2008-01-01

Numerical Study of Limit Cycle Oscillation Using Conventional and Supercritical Airfoils

Felipe Manuel Loo
University of Miami, f.loo@umiami.edu

UNIVERSITY OF MIAMI

NUMERICAL STUDY OF LIMIT CYCLE OSCILLATION USING A CONVENTIONAL AND SUPERCRITICAL AIRFOIL

By

Felipe M. Loo

A THESIS

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science

Coral Gables, Florida

December 2008

UNIVERSITY OF MIAMI

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

NUMERICAL STUDY OF LIMIT CYCLE OSCILLATION USING A CONVENTIONAL AND SUPERCRITICAL AIRFOIL

Felipe M. Loo

Approved:

GeCheng Zha, Ph.D. Professor, Department of Mechanical and Aerospace Engineering	Terri A. Scandura, Ph.D. Dean of the Graduate School
Singiresu S. Rao, Ph.D. Chairman, Department of Mechanical and Aerospace Engineering	Hongtan Liu, Ph.D. Professor, Department of Mechanical and Aerospace Engineering
Xiangying Chen, Ph.D. Professor, Department of Mechanical and Aerospace Engineering	Alexander Pons, Ph.D. Director, Career Services, Office of Dean, School of Business Administration

LOO, FELIPE (M.S., Mechanical Engineering)

Numerical Study of Limit Cycle Oscillation
Using Conventional and Supercritical Airfoils (December, 2008)

Abstract of a thesis at the University of Miami

Thesis supervised by Professor Gecheng Zha
No. of pages in text.(94)

Limit Cycle Oscillation is a type of aircraft wing structural vibration caused by the non-linearity of the system. The objective of this thesis is to provide a numerical study of this aeroelastic behavior. A CFD solver is used to simulate airfoils displaying such an aeroelastic behavior under certain airflow conditions. Two types of airfoils are used for this numerical study, including the NACA64a010 airfoil, and the supercritical NLR 7301 airfoil. The CFD simulation of limit cycle oscillation (LCO) can be obtained by using published flow and structural parameters. Final results from the CFD solver capture LCO, as well as flutter, behaviors for both wings. These CFD results can be obtained by using two different solution schemes, including the Roe and Zha scheme. The pressure coefficient and skin friction coefficient distributions are computed using the CFD results for LCO and flutter simulations of these two airfoils, and they provide a physical understanding of these aeroelastic behaviors.

Contents

List of Figures vi

List of Tables xi

1 Introduction **1**

1.1 Flutter Oscillations 2

1.2 Limit Cycle Oscillations 3

2 Governing Flow Equations **4**

2.1 Formulation of Navier-Stokes Equation 4

3 Governing Structural Equations **10**

3.1 Heave and Pitching Equations 11

4 Discretization of Equations **14**

4.1 Structural Equation Formulation 15

4.2 Flow Equations 16

5 The CFD solver **18**

5.1 CFD Simulation Process 19

5.2 Upwind Schemes 20

5.2.1 The Roe Scheme 20

5.2.2 The Zha Scheme 23

6 CFD Mesh Generation **26**

6.1 Mesh Generation Mathematics 27

7 Results **31**

7.1 Method of CFD Computation 32

7.2 Validation Cases 33

7.3 LCO and Flutter Cases: NACA 64a010 and NLR 7301 35

7.4 Results I: Validation Cases 38

7.5 Results II: LCO and Flutter Cases 43

7.6 Results III: Skin Friction and Pressure Distributions 52

8 Conclusion **62**

Appendix A CFD Post-Processing Code **64**

A.1 Code: cdlt_all.f90 64

Appendix B The Scientific Subroutine Library **77**

B.1 The Discrete Fourier Transform Subroutine 77

Appendix C Transformations of Shear Stresses **79**

Appendix D Derivation of Equations **80**

D.1 Pitching Mode without Damping 80

D.2 Pitching Mode with Damping 82

D.3 Heave Mode without Damping 84

D.4 Heave Mode with Damping 86

D.5 The CFD Structural Equations 87

Appendix E Derivation of Flow and Structural Parameters **89**

References **92**

List of Figures

3.1 Profile Diagram of an Airfoil 11

6.1 Mesh boundary descriptions using O-type topology. 28

7.1 NACA 64a010 geometric profile 35

7.2 NLR 7301 geometric profile 35

7.3 Pressure distribution of NLR 7301 for steady state condition (Case III), $M_a = 0.78$, Re= 1700000.0, experimental data obtained from Reference [9] 38

7.4 Pressure Distribution of NACA 64a010 for force-induced vibrating condition (Case II), $M_a = 0.8$, Re = 12560000.0, Experimental data obtained from Reference [33] 39

7.5 Displacement trajectory of a cylinder in laminar vortex induced oscillating condition (Case I), GCL = 0.0 $\zeta = 0.63326$, $\mu_s = 1.2732$, $\bar{u} = 1.5915$ 39

7.6 Drag and lift coefficient profile for a cylinder in laminar vortex induced condition (Case I), GCL = 0, $\zeta = 0.63326$, $\mu_s = 1.2732$, $\bar{u} = 1.5915$. 40

7.7 Lift coefficient profile of NACA 64a010 for force induced vibrating condition (Case II), at $\alpha_o = 0.0$, $\alpha_A = 1.01$, $K_c = 0.202$, Re = 12560000.0, and $\mathrm{M}_a = 0.8$ 40

7.8 Torque coefficient profile of NACA 64a010 for force induced vibrating condition (Case II), at $\alpha_o = 0.0$, $\alpha_A = 1.01$, $K_c = 0.202$, Re = 12560000.0, and $\mathrm{M}_a = 0.8$ 41

7.9 Subfigures A to J show the time history of Mach contour plots for NACA 64a010 in critical flutter condition (Case V) in order to capture oscillatory motion of shock waves 42

7.10 Heave oscillation of NACA 64a010 for critical flutter condition (Case V), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$ 45

7.11 Pitch oscillation of NACA 64a010 for critical flutter condition (Case V), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$ 45

7.12 Heave oscillation of NACA 64a010 for damping condition (Case IV), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$. 46

7.13 Pitch oscillation of NACA 64a010 for damping condition (Case IV), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$. 46

7.14 Heave oscillation of NACA 64a010 for flutter condition (Case VI), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$. 47

7.15 Pitch oscillation of NACA 64a010 for flutter condition (Case VI), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.825$. 47

7.16 Heave oscillation of of NACA 64a010 for LCO condition (Case VII), $\alpha_o = 0.0$, Re = 12560000.0, $\mathrm{M}_a = 0.925$ 47

7.17 Pitch oscillation of NACA 64a010 for LCO condition (Case VII), $\alpha_o =$ 0.0, Re = 12560000.0, $M_a = 0.925$ 48
7.18 Heave oscillation of NLR 7301 for flutter condition (Case VIII), $\alpha_o =$ 0.65, Re = 1700000.0, $M_a = 0.753$ 48
7.19 Pitch oscillation of NLR 7301 for flutter condition (Case VIII), $\alpha_o =$ 0.65, Re = 1700000.0, $M_a = 0.753$ 48
7.20 Left: Heave phase diagram of NLR 7301 for flutter condition (Case VIII). Right: Pitch phase diagram of NLR 7301 for flutter condition (Case VIII) 49
7.21 Heave oscillation of NLR 7301 for LCO condition (Case IX), $\alpha_o = 0.0$, Re = 1700000.0, $M_a = 0.753$ 49
7.22 Pitch oscillation of NLR 7301 for LCO condition (Case IX), $\alpha_o = 0.0$, Re = 1700000.0, $M_a = 0.753$ 49
7.23 Left: Heave phase diagram for NLR 7301 in LCO condition (Case IX). Right: Pitch phase diagram for NLR 7301 in LCO condition (Case IX) 50
7.24 Instantaneous Skin friction distribution of NACA 64a010 in flutter oscillating condition (Case V) 53
7.25 Real-part of the unsteady pressure distribution of NACA 64a010 in flutter oscillating condition (Case V) 53
7.26 Imaginary-part of the unsteady pressure distribution of NACA 64a010 in flutter oscillating condition (Case V) 54
7.27 Time averaged pressure distribution of NACA 64a010 in flutter oscillating condition (Case V) 55

7.28 Instantaneous skin friction distribution of NACA 64a010 in LCO oscillating condition (Case VII) 55

7.29 Real-part of the unsteady pressure distribution of NACA 64a010 in LCO oscillating condition (Case VI) 56

7.30 Imaginary-part of the unsteady pressure distribution of NACA64a010 in LCO oscillating condition (Case VI) 57

7.31 Time averaged pressure distribution of NACA 64a010 in LCO oscillating condition (Case VI) 57

7.32 Instantaneous skin friction distribution of NACA 64a010 in damped oscillating condition (Case IV) 58

7.33 Real-part of the unsteady pressure distribution of NACA 64a010 in damped oscillating condition (Case IV) 58

7.34 Imaginary-part of the unsteady pressure distribution of NACA 64a010 in damped oscillating condition (Case VI) 59

7.35 Time averaged pressure distribution of NACA 64a010 in damped oscillating condition (Case VI) 59

7.36 Instantaneous skin friction distribution of NLR 7301 in LCO condition (Case IX) 60

7.37 Real-part of the unsteady pressure distribution of NLR 7301 in LCO condition (Case IX) 60

7.38 Imaginary-part of the unsteady pressure distribution of NLR 7301 in LCO condition (Case IX) 61

7.39 Time averaged pressure distribution of NLR 7301 in LCO condition (Case IX) 61

List of Tables

7.1 Flow and structural parameters for CFD simulation runs to obtain results for test and validation purposes 34

7.2 Flow and structural parameters for CFD simulations of NACA64a010 36

7.3 Flow and Structural parameters for CFD simulations of NLR 7301 . . 37

7.4 Comparison of Case VIII (flutter) and Case IX (LCO) of NLR 7301 with experimental data available in Reference [9] 50

7.5 Comparison of Case VIII (flutter) and Case IX (LCO) of NACA 64a010 with experimental data available in Reference [4] 51

Chapter 1

Introduction

Modern aeroelasticity is a multidisciplinary science of the study of flow behaviors around structures [1]. When computers were not yet available to assist in the computational efforts to solving complex , analytical methods of solutions to describe flow phenomena such as shock waves and flutter were mainly used [2]. In the course of time, advancements in computer applications provided the capability to solve more complex aeroelastic problems using computational fluid dynamics (CFD). Also, improved experimental tools have been devised for measuring a variety of structural and flow parameters, thus allowing for a better physical understanding, as well as the discovery, of aeroelastic behaviors. Also, the development of efficient solution schemes for CFD provided for more accurate aeroelastic computations. These advancements have collectively contributed in the discovery of an aeroelastic behavior known as limit cycle oscillation (LCO). In recent times, there has been several scientific works devoted to the understanding of the physics behind LCO behaviors. The objective of this thesis is to provide a numerical study of LCO using a CFD solver in order to provide a physical understanding of, as well as to prove the capability of the CFD solver to capture, this aeroelastic phenomenon. This study begins with the understanding of the fundamental concepts behind LCO.

1.1 Flutter Oscillations

The fundamental concept of LCO is similar to that of the aeroelastic phenomenon known as flutter. This occurs when a wing undergoes a flow-induced, self-sustainable vibration with amplitude continuously increasing, eventually stabilizing to a constant amplitude vibration. There exists a critical flow condition that causes the wing to undergo a periodic heave and pitching motion with constant amplitude, as demonstrated in References [3] [4]. This is known as the critical flutter phenomenon. In real case scenarios, a wing that is going through a regular flutter behavior is vulnerable to structural failures. This could be avoided by knowing the flow conditions that causes critical flutter, and designing a wing that could at least support the aerodynamic forces at critical flutter conditions. According to Reference [5], one of the aerodynamic attributes of flutter is that the shock waves move almost in a sinusoidal manner and remains present during the complete cycle of oscillation, although its strength varies. This effect has been demonstrated in experimental and simulation tests, such as in Reference [6]. This has also been confirmed by results obtained from the CFD solver used for this thesis study. The periodical motion of shock waves around an airfoil causes it to vibrate periodically in heave and pitch.

1.2 Limit Cycle Oscillations

According to literature, a wing subject to a certain flow condition may exhibit two different flow phenomena, including limit cycle oscillation and flutter [7]. These aeroelastic behaviors are both similar in terms of their aerodynamic nature, and the condition that both exhibit constant amplitude structural oscillations after a certain period of time. They are basically described when the amplitude of the structural oscillations grow in time, and gradually stabilizing into a state of constant amplitude oscillation. Similarly to flutter, the aerodynamic nature of LCO is that of a shock wave motion around the structures that induces a periodic flow separation at the trailing edge, thus providing a damping effect that stabilizes the structural oscillation of the object [8]. The difference between these two aeroelastic phenomena is in the growth rate and amplitude size of the structural oscillation during their transient phase. This has been demonstrated in various studies, such as in References [9], [10], [11], and [12]. The objective behind these studies has been to investigate if LCO could be used to extend the operational flight regime, even though it is seen as an undesirable vibration that limits the aircraft flight performance. This thesis attempts to investigate LCO and flutter behaviors by using a CFD solver to capture these aeroelastic behaviors from the NLR 7301 and the NACA 64a010 wing,

Chapter 2

Governing Flow Equations

The CFD solver that is used to capture LCO uses the Reynolds averaged Navier-Stokes equations (RANS) to obtain a solution of the flowfield. These equations arise from applying the concept of mass, momentum, and energy balance equations. The RANS model is supplemented with turbulence model such as the Baldwin Lomax model, which is what this thesis study uses to obtain its results. The RANS model that is embedded in the CFD solver can be used to simulate two and three dimensional flow around moving or non-moving objects, as provided in Reference [13]. This study focuses on CFD computations of the two dimensional geometry of both the NLR 3701 and the NACA 64a010 wings.

2.1 Formulation of Navier-Stokes Equation

The Navier-Stokes equations consists of the three principal equations of heat transfer and fluid dynamics; namely, the conservation of mass, the conservation of momentum, and the conservation of energy [14]. The RANS equations can be collectively combined to form a single vector equation. This equation is then expanded with a source term representing external heating and forces. The end result is the integral form the Navier-Stokes equation, expressed as:

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{Q}d\Omega + \oint_{\partial\Omega}\left(\vec{W}_c - \vec{W}_v\right)dS = \int_{\Omega}\vec{K}d\Omega \tag{2.1}$$

The vector $\vec{Q}$ is known as the conservative vector, and it consists of parameters related to the physical properties of the flow. The vector $\vec{W}_c$ represents the convective flux vector, which contains the expressions related to the convective transport of quantities in the fluid. The vector $\vec{W}_v$ represents the viscous flux vector, which contains the expressions related to viscous stresses. Lastly, the vector $\vec{K}$ represents the source term which is comprised of volume sources due to body forces and volumetric heating. Supposed that the convective and viscous fluxes are continuous such that a first-order differentiation of these vectors is possible, then the Navier-Stokes equation can be transformed from an integral to a differential form by first applying Gauss' theorem [15], yielding:

$$\frac{\partial}{\partial t}\int_{\Omega}\vec{Q}d\Omega + \int_{\Omega}\nabla\cdot\left(\vec{W}_c - \vec{W}_v\right)d\Omega = \int_{\Omega}\vec{K}d\Omega. \tag{2.2}$$

Equation 2.2 can then be integrated with respect to a control volume, Ω, resulting in the desired differential form of the Navier-Stokes equation, that is:

$$\frac{\partial\vec{Q}}{\partial t} + \vec{\nabla}\cdot\left(\vec{W}_c - \vec{W}_v\right) = \vec{K} \tag{2.3}$$

The Cartesian coordinates of equation 2.3 are transformed into curvilinear coordinates, as shown in equation 2.4. This is done by applying the equations of metric transformation [16], leading to equation 2.5, which is the Navier-Stokes equation in non-dimensional form and in the computational space.

$$\begin{aligned} \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \zeta &= \zeta(x, y, z, t) \end{aligned} \tag{2.4}$$

$$\frac{\partial \vec{Q}^*}{\partial t} + \frac{\partial \vec{W}_{c,1}}{\partial \xi} + \frac{\partial \vec{W}_{c,2}}{\partial \eta} + \frac{\partial \vec{W}_{c,3}}{\partial \zeta} = \frac{1}{\mathrm{Re}} \left(\frac{\partial \vec{W}_{v,1}}{\partial \xi} + \frac{\partial \vec{W}_{v,2}}{\partial \eta} + \frac{\partial \vec{W}_{v,3}}{\partial \zeta} \right) \tag{2.5}$$

The vector $\vec{Q}^*$ of equation 2.5 is defined as $\vec{Q}^* = \frac{\vec{Q}}{J}$, where J is the determinant of the coordinate transformation Jacobian. Equation 2.5 can be supplemented with moving grid models so it can be applied for cases involving both moving and non-moving grids. Also, since equation 2.5 is suitable for laminar CFD computations, it is supplemented with a turbulence model in order to solve for problems involving turbulent flows. The CFD solver that is used for this study makes use of the curvilinear, nondimensional, moving-grid form of the Reynolds-averaged Navier-Stokes equation with Favre mass averaged terms for the Baldwin-Lomax turbulence model. The conservative, convective and viscous vectors of equation 2.5, are defined in equations 2.6, 2.7, and 2.8. They are implemented in the present CFD solver.

$$\vec{Q} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{\omega} \\ \bar{\rho}\tilde{e} \end{bmatrix} \tag{2.6}$$

$$\begin{aligned} \vec{W}_{c,1} &= \frac{1}{J}\left(\xi_t\vec{Q} + \xi_x\vec{E}_c + \xi_y\vec{F}_c + \xi_z\vec{G}_c\right) \\ \vec{W}_{c,2} &= \frac{1}{J}\left(\eta_t\vec{Q} + \eta_x\vec{E}_c + \eta_y\vec{F}_c + \eta_z\vec{G}_c\right) \\ \vec{F}_{c,3} &= \frac{1}{J}\left(\zeta_t\vec{Q} + \zeta_x\vec{E}_c + \zeta_y\vec{F}_c + \zeta_z\vec{G}_c\right) \end{aligned} \tag{2.7}$$

$$\begin{aligned} \vec{W}_{v,1} &= \frac{1}{J}\left(\xi_t\vec{Q} + \xi_x\vec{E}_v + \xi_y\vec{F}_v + \xi_z\vec{G}_v\right) \\ \vec{W}_{v,2} &= \frac{1}{J}\left(\eta_t\vec{Q} + \eta_x\vec{E}_v + \eta_y\vec{F}_v + \eta_z\vec{G}_v\right) \\ \vec{W}_{v,3} &= \frac{1}{J}\left(\zeta_t\vec{Q} + \zeta_x\vec{E}_v + \zeta_y\vec{F}_v + \zeta_z\vec{G}_v\right) \end{aligned} \tag{2.8}$$

where:

$$\vec{E}_c = \begin{bmatrix} \bar{\rho}\tilde{u} \\ \bar{\rho}\tilde{u}\tilde{u} + \tilde{p} \\ \bar{\rho}\tilde{u}\tilde{v} \\ \bar{\rho}\tilde{u}\tilde{\omega} \\ (\bar{\rho}\tilde{e} + \tilde{p})\,\tilde{u} \end{bmatrix}, \ \vec{F}_c = \begin{bmatrix} \bar{\rho}\tilde{v} \\ \bar{\rho}\tilde{u}\tilde{v} \\ \bar{\rho}\tilde{v}\tilde{v} + \tilde{p} \\ \bar{\rho}\tilde{\omega}\tilde{v} \\ (\bar{\rho}\tilde{e} + \tilde{p})\,\tilde{v} \end{bmatrix}, \ \vec{G}_c = \begin{bmatrix} \bar{\rho}\tilde{\omega} \\ \bar{\rho}\tilde{u}\tilde{\omega} \\ \bar{\rho}\tilde{v}\tilde{\omega} \\ \bar{\rho}\tilde{\omega}\tilde{\omega} + \tilde{p} \\ (\bar{\rho}\tilde{e} + \tilde{p})\,\tilde{\omega} \end{bmatrix}, \tag{2.9}$$

and

$$\vec{E}_v = \begin{bmatrix} 0 \\ \bar{\tau}_{xx} - \overline{\rho u'' u''} \\ \bar{\tau}_{xy} - \overline{\rho u'' \upsilon''} \\ \bar{\tau}_{xz} - \overline{\rho u'' \omega''} \\ Q_x \end{bmatrix}, \vec{F}_v = \begin{bmatrix} 0 \\ \bar{\tau}_{yx} - \overline{\rho \upsilon'' u''} \\ \bar{\tau}_{yy} - \overline{\rho \upsilon'' \upsilon''} \\ \bar{\tau}_{yz} - \overline{\rho \upsilon'' \omega''} \\ Q_y \end{bmatrix}, \vec{G}_v = \begin{bmatrix} 0 \\ \bar{\tau}_{zx} - \overline{\rho \omega'' u''} \\ \bar{\tau}_{zy} - \overline{\rho \omega'' \upsilon''} \\ \bar{\tau}_{zz} - \overline{\rho \omega'' \omega''} \\ Q_z \end{bmatrix}. \tag{2.10}$$

The pressure variable $\tilde{p}$ of equation 2.9 is the force exerted on the fluid flow. The variable τ_{ij}, and $\bar{\rho}\tilde{e}$ are the shear stress of the flow around the body, and the total energy. The variables u, υ, and ω are the Cartesian components of the flow velocity. The expressions for the heat flux and shear stress, as shown in equations 2.11 and 2.12, are in the Cartesian coordinate space, and it is modified with Favre mass-averaged expressions for turbulence motion. These expressions use Einstein summation notations, in order to accommodate the three Cartesian components of shear stress, and heat transfer variables. A systematic method of replacing the ijk with xyz notation can be used in order to obtain each component.

$$\bar{\tau}_{ij} = -\frac{2}{3}\tilde{\mu}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} + \mu\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \tag{2.11}$$

$$Q_i = \tilde{u}_i\left(\bar{\tau}_{ij} - \overline{\rho u'' u''}\right) - \left(\bar{q}_i - C_p\overline{\rho T'' u''}\right) \tag{2.12}$$

The variable $\bar{q}_i$ of equation 2.12 is the mean molecular heat flux defined as $\bar{q}_i = -\frac{\tilde{\mu}}{(\gamma-1)\,\mathrm{Pr}}\frac{\partial a^2}{\partial x_i}$, in which $\tilde{\mu}$ is the turbulent-based viscosity, which, in turn, is determined using Sutherland's formula, and a is the speed of sound, as determined by $a = \sqrt{\gamma R T_\infty}$.The thermodynamic state that closes the system is given by equation 2.13 as

$$\bar{\rho}\tilde{e} = \frac{\tilde{\rho}}{(\gamma - 1)} + \frac{1}{2}\bar{\rho}\left(\tilde{u}^2 + \tilde{v}^2 + \tilde{\omega}^2\right) + k, \tag{2.13}$$

where γ is the ratio of specific heats, and k is the Favre mass-averaged turbulence kinetic energy.

Chapter 3

Governing Structural Equations

In order to make the CFD solver capable of solving the fluid flow around a moving geometry, the governing flow equations need to be coupled with governing structural equations. In the case of a wing with infinitely large span-to-chord ratio, the governing structural equations for two dimensions can be used, including the heave and pitching structural equations. As shown in figure 3.1, the heave and pitching motion of an airfoil is the vertical and rotational motion, respectively. The CFD solver is also capable of simulating three dimensional cases with cross-section deflections [13]. For this kind of problems, the modal equations are used. The present work is focused on two dimensional airfoils which is rigid with no deflection.

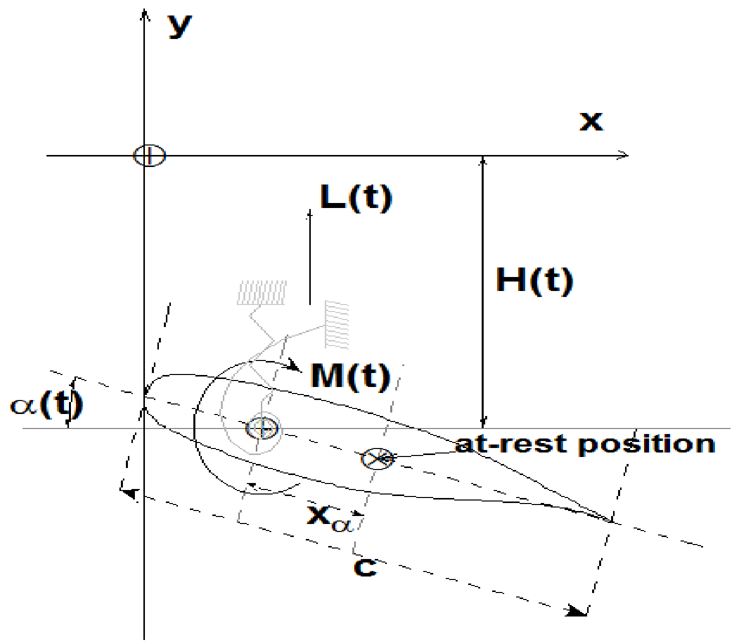

Figure 3.1: Profile Diagram of an Airfoil

3.1 Heave and Pitching Equations

Using Newton's law of force summation, the heave structural equation can be expressed as follows:

$$m\ddot{h} + mcx_\alpha\ddot{\alpha} + \Phi_h\dot{h} + k_h h = L. \tag{3.1}$$

Equation 3.1 can be applied for two dimensional flow problems involving moving structures such as wings with large span-to-chord ratios. The variable m is the mass of the wing, c is the chord length, h is the vertical displacement along the rotating axis, α is the rotational displacement around its rotating axis, Φ_h is the translational damping factor along its rotating axis, k_h is vertical stiffness factor, and L is the lift per unit span. The damping variable Φ_h gives a measure of resistance of the wing against flow-induced heave vibrations. However, the damping value for real case scenarios is so small compared to aerodynamic forces, that it is practically zero in CFD computations. Equation 3.1 is non-dimensionalized using the procedure provided in Appendices D.3 and D.4, resulting in the following:

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + 2\Phi_h^* \frac{\omega_h}{\omega_\alpha} \dot{h}^* + \frac{\omega_h^2}{\omega_\alpha^2} h^* = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_L. \tag{3.2}$$

The Newton's law of moment summation can be used to obtain the pitching structural equation as follow:

$$mcx_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + \Phi_\alpha \dot{\alpha} + k_\alpha \alpha = M. \tag{3.3}$$

The variable I_α is the moment of inertia around its rotating axis. The variable Φ_α is rotational damping factor. k_α is the rotational stiffness factor. Lastly, M is the moment around its rotating axis, as indicated in figure 3.1. Equation 3.3 is non-dimensionalized using the procedure provided in Appendix D.1 and D.2, resulting in the following:

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + 2\Phi_\alpha^* r_\alpha^2 \dot{\alpha}^* + r_\alpha^2 \alpha = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_M. \tag{3.4}$$

The rotational damping variable is also a measure of resistance of the wing against flow-induced pitching vibration. It is also considered to be very small in real case flow problems, that it can be set to zero in CFD computations. Nevertheless, the CFD solver could take both rotational and translational damping values for solving the governing structural equations.

Chapter 4

Discretization of Equations

In order to simulate the fluid flow around a moving object, the CFD solver makes use of the Fully Coupled Solution Methodology to obtain the flowfield solution at every timestep of the simulation run [17] [18]. This fully coupling solution method consists of running pseudo-time steps during every physical time step, in order to obtain a converging solution of the RANS equations. During each pseudo-time step, the CFD solver finds an iterative solution of the structural equations, which is then coupled with the RANS equations so that they can be solved for at the current pseudo time step. If the computed RANS solution does not converge at this pseudo-time step, the solution of structural equations is solved for the next pseudo-time level, and the same process is repeated until a final converging solution of the RANS equation is obtained. The same overall process is then repeated at the next physical time step. This fully coupled solution methodology is made possible by the discretization of both the RANS equations and structural equations, which is the focus of the present chapter.

4.1 Structural Equation Formulation

The two non-dimensional structural equations, as given by equations 3.2 and 3.4, can be combined to form a matrix equation, as follows:

$$[M]\,\frac{\partial \vec{S}}{\partial t} + [K]\,\vec{S} = \vec{q}, \tag{4.1}$$

where:

$$[M] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -x_\alpha \\ 0 & 0 & 1 & 0 \\ 0 & -x_\alpha & 0 & r_\alpha^2 \end{bmatrix}, \quad [K] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ \frac{\omega_h^2}{\omega_\alpha^2} & 2\Phi_h \frac{\omega_h}{\omega_\alpha} & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & r_\alpha^2 & 2\Phi_\alpha r_\alpha^2 \end{bmatrix}, \tag{4.2}$$

and

$$\vec{q} = \begin{bmatrix} 0 \\ \frac{2}{\pi}\frac{U^{*2}}{\mu} C_L \\ 0 \\ \frac{2}{\pi}\frac{U^{*2}}{\mu} C_M \end{bmatrix}, \quad \vec{S} = \begin{bmatrix} h^* \\ \dot{h}^* \\ \alpha^* \\ \dot{\alpha}^* \end{bmatrix}. \tag{4.3}$$

In equation 4.1, the matrices $[M]$ and $[K]$ are the mass matrix, and the stiffness matrix, respectively, and the vector $\vec{S}$ is structural conservative vector. In equations 4.2 and 4.3, x_α is the nondimensional distance between the center of mass and axis of rotation, r_α is the radius of gyration, μ is the mass ratio, ω_α is the pitch frequency, ω_h is the heave frequency, Φ_h is the heave damping factor, Φ_α is the pitch damping factor, C_L is coefficient of lift, and C_M is coefficient of moment. More details about these parameters are available in Appendix D.

Equation 4.1 is discretized using the 3-step backward differencing techniques, thus yielding equation 4.4. The solver then solves the matrix equation 4.4 for $\vec{S}$ using the

Gauss-Seidel Line relaxation iterative techniques involving pseudo-time steps toward a converging solution.

$$\left(\frac{1}{\Delta\tau}[I]+\frac{1.5}{\Delta t}[M]+[K]\right)\delta\vec{S}^{n+1,m+1}=-[M]\frac{3\vec{S}^{n+1,m-1}-4\vec{S}^{n}+\vec{S}^{n-1}}{2\Delta t}-[K]\vec{S}^{n+1,m}+\vec{q}^{n+1,m+1} \tag{4.4}$$

4.2 Flow Equations

In order to discretize the RANS equations, it must be transformed from a nonlinear to linearized form. This can be achieved by imposing the Method of Lines, as described in Reference [15]. The outcome from this method is the vector $\vec{R}$, also known as the residual vector, which contains the viscous and convective vectors, as given by equations 2.7 and 2.8. Then, using the same implicit discretization technique as used for the structural equation, the RANS equation becomes equation 4.5, as follows:

$$\left[\left(\frac{1}{\Delta t_I^*}+\frac{3}{2\Delta t}\left(\Omega\left[M\right]\right)_I^{n+1}+\left(\frac{\partial\vec{R}}{\partial\vec{Q}}\right)\right)\right]\Delta\vec{Q}^*=-\left(\vec{R}_I^*\right)^l, \tag{4.5}$$

where $\vec{R}$ is the residual vector, expressed as:

$$R=-\int_s\left[\left(W_{c,1}-W_{v,1}\right)\mathbf{i}+\left(W_{c,2}-W_{v,2}\right)\mathbf{j}+\left(W_{c,3}-W_{v,3}\right)\mathbf{k}\right]\cdot d\mathbf{s}. \tag{4.6}$$

The term $\frac{\partial\vec{R}}{\partial\vec{Q}}$ of equation 4.5 is known as the flux Jacobian. The CFD solver evaluates the convective flux vectors in equation 4.5 using specialized schemes such as central scheme, flux-Vector Splitting scheme, etc. For this study, the Roe scheme [15] [19] and Zha scheme [17] [20] are used to solve these vectors for CFD computation.

Chapter 5

The CFD solver

The present CFD solver that is used to obtain LCO simulations with the NACA 64a010 and the NLR 7301 wing is available at the CFD lab of the college of engineering of the University of Miami. A complete technical description of this aeroelastic solver can be found in Reference [13]. It is used to simulate airflow around moving or non-moving two dimensional airfoils, generating time-dependent CFD results of the aerodynamic properties of the flowfield. It can also simulate fluid flow around three dimensional moving or non-moving objects such as wings and airplanes. It makes use of Reynolds-averaged Navier-Stokes equations, supplemented with the turbulence model known as the Baldwin-Lomax model, as described in Reference [21]. As mentioned in section 4.2, the fully coupled solution methodology is employed to obtain CFD solution of the flowfield around moving objects at every physical time step. In the present chapter, the overall process of running the CFD solver for solving the flow field of moving structures is described. Also, the solution schemes that are used to solve the convective flux vectors of RANS equations are described.

Two particular schemes are used, including the Roe scheme [15] [19] and the Zha scheme [17] [20].

5.1 CFD Simulation Process

In order to obtain CFD solutions of the flowfield around a moving object, the CFD solver needs to go through a series of CFD simulation pre-runs that require different initial flow solutions, before running the true CFD run that provides the necessary CFD results. First, an initial solution is created to run the program with laminar flow. This flowfield solution, like all subsequent solutions, consists of initial values of the flow conservative variables, as shown in equation 2.6. The final CFD solution that is generated by the first pre- run serves as the initial solution for the second pre-run, which simulates turbulent flow around a non-moving object. The final flowfield solution from the second pre-run is used as the initial solution for the third pre-run. This simulation pre-run generates solution in pseudo-time steps, and it generates a pair of CFD solutions; one for the final physical time step, and the other for the second final physical time step. These two CFD solutions are then used as the initial solutions for the next and true run, which simulates turbulence flows around a flow-induced vibrating airfoil.

During the third simulation pre-run, the program makes use of the fully coupled methodology to obtain a converging solution of the flowfield at every physical time step. During the pseudo-time iteration process, the program basically finds the solutions of the flow equations by first acquiring the solution of the structural equation at the current pseudo-time step. This cycle repeats until both the residuals of the RANS solutions reach machine zero, or if the maximum allowable pseudo time step is reached, whichever comes first.

5.2 Upwind Schemes

As explained in section 4.2, the two solution schemes that can be used to solve the RANS inviscid convective flux vector are the Roe scheme [15] [19] and Zha scheme [17] [20]. These solution schemes are based on the physical properties of Euler equations that define the physical characteristic of flow. Hence, they are referred as upwind schemes. They are different from central schemes, which are based on the idea of averaging out the conservative variables to the left and to the right of control volume, thus not reflecting the actual physical flow characteristics. There are several categories of upwind schemes, including flux-vector splitting, flux-difference splitting, total variation diminishing, and fluctuation-splitting schemes [15].

5.2.1 The Roe Scheme

The governing principle behind the Roe Scheme is to evaluate the convective fluxes at the face of a control volume from the left and right state by solving the Riemann problem. It belongs to the category of flux-difference splitting schemes. It can be used for flow fields that are discretized based on the based on cell-centered scheme or dual control-volume. In the case of the CFD solver, the cell-centered volume approach is used. The Roe scheme can be expressed in general as:

$$\left(\vec{F}_c\right)_R - \left(\vec{F}_c\right)_L = \left(\bar{A}_{Roe}\right)_{I+\frac{1}{2}} \left(\vec{Q}_R - \vec{Q}_L\right). \tag{5.1}$$

In equation 5.1, the matrix variable $\bar{A}_{Roe}$ denotes the so called *Roe matrix*, and the subscripts and L and R represent the left and right state, respectively. The Roe matrix is identical to the convective flux Jacobian, as described in references [15], and [22]. The convective interface flux can be evaluated at the faces of a control volume, as follow [23]:

$$\left(\vec{F}_c\right)_{I+\frac{1}{2}} = \frac{1}{2}\left[\vec{F}_c\left(\vec{W}_R\right) + \vec{F}_c\left(\vec{W}_L\right) - \left|\bar{A}_{Roe}\right|_{I+\frac{1}{2}} \left(\vec{W}_R - \vec{W}_L\right)\right] \tag{5.2}$$

According to References [24] and [13], The Roe matrix $\bar{A}_{Roe}$ is a 6 x 6 matrix and has the form $A = T\Lambda T^{-1}$, where T, T^{-1}, and Λ are the right eigenvector, left eigenvector, and the eigenvalue matrix of A, respectively. By replacing the variables of T, T^{-1}, and Λ with the corresponding Roe-averaged counterparts, the Roe matrix $\bar{A}_{Roe}$ can be obtained. The expression for T, and Λ are given as follow:

$$\Lambda = \begin{pmatrix} U+C & 0 & 0 & 0 & 0 & 0 \\ 0 & U-C & 0 & 0 & 0 & 0 \\ 0 & 0 & U & 0 & 0 & 0 \\ 0 & 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & 0 & U & 0 \\ 0 & 0 & 0 & 0 & 0 & U \end{pmatrix} \tag{5.3}$$

$$T = \begin{pmatrix} \frac{1}{2h} & \frac{1}{2h} & 0 & 0 & -\frac{1}{h} & 0 \\ \frac{u+c\hat{l}_x}{2h} & \frac{u-c\hat{l}_x}{2h} & \hat{m}_x & \hat{n}_x & -\frac{u}{h} & 0 \\ \frac{\upsilon+c\hat{l}_y}{2h} & \frac{\upsilon-c\hat{l}_y}{2h} & \hat{m}_y & \hat{n}_y & -\frac{\upsilon}{h} & 0 \\ \frac{\omega+c\hat{l}_z}{2h} & \frac{\omega-c\hat{l}_z}{2h} & \hat{m}_z & \hat{n}_z & -\frac{\omega}{h} & 0 \\ \frac{c\hat{U}+\gamma e-(\gamma-1)q}{2h} & \frac{-c\hat{U}+\gamma e-(\gamma-1)q}{2h} & \widehat{V} & \widehat{W} & -\frac{q}{h} & 0 \\ \frac{v}{2h} & \frac{v}{2h} & 0 & 0 & -\frac{v}{h} & 0 \end{pmatrix}, \tag{5.4}$$

where the static enthalpy h is calculated as,

$$h = \frac{c^2}{\gamma - 1}, \tag{5.5}$$

and the variable q is the flow kinetic energy, expressed as,

$$q = \frac{1}{2}\left(u^2 + \upsilon^2 + \omega^2\right). \tag{5.6}$$

$\hat{l}$ is the unit vector normal to the ξ surface pointing to the direction that ξ increases, and it can be expressed as:

$$\hat{l} = \frac{\vec{l}}{\left|\vec{l}\right|} \tag{5.7}$$

$\hat{m}$, $\hat{n}$ and $\hat{l}$ are mutually orthogonal unit vectors; that is, $\hat{l}\bullet\,\hat{m} = 0$, $\hat{l} \bullet \hat{n} = 0$, and $\hat{m}\bullet\hat{n} = 0$. Let $\vec{V} = (u, \upsilon, \omega)$ be the velocity vector, $\hat{U}$, $\hat{V}$, and $\hat{W}$ are then determined by,

$$\hat{U} = \vec{V} \cdot \hat{l} \tag{5.8}$$

$$\hat{V} = \vec{V} \cdot \hat{m} \tag{5.9}$$

$$\hat{W} = \vec{V} \cdot \hat{n} \tag{5.10}$$

The Roe scheme that is used in the CFD solver is based on the above formulations, and it can be used for moving cases.

5.2.2 The Zha Scheme

The Zha scheme is based on the concept of Convective Upwind Split Pressure (CUSP) scheme, as suggested in References [25], [26], [27], and [28]. Basically, the governing principle behind CUSP schemes is to decompose the vector of the convective fluxes into two parts, including a convective and a pressure part. The CUSP schemes can be categorized into two types, including H-CUSP and E-CUSP schemes. The Zha scheme belongs to the E-CUSP category. The main feature about E-CUSP schemes is that the total energy is place in the convective vector, whereas the H-CUSP schemes, as well as other upwind schemes, have the total enthalpy from the energy equation in the convective vectors. The E-CUSP scheme developed by Zha has the advantages of low diffusion and efficient calculations using a scalar dissipation term. The general expression for CUSP schemes is as follows:

$$\left(\vec{F}_c\right)_{I+\frac{1}{2}} = \frac{1}{2}\left[\vec{F}_c\left(\vec{Q}_R\right) + \vec{F}_c\left(\vec{Q}_L\right)\right] - \vec{D}_{I+\frac{1}{2}}. \tag{5.11}$$

where $\left(\vec{F}_c\right)_{I+\frac{1}{2}}$ is the interface flux, and $\vec{D}_{I+\frac{1}{2}}$ is the dissipation term. The general expression of the interface flux that is evaluated by the Zha scheme is as follow:

$$\vec{F}_{I+\frac{1}{2}} = \frac{1}{2}\left[(\rho u)_{\frac{1}{2}}\left(\mathbf{q}_L^c + \mathbf{q}_R^c\right) - |\rho u|_{\frac{1}{2}}\left(\mathbf{q}_L^c - \mathbf{q}_R^c\right)\right] + \begin{pmatrix} 0 \\ \text{Þ}^+ p \\ \frac{1}{2}p\left(u + a_{\frac{1}{2}}\right) \end{pmatrix}_L + \begin{pmatrix} 0 \\ \text{Þ}^- p \\ \frac{1}{2}p\left(u - a_{\frac{1}{2}}\right) \end{pmatrix}_R \tag{5.12}$$

Where, the interface mass flux is evaluated as:

$$(\rho u)_{\frac{1}{2}} = \left(\rho_L u_L^+ + \rho_R u_R^+\right) \tag{5.13}$$

$$\mathbf{q}^c = \begin{pmatrix} 1 \\ u \\ e \end{pmatrix} \tag{5.14}$$

$$u_L^+ = a_{\frac{1}{2}} \left\{ \frac{M_L + |M_L|}{2} + \alpha_L \left[\frac{1}{4} (M_L + 1)^2 - \frac{M_L + |M_L|}{2} \right] \right\} \tag{5.15}$$

$$u_R^+ = a_{\frac{1}{2}} \left\{ \frac{M_R - |M_R|}{2} + \alpha_R \left[\frac{1}{4} (M_R - 1)^2 - \frac{M_R - |M_R|}{2} \right] \right\} \tag{5.16}$$

The variables α_L and α_R are evaluated as:

$$a_L = \frac{2\,(p/\rho)_L}{(p/\rho)_L + (p/\rho)_R}, \qquad a_R = \frac{2\,(p/\rho)_R}{(p/\rho)_L + (p/\rho)_R} \tag{5.17}$$

The interface speed of sound $a_{\frac{1}{2}}$, and Mach number are evaluated as:

$$a_{\frac{1}{2}} = \frac{1}{2}\,(a_L + a_R) \tag{5.18}$$

$$M_L = \frac{u_L}{a_{\frac{1}{2}}}, \qquad M_R = \frac{u_R}{a_{\frac{1}{2}}} \tag{5.19}$$

The pressure splitting coefficient is:

$$Þ^{\pm} = \frac{1}{4}\left(M \pm 1\right)^2\left(2 \mp M\right) \pm \alpha M\left(M^2 - 1\right)^2, \qquad \alpha = \frac{3}{16} \tag{5.20}$$

For $u > a$, $\vec{F}_{\frac{1}{2}} = \vec{F}_L$; and for $u < -a$, $\vec{F}_{\frac{1}{2}} = \vec{F}_R$.

More details about the parameters in equation 5.12 can be found in References [20], [29], and [30]. The CFD solver can be run to solve the turbulent flow around a moving geometry by using the Zha scheme.

Chapter 6

CFD Mesh Generation

In order to perform the CFD simulation of the flowfield around a NACA 64a010 and NLR 7301 wing geometry, a mesh coordinate file linked must be generated first. To begin with, the mesh coordinate file can be generated by running a Fortran code developed by Chen [17]. The mesh coordinate file obtained from Chen's Fortran code is then used to generate the initial flowfield solution which has the initial flow properties at each coordinate point. The most complicated phase in performing a CFD computation is the generation of the mesh coordinate file. In the following sections, the specific technique used to generate the mesh coordinate files of NACA 64a010, NLR 7301, as well as a cylinder, is described. Also, the mesh coordinate system and boundary conditions are described.

6.1 Mesh Generation Mathematics

The method that is used to generate the mesh coordinates around the object of interest is known as the algebraic grid generation method. Before implementing this method, the topology of the mesh needs to be defined. For the present study, the O-type topology is used, and the boundary conditions of this topology are schematically depicted in figure 6.1.This particular mesh of O-type topology consists of one block, which is the terminology used to name a topology region. The entire mesh itself consists of two zones, including the fine and coarse mesh zones. This form of mesh structure allows for the CFD solver to reposition the coordinate point of the mesh files after every physical time step more effectively.

The mesh points along the surface of the object, or boundary I, are calculated using a formula based on clustered geometry grid generation techniques. The mesh points of the outer wall, or boundaries II and V, are equally distributed along their entire lengths. The mesh points for boundaries III and IV are calculated using the same clustering technique as used for boundary I. The boundary points that surround the fine mesh zone serve as boundary conditions to solve for the fine mesh coordinates using elliptic partial differential equations (PDE) for grid generation [16].

The computational finite difference analog of PDE in x coordinates is as follows:

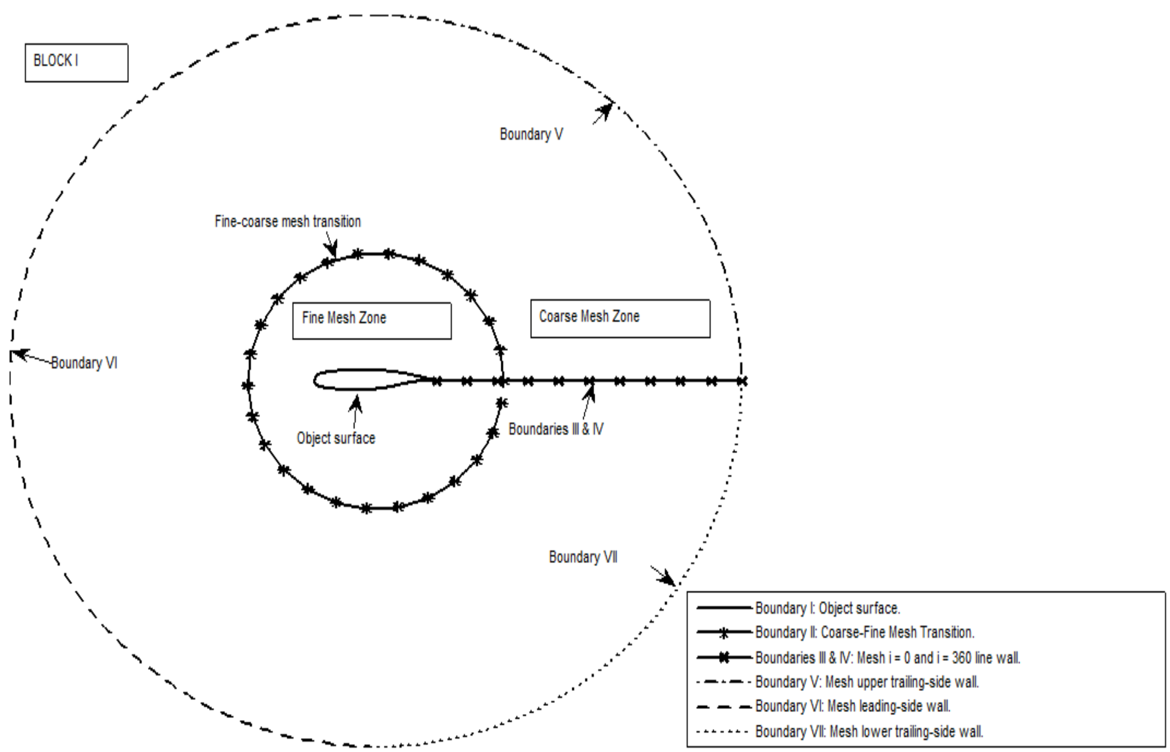

Figure 6.1: Mesh boundary descriptions using O-type topology.

$$
\begin{gathered}
\omega\left(cx_{i,j+1}^{k+1}\right)-2\left(a+c\right)x_{i,j}^{k+1}+\omega\left(cx_{i,j-1}^{k+1}\right)= \\
-2\left(1-\omega\right)\left[\tfrac{a}{(\Delta\zeta)^2}+\tfrac{c}{(\Delta\eta)^2}\right]x_{i,j}^{k}+ \\
\tfrac{\omega b}{2}\left(x_{i+1,j+1}^{b}-x_{i+1,j-1}^{k}+x_{i-1,j-1}^{k+1}+x_{i-1,j+1}^{k+1}\right)- \\
\omega a\left(x_{i+1,j}^{k}+x_{i-1,j}^{k+1}\right)
\end{gathered} \tag{6.1}
$$

The PDE for y-coordinates is the same as equation 6.1, only that the variable x is replaced with y. These equations are also known as the Successive Overrelaxation Line Gauss-Seidel form of Poisson's equations. As indicated in equation 6.1, the computational coordinates i, j, and k provide a means to identify the physical coordinate system of the mesh field. Basically, the i coordinate axis starts from the trailing edge of the object surface, wraps counterclockwise around boundary I, and ends at the same trailing edge. The j-coordinate axis starts from the surface of the object, continues outward from the surface of the object, and ends at the wall of the block. The PDEs are solved using an iterative convergence technique known as Thomas algorithm [19].

The formula that is used to generate the clustered geometry grid of the coarse mesh region, as shown in figure 6.1, is as follows:

$$
y_{i,j}=H\frac{(\beta-1)\left[(\beta+1)/(\beta-1)\right]^{1-j}}{\left[(\beta+1)/(\beta-1)\right]j+1},\quad j=1,J_{\max} \tag{6.2}
$$

Equation 6.2 is based on algebraic grid generation techniques [16]. The variable H is the physical length of boundary III (boundaries III and IV have equal lengths). The variable β is the cluster parameter whose value can be anything between 1 and

∞. The closer the variable β is to 1, the greater the number points become clustered toward $j = 1$. More technical details about algebraic mesh generation techniques can be found in References [31] and [32].

Chapter 7

Results

The goal of this thesis is to perform CFD simulations of flow-induced vibrations of NACA 64a010 and NLR 7301 in order to capture LCO behaviors using both the Zha and Roe scheme. After several CFD simulation trials, LCO and flutter behaviors have been successfully captures using both schemes. The results given in the present chapter are obtained using the Roe scheme. The CFD results associated with these flow behaviors are post-processed in order to obtain the pressure coefficient distribution and the skin friction coefficient distribution. These plots help demonstrate the physics behind limit cycle oscillations, flutter, and damping vibrations. In the following sections, the flow and structural parameters that are used to run the CFD simulations of each airfoil are provided in tables 7.1, 7.2, 7.3. In these tables, the mesh file that is used for cases involving the NACA 64a010 and NLR 7301 airfoil have a grid size of 280x65. Also, mesh file that is used for cases involving a cylinder has a mesh grid size of 120x80.

7.1 Method of CFD Computation

The methods used to run the CFD solver consists of certain simulation pre-runs and the final run, depending on the type of problem, and thus, requiring different input parameters. The formulas that are used to obtain the flow and structural parameters for the CFD solver can be found in Appendix E. In this thesis study, three different categories of CFD computation are performed, including turbulent flow around a stationary geometry, turbulent flow around force-induced vibrating geometry, and turbulent/laminar flow around a flow-induced vibrating geometry. The general process for running the CFD for each of these types of problem has been described in section 5.1.

In order to run all three types of CFD simulation, it is necessary to obtain the flow parameters, including the Reynolds Number Re, Mach number Ma, dimensionless stagnation pressure P_o, dimensionless static pressure P^*, dimensional total temperature T_o, and the Specific heat ratio γ. These five flow parameters must be calculated using the formulas provided in Appendix E. If the flow problem belongs to the category of flow-induced or force-induced vibrating objects, certain structural parameters must be obtained. For the case of force-induced vibrating objects, the reduced pitch frequency ω_α^* is require to simulate the pitching motion as a function of time as $\alpha(t) = \alpha_m + \alpha_o \sin(\omega_\alpha^* t)$, where α_m and α_o are the mean angle of attack and the amplitude of oscillation, respectively. For the case of a flow-induced vibrating object, several structural parameters must be acquired or calculated from given flow parameters. The necessary structural parameters are the dimensional velocity U_∞, mass ratio μ, dimensional pitch frequency ω_α, ratio between pitch and heave frequency $\frac{\omega_\alpha}{\omega_h}$, moment arm length a, initial angle of attack α_o, unbalance distance x_a, and the squared radius of gyration r_a^2. Out of these structural parameters, the dimensional velocity has to be calculated from equation E.6 in Appendix E.

7.2 Validation Cases

The CFD solver and the mesh files of NACA 64a010 and NLR 7301 are tested by validating certain simulation runs with experimental and computational data. Table 7.1 provides the initial input parameters for three validity cases.

Structural and Flow Parameters	**Case I**	**Case II**	**Case III**
Reynolds Number: Re $(-)$	500	12560000	1700000
Mach Number: M $(-)$	0.2	0.8	0.753
Freestream temperature: T (K)	–	–	498.6
Static Pressure: P (Pa)	–	–	4.418
Specific heat ratio: γ $(-)$	1.4	1.4	1.4
Dimensionless static pressure: P^* $(-)$	17.85714	1.116071	1.259743
Total Pressure: P_o $(-)$	18.36216	1.701272	1.834694
Total Temperature: T_o $(-)$	1.008	1.128	1.113402
Viscosity: ν (x10^{-5})	–	1.74	1.74
Velocity: U_∞ $(\frac{\mathrm{m}}{\mathrm{s}})$	–	315.678	–
Velocity Index: VI $(-)$	–	1.278	–
Mass Ratio: μ $(-)$	1.2732	60	–
Reduced Velocity: U^* $(-)$	1.59155	9.899345	–
Pitch Frequency: ω_α $(1/s)$	0.046940	–	–
Reduced Pitch Frequency: ω_α^* $(-)$	–	0.202	–
Frequency Ratio: $\frac{\omega_\alpha}{\omega_h}$ $(-)$	–	1	–
Heave Damping Factor: Φ_h $(-)$	–	0	–
Pitch Damping Factor: Φ_α $(-)$	–	0	–
Damping ratio: $(-)$	0.633257	–	–
Number of Cycles: NC $(-)$	–	–	–
Physical Time Step: t_s $(-)$	0.05	0.3	0.3
Chord Length: c $(-)$	1.0	1.0	1.0
Initial Angle of Attack: α_o (deg)	–	0.0	0.08
Moment arm length: a $(-)$	0.0	0.0	–
Unbalance distance: x_a $(-)$	–	–	–
Radius of gyration: r_a^2 $(-)$	–	–	–
Case I: Cylinder, vortex indcued oscillating motion			
Case II: NACA 64a010, force induced oscillating motion			
Case III: NLR 7301, steady state (non-moving) condition			

Table 7.1: Flow and structural parameters for CFD simulation runs to obtain results for test and validation purposes

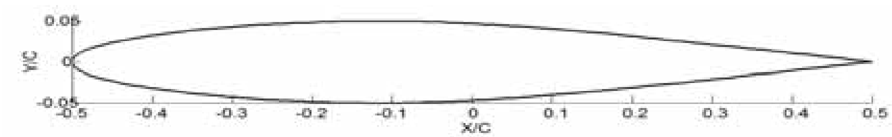

Figure 7.1: NACA 64a010 geometric profile

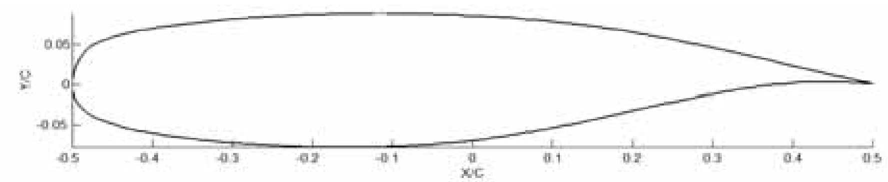

Figure 7.2: NLR 7301 geometric profile

7.3 LCO and Flutter Cases: NACA 64a010 and NLR 7301

One of the airfoils that are used to obtain CFD simulations of LCO and flutter is the NACA 64a010 conventional airfoil. This airfoil is symmetric about its chord line, as shown in figure 7.1. The coordinates of the airfoil shape can be obtained from Reference [33]. The initial structural and fluid parameters that is used for the damping, flutter, critical flutter, and LCO simulations are given in table 7.2.

The other airfoil that is used is the NLR 7301 supercritical airfoil. This airfoil is not symmetrical with respect to its chord line, as shown in figure 7.2. The coordinates of the airfoil can be obtained from Reference [34]. Like for the conventional airfoil, the structural and fluid parameter that is used to obtain flutter and LCO plots are given in table 7.3.

The results of the CFD simulations, using the Roe upwind scheme, are provided for both the NACA 64a010 and NLR 7301 airfoils in the following sections. The input parameters that are used to simulate these airfoil vibrations are provided in tables 7.2 and 7.3.

Structural and Flow Parameters	**Case IV**	**Case V**	**Case VI**	**Case VII**
Reynolds Number: Re $(-)$	12560000	12560000	12560000	12560000
Mach Number: M $(-)$	0.825	0.825	0.825	0.925
Freestream temperature: T (K)	277.8	277.8	277.8	277.8
Static Pressure: P (Pa)	55157.2	55157.2	55157.2	55157.2
Specific heat ratio: γ $(-)$	1.4	1.4	1.4	1.4
Dimensionless static pressure: P^* $(-)$	0.834811	1.049455	1.049455	1.049455
Total Pressure: P_o $(-)$	1.451108	1.640421	1.640421	1.640421
Total Temperature: T_o $(-)$	1.171125	1.136125	1.136125	1.136125
Viscosity: ν (x10^{-5})	1.73891	1.73891	1.73891	1.73891
Velocity: U_∞ $(\frac{\mathrm{m}}{\mathrm{s}})$	315.6782	315.6782	315.6782	315.6782
Velocity Index: VI $(-)$	0.55	0.615	0.7	5.5
Mass Ratio: μ $(-)$	60	60	60	60
Reduced Velocity: U^* $(-)$	4.26029	4.763769	4.763769	42.6028
Pitch Frequency: ω_α $(1/s)$	296.392	265.066	232.879	29.6392
Reduced Pitch Frequency: ω_α^* $(-)$	0.46945	0.419836	0.368856	0.46945
Frequency Ratio: $\frac{\omega_\alpha}{\omega_h}$ $(-)$	1	1	1	1
Heave Damping Factor: Φ_h $(-)$	0	0	0	0
Pitch Damping Factor: Φ_α $(-)$	0	0	0	0
Number of Cycles: NC $(-)$	2	2	2	12
Physical Time Step: t_s $(-)$	0.3	0.3	0.3	0.3
Chord Length: c $(-)$	1.0	1.0	1.0	1.0
Initial Angle of Attack: α_o (deg)	0.0	0.0	0.0	0.0
Moment arm length: a $(-)$	-2.0	-2.0	-2.0	-2.0
Unbalance distance: x_a $(-)$	1.8	1.8	1.8	1.8
Radius of gyration: r_a^2 $(-)$	3.48	3.48	3.48	3.48
Case IV: NACA 64a010, damped oscillating condition				
Case V: NACA 64a010, Critical flutter oscillating condition				
Case VI: NACA 64a010, Flutter oscillating condition				
Case VII: NACA 64a010, LCO condition				

Table 7.2: Flow and structural parameters for CFD simulations of NACA64a010

Structural and Flow Parameters	Case VIII	Case IX
Reynolds Number: Re $(-)$	1695000	1695000
Mach Number: M $(-)$	0.77	0.77
Freestream temperature: T (K)	310	310
Static Pressure: P (Pa)	45000	45000
Specific heat ratio: γ $(-)$	1.4	1.4
Dimensionless static pressure: P^* $(-)$	1.204732	1.204732
Total Pressure: P_o $(-)$	1.78330	1.78330
Total Temperature: T_o $(-)$	1.11858	1.11858
Viscosity: ν (x10^{-5})	1.74	1.74
Velocity: U_∞ $(\frac{\text{m}}{\text{s}})$	257.2398	257.2398
Velocity Index: VI $(-)$	.190	.190
Mass Ratio: μ $(-)$	1077.2	1077.2
Reduced Velocity: U^* $(-)$	0.31989	0.31989
Pitch Frequency: ω_α $(1/s)$	274.3014	274.3014
Reduced Pitch Frequency: ω^*_α $(-)$	$-$	$-$
Frequency Ratio: $\frac{\omega_\alpha}{\omega_h}$ $(-)$	0.761	0.761
Heave Damping Factor: Φ_h $(-)$	$-$	$-$
Pitch Damping Factor: Φ_α $(-)$	$-$	$-$
Damping ratio: $(-)$	$-$	$-$
Number of Cycles: NC $(-)$	$-$	$-$
Non-D. Physical Time Step: t_s $(-)$	0.3	0.3
Chord Length: c $(-)$	1.0	1.0
Initial Angle of Attack: α_o (deg)	0.65	0.0
Moment arm length: a $(-)$	-0.25	-0.25
Unbalance distance: x_a $(-)$	0.086	0.086
Radius of gyration: r_a^2 $(-)$	0.155236	0.155236
Case VIII: NLR 7301, flutter oscillating condition		
Case IX: NLR 7301, LCO oscillating condition		

Table 7.3: Flow and Structural parameters for CFD simulations of NLR 7301

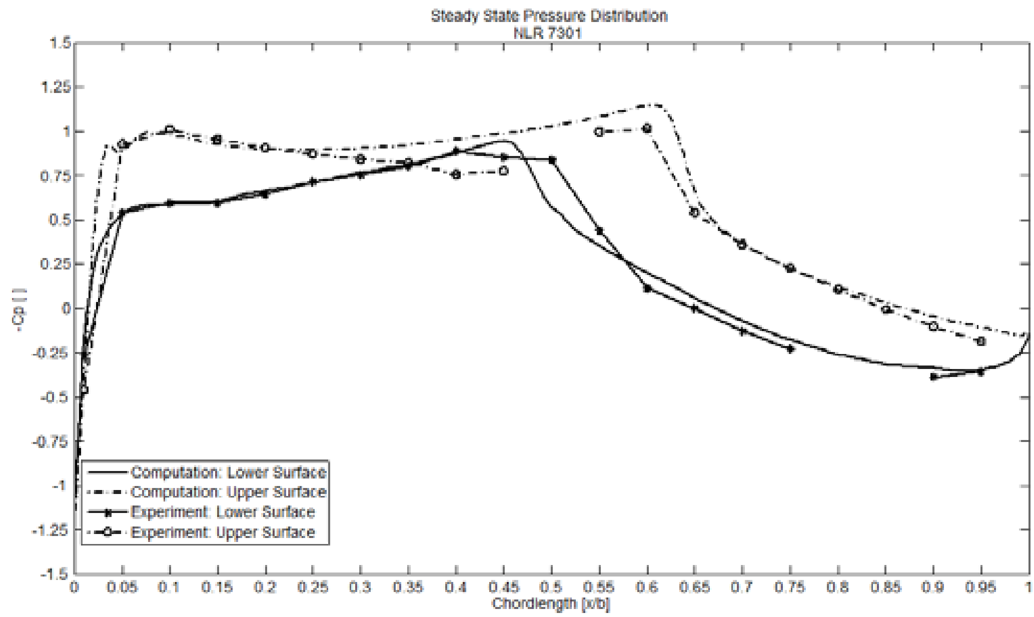

Figure 7.3: Pressure distribution of NLR 7301 for steady state condition (Case III), $M_a = 0.78$, Re= 1700000.0, experimental data obtained from Reference [9]

7.4 Results I: Validation Cases

Figures 7.3 to 7.8 show the pressure distribution, lift and drag coefficients, and displacement profiles that are obtained for different cases of flow and structural conditions, as explained in table 7.1. Figures 7.7 and 7.8 show the computed moment coefficient and lift coefficient, respectively, for a NACA 64a010 in force vibration, with experimental data for comparison [33]. The computed moment coefficient does not agree as accurately with experimental data as it does for the lift coefficient, although these results are similar to those provided in References [4] and [35]. Figures 7.5 and 7.6 are CFD results for a cylinder using a geometry conservation law (GCL) condition of 1 [29]. Lastly, figure 7.9 shows the time evolution of the Mach contour plots for NACA 64a010 in critical flutter condition. The Mach contour plots are shown for every dimensionless timestep of 3.

As shown in the figures 7.3 to 7.8, the CFD solver is validated from CFD results

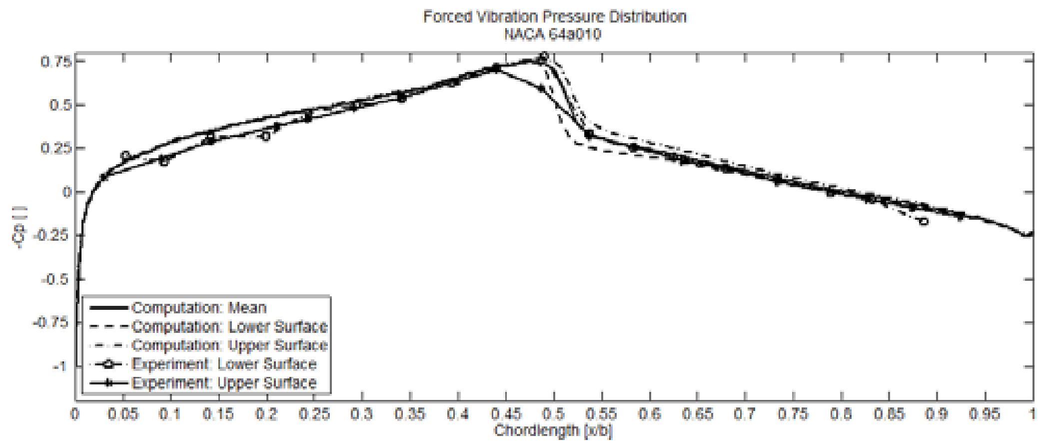

Figure 7.4: Pressure Distribution of NACA 64a010 for force-induced vibrating condition (Case II), $M_a = 0.8$, $Re = 12560000.0$, Experimental data obtained from Reference [33]

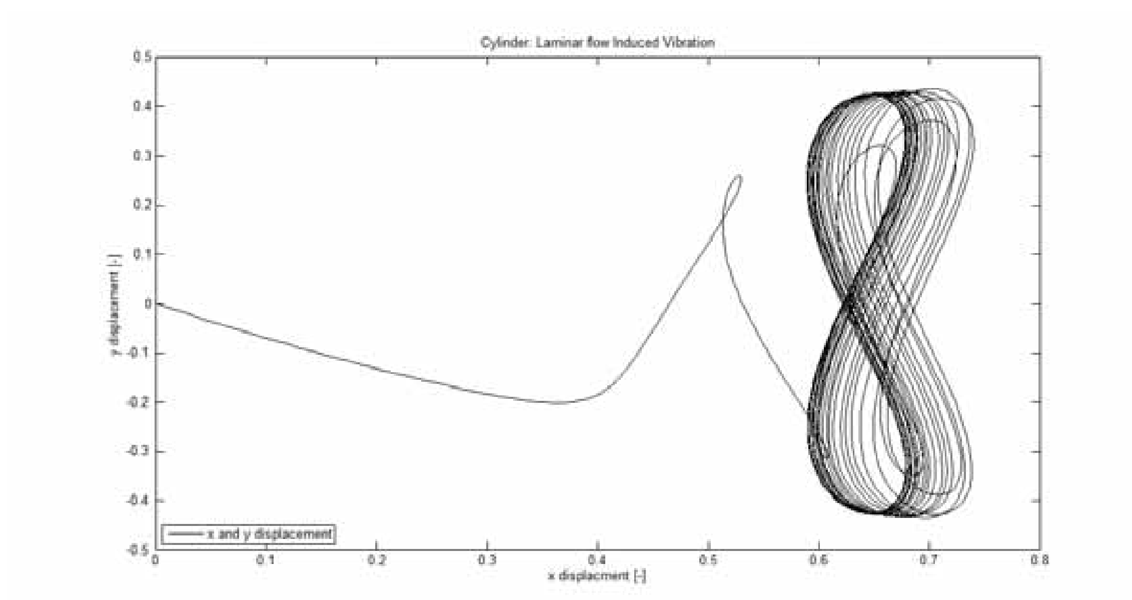

Figure 7.5: Displacement trajectory of a cylinder in laminar vortex induced oscillating condition (Case I), GCL $= 0.0$ $\zeta = 0.63326$, $\mu_s = 1.2732$, $\bar{u} = 1.5915$

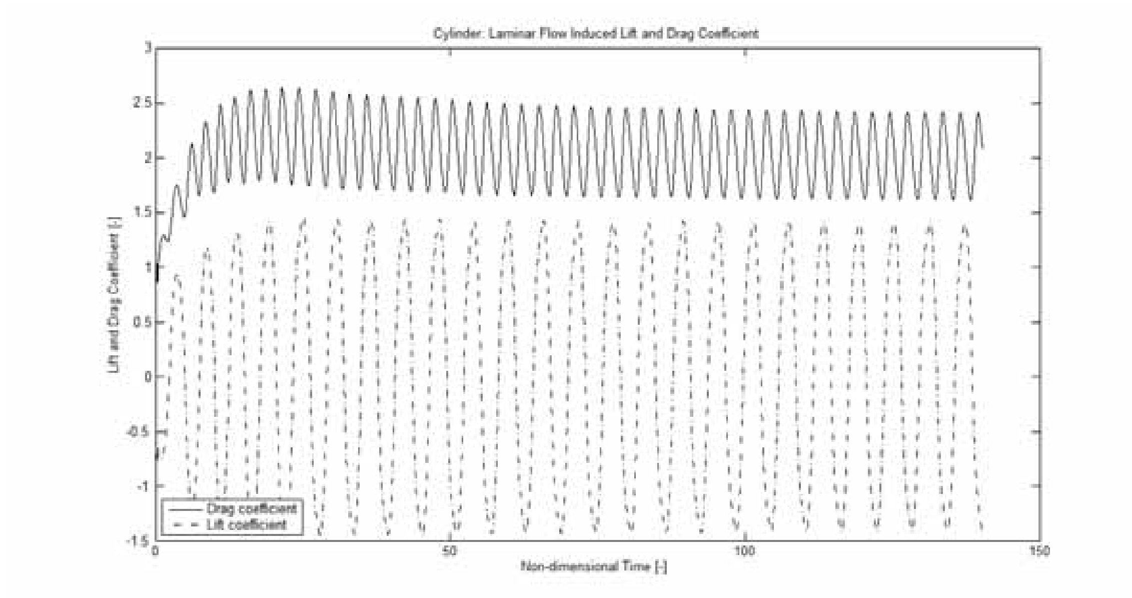

Figure 7.6: Drag and lift coefficient profile for a cylinder in laminar vortex induced condition (Case I), GCL = 0, $\zeta = 0.63326$, $\mu_s = 1.2732$, $\bar{u} = 1.5915$

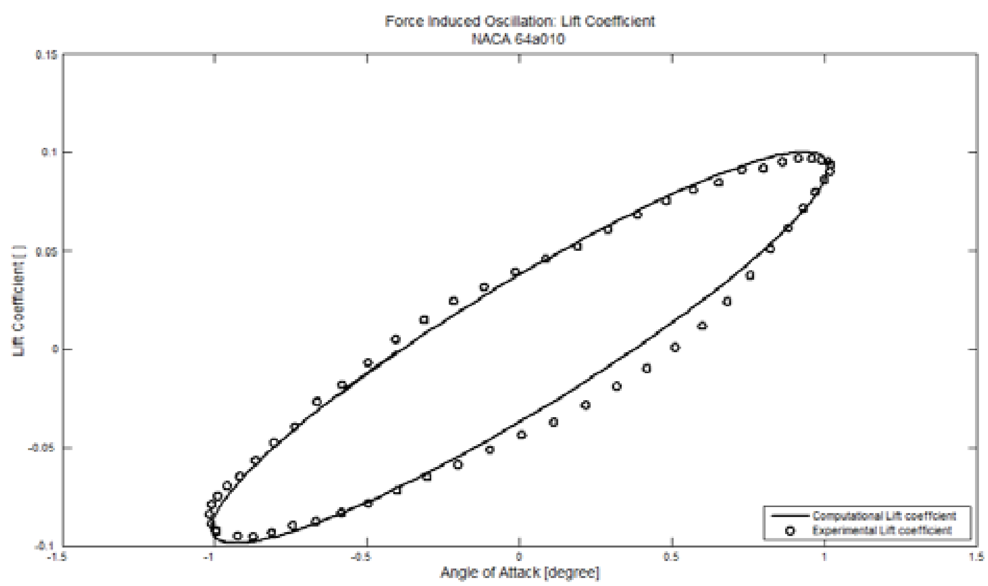

Figure 7.7: Lift coefficient profile of NACA 64a010 for force induced vibrating condition (Case II), at $\alpha_o = 0.0$, $\alpha_A = 1.01$, $K_c = 0.202$, Re = 12560000.0, and $\mathrm{M}_a = 0.8$

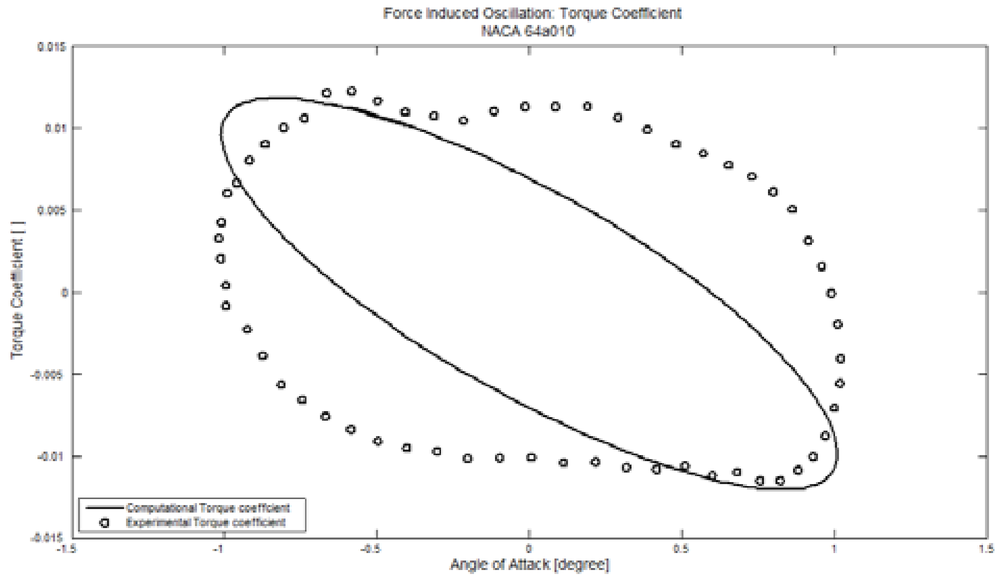

Figure 7.8: Torque coefficient profile of NACA 64a010 for force induced vibrating condition (Case II), at $\alpha_o = 0.0$, $\alpha_A = 1.01$, $K_c = 0.202$, Re $= 12560000.0$, and $\mathrm{M}_a = 0.8$

related to steady-state and unsteady-state conditions. The flow and structural parameters corresponding to these results are shown in table 7.3. As shown in figure 7.3, the pressure distribution for NLR 7301 in steady-state condition closely matches with experimental data obtained from Dietz [9]. As shown in figure 7.4, the pressure distribution for NACA 64a010 in forced pitching oscillating condition also matches with experimental data obtained from Davis [33]. The time evolution of the center displacement, lift and drag coefficients of a cylinder in vortex induced oscillating condition, as shown in figures 7.5 and 7.6, are relatively similar to results obtained by Prananta and Bohbot [3], [4], and others. The time evolution of the lift and torque coefficients of NACA 64a010 in force induced oscillating condition, as shown in figures 7.7 and 7.8, coincide with experimental data obtained from Davis [33], but the computed moment coefficient is not as accurate as those for the lift coefficient. Lastly, the description of critical flutter conditions in term of oscillatory motion of shock waves around the structure, as provided in Reference [5], can be confirmed from the

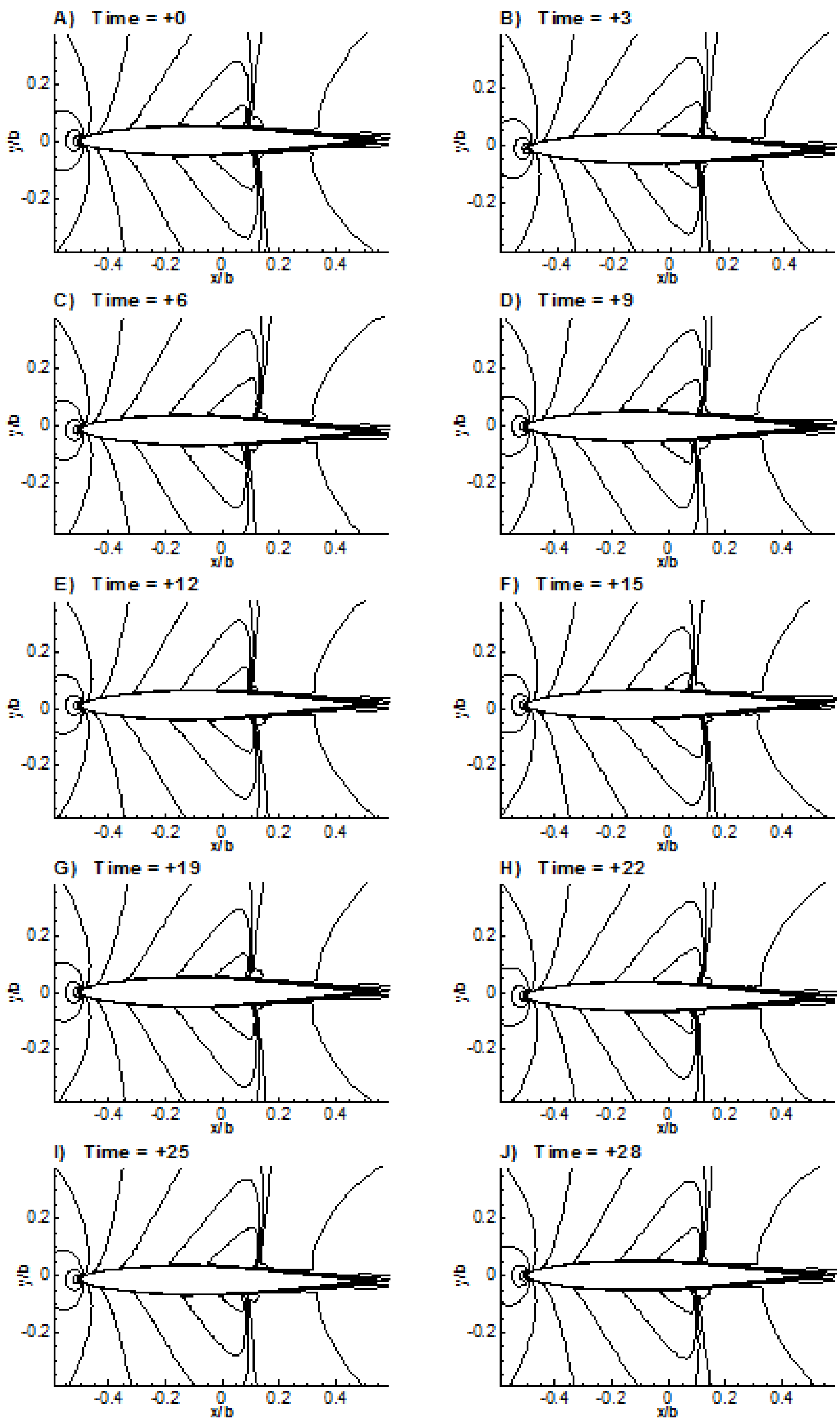

Figure 7.9: Subfigures A to J show the time history of Mach contour plots for NACA 64a010 in critical flutter condition (Case V) in order to capture oscillatory motion of shock waves

Mach contour plots given in figure 7.9. All these CFD results validate the mesh files used for the CFD simulations of NACA 64a010 and NLR 7301, and they show the consistency of the Baldwin Lomax turbulence model that is used by the CFD solver in computing flow properties.

7.5 Results II: LCO and Flutter Cases

Figures 7.10 to 7.17 show the heave and pitching motions of NACA 61a0101 for the damping, critical stable, flutter, and LCO conditions. Figures 7.21, 7.22, 7.18 and 7.22 show the heave and pitching motions of NLR 7301 for flutter and LCO conditions. These results are obtained using input structural parameters using the method of calculations given in Appendix E. Figures 7.20 and 7.23 demonstrate the phase diagram for the LCO oscillations of NLR 7301, as corresponding to figures 7.21 and 7.22. Tables 7.4 and 7.5 provide experimental and computational comparison with the LCO and flutter simulation results.

The time history of the heave and pitching motion of NACA 64a010 for damped, critical flutter, and flutter conditions, as shown in figures 7.10 to 7.15, are consistent with those given in References [36] and [37]. These oscillation plots are obtained by varying the reduced pitch frequency. There exists a critical pitch frequency at which the heave and pitching motions remains constant in amplitude, as shown in figures 7.10 and 7.11. This condition is the called the critical flutter effect [2]. As for the flutter and LCO plots obtained for NLR 7301, as shown in figures 7.18, 7.19, 7.21, and 7.22, they are obtained by varying the initial angle of attack, and keeping the relevant flow properties like Mach number and Reynolds number the same; that is, these flow parameters do not change for cases of different angle of attack.

The limit cycle oscillation (LCO) for both NLR 7301 and NACA 64a010 are shown in figures 7.16, 7.17, 7.21, and 7.22. The appearance of these LCO plots are consistent with the computational and experimental results in References [11], [17], [9], [10], [12], etc. The physics behind the flow that causes flutter and LCO behavior can be understood in terms of the skin friction and pressure coefficients obtained for both airfoils.

As indicated in table 7.4, the flutter properties that are obtained for case VIII (NLR 7301 in flutter oscillating condition) are consistent with experimental data obtained from Reference [9]. However, as indicated in 7.4, the heave and pitching amplitudes for case IX (NLR 7301 in LCO oscillating condition) are approximately 2 times larger than those for case VIII. This observation serves as a description of the difference between LCO and flutter oscillating conditions for NLR 7301. Also, as indicated in table 7.5, the critical flutter and LCO properties corresponding to cases V and VII are consistent with computational data obtained from Reference [4]. It can also be observed from table 7.5 that the heave and pitching amplitudes for case VII (NACA 64a010 LCO oscillating condition) is approximately 9 and 6 times larger than those for case VI (NACA 64a010 in critical flutter oscillating condition). This observation may also serve as a description of LCO and flutter oscillating conditions for NACA 64a010. Both of these observations provide a general description that LCO amplitudes are relatively larger than flutter amplitudes.

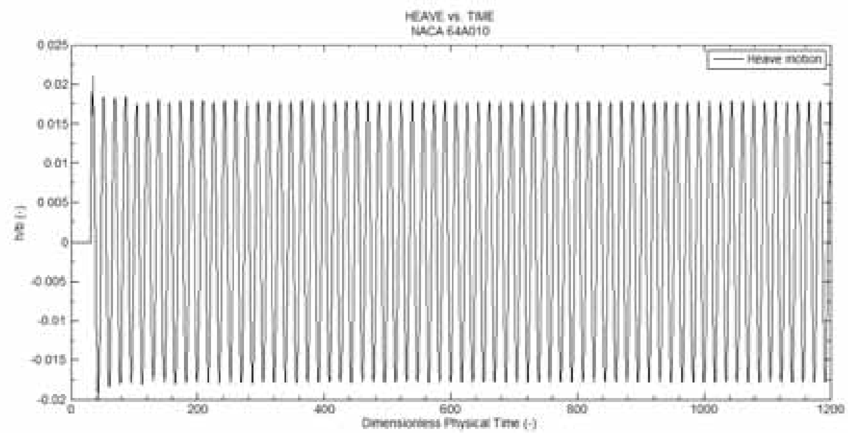

Figure 7.10: Heave oscillation of NACA 64a010 for critical flutter condition (Case V), $\alpha_o = 0.0$, Re $= 12560000.0$, $\mathrm{M}_a = 0.825$

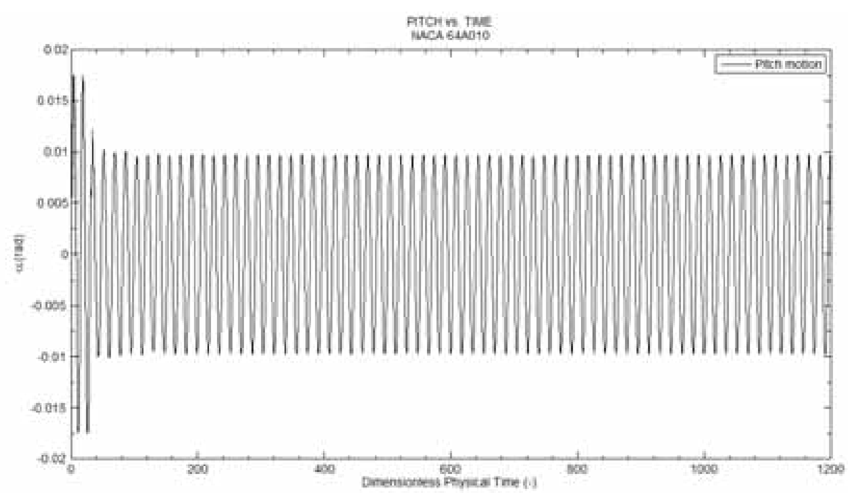

Figure 7.11: Pitch oscillation of NACA 64a010 for critical flutter condition (Case V), $\alpha_o = 0.0$, Re $= 12560000.0$, $\mathrm{M}_a = 0.825$

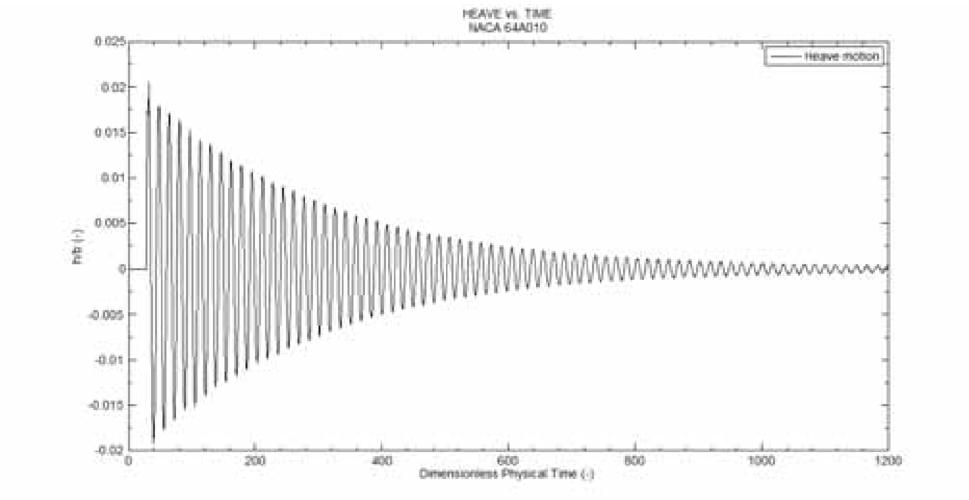

Figure 7.12: Heave oscillation of NACA 64a010 for damping condition (Case IV), $\alpha_o = 0.0$, Re $= 12560000.0$, $\mathrm{M}_a = 0.825$.

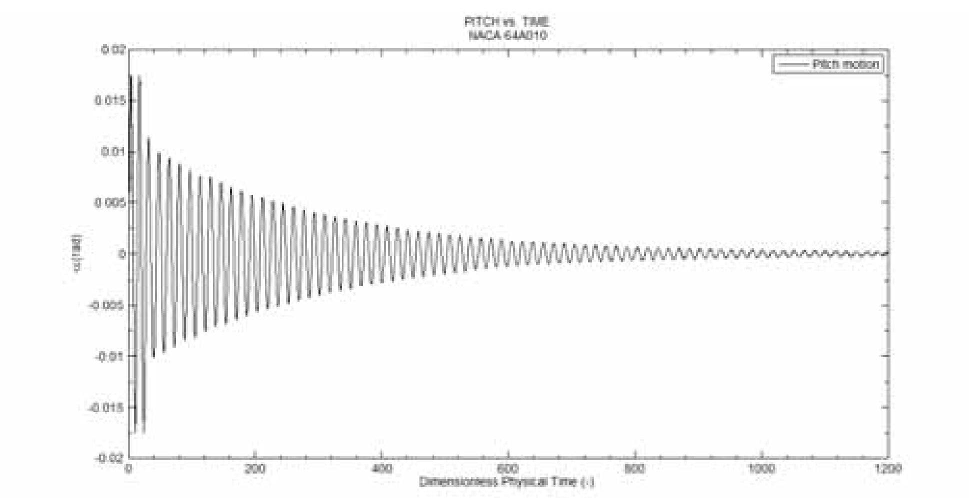

Figure 7.13: Pitch oscillation of NACA 64a010 for damping condition (Case IV), $\alpha_o = 0.0$, Re $= 12560000.0$, $\mathrm{M}_a = 0.825$.

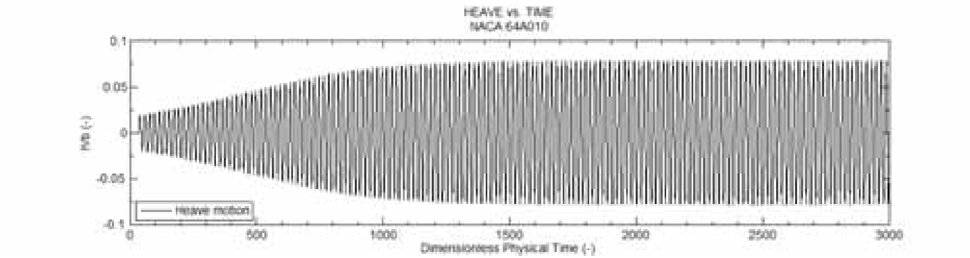

Figure 7.14: Heave oscillation of NACA 64a010 for flutter condition (Case VI), $\alpha_o =$ 0.0, Re = 12560000.0, $M_a = 0.825$.

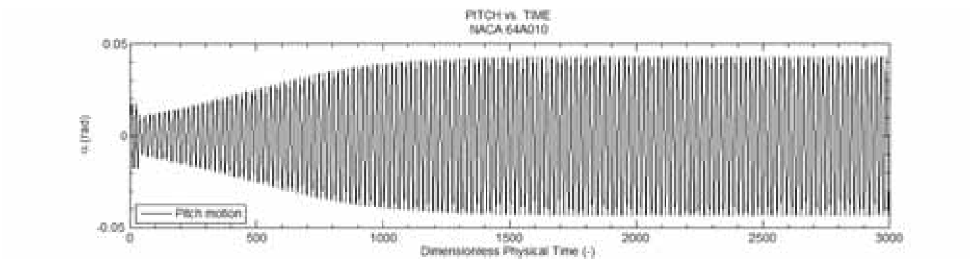

Figure 7.15: Pitch oscillation of NACA 64a010 for flutter condition (Case VI), $\alpha_o =$ 0.0, Re = 12560000.0, $M_a = 0.825$.

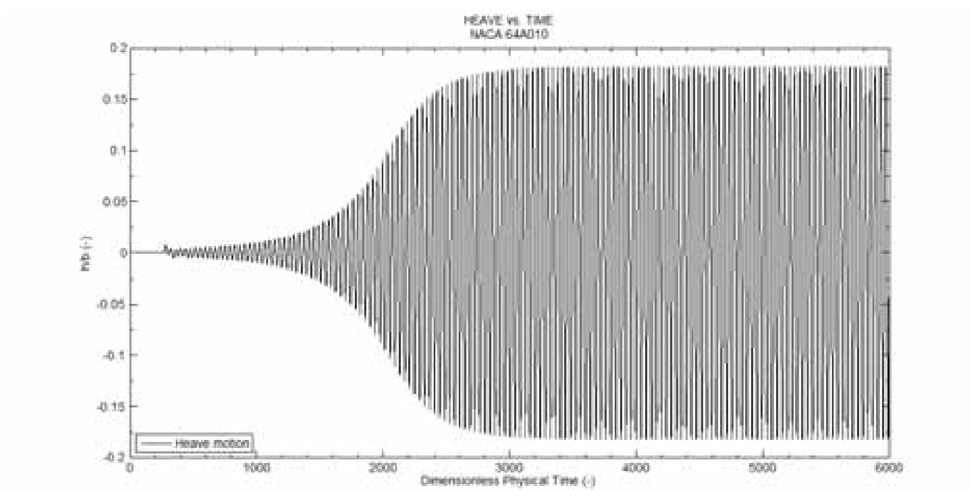

Figure 7.16: Heave oscillation of of NACA 64a010 for LCO condition (Case VII), $\alpha_o = 0.0$, Re = 12560000.0, $M_a = 0.925$

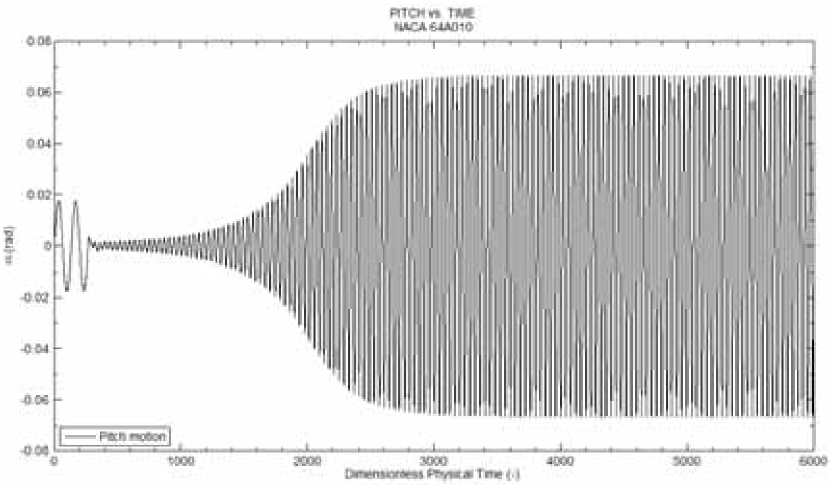

Figure 7.17: Pitch oscillation of NACA 64a010 for LCO condition (Case VII), $\alpha_o =$ 0.0, Re = 12560000.0, $M_a = 0.925$

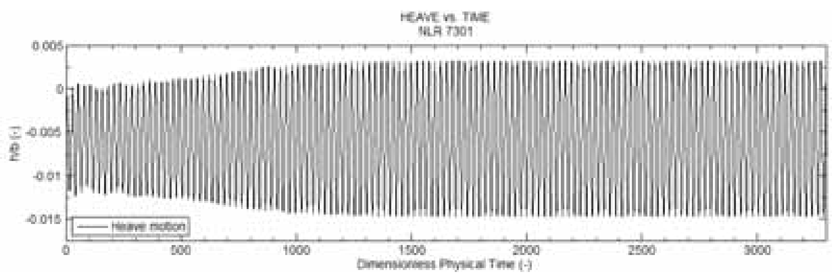

Figure 7.18: Heave oscillation of NLR 7301 for flutter condition (Case VIII), $\alpha_o =$ 0.65, Re = 1700000.0, $M_a = 0.753$

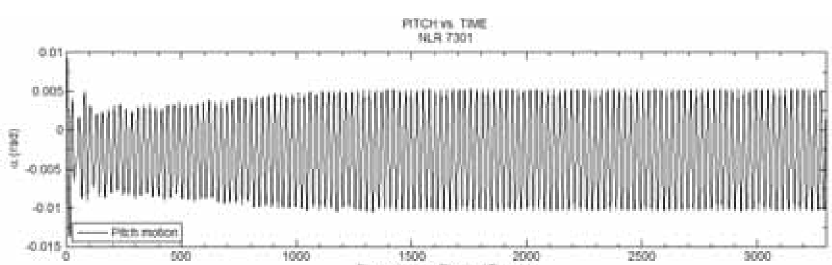

Figure 7.19: Pitch oscillation of NLR 7301 for flutter condition (Case VIII), $\alpha_o = 0.65$, Re = 1700000.0, $M_a = 0.753$

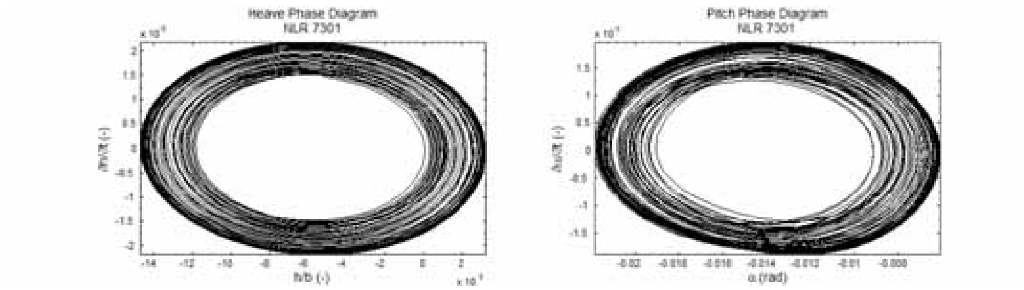

Figure 7.20: Left: Heave phase diagram of NLR 7301 for flutter condition (Case VIII). Right: Pitch phase diagram of NLR 7301 for flutter condition (Case VIII)

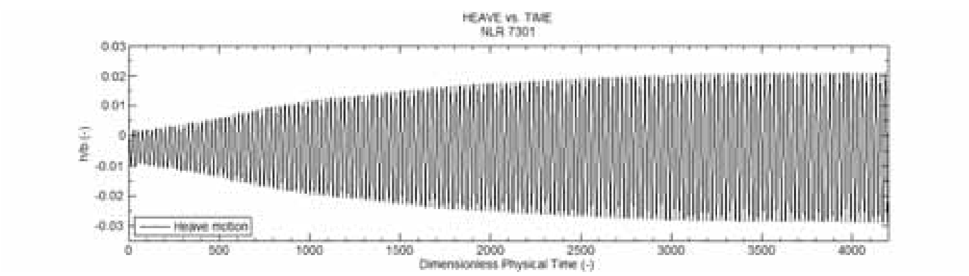

Figure 7.21: Heave oscillation of NLR 7301 for LCO condition (Case IX), $\alpha_o = 0.0$, Re = 1700000.0, $M_a = 0.753$

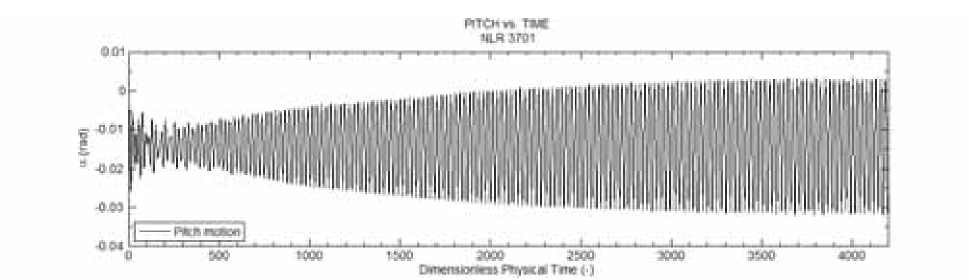

Figure 7.22: Pitch oscillation of NLR 7301 for LCO condition (Case IX), $\alpha_o = 0.0$, Re = 1700000.0, $M_a = 0.753$

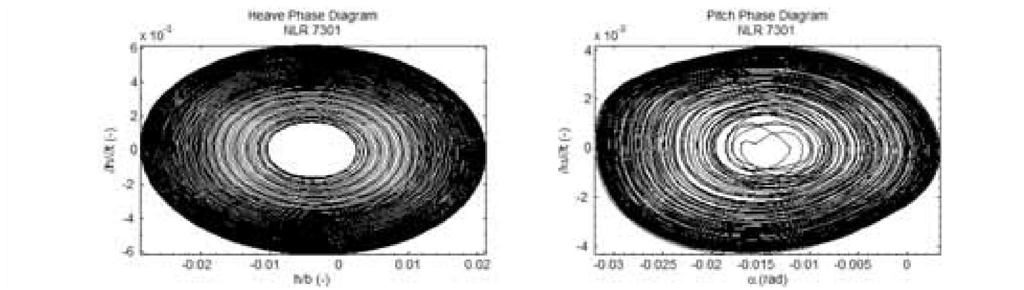

Figure 7.23: Left: Heave phase diagram for NLR 7301 in LCO condition (Case IX). Right: Pitch phase diagram for NLR 7301 in LCO condition (Case IX)

NLR 7301			
Structural and flow parameters	**Exp. [9]**	**Case VIII**	**Case IX**
Reynolds Number: Re $(-)$	1700000	169500	169500
Mach Number: Ma $(-)$	0.768	0.77	0.77
Freestream Temperature: T (K)	274	310	310
Stagnation pressure: P (Pa)	45000	45000	45000
Velocity: U_∞ $(\frac{\mathrm{m}}{\mathrm{s}})$	255	77.17	77.17
Density: ρ $(\frac{\mathrm{kg}}{\mathrm{m}^3})$	.388	506	506
Initial Angle of Attack: α_o (deg)	1.9	0.65	0.0
Velocity Index: VI $(-)$	.204	.190	.190
Mass Ratio: μ $(-)$	942	1077.2	1077.2
Reduced Pitch Frequency: ω^*_α $(-)$	0.242	.320	.320
LCO and Flutter properties			
Percent heave amplitude: $\Delta H/2$ (%)	0.365	0.425	1.25
Pitching amplitude: $\Delta\alpha/2$ (deg)	0.3	0.425	1.0
Mean Lift coefficient: c_L $(-)$	0.272	.29	.151
Mean Moment coefficient: c_M $(-)$	-0.082	-0.082	-0.068

Table 7.4: Comparison of Case VIII (flutter) and Case IX (LCO) of NLR 7301 with experimental data available in Reference [9]

NACA 64a010				
Structural and flow parameters	**Comp. [4]**	**Case V**	**Comp. [4]**	**Case VII**
Reynolds Number: Re $(-)$	12560000	12560000	12560000	12560000
Mach Number: M $(-)$	0.825	0.825	0.925	0.925
Initial Angle of Attack: α_o (deg)	0.0	0.0	0.0	0.0
Mass Ratio: μ $(-)$	60	60	60	60
LCO and Flutter properties				
Velocity Index: VI $(-)$	0.75	0.55	3.5	5.5
Percent heave amplitude: $\Delta H/2$ (%)	–	0.88	–	9.0
Pitching amplitude: $\Delta\alpha/2$ (deg)	–	0.01	–	0.063

Table 7.5: Comparison of Case VIII (flutter) and Case IX (LCO) of NACA 64a010 with experimental data available in Reference [4]

7.6 Results III: Skin Friction and Pressure Distributions

Figures 7.24 to 7.39 show the skin friction and time averaged pressure distributions for both NLR 7301 and NACA 64a010 for different unsteady state (flow-induced oscillating) conditions. These plots are obtained from a post-processing code provided in Appendix A. This code basically post-processes the time-dependent results generated by the CFD simulations, and it makes use of library DISLIN which plots the post-processed data. This post-processing code makes use of the Fortran library that evaluates the real-par and imaginary part of the unsteady pressure distributions. More details about this library are provided in Appendix B.

Figures 7.24 to 7.39 show the skin friction and pressure distributions for NACA 64a010 and NLR 7301. The skin friction distribution for NACA 64a010 in critical flutter condition shows a sharp reduction as approaching the trailing side of both sides of the airfoil, and this drop of skin friction occurs at the same location of both sides of the airfoil. This is consistent with the symmetric aspect of the wing. Since skin friction is reversely proportional to the flow velocity around the airfoil, these figures indicate that the flutter condition for NACA 64a010 is caused by the flow separation on the trailing region of the wing. The corresponding pressure distribution, as shown in figure 7.31, is consistent with this type of skin friction distribution.

Unlike the time averaged pressure distribution of NACA 64a010 in critical flutter condition, as shown in figure 7.27, the pressure distribution for NACA 64a010 in LCO does not go up and down as it approaches the trailing side. Instead, the LCO pressure distribution seems to increase at a decreasing rate from the leading to the trailing side of the wing. This increasing pressure distribution seems to be compensated by the non-reducing skin friction distribution of NACA 64a010 in LCO condition, as shown in figure 7.28.

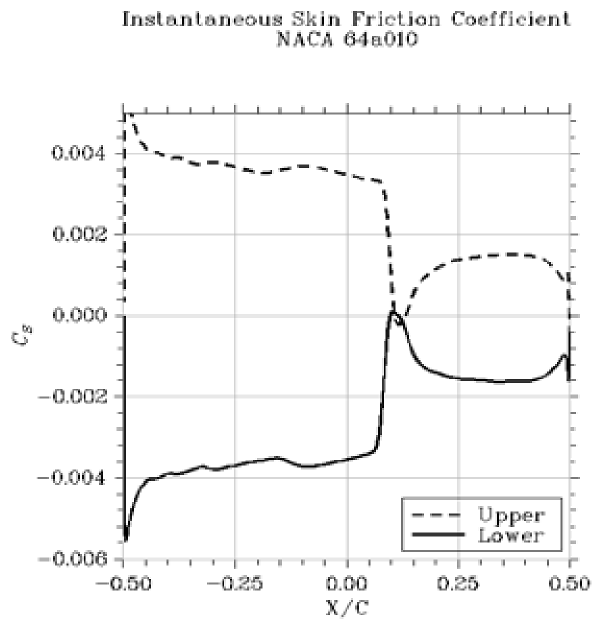

Figure 7.24: Instantaneous Skin friction distribution of NACA 64a010 in flutter oscillating condition (Case V)

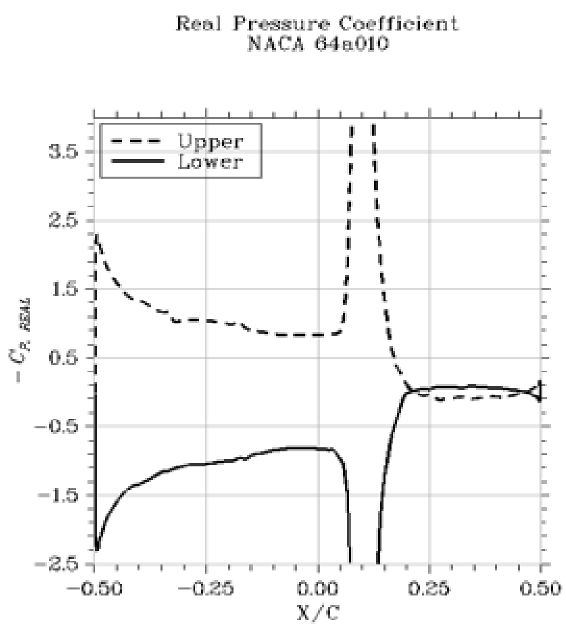

Figure 7.25: Real-part of the unsteady pressure distribution of NACA 64a010 in flutter oscillating condition (Case V)

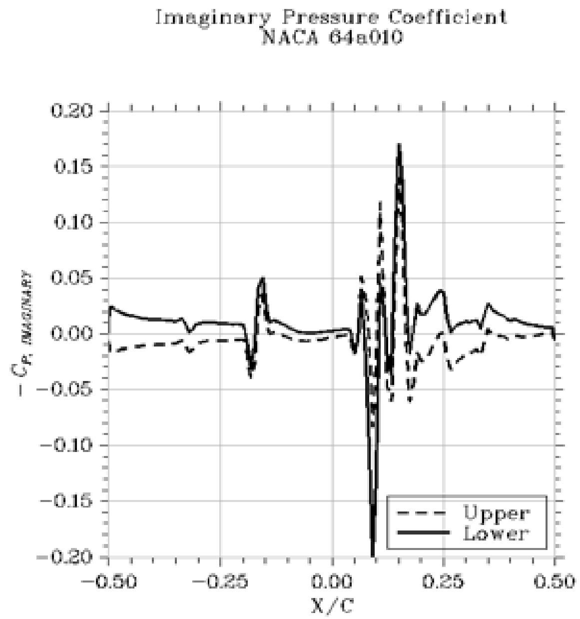

Figure 7.26: Imaginary-part of the unsteady pressure distribution of NACA 64a010 in flutter oscillating condition (Case V)

As shown in figures 7.24, 7.27, 7.36, and 7.39, the skin friction and pressure coefficient of NLR 7301 in LCO condition are similar in characteristic to those for NACA 64a010 in critical flutter condition. This indicates that the physics that causes LCO condition for NLR 7301 is similar to that of NACA 64a010 in critical flutter condition. This could also indicate that the NLR 7301 wing has been designed so that it can support the type of aerodynamic loadings associated with those of NACA 64a010 in flutter condition. However, unlike the pressure distribution for NACA 64a010 in critical flutter condition, the pressure distribution for NLR 7301 in LCO condition, as shown in figure 7.39, does not show overlapping curves. That is, the pressure distribution of the lower surface of the wing is higher than that of the upper surface. This could indicate that the NLR 7301 wing is designed so that it could be more dynamically stable than conventional wings.

The real and imaginary part of the unsteady time averaged pressure distribution of NACA 64a010 in damping condition is much more linear than those of flutter and

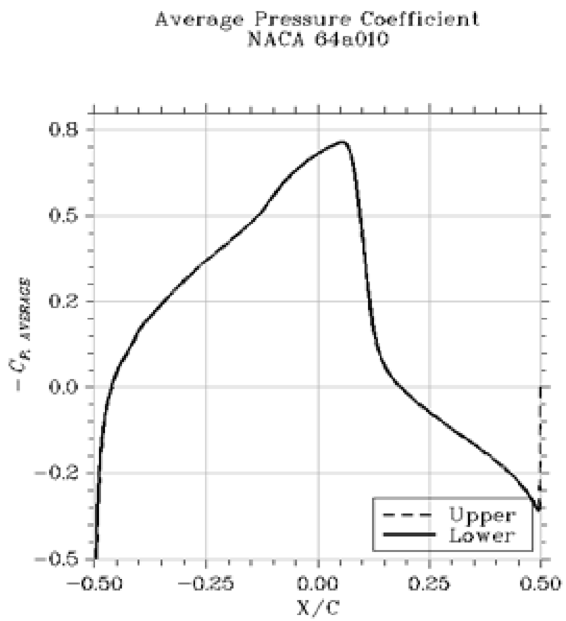

Figure 7.27: Time averaged pressure distribution of NACA 64a010 in flutter oscillating condition (Case V)

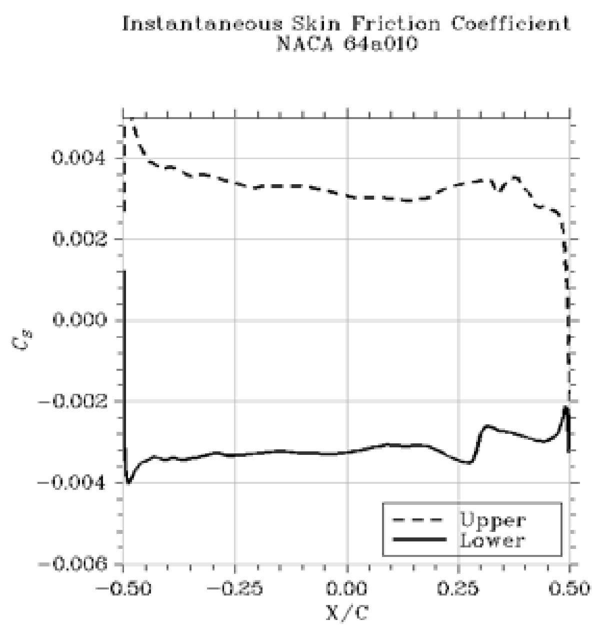

Figure 7.28: Instantaneous skin friction distribution of NACA 64a010 in LCO oscillating condition (Case VII)

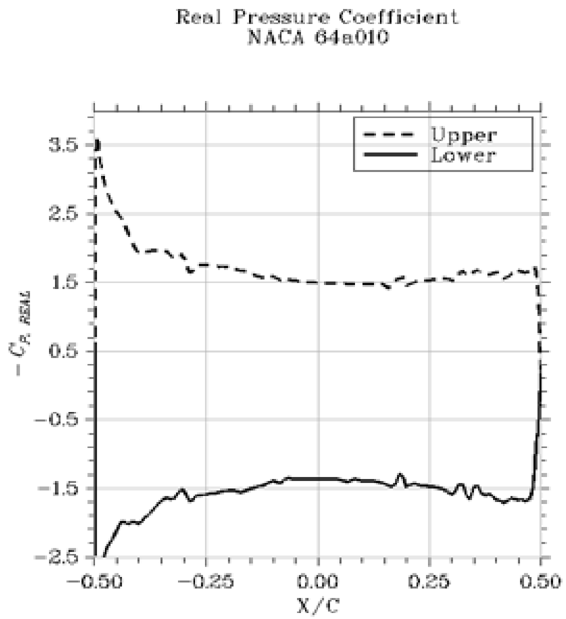

Figure 7.29: Real-part of the unsteady pressure distribution of NACA 64a010 in LCO oscillating condition (Case VI)

LCO behaviors. The real and imaginary part of the unsteady pressure distributions are obtained by using the Fast Fourier transform formula, as described in Reference B.1. These plots could serve as a measure of the variation of the unsteady pressure distribution. Figures 7.25, 7.26, 7.33, and 7.34 indicate that as oscillation is dampen out in time, the time-evolution of the pressure distribution become more linear, so that the real and imaginary-part of the unsteady pressure distribution become more linear. That is, the pressure distribution in time gets closer to the time averaged pressure distribution. The imaginary and real parts of the pressure distribution for NACA 64a010 in critical flutter condition are similar to the results obtained from Davis [33].

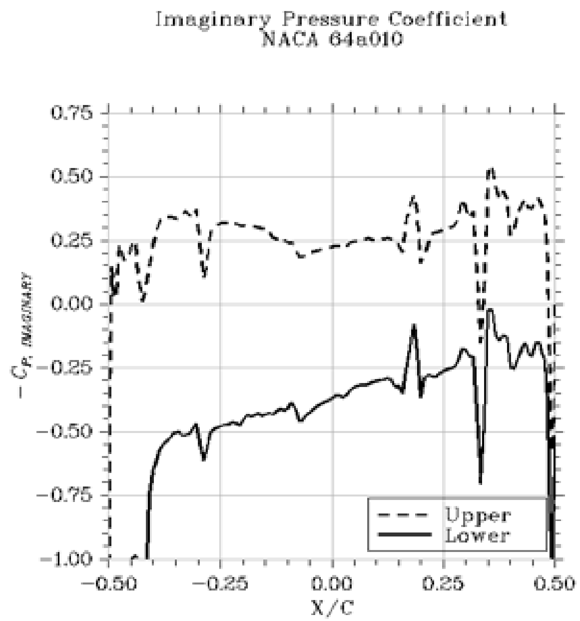

Figure 7.30: Imaginary-part of the unsteady pressure distribution of NACA64a010 in LCO oscillating condition (Case VI)

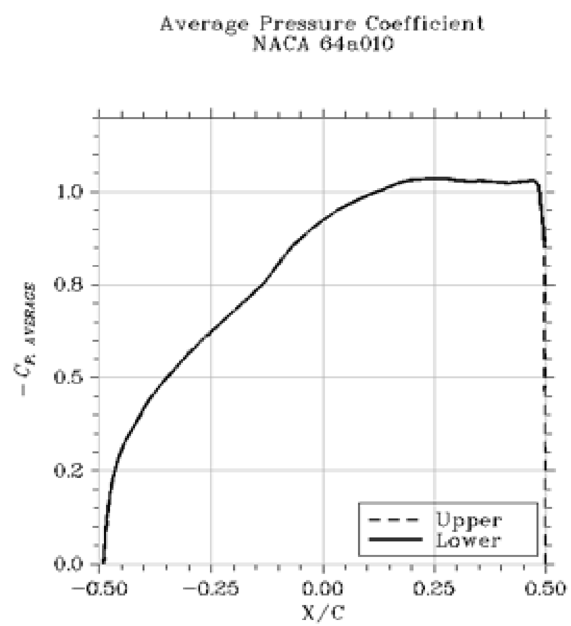

Figure 7.31: Time averaged pressure distribution of NACA 64a010 in LCO oscillating condition (Case VI)

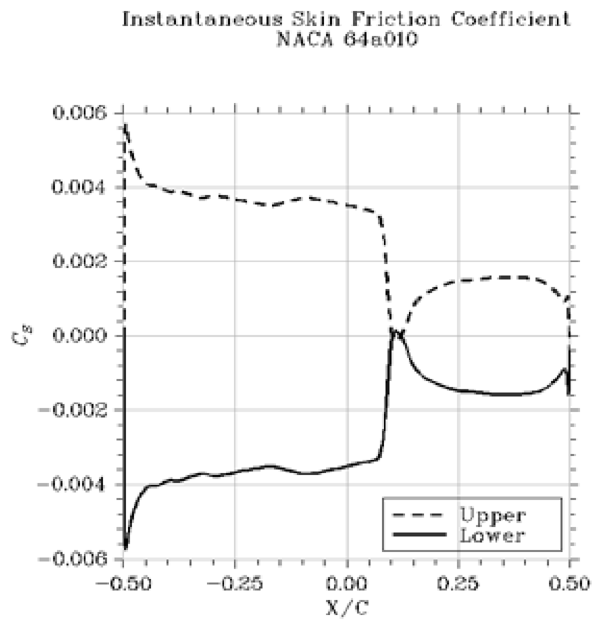

Figure 7.32: Instantaneous skin friction distribution of NACA 64a010 in damped oscillating condition (Case IV)

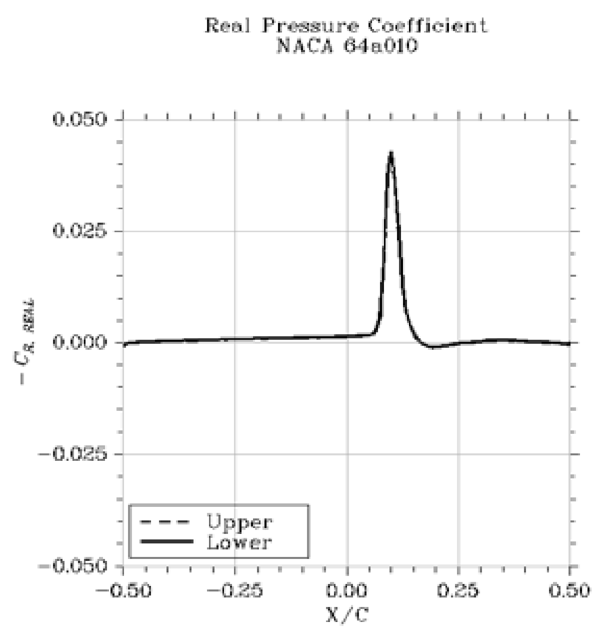

Figure 7.33: Real-part of the unsteady pressure distribution of NACA 64a010 in damped oscillating condition (Case IV)

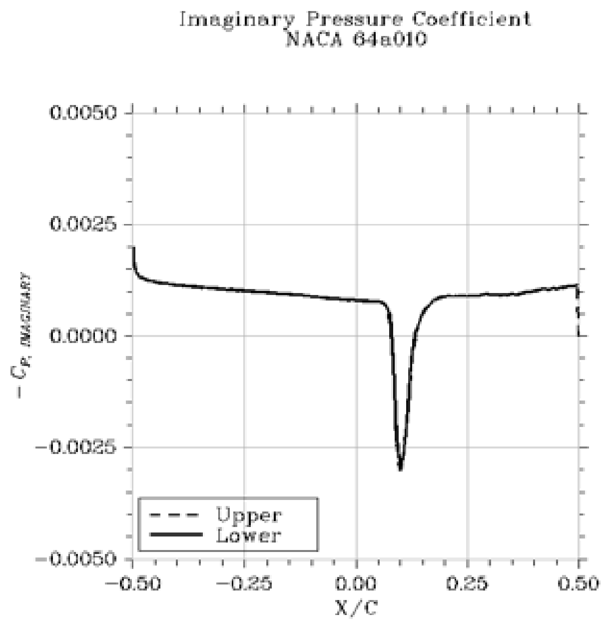

Figure 7.34: Imaginary-part of the unsteady pressure distribution of NACA 64a010 in damped oscillating condition (Case VI)

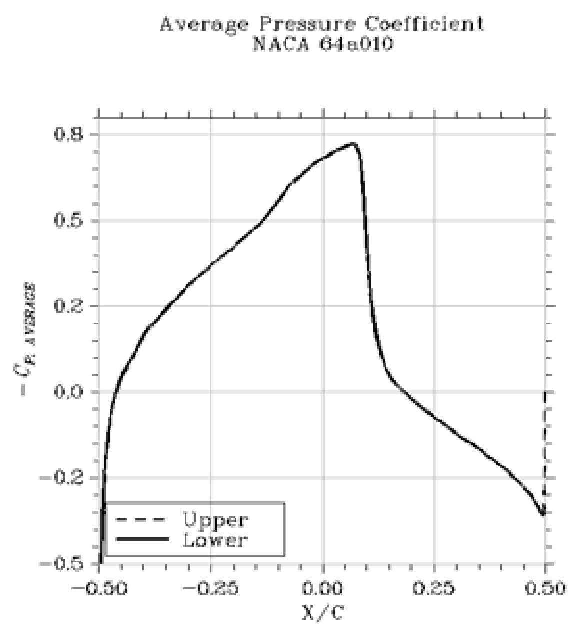

Figure 7.35: Time averaged pressure distribution of NACA 64a010 in damped oscillating condition (Case VI)

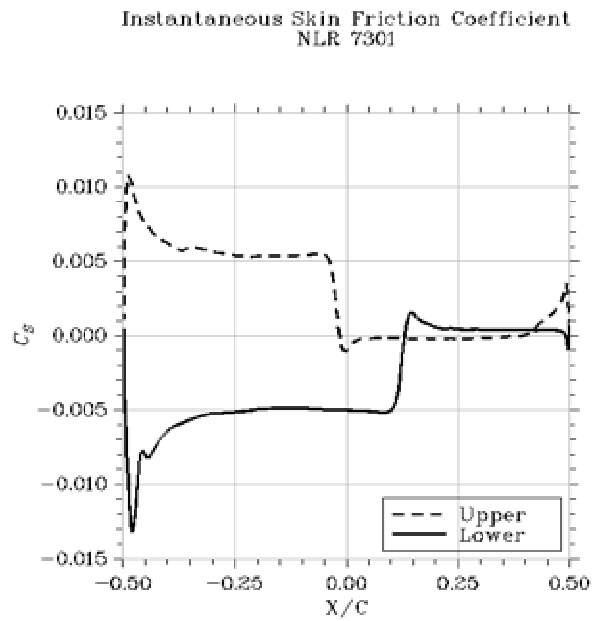

Figure 7.36: Instantaneous skin friction distribution of NLR 7301 in LCO condition (Case IX)

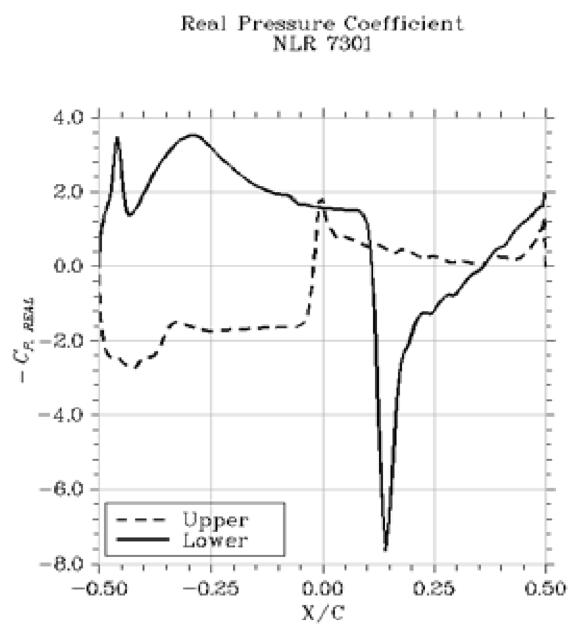

Figure 7.37: Real-part of the unsteady pressure distribution of NLR 7301 in LCO condition (Case IX)

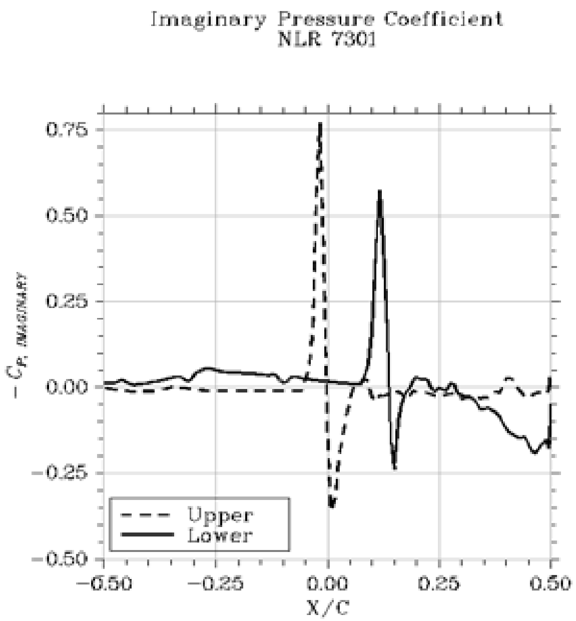

Figure 7.38: Imaginary-part of the unsteady pressure distribution of NLR 7301 in LCO condition (Case IX)

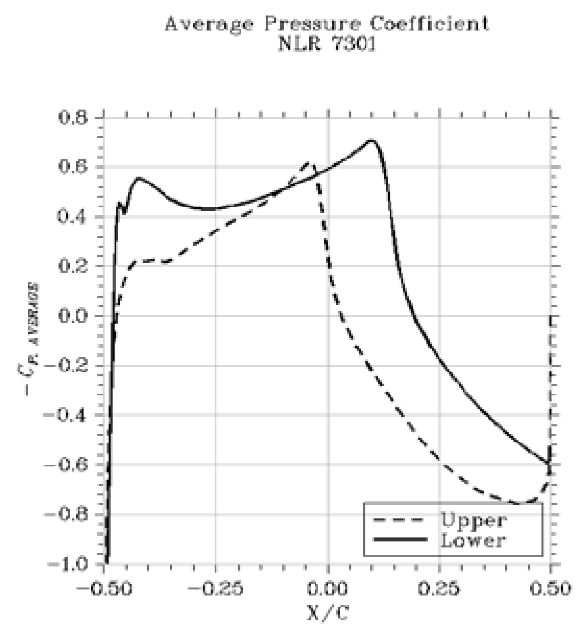

Figure 7.39: Time averaged pressure distribution of NLR 7301 in LCO condition (Case IX)

Chapter 8

Conclusion

As shown in the plots given in the previous chapter, the CFD solver is able to capture limit cycle oscillation (LCO) for both the NLR 7301 and NACA 64a010 airfoils. Both the Zha scheme and Roe scheme can be used to obtain these results, but require different flow parameters to obtain similar results. It has been observed though, that the CFD solver can be run with the Zha scheme using a Courant-Friedrichs-Lewy (CFL) value of 1.0. On the other hand, the Roe scheme could be used with CFL values that are either higher or lower than 1.0. Since the Roe scheme does not have this CFL constraint, it is ultimately used to obtain both the flutter and LCO behaviors for both the NACA 64a010 and NLR 7301 airfoils, as well as the results to validate the CFD solver and the mesh grid of both airfoils.

The results obtained from these CFD simulations provide an understanding of the LCO and flutter behaviors for both the NACA 64a010 and NLR 7301 wing. The surface pressure distribution and the skin friction distribution helps in understanding the physics of these two flow behaviors. The understanding of the CFD simulation process to capture LCO and flutter behaviors for both the NACA 64a010 and the NLR 7301 airfoils, and also, the knowledge of the flow and structural parameters that are required to obtain these aeroelastic plots, could serve as an indispensable guideline for future CFD studies of LCO using two and three dimensional geometries.

Appendix A

CFD Post-Processing Code

The skin friction coefficient and unsteady pressure distribution plots displayed in section 7.6 are all generated using a post-processing code made by the author of this report. This code is written in FORTRAN 77, and it makes use of non-commercial plotting library called DISLIN©. This is an advanced plotting library that is available at the website http://www.mps.mpg.de/dislin/. This code also makes use of another Fortran library called SSL2©. This is a library that provides various advanced mathematical subroutines. Specifically, it performs Fourier interpolations of time dependent data, such as the CFD unsteady time-dependent fluid flow parameters. This library is used to obtain the imaginary and real part of the unsteady pressure distribution. The output that is generated by this code is plotted using the DISLIN plotting library. This code is documented in the following section.

A.1 Code: cdlt_all.f90

cdlt_all.f90

```
      PROGRAM CDLT_ALL

      USE DISLIN
      IMPLICIT NONE
      INTEGER, PARAMETER :: N=281, NN=141, M=280, &
      TIME_STEPS=500
      DOUBLE PRECISION, DIMENSION(N) :: Y11, Y12, Y13, &
```

```
      Y14, Y15, Y16, Y17, Y18, Y19
      DOUBLE PRECISION, DIMENSION(M) :: Y1, Y2, Y3, Y4, &
      Y5, Y6, Y7, Y8, Y9, Y110, Y111
      DOUBLE PRECISION, DIMENSION(TIME_STEPS, M):: Z1,&
      Z2, Z3, Z4, Z5, Z6, Z7, Z8, Z9, Z110, Z111
      DOUBLE PRECISION, DIMENSION(TIME_STEPS,N):: Z11, &
      Z12, Z13, Z14, Z15
      DOUBLE PRECISION, DIMENSION(M) :: THETA2, THETA3
      DOUBLE PRECISION, DIMENSION(TIME_STEPS, M):: THETA3Z
      DOUBLE PRECISION, DIMENSION(N) :: X1, X2, X11
      REAL, DIMENSION (NN) :: XRAY,Y1RAY,Y2RAY, Y3RAY, &
      Y4RAY, Y5RAY, Y7RAY, Y8RAY, Y9RAY
      REAL, DIMENSION (NN) :: Y01RAY, Y02RAY, Y03RAY, &
      Y04RAY, Y07RAY,Y08RAY, Y09RAY

      REAL, DIMENSION (TIME_STEPS, NN) :: XRAYZ,Y1RAYZ, &
      Y2RAYZ, Y3RAYZ, Y4RAYZ,Y5RAYZ, Y7RAYZ, Y8RAYZ, Y9RAYZ
      REAL, DIMENSION (TIME_STEPS, NN) :: Y01RAYZ, Y02RAYZ, &
      Y03RAYZ, Y04RAYZ,Y07RAYZ, Y08RAYZ, Y09RAYZ

      REAL, DIMENSION (NN) :: X11RAY, X22RAY
      DOUBLE PRECISION :: DUM1, DUM2, DUM3, DUM4, DUM5, &
      DUM6, DX, DX2
      REAL, PARAMETER :: PI=3.1415926
      REAL :: FPI,STEP,X
      INTEGER :: I,J, K, II, IC, I2, I3, IJ, I21, &
      FLAG, ISN, JJ, ICONN
      CHARACTER (LEN=20) FILENAME1, LINESS, CBUF*24
      CHARACTER (LEN=20) FILENAME2, FILENAME3

      REAL, ALLOCATABLE, DIMENSION(:,:) :: Y7COFF1, Y07COFF1,&
      Y7COFF2, Y07COFF2, &
      Y8COFF1, Y8COFF2, Y08COFF1, Y08COFF2
      REAL, ALLOCATABLE, DIMENSION(:) :: A
      REAL, ALLOCATABLE, DIMENSION(:,:) :: AA
      FPI=PI/180.
      STEP=360./(N-1)

      CALL METAFL('EPS')
      CALL PAGE (2000, 2000)
      WRITE(FILENAME1, '("cdlt_all.his")')
      OPEN(UNIT = 1, FILE = FILENAME1, FORM = 'UNFORMATTED',&
      ACTION = 'READ', STATUS = 'OLD')

      WRITE(FILENAME2, '("ch_nlr")')
      OPEN(UNIT = 2, FILE = FILENAME2, FORM = 'FORMATTED',&
      ACTION = 'READ', STATUS = 'OLD')
```

```

      WRITE(FILENAME3, '("ch_nlr")')
      OPEN(UNIT = 3, FILE = FILENAME3, FORM = 'FORMATTED',&
      ACTION = 'READ', STATUS = 'OLD')


!     Reading x-wise position for plots.

      DO I = 1, 281
      READ(2,300) X11(I)
      ENDDO
      DO J =1, 141
      X11RAY(J)= REAL(X11(J))
      X22RAY(J)= REAL(X11(J+140))
      END DO

!     Start main time iteration loop
      DO I = 1, TIME_STEPS

!     Several input parameters are read
      READ(1) (Y1(J), J=1,280)
      DO k = 1, 280
      Z1(I,k)= Y1(k)
      END DO

      READ(1) (Y2(J), J=1,280)
      DO k = 1, 280
      Z2(I,k)= Y2(k)
      END DO

      READ(1) (Y3(J), J=1,280)
      DO k = 1, 280
      Z3(I,k)= Y3(k)
      END DO

      READ(1) (Y4(J), J=1,280)
      DO k = 1, 280
      Z4(I,k)= Y4(k)
      END DO

      READ(1) (Y5(J), J=1,280)
      DO k = 1, 280
      Z5(I,k)= Y5(k)
      END DO

      READ(1) (Y6(J), J=1,280)
      DO k = 1, 280
```

```
      Z6(I,k)= Y6(k)
      END DO

      READ(1) (Y7(J), J=1,280)
      DO k = 1, 280
      Z7(I,k)= Y7(k)
      END DO

      READ(1) (Y11(J), J=1,281)
      DO k = 1, 281
      Z11(I,k)= Y11(k)
      END DO

      READ(1) (Y12(J), J=1,281)
      DO k = 1, 281
      Z12(I,k)= Y12(k)
      END DO

      READ(1) (Y13(J), J=1,281)
      DO k = 1, 281
      Z13(I,k)= Y13(k)
     END DO

!     Converting pressure coefficient, Y11, from double precision
      to real type

      DO k =1,141
      Y7RAYZ(i,k)=REAL(Y11(k))
      Y07RAYZ(i,k)=REAL(Y11(k+140))
      Y7RAYZ(i,k)=-2.*(Y7RAYZ(i,k)-1.)
      Y07RAYZ(i,k)=-2.*(Y07RAYZ(i,k)-1.)
       END DO

!     Angular projection of line perpendicular to  wing surface,
!     positive counterclockwise.

      DO J = 1, 280
      DX2 = Y12(J+1) - Y12(J)
      IF (DX2.GT.0.) THEN
      THETA3(J) = ATAN((Y13(J+1)-Y13(J))/(Y12(J+1)-Y12(J)))
      ELSE
      THETA3(J) = ATAN((Y13(J)-Y13(J-1))/(Y12(J)-Y12(J-1)))
      ENDIF
      END DO
      DO k = 1, 280
      THETA3Z(I,k)=THETA3(k)
      END DO
```

```
      IJ = 0
      DO J = 1,280
      IJ = IJ + 1
      DX = X11(IJ+1)-X11(IJ)

      IF(DX.LT.0) THEN

!     Shear Stress in the x-direction
      Y110(J)= -(Y1(J)-Y5(J))*sin(THETA3(J))*cos(THETA3(J))+&
      Y3(J)*(cos(THETA3(J))*cos(THETA3(J)) -&
       sin(THETA3(J))*sin(THETA3(J)))

!     Shear Stress in the y-direction
      Y111(J)= Y1(J)*cos(THETA3(J))*cos(THETA3(J))+&
      Y5(J)*sin(THETA3(J))*sin(THETA3(J))+&
      2.*Y3(J)*cos(THETA3(J))*SIN(THETA3(J))

!     Multiplication of shear stresses by negative sign
      Y110(J) = Y110(J)
      Y111(J) = Y111(J)

      ELSE

!     Shear stresses in the x-direction
      Y110(J)= -(Y1(J)-Y5(J))*sin(THETA3(J))*cos(THETA3(J))+&
      Y3(J)*(cos(THETA3(J))*cos(THETA3(J)) -&
      sin(THETA3(J))*sin(THETA3(J)))

!     Shear stresses in the y-direction
      Y111(J)= Y1(J)*cos(THETA3(J))*cos(THETA3(J))+&
      Y5(J)*sin(THETA3(J))*sin(THETA3(J))+&
      2.*Y3(J)*cos(THETA3(J))*SIN(THETA3(J))

      ENDIF
      END DO

      DO k=1,280
      Z110(I,J)=Y110(J)
      Z111(I,J)=Y111(J)
      END DO

      DO J = 1, 141
      IF (J.EQ.1)THEN
      Y8RAY(J) = 2.*REAL(.5*(Y110(1)+Y110(280)))
      ELSE
      IF(J.EQ.141) THEN
```

```
      Y8RAY(J) = 2.*REAL(.5*(Y110(141)+Y110(140)))
      ELSE
      Y8RAY(J) = 2.*REAL(.5*(Y110(J)+Y110(J-1)))
      END IF
      END IF
      END DO

      II  = 1
      DO  J = 141, 281
      IF (J.EQ.141)THEN
      Y08RAY(I) = 2.*REAL(.5*(Y110(140)+Y110(141)))
      II = II + 1
      ELSE
      IF(J.EQ.281) THEN
      Y08RAY(II) = 2.*REAL(.5*(Y110(280)+Y110(1)))
      II = II + 1
      ELSE
      Y08RAY(II) = 2.*REAL(.5*(Y110(J)+Y110(J-1)))
      II = II + 1
      END IF
      END IF
      END DO

      DO J = 1, 141
      IF (J.EQ.1)THEN
      Y9RAY(J) = 2.*REAL(.5*(Y111(1)+Y111(280)))
      ELSE
      IF(J.EQ.141) THEN
      Y9RAY(J) = 2.*REAL(.5*(Y111(141)+Y111(140)))
      ELSE
      Y9RAY(J) = 2.*REAL(.5*(Y111(J)+Y111(J-1)))
      END IF
      END IF
      END DO

      II  = 1
      DO  J = 141, 281
      IF (J.EQ.141)THEN
      Y09RAY(I) = 2.*REAL(.5*(Y111(140)+Y111(141)))
      II = II + 1
      ELSE
      IF(J.EQ.281) THEN
      Y09RAY(II) = 2.*REAL(.5*(Y111(280)+Y111(1)))
      II = II + 1
      ELSE
      Y09RAY(II) = 2.*REAL(.5*(Y111(J)+Y111(J-1)))
      II = II + 1
```

```
      END IF
      END IF
      END DO

!    Store non-time-dependent Y8, Y08, Y9,Y09 on
!    time-dependent parameters

      DO k=1,141
      Y8RAYZ(I,k)=Y8RAY(k)
      Y08RAYZ(I,k)=Y08RAY(k)
      Y9RAYZ(I,k)=Y9RAY(k)
      Y09RAYZ(I,k)=Y09RAY(k)
      END DO

!     End of main time iteration loop

      END DO

!     Calculate time steps that's required for FT
!     analysis subroutine.

      I3 = 0
      FLAG = 0
      DO WHILE(FLAG.NE.1)
      I3 = I3 + 1
      I2 = 2**I3
      IF(I2.GT.TIME_STEPS.OR.I2.EQ.TIME_STEPS) THEN
      IF(I2.EQ.TIME_STEPS)THEN
      FLAG = 1
      END IF
      IF(I2.GT.TIME_STEPS)THEN
      FLAG = 1
      I3 = I3 - 1
      I2 = 2**I3
      END IF
      END IF
      END DO

      ALLOCATE (A(I2))
      ALLOCATE (AA(I2,141))

!     Calculate FT output for upper surface of wing

      DO J = 1, 141
      DO k=1,I2
      A(k)=Y07RAYZ(k,J)
      ENDDO
```

```
      ISN = 1
      I21 = I2/2 - 1
      CALL RFT(A,I2,ISN,ICONN)
      DO k = 1, I2
      AA(k,J)= A(k)
      END DO
      END DO

      ALLOCATE (Y07COFF1(I21,141))
      ALLOCATE (Y07COFF2(I21, 141))

!    Store FT output on time-dependent variables

      jj = 1
     DO J = 1, 141
     DO k=1,I21
     Y07COFF1(k,J)=AA(2*k+1,J)
     Y07COFF2(k,J)=AA(2*k+2,J)
      jj = jj + 1
      END DO
      END DO

!     Calculate FT output for lower surface of wing

      DO J = 1, 141
      DO k=1,I2
      A(k)=Y7RAYZ(k,J)
      ENDDO
      ISN = 1
      I21 = I2/2 - 1
      CALL RFT(A,I2,ISN,ICONN)
      DO k = 1, I2
      AA(k,J)= A(k)
      END DO
      END DO

      ALLOCATE (Y7COFF1(I21,141))
      ALLOCATE (Y7COFF2(I21,141))

!     Store FT output on time-dependent variables

      DO J = 1, 141
      DO k=1,I21
      Y7COFF1(k,J)=AA(2*k+1,J)
      Y7COFF2(k,J)=AA(2*k+2,J)
      END DO
      END DO
```

```
100     FORMAT(280(e20.14,1x))
200     FORMAT(281(e20.14,1x))
300     FORMAT(E20.14)

!      Plot of Skin Coefficients

       CALL DISINI()
       CALL COMPLX()
       CALL AXSPOS(450,1800)
       CALL AXSLEN(1200,1200)
       CALL CHNCRV('LINE')
       CALL NAME('Chord Position  X/C','X')
       CALL NAME('Skin Friction Coefficient ','Y')

       CALL LABDIG(2,'X')
       CALL LABDIG(3,'Y')
       CALL LEGTIT(' ')
       CALL TICKS(10,'XY')

       CALL LEGINI(CBUF,2,8)
       CALL LEGLIN(CBUF,'Lower Cp',1)
       CALL LEGLIN(CBUF,'Upper Cp',2)
       CALL LEGTIT(' ')
       CALL TITLIN('Skin Friction Coefficient',1)
       CALL TITLIN('NLR 7301',2)
       IC=INTRGB(1.,1.,1.)
       CALL AXSBGD(IC)

       CALL GRAF(-.5,.5,-.5,.25,&
       -.008,.008,-.008,.001)
       CALL SETRGB(0.7,0.7,0.7)
       CALL GRID(1,1)

       CALL COLOR('FORE')
       CALL TITLE()

       DO k=1,141
       Y8RAY(k)=Y8RAYZ(TIME_STEPS,k)
       Y08RAY(k)=Y08RAYZ(TIME_STEPS,k)
       END DO

       CALL SETRGB(0.,0.,0.)
       CALL CURVE(X11RAY,Y8RAY,141)
       CALL SETRGB(0.,0.,0.)
       CALL CURVE(X22RAY,Y08RAY,141)
       CALL LEGEND(CBUF,5)
```

```
      CALL DISFIN()

!    Plot of Pressure Coefficients

      CALL DISINI()
      CALL COMPLX()
      CALL AXSPOS(450,1800)
      CALL AXSLEN(1200,1200)
      CALL CHNCRV('LINE')
      CALL NAME('Chord Position  X/C','X')
      CALL NAME('Pressure Coefficient ','Y')

      CALL LABDIG(2,'X')
      CALL LABDIG(2,'Y')
      CALL LEGTIT(' ')
      CALL TICKS(10,'XY')

      CALL LEGINI(CBUF,2,8)
      CALL LEGLIN(CBUF,'Lower Cp',1)
      CALL LEGLIN(CBUF,'Upper Cp',2)
      CALL LEGTIT(' ')
      CALL TITLIN('Pressure Coefficient',1)
      CALL TITLIN('NLR 7301',2)
      IC=INTRGB(1.,1.,1.)
      CALL AXSBGD(IC)

      CALL GRAF(-.5,.5,-.5,.25,&
      -.5,1.,-.5,.25)
      CALL SETRGB(0.7,0.7,0.7)
      CALL GRID(1,1)

      CALL COLOR('FORE')
      CALL TITLE()

      DO k=1,141
      Y7RAY(k)=Y7RAYZ(TIME_STEPS,k)
      Y07RAY(k)=Y07RAYZ(TIME_STEPS,k)
      END DO

      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X11RAY,Y7RAY,141)
      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X22RAY,Y07RAY,141)
      CALL LEGEND(CBUF,5)
      CALL DISFIN()

!     Plot of Real Pressure Distribution
```

```

      CALL DISINI()
      CALL COMPLX()
      CALL AXSPOS(450,1800)
      CALL AXSLEN(1200,1200)
      CALL CHNCRV('LINE')
      CALL NAME('Chord Position  X/C','X')
      CALL NAME('Real Pressure Coefficient ','Y')

      CALL LABDIG(2,'X')
      CALL LABDIG(1,'Y')
      CALL LEGTIT(' ')
      CALL TICKS(10,'XY')

      CALL LEGINI(CBUF,2,8)
      CALL LEGLIN(CBUF,'Lower Cp',1)
      CALL LEGLIN(CBUF,'Upper Cp',2)
      CALL LEGTIT(' ')
      CALL TITLIN('Real Pressure Coefficient',1)
      CALL TITLIN('NLR 7301',2)
      IC=INTRGB(1.,1.,1.)
      CALL AXSBGD(IC)

      CALL GRAF(-.5,.5,-.5,.25,&
      -18.,16.,-18.,2.)
      CALL SETRGB(0.7,0.7,0.7)
      CALL GRID(1,1)

      CALL COLOR('FORE')
      CALL TITLE()

      DO k=1,141
      Y7RAY(k)=Y7COFF1(2,k)
      Y07RAY(k)=Y07COFF1(2,k)
      END DO

      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X11RAY,Y7RAY,141)
      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X22RAY,Y07RAY,141)
      CALL LEGEND(CBUF,5)
      CALL DISFIN()

!     Plot of Imaginary Pressure Distribution

      CALL DISINI()
      CALL COMPLX()
```

```
      CALL AXSPOS(450,1800)
      CALL AXSLEN(1200,1200)
      CALL CHNCRV('LINE')
      CALL NAME('Chord Position  X/C','X')
      CALL NAME('Imaginary Pressure Coefficient ','Y')

      CALL LABDIG(2,'X')
      CALL LABDIG(1,'Y')
      CALL LEGTIT(' ')
      CALL TICKS(10,'XY')

      CALL LEGINI(CBUF,2,8)
      CALL LEGLIN(CBUF,'Lower Cp',1)
      CALL LEGLIN(CBUF,'Upper Cp',2)
      CALL LEGTIT(' ')
      CALL TITLIN('Imaginary Pressure Coefficient',1)
      CALL TITLIN('NLR 7301',2)
      IC=INTRGB(1.,1.,1.)
      CALL AXSBGD(IC)

      CALL GRAF(-.5,.5,-.5,.25,&
      -8.,8.,-8.,2.)
      CALL SETRGB(0.7,0.7,0.7)
      CALL GRID(1,1)

      CALL COLOR('FORE')
      CALL TITLE()

      DO k=1,141
      Y7RAY(k)=Y7COFF2(2,k)
      Y07RAY(k)=Y07COFF2(2,k)
      END DO

      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X11RAY,Y7RAY,141)
      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X22RAY,Y07RAY,141)
      CALL LEGEND(CBUF,5)
      CALL DISFIN()

!     Plot of Mean Pressure Distribution

      CALL DISINI()
      CALL COMPLX()
      CALL AXSPOS(450,1800)
      CALL AXSLEN(1200,1200)
      CALL CHNCRV('LINE')
```

```
      CALL NAME('Chord Position  X/C','X')
      CALL NAME('Mean Pressure Coefficient ','Y')

      CALL LABDIG(2,'X')
      CALL LABDIG(1,'Y')
      CALL LEGTIT(' ')
      CALL TICKS(10,'XY')

      CALL LEGINI(CBUF,2,8)
      CALL LEGLIN(CBUF,'Lower Cp',1)
      CALL LEGLIN(CBUF,'Upper Cp',2)
      CALL LEGTIT(' ')
      CALL TITLIN('Mean Pressure Coefficient',1)
      CALL TITLIN('NLR 7301',2)
      IC=INTRGB(1.,1.,1.)
      CALL AXSBGD(IC)

      CALL GRAF(-.5,.5,-.5,.25,&
      -18.,16.,-18.,2.)
      CALL SETRGB(0.7,0.7,0.7)
      CALL GRID(1,1)

      CALL COLOR('FORE')
      CALL TITLE()

      DO k=1,141
      Y7RAY(k)=Y7COFF1(1,k)
      Y07RAY(k)=Y07COFF1(1,k)
      Y7RAY(k)= Y7RAY(k)
      Y07RAY(k)=Y07RAY(k)
      END DO

      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X11RAY,Y7RAY,141)
      CALL SETRGB(0.,0.,0.)
      CALL CURVE(X22RAY,Y07RAY,141)
      CALL LEGEND(CBUF,5)
      CALL DISFIN()

      STOP
      END PROGRAM CDLT_ALL

```
cdlt_all.f90

Appendix B

The Scientific Subroutine Library

A commercial Fortran math library, called SSL2 is employed for performing a Fourier interpolation of the time dependent CFD data, like the pressure, lift and moment coefficients [38]. Basically, this is a Fortran library that is installed for usage in Fortran codes. *Discrete Fourier Transform* is the name of the particular subroutine burrowed from this library. The procedure of using this subroutine is available in its manual, which could be found at `http://www.lahey.com/docs/ssl2_lin62.pdf`.

B.1 The Discrete Fourier Transform Subroutine

In chapter 7.3, the real and imaginary components of the Fourier interpolation of time dependent CFD data are plotted. As mentioned above, this is done by a Fortran subroutine named *Discrete Fourier Transformation* that is available in a Fortran scientific subroutine library. The input that is needed for the DFT subroutine is basically an array of time dependent data such as the flow coefficients given in chapter 7.3. The size of the input data has to be equal to 2^i, where the variable i is a non-negative integer. The subroutine is capable of computing either the inverse or non-inverse Fourier transforms. In this case, the inverse Fourier transform is used, so that the imaginary and real coefficients from the general Fourier equation are obtained.

The general expressions for both inverse and non-inverse Fourier transforms are given in B.1 and B.2, respectively.

$$
\begin{aligned}
a_k &= \frac{2}{n}\sum_{j=0}^{n-1} x_j \cos\frac{2\pi kj}{n}, \quad k = 0, ..., \frac{n}{2} \\
b_k &= \frac{2}{n}\sum_{j=0}^{n-1} x_j \sin\frac{2\pi kj}{n}, \quad k = 1, ..., \frac{n}{2} - 1
\end{aligned} \tag{B.1}
$$

$$
x_j = \frac{1}{2}a_o + \sum_{j=0}^{n-1}\left(a_k \cos\frac{2\pi kj}{n} + b_k \sin\frac{2\pi kj}{n}\right) + \frac{1}{2}a_{\frac{n}{2}} \cos \pi j, \quad j = 0, ..., n-1 \tag{B.2}
$$

Appendix C

Transformations of Shear Stresses

As displayed in chapter 7.3, the shear stress coefficients along the surface of an airfoil are plotted. These values are calculated indirectly from the subroutine in the CFD solver called *cdltj.f* [13]. Basically, this subroutine generates the shear stresses congruent with the Cartesian coordinates. Therefore, these parameters undergo a transformation such that the new values are shear stress parallel and perpendicular to the surface of the airfoil. In this case, the new parallel shear stress, because it is representative of the skin friction coefficients. The transformation equation used to carry out the necessary transformations is expressed as [39]:

$$\tau_{x'y'} = -\left(\sigma_x - \sigma_y\right)\sin\theta\cos\theta + \tau_{xy}\left(\cos^2\theta - \sin^2\theta\right), \tag{C.1}$$

where the definitions of θ, $\tau_{x'y'}$, σ_x, σ_y, and τ_{xy} are defined as the angular projection, the x-direction stress, the y-direction stress, and the shear stress, respectively, associated with a segment of a shape such as a wing.

Appendix D

Derivation of Equations

In deriving the governing pitch and heave equations, it is necessary to know the direction of force and moment applied. This is important because, if the directions are not correct, than the values of heave and pitch will not be correct, causing miscalculations of the conservative variables. The procedure presented in the following is given in detail, so as to provide a full comprehension of the nondimensionalized parameters, most of which are embedded in the CFD solver.

D.1 Pitching Mode without Damping

By applying Newton's law of force summation, the pitch governing equation, without damping, can be formulated as follow:

$$mcx_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + k_\alpha \alpha = M \tag{D.1}$$

Eq. D.1 is divided by $m.$ and c to obtain

$$x_\alpha \ddot{h} + \frac{I_\alpha}{mc}\ddot{\alpha} + \frac{k_\alpha}{mc}\alpha = \frac{1}{mc}M. \tag{D.2}$$

Hence, both m and c are removed from the first term of eq. D.1. Coincidentally, this process eliminates the need to non-dimensionalize I_α and k_α, as explained later. Eq.D.2 is then divided by c in order to non-dimensionalize the chord length.

$$\frac{x_\alpha}{c}\ddot{h} + \frac{I_\alpha}{mc^2}\ddot{\alpha} + \frac{k_\alpha}{mc^2}\ddot{\alpha} = \frac{1}{mc^2}M \tag{D.3}$$

The time parameter is then non-dimensionalized by dividing Eq. D.3 by $\frac{U_\infty^2}{c^2}$, as follow:

$$\frac{x_\alpha c^2}{cU_\infty^2}\ddot{h} + \frac{I_\alpha c^2}{mc^2U_\infty^2}\ddot{\alpha} + \frac{k_\alpha c^2}{mc^2U_\infty^2}\ddot{\alpha} = \frac{c^2}{mc^2U_\infty^2}M, \tag{D.4}$$

which can be simplified as:

$$x_\alpha\ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{r_\alpha^2 k_\alpha c^2}{I_\alpha U_\infty^2}\alpha = \frac{c^2}{mc^2U_\infty^2}M. \tag{D.5}$$

Eq. D.5 is multiplied by $\frac{\frac{\rho_\infty}{2}}{\frac{\rho_\infty}{2}}$, as follow:

$$x_\alpha\ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{r_\alpha^2 k_\alpha c^2}{I_\alpha U_\infty^2}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{mc^2U_\infty^2\frac{\rho_\infty}{2}}M, \tag{D.6}$$

which can be simplified by using the definition of Moment coefficient, that is, $C_m = \frac{M}{\frac{\rho_\infty}{2}U_\infty^2c^2}$.

$$x_\alpha\ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{r_\alpha^2 k_\alpha c^2}{I_\alpha U_\infty^2}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{m}C_M \tag{D.7}$$

Eq. D.7 is further modified by applying the definition of viscosity, that is, $\mu = \frac{m}{\pi\rho_\infty b^2}$.

$$x_\alpha\ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{r_\alpha^2 k_\alpha c^2}{I_\alpha U_\infty^2}\alpha = \frac{2}{\mu\pi}C_M \tag{D.8}$$

Eq. D.8 is further modified by applying the definition of reduced frequency, that is, $\omega_\alpha^{2*} = \frac{k_\alpha c^2}{I_\alpha U_\infty^2}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + r_\alpha^2 \omega_\alpha^{2*} \alpha = \frac{2}{\mu\pi} C_M \tag{D.9}$$

Eq. D.9 is the governing equation of the non-damped pitching motion of an airfoil in its pure form. In order to use this equation in the CFD solver, eq. D.9 is multiplied on both sides by $\frac{U_\infty^2}{c^2\omega_\alpha^2}$, which is defined as the reduce velocity parameter U^*. This process rescales the dimensionless time variable such that $t^* = t^* \frac{c\omega_\alpha}{U_\infty}$, and also, it dimensionalizes the natural frequency such that $\omega_\alpha = \omega_\alpha^* \frac{U_\infty}{c}$. Eq. 9 is then modified as follow:

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + r_\alpha^2 \alpha = \frac{2}{\pi} \frac{U_\infty^2}{\mu c^2 \omega_\alpha^2} C_M \tag{D.10}$$

By applying the substitution $U^* = \frac{U_\infty}{b\omega_\alpha}$, eq. D.10 becomes D.11, which is the desired non-damped pitching equation for the CFD solver,

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + r_\alpha^2 \alpha = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_M. \tag{D.11}$$

D.2 Pitching Mode with Damping

By applying Newton's law of force summation, the equation of motion for pitching mode with damping factor can be formulated as follow:

$$mcx_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + \Phi_\alpha \dot{\alpha} + k_\alpha \alpha = M. \tag{D.12}$$

Just like in section D.1, eq. D.12 is divided by m and c in order to eliminate these variables from the first term of eq. D.12. Then, it is divided by c and $\frac{U_\infty^2}{c^2}$ in order to non-dimensionalize the chord length and time parameter. Then it is multiplied by $\frac{\frac{\rho_\infty}{2}}{\frac{\rho_\infty}{2}}$ on the right side. This results in the following equation,

$$\frac{x_\alpha c^2}{cU_\infty^2}\ddot{h} + \frac{I_\alpha c^2}{mc^2U_\infty^2}\ddot{\alpha} + \frac{\Phi_\alpha c^2}{mc^2U_\infty^2}\dot{\alpha} + \frac{k_\alpha c^2}{mc^2U_\infty^2}\ddot{\alpha} = \frac{c^2}{mc^2U_\infty^2}M, \tag{D.13}$$

which can be simplified as:

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{\Phi_\alpha c}{mc^2U_\infty}\dot{\alpha}^* + \frac{k_\alpha c^2}{mc^2U_\infty^2}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{mc^2U_\infty^2\frac{\rho_\infty}{2}}M \tag{D.14}$$

Eq. D.14 is modified by applying the definition of reduced frequency, that is, $\omega_\alpha^{2*} = \frac{k_\alpha c^2}{I_\alpha U_\infty^2}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{\Phi_\alpha \omega_\alpha^* I_\alpha^{\frac{1}{2}}}{mc^2k_\alpha^{\frac{1}{2}}}\dot{\alpha}^* + \frac{I_\alpha\omega_\alpha^{2*}}{mc^2}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{mc^2U_\infty^2\frac{\rho_\infty}{2}}M \tag{D.15}$$

Eq. D.15 is further modified by applying the definition of Radius of gyration, that is, $r_\alpha^2 = \frac{I_\alpha}{mc^2}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + \frac{\Phi_\alpha r_\alpha^2 \omega_\alpha^*}{I_\alpha^{\frac{1}{2}}k_\alpha^{\frac{1}{2}}}\dot{\alpha}^* + r_\alpha^2\omega_\alpha^{2*}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{mc^2U_\infty^2\frac{\rho_\infty}{2}}M \tag{D.16}$$

Eq. D.16 is modified by applying the definition of reduced damping coefficient, that is, $\Phi_\alpha^* = \frac{\Omega}{2\sqrt{I_\alpha k_\alpha}}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + 2\Omega^* r_\alpha^2\omega_\alpha^*\dot{\alpha}^* + r_\alpha^2\omega_\alpha^{2*}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{mc^2U_\infty^2\frac{\rho_\infty}{2}}M \tag{D.17}$$

Eq. D.17 is modified by applying the definition of Moment Coefficient, that is, $C_m = \frac{M}{\frac{\rho_\infty}{2}U_\infty^2c^2}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + 2\Phi_\alpha^* r_\alpha^2\omega_\alpha^*\dot{\alpha}^* + r_\alpha^2\omega_\alpha^{2*}\alpha = \frac{c^2\frac{\rho_\infty}{2}}{m}C_M \tag{D.18}$$

Eq. D.18 is then modified by applying the definition of viscosity, that is, $\mu = \frac{m}{\pi\rho_\infty b^2}$.

$$x_\alpha \ddot{h}^* + r_\alpha^2\ddot{\alpha}^* + 2\Phi_\alpha^* r_\alpha^2\omega_\alpha^*\dot{\alpha}^* + r_\alpha^2\omega_\alpha^{2*}\alpha = \frac{2}{\mu\pi}C_M \tag{D.19}$$

Eq. D.19 is the equation of damped pitching motion in its pure form. Just as the non-damped pitch equation in section D.1, eq. D.19 is multiplied by $\frac{U_\infty^2}{c^2\omega_\alpha^2}$ on both sides, so that the dimensionless time is rescaled, and the uncoupled pitch frequency is dimensionalized. The outcome of this procedure is the desired damped pitching equation that's used in the CFD solver, as follow:

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + 2\Phi_\alpha^* r_\alpha^2 \dot{\alpha}^* + r_\alpha^2 \alpha = \frac{2}{\pi}\frac{U^{*2}}{\mu} C_M. \tag{D.20}$$

D.3 Heave Mode without Damping

By applying Newton's law of moment summation, the equation of motion for pitching mode is formulated as follow:

$$m\ddot{h} + mcx_\alpha \ddot{\alpha} + k_h h = L \tag{D.21}$$

In order to eliminate the variable m, eq. D.21 is divided by m.

$$\ddot{h} + cx_\alpha \ddot{\alpha} + \frac{k_h}{m} h = \frac{1}{m} L \tag{D.22}$$

Then, in order to eliminate the variable c, eq. D.22 is divided by c .

$$\frac{1}{c}\ddot{h} + x_\alpha \ddot{\alpha} + \frac{k_h}{mc} h = \frac{1}{mc} L \tag{D.23}$$

The last two steps eliminate the need to non-dimensionalize k_h. Eq. D.23 is then divided by $\frac{U_\infty^2}{c^2}$ in order to non-dimensionalize the time parameter, as follow:

$$\frac{c^2}{cU_\infty^2}\ddot{h} + \frac{x_\alpha c^2}{U_\infty^2}\ddot{\alpha} + \frac{k_h c^2}{mcU_\infty^2} h = \frac{c^2}{mcU_\infty^2} L, \tag{D.24}$$

which can be simplified as:

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{k_h c^2}{m U_\infty^2} h^* = \frac{c^2}{m c U_\infty^2} L. \tag{D.25}$$

Eq. D.25 is modified by multiplying the right side by $\frac{\frac{\rho_\infty}{2}}{\frac{\rho_\infty}{2}}$.

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{k_h c^2}{m U_\infty^2} h^* = \frac{c^2 \frac{\rho_\infty}{2}}{m c U_\infty^2 \frac{\rho_\infty}{2}} L \tag{D.26}$$

Eq. D.26 is then modified by applying the definition of Moment coefficient, that is, $C_L = \frac{L}{\frac{\rho_\infty}{2} U_\infty^2 c}$.

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{k_h c^2}{m U_\infty^2} h^* = \frac{c^2 \frac{\rho_\infty}{2}}{m} C_L \tag{D.27}$$

Eq. D.27 is then modified by applying the definition of viscosity, that is, $\mu = \frac{m}{\pi \rho_\infty b^2}$.

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{k_h c^2}{m U_\infty^2} h^* = \frac{2}{\mu \pi} C_L \tag{D.28}$$

Eq. D.28 is then modified by applying the definition of reduced plunge frequency, that is, $\omega_h^{2*} = \frac{k_h c^2}{m U_\infty^2}$.

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \omega_h^{2*} h^* = \frac{2}{\mu \pi} C_L \tag{D.29}$$

Eq. D.29 is the governing equation of the non-damped plunging motion in its pure form. In order to use eq. D.29 in the CFD solver, eq. D.29 is multiplied by $\frac{U_\infty^2}{c^2 \omega_\alpha^2}$ on both sides, so that the dimensionless time parameter is rescaled, and the uncoupled frequency dimensionalized. The outcome from this procedure is the desired non-damped plunging equation for the CFD solver, as follow:

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{\omega_h^2}{\omega_\alpha^2} h^* = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_L. \tag{D.30}$$

D.4 Heave Mode with Damping

By applying Newton's law of moment summation, the equation of motion for pitching mode with damping can be formulated as follow:

$$m\ddot{h} + mcx_\alpha\ddot{\alpha} + \Phi_h\dot{h} + k_h h = L. \tag{D.31}$$

Just like in section D.3, eq.D.31 is divided by m and c in order to eliminate these variables from the first term of eq. D.12. Then, it is divided by c and $\frac{U_\infty^2}{c^2}$ in order to non-dimensionalized the chord length and time parameter. Then it is multiplied by $\frac{\frac{\rho_\infty}{2}}{\frac{\rho_\infty}{2}}$ on the right side. This results in the following equation,

$$\ddot{h}^* + x_\alpha\ddot{\alpha}^* + \frac{\Phi_h c}{mU_\infty}\dot{h}^* + \frac{k_h c^2}{mU_\infty^2}h^* = \frac{c^2\frac{\rho_\infty}{2}}{mcU_\infty^2\frac{\rho_\infty}{2}}L \tag{D.32}$$

Eq. D.32 is modified by applying the definition of Moment coefficient, that is, $C_L = \frac{L}{\frac{\rho_\infty}{2}U_\infty^2 c}$.

$$\ddot{h}^* + x_\alpha\ddot{\alpha}^* + \frac{\Phi_h c}{mU_\infty}\dot{h}^* + \frac{k_h c^2}{mU_\infty^2}h^* = \frac{c^2\frac{\rho_\infty}{2}}{m}C_L \tag{D.33}$$

Eq. D.33 is then modified by applying the definition of viscosity, that is, $\mu = \frac{m}{\pi\rho_\infty b^2}$.

$$\ddot{h}^* + x_\alpha\ddot{\alpha}^* + \frac{\Phi_h c}{mU_\infty}\dot{h}^* + \frac{k_h c^2}{mU_\infty^2}h^* = \frac{2}{\mu\pi}C_L \tag{D.34}$$

Eq. D.34 is then modified by applying the definition of reduced plunge frequency, that is, $\omega_h^{2*} = \frac{k_h c^2}{mU_\infty^2}$.

$$\ddot{h}^* + x_\alpha\ddot{\alpha}^* + \frac{\Phi_h\omega_h^*}{m^{\frac{1}{2}}k_h^{\frac{1}{2}}}\dot{h}^* + \omega_h^{2*}h^* = \frac{2}{\mu\pi}C_L \tag{D.35}$$

Eq. D.35 is then modified by applying the definition of reduced damping coefficient, that is, $\Phi_h^* = \frac{\Phi}{2\sqrt{mk_h}}$.

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + 2\Phi_h^* \omega_h^* \dot{h}^* + \omega_h^{2*} h^* = \frac{2}{\mu\pi} C_L \tag{D.36}$$

Eq. D.36 is the equation of damped plunging motion in its pure form. In order to use eq.D.36 in the CFD solver, it is multiplied by $\frac{U_\infty^2}{c^2\omega_\alpha^2}$ on both sides, so that the dimensionless time parameter is rescaled, and the uncoupled frequency dimensionalized. The outcome from this procedure is the desired damped plunging equation for the CFD solver, as follow:

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + 2\Phi_h^* \frac{\omega_h}{\omega_\alpha} \dot{h}^* + \frac{\omega_h^2}{\omega_\alpha^2} h^* = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_L. \tag{D.37}$$

D.5 The CFD Structural Equations

Equations D.20 and D.37 are combined to form the governing structural matrix equation, as shown in eq. 4.1. However, the matrix equation that is actually coded in the CFD solver is not exactly the same as eq. 4.1, because the former is actually non-dimensionalized by the mid-chord b, whereas as the latter is non-dimensionalized by the full chord length c. If the mid-chord length b is used, then this would only modify the constant coefficients on the right side of the governing equations, such that equations D.11, D.20, D.30, and D.37 become:

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + r_\alpha^2 \alpha = \frac{1}{\pi} \frac{U_\infty^2}{\mu c^2 \omega_\alpha^2} C_{M,} \tag{D.38}$$

$$x_\alpha \ddot{h}^* + r_\alpha^2 \ddot{\alpha}^* + 2\Phi_\alpha^* r_\alpha^2 \dot{\alpha}^* + r_\alpha^2 \alpha = \frac{1}{\pi} \frac{U^{*2}}{\mu} C_M, \tag{D.39}$$

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + \frac{\omega_h^2}{\omega_\alpha^2} h^* = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_L, \text{ and} \tag{D.40}$$

$$\ddot{h}^* + x_\alpha \ddot{\alpha}^* + 2\Phi_h^* \frac{\omega_h}{\omega_\alpha} \dot{h}^* + \frac{\omega_h^2}{\omega_\alpha^2} h^* = \frac{2}{\pi} \frac{U^{*2}}{\mu} C_L. \tag{D.41}$$

Appendix E

Derivation of Flow and Structural Parameters

In order to simulate a particular type of flow induced vibration of an airfoil, the CFD solver requires the value of certain flow and structural parameters. Certain flow parameters are predetermined for the CFD solver, including the Reynolds Number, Re, Mach number M, and the specific heat ratio γ. These parameters belong to the freestream conditions. Other flow parameters have to be calculated using an appropiate mathematical formula. The formulas for most of these parameters can be found in References [40] and [14]. To begin with, the dimensionless freestream pressure is defined as:

$$P_\infty^* = \frac{M_\infty^2}{\gamma}. \tag{E.1}$$

The ratio of stagnation pressure to static pressure is expressed as:

$$\frac{P_o^*}{P_\infty^*} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma-1}}. \tag{E.2}$$

The stagnation pressure P_o^* can then be calculated from equations E.1 and E.2. The ratio of stagnation to static temperature can be expressed as [40] [14]:

$$\frac{T_o}{T_\infty} = \left(1 + \frac{\gamma - 1}{2} M^2\right). \tag{E.3}$$

Equation E.3 can defined as the dimensionless stagnation temperature, or $T_o^* = \frac{T_o}{T_\infty}$. In all, the dimensionless static pressure, stagnation pressure, and static temperature are flow parameters that are required to run the CFD solver.

Certain structural parameters are predetermined for the CFD solver, such as the unbalance distance x_a, mass ratio μ, and radius of gyration r_a. Other structural parameters have to be calculated using an appropiate mathematical formula. To begin with, the flow viscosity is calculated using Sutherland's law of viscosity, expressed as:

$$\nu = 1.7894 \times 10^5 \left(\frac{T_\infty}{288.16} \right)^{1.5} \left(\frac{288.16 + 110}{T_\infty + 110} \right). \tag{E.4}$$

The dimensional free stream velocity can be calculated from the formula of Reynolds number, that is,

$$U_\infty = \frac{\mathrm{Re}\,\mu}{c\rho}, \tag{E.5}$$

where $\rho = \frac{P}{RT}$. Hence, the reduced freestream velocity U_∞ can be expressed as:

$$U_\infty = \frac{\mathrm{Re}\,\mu R T_\infty}{P_\infty}. \tag{E.6}$$

The value of velocity index VI can also be predetermined for CFD computation. The flutter index serves as a dimensionless scalar quantity that relates flow dynamics to structural dynamics. From a given value of velocity index VI, the reduced velocity U_∞^* can be obtained. The velocity index VI is mathematically expressed as [29]:

$$VI = \frac{U_\infty^*}{\sqrt{\mu}} \tag{E.7}$$

Finally, both the reduced pitch and heave frequencies are defined in pages 83 and 87, and they can be obtained as follow:

$$\omega_\alpha^* = \frac{\omega_\alpha c}{U_\infty}, \tag{E.8}$$

$$\omega_h^* = \frac{\omega_h c}{U_\infty}. \tag{E.9}$$

References

[1] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L., *Aeroelasticity*, New York : Dover Publications, 1996.

[2] Fung, Y. C., *An Introduction to the Theory of Aeroelasticity*, New York: Dover Publications, 1969.

[3] Prananta, B. B., Hounjet, M. H. L., and Zwan, R. J., "Two Dimensional Transonic Aeroelastic Analysis Using Thin-Layer Navier-Stokes Methods," *Journal of Fluids and Structures*, Vol. 12, 1998, pp. 655–676.

[4] Bohbot, J. and Darracq, D., "Time Domain Analysis of Two D.O.F. Airfoil Flutter Usingan Euler Turbulent Navier-Stokes Implicit Solver," *International Forum on Aeroelasticity and Structural Dynamics*, Vol. 5, 1990.

[5] Tijdeman, H. and Seebass, R., "Transonic Flow Past Airfoils," *Annual Review of Fluid Mechanics*, Vol. 12, 1980, pp. 181 – 222.

[6] Tang, L., Bartels, R. E., P., C., and Liu, D. D., "Numerical Investigation of Transonic Limit Cycle Oscillations of a Two-Dimensional Supercritical Wing," *Journal of Fluids and Structures*, Vol. 17, 2003, pp. 29–41.

[7] Edwards, J. W., "Transonic Shock Oscillations and Wing Flutter Calculated with an Interactive Boundary Layer Coupling method," EUROMECH-Colloquium 349, Simulation of Fluid-Structure Interaction in Aeronautics, Gottingen, Germany, 1996.

[8] Bendiksen, O. O., "Transonic Limit Cycle Flutter LCO," AIAA Paper 2004-1694, 2004.

[9] Dietz, G., Schewe, G., and Mai, H., "Experiments on Heave-Pitch Limit Cycle Oscillations of Supercritical Airfoil Close to the Transonic Dip," *Journal of Fluids and Structures*, Vol. 19, 2001, pp. 1–16.

[10] Geissler, W., "Numerical Study of Buffet and Transonic Flutter on the NLR 7301 Airfoil," *Aerospace Science and Technology*, Vol. 7, 2003, pp. 540–550.

[11] Castro, B. M., Jones, K. D., Ekaterinaris, J. A., and Platzer, M. F., "Analysis of the Effect of Porous Wall Interference on Transonic Airfoil Flutter," AIAA Paper 2001-2725, 2001.

[12] Schewe, G., Knifer, A., Mai, H., and Dietz, G., "Experimental and Numerical Investigation of Non-linear Effects in Transonic Flutter," Tech. Rep. DLR IB 232

- 2002 J 01, German Aerospace Center (DLR) Institute of Aeroelasticity (23200) Göttingen, 2002.

[13] Hu, Z., *Parallel Computation of FLuid-Structural Interactions Using High Resolution Upwind schemes*, Ph.D. thesis, Dept. of Mechanical and Aerospace Engineering, University of Miami, May 2005.

[14] Anderson, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill, 2001.

[15] Blazek, J., *Computational Fluid Dynamics: Principles and Applications*, Elsevier, 2001.

[16] Hoffman, K. A. Chiang, S. T., *Computational Fluid Dynamics: Volume I*, EES-books, 2004.

[17] Chen, X. and Zha, G., "Fully Coupled Fluid-Structural Interactions Using an Efficient High Resolution Upwind Scheme," *Journal of Fluids and Structures*, Vol. 20, 2005, pp. 1105–1125.

[18] Chen, X. and Zha, G., "Numerical Simulation of 3 D Wing Flutter with Fully Coupled Fluid-Structural Interaction," AIAA Paper 2006-697, 2006.

[19] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Taylor & Francis, 1997.

[20] Zha, G., "A Low Diffusion E-CUSP Upwind Scheme for Transonic Flows," AIAA Paper 2004-2707, 2004.

[21] Baldwin, B. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Seperated Turbulent flows," *AIAA Paper 78-257*, 1978.

[22] Pulliam, T. H. and Steger, J. L., "Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms," *AIAA Paper*, 1985, pp. 85–360.

[23] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Computational Physics*, Vol. 43, 1981, pp. 357–372.

[24] Wang, B. and Zha, G., "Comparison of a Low Diffusion E-CUSP and the Roe Scheme for RANS Calculation," AIAA Paper 2008-0596, 2008.

[25] Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, 1983, pp. 327–355.

[26] Jameson, A., "Artificial Diffusion, Upwind Biasing, Limiters and their Effect on Accuracy and Multigrid Convergence in Transonic and Hypersonic Flow," *AIAA Paper 93-3559*, 1993.

[27] Jameson, A., "Positive Schemes and Schock Modelling for Compressible Flow," *Int. J. Numerical Methods in Fluids*, Vol. 20, 1995, pp. 743–776.

[28] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics II: Artificial Diffusion and Discrete Schock Structures," *Int. J. Computational Fluid Dynamics*, Vol. 5, 1995, pp. 1–38.

[29] Chen, X., Zha, G., and Hu, Z., "Numerical Simulation of Flow Induced Vibration Based on Fully Coupled Fluid structural Interations," AIAA Paper 2004-2240, 2004.

[30] Zha, G., Shen, Y., and Wang, B., "Calculation of Transonic Flows Using WENO Method with a Low Diffusion E-CUSP Upwind Scheme," AIAA-2008-745, Jan. 2008.

[31] Knupp, P. M. and Steinberg, S., *Fundamentals of Grid Generation*, CRC Press, 1994.

[32] Thompson, J. F., Soni, B. K., and Weatherill, N. P., *Handbook of Grid Generation*, CRC Press, 1999.

[33] Davis, S. S., "NACA 64A010 Oscillatory Pitching," *Compendium of Unsteady Aerodynamics Measurements*, 1982.

[34] Zwaan, R. J., "NLR 7301 Supercritical Airfoil Oscillatory Pitching and Oscillatory Flap," *Compendium of Unsteady Aerodynamic Measurements*, 1982.

[35] McMullen, M., Jameson, A., and Alonso, J., "Application of Non-Linear Frequency Domain Solver to the Euler and Navier-Stokes Equations," *AIAA Paper-2002-0120*, 2002.

[36] Isogai, K., "Numerical Study of Transonic Flutter of A Two-dimensional Airfoil," *AIAA Journal*, Vol. 15, 1977, pp. 1728–1735.

[37] Isogai, K., "Transonic Dip Mechanism of Flutter of a Sweptback Wing: Part II," Tech. Rep. Technial Report 617, National Aerospace Laboratory of Japan, 1980.

[38] Lahey Computer Systems, Inc., *SSL II User Manual (Scientific Subroutine Library)*.

[39] Budynas, R. G., *Advanced Strength and Applied Stress Analysis*, McGraw-Hill, 2nd ed., 1999.

[40] Anderson, J. D., *Introduction to Flight*, McGraw-Hill, 2005.