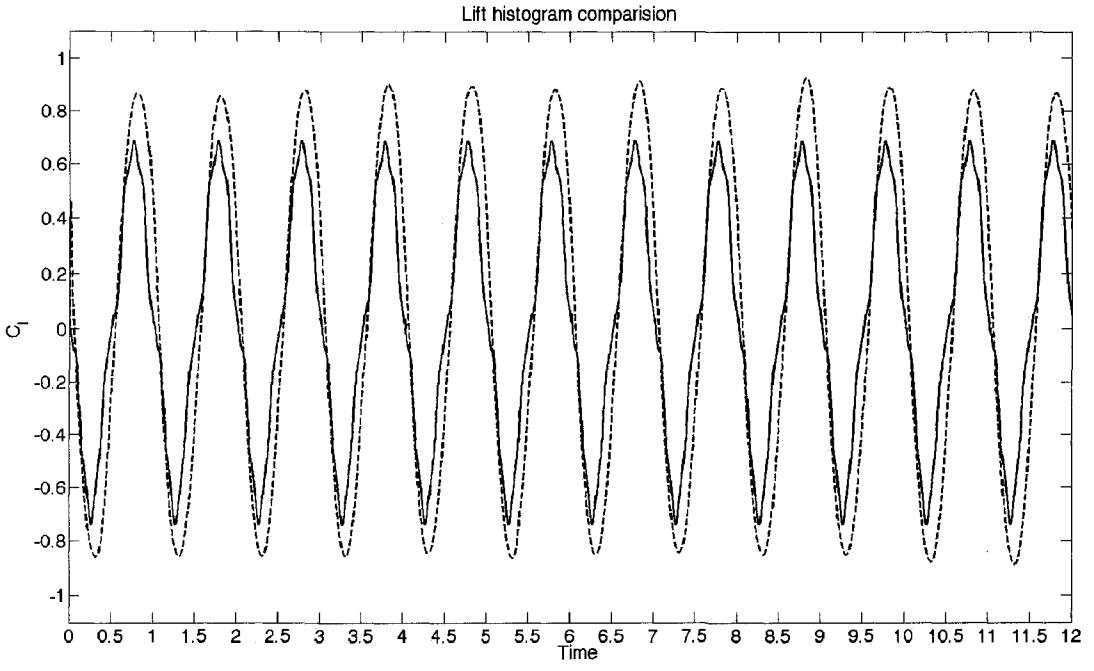


Figure 6.22: Lift histogram comparisons (Solid line- Immersed boundary method, Dotted line- FLUENT)

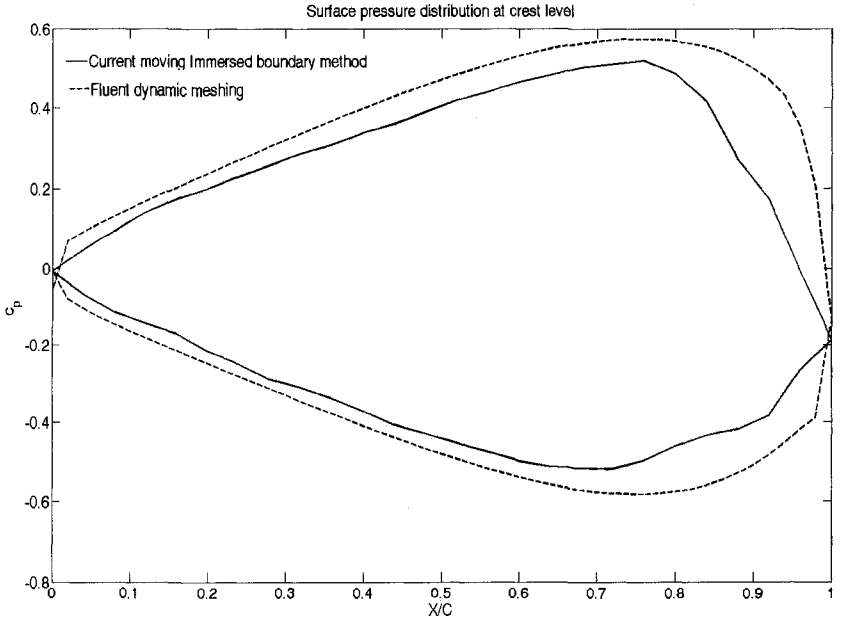


Figure 6.23: Surface pressure distributions on the plate at the maximum amplitude position

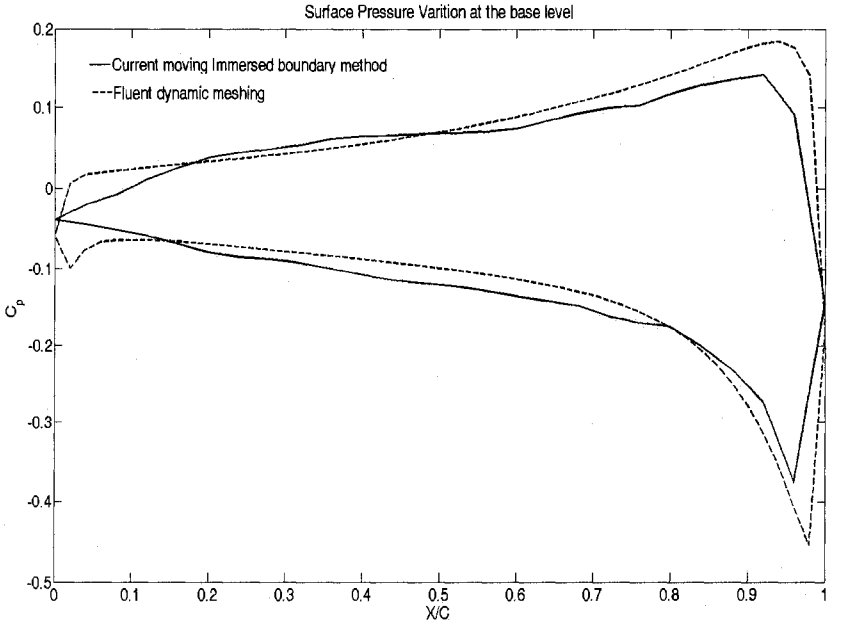
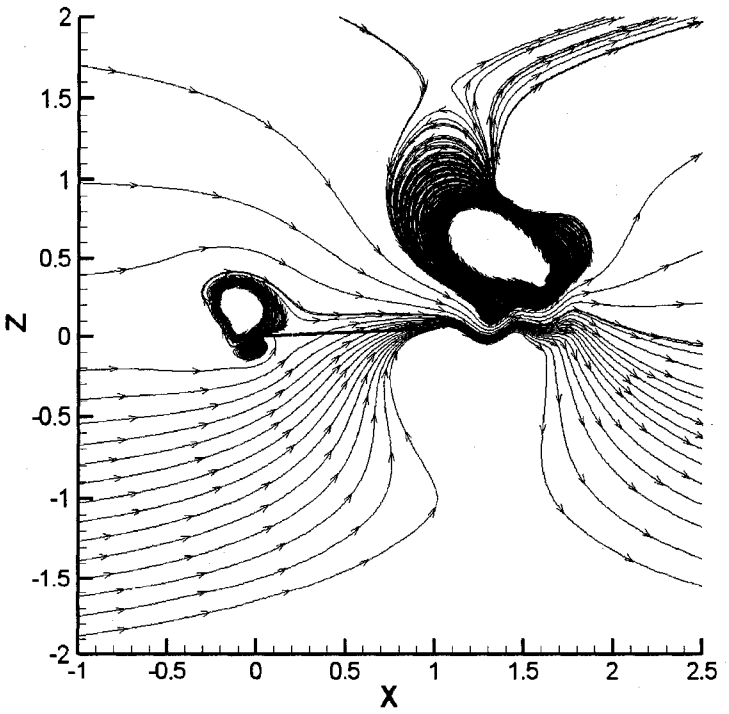


Figure 6.24: Surface Pressure distributions on the plate at the base level (zero amplitude position)



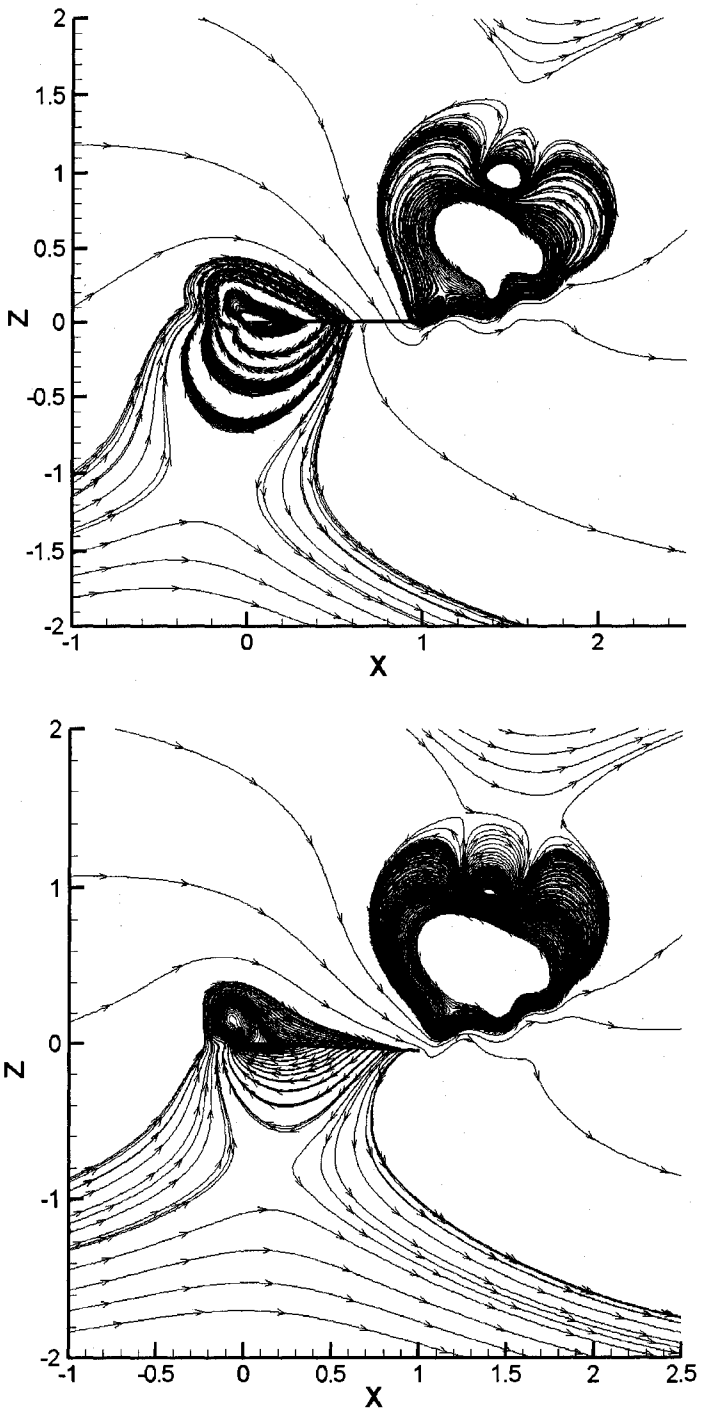
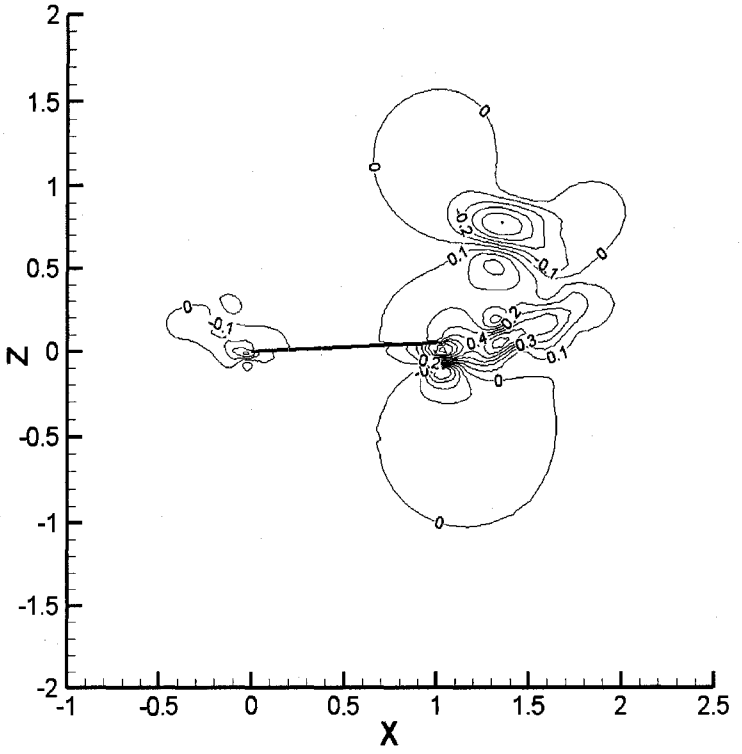


Figure 6.25: Stream lines with respect the plate position at maximum, baseline and minimum amplitude positions

Figures 6.25 and 6.26 show the streamlines and the time varying x-velocity contours at different positions of the plate. From the plots the stirring of the fluid are significant as it passes through the base level position. The separation bubbles at the trailing edge can also be observed from the stream line plots.



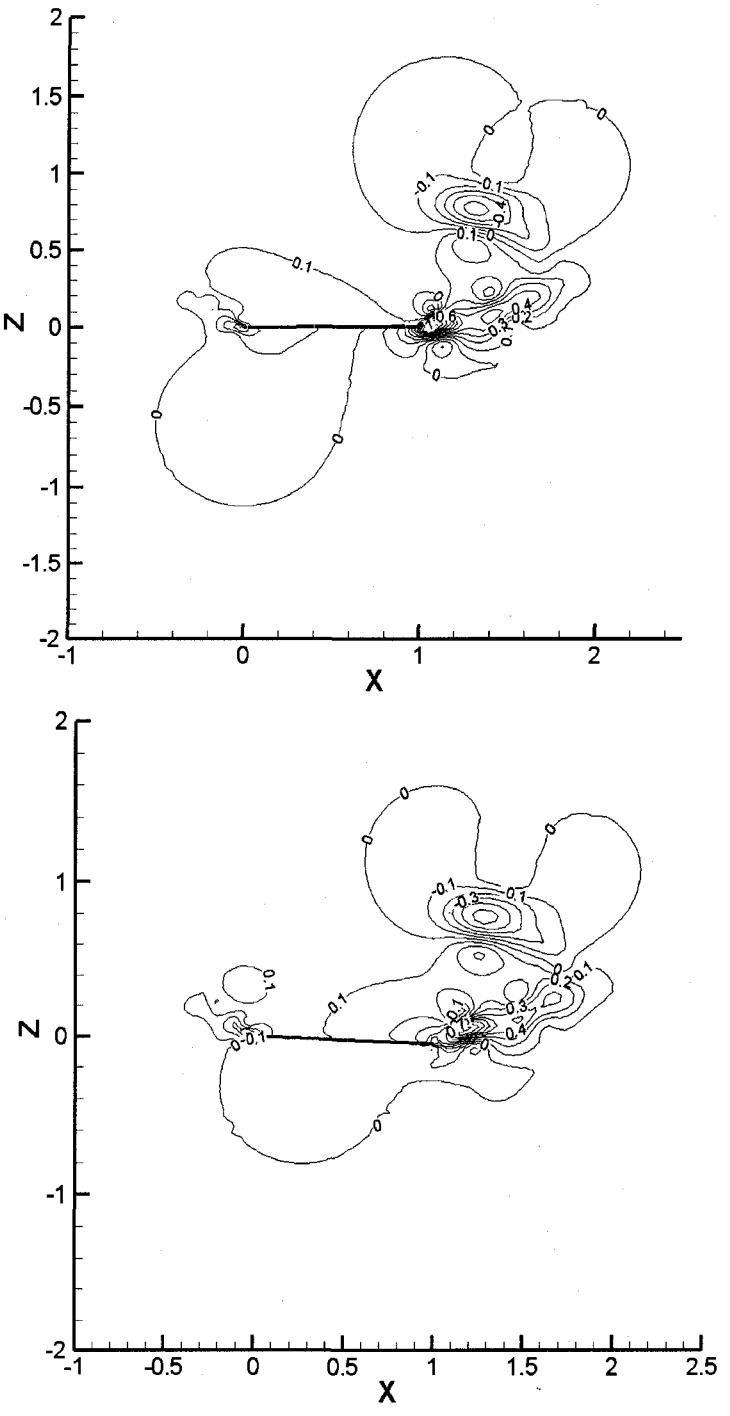


Figure 6.26: Time varying x-velocity contours with respect the plate position at maximum, baseline and minimum amplitude positions

6.7 Order of convergence studies

According to Roache [45] a CFD code uses a numerical algorithm that provides the theoretical order of convergence, however, the boundary conditions, numerical models, and grid will reduce this order. The Methods for examining the spatial and temporal convergence of are presented by Roache [45]. This method involves performing simulation on two or more finer grids. When the grid is refined (grid cells become smaller and the number of cells in the flow domain increase) and the time step is refined (reduced) the spatial and temporal discretization errors, respectively, should asymptotically approach zero, excluding computer round-off error. The solution error according to Roache is defined as the difference between the discrete solution and the exact solution.

$$Error = f(h) - f_{exact} = Ch^p \tag{6.3}$$

where p represents the order of convergence and h is the measure of the grid spacing. When three levels of grids are used a more direct evaluation of P can be done using a constant grid refinement ratio r and assuming that the solution at the finest grid to be "exact", we obtain

$$P = \ln\left(\frac{f_3 - f_2}{f_2 - f_1}\right) \Big/ \ln(r) \qquad 6.4$$

In order to calculate the spatial order of convergence, different grids need to be generated. Four uniformly spaced, successively refined meshes were used for this analysis. The approach that was taken in this study is to first generate a very fine grid. A coarser level grid is then built by removing every other grid line in each coordinate direction. The same procedure is also used to build the next level of coarse grids. The grid sizes used in the current work are 71*71, 141*141, 281*281 and 561*561. The grid refinement ratio r then equals to 2.

The grid convergence order is also examined by the averaged flow property changes in the computational domain. The L_∞ norm of the streamwise velocity components can be written as,

$$e_N^\infty = \max_{i=1,N^2} \left| u_i^{(N)} - u_i^{exact} \right| \qquad 6.5$$

where $u_i^{(N)}$ denotes the streamwise component of the velocity at node i of the N^2 grid and u_i^{exact} represents the exact solution at point i.

Flow over a circular cylinder

The variation of L_∞ norm of error with grid spacing is calculated for flow over a circular cylinder at Reynolds number of 40. Figure 6.27 shows the variation of the L_∞ norm of the error with grid spacing in a log-log scale. This is evident from the figure 6.27 the solver for this flow convenes at a rate which is close to second order. The line with slope of 1 and 2 are also shown in the figure for reference.

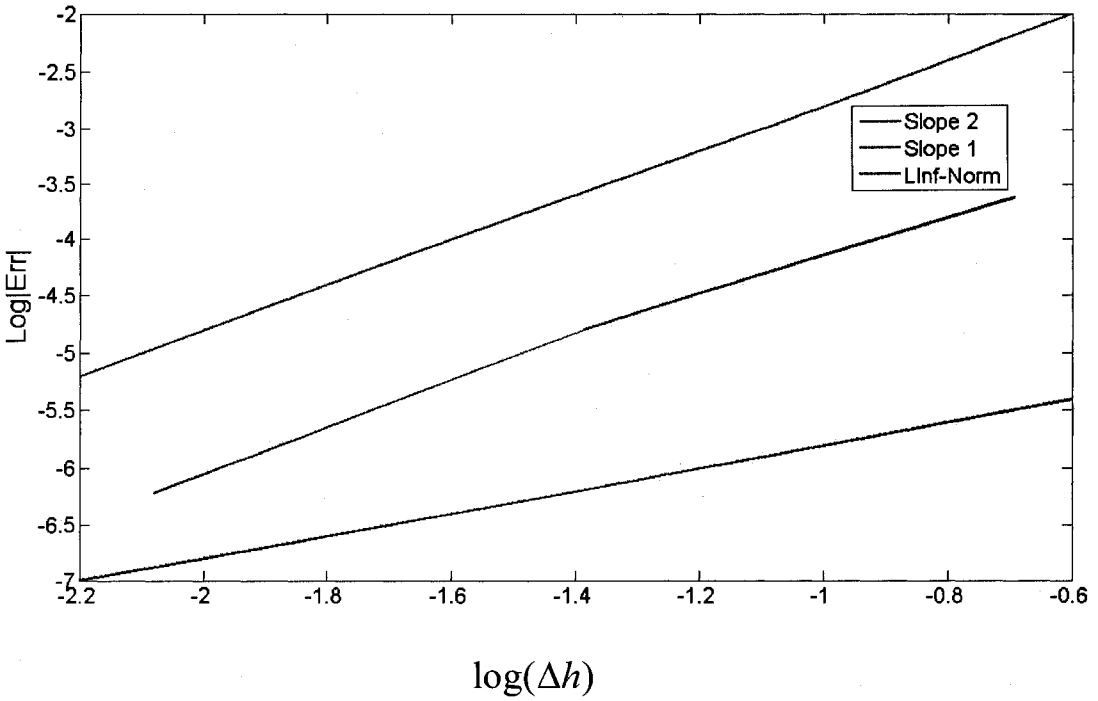


Figure 6.27: Convergence of L_∞ norm of error for the velocity field for flow over a circular cylinder

Flow over a NACA0012 airfoil

The variation of L_∞ norm of error with grid spacing is calculated for flow over a NACA0012 airfoil at Reynolds number of 500. Figure 6.28 shows the variation of the L_∞ norm of the error with grid spacing in a log-log scale. This is evident from the figure 6.28 the solver for this flow converges at a rate which is close to second order. The line with slope of 1 and 2 are also shown in the figure for reference.

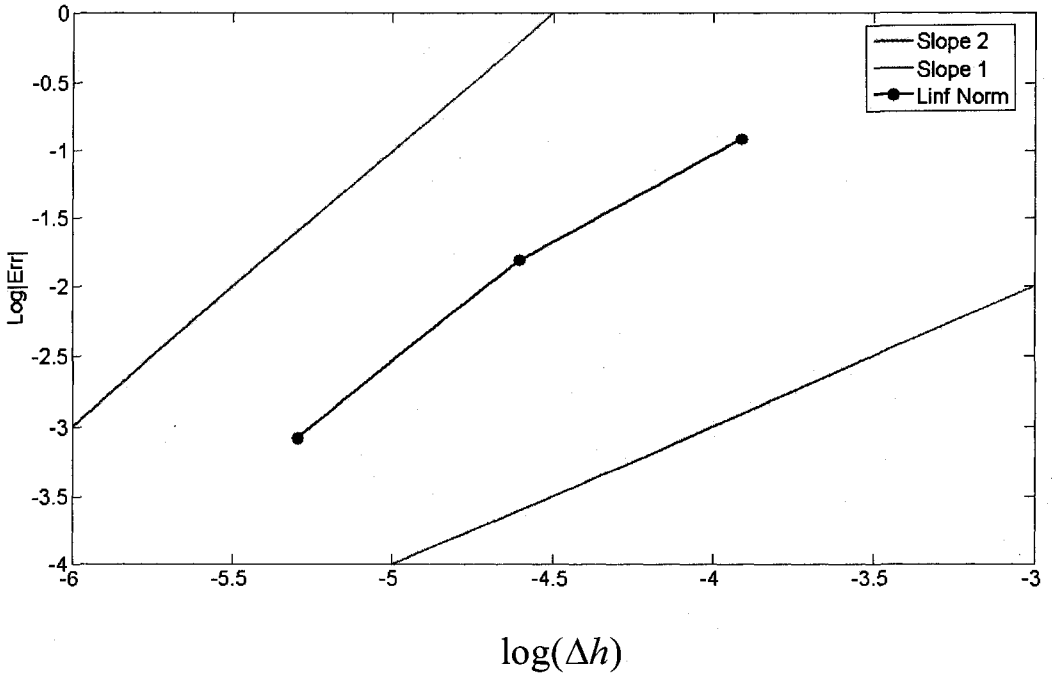


Figure 6.28: Convergence of L_∞ norm of error for the velocity field for flow over a NACA0012 airfoil

CHAPTER VII

FLIUD STRUCTURE INTERACTION OVER A PASSIVE FLEXIBLE FLAT PLATE

The developed fluid-structure coupled solver is first used to study a two-dimensional flow over an infinitesimally thin flat plate hinged at leading edge. All three remaining edges are free to move and are unconstrained. The simplicity associated with the wealth of properties involved make the flow over a flexible flapping plate a great start up problem to investigate the coupled fluid-structure solver. In this case we study the response of the fin due the fluid dynamic forcing acting on it. The flow dynamics are solved on a fixed Cartesian grid by interpolating values to the nearby grid points using the immersed boundary technique. The overall fluid dynamic forcing acting on the immersed boundary is calculated by extrapolation from the grid points and this is used as an external force to the structural dynamics solver. The coupling between the fluid and the structural solver is done in such a way that at every time step there is a switch between the fluid solver and the structural solver, where the boundary conditions are passed at the end of each individual solver time step. No sub iterations between the solutions takes place.

7.1 Problem description

The flow over a two dimensional passively flapping flat plate, which is hinged at leading edge and free to move in span wise direction at the trailing edge is as shown in figure 7.1. This flow over the passive flat flapping flat plate is performed using the coupled Navier-stokes immersed boundary with the subdivision finite element solver on this hinged flat plate at Reynolds number of 20,000. The plate is considered infinitesimally thin. The plate is made of mylar which can be stretched, deformed and bent and it resists these forces by its elastic nature.

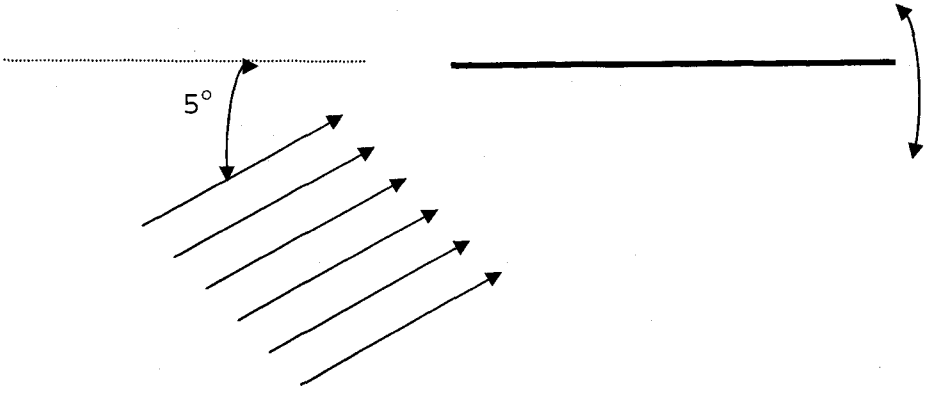


Figure 7.1: Schematic representation of the Mylar flapping flat plate hinged at the leading edge

The mylar plate is a biaxially-oriented polyethylene terephthalate polyester film with high tensile strength and chemical/dimensional stability. Some of the

interested properties are: Young's modulus (MD) 2.8 Gpa; Poisson ratio0.37 and Density of $1.39g/cc$.

7.2 Computational domain

In this section both the computational domain and the grids used by both the fluid solver as well as the structural solver are discussed. Different grids are employed by both the solvers. The node positions of both solvers are different since different grid orientations are used for different solvers. The values are interpolated to nearest grid points for both solvers to either input or output the appropriate boundary conditions from the fluid and structural solver and vice versa.

Combined Navier-Stokes immersed boundary solver

The computational fluid dynamic grid used in the laminar flow over a passive flapping flat plate is generated is shown in figure 7.2. The computational domain is x/c=20 in length, the plate is located at the base level located in the x-axis between 0 and 1, z/c=20 in height, and y/c=0.2 in width. The grid is clustered near the inlet and is stretched using a hyperbolic stretching function toward the exit in the stream wise direction and cross stream directions.

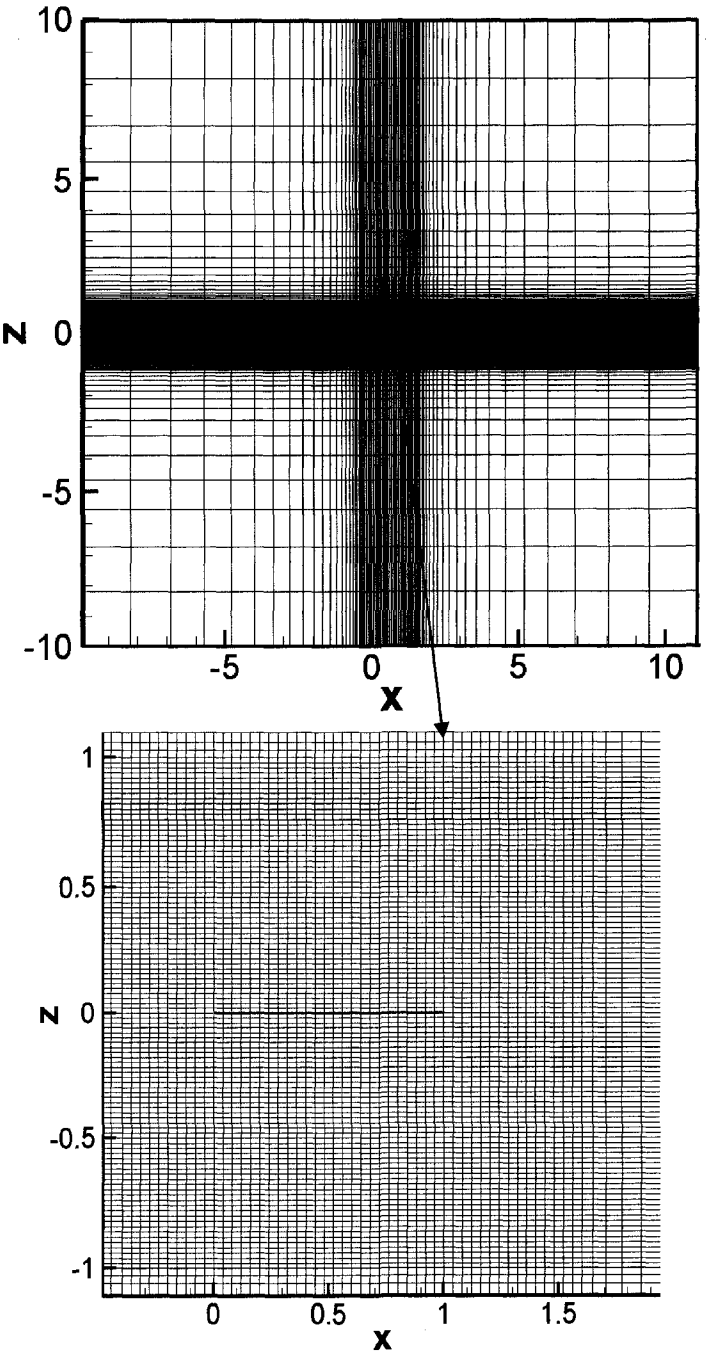


Figure 7.2: Computation fluid dynamics grid with the plate at baseline position before the flow is impulsively started

The inlet boundary conditions for the laminar flow over a passive flapping flat plate are assumed uniform for all variables, where U = cos(α), V = sin(α). The flow is essentially laminar and hence no turbulence model used. On the symmetry boundaries in the span wise direction, the mirror-image reflections for the grid and the flow variables are used. The exit boundary condition is imposed by assuming zero stream wise diffusion. At the outer boundary, the corresponding free stream recovery is assumed. The wall boundary condition is zero value for all variables, that is $u = v = 0$.

Subdivision finite element solver

The computational structural dynamic grid used to input the pressures on to the surface of the plate is as shown in figure 7.3. The computational domain is x/c=1 in length, y/c=0.2 in horizontal cross stream or span wise direction.

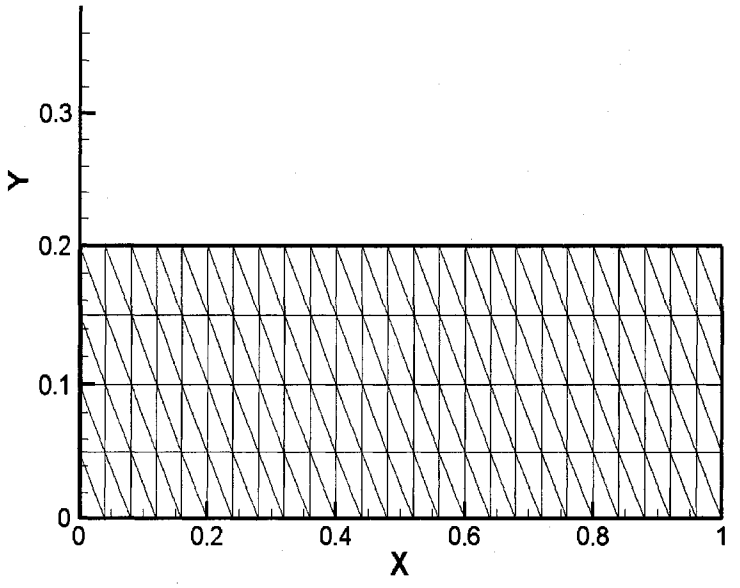


Figure 7.3: Unstructured triangular mesh used in the structural code for the flexible flapping plate hinged at x=0 at the starting position

An unstructured triangular grid of equal size is generated and used in the computations. The pressure loading is applied on the center line at y=0 and mirror image reflections for loading are used at the other lines in the span wise direction.

7.3 Coupling CFD and the CSD solvers

The computational techniques used for both the CFD and the CSD solvers are different (as suggested in chapter II and III) and this increases the complexity of the problem. The usage of different CFD as well as CSD solvers here to compute the problem works only when there is a proper coupling between the codes. There are

different techniques for exchanging the information, between the CFD and the CSD codes, which was suggested by Giannopapa [12] for a study of Fluid structure interaction in flexible vessels. One of the techniques described is used for the present coupling.

1) The most basic approach Giannopapa [12] describes is non-iterative over all time. The Navier-Stokes equation and the thin shell formulation equations are solved separately for the whole computational domain, which remains the same for the solvers. The major underlining fact is that the effect of the structure on the fluid is considered to be negligible. Figure 7.4 shows the coupling procedure followed for this kind of formulation

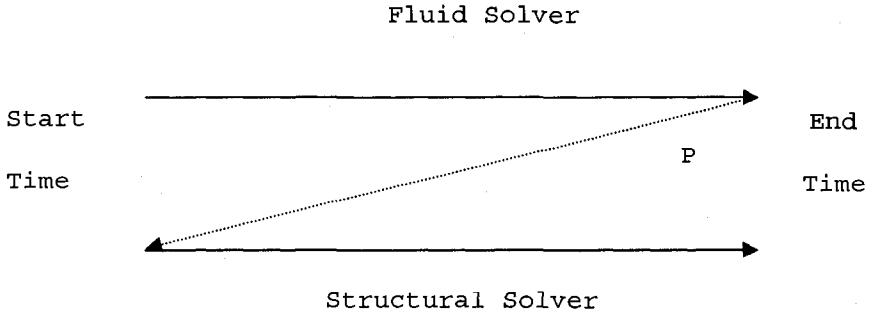


Figure 7.4: Non-iterative scheme over all time

2) The second technique would be to couple both the solvers in an iterative manner and hence the name

iterative over all time. This technique is a combination of the above technique with the exception that the solution from the CSD code is used as a transient boundary condition on the fluid unlike the 1st techniques where the structural effect was neglected. The coupling procedure is shown below in figure 7.5

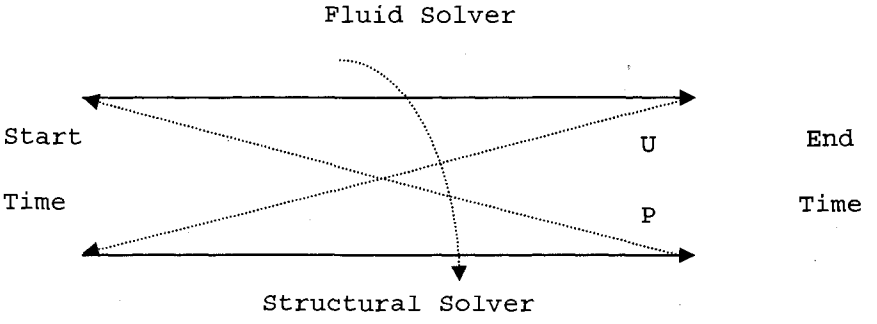


Figure 7.5: Iterative schemes over all time

3) The other technique described by Giannopapa [12] is similar to the above two techniques with the exception that the information is passed from the fluid/structural solution to the structural/fluid solution at the end of every individual time step. The schematic techniques are depicted below in the figure 7.6. This is the technique that is used in this thesis where the pressure force acting on the structure is calculated from passing the total pressure force as an input to the structural boundary (i.e.

plate) and the total deformation and velocity information is passed into the fluid solver where the plate velocity is forced at the nearby grid points using the two-direction interpolation techniques suggested in the earlier chapter.

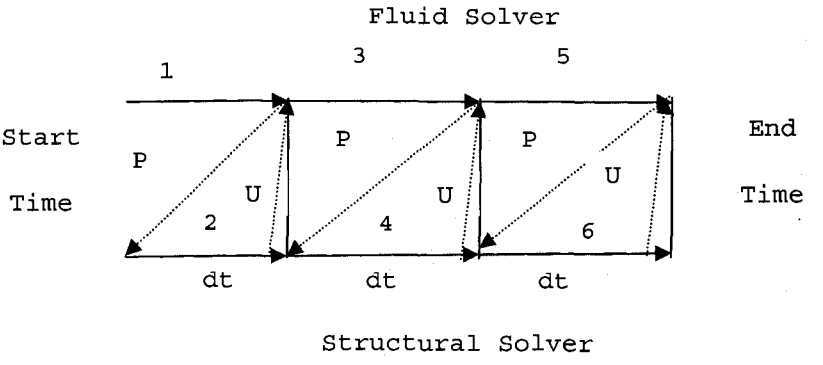


Figure 7.6: Non-iterative schemes over each time step

7.4 Results and discussion

A grid independence study for laminar flow over a flapping flat plate has been performed, and the results are as shown in figure 7.7. Two different fluid grids of 91×161 and 91×181, in the stream wise and the wall-normal directions, respectively are generated. The structural solver remains the same but two different grids of increasing mesh density are used on the fluid solver. The surface pressure distributions on the plate after the plate has stabilized are computed and compared on these two different grids. The computed pressure distributions

using these two different grids basically collapse indicating the flow solutions are grid independent. The 91x181 grid is used later where the contours of vorticity are used to visualize the simulation results. The instantaneous positions of the plate as well as the bending stress are also plotted at those time steps to visualize the structural simulation results.

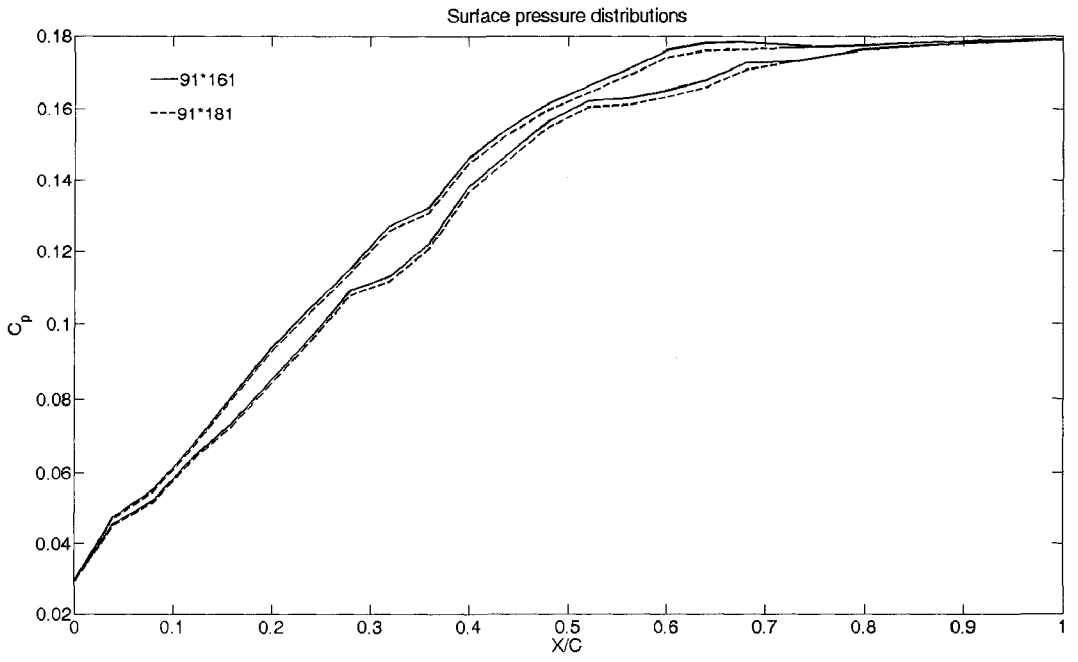


Figure 7.7: Surface pressure variation at t=1.3

In order to study the response of the plate to the incoming flow the tail displacement with respect to time and the lift variation are plotted as shown in figures 7.8 and 7.9 respectively. From figure 7.8 it can be interpreted that initially as soon as the flow is

impulsively started the plate moves randomly for a little while, then it experiences a periodic flapping, finally aligning itself close to the incoming flow angle. With the motion of the plate, the overall aerodynamic properties of the plate change. The time history of the lift through the whole process is plotted in the figure 7.9. It is clear from the figure that the lift fluctuation follows the frequency of the flapping.

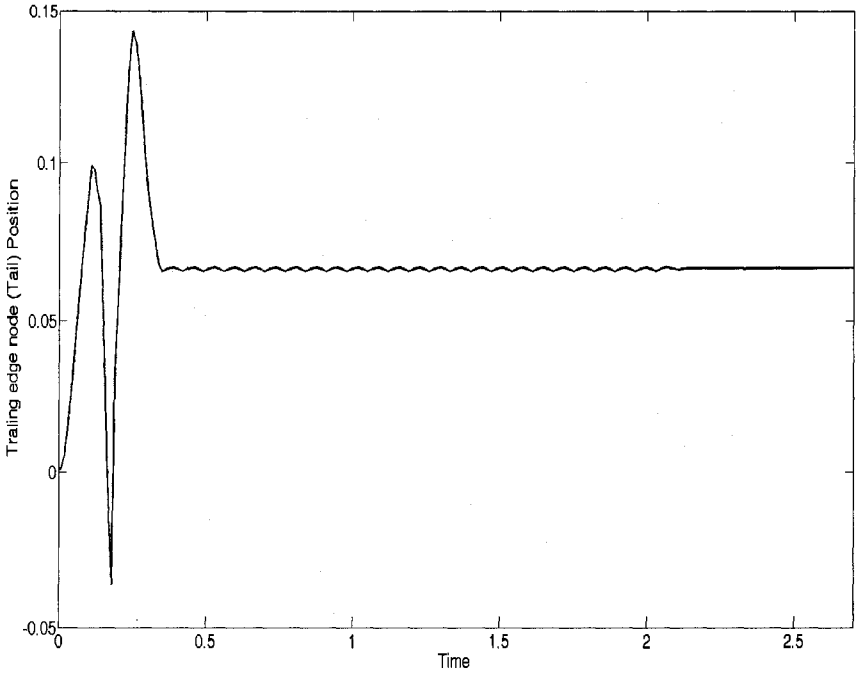


Figure 7.8: Tail amplitude variations with respect to time

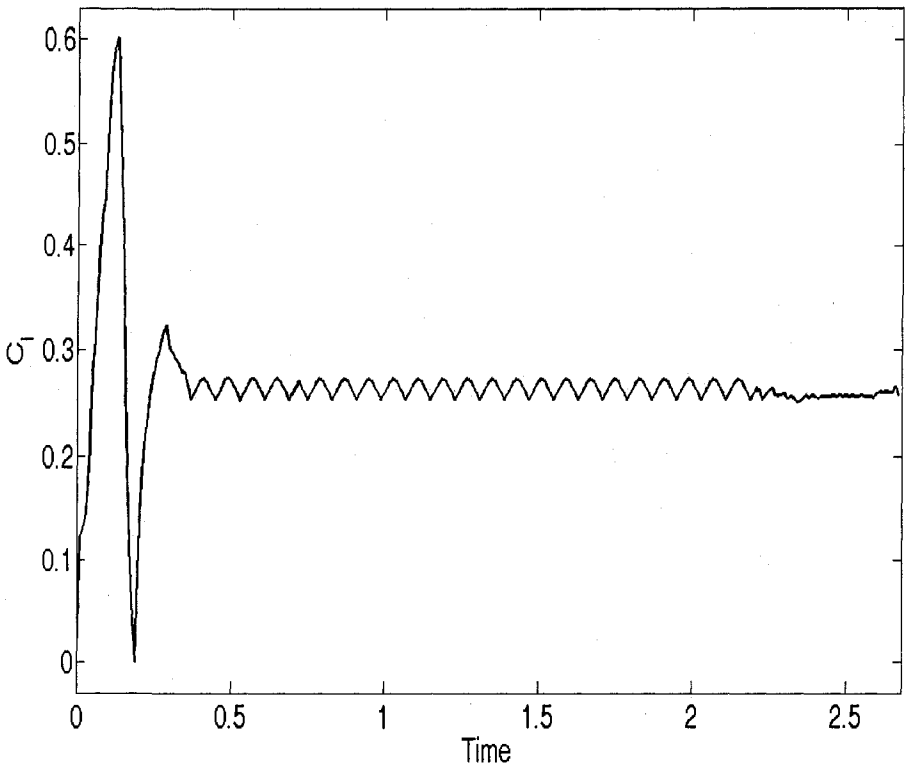


Figure 7.9: Lift histogram of the flexible flat plate

The instantaneous position of plate and the contours of vorticity are used to visualize the flow. In addition to this the bending stress and the pressure force on the plate are plotted in the following pages. The flow is from left to right when visualizing the vorticity contours. The same is true for position of the plate with the leading edge at the left end hinged. The vorticities and the plate positions are plotted at an equal time interval of 0.1 sec in between them.

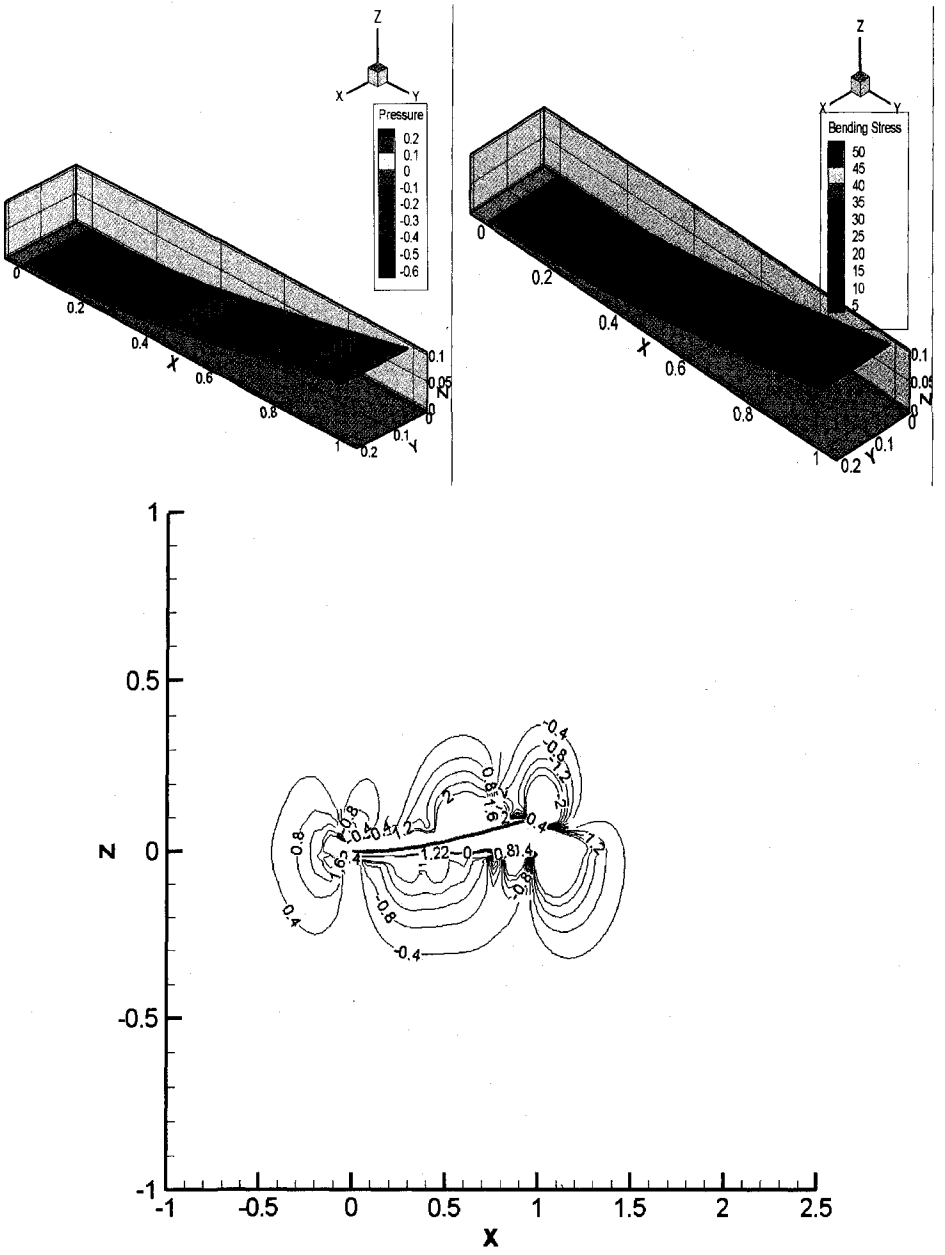


Figure 7.10 a) Pressure, bending stress and y-vorticity magnitude after 0.1 sec

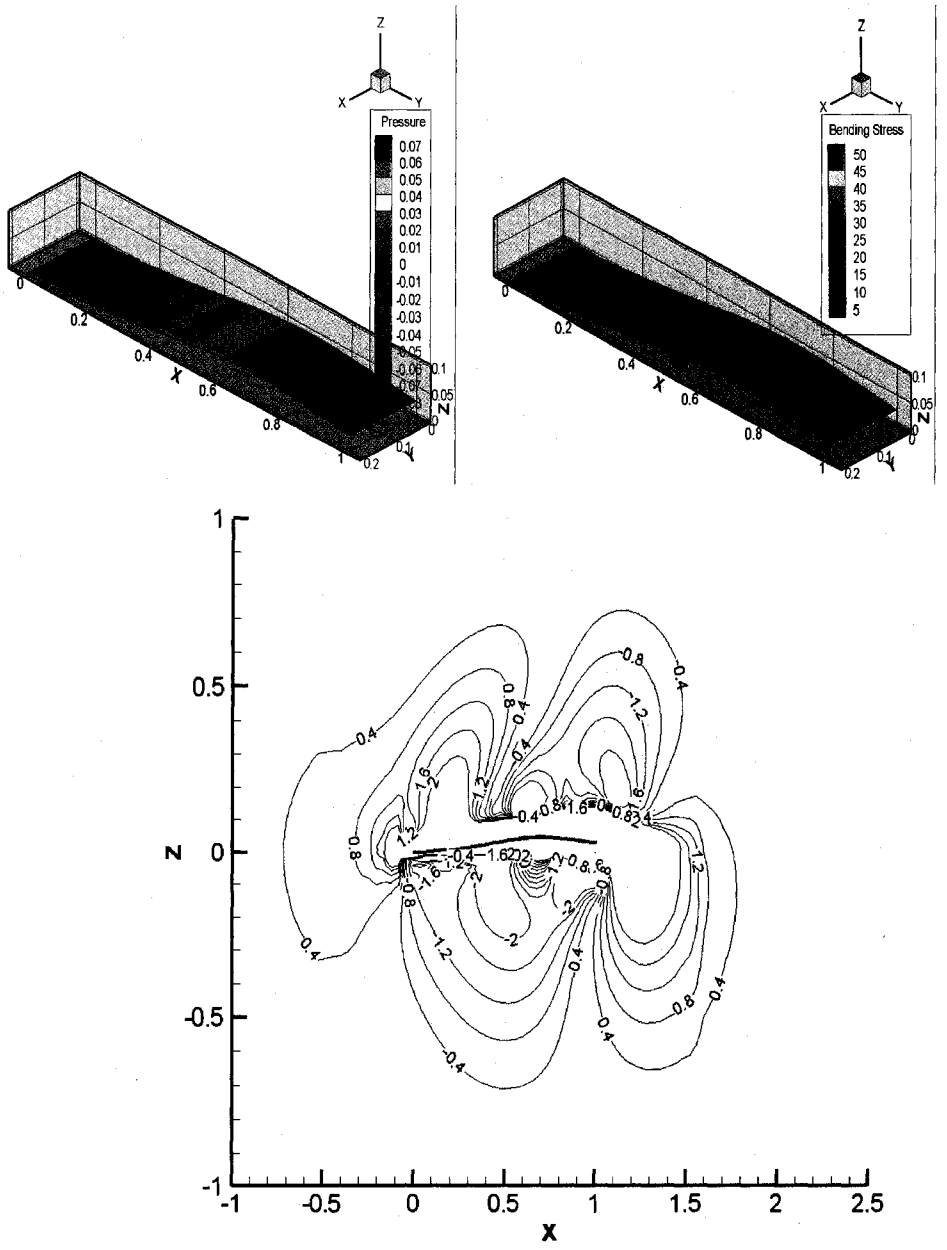


Figure 7.10 b) Pressure, bending stress and y-vorticity magnitude after 0.2 sec

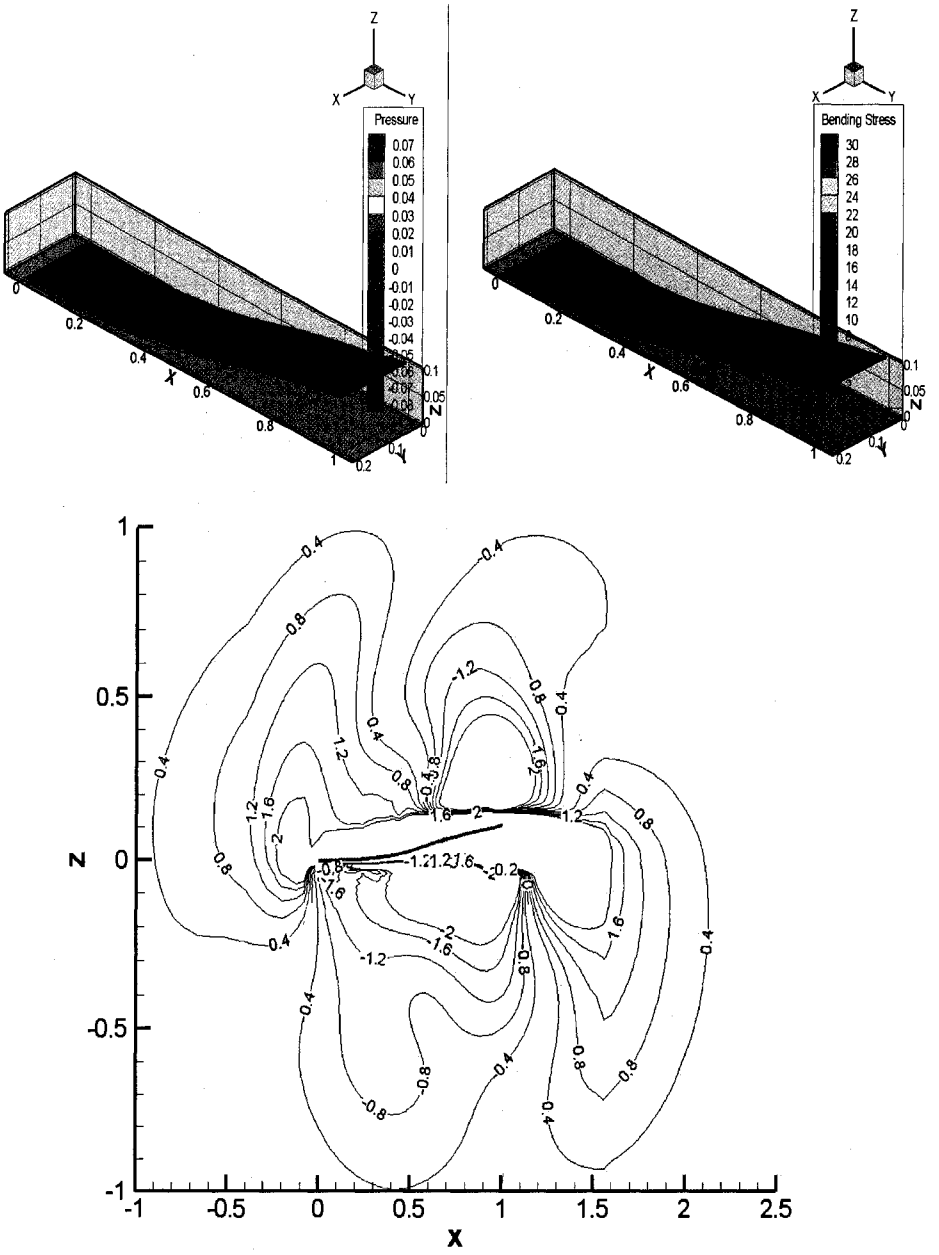


Figure 7.10 c) Pressure, bending stress and y-vorticity magnitude after 0.3 sec

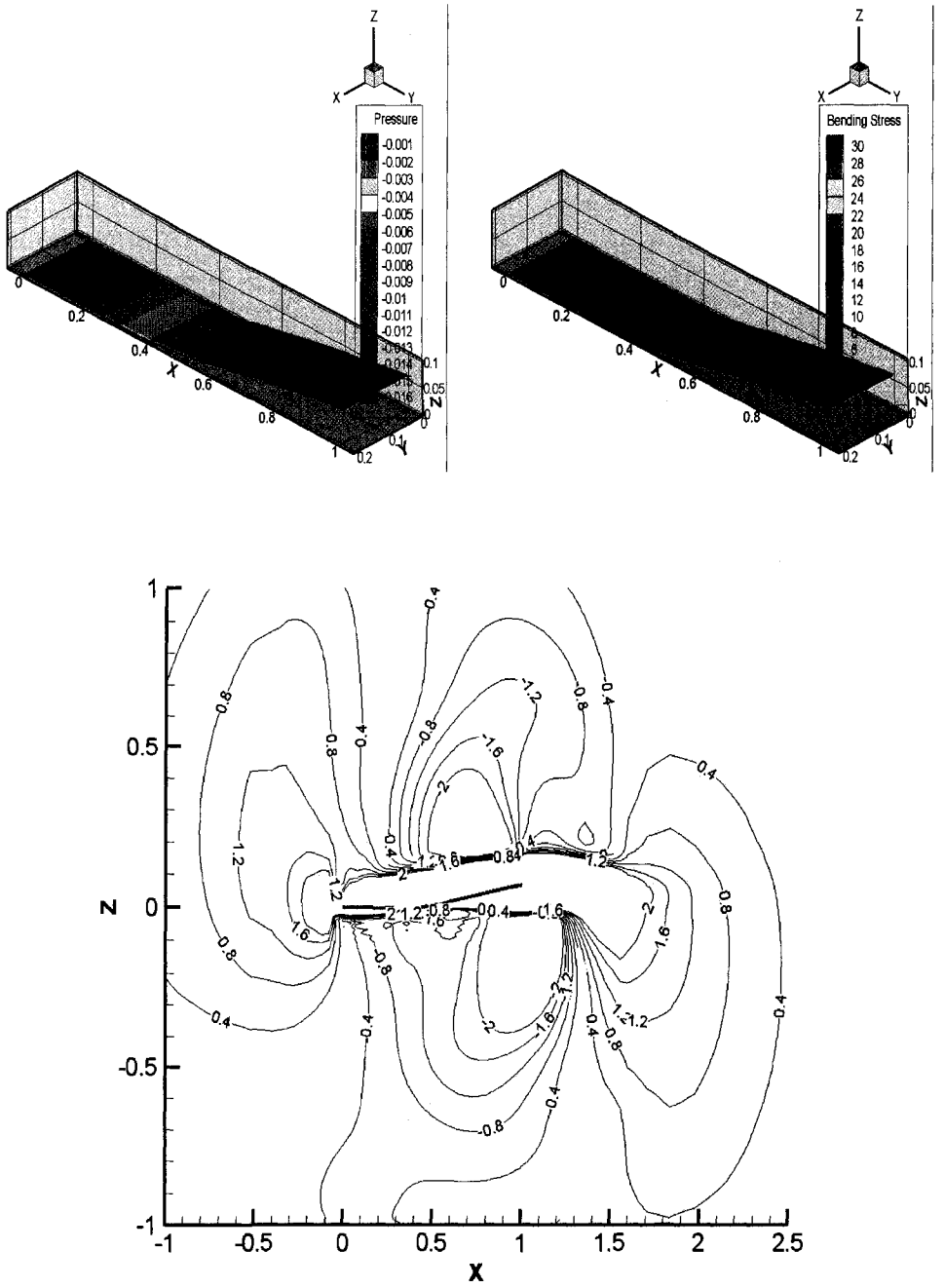


Figure 7.10 d) Pressure, bending stress and y-vorticity magnitude after 0.4 sec

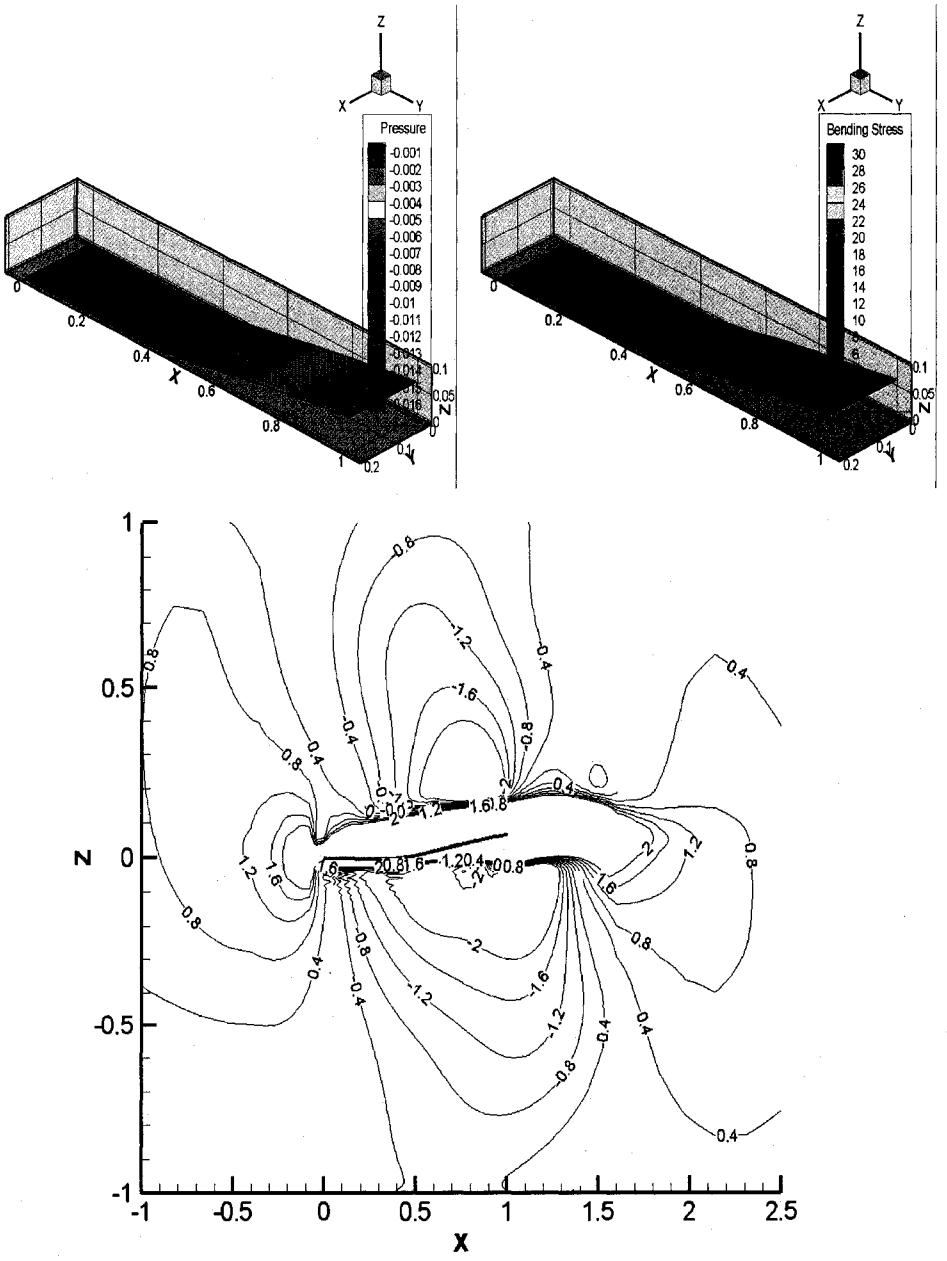


Figure 7.10 e) Pressure, bending stress and y-vorticity magnitude after 0.5 sec

CHAPTER VIII

MODELING FLUID STRUCTURE INTERACTION OVER A FLEXIBLE FIN ATTACHED TO THE UPPER SURFACE OF A NACA0012 AIRFOIL

This chapter explores a new concept of post-stall flow control for airfoils and wings by using a thin flexible fin attached on the upper surface of an airfoil to passively manipulate flow structures in the fully separated flows for drag reduction and lift enhancement. The flexible fin used for passive control in this chapter is a thin Mylar film. The fin is 0.25c long and is fixed at 0.1c on the upper surface of the NACA0012 airfoil. The Mylar thin film is a biaxially-oriented polyethylene terephthalate polyester film with high tensile strength and chemical/dimensional stability. The velocity and vorticity fields and the fin kinematics are given to provide insights into the physical mechanisms of the post-stall flow control. The lift and the drag coefficients are computed. The average drag co-efficient from experiments at angle of attack of 18^0 is compared with the running average drag from the computations. The average drag coefficients from the experiments as well as the computations at various angles of attack are also compared. The average viscous and pressure drags at various angles of attack are computed and compared with

the baseline NACA0012 airfoil.

8.1 Experimental set-up

Experiments were conducted in the water tunnel (The Rolling Hills Research Corporation Model 1520) in the Fluid Mechanics Laboratory at Western Michigan University by Dr. Liu and Dr. Montefort. The test section is nominally 15 in wide, 20 in high and 60 in long. The tempered glass, 3/8 in thick on the sidewalls and 1/2 in thick on the bottom, is mounted with silicon rubber, allowing good optical access from the top, bottom, both sides and rear for flow visualization and Particle Image Velocimetry (PIV) measurements. The tunnel is operated as a continuous flow channel and the water level in the test section is typically adjusted to be roughly 50 mm below the top of the walls. The free water surface provides simple access to the model and easy setup of an external force balance. There is a 6:1 contraction section before the test section for turbulence reduction and avoidance of local separation and vorticity development. The test section flow velocity is variable from 0 up to 0.3 m/s. In the test section, the turbulence intensity is less than 0.1%, and the velocity non-uniformity is less than 2%, and the mean flow angularity is less than 1% in both the pitch and yaw angles.

A plastic NACA0012 airfoil section model that was built by a rapid prototype machine was tested. The chord and span of the model were 10 in and 12 in, respectively. A clear Mylar (PET-polyester) film was used as a flexible fin. The Young's modulus and Poisson ratio for Mylar were 2.8 GPa and 0.37, respectively. The Mylar film was attached to the upper surface of the airfoil by Scotch tape. Figure 8.1 shows a flexible Mylar fin attached on the NACA0012 model in a typical PIV image.

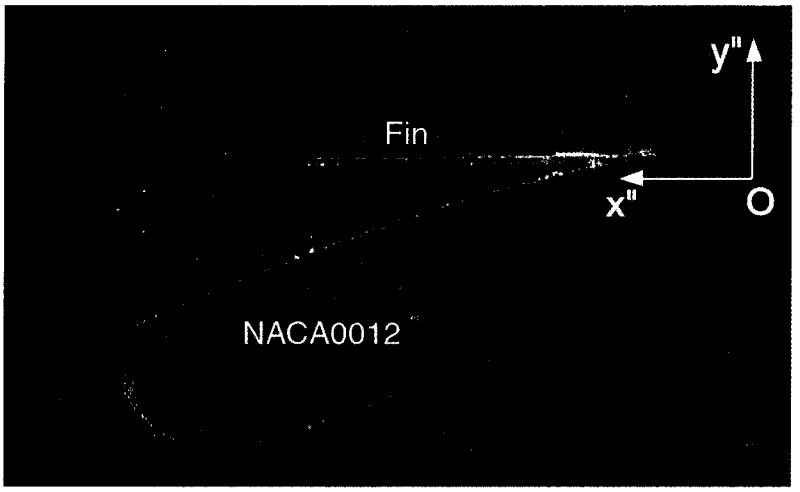


Figure 8.1: Experimental setup of the fin attached to a NACA0012 airfoil (Courtesy of Dr. Liu and Dr. Montefort)

8.2 Computational set-up

The flow over a passively flapping flat plate attached to the upper surface of a NACA0012 airfoil, which is hinged at leading edge and free to move in span

wise direction at the trailing edge is as shown in figure 8.2. This flow is computed on a C-Grid fitted on top of a NACA0012 airfoil is generated using Meshpilot [39]. The Reynolds number based on the chord length of the airfoil is fixed as 63,000 with a 18^0 angle of attack. The plate is considered infinitesimally thin. The leading edge of flexible plate as described earlier is fixed at 0.1 chord length to the upper surface of the airfoil.

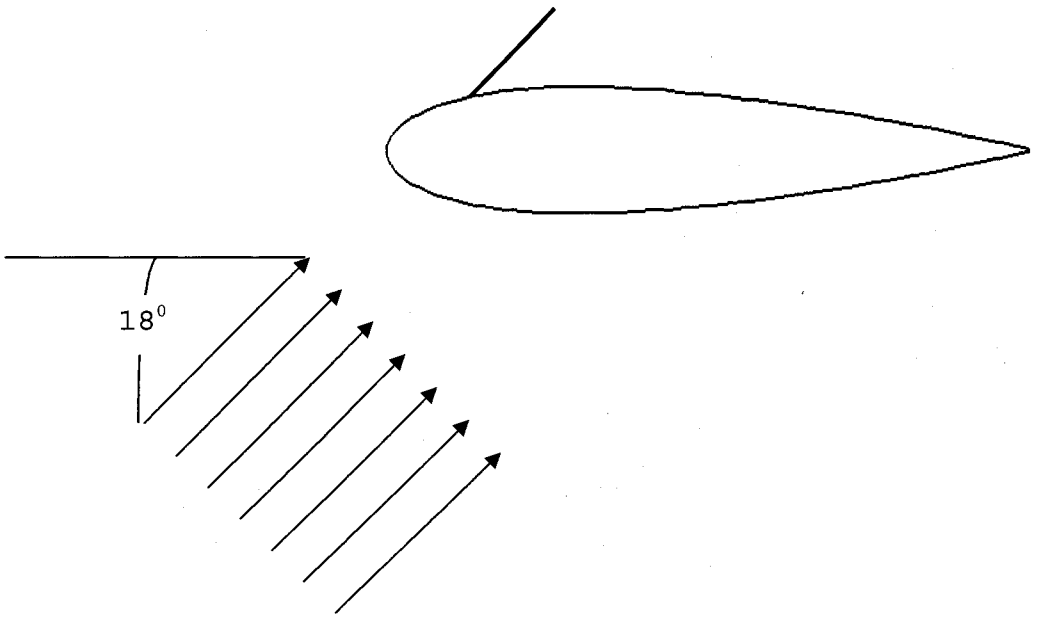


Figure 8.2: Schematic representation of the Mylar flapping flat plate attached to the upper surface of NACA0012 airfoil

8.3 Domain and grid details

The c-type computational grid used in the CFD solver

for turbulent flow over a NACA0012 is generated using Meshpilot [39] software is shown in the figure 8.3. The computational domain is x/c=30 in length, the airfoil located in the x-axis between 0 and 1, z/c=30 in height, and y/c=0.2 in width.

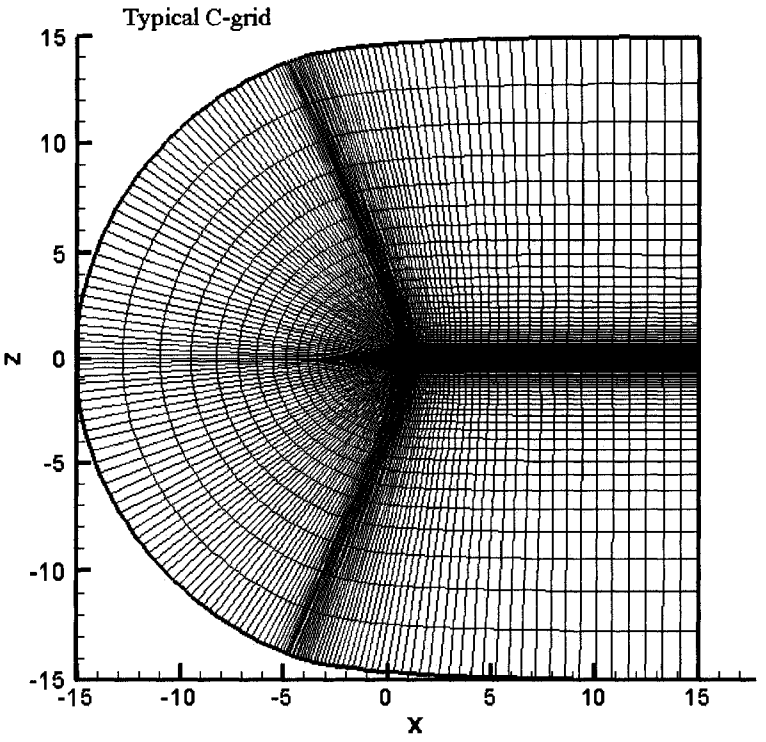


Figure 8.3: Body fitted C-Grid generated using Meshpiolt

This means the computational domain extends from 15 chord lengths upstream of the airfoil to 15 chord lengths downstream of the airfoil in the stream wise direction and 15 chord lengths in either direction normal to the airfoil in cross stream direction and 0.2 chord lengths

in the span wise direction. The zoomed up plot of the grid with the fin attached before the start of the simulation is shown in figure 8.4. It is evident that the grids are clustered near the NACA0012 airfoil and along the centerline of the wake.

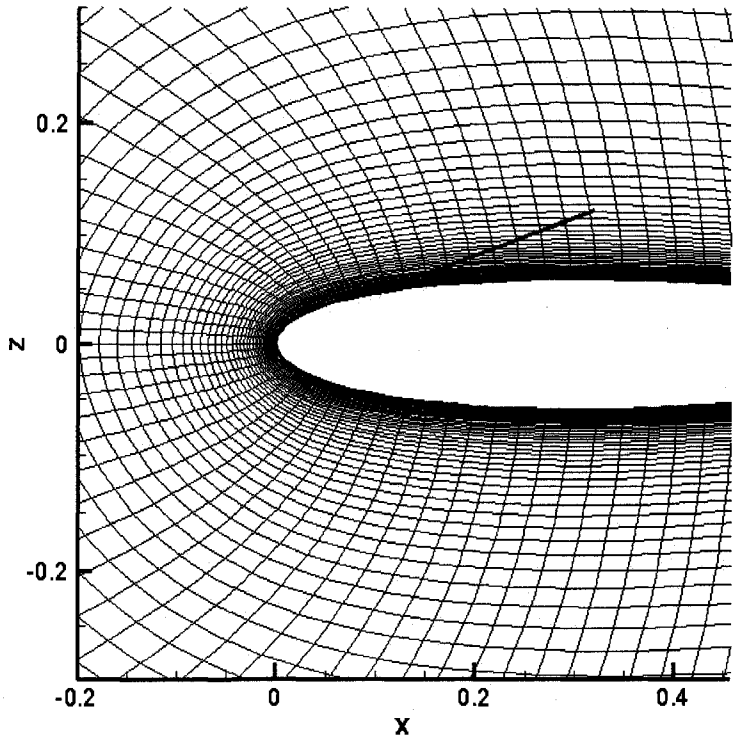


Figure 8.4: Zoomed up body-fitted grid showing the flexible fin orientation before the start of simulation

The computational structural dynamic grid used to input the pressures on to the surface of the plate is as shown in figure 8.5. The computational domain is x/c=0.25 in length, y/c=0.2 in horizontal cross stream or span

wise direction.

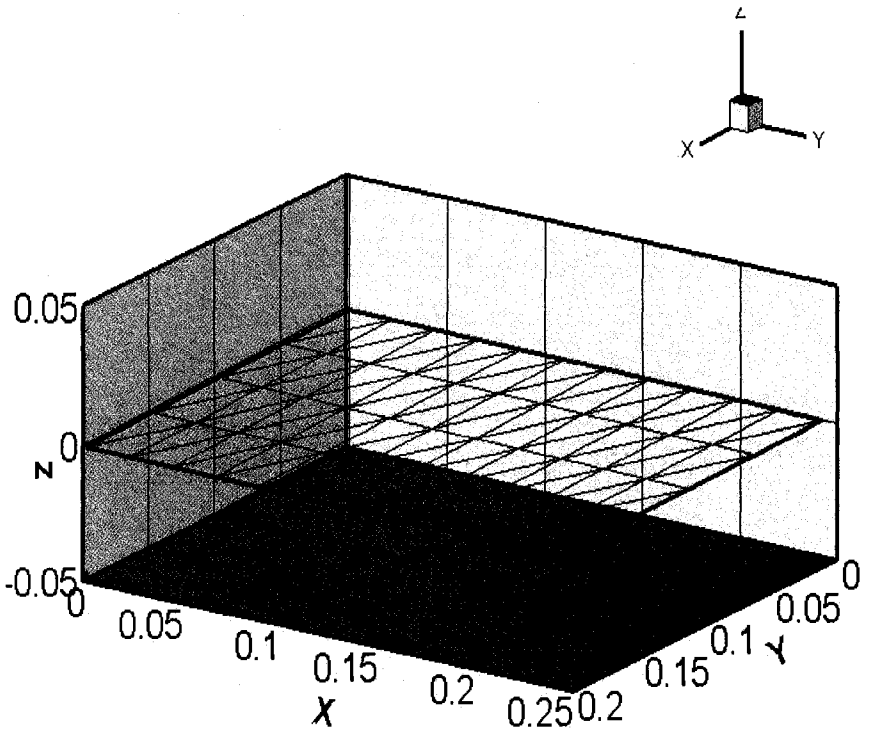


Figure 8.5: Unstructured triangular mesh for the flexible flapping plate hinged at x=0 at the start of the simulation

An unstructured triangular grid of equal size is generated and used in the computations. The pressure loading is applied on the center line at y=0 and mirror image reflections for loading are used at the other lines in the span wise direction.

The boundary conditions used on the CFD solver are the same ones used in the chapter IV when simulating when flow over a NACA0012 airfoil using a C-type grid. The conditions on the plate are same as suggested in the

chapter VII. The coupling between the CFD and CSD solvers is same as the one used in chapter VII.

8.4 Results and discussion

In order to understand the underlying mechanism behind the flexible fin effect on the NACA0012 airfoil the development streamlines is studied. In order to understand how the global aerodynamic properties vary firstly the lift histogram from the computations with the fin is compared to the baseline NACA0012 airfoil case. Secondly running average pressure, viscous and the total drag are computed and compared with the average drag coefficient from the experiments. Then the time averaged drag coefficients of the baseline NACA0012 airfoil as well as one with the fin are computed at different angles of attack and compared with the experimental counter parts. Finally the drag computed with the fin is decomposed into pressure and viscous drag at various angels of attack and compared with the baseline cases. Figure 8.6 shows the time history variation of the fluid dynamic forcing in the stream wise (drag) direction. This is compared to the average drag obtained from the experiments. From the figure 8.6 one can see that the running average drag from the computations approaches the experiments.

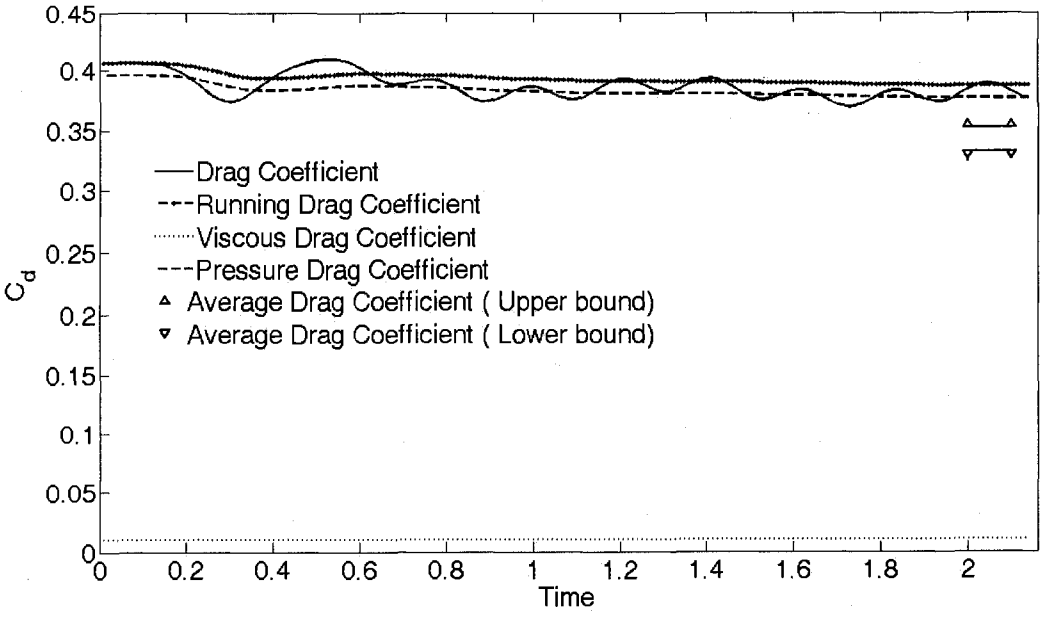


Figure 8.6: Running Pressure, viscous and total drag coefficients compared with the experiments

The time history of the lift co-efficient is also computed and plotted in figure 8.7. The average lift coefficient with fin from computation is 1.3123 compared to 1.178 from the baseline NACA0012 airfoil case. This means that there is a lift enhancement of 11.4% when a fin is attached to a NACA0012 airfoil compared to the baseline airfoil case at angle of attack of 18. The fin kinematics at the trailing edge are computed and plotted in figure 8.8.

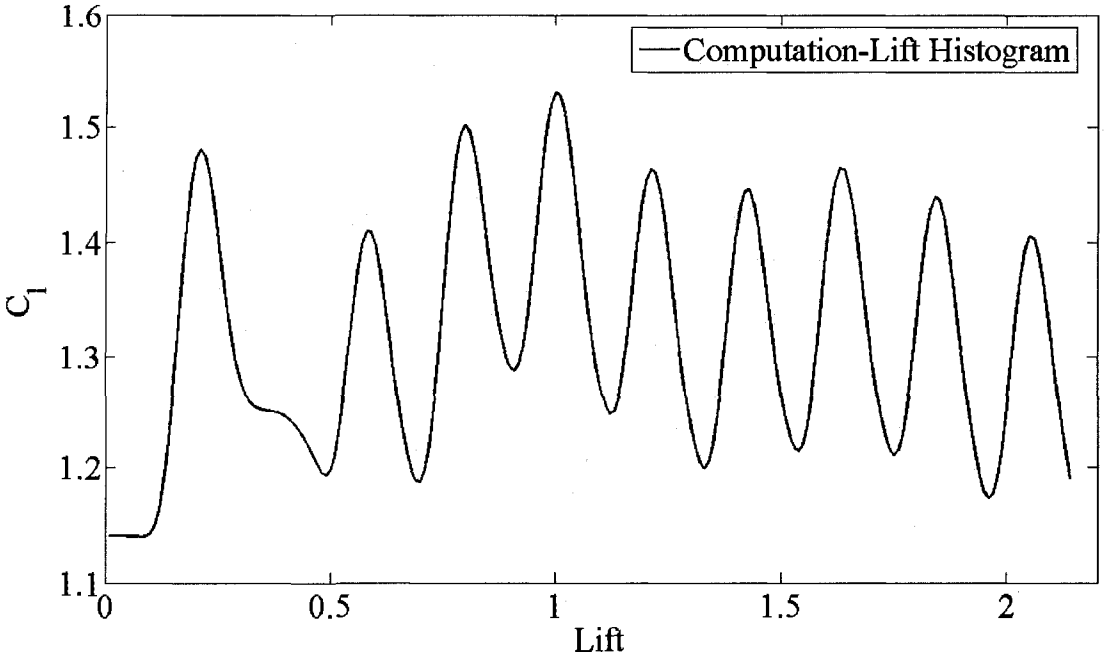


Figure 8.7: Lift coefficient history with the fin

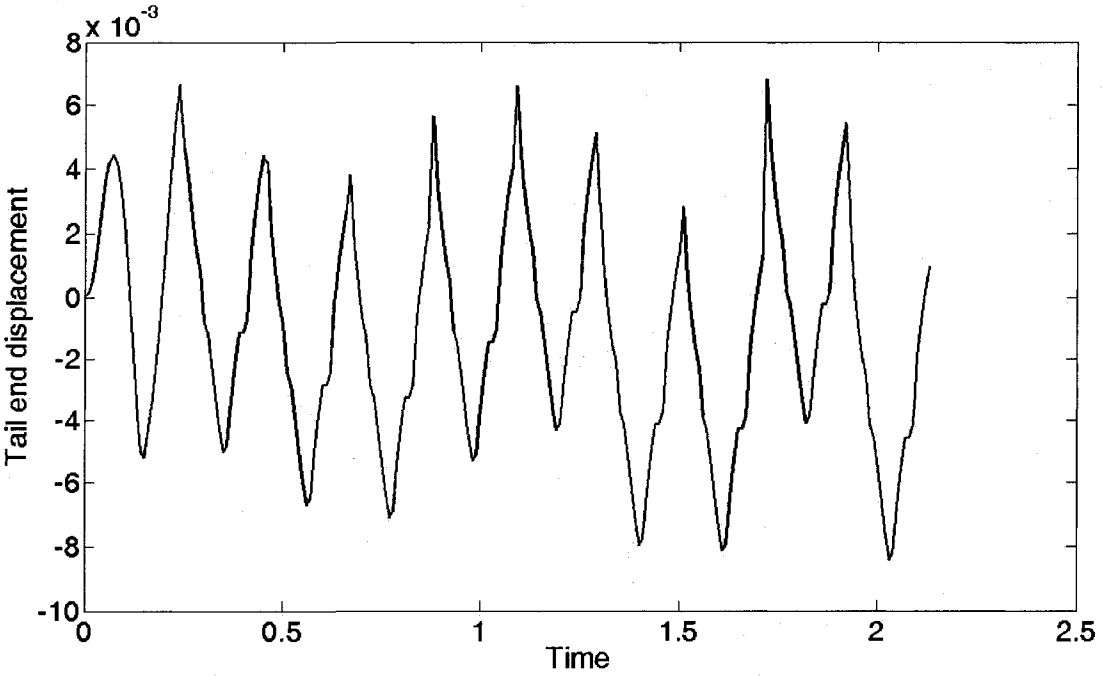


Figure 8.8: Tail end displacements with respect to time

Next the time varying U-velocity contours and stream lines are plotted to give more insight into the computations. In each plot the flow is from left to right side of the page. We can see from the following figures, the region behind the plate the flow separates and a low velocity zone is created. The size of the zone increases as the flow evolves. The streamlines behind the plate show a creation of small separation bubble. The plots are taken when the physical time step of the flow is 1.8, 1.87, 1.96, 2.06 and 2.14 seconds respectively.

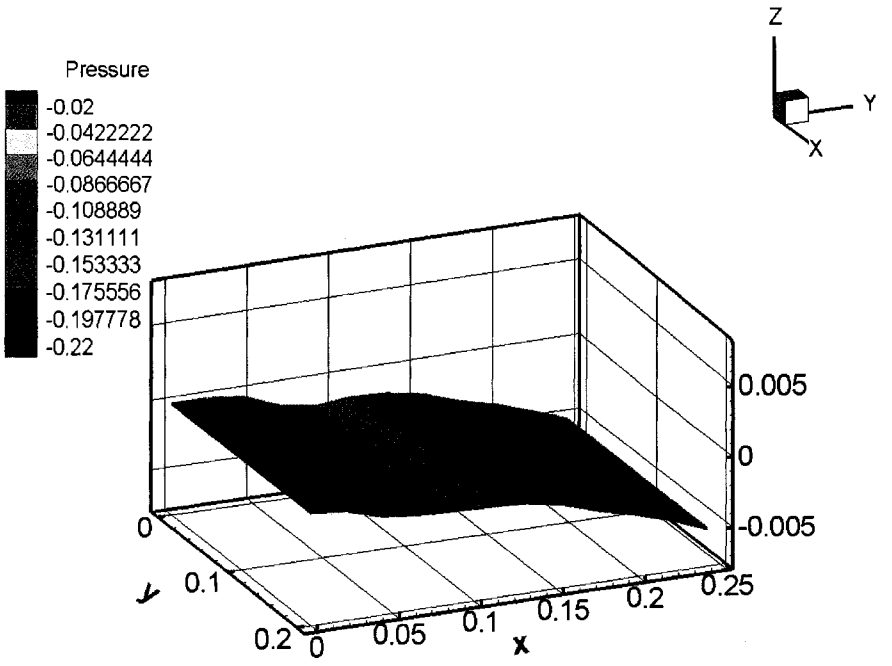


Figure 8.9 a-1) Pressure forces on the fin after 1.8 sec

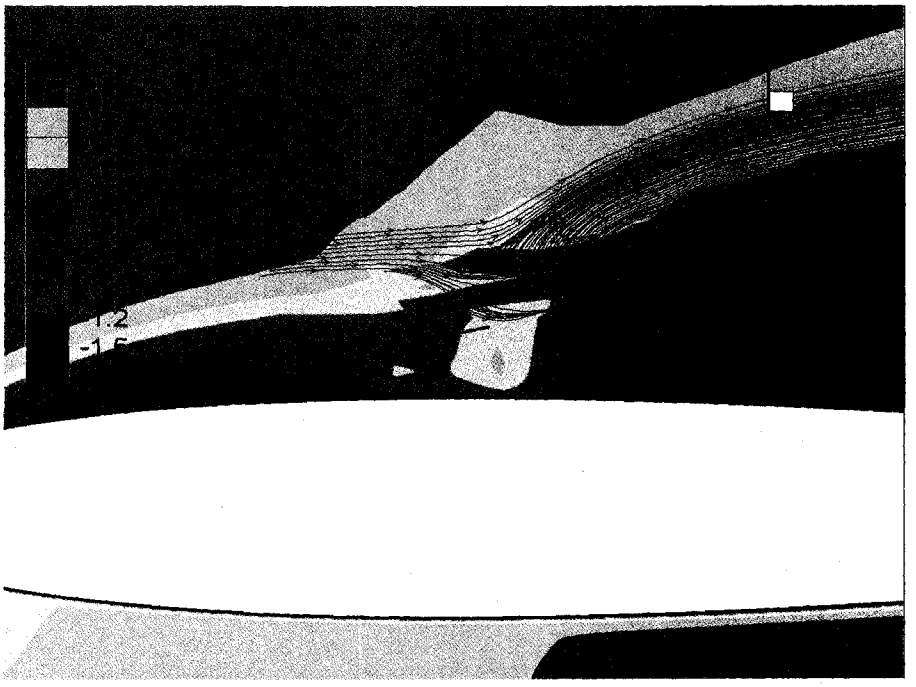


Figure 8.9 a-2) Stream lines with U-velocity contours after 1.8 sec

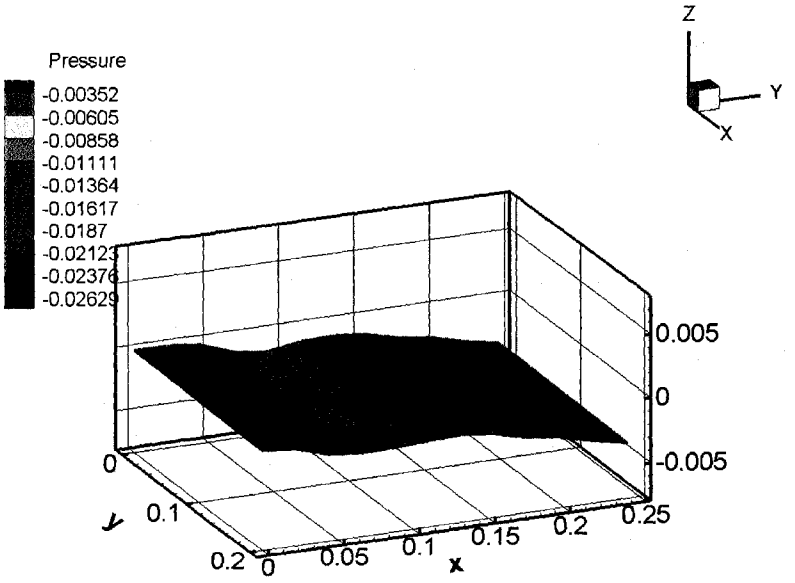


Figure 8.9 b-1) Pressure forces on the fin after 1.87 sec

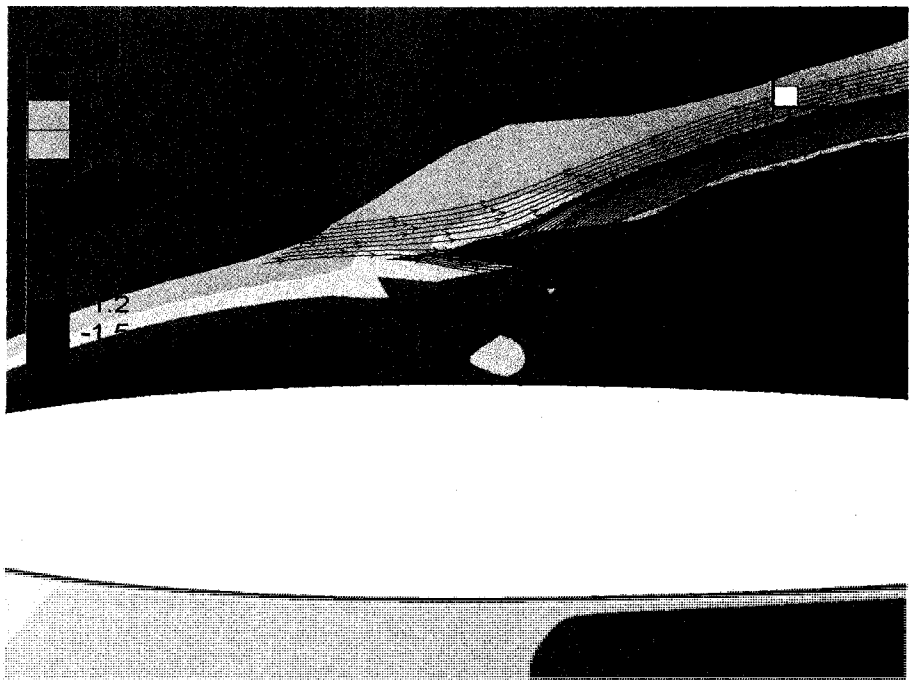


Figure 8.9 b-2) Stream lines and U-velocity contours after 1.87 sec

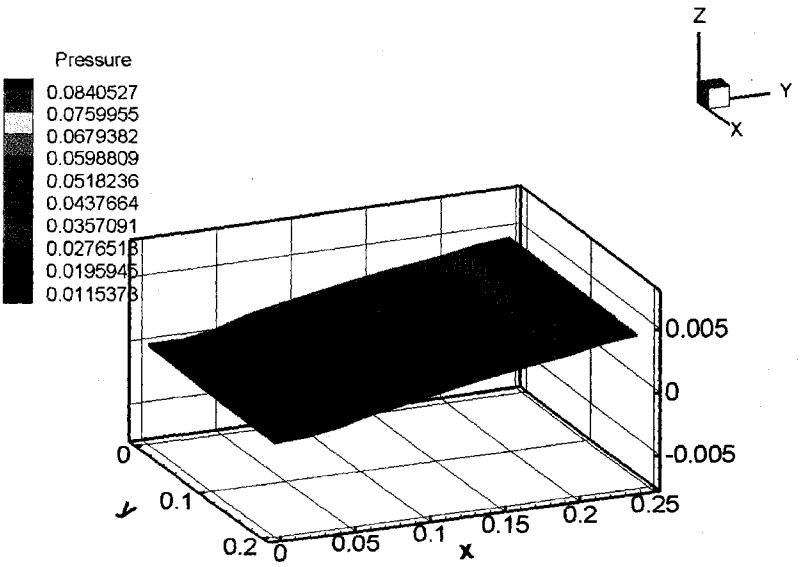


Figure 8.9 c-1) Pressure force on the fin after 1.96 sec

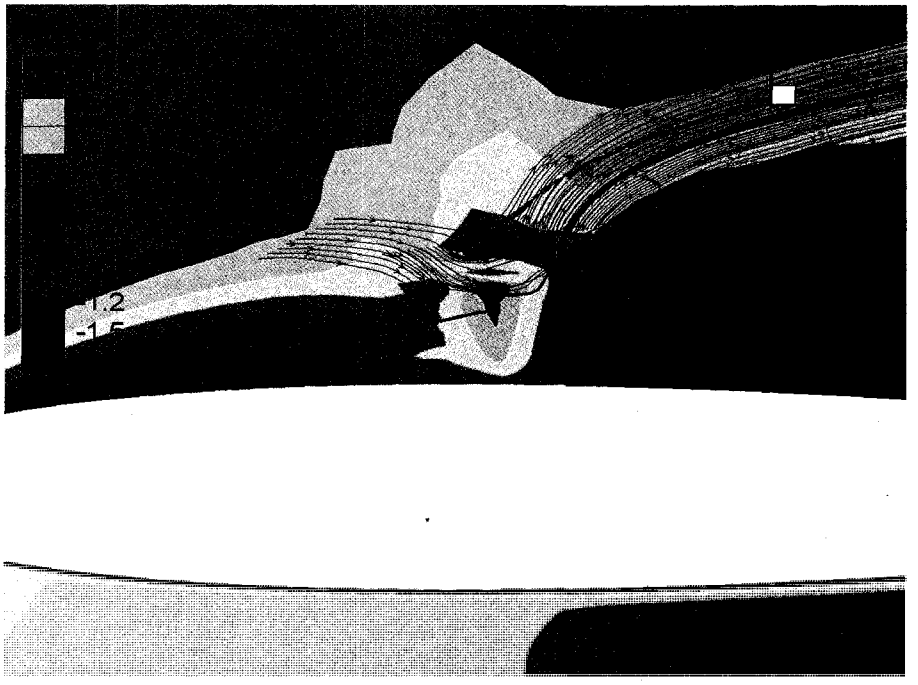


Figure 8.9 c-2) Stream lines and U-velocity contour after 1.96 sec

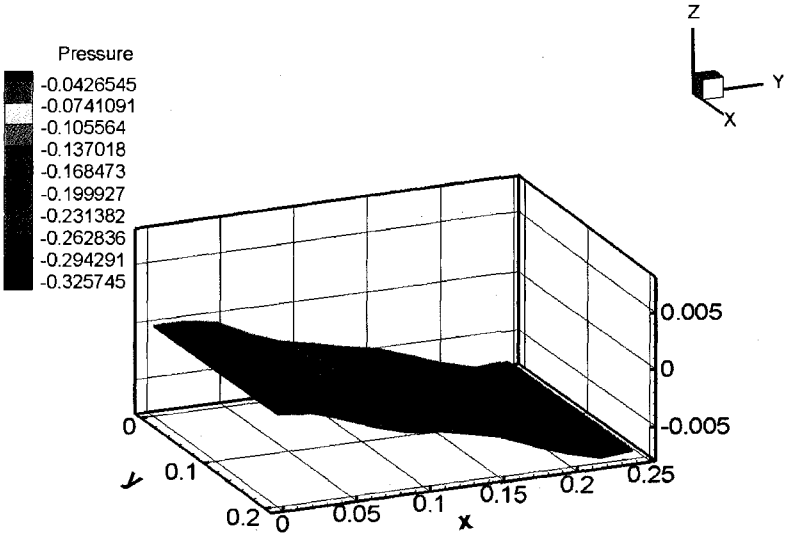


Figure 8.9 d-1) pressure force on the fin after 2.06 sec

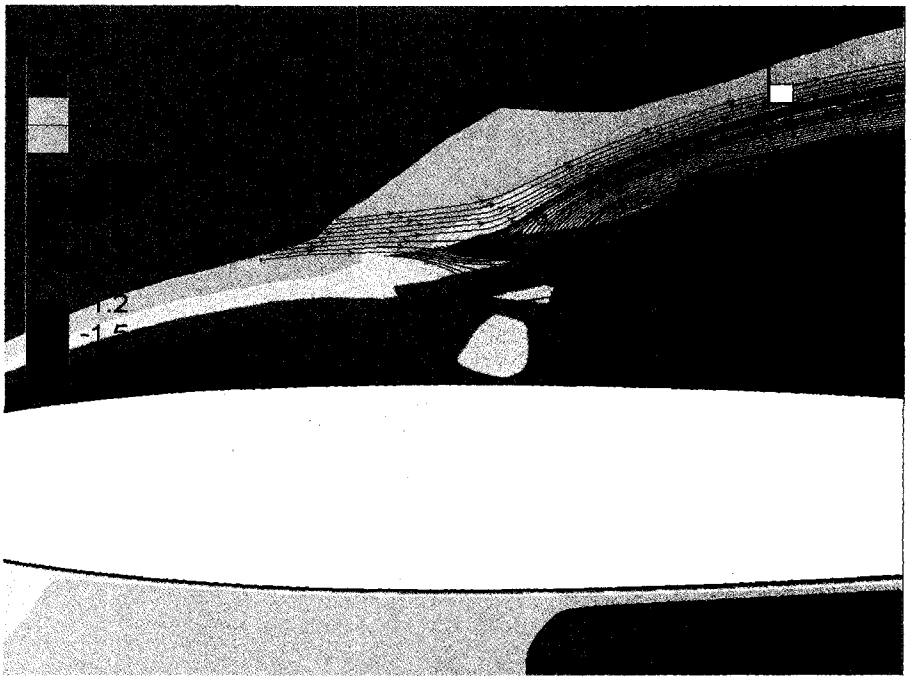


Figure 8.9 d-2) Stream lines and U-velocity contours after 2.06 sec

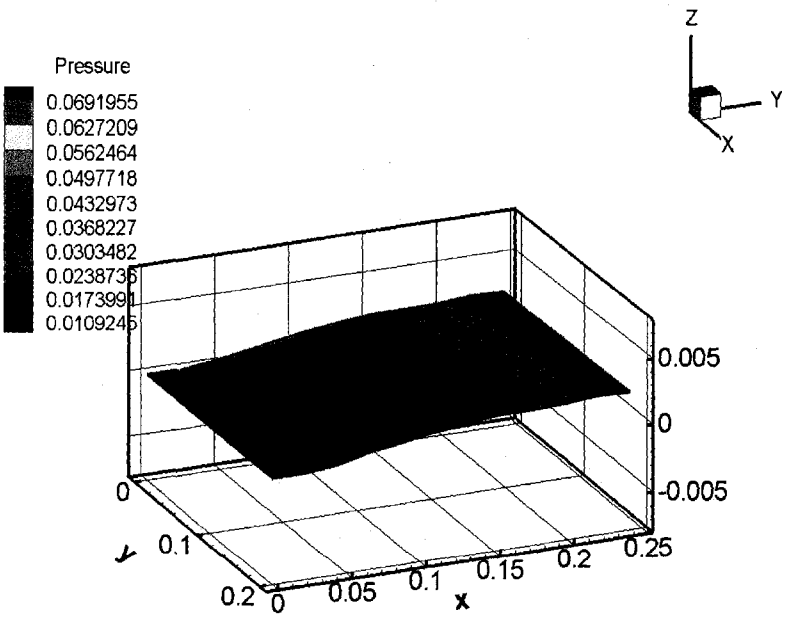


Figure 8.9 e-1) pressure forces on the fin after 2.14 sec

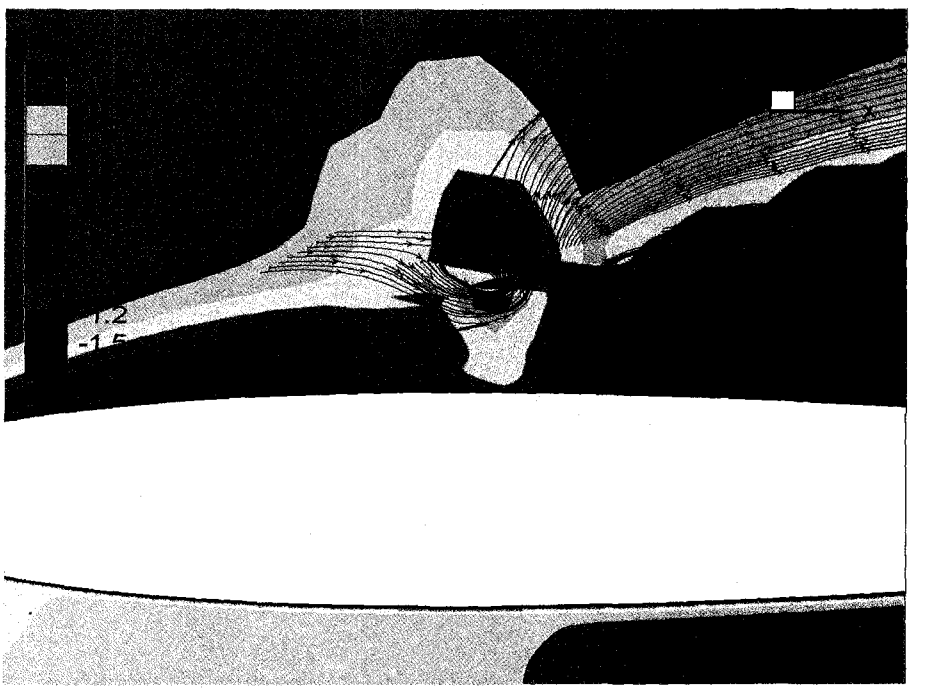


Figure 8.9 e-2) Stream lines and U-velocity contours after 2.14 sec

Finally the time averaged drag coefficient obtained from the experimental baseline and the fin cases at various angles of attack are compared with the computations. From figure 8.10 one can see that the numerical computations follow the trend similar to experiments, drag reduction at higher angles of attack and drag increment at lower angles of attack when a fin is attached to a NACA0012 airfoil compared to the baseline NACA0012 airfoil case. Further this effect was studied by decomposing the overall drag coefficient into

viscous drag and pressure drag as plotted in figures 8.11 and 8.12.

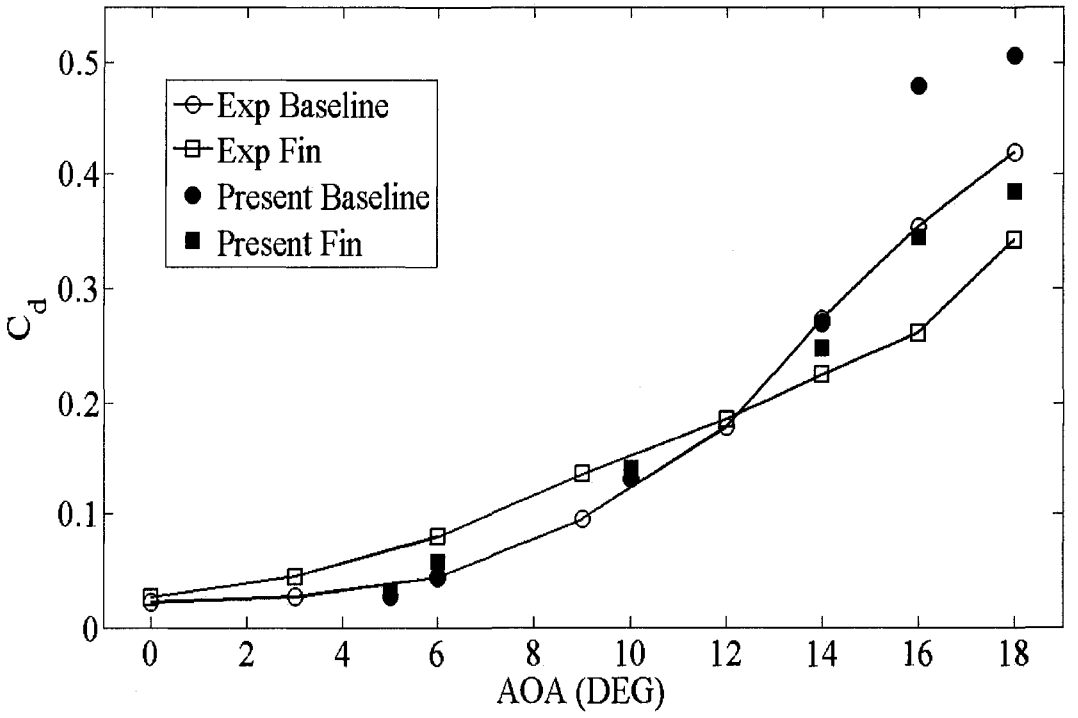


Figure 8.10: Drag coefficient as a function of angle of attack for baseline and fin

From the viscous drag plots at various angles of attack one can see at lower angles of attack the viscous drag with the fin is very high compared to the baseline case. At lower angles of attack the fin motion or flutter is very small and it quickly settles to a steady state, hence becoming an intrusive object to the attached boundary layer, there by increasing the skin friction drag coefficient. This means that at higher angles of

attack the fin passively alters the flow structures to suppress separation and reduce the overall pressure drag. There is a significant viscous drag reduction at higher angles of attack with presence of the fin compared to the baseline airfoil. Hence we can conclude that the overall drag reduction at higher angles of attack with the fin is due to combined reduction of viscous and pressure drags compared to the baseline NACA0012 airfoil.

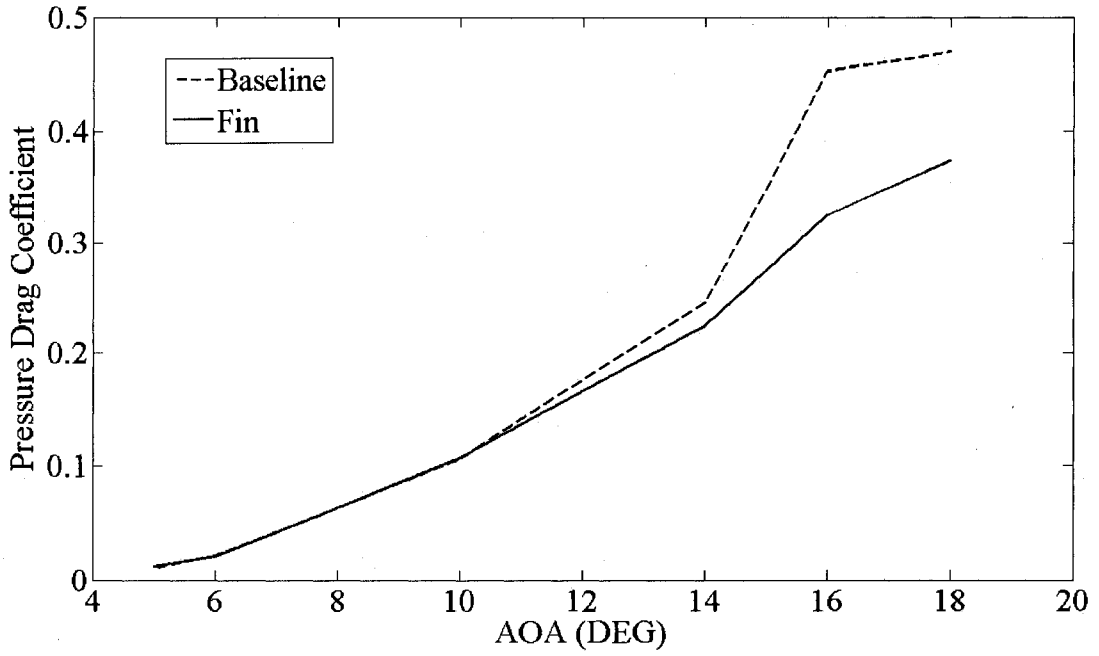


Figure 8.11: Pressure drag as a function of angle of attack for baseline and fin (computations)

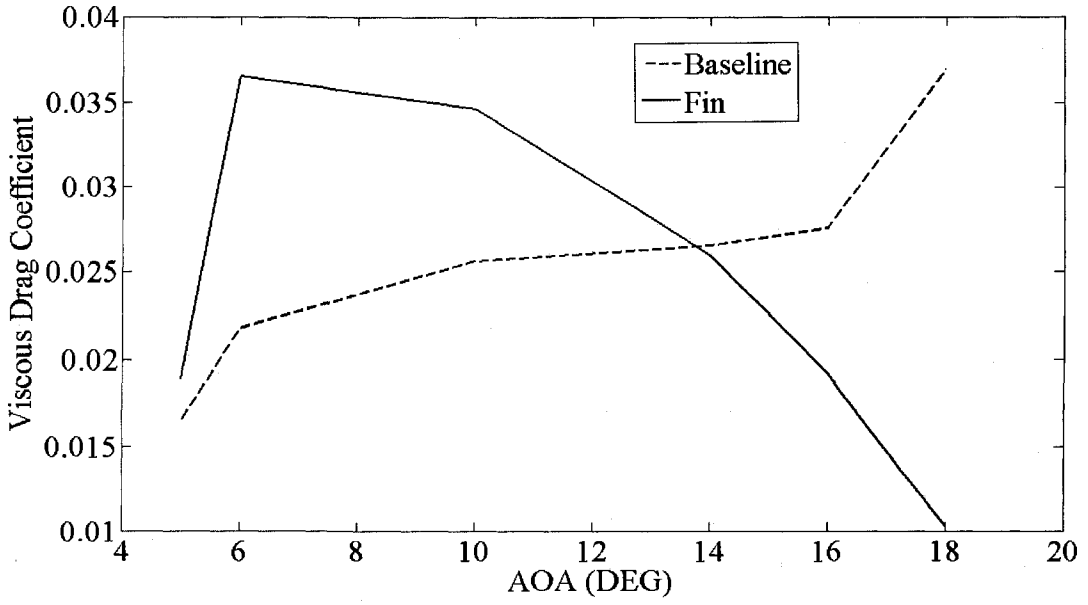


Figure 8.12: Viscous drag as a function of angle of attack for baseline and fin (computations)

CHAPTER IX

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis we developed a new fluid structure interaction (FSI) solver where the fluids code and the structural code are solved separately in a segregated manner and the information at the boundary is passed at the end of each time step. The motion of the boundary is modeled using a new novel procedure called immersed boundary method. The distant advantage of this method is the inherent capacity to solve transient moving boundary problems on fixed Cartesian grids, which eliminates the need for complex mesh regenerations schemes. This not only helps in cutting the computational time but is much easier to numerically add them to any developed in-house CFD code. The direct forcing method of Mohd-yusof [21] used for computing the force that is used to enforce the no-slip condition along the immersed boundary is determined implicitly , which facilitates the use of larger CFL numbers as compared to feed back forcing methods. The individual solvers used in the coupled fluid structure interaction solver are the Navier-Stokes Reynolds averaged solver which is discretized using the finite difference method and the structural solver discretized using finite element method. The CFD solver

is verified and validated for turbulent flows by performing numerical simulations around a NACA0012 airfoil. The CSD solver is also validated against analytical results. Then a number of simulations were modeled starting from thick surfaces to infinitesimally thin surfaces, involving both stationary and moving boundary, with the Navier-stokes solver combined with the immersed boundary method for forcing the no-slip along the boundary. The spatial and temporal order of convergence studies showed that the immersed boundary solver almost approaches second order.

This above discussed immersed boundary technique was used in combination with the Navier-Stokes method and coupled with the subdivision finite element solver for modeling a flow induced passively flexible plate as well as a passive flexible plate attached to the upper surface of the NACA0012 airfoil. The results from the flow induced flapping plate show three states, first a non-linear flapping state, then a periodic flapping stare followed by a near steady state.

The results from the fluid structure interaction study around a NACA0012 airfoil with a flexible fin show that there is an over all drag reduction with fin attached compared to the baseline case in both the experiments formed by Dr. Liu & Dr. Montefort [52] at the

fluids lab at western Michigan University as well as the computations. The tail end frequency of the plate was plotted and compared with the experiments show that the computations were not able to match the experiments in terms of the frequency of the plate motion. The computed time averaged drag coefficients at different angles of attack showed a drag reduction at higher angles of attack with the fin compared to the baseline airfoil.

The fin attached to the upper surface of the NACA0012 airfoil at post-stall angles of attack alters the flow structures to suppress separation, enhance lift and reduces the drag. The implementation of this study has many practical advantages. The present work of applying the flexible passive and active elements on airfoils and wings would be beneficial to a number of problems like gust alleviation for MAV, separation control on LPT blades impinged by unsteady wake, drag reduction of bluff bodies.

Future directions

The level of accuracy of 2D numerical solutions is not possible compared to the experimental values for large separation high angle of attacks flows. The accuracy of grid resolution, accuracy of numerical coupling, accuracy of the immersed boundary methods as

well turbulence modeling are some of the factors that may cause inconsistency.

There are three major studies that need to done in the future as an extension of our research

1. The present immersed boundary scheme devised and applied for various flows is a combination of 1D and 2D linear interpolation schemes. In order for these schemes to be applicable to high Reynolds number flows the grid needs to refined in two or three directions, which might increase the computational cost. Hence higher order interpolation techniques need to be explored and tested.

2. Different grid independent studies need to done to accurately access the discrepancy between the computations and the experiments for the flow control using a flexible fin attached to the NACA0012 airfoil case. Increasing the structural as well as the fluid mesh densities, changing the time step size by increasing the number of time steps in a period of flapping are a couple of strategies that need to be worked out to get a better understanding on the

accuracy of the present FSI solver.

3. The present FSI solver uses unsteady RANS modeling for predicting the turbulent scales in the flow. An extension of our present research would be to employ the same modeling approach for structure, but to combine the immersed boundary approach with a LES model.

BIBLIOGRAPHY

1. Lippisch, A. M., Man Power Flight in 1929, Journal of the Royal Aeronautical Society, Vol. 64, 1960, pp. 395-398.

2. McGowan, A. R. AVST Morphing Project Research Summaries in Fiscal Year 2001, NASA TM-2002-211769, 2001.

3. Glezer, A. and Amitay, M., Synthetic Jets, Annual Review of Fluid Mechanics, 2002.

4. Mittal, R., Kotapati, R. B., Cattafesta, L. N., Numerical Stduy of Resonant Interactions and Flow Control in a Canonical Separated Flow, AIAA paper 2005-1261, 2005.

5. Gad-el-Hak, M., Flow Control: Passive, Active and Reactive Flow Management, Cambridge University Press, 2000.

6. Atik, H., Kim, C. Y., Van Dommelen, L. L., and Walker, J. D. A., Boundary-Layer Separation Control on a Thin Airfoil Using Local Suction, Journal of Fluid Mechanics, 2005.

7. Jones, K. D., Nakashima, M., Bradshaw, C. J., Papadopoulos, J., and Plazer, M. F., On Flow Separation Control by Means of Flapping Wings, Chapter 5, Biomechanics of swimming and flying Eds, 2004.

8. Munday, D. and Jacob, J., Active Control of Separation on a Wing with Oscillating Camber, AIAA J. of Aircraft, Vol. 39, No. 1, 2002.

9. Tallec, P. L. and Mouro, J., Fluid structure interaction with large structural displacements, Computer methods in applied mechanics and engineering, 190:3039-3067, 2001.

10. Bloom, F. J., A monolithical fluid-structure interaction algorithm applied to the piston problem, Computer methods in applied mechanics and engineering, 167:369-391, 1998.

11. Wall, w., Fluid-struktur-Interaktion mit stabilisierten Finiten Elementen, University of Stuttgart, Dissertation, Institute for Baustatik Bericht, Stuttgart, 1999.

12. Giannopapa, C.G., Fluid structure interaction in flexible vessels, Doctoral Dissertation, Kings College London, 2004.

13. Liou, W. W. and Pantula, S. R., Unsteady Flow Calculations for Flexible Thin Plate, AIAA Paper, 2007-4339, 2007.

14. Fadlun, E.A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics, 161, 35-60, 2000.

15. Launder, B. E. and Sharma, B. I., Application of the Energy-Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc, Letters in Heat and Mass Transfer, Vol. 1, No. 2, pp. 131-138, 1974.

16. Peskin, C. S., Flow patterns around heart valves: A numerical method, Journal of Computational Physics, 10:252-271, 1972.

17. Goldstein, D., Handler, R., Sirovich L., Modeling a no-slip flow boundary with an external force field. Journal of Computational Physics,105:354 -366, 1993.

18. Mittal, R., Iaccarino, G., Immersed boundary methods, Annual review of Fluid Mechanics, 37, 239-261, 2005.

19. Goldstein, D., Tuan, T. C., Secondary flow induced by riblets, Journal of Fluid Mechanics, 363:115-51, 1998.

20. Saiki, E. M., Biringen, S., Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, Journal of Computational Physics, 123:450-65, 1996.

21. Mohd-Yusof,J., Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries, CTR Annual Research Briefs, NASA Ames/Stanford University, 1997.

22. Kim, J., Kim, D., Choi, H., An immersed-boundary finite-volume method for simulations of flow in complex geometries, Journal of Computational Physics, 171:132-50, 2001.

23. Majumdar, S., Iaccarino, G., Durbin, P., RANS solvers with adaptive structured boundary non-conforming grids, CTR Annual Research Briefs, NASA Ames/Stanford University, 2001.

24. Davidsson, L., class notes, Applied Mechanics Department, Chalmers university of technology, http://www.tfd.chalmers.se/~lada/comp_fluiddynamics.

25. Launder, B.E., and Sharma, B.I., (1974), Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc, Letters in Heat and Mass Transfer, Vol. 1, 1974, pp. 131-138

26. Jameson, A., and Baker, T.J, (1983), Solution of the Euler equations for complex configurations, AIAA Paper 83-1929

27. Simo, J., C., and Fox, D., D., On a stress resultant geometrically exact shell model. Part I: formulation and optimal parameterization. *Computer Methods in Applied Mechanics and Engineering*, 72:267-304, 1989

28. Cirak, F., Ortiz, M., and Schroder, P., Subdivision surface: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering*, 47: 2039-2072, 2000

29. Loop, C., T., Smooth subdivision surfaces based on triangles. Master's thesis, The University of Utah, August 1987

30. Hughes, T.J., Liu, W.K., Zimmermann, T.K., Lagrangian-Eulerian finite element formulation for incompressible viscous flows. Computational Methods in Applied Mechanics and Engineering, 29:329-349, 1981.

31. Gilmanov, A. and Sotiropoulos, F., A hybrid immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, Journal of computational physics, 2005.

32. Zeeuw, De and Powell, K.G., An adaptively refined Cartesian mesh solver for the Euler equations, AIAA Paper, 91-1542-cp.

33. Bayyuk, S., A., Powell, K.G. and van Leer, B., A simulation technique for 2-D unsteady inviscid flows around arbitrary moving and deforming bodies of arbitrary geometry, AIAA Paper 93-3391-CP, 1993.

34. Quirk, J.J., An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies, Computational Fluids 23, pp 125-142, 1994.

35. Udaykumar, H.S., Shyy, W., Rao, M.M., Elafint, A mixed Eulerian- -Lagrangian method for fluid flows with complex and moving boundaries, International Journal of Numerical Methods Fluids 22-691, 1996.

36. Hegna, H.A., The numerical solution of incompressible turbulent flow over airfoils, *AIAA Paper*, AIAA-81-0047, 1981.

37. Jacobs, E., and Sherman, A., Airfoil section characteristics as affected by variations of the Reynolds Number, NACA Report 536, 1937.

38. Launder, B.E., and Shima, N., (1989), "Second-moment closure for the near-wall sub layer: development and

application," AIAA Journal, Vol. 27, No. 10, pp. 1319-1325.

39. MeshPiolt, Shore CFD, Software Package, Scotland, UK, 2003.

40. Yang, B.R. and Bhatti, M.A, Nonlinear static and dynamic analysis of plates, Journal of engineering mechanics, ASCE 111:175-187, 1985.

41. Fornberg B., A numerical study of steady viscous flow past a circular cylinder, Journal of Fluid Mechanics, 98:819-855, 1980.

42. Kiris, C. and Kwak, D., Numerical solution of incompressible Navier-Stokes equations using a fractional-step approach, Computers and Fluids, 30:829-851, 2001.

43. Peng , y., Shu, C. and Chew, Y.T., Niu , X.D., Lu X.Y., Application of multi-block approach in the immersed boundary-lattice Boltzmann method for viscous fluid flows , Journal of Computational Physics, 218 460-478, 2006.

44. Lockard, D.P., Luo, L.S., Milder, S.D. and Singer, B.A., Evaluation of power flow for aerodynamic applications, Journal of Statistical Physics 107 423-478, 2002.

45. Roache, P.J., "Quantification of uncertainty in computational fluid dynamics," Ann. Rev. Fluid Mech., 29, pp.123-160 , 1997.

46. LeVeque, R.J. and Oliger, J., Numerical-methods based on additive splittings for hyperbolic partial-differential equations, Mathematics of Computation, 40 469-497, 1983.

47. Tianshu, L., Montefort, J., Liou, W.W., Pantula, S. R. and Shams, Q., Lift enhancement by Static Extended Trailing Edge, Journal of Aircraft 2007,Vol 44, 1939-1947.

48. Ahuja, S., and Rowley, C. W., Low-dimensional models for feedback Stabilization of unstable steady states. AIAA Paper 2008-553.

49. Yu, D., Mei, R. and Shyy, W., A multi-block lattice Boltzmann method for viscous fluid flows, International Journal of Numerical Methods of Fluids 39 99-120, 2002.

50. Van Driest, E.R., (1956), "On turbulent flow near a wall," Journal of Aerospace Science, Vol. 23, pp. 1007-1011.

51. Sotiropoulos, F., and Abdallah, S., (1992), "A Primitive variable method for the solution of three-dimensional incompressible viscous flows," Journal of Computational Physics, Vol. 103, pp. 336-349.

52. Tianshu, L., Montefort, J., Liou, W.W., Pantula, S. R., Yang, Y., and Shams, Q., Post-Stall Flow Control Using Flexible Fin on Airfoil, AIAA 09. (paper Accepted)

53. Tyagi, M. and Acharya,S., Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method, International journal for numerical methods in fluids, 48:691-722, 2005.

54. Fluent 6.3 User's Manual, Fluent inc., 2008.