



Cape Peninsula
University of Technology

Faculty of Engineering

Department of Mechanical Engineering

Smart Alignment System Research Group

**A Novel Approach to Control the Shape Change of a
Reconfigurable Wing Using Shape Memory Alloy**

by

Zhe Xing

**Thesis submitted towards partial fulfilment of the Masters Degree in
Technology: Mechanical Engineering (MTechME)**

Under the supervision of

Dr. Oscar Philander (DTech, CPUT Bellville Campus)

CAPE TOWN, 2007

Abstract

Engineers and Technologists have found several approaches to control the shape of an aerofoil and improve the performance of a wing at different flow regimes; this research has been done at 2D level. In this work, a novel approach has been developed. The inspiration for this work comes from biological research. A 3D wing body has been modeled and flow conditions around it were simulated by advanced computer technology. The fabrication of the wing, based on the design optimization model, has been conducted using rapid prototyping technology. The unique thermal and mechanical properties that are exhibited by shape memory alloys (SMAs) have presented an exciting design possibility in the field of aerospace engineering. This kind of smart material was incorporated in the wing structure and when activated can alter the shape of the wing, thus effectively increasing the efficiency of a wing in flight, at several different flow regimes.

Acknowledgements

Firstly I would like to thank Dr. O. Philander my supervisor for introducing this exciting field of study to me. He not only provided a high level of technology regarding Shape Memory Alloys, but also provided me with many chances to experiment myself. Dr. Philander greatly assisted me in adapting to a culture difference.

The financial assistance given to me by the Cape Peninsula University of Technology and the Department of Mechanical Engineering is through the process of my study.

I would like to thank the members of the SMART alignment research group; we assisted each other during the research process and shared in the success together.

Prof. G. Oliver's excellent knowledge of programming languages, Continuum Mechanics and the Finite Element Method has always been an inspiration to

me to finish this research.

Prof. B. Sun's encyclopedically supporting is very important to me and other Chinese students.

To the lecturers in the Department of Mechanical Engineering especially Ms. R. Ziegler, Mr. M. N. Riddles for your undying support me.

A special word of thanks to my parents Jikun Xing and Yeping Na, my little sister Na Xing. Your love and support is energy for my life.

Table of Contents

Chapter 1	Introduction	1
1.1.	Problem Statement	1
1.2.	Objectives	4
1.3.	Background.....	5
1.3.1.	Smart Materials.....	5
1.3.2.	Wing Design	9
1.3.3.	Finite Element Method.....	12
1.3.4.	Rapid Prototype and Vacuum Casting	13
1.4.	Literature Review.....	15
1.4.1.	Review of Smart Materials.....	15
1.4.2.	Review of Smart Structure.....	19
1.5.	The Inspiration Come From Biology.....	24
1.6.	Scope of the Thesis	25
Chapter 2	Airfoil Design	26
2.1.	Introduction.....	26
2.2.	Kutta Condition	27
2.3.	Kutta-Joukowski Theorem.....	27
2.4.	Circulation about a body in a flow field.....	28

2.5.	The Airfoil Geometrical Variables	30
2.6.	The Method of Airfoil Design.....	33
2.7.	Introduction of Airfoil Vortex Sheet Models.....	35
2.8.	Biot-Savart Law.....	36
2.9.	Algorithm of the Vortex Panel Method.....	38
2.10.	Aerodynamic Properties of a Finite Wing.....	50
Chapter 3 Conceptualization and FE Simulation		54
3.1.	<i>The MSC.Nastran Analysis for the Wing Design.....</i>	<i>54</i>
3.1.1.	The Structure of the Wing.....	55
3.1.2.	The Comparison	57
3.1.3.	Wing modeling.....	66
3.2.	<i>The MSC.Dytran Analysis for the Wing Design.....</i>	<i>70</i>
Chapter 4 The Manufacture of the Conformable Wing		76
4.1.	The LOM Process.....	76
4.2.	Vacuum Casting Procedure	79
Chapter 5 Conclusions.....		86
Appendix A Python codes used for get the shape information NACA series aerofoil		88
Appendix B Python codes used for get pressure coefficient of NACA series aerofoil and Boeing F aerofoil		97
Appendix C Python codes used for get the lift coefficient and induce drag coefficient for finite wing model.....		102

Appendix D Some graphics come from MSC.Dytran Analyses	
.....	111
Reference	
.....	115

List of Figures

Figure 1.1: Direction of flow at the wing tips due to pressure differences on the upper and lower surfaces of airplane wings [5].....	2
Figure 1.2: The tip vortex [5].....	2
Figure 1.3: The information of NACA series airfoil in python program	12
Figure 1.4: Austenite Phase [2].....	16
Figure 1.5: Martensite Phase [2].....	16
Figure 1.6: Transformation Cycle of A SMA [2]	17
Figure 1.7: Temperature-induced phase transformation between austenite and twinned Martensite in the absence of an applied load. [2].....	18
Figure 1.8: Hysteretic relation between austenite and.....	18
Figure 1.9: Stress-induced phase transformation and	18
Figure 1.10: Quasiplastic behavior and residual strain generated when $T < M_f$ [2]	19
Figure 1.11: Experiment using model [9]	20
Figure 1.12: One Wind Tunnel Models Come From	21
Figure 1.13: Some Other Design [9]	21
Figure 1.14: F-15 [10].....	22
Figure 1.15: F-15 and its smart structure [10].....	22
Figure 1.16: A morphing wing [12].....	23
Figure 1.17: Hinge less shape memory alloy Flap [12].....	23
Figure 1.18: Reconfigurable wing from the Technical University of Berlin [13].	23
Figure 1.19: The birds' wing.....	24
Figure 1.20: The most efficient wing shape [1]	25
Figure 2.1: Flow around an airfoil with zero circulation. [11]	27
Figure 2.2: Flow past an airfoil in a real fluid. [11].....	27
Figure 2.3: Kutta condition [5].....	28

Figure 2.4: Vortex behavior [5].....	29
Figure 2.5: 2-D Vortex behavior [5].....	29
Figure 2.6: Airfoil Geometrical Variables [11].....	31
Figure 2.7: The pressure distribution [11].....	32
Figure 2.8: The streamlines of the flow around an airfoil	32
Figure 2.9: Approximate the vortex sheet by a series of straight panels. [11]	34
Figure 2.10: Replacement of an Airfoil by Vortex Panels of Linearly Varying Strength. [11]	35
Figure 2.11: The vortex sheet on the airfoil surface [11].....	35
Figure 2.12: The sheet strength distribution [5]	36
Figure 2.13: The velocity increment du_θ induced by vortex element ds . [11]	37
Figure 2.14: Strength Distribution around the Arbitrary Shaped Airfoil [5]... 39	39
Figure 2.15: Replacement of an Airfoil by Vortex Panels of Linearly Varying Strength.....	39
Figure 2.16: The reference of the pressure coefficient [11].....	47
Figure 2.17: The shape of the airfoil	48
Figure 2.18: The comparison of the pressure coefficient.....	49
Figure 2.19: Pressure Distribution for Boeing F Aerofoil.....	49
Figure 2.20: The dimension of the finite wing model.....	50
Figure 2.21: Lift Coefficient.....	52
Figure 2.22 Lift Coefficient and Induce drag Coefficient.....	53
Figure 3.1: The Wing Models.....	55
Figure 3.2: The Wings' Model's analysis.....	57
Figure 3.3: The wing_40's deformation.....	59
Figure 3.4: The wing_60's deformation.....	59
Figure 3.5: The wing_70's deformation.....	62
Figure 3.6: The wing_80's deformation.....	62
Figure 3.7: The wing_0's deformation.....	65
Figure 3.8: The wing_3's deformation.....	65

Figure 3.9: Unexploded view for structure	67
Figure 3.10: Explode view for the wing structure	67
Figure 3.11: Comparison 1 (between the straight wing and the bending wing)	68
Figure 3.12: Comparison 2 (between the straight wing and the bending wing)	69
Figure 3.13: Comparison 3 (between the straight wing and the bending wing)	69
Figure 3.14: Some Parameters for MSC.Dytran simulation	70
Figure 3.15: The Insight View of Element Velocity Field	71
Figure 3.16: Comparison 1 of air flow speed between straight wing and bending wing (The wing in Figure 3.11).....	72
Figure 3.17: Comparison 2 of air flow speed between straight wing and bending wing (The wing in Figure 3.12).....	73
Figure 3.18: Comparison 3 of air flow speed between straight wing and bending wing (The wing in Figure 3.13).....	74
Figure 4.1: LOM machine	77
Figure 4.2: A typical LOM machine [19]	78
Figure 4.3: The procedure of RP in LOM [19].....	78
Figure 4.4: The final models of the wing parts	79
Figure 4.5: Fixing the master model in the casting frame	80
Figure 4.6: Mixing, degassing and pouring the silicone rubber [7].....	80
Figure 4.7: Vacuum casting machine.....	81
Figure 4.8: Opening the mold	82
Figure 4.9: Casting and heating the mold [7].....	82
Figure 4.10: Casting a part [7]	82
Figure 4.11: The finished part	83
Figure 4.12: The assembling	85

List of Tables

Table 2.1: The data comparison	48
Table 2.2: The detail information for the wing model	52
Table 2.3: Lift Coefficient in different AOA	52
Table 2.4: Lift Coefficient and Drag Coefficient	53
Table 3.1: Wing_40 analyse.....	58
Table 3.2: Wing_60 analyse.....	60
Table 3.3: Wing_70 analyse.....	61
Table 3.4: Wing_80 analyse.....	63
Table 3.5: Wing_0 analyse.....	64
Table 3.6: Wing_3 analyse.....	66
Table 4.1: The technical overview.....	84

Chapter 1 Introduction

1.1. Problem Statement

Aircraft design involving fixed wing structures such as those found on all commercial airplanes often involves a sacrifice in design point efficiency to account for off-design flight conditions. Most airplane wings are designed to perform optimally at cruising speeds [1]. These off-design flight conditions include take off, ascending, descending, and landing. The take-off condition usually draws the highest power from the engines to get the airplane airborne. The velocity of airflow around the subsonic airplane wings is less than 300km/h and thus ideal gas conditions exist for these regimes. Conventional airplanes use hydraulic actuation systems to operate flaps during take-off conditions to increase the surface area and angle of attack of the airplane wing to increase lift. During conventional flight of all aircraft, high pressure is produced on the lower surface of wings while low pressure is produced on its upper surface. This produces a condition where the flow would always want to move from the lower wing surface to the upper wing surface at the wing tip (see Fig. 1.1). This condition produces huge vortices at the wing tips and thus reduces the efficiency of the entire aircraft (see Fig. 1.2).

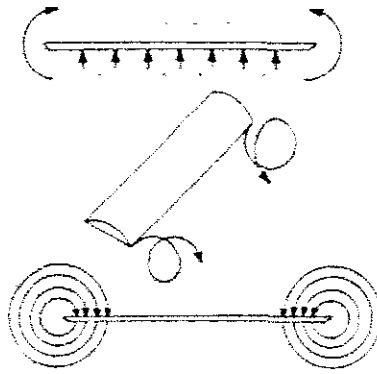


Figure 1.1: Direction of flow at the wing tips due to pressure differences on the upper and lower surfaces of airplane wings [5]

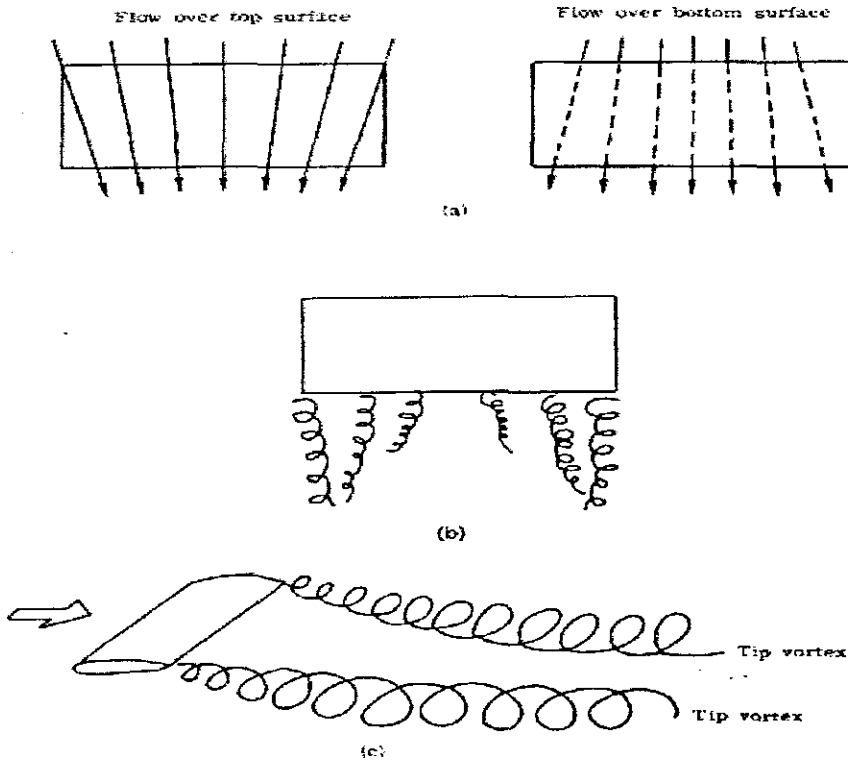


Figure 1.2: The tip vortex [5]

In recent years, various approaches have been employed to use smart materials in airplane wings to make them reconfigurable and thus improve their performance for different flight conditions, i.e., take-off, ascending, etc, and to reduce the magnitude of the vortices formed at the wing tips of conventional

aircraft during flight.

Advances in the technology of smart materials have provided the opportunity to make this idea a reality. Previous investigations into conformable wings have produced favorable results [1]. These investigations have however only focused on changing the shape of the aerofoil to suit different flight conditions. The question that this research is trying to solve is simple: "Is there a novel approach to change the entire configuration of a three-dimensional airplane wing to make it more efficient for the different flight conditions?" The answer to this question is the primary focus of this thesis.

The unusual characteristics of smart materials, such as high strength to weight ratios, the ability to change their shape, size, mechanical and thermal properties to name a few, have made these materials attractive for use in military, aerospace, medical, automotive and commercial applications. This has been the trend from the beginning of twenty first century. The engineering use of these materials includes vibration control or damping and transformation or morphology of structures.

This research focuses on the conceptualization, design and the manufacture process of a model of a reconfigurable or morphing airplane wing. The shape memory alloys embedded into the airplane wing could use their shape memory ability to produce actuation for the morphing. Unfortunately there are no resources available for testing the model, but even if there were it would constitute a major effort beyond the scope of this work.

The shape memory effect exhibited by shape memory alloys consists out of three material behaviors, i.e. a load induced martensitic phase transformation also known as quasi-plasticity that produces an actuation stroke, a temperature induced phase transformation from a low temperature martensitic phase structure to a high temperature austenitic phase structure that produces the

actuation stroke, and another temperature induced phase transformation from austenite to martensite during cooling that brings the material to its original configuration.

1.2. Objectives

As stated previously, the main aim of this research is to conceptualize, design investigate and adopt a suitable process for the manufacture of the model of a reconfigurable or morphing airplane wing. Morphing to be accomplished by embedded SMA's. It was decided that the initial investigation into finding an answer to this question was to develop a scale model of conventional straight wing that will typically be suited for an unmanned aircraft.

An investigation into wing design inspired by biological research will first be conducted. It is well known that biological creatures such as birds and fish use their wings and fins respectively in the most efficient ways to move around. The concepts for conformable wings discussed in this thesis will thus be based on biological creatures.

For this study a commercial airplane wing shape, based on the Boeing747-400 airplane wing will be used. Coupled to this, the wing will use the supercritical airfoil shape. This airfoil and wing shape is used by most modern airplane manufacturers. In order to test the performance of the airfoil shape for low speed flight conditions and various angles of attack an investigation of wing design will be conducted. A python program based on an analytical method called the Vortex Panel method will be developed to determine the airfoil's flight characteristics at low speed conditions. The program using the Vortex Panel method is a powerful design tool that will be able to determine pressure distributions for any arbitrary shape placed in a two-dimensional flow field. The development of this program will form a critical part in the conformable wing research project.

The results of this investigation will then be used to develop the geometry for

different concepts for conformable wings. Models will then be created using 3D modeling software and imported into a finite element package called MSC.Patran and MSC.Nastran to perform simulations on the deformation behavior of the different concepts. The concepts that produce the most favorable deformations, i.e. displacements and rotations, will then be used in flow simulations using MSC.Dytran to determine velocity and pressure distributions of the different conformable wings.

The concept for the conformable wing that produces the best lift will then be modeled using laminated object manufacturing (LOM) prototyping technology. This process will produce “masters” that will be used in the vacuum casting process manufacture the components, that when assembled will produce the final model of the morphing wing.

1.3. Background

1.3.1. Smart Materials

According to recent surveys, scientists have made amazing developments in the design of electronics and machinery using quite unusual materials. These kinds of special materials have properties that scientists can easily manipulate. Some of these materials have the ability to change shape or size simply by adding a little bit of heat, or to change from a liquid to a solid almost instantly when placed near a magnetic field; these materials are called smart materials.

Smart materials have one or more properties that can be dramatically altered. Most everyday materials have physical properties, which cannot be significantly altered, whereas a smart material with variable viscosity may turn from a fluid which flows easily to a solid. Varieties of smart materials already exist, and are being researched extensively. These include piezoelectric materials, magneto-rheostatic materials, electro-rheostatic materials, and shape memory alloys. Some everyday items are already incorporating smart materials and the number

of applications for them is growing steadily [2].

Each individual type of smart material has a different property which can be significantly altered, such as viscosity, volume, and conductivity. The property that can be altered influences what types of applications the smart material can be used for.

● **Piezoelectric Materials**

Piezoelectric materials have two unique properties which are interrelated. When a piezoelectric material is deformed, it gives off a small but measurable electrical discharge. Alternately, when an electrical current is passed through a piezoelectric material it experiences a significant increase in size (Up to a 4% change in volume).

Piezoelectric materials are most widely used as sensors in different environments. They are often used to measure fluid compositions, fluid density, fluid viscosity, or the force of an impact. An example of a piezoelectric material in everyday life is the airbag sensor in your car. The material senses the force of an impact on the car and sends an electric charge deploying the airbag [2].

● **Electro-rheostatic and Magneto-rheostatic**

Electro-rheostatic (ER) and magneto-rheostatic (MR) materials are fluids, which can experience a dramatic change in their viscosity. These fluids can change from a thick fluid to nearly a solid substance within the span of a millisecond when exposed to a magnetic or electric field; the effect can be completely reversed just as quickly when the field is removed. MR fluids experience a viscosity change when exposed to a magnetic field, while ER fluids experience similar changes in an electric field. The composition of each type of smart fluid varies widely. The most common form of MR fluid consists of tiny iron particles suspended in oil, while ER fluids can be as simple as milk chocolate or cornstarch and oil.

MR fluids are being developed for use in car shocks, damping washing machine vibration, prosthetic limbs, exercise equipment, and surface polishing of machine parts. ER fluids have mainly been developed for use in clutches and valves, as well as engine mounts designed to reduce noise and vibration in vehicles [2].

- **Electrostrictive Ceramics**

In the majority of current transducer designs the active element is a ferroelectric or piezoelectric ceramic from the lead zirconate titanate (PZT) family of compositions. These materials have a large strain stress product and are relatively robust but, when operating at high fields, they exhibit hysteresis in the strain versus field ($S = dE$) and the polarization versus field ($D = \epsilon^T E$) relationships, which can complicate the control aspect of the design.

- **Shape Memory Alloys**

Shape memory alloys (SMA's) are metals, which exhibit two very unique properties, pseudo-elasticity, and the shape memory effect. Arne Olander [2] first observed these unusual properties in 1938, but not until the 1960's were any serious research advances made in the field of shape memory alloys. The most effective and widely used alloys include NiTi (Nickel - Titanium), CuZnAl, CuAlNi, and AuCd.

The use of NiTi as a biomaterial is fascinating because of its superelasticity and shape memory effect, which are completely new properties compared to the conventional metal alloys.

The two unique properties described above are made possible through a solid state phase change, which is a molecular rearrangement, which occurs in the shape memory alloy. Typically when one thinks of a phase change a solid to liquid or liquid to gas change is the first idea that comes to mind. A solid state phase change is similar in that a molecular rearrangement is occurring, but the

molecules remain closely packed so that the substance remains a solid. In most shape memory alloys, a temperature change of only about 10 is necessary to initiate this phase change. The two phases, which occur in shape memory alloys, are Martensite, and Austenite.

Martensite is the relatively soft and easily deformed phase of shape memory alloys, which exists at lower temperatures. The molecular structure in this phase is twinned. Austenite, the stronger phase of shape memory alloys, occurs at higher temperatures. The shape of the Austenite structure is cubic; the undeformed Martensite phase is the same size and shape as the cubic Austenite phase on a macroscopic scale, so that no change in size or shape is visible in shape memory alloys until the Martensite is deformed [3].

When Martensite NiTi is heated, it begins to change into austenite. The temperature at which this phenomenon starts is called austenite start temperature (A_s). The temperature at which this phenomenon is complete is called austenite finish temperature (A_f). When austenite NiTi is cooled, it begins to change onto Martensite. The temperature at which this phenomenon starts is called Martensite start temperature (M_s). The temperature at which Martensite is again completely reverted is called Martensite finish temperature (M_f)

The shape memory effect is observed when the temperature of a piece of shape memory alloy is cooled to below the temperature M_f . At this stage the alloy is completely composed of Martensite which can be easily deformed. After distorting the SMA the original shape can be recovered simply by heating the wire above the temperature A_f . The heat transferred to the wire is the power driving the molecular rearrangement of the alloy, similar to heat melting ice into water, but the alloy remains solid. The deformed Martensite is now transformed to the cubic Austenite phase, which is configured in the original shape of the wire.

Some of the main advantages of shape memory alloys include:

- Bio-compatibility
- Diverse Fields of Application
- Good Mechanical Properties (strong, corrosion resistant)

There are still some difficulties with shape memory alloys that must be overcome before they can live up to their full potential [2]. These alloys are still relatively expensive to manufacture and machine compared to other materials such as steel and aluminum. Most SMA's have poor fatigue properties.

1.3.2. Wing Design

Aircraft design involving fixed wing structures often involves a sacrifice in design point efficiency to account for off-design flight conditions. These compromises can be eliminated with the implementation of a wing that can adapt to its current flow condition. One factor that greatly influences a wing's efficiency at a particular flow regime is the shape of its cross-section, or airfoil. Some airfoils are aerodynamically efficient at subsonic speeds, but typically create undesirable shock waves early in the transonic regime. Other airfoils are tailored to minimize wave drag near the speed of sound, but do not produce favorable pressure distributions at low subsonic speeds. The process of airfoil design proceeds from a knowledge of the boundary layer properties and the relation between geometry and pressure distribution. The goal of an airfoil design varies. Some airfoils are designed to produce low drag (and may not be required to generate lift at all.) Some sections may need to produce low drag while producing a given amount of lift. In some cases, the drag doesn't really matter; it is maximum lift that is important. The section may be required to achieve this performance with a constraint on thickness, or pitching moment, or off-design performance, or other unusual constraints. Some of these are discussed further in the section on historical examples.

There are two kinds of way to design an airfoil. One approach is to use an airfoil that was already designed by someone who knew what he or she was doing.

This "design by authority" works well when the goals of a particular design problem happen to coincide with the goals of the original airfoil design [4]. This is rarely the case; although sometimes existing airfoils are good enough. The advantage to this approach is that there are test data available. No surprises, such as an unexpected early stall, are likely. On the other hand, available tools are now sufficiently refined that one can be reasonably sure that the predicted performance can be achieved. The use of "designer airfoils" specifically tailored to the needs of a given project is now very common. Airfoils are typically designed with a primary flight condition in mind, for example cruise. These airfoils are generally sub-optimal for the design point because they are fixed and must be used at takeoff, landing, climb, and descent in addition to the design point condition. It would be desirable to have an airfoil with the ability to adapt to its current flow regime and alter its shape to remain efficient at any speed.

The National Advisory Committee for Aeronautics (NACA) developed NACA series airfoil; its design was rather arbitrary with nothing to guide the designer except past experience with known shapes and experimentation with modifications to those shapes. This methodology began to change in the early 1930s with the publishing of a NACA report entitled "The Characteristics of 78 Related Airfoil Sections from Tests in the Variable Density Wind Tunnel" [4]. In this landmark report [1], the authors noted that there were many similarities between the airfoils that were most successful, and the two primary variables that affect those shapes are the slopes of the airfoil mean camber line and the thickness distribution above and below this line. They then presented a series of equations incorporating these two variables that could be used to generate an entire family of related airfoil shapes. As airfoil design became more sophisticated, this basic approach was modified to include additional variables, but these two basic geometrical values remained at the heart of all NACA airfoil series.

For instance, the first family of airfoils designed using this approach became

known as the NACA Four-Digit Series. The first digit specifies the maximum camber (m) in percentage of the chord (airfoil length), the second indicates the position of the maximum camber (p) in tenths of chord, and the last two numbers provide the maximum thickness (t) of the airfoil in percentage of chord. For example, the NACA 2415 airfoil has a maximum thickness of 15% with a camber of 2% located 40% back from the airfoil leading edge (or $0.4c$). Utilizing these m , p , and t values, we can compute the coordinates for an entire airfoil using the following relationships:

- 1 Pick values of x from 0 to the maximum chord c .
- 2 Compute the mean camber line coordinates by plugging the values of m and p into the equations (1.1) for each of the x coordinates.

$$Y_c = \frac{m}{p^2}(2px - x^2) \quad \text{from } x = 0 \text{ to } x = p \quad (1.1)$$

$$Y_c = \frac{m}{(1-p^2)}[(1-2p) + 2px - x^2] \quad \text{from } x = p \text{ to } x = c$$

- 3 Calculate the thickness distribution above (+) and below (-) the mean line by plugging the value of t into the equation (1.2) for each of the x coordinates.

$$Y_t = \frac{t}{0.2}(0.2969\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4) \quad (1.2)$$

- 4 Determine the final coordinates for the airfoil upper surface (x_u, y_u) and lower surface (x_l, y_l) using the following relationships.

$$\begin{aligned} x_u &= x - y_t \sin \theta \\ y_u &= y_c - y_t \cos \theta \\ x_l &= x + y_t \sin \theta \\ y_l &= y_c + y_t \cos \theta \end{aligned} \quad (1.3)$$

For getting the information of the shape of the aerofoil, the Python code had

been used to write the interface shown as figure 1.3. The detail information of this framework can be found in appendix A.

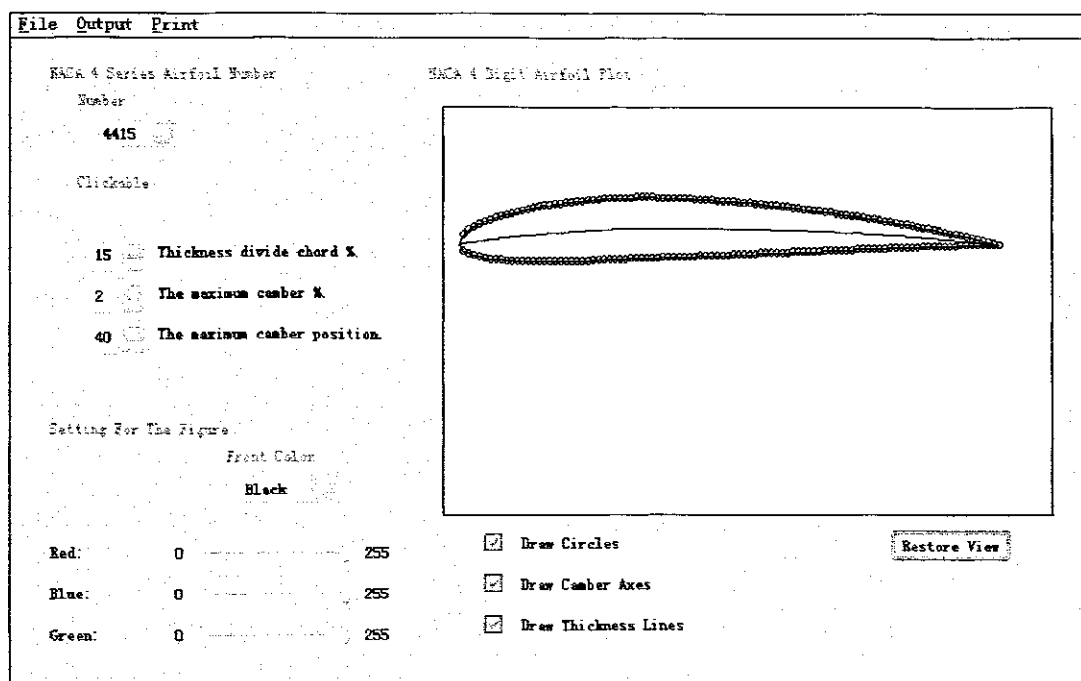


Figure 1.3: The information of NACA series airfoil in python program

1.3.3. Finite Element Method

The finite element method is a numerical procedure for solving the differential equations of physics and engineering. The method had its birth in the aerospace industry in the early 1950s and was first presented in the publication by Turner, Clough, Martin, and Topp (1956) [5]. This publication stimulated other researches and resulted in several technical articles that discussed the application of the method to structural and solid mechanics. An important theoretical contribution was made in 1963 when Melosh [5] illustrated shown that the finite element method was really a variation of the well known Raleigh-Ritz procedure. In structural problems, the method produces a set of linear equilibrium equations by minimizing the potential energy of the system.

The connecting of the finite element method with a minimization procedure quickly led to its use in other engineering areas. The method was applied to problems governed by the Laplace or the Poisson equations because these equations are closely related to the minimization of a functional. The first publications showed the application of the finite element method to conduction heat transfer. The application to fluid mechanics, particularly to the flow through a porous media, followed immediately.

The range of applications for the finite element method was enlarged when other investigations showed that the element equations related to structural mechanics, heat transfer, and fluid mechanics could also be derived by using a weighted residual procedure such as Galerkin's method [5] or the least-square approach. This knowledge is a very important contribution to the theory because it allows the finite element method (FEM) to be applied to any differential equation. The more general theoretical basis eliminates the need for a functional formulation of the physical problem.

The widespread use of personal computers, which have the power to solve problems solvable in the past only on mainframe computers, has influenced the tabulated format of this book. Computer programs for structural analysis, employing techniques such as the finite element method, are also available for general use. These programs are very powerful; however, in many cases, elements of structural systems can be analyzed quite effectively independently without the need for an elaborate finite element model.

1.3.4. Rapid Prototype and Vacuum Casting

The term rapid prototyping (RP) refers to a class of technologies that can automatically construct physical models from Computer-Aided Design (CAD) data. These "three dimensional printers" allow designers to quickly create tangible prototypes of their designs, rather than just two-dimensional pictures.

Such models have numerous uses. They make excellent visual aids for communicating ideas with co-workers or customers. In addition, prototypes can be used for design testing. For example, an aerospace engineer might mount a model airfoil in a wind tunnel to measure lift and drag forces. Designers have always utilized prototypes; rapid prototyping (RP) allows us to be made faster and less expensively [6].

Vacuum Casting is the most widely used process for replicating Rapid Prototyping models. Vacuum casting stands for a duplication technique which has been especially developed to produce small series of functioning prototypes and patterns of plastics. In this technique, the forms are duplicated upon a silicone rubber mould making use of a vacuum chamber that avoids air inclusions in the mould and the work piece.

Manufacture is based on the prototype. After having defined the parting plane and the dead head, the part is fixed inside a frame. Afterwards, this frame is filled with silicone rubber and heated under vacuum. This way, the air emits out of the silicone and the mould gets as strong as necessary.

Having removed the master model, the mould is closed again and filled with the liquid material (under vacuum). The casting resins used in vacuum casting is 2 component- polyurethane or epoxy resin systems. In initial state, both components are available in liquid state, but at different viscosity. The product, which emerges in the mould after curing, is represented by a duroplastic of different characteristics. After having cured the plastic, the generated parts are removed from the mould and finished. Afterwards, the mould can be used for further moldings [7]. The technique can be made more efficient by molding several parts within one mould (multiple moulds).

1.4. Literature Review

1.4.1. Review of Smart Materials

Shape memory alloys (SMA) are set apart from other metals by their unique crystalline characteristics. They exhibit one of two crystalline lattice structures depending on the applied stress and temperature. The low temperature phase, Martensite, is a simple monoclinic arrangement and the high temperature phase, austenite, is a body centered cubic arrangement. The material properties of SMA are dependant on the amount of each phase present at a given time.

The Martensitic phase of an SMA can have a lattice arrangement in one of 24 variants depending on the local stress field, and can appear as twinned or detwinned [8]. In its Martensitic phase, SMA can be deformed easily as these variants interchange to those that align more favorably with the local stress field. Heating the SMA will force a return to its austenitic phase and bring the SMA back to its original macroscopic shape. The recovered strain is called the transformation strain. Two-way shape memory effect can be achieved by inducing a plastic deformation in the material such that a transformation into Martensite results in a macroscopic shape change without an applied stress. This is called two-way shape memory effect because macroscopic shape change occurs in both the heating and cooling directions, even in the absence of external stresses.

Because they are both commonly attainable and easy to implement as linear actuators, SMA wires will be used in this research. Two-way trained SMA wire actuators are attached to points on the inside of an airfoil after they have been cooled below the Martensitic finish temperature. When the wires are heated above the austenitic start temperature, they begin to contract, recovering the transformation strain and deforming the airfoil. Upon cooling below the Martensitic finish temperature, the SMA actuators are brought back to their

original configuration, returning the airfoil to its original shape.

Shape memory alloys undergo a solid-state phase change once they have reached their transformation temperature. This type of metal has two main phases associated with the shape memory effect, austenite and Martensite. Austenite, shown in Fig 1.4, is the high temperature phase where the alloy has a cubic crystal structure. Martensite, the low temperature phase, has a monoclinic crystal structure as shown in Fig 1.5.

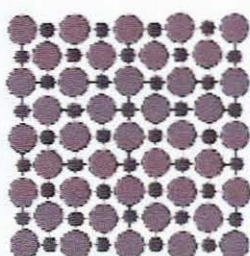


Figure 1.4: Austenite Phase [2]

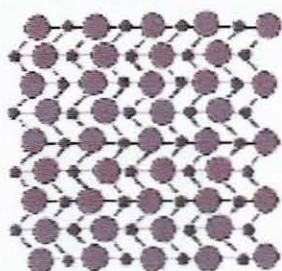


Figure 1.5: Martensite Phase [2]

SMAs are able to return to either phase by either gaining or losing heat. The specific transformation temperature varies depending upon the exact chemical composition [2]. For example, slight variances in Nitinol's composition, which is made from approximately equal amounts of Nickel and Titanium, can cause the transformation temperatures to vary from below 0°C to above 100°C .

The shape memory effect in SMAs comes from the phase change in the alloy's atomic structure. As the material cools to form the Martensite phase it creates

alternating bands or layers. Each layer is tilted in an opposite direction from the layer above and below it. This layer opposition creates a balance in the overall structure to where there is no shape change from austenite to Martensite. In order to return to austenite the crystal structure uses thermal energy to unscrew its atomic formation. This happens because, atomically, lower energy is preferred. At a certain temperature the cubic arrangement of Austenite has a lower energy than the monoclinic arrangement of Martensite. At this point the structure reverts to the Austenite crystal formation.

The different crystal structures of shape memory alloys allows for a rare thermo - elastic Martensitic transformation. The tilted layers in the Martensite phase allow for the material to undergo a shape change from an applied stress. This type of deformation does not damage the structure of the material. The atomic structure only allows the stress to tilt the layers in the same direction. Because the layers can only move in one direction, the material is able to remember its previous crystal formation. When heated the material springs back to its austenite phase causing it to return to its original shape. This cycle is shown in Fig 1.6.

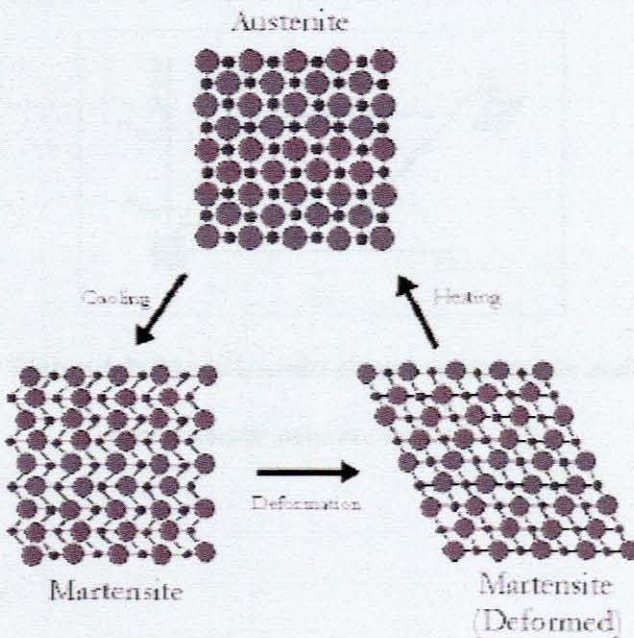


Figure 1.6: Transformation Cycle of A SMA [2]

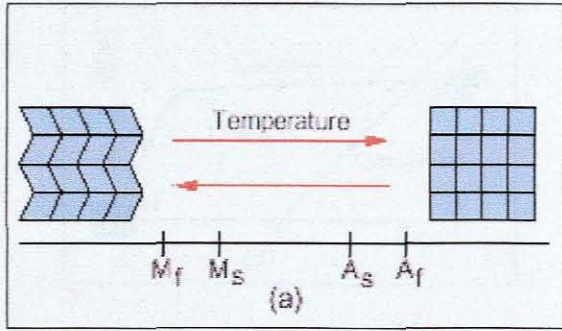


Figure 1.7: Temperature-induced phase transformation between austenite and twinned Martensite in the absence of an applied load. [2]

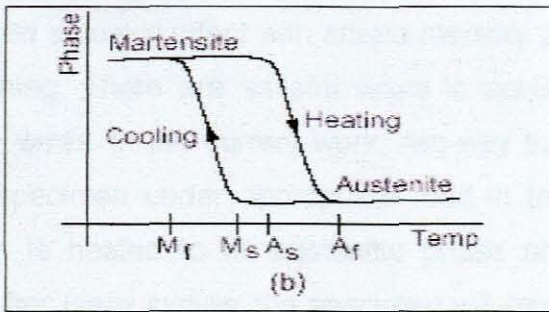


Figure 1.8: Hysteretic relation between austenite and Martensite as a function of temperature. [2]

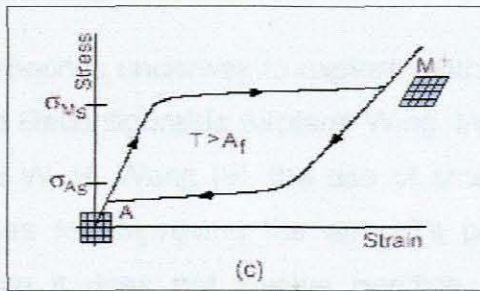


Figure 1.9: Stress-induced phase transformation and pseudoelastic behavior for $T > A_f$. [2]

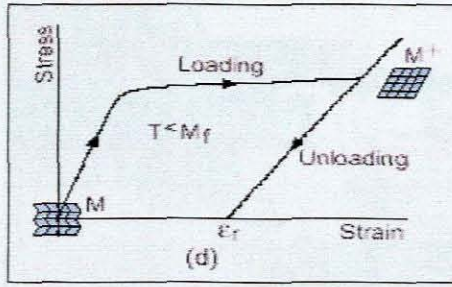


Figure 1.10: Quasiplastic behavior and residual strain generated when $T < M_f$ [2]

To achieve the desired actuator effect with shape memory alloy wires, they must exhibit two-way training. There are several ways to achieve two-way SME in shape memory alloy wires. In the current work, two-way training is achieved by first deforming the specimen under appropriate load in the Martensitic phase. Then, the specimen is heated to its austenitic phase and cooled back to a Martensitic phase. After many cycles, the specimen will begin to exhibit two-way shape memory effect [2].

1.4.2. Review of Smart Structure

There are several approaches underway to explore methods of using the Shape Memory Alloy Actuated Reconfigurable Airplane Wing. In the technical note of J. T. Oh, H. C. Park and W. H. Wang [9], the use of smart structure technology offers some possibilities for improving the aircraft's performance. This work sounds promising since it does not involve bending or twisting a structure purposely designed rigid for flight. However, this is not the case with most approaches like that of the German Aerospace Centre (DLR). Here they proposed a belt rib type construction that was capable of camber changes [F. Campanile, et. al., 2000]. They used a system of ribs connected to an outer belt that would have varying stiffness in the joints. By applying a moment to the structure and varying this stiffness they could control the camber.

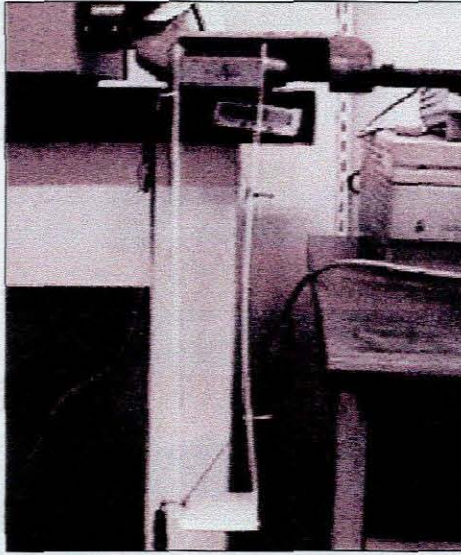


Figure 1.11: Experiment using model [9]

The Aerospace Engineering Department of the Texas A&M University has done some innovative research in shape memory alloy actuated reconfigurable airplane wings [8]. In their research, they created a reconfigurable wing whose airfoil can achieve the potential large-scale deflections necessary to maintain high efficiency as flow conditions change. A high force, high displacement actuator is needed to achieve the deformations required for full airfoil reconfigurability. The wires are heated by electrical current and cooled by natural convection. The wires were put through a series of tests and demonstrated recoverable transformation strain of about 3.7%. A Differential Scanning Calorimeter (DSC) was used to determine the start and finish austenite and martensite transformation temperatures. Quasi-static tests were conducted on a load frame to determine the modulus of elasticity for both the martensitic and austenitic phases of the material.

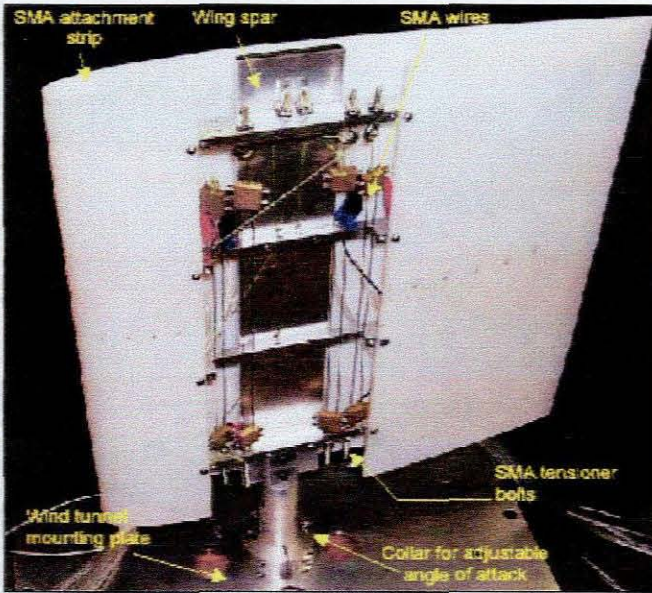


Figure 1.12: One Wind Tunnel Models Come From Aerospace Engineering Department, Texas A&M University [9]

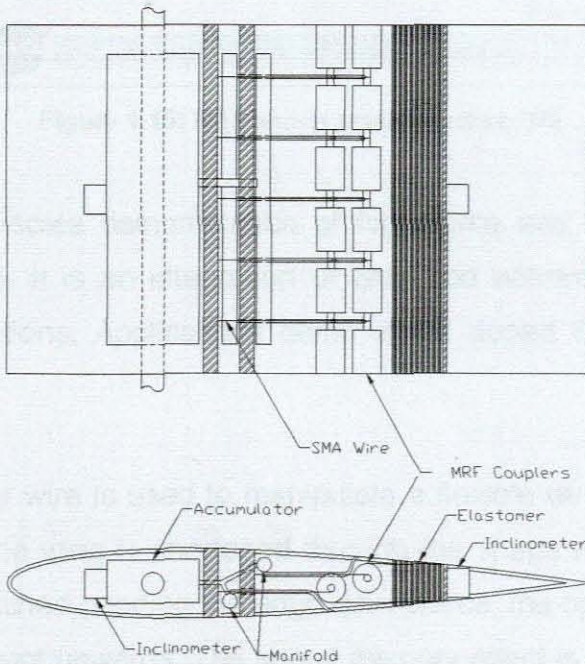


Figure 1.13: Some Other Design [9]

The following pictures have illustrates the SAMPSON Modified F-15 Inlet with smart structures actuation:

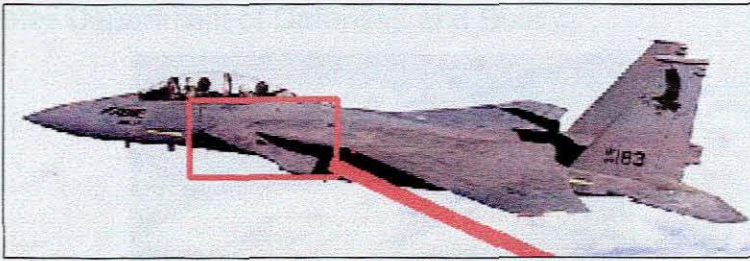


Figure 1.14: F-15 [10]

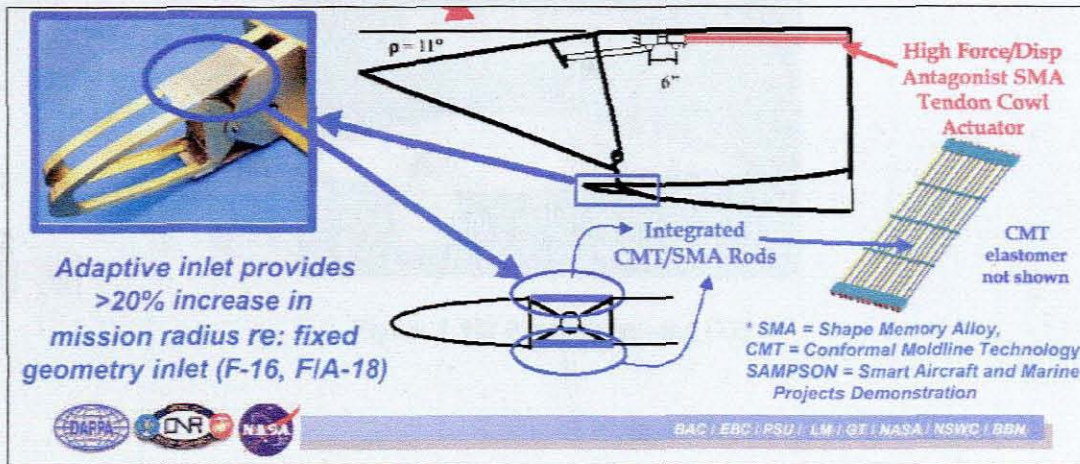


Figure 1.15: F-15 and its smart structure [10]

The F-15 is a full-scale demonstration of high force and displacement smart materials actuation. It is an integration of SMA rod actuators within compliant structure configurations. Applications demo of Pd doped SMA (high transition temperature) [10].

The shape memory wire is used to manipulate a flexible wing surface. The wire on the bottom of the wing is shortened through the shape memory effect, while the top wire is stretched bending the edge downwards, the opposite occurs when the wing must be bent upwards. The shape memory effect is induced in the wires simply by heating them with an electric current, which is easily supplied through electrical wiring, eliminating the need for large hydraulic lines. By removing the hydraulic system, aircraft weight, maintenance costs, and repair time are all reduced. The smart wing system is currently being developed cooperatively through the Defense Advanced Research Project Agency (DARPA, a branch of

the United States Department of Defense), and Boeing.

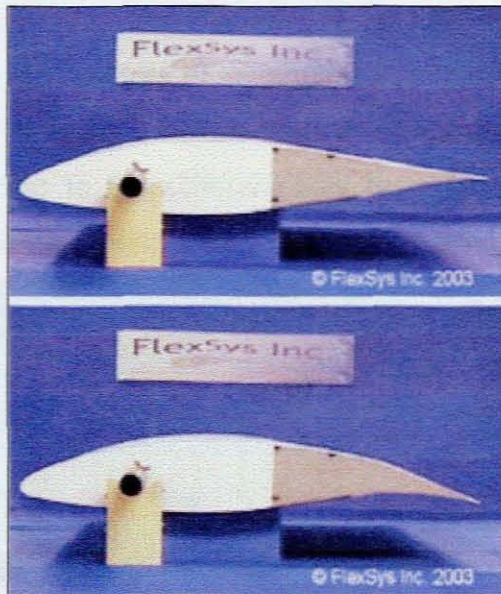


Figure 1.16: A morphing wing [12]

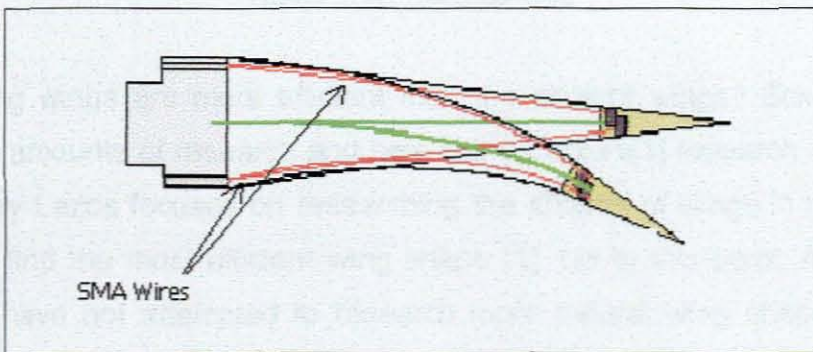


Figure 1.17: Hinge less shape memory alloy Flap [12]

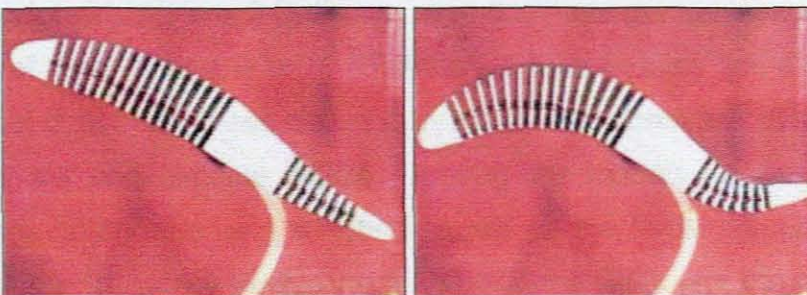


Figure 1.18: Reconfigurable wing from the Technical University of Berlin [13].

1.5. The Inspiration Come From Biology

Human beings always can get lots of inspiration from birds' flying; it's well known that the natural wing is more efficient than the airplane wing because it can change shape according to the different flow regimes. When birds fly at low speed (landing), they always bend their wing as illustrated in Figures 1.19.



Figure 1.19: The birds' wing

Are bending wings are more efficient than the straight wings? Scientists have done large amounts of research and here Barry Lazos's[1] research inspired this paper. Barry Lazos focuses on researching the shapes of wings in nature in an attempt to find the most efficient wing shape [1]. Up to this point, most aircraft designers have not attempted to research more natural wing shapes because they have not had the ability to make models of the complex shapes for testing in wind tunnels. However, Lazos used a technology called stereo lithography which is able to sculpt virtually any 3-D shape using computer guided lasers to form the objects out of a polyurethane resin (Hoffman). Lazos designs some of his wings based almost directly on those occurring in nature and some as hybrids between various animals and airplane wings. Among the models he has tested are seagull wings, shark fins, and a hybrid known as the hyper elliptic, cambered span (Hoffman). In Lazos' studies he found that the wing with the hyper elliptic shape has a 15% improvement in its lift to drag ratio as compared with a normal airplane wing [14].



Figure 1.20: The most efficient wing shape [1]

The photos in figure 1.20 above from LHS to RHS are: seagull wings, shark fins, and a hybrid known as the hyper elliptic. The seagull wings are optimized for slow horizontal flight, the shark fin has good performance in reduce the drag in low speed, and the last conceptual wing has the best lift-to-drag performance. Unfortunately, these nature inspired designs are a bit complex for computer simulations of fluid dynamics so Lazos in unable to say for certain why this particular shape is so efficient.

1.6. Scope of the Thesis

Chapter 2 gives details about a wing model which is designed to apply NiTi shape memory alloys wires in. The aerofoil is chosen from the Boeing Company. It is a supercritical aerofoil and good for high speed cruising flight conditions, the vortex panel method is used to calculate the pressure coefficient for the aerofoil.

Chapter 3 shows the results from the Finite Element simulations, where the best bending wing shape can be derived from them. This section also shows why the bending wings are more efficient than the straight wings.

Chapter 4 introduces the procedure of the manufacture process in producing the model of the morphing wing.

Chapter 5 covers the discussions and conclusions.

Chapter 2 Airfoil Design

2.1. Introduction

This chapter introduces a numerical method, the vortex-panel method, used to determine the aerodynamic characteristics such as the flow velocity, circulation density, and coefficient of pressure over an airfoil as a function of the geometric angle of attack for bodies that experience lift when placed in a low speed flow field where the fluid is said to be inviscid and has constant density. These aerodynamic characteristics can be used to determine the airfoil's lift, drag and moment. This method is often employed for flow calculations on multiple surfaces, such as airfoils, flaps, and slots or cascades representing axial compressors or turbines and many other problems, for which exact solutions are, in general, not available. This numerical method is based on the Kutta Condition and the Kutta-Joukowski theorem. The latter is used to determine the force (i.e., Lift) experienced by a body in a uniform flow field while the Kutta condition describes the environment for a smooth flow past the trailing edge of an airfoil. To this end, the chapter starts off with the definitions of these two important principles for bodies in a flow field.

After discussing the theories used in this numerical wing design method, and describing the numerical architecture for the vortex-panel method, a python computer code is developed to evaluate the aerodynamic characteristics of an airfoil with arbitrary thickness and camber. The results obtained from the python program for the coefficient of pressure of a NACA 2412 airfoil is then compared to existing numerical results for this airfoil and a good agreement was obtained, thus validating the use of the program.

2.2. Kutta Condition

A body with a sharp trailing edge in motion through a fluid creates about itself a circulation of sufficient strength to hold the rear stagnation point at the trailing edge of the finite angle and makes the flow along the trailing edge of the bisector angle smooth (see Fig. 2.2). For a body with a cusped trailing edge, where the upper and lower surfaces meet tangentially, a smooth flow at the trailing edge requires equal velocities on both sides of the edge in the tangential direction.

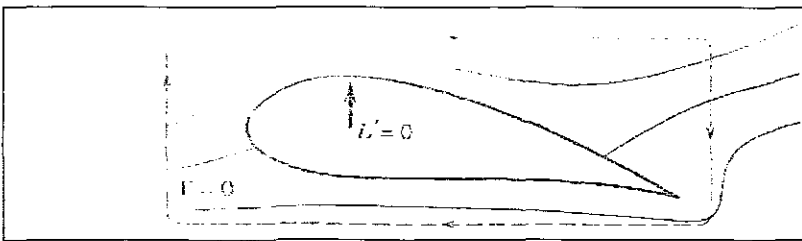


Figure 2.1: Flow around an airfoil with zero circulation. [11]

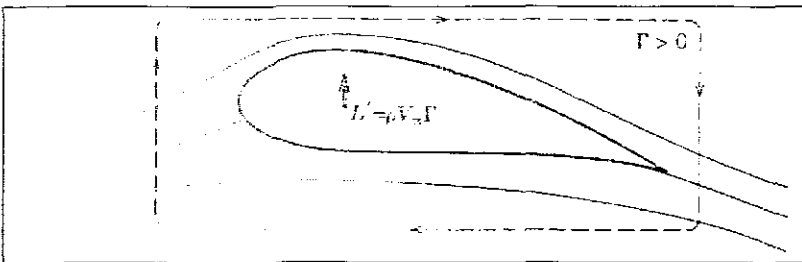


Figure 2.2: Flow past an airfoil in a real fluid. [11]

2.3. Kutta-Joukowski Theorem

The force experienced by a body in a uniform stream is equal to the product of the fluid density, stream velocity, and circulation and has a direction perpendicular to the stream velocity (see Fig 2.2).

2.4. Circulation about a body in a flow field

Consider an airfoil initially at rest. Since $V=0$, the circulation Γ_1 about the circuit around the airfoil must be zero as well (see Fig. 2.3). When the airfoil is set into motion, it will develop a nonzero circulation, Γ_4 , (see Fig. 2.3). Experimental flow observations show that a starting vortex of circulation Γ_3 is shed from the trailing edge. This shedding is associated with the Kutta condition being satisfied for every instant in time. The vortex is swept downstream as seen by an observer on the airfoil.

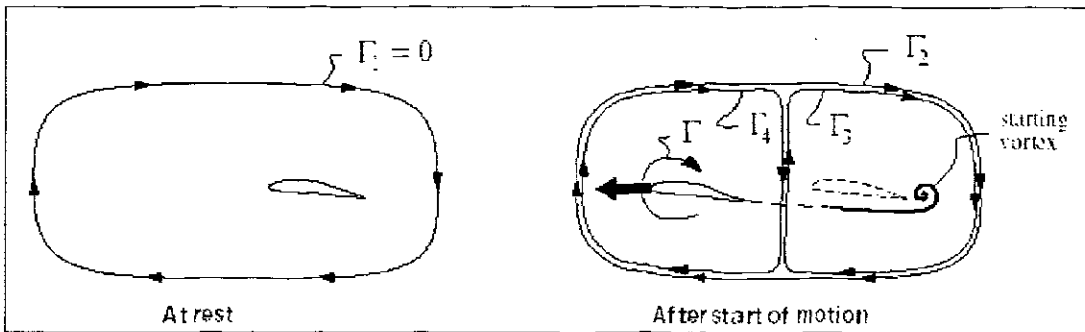


Figure 2.3: Kutta condition [5]

Construction of the circulation circuits gives us:

$$\Gamma_2 = \Gamma_3 + \Gamma_4 \quad (2.1)$$

And by this overall Γ_2 must be the same as the outer circuit's Γ_1 before the airfoil starts to move.

$$\Gamma_2 = \Gamma_1 = 0 \quad (2.2)$$

Therefore, the airfoil and the starting vortex must have equal and opposite circulations.

$$\Gamma_3 = -\Gamma_4 \quad (2.3)$$

It's important that circulation about any circuit is the same to any non-rotating observer. Hence, Kelvin's Theorem applies in both stationary and moving frames of reference. The airfoil and the starting vortex also have the same equal and opposite circulations in either frame.

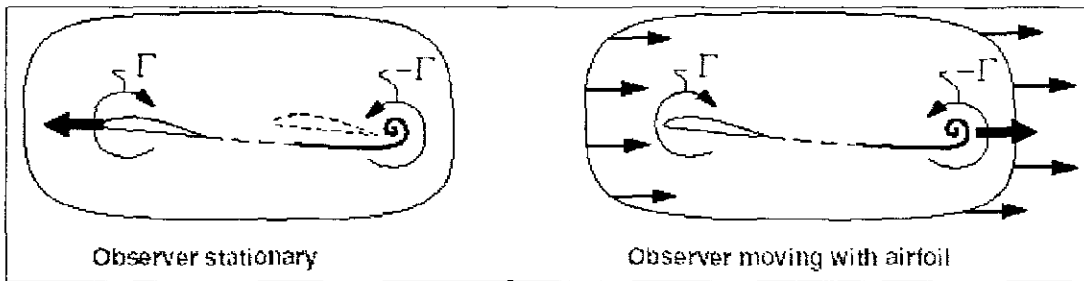


Figure 2.4: Vortex behavior [5]

A long time after the start of motion, the starting vortex is very far downstream behind the airfoil, and has no influence on the flow field about the airfoil. Therefore, the shedding of the starting vortex may be disregarded when considering any steady 2-D airfoil flow. Shed vortices must however still be considered when analyzing unsteady airfoil flows.

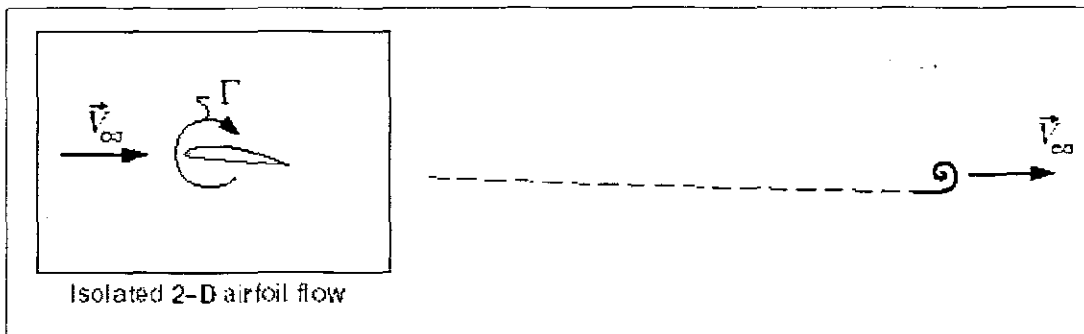


Figure 2.5: 2-D Vortex behavior [5]

2.5. The Airfoil Geometrical Variables

In 1929, the National Advisory Committee for Aeronautics (NACA) began studying the characteristics of a systematic series of airfoils in an effort to find the shape that was best suited for specific purposes. Families of airfoils that were constructed according to a certain plan were tested and their characteristics recorded. The airfoils were composed of a thickness envelope wrapped around a mean camber line (see Fig. 2.6). The mean camber line lies halfway between the upper and lower surfaces of the airfoil and intersects the chord line at the leading and trailing edges (LE and TE respectively).

The various families of airfoils were designed to show the effects of varying geometric variables on the important aerodynamic characteristics, such as lift, drag, and moment, as functions of the angle of attack. The geometric angle of attack, α , is defined as the angle between the flight path and the chord line of the airfoil (see Fig. 2.6). The geometrical variables of the airfoil include:

- the maximum camber, z_c , of the mean camber line and its distance, x_c , behind the leading edge;
- the maximum thickness, t_{max} , and its distance, x_t , behind the leading edge,
- the radius of curvature, r_0 , of the surface at the leading edge; and
- the trailing edge angle between the upper and lower surfaces of the trailing edge [11].

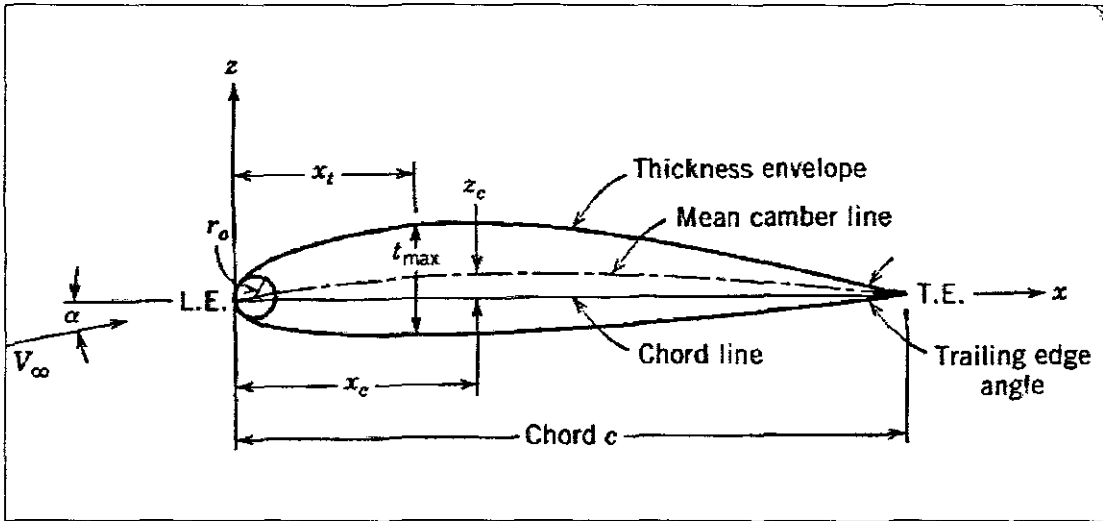


Figure 2.6: Airfoil Geometrical Variables [11]

Theoretical studies and wind tunnel experiments show the effects of these variables that facilitates the choice of shapes for specific applications. The lifting characteristics of an airfoil below the stall are negligibly influenced by viscosity. Furthermore, the result of the pressure forces on the airfoil is only slightly influenced by the thickness envelope, provided that the ratio of maximum thickness to chord is small, the maximum mean camber is small, and the airfoil is acting at a small angle of attack. These three conditions are usually fulfilled in the normal operation of the usual types of airfoils.

In an inviscid fluid of constant density, the lift is proportional to the angle of attack and to the dynamic pressure. The pressure distribution will have the characteristics shown in Fig. 2.7.

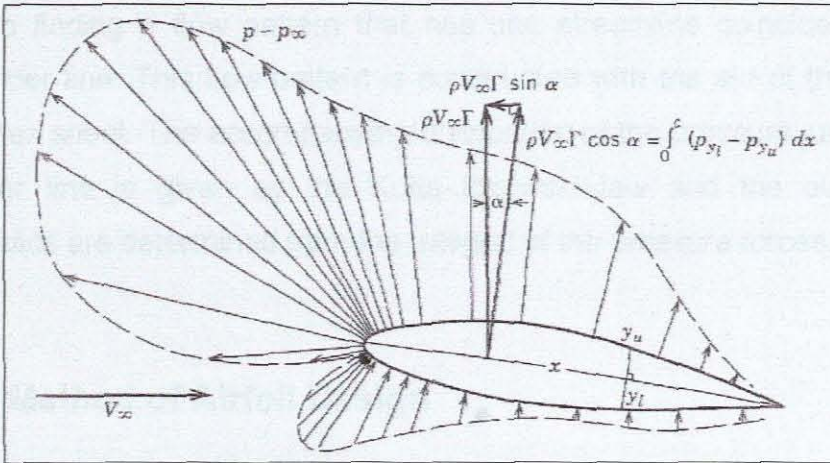


Figure 2.7: The pressure distribution [11]

At low angles of attack the streamlines follow the airfoils surface smoothly, although the boundary layer causes some deviation (See Fig 2.8a).

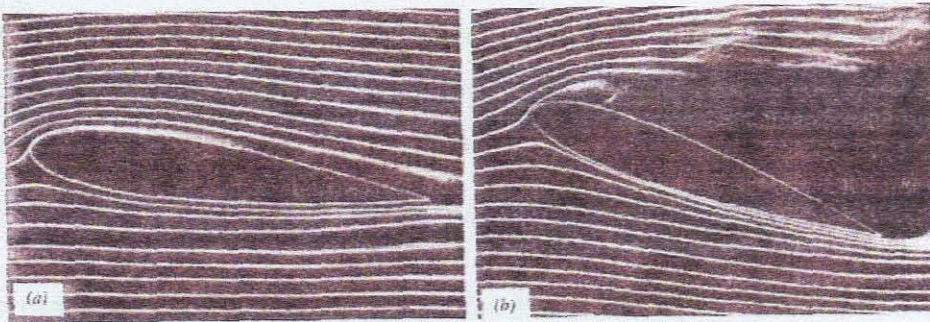


Figure 2.8: The streamlines of the flow around an airfoil
(a) Low α . (b) Stalling angle. [11]

At angles of attack greater than the stalling angle, the flow separates on the upper surface, and the Kutta Condition no longer holds and large vortices are intermittently formed (See Fig 2.8b) and shed. At these high angles of attack the flow becomes unstable and the lift decreases dramatically. This decrease in the lift is also accompanied by an increase in the drag and large changes in the moment exerted on the airfoil by the resulting change in pressure distribution.

The theoretical kinematic problem of the un-stalled airfoil in a flow field is

resolved to finding a flow pattern that has one streamline coincident with the mean camber line. This flow pattern is constructed with the aid of the so-called bound vortex sheet. The accompanying distribution of the pressure jumps across the camber line is given by the Kutta-Jokowski law and the overall lifting characteristics are determined from the integral of the pressure forces.

2.6. The Method of Airfoil Design

The first step in airfoil design is choosing a suitable method of design that has the proper balance of reliability and speed for the given application. Most conventional aircrafts use airfoils that are cambered with arbitrary thickness. The calculation of the aerodynamic characteristics, such as lift, drag, and moment for thick highly cambered, slotted surfaces, with single or multiple flaps and mutual interference effects among wings, fuselages, nacelles, and so forth, require in general the use of numerical methods. One such method, the *source panel method*, only considers non-lifting bodies. This method can however be modified to include lifting bodies by simply including circulation effects, the strength of which is fixed by the Kutta Condition.

The accuracy of the latter numerical method in practical flow problems are limited only by a few factors that include:

- the designer's skill to adequately represent the airfoil's surface by source and vortex panels,
- the number of simultaneous linear algebraic equations that a computer can handle expeditiously,
- the accuracy to which the effects of viscosity and, at high speeds, compressibility can be included in the computations.

The vortex-panel method involves the representation of an airfoil by a closed

polygon of vortex panels (see Fig. 2.9). The airfoil and wing problems can be solved by means of a vortex-panel distribution alone. The circulation of the fuselage and nacelle characteristics and their interference flows dictates the use of source and, possibly, doublet as well as vortex-panels. The vortex-panel introduces a circulation density on each panel that varies linearly from one corner of the panel to the other and is continuous across the corners (see Fig. 2.9). The Kutta Condition is thus easily incorporated into this formulation, and the numerical formulation is stable unless a large number of panels are chosen for a cusped trailing edge. The panels, m , in number, are assumed to be planar and are named in the clockwise direction, starting from the trailing edge. The "boundary points", selected on the surface of the airfoil, are the intersections of neighboring vortex panels. The condition that the airfoil be a streamline is met approximately by applying the condition of zero normal velocity components at "control points" [11], specified as the midpoints of the panels.

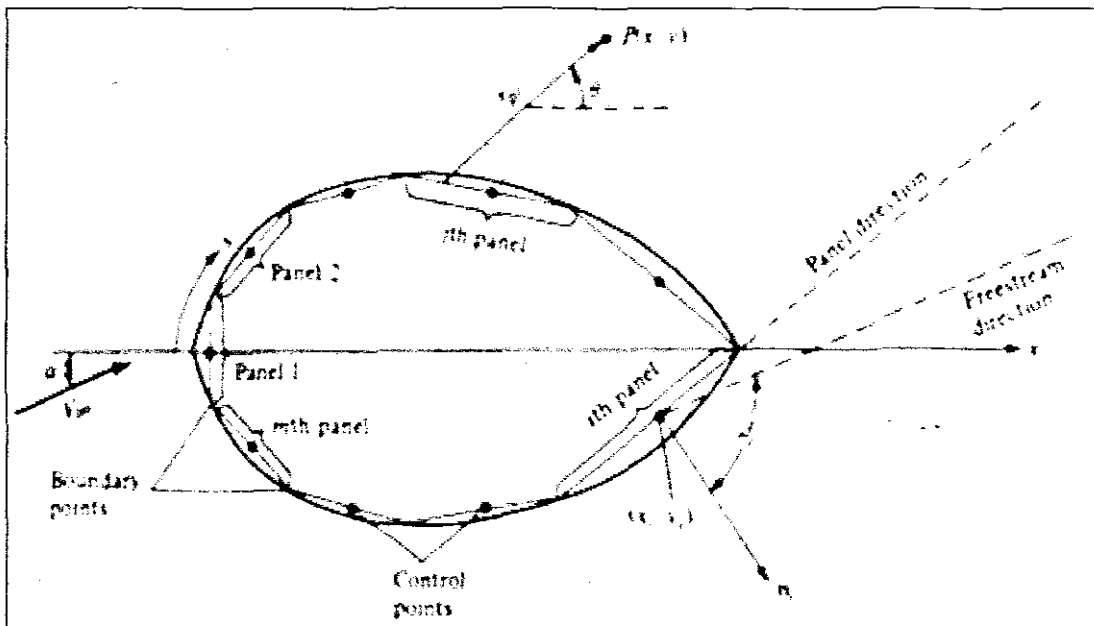


Figure 2.9: Approximate the vortex sheet by a series of straight panels. [11]

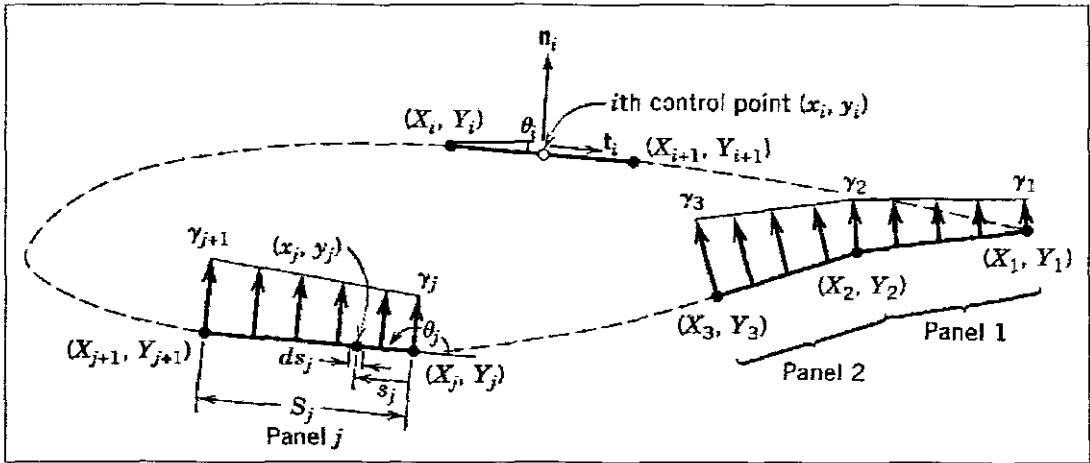


Figure 2.10: Replacement of an Airfoil by Vortex Panels of Linearly Varying Strength. [11]

2.7. Introduction of Airfoil Vortex Sheet Models

An accurate means of representing the flow about an airfoil in a uniform flow is to place a vortex sheet on the airfoil surface (see Fig. 2.11).

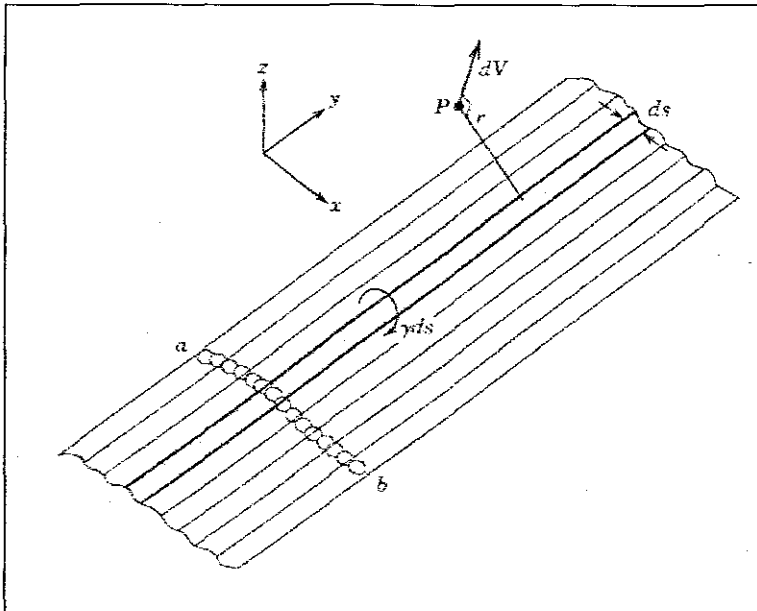


Figure 2.11: The vortex sheet on the airfoil surface [11]

The total velocity $V(x, z)$, which is the vector sum of the freestream velocity, V_x , and the vortex-sheet velocity, V_z , can be forced parallel to the airfoil surface by suitably setting the sheet strength distribution $\gamma(s)$ (see Fig.2.12).

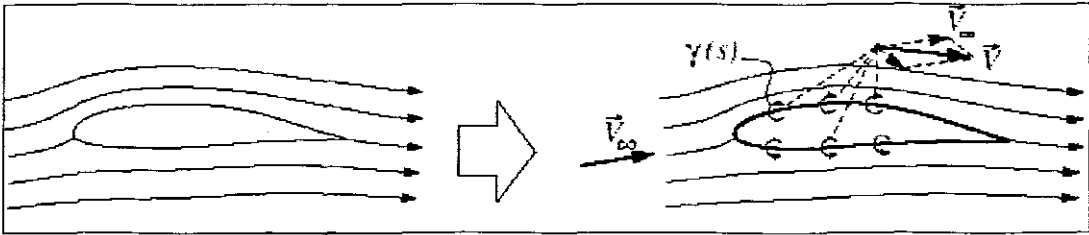


Figure 2.12: The sheet strength distribution [5]

A panel method is normally used to numerically compute $\gamma(s)$. By using a sufficient number of panels, this result can be made as accurate as needed. The main drawback of such numerical calculations is that they give limited insight into how the flow is influenced by changes in the angle of attack or the airfoil geometry. Such insight, which is important for effective aerodynamic design and engineering, is much better provided by simple approximate analytic solutions. The panel method can still be used for accuracy when it's needed.

The sheet strength distribution $\gamma(s)$ is given by:

$$\gamma = \lim_{\Delta s \rightarrow 0} \left[\frac{1}{\Delta s} \oint V \cdot ds \right] \quad (2.4)$$

2.8. Biot-Savart Law

Another very important theoretical aspect for the vortex panel method is Biot-Savart Law. It gives the relation of the velocity induced in the region surrounding a vortex filament (i.e., a line coinciding with the axis of rotation of successive fluid

elements). If P is a point at a distance r from an element, ds , of a vortex filament, then the increment of velocity induced at P by ds is given by the Biot-Savart Law (see Fig. 2.13) [11]:

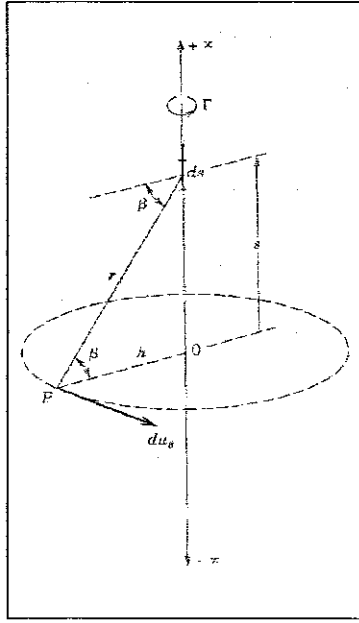


Figure 2.13: The velocity increment du_θ induced by vortex element ds . [11]

$$du_\theta = \frac{\Gamma \cos \beta ds}{4\pi r^2} \quad (2.5)$$

Here Γ is the strength of the vortex filament, r is the distance between the filament and any arbitrary point, P , and β is the angle between the length r and the normal to the filament. The direction of the velocity increment induced by ds is perpendicular to a plane containing the length ds and the point P . The velocity at P induced by the entire filament is obtained by integration:

$$u_\theta = \frac{\Gamma}{4\pi} \int_{-\infty}^{\infty} \frac{\cos \beta ds}{r^2} \quad (2.6)$$

$$r = h \sec \beta; \quad s = h \tan \beta$$

h is the perpendicular distance between the filament and the point P . β is integrated between the limits $\pm\pi/2$ with the result,

$$u_{\theta} = \frac{\Gamma}{2\pi h} \quad (2.7)$$

This is precisely the velocity induced at P , arising from a point vortex at the origin in a two-dimensional flow.

In the vortex-panel method each panel could be considered to have multiple vortex filaments. Each panel will thus have a different circulation density distribution, γ . If we now integrate over the entire airfoil (discretised as panels) with respect to γ , we find the total strength of circulation Γ . This total circulation strength is then used to determine the overall lift of an airfoil (i.e., using the Kutta-Joukowski theorem) as follows:

$$\text{Lift} = \rho V_{\infty} \Gamma$$

Here ρ is the density of the fluid in the surrounding flow field and V_{∞} is the reference velocity of the fluid stream.

2.9. Algorithm of the Vortex Panel Method

During these days, the low speed aerodynamic characteristics of the airfoil that have the arbitrary shape, thickness, camber and orientation are always calculated by vortex panel method.

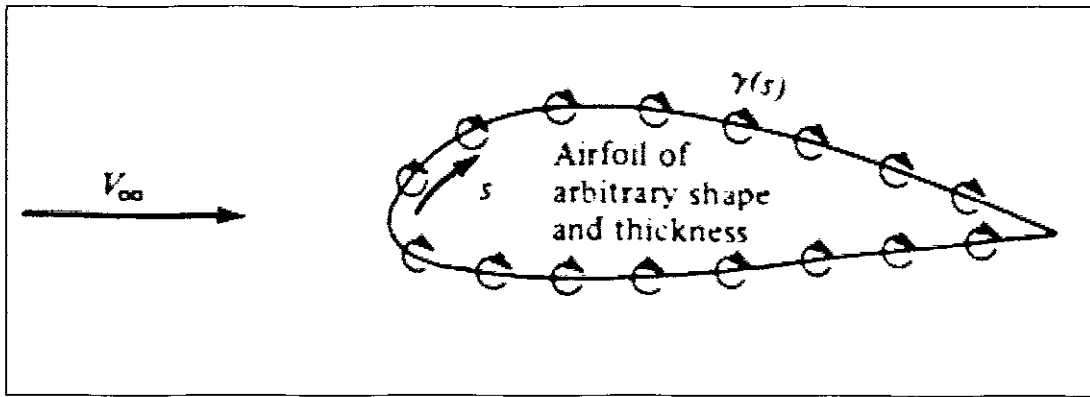


Figure 2.14: Strength Distribution around the Arbitrary Shaped Airfoil [5]

The figure 2.15 is very important for numerical calculation. By using the geometric relationship shown in this figure, the pressure coefficient can be calculated using vortex panel method.

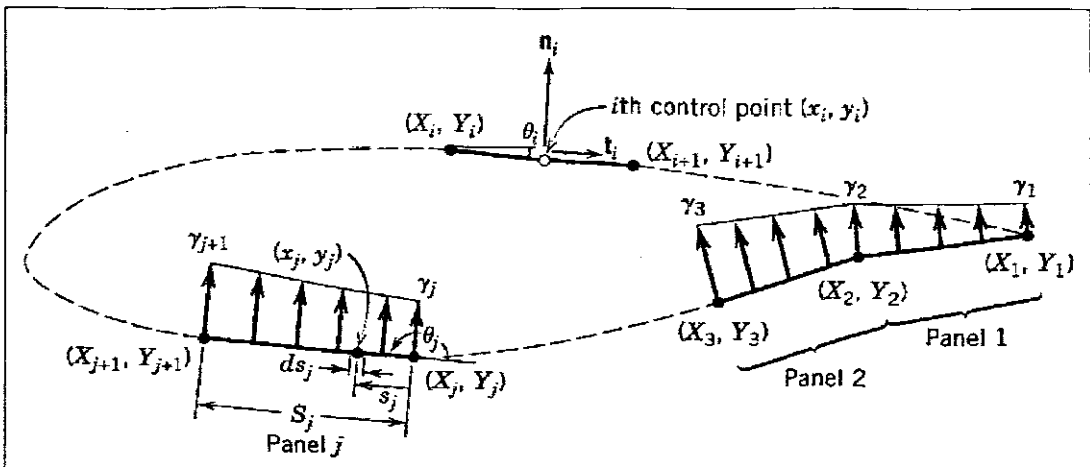


Figure 2.15: Replacement of an Airfoil by Vortex Panels of Linearly Varying Strength

[11]

Velocity potential introduced at P due to the jth panel, here p is any arbitrary point, θ_{pi} is the angle between P and jth panel, ds_j is the length of the panel j.

$$\Delta\phi_j = -\frac{1}{2\pi} \int_j \theta_{pj} \gamma_j ds_j \quad (2.8)$$

Where γ_j is a constant over the j th panel and

$$\theta_{pj} = \tan^{-1} \frac{y - y_j}{x - x_j} \quad (2.9)$$

Velocity potential introduced at P due to all the panels:

$$\phi(P) = \sum_{j=1}^m \Delta\phi_j = -\sum_{j=1}^m \frac{\gamma_j}{2\pi} \int_j \theta_{pj} ds_j \quad (2.10)$$

Put P at the control point of the i th panel, that means point P is located at (x_i, y_i) , then:

$$\theta_{ij} = \tan^{-1} \frac{y_i - y_j}{x_i - x_j} \quad (2.11)$$

And

$$\phi(x_i, y_i) = -\sum_{j=1}^n \frac{\gamma_j}{2\pi} \int_j \theta_{ij} ds_j \quad (2.12)$$

In the presence of a uniform flow V_∞ at an angle of attack α and m vortex panels, the velocity at the i th control point (x_i, y_i) is:

$$\phi(x_i, y_j) = V_\infty (x_i \cos \alpha + y_i \sin \alpha) - \sum_{j=1}^m \int_j \frac{\gamma(s_j)}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ds_j \quad (2.13)$$

Here

$$\gamma(s_j) = \gamma_j + (\gamma_{j+1} - \gamma_j) \frac{s_j}{S_j} \quad (2.14)$$

In equation 2.13, (x_j, y_j) represent coordinates of an arbitrary point on the j th panel of length S_j , which is at a distance s_j measured from the leading edge of the panel. The integration is performed along the entire panel from (X_j, Y_j) to (X_{j+1}, Y_{j+1}) . Here, capital letters are used to denote the coordinates of boundary points [11].

The boundary condition can be explained as function (2.15). It requires that the velocity in the direction of the unit outward normal vector n_i vanishes at the i th control point, so that:

$$\frac{\partial}{\partial n_i} \phi(x_i, y_i) = 0; \quad i = 1, 2, \dots, m \quad (2.15)$$

$$\frac{\partial}{\partial n_i} \phi(x_i, y_i) = \frac{\partial}{\partial n_i} \left[V_\infty (x_i \cos \alpha + y_i \sin \alpha) - \sum_{j=1}^m \int_j \frac{\gamma(s_j)}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ds_j \right] \quad (2.16)$$

On the RHS, the first part is equal to:

$$\begin{aligned} \frac{\partial}{\partial n_i} V_\infty (x_i \cos \alpha + y_i \sin \alpha) &= V_\infty \left(\cos \alpha \frac{\partial x_i}{\partial n_i} + \sin \alpha \frac{\partial y_i}{\partial n_i} \right) \\ &= V_\infty (\cos \alpha \sin \theta_i - \sin \alpha \cos \theta_i) \\ &= V_\infty \sin(\theta_i - \alpha) \end{aligned} \quad (2.17)$$

The second part is equal to:

$$\begin{aligned}
 \frac{\partial}{\partial n_i} \sum_{j=1}^m \int_j \frac{\gamma(s_j)}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ds_j &= \sum_{j=1}^m \frac{\partial}{\partial n_i} \int_j \frac{\gamma(s_j)}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ds_j \\
 &= \sum_{j=1}^m \frac{\partial}{\partial n_i} \int_j \frac{\gamma_j + (\gamma_{j+1} - \gamma_j) \frac{s_j}{S_j}}{2\pi} \tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) ds_j \\
 &= \sum_{j=1}^m \int_j \frac{\gamma_j + (\gamma_{j+1} - \gamma_j) \frac{s_j}{S_j}}{2\pi} \frac{\partial}{\partial n_i} \left[\tan^{-1} \left(\frac{y_i - y_j}{x_i - x_j} \right) \right] ds_j
 \end{aligned} \tag{2.18}$$

Carrying out the involved differentiation and integration for the panels, we obtain:

$$\sum_{j=1}^m (C_{n1_y} \gamma'_j + C_{n2_y} \gamma'_{j+1}) = \sin(\theta_i - \alpha); \quad i = 1, 2, \dots, m \tag{2.19}$$

In which $\gamma' = \gamma / 2\pi V_\infty$ is a dimensionless circulation density, θ_i is the orientation angle of the i th panel measured from the x axis to the panel surface.

$$\begin{aligned}
 C_{n1_y} &= 0.5 \cos(\theta_i - \theta_j) \ln \left\{ 1 + \frac{S_j^2 + 2[-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2} \right\} \\
 &+ \sin(\theta_i - \theta_j) \tan^{-1} \left\{ \frac{[(x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j] S_j}{[(x_i - X_j)^2 + (y_i - Y_j)^2] + [-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j} \right\} \\
 &- C_{n2_y}
 \end{aligned} \tag{2.20}$$

$$\begin{aligned}
C_{n2_y} &= \cos(\theta_i - \theta_j) + 0.5[(x_i - X_j) \cos(\theta_i - 2\theta_j) + (y_i - Y_j) \sin(\theta_i - 2\theta_j)] \\
&\times \ln \left\{ 1 + \frac{S_j^2 + 2[-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2} \right\} / S_j \\
&\quad - \left(\frac{[-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] \sin(\theta_i - \theta_j)}{+ \cos(\theta_i - \theta_j) [(x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j]} \right) \\
&\times \tan^{-1} \left\{ \frac{[(x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j] S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2 + [-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j} \right\} / S_j \}
\end{aligned} \tag{2.21}$$

Or the simplified format:

$$\begin{aligned}
C_{m_y} &= 0.5DF + CG - C_{n2_y} \\
C_{n2_y} &= D + 0.5QF / S_j - (AC + DE)G / S_j
\end{aligned} \tag{2.22}$$

The constants shown in these and some later expressions are defined as:

$$A = -(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j \tag{2.23}$$

$$B = (x_i - X_j)^2 + (y_i - Y_j)^2 \tag{2.24}$$

$$C = \sin(\theta_i - \theta_j) \tag{2.25}$$

$$D = \cos(\theta_i - \theta_j) \tag{2.26}$$

$$E = (x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j \tag{2.27}$$

$$F = \ln \left(1 + \frac{S_j^2 + 2AS_j}{B} \right) \tag{2.28}$$

$$G = \tan^{-1} \left(\frac{ES_j}{B + AS_j} \right) \quad (2.29)$$

$$P = (x_i - X_j) \sin(\theta_i - 2\theta_j) + (y_i - Y_j) \cos(\theta_i - 2\theta_j) \quad (2.30)$$

$$Q = (x_i - X_j) \cos(\theta_i - 2\theta_j) + (y_i - Y_j) \sin(\theta_i - 2\theta_j) \quad (2.31)$$

These constants are functions of ith "control points", those of the boundary points of the jth vortex panel, and the orientation angles of both ith and jth panels.

For $i=j$, the coefficients have simplified values:

$$C_{n1_u} = -1 \quad \text{and} \quad C_{n2_u} = 1 \quad (2.32)$$

To ensure a smooth flow at the trailing edge, the Kutta condition is applied, in the present notation, it becomes:

$$\gamma_i' + \gamma_{m+1}' = 0 \quad (2.33)$$

There are $(m+1)$ equations that are sufficient to solve the $(m+1)$ unknown γ_i' values. We may rewrite this system of simultaneous equations in a more convenient form:

$$\sum_{j=1}^{m+1} A_{ij} \gamma_j' = RHS_i; \quad i = 1, 2, \dots, m+1 \quad (2.34)$$

In which, for $i < m+1$:

$$\begin{aligned}
 A_{n_j} &= C_{n1_{i1}} \\
 A_{n_j} &= C_{n1_{ij}} + C_{n2_{j-1}}; \quad j = 2, 3, \dots, m \\
 A_{n_{m+1}} &= C_{n2_{jm}}
 \end{aligned} \tag{2.35}$$

$$RHS_i = \sin(\theta_i - \alpha) \tag{2.36}$$

And, for $i=m+1$:

$$\begin{aligned}
 A_{n_j} &= 1 = A_{n_{m+1}} \\
 A_{n_j} &= 0; \quad j = 2, 3, \dots, m \\
 RHS_i &= 0
 \end{aligned} \tag{2.37}$$

After the determination of the unknown circulation densities, proceed to compute the velocity and pressure at the control points. At such a point, the velocity has only a tangential component at the panel surface because of the vanishing of the normal component there. Thus, if we let t_i designate the unit tangential vector on the i th panel, the local dimensionless velocity defined as $(\partial\phi/\partial t_i)/V_\infty$ can be computed, which has the expression:

$$V_i = \cos(\theta_i - \alpha) + \sum_{j=1}^m (C_{i1_j} \gamma_j' + C_{i2_j} \gamma_{j+1}'); \quad i = 1, 2, \dots, m \tag{2.38}$$

In which:

$$\begin{aligned}
 C_{i1_j} &= 0.5 \sin(\theta_i - \theta_j) \ln \left\{ 1 + \frac{S_j^2 + 2[-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2} \right\} \\
 &\quad - \cos(\theta_i - \theta_j) \tan^{-1} \left\{ \frac{[(x_i - X_j) \sin \theta_j - (y_i - Y_j) \cos \theta_j] S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2 + [-(x_i - X_j) \cos \theta_j - (y_i - Y_j) \sin \theta_j] S_j} \right\} \\
 &\quad - C_{i2_j}
 \end{aligned} \tag{2.39}$$

$$\begin{aligned}
C_{t2_y} &= \sin(\theta_i - \theta_j) + 0.5[(x_i - X_j)\sin(\theta_i - 2\theta_j) + (y_i - Y_j)\cos(\theta_i - 2\theta_j)] \\
&\times \ln \left\{ 1 + \frac{S_j^2 + 2[-(x_i - X_j)\cos\theta_j - (y_i - Y_j)\sin\theta_j]S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2} \right\} / S_j \\
&+ \left(\begin{aligned} &[-(x_i - X_j)\cos\theta_j - (y_i - Y_j)\sin\theta_j]\cos(\theta_i - \theta_j) \\ &-\sin(\theta_i - \theta_j)[(x_i - X_j)\sin\theta_j - (y_i - Y_j)\cos\theta_j] \end{aligned} \right) \\
&\times \tan^{-1} \left\{ \frac{[(x_i - X_j)\sin\theta_j - (y_i - Y_j)\cos\theta_j]S_j}{(x_i - X_j)^2 + (y_i - Y_j)^2 + [-(x_i - X_j)\cos\theta_j - (y_i - Y_j)\sin\theta_j]S_j} \right\} / S_j
\end{aligned}$$

$$C_{t1_u} = C_{t2_u} = \pi/2 \quad (2.40)$$

Or

$$\begin{aligned}
C_{t1_y} &= 0.5CF - DG - C_{t2_y} \\
C_{t2_y} &= C + 0.5PF/S_j + (AD - CE)G/S_j \\
C_{t1_u} &= C_{t2_u} = \pi/2
\end{aligned} \quad (2.41)$$

The expression in the parentheses following the summation symbol has the physical meaning of the tangential velocity at the i th control point induced by the vortices distributed on the j th panel. To facilitate computer programming the equations had been simplified:

$$V_i = \cos(\theta_i - \alpha) + \sum_{j=1}^m A_{t_y} \gamma_j^1; \quad i = 1, 2, \dots, m \quad (2.42)$$

Where the tangential-velocity influence coefficients are defined as follows:

$$\begin{aligned}
A_{t_y} &= C_{t1_i} \\
A_{t_y} &= C_{t1_j} + C_{t2_{j-1}}; \quad j = 2, 3, \dots, m \\
A_{t_{m+1}} &= C_{t2_m}
\end{aligned} \quad (2.43)$$

The pressure coefficient at the i th control point is:

$$C_{pi} = 1 - V_i^2 \quad (2.44)$$

The architecture for the numerical solution to finding the coefficient of pressure for an airfoil of arbitrary thickness and camber as a function of the angle of attack has been described successfully above. A python code (see Appendix 2) is now developed for ease of these computations. The panel boundary points for the NACA 2412 airfoil was obtained from literature and used as input to the python program. A total of 12 panels were used for an angle of attack of 8° . The results for the coefficient of pressure obtained from the python program were plotted and compared to results found in literature (see Table 2.1). A good agreement was found between these two results, which validate the use of the python program (see Fig 2.18). Fig 2.16 shows results for the coefficient of pressure for an angle of attack of 8° . The figure gives results for 12-, 48-, and 120-panel solution.

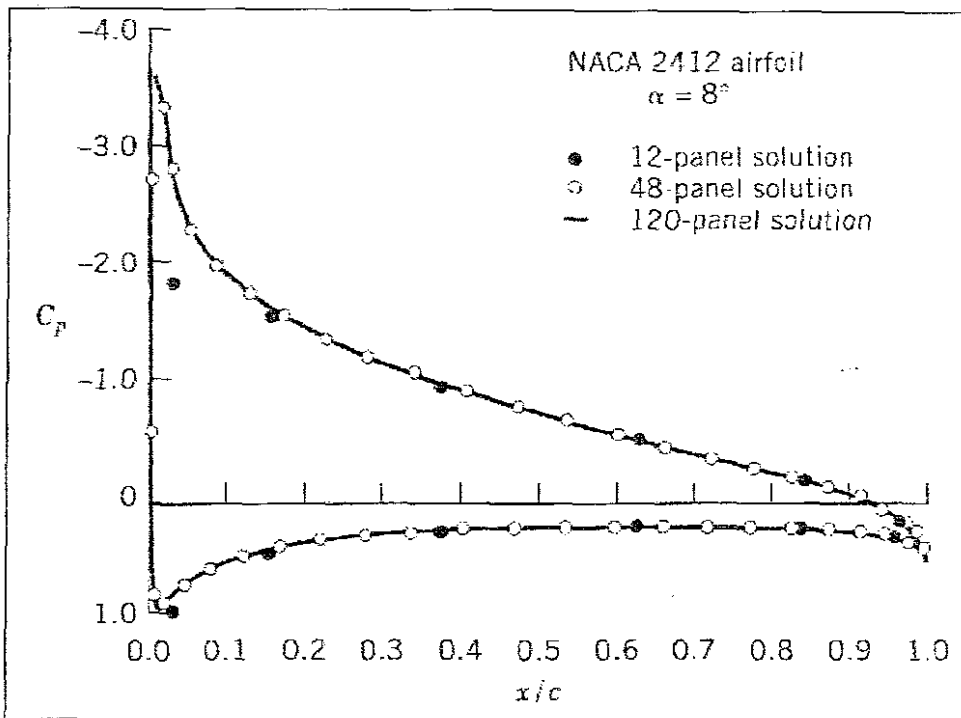
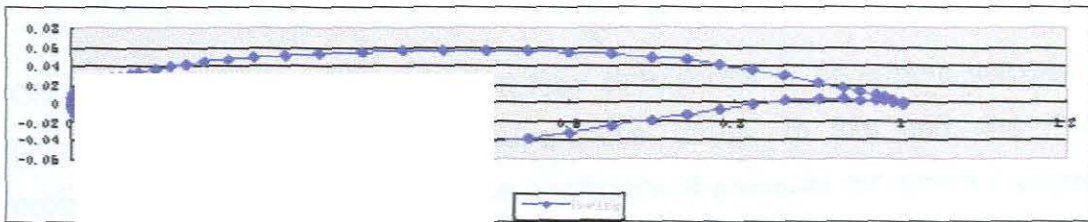


Figure 2.16: The reference of the pressure coefficient [11]

Table 2-1: The data comparison

Index	CP(l) from reference	CP(l) from Python code
1	0.2630	0.26296473
2	0.1969	0.19690378
3	0.2097	0.20973462
4	0.2667	0.2666984
5	0.4707	0.47071002
6	0.9929	0.99287832
7	-1.8101	-1.81010084
8	-1.5088	-1.50884427
9	-0.9334	-0.93343659
10	-0.5099	-0.50990103
11	-0.1688	-0.16878934
12	-0.1674	-0.16741964



The shape of the airfoil

As de
Boeing
this in
aerofo
of 20
progra

rcritical airfoil (see Fig. 2.17) based on the
il with a cusped trailing edge) will be used in
f this aerofoil were obtained from the NASG
right belongs to the Boeing Company. A total
is airfoil was used as input to the python
ire was obtained as a function of an angle of

attack of 8° . The distribution of the coefficient of pressure for the supercritical airfoil was plotted together with the results of the NACA 2412 (see Fig 2.18) clearly shows the difference in the values obtained for the coefficient of pressure for an airfoil with a cusped trailing edge.

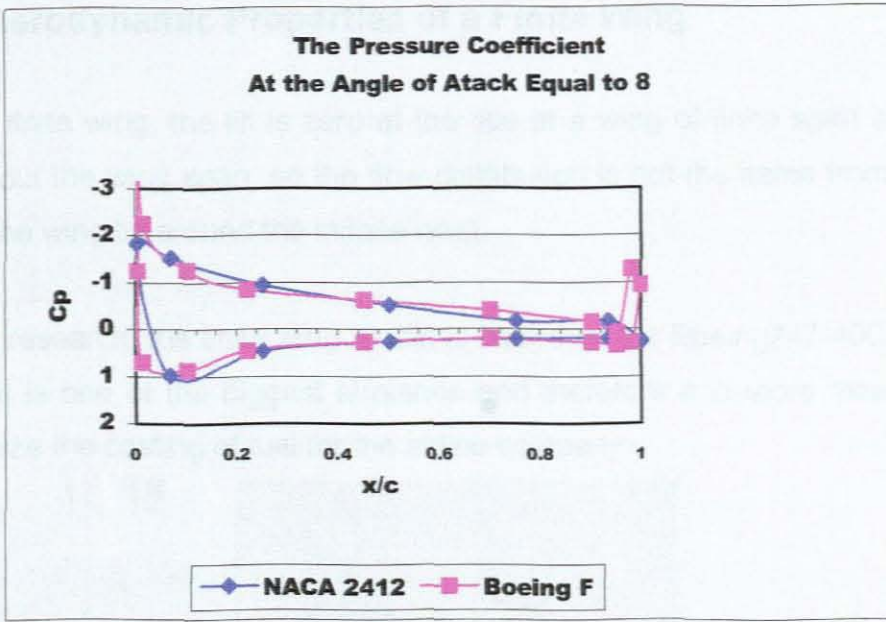


Figure 2.18: The comparison of the pressure coefficient

Chapter 3 discusses finite element flow simulations to determine velocity and pressure distributions over the supercritical airfoil. To this end, the Python program was used to determine the coefficient of pressure for different angles of attack for the supercritical airfoil. The angle of attack (AOA) was varied from 0 degrees to 10 degrees. The result of this computation can be seen in Fig. 2.19.

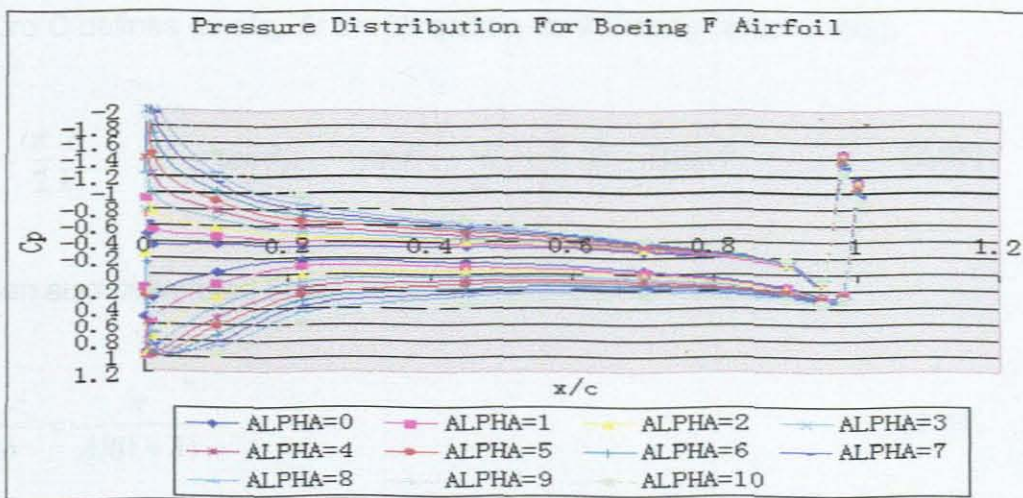


Figure 2.19: Pressure Distribution for Boeing F Aerofoil

2.10. Aerodynamic Properties of a Finite Wing

For the finite wing, the lift is zero at the tips of a wing of finite span and varies throughout the wing span, so the flow distribution is not the same from the wing root to the wing tip around the infinite wing.

For this research, the finite wing model is based on the Boeing 747-400, because of that it is one of the biggest airplanes and therefore it is more meaningful to economize the costing of fuel for the airline company.

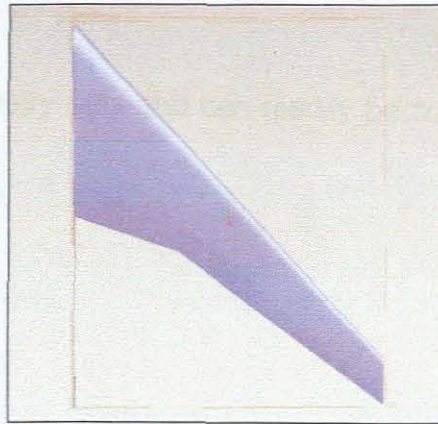


Figure 2.20: The dimension of the finite wing model

Dimensionless spanwise distance Y is defined as $y/0.5b$ and dimensionless chord C defines as c/c_s . At the j th station on a linearly tapered wing,

$$\theta_j = \frac{\pi j}{2k}; \quad Y_j = \cos \theta_j; \quad \text{and} \quad C_j = 1 - (1 - \lambda) \cos \theta_j \quad (2.45)$$

It can also be verified that:

$$\frac{m_{0s} c_s}{4b} = \frac{\pi}{AR(1 + \lambda)} \quad (2.46)$$

It is more convenient to simplify A_n by A_N and to replace n by $2N-1$ elsewhere in

that equation, where $N=1, 2, \dots, k$.

$$\sum_{N=1}^k D_{jN} A_N = \alpha_{a_j}; \quad j = 1, 2, \dots, k \quad (2.47)$$

In which, for a uniform distribution of $m_0=2\pi$,

$$D_{jN} = \left[\frac{1}{C_j} + \frac{(2N-1)\pi}{AR(1+\lambda)\sin\theta_j} \right] \sin(2N-1)\theta_j \quad (2.48)$$

D_{jN} is a function of geometry only and can readily be computed for given values of j and N .

$$C_{D_i} = \frac{\pi^3}{AR(1+\lambda)^2} \sum_{N=1}^k (2N-1) A_N^2 \quad (2.49)$$

$$C_{L_j} = \frac{2\pi}{C_j} \sum_{N=1}^k A_N \sin(2N-1)\theta_j \quad (2.50)$$

$$\alpha_{y_i} = \frac{\pi}{AR(1+\lambda)} \sum_{N=1}^k (2N-1) A_N \frac{\sin(2N-1)\theta_j}{\sin\theta_j} \quad (2.51)$$

$$\bar{m} = \frac{1}{1+\lambda} \int_0^1 m C dY \quad (2.52)$$

The integral in the last equation can be evaluated numerically using the rule:

$$\bar{m} = \frac{1}{2(1+\lambda)} \sum_{j=1}^{k-1} (m_j C_j + m_{j+1} C_{j+1}) (Y_j - Y_{j+1}) \quad (2.53)$$

The detail information for the wing model:

Table 2-2: The detail information for the wing model

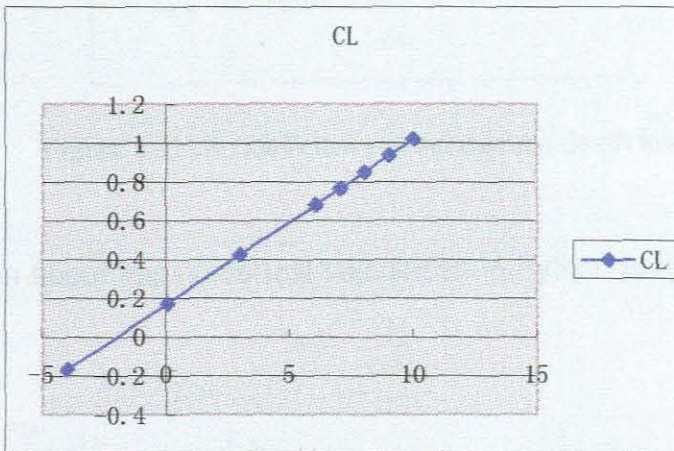
Taper ratio	0.239986
Total area	1871.928cm ²
Total span	115.88cm
Aspect ratio	7.173445

The coefficient of lift according to the different angle of attack:

Table 2-3: Lift Coefficient in different AOA

AOA	Cl
-4	-0.169947648
0	0.169947648
3	0.42486912
6	0.679790591
7	0.764764474
8	0.849738239
9	0.934712122
10	1.019686005

The lift coefficient for the wing model:

**Figure 2.21:** Lift Coefficient

The induced drag coefficient and the lift coefficient:

Table 2-4: Lift Coefficient and Drag Coefficient

α	$C_{di} = CL^2 / \pi A Re$	CL
$\alpha = -4$	0.00197847	-0.16995
$\alpha = 0$	0.00197847	0.169948
$\alpha = 3$	0.012365438	0.424869
$\alpha = 6$	0.031655521	0.679791
$\alpha = 7$	0.040064025	0.764764
$\alpha = 8$	0.049461751	0.849738
$\alpha = 9$	0.059848726	0.934712
$\alpha = 10$	0.071224938	1.019686

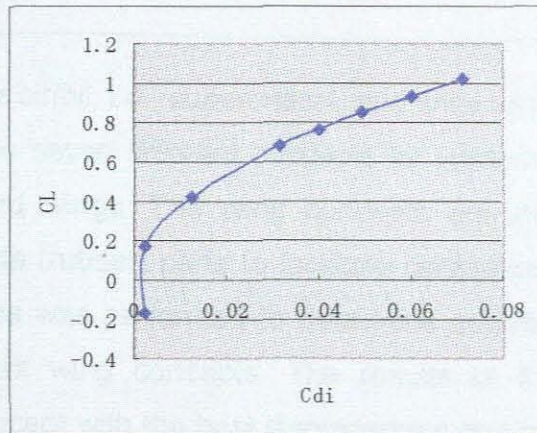


Figure 2.22 Lift Coefficient and Induce drug Coefficient

The information about this numerical calculation in Python program can be found in appendix C.

Chapter 3 Conceptualiza -tion and FE Simulation

This Chapter uses the airfoil, i.e., supercritical, and finite wing shape described in Chapter 2 to introduce seven different concepts for conformability based on the natural shapes of bird wings. The wing concepts are all modeled with rigid (aluminum) and flexible (rubber) parts to facilitate conformability. An initial set of finite element analyses was performed to determine possible deformed shapes for the seven different wing concepts. The results of these analyses were compared and the concept with the best displacement and rotation was selected. A further finite element analysis was now performed on an initially straight wing and three deformed wing shapes to study the differences in velocity and pressure distributions over these wings. From these results, the deformed wing would provide the best lift was selected.

3.1. The MSC.Nastran Analysis for the Wing Design

The MSC.Nastran product family is a model that can be used to analyze product designs ranging from simple components to complex structures and systems [22]. The MSC.Nastran Basic product configuration is valuable when performing linear static, normal modes, and buckling analyses on models of unlimited size. It can

reduce the time and cost required to get the new products to market by using simulation instead of the traditional build-and-break process. It can be applied for the strength and fatigue of aircraft structures, such as fuselage, wings and flaps, and landing gear.

In this thesis, MSC.Nastran is used to calculate the deformation of the wing parts in different kind of loading conditions. MSC.Patran is used as the pre- and postprocessor for the MSC.Nastran analyses.

3.1.1. The Structure of the Wing

According to the wing's model that has been designed before, for getting various wings shapes, the model is separated into several different parts, as shown in the following figure 3.1:

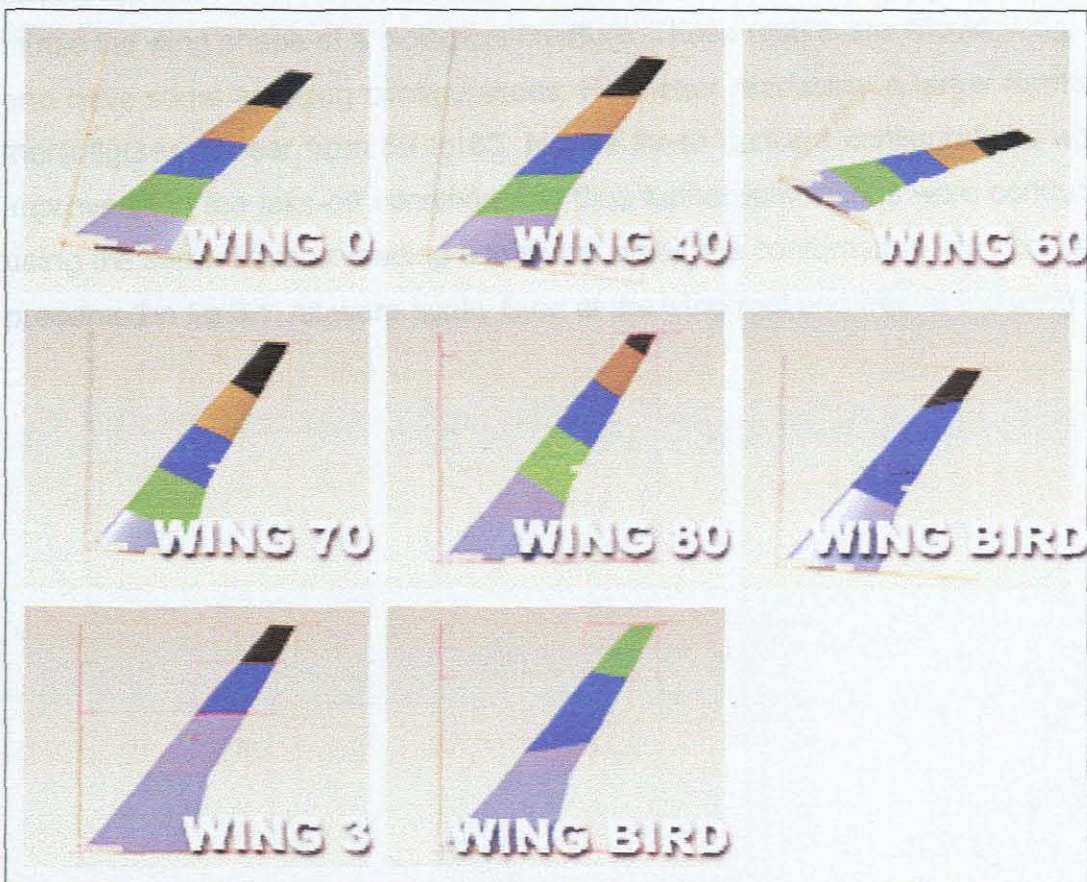


Figure 3.1: The Wing Models

These concepts were used because it is envisaged that they will produce different kinds of deformations of the wing shape chosen in Chapter 2. Each of these wing concepts are made of different materials, i.e. flexible (rubber) and rigid (aluminum). WING 0 has components that are all parallel to each other. WING 40 has components which are at an angle of 40° to the leading edge. Wings 60, 70, and 80 follow the same approach in that they are angled at 60° , 70° , and 80° respectively to the leading edge. The two WING BIRD concepts were chosen because they would typically mimic a bird's wing movements. The WING 3 concept was chosen to determine if it would simulate typical winglet behaviors.

Initially different loading conditions were used to determine the deformation (displacement and rotation) of the conformable wing concepts that would best mimic the wing shape of a biological creature. These results are shown in fig. 3.2 and have some amusing configurations. For better readability a larger format of these figures appear from 59 to 65. Not all these concept configurations would improve lift in the take-off condition and thus further simulations were conducted using the same transfer loading conditions on all the conformable wing concepts presented in fig 3.1. All were rigidly fixed at the wing root and free at the wing tip.

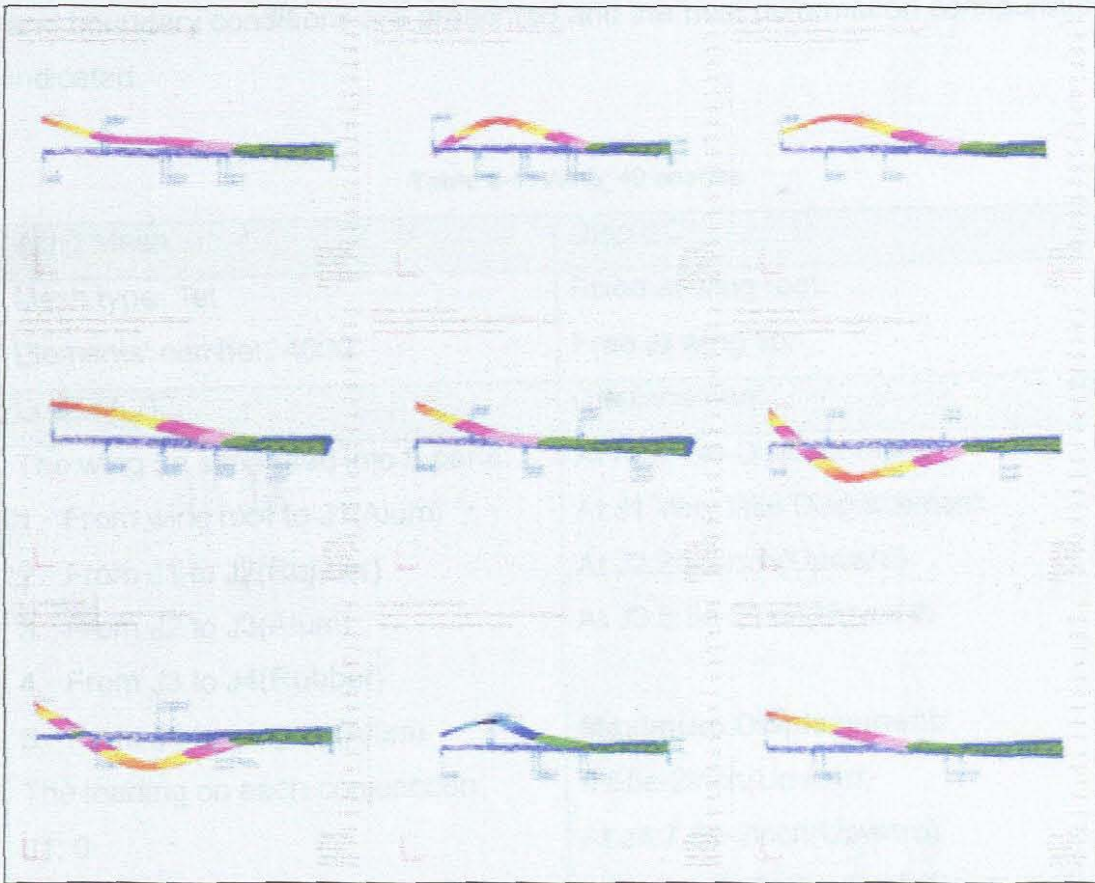


Figure 3.2: The Wings' Model's analysis

The loads that were applied at each junction are as follows (starting from the root to tip): junction 1: 0N, junction 2: 89N upward, junction 3: 89N upward, junction 4: 133.5N upward, and junction 5: 89N downward. A comparison of these results is presented below.

Here all the loads are reasonable assumptions i.e., the deformation caused would be ascribed due to similar loading conditions. The results are useful for comparison and help us to choose one wing's structure that has the best performance in terms of displacement and rotation.

3.1.2. The Comparison

A comparison of the 7 conformable wing concepts, subjected to the same loading

and boundary conditions are presented and the best deformation configuration is indicated.

Table 3-1: Wing_40 analyse

Wing Mesh	Disp BC
Mesh type: Tet Elements' number: 4000	Fixed at wing root. Free at wing tip.
Loading BC	Displacement
The wing be separated into 5 parts: 1. From wing root to J1(Alum) 2. From J1 to J2(Rubber) 3. From J2 to J3(Alum) 4. From J3 to J4(Rubber) 5. From J4 to wing tip(Alum) The loading on each conjunction: J1: 0 J2: 89N Upward J3: 89N Upward J4: 133.5N Upward Wing tip:89N Downward	At root: No Displacement. At J1:Very little Displacement. At J2:2e-2inch(Upward) At J3:5.8e-2inch(Upward) Maximum Displacement: 9.65e-2inch(Upward) At J4:7.5e-2inch(Upward) AT Wing tip:6.43e-3inch(Downward) Twisting: At wing tip: At wing tip clockwise.

The simulation result is shown in figure 3.3.

Wing_40 the maximum displacement is 9.65e-2 m.

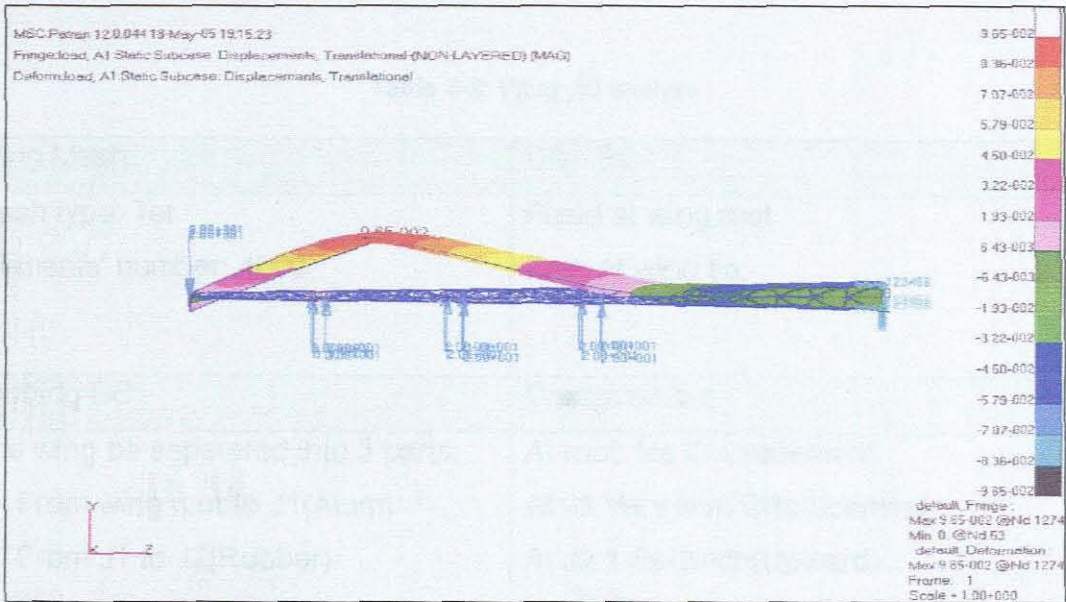


Figure 3.3: The wing_40's deformation

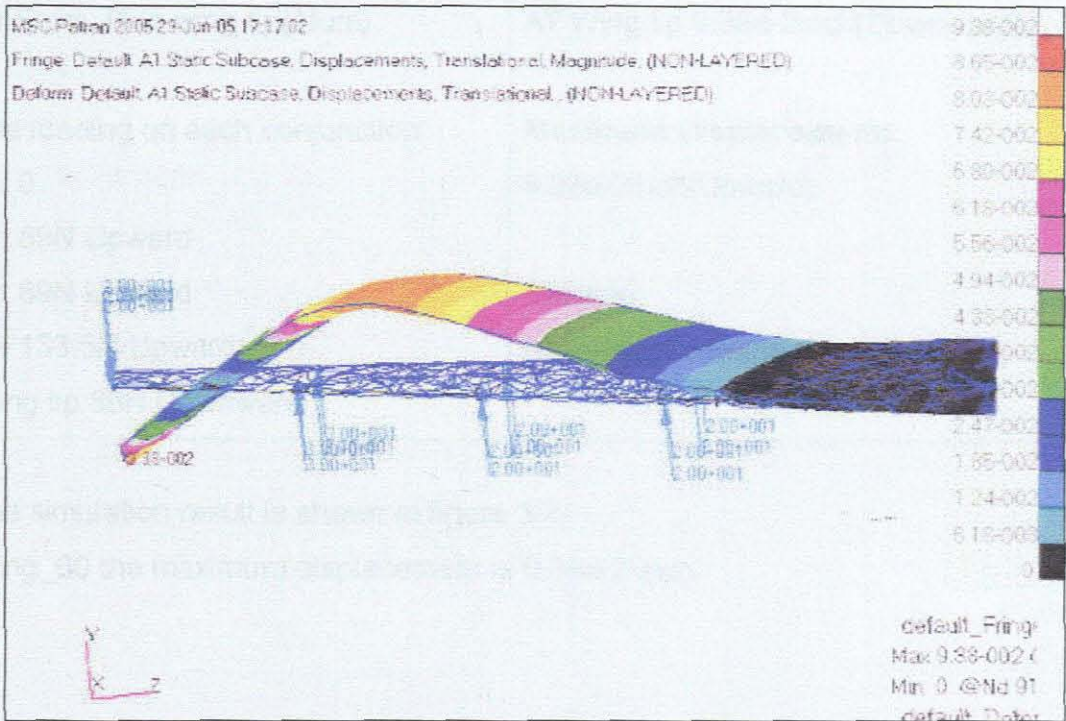


Figure 3.4: The wing_60's deformation

Table 3-2: Wing_60 analyse

Wing Mesh	Disp BC
Mesh type: Tet Elements' number: 4000	Fixed at wing root. Free at wing tip.
Loading BC	Displacement
The wing be separated into 5 parts: 6. From wing root to J1(Alum) 7. From J1 to J2(Rubber) 8. From J2 to J3(Alum) 9. From J3 to J4(Rubber) 10. From J4 to wing tip(Alum)	At root: No Displacement. At J1: Very little Displacement. At J2: 1.5e-2inch(Upward) At J3: 5.8e-2inch(Upward) At J4: 4.5e-2inch(Upward) AT Wing tip: 9.38e-2inch(Downward)
The loading on each conjunction: J1: 0 J2: 89N Upward J3: 89N Upward J4: 133.5N Upward Wing tip: 89N Downward	Maximum Displacement: 9.38e-2inch(Upward) Twisting: At wing tip: At wing tip clockwise.

The simulation result is shown in figure 3.4.

Wing_60 the maximum displacement is 9.38e-2 inch.

Table 3-3: Wing_70 analyse

Wing Mesh	Disp BC
Mesh type: Tet Elements' number: 4000	Fixed at wing root. Free at wing tip.
Loading BC	Displacement
The wing be separated into 5 parts:	At root: No Displacement.
11. From wing root to J1(Alum)	At J1: Very little Displacement.
12. From J1 to J2(Rubber)	At J2: 1.73e-2inch(Upward)
13. From J2 to J3(Alum)	At J3: 6.92e-2inch(Upward)
14. From J3 to J4(Rubber)	At J4: 5e-2inch(Upward)
15. From J4 to wing tip(Alum)	AT Wing tip: 1.31e-1inch(Downward)
The loading on each conjunction:	Maximum Displacement:
J1: 0	1.31e-1inch(Upward)
J2: 89N Upward	
J3: 89N Upward	Twisting:
J4: 133.5N Upward	At wing tip: At wing tip clockwise.
Wing tip: 89N Downward	

The simulation result is shown in figure 3.5.

Wing_70 the maximum displacement is 1.31e-1 inch.

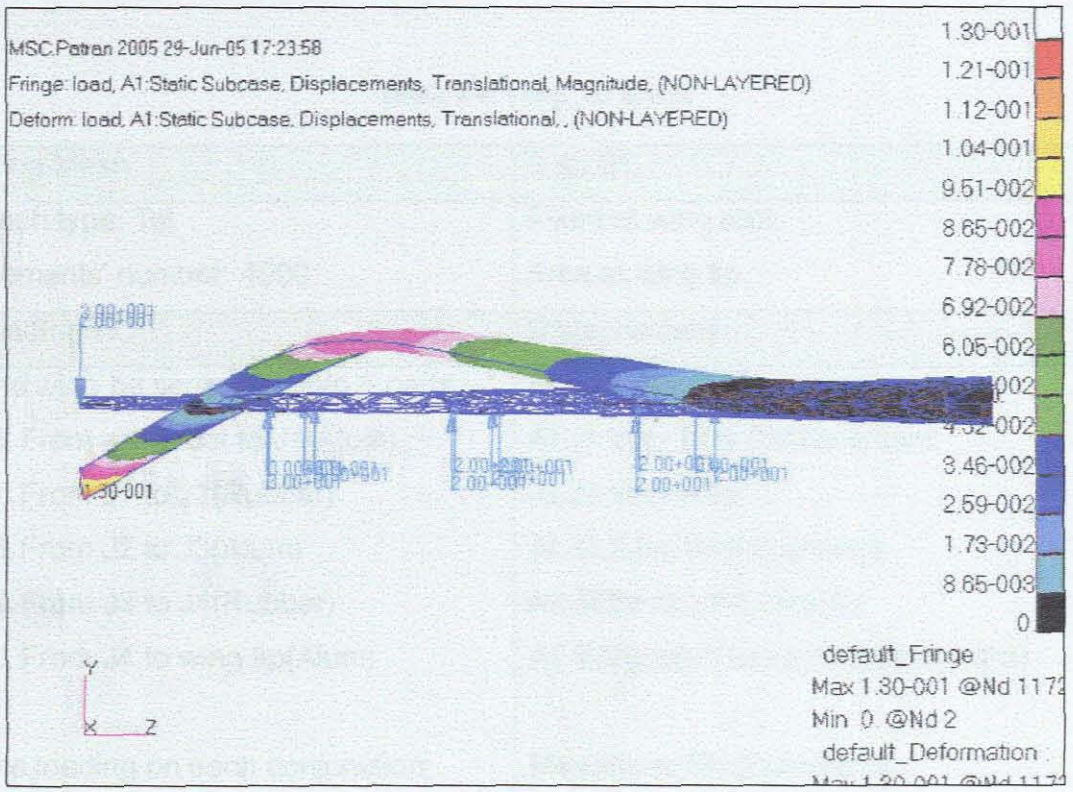


Figure 3.5: The wing_70's deformation

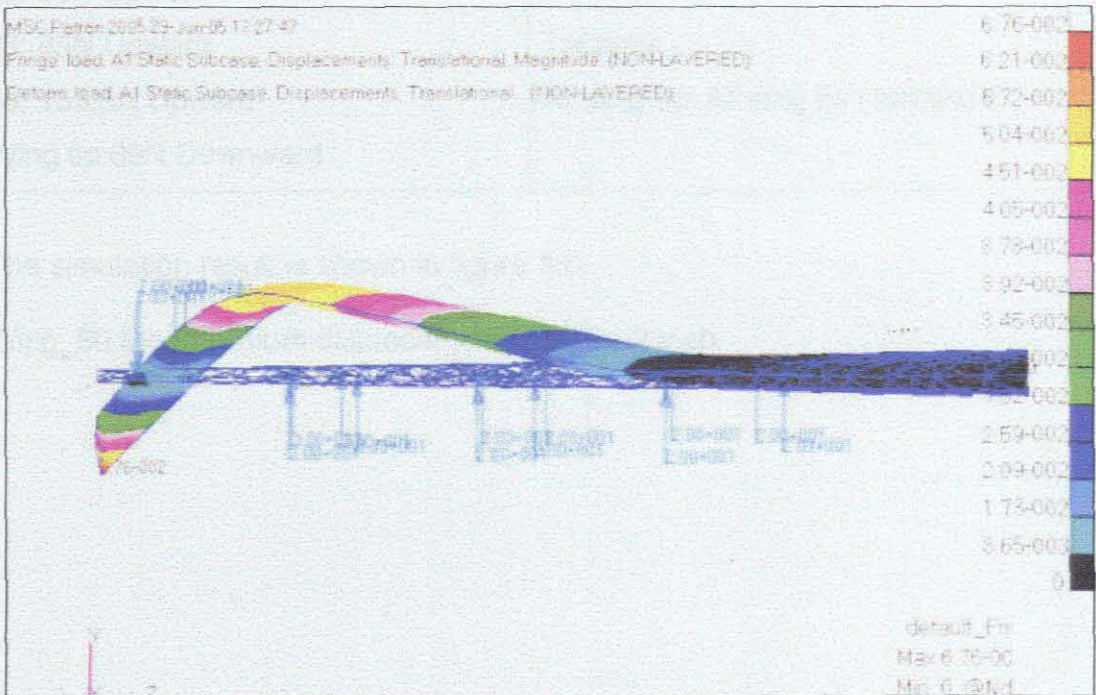


Figure 3.6: The wing_80's deformation

Table 3-4: Wing_80 analys

Wing Mesh	Disp BC
Mesh type: Tet Elements' number: 4000	Fixed at wing root. Free at wing tip.
Loading BC	Displacement
The wing be separated into 5 parts: 16. From wing root to J1(Alum) 17. From J1 to J2(Rubber) 18. From J2 to J3(Alum) 19. From J3 to J4(Rubber) 20. From J4 to wing tip(Alum)	At root: No Displacement. At J1: Very little Displacement. At J2: Very little At J3: 2.5e-2inch(Upward) At J4: 6e-2inch(Upward) AT Wing tip: 6.76e-2inch(Downward)
The loading on each conjunction: J1: 0 J2: 89N Upward J3: 89N Upward J4: 133.5N Upward Wing tip: 89N Downward	Maximum Displacement: 6.76e-2inch(Upward) Twisting: At wing tip: At wing tip clockwise.

The simulation result is shown in figure 3.6.

Wing_80 the maximum displacement is 6.76e-2 inch.

Table 3-5: Wing_0 analyse

Wing Mesh	Disp BC
Mesh type: Tet	Fixed at wing root.
Elements' number: 4000	Free at wing tip.
Loading BC	Displacement
The wing be separated into 5 parts:	At root: No Displacement.
21. From wing root to J1(Alum)	At J1: Very little Displacement.
22. From J1 to J2(Rubber)	At J2: 1.85e-2inch(Upward)
23. From J2 to J3(Alum)	At J3: 6.8e-2inch(Upward)
24. From J3 to J4(Rubber)	
25. From J4 to wing tip(Alum)	Maximum Displacement:
	9.27e-2inch(Upward)
The loading on each conjunction:	At J4: 8.1e-2inch(Upward)
J1: 0	AT Wing tip: 4.33e-2inch(Downward)
J2: 89N Upward	
J3: 89N Upward	Twisting:
J4: 133.5N Upward	At wing tip: At wing tip clockwise.
Wing tip: 89N Downward	

The simulation result is shown in figure 3.7.

Wing_0 the maximum displacement is 9.27e-2inch

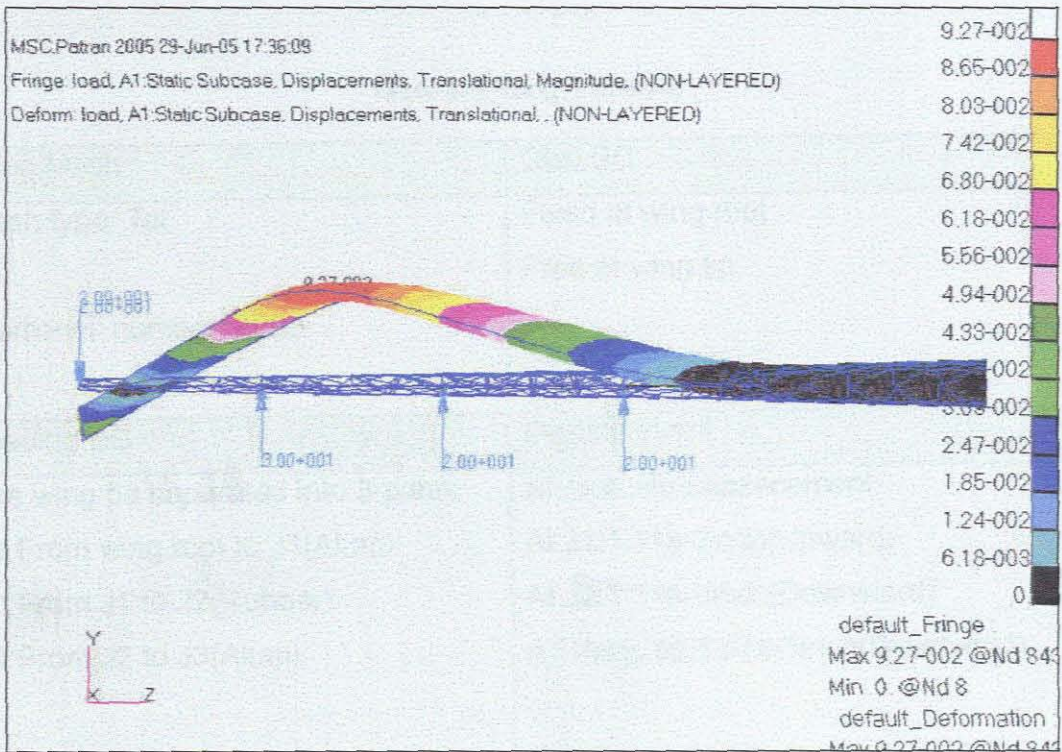


Figure 3.7: The wing_0's deformation

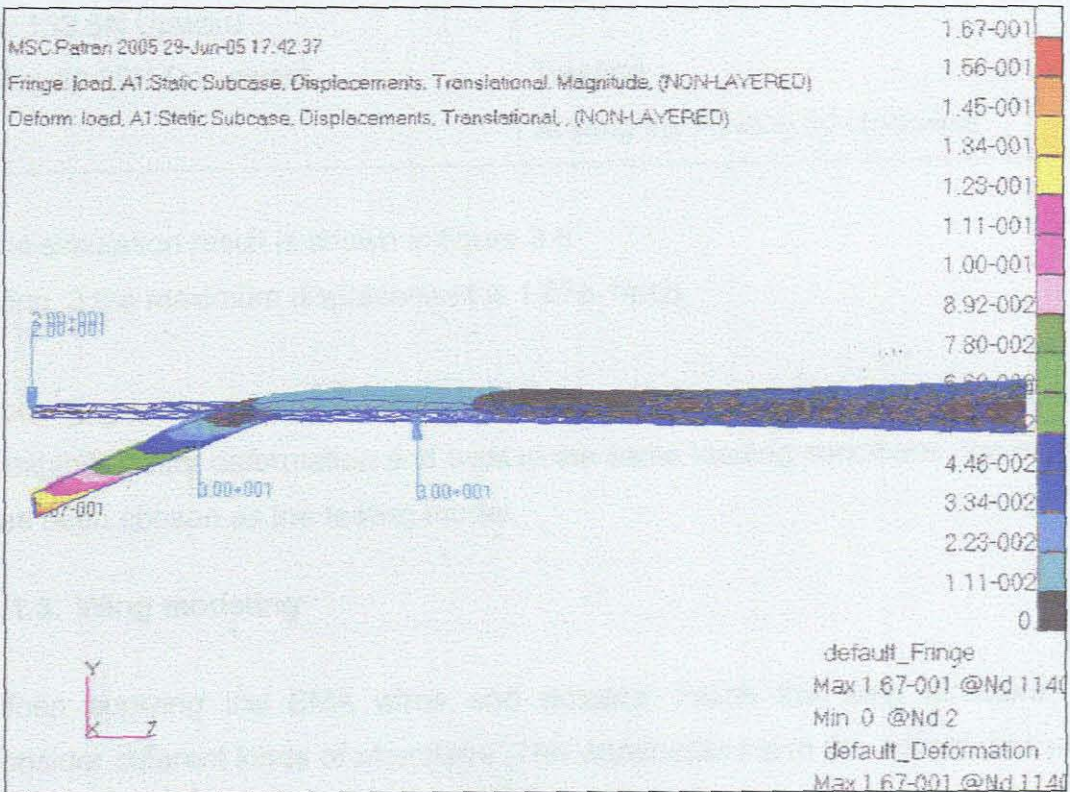


Figure 3.8: The wing_3's deformation

Table 3-6: Wing_3 analyse

Wing Mesh	Disp BC
Mesh type: Tet Elements' number: 4000	Fixed at wing root. Free at wing tip.
Loading BC	Displacement
The wing be separated into 3 parts: 26. From wing root to J1(Alum) 27. From J1 to J2(Rubber) 28. From J2 to J3(Alum)	At root: No Displacement. At J1: 1.11e-2inch(Upward) At J2: 1.11e-2inch(Downward) AT Wing tip: 1.67e-1inch(Downward)
The loading on each conjunction: J1: 133.5N Upward J2: 133.5N Upward Wing tip: 89N Downward	Maximum Displacement: 1.67e-1inch(Downward) Twisting: At wing tip: At wing tip clockwise.

The simulation result is shown in figure 3.8.

Wing_3 the maximum displacement is 1.67e-1inch

According to the results of the above comparisons, wing_70 is the best because it exhibits better deformation and twist in the same loading conditions, therefore it has been chosen as the testing model.

3.1.3. Wing modeling

When applying the SMA wires and actuator inside the wing structure, we consider different kinds of structures. The weaknesses and the advantages were compared and according to the structures of wing_70, we separated the structure

into 4 parts: the first and the third part are assumed to be made up of aluminums, and the second part and fourth parts are assumed to be made up of rubber, as shown in figure [Fig 3.9], [Fig 3.10].

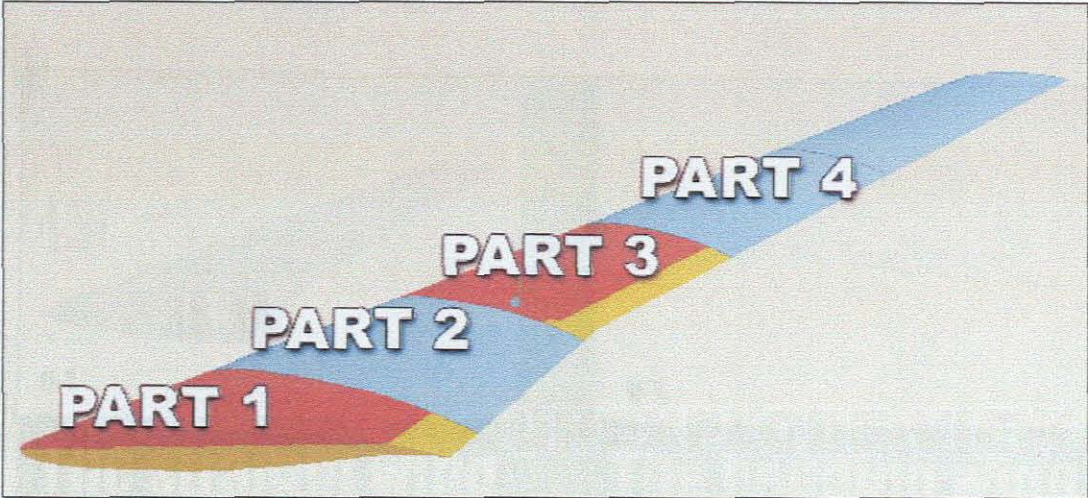


Figure 3.9: Unexploded view for structure

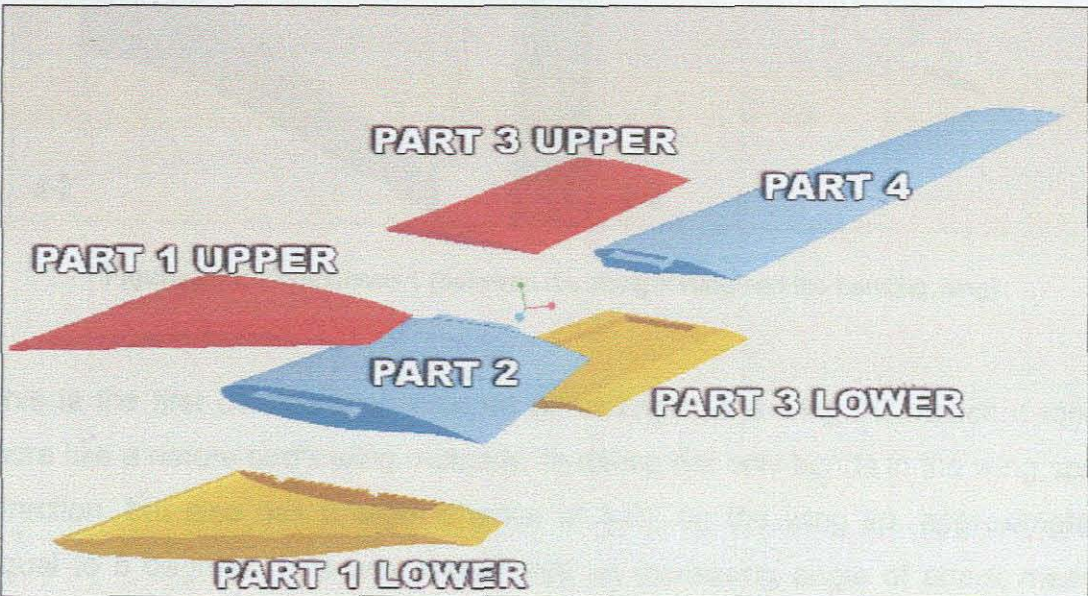


Figure 3.10: Explode view for the wing structure

The approach in controlling structural deformation is to incorporate actuators into part 1 and 3 of the wing structure, in which actuation strain can be regulated. In

such studies, SMA wires were used as embedded actuator to control the deflection in part 2 and part 4. After actuation it is envisaged that the deformation will result as follows:

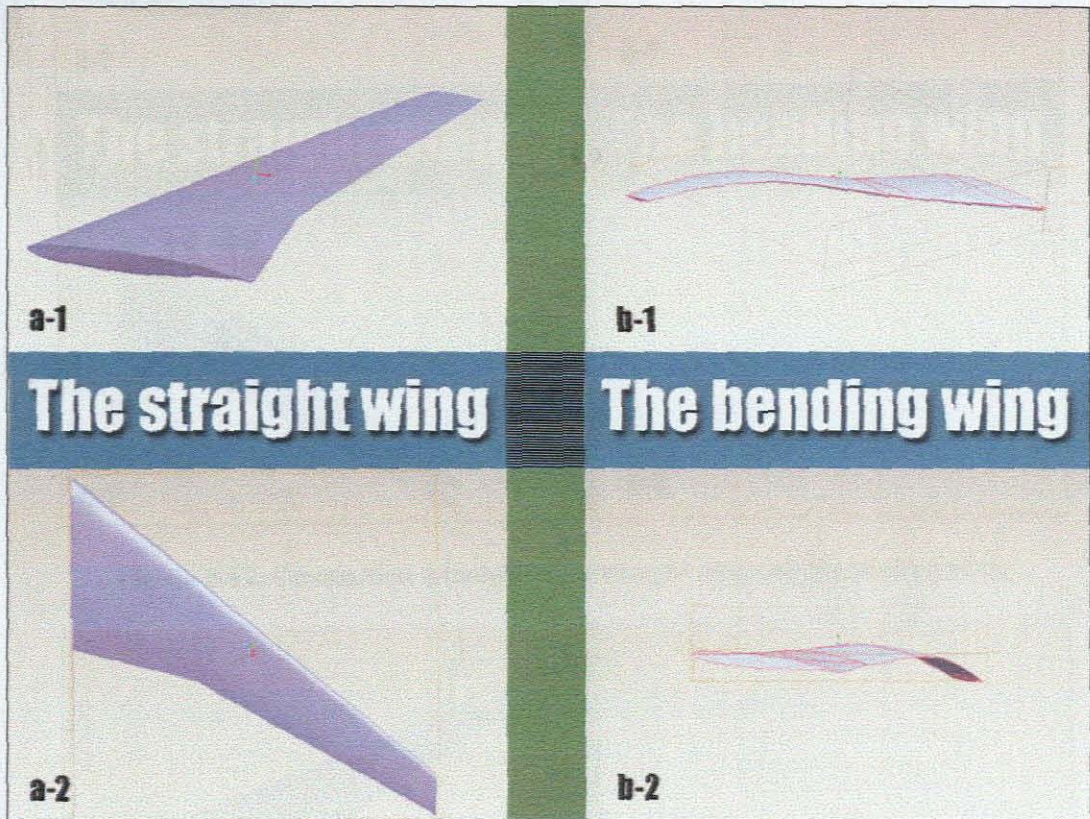


Figure 3.11: Comparison 1 (between the straight wing and the bending wing)

This is the first configuration created according to the wing 70 model; it looks more like a nature bird's wing. Actually, this wing not only bends in the wing span direction, but also has a slight change of twist on the wing tip, approximately equal to 5 degrees. It is well known that an increasing angle of attack means more lift can be found on the wing. But for this bending wing, this kind of twist is not in accordance with the torsion.

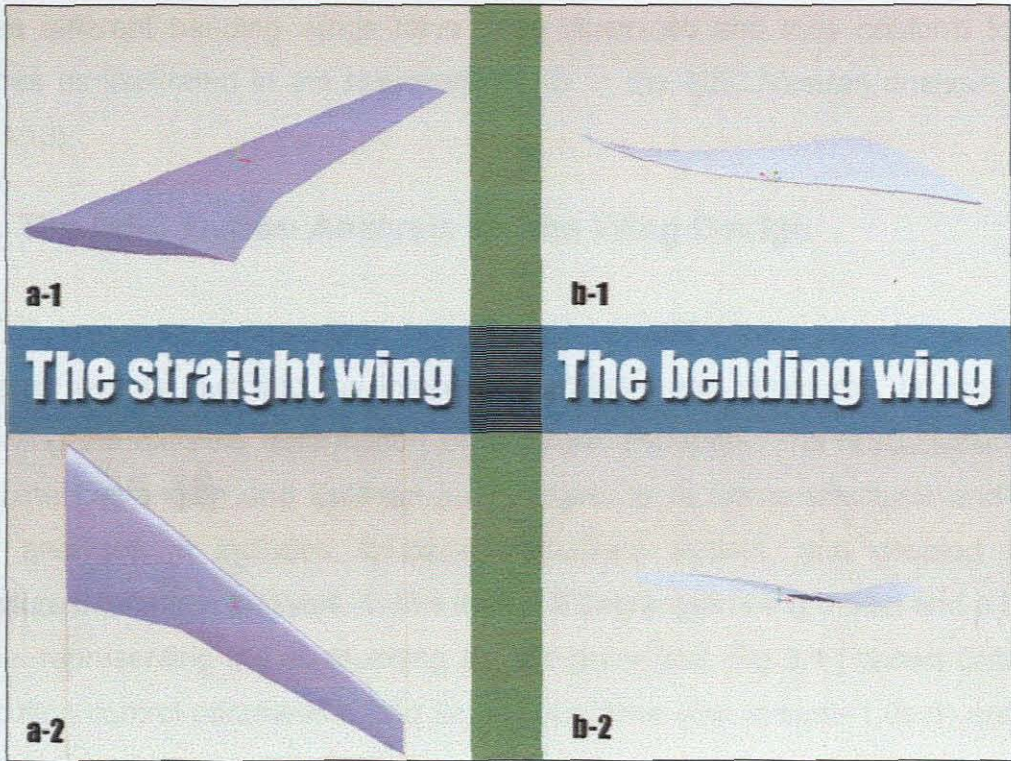


Figure 3.12: Comparison 2 (between the straight wing and the bending wing)

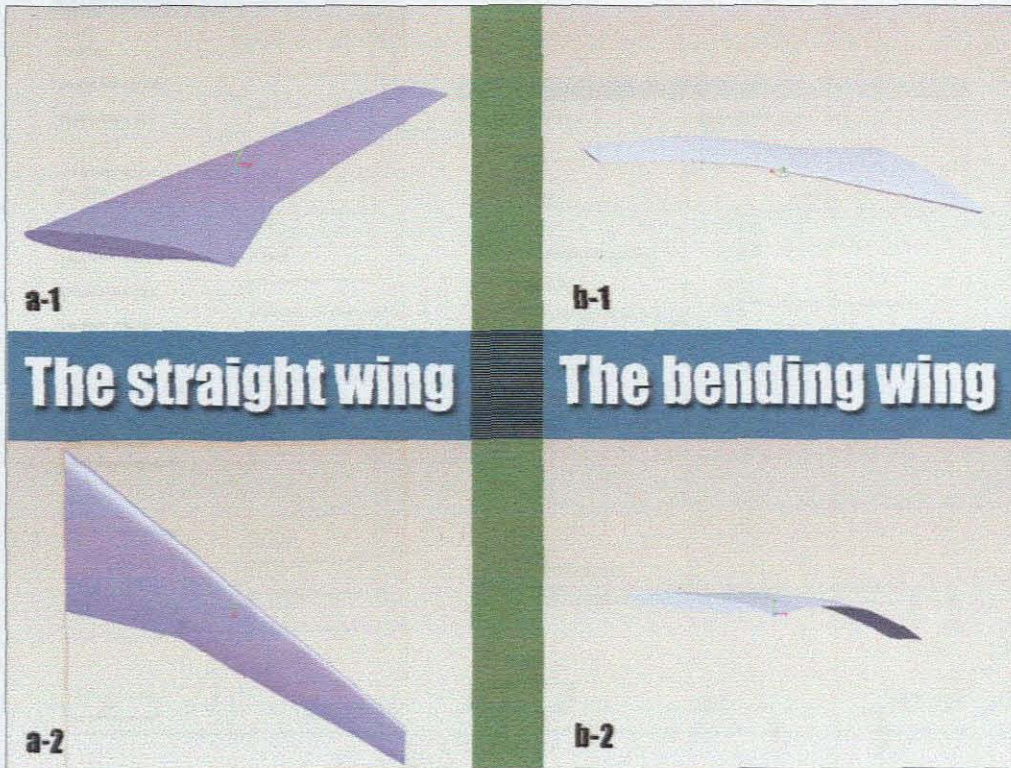


Figure 3.13: Comparison 3 (between the straight wing and the bending wing)

Three different bending wings have been illustrated and they conform to the shapes as illustrated in the research results in the MSC.Nastran analysis (see Fig 3.13).

3.2. The MSC.Dytran Analysis for the Wing Design

MSC.Dytran was designed to help people simulate extreme, short-duration dynamic events involving the interaction of fluids and structures, or problems involving the extreme deformation of structural materials. The MSC.Dytran can combine Lagrangian and Eulerian technologies to do some structural and fluid flow analyses. It includes structure-to-structure contact, and coupled fluid-structure interaction analysis. In this thesis, a Lagrangian wing model and a Euler model representing the surrounding air, are presented. Fig 3.14 shows Dytran's execution control parameters (end time=0.02s, time step at start=1.0e-6) and the input options for air.

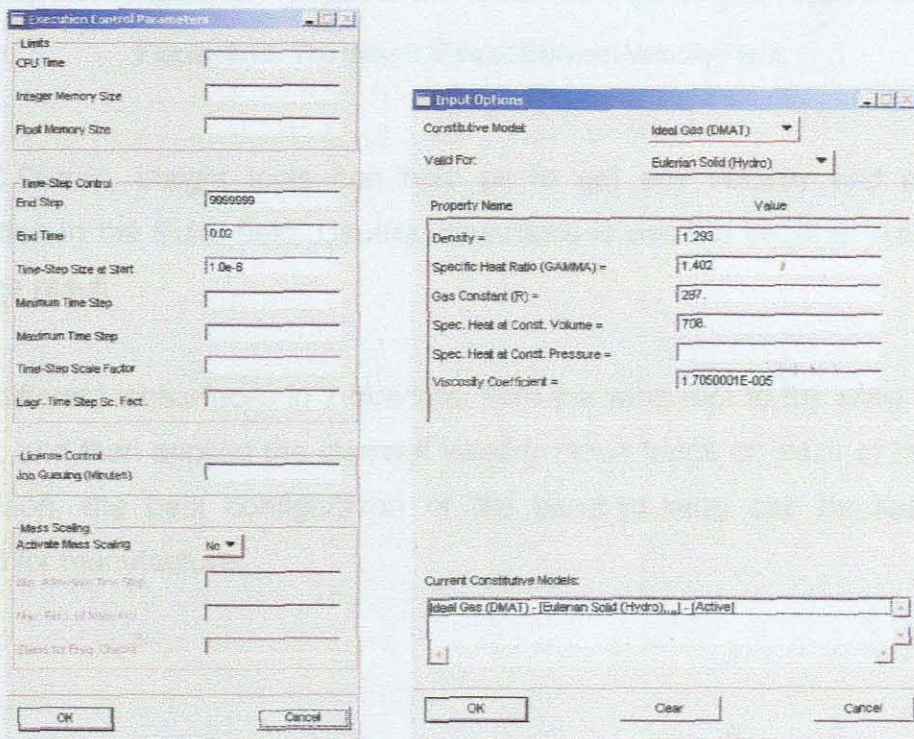


Figure 3.14: Some Parameters for MSC.Dytran simulation.

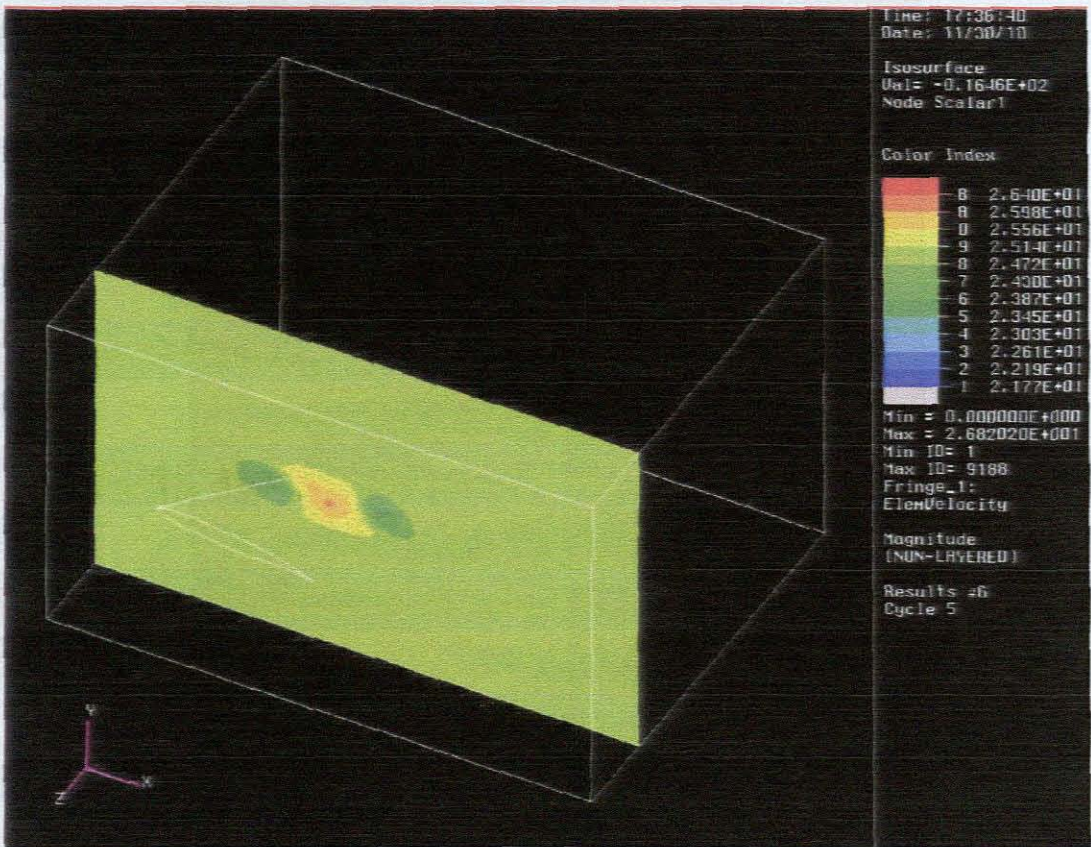


Figure 3.15: The Insight View of Element Velocity Field

In MSC.Dytran, insight tools can help us to get any velocity and pressure information in the Euler field. Usually, isosurface is used to become a carrier of the fringe result.

Seven different isosurfaces in z direction from the wing root to the wing tip were created, and then applied the element velocity result fringe on each of these. By comparison, the best configuration of the bending wing can be found and adopted for manufacture.

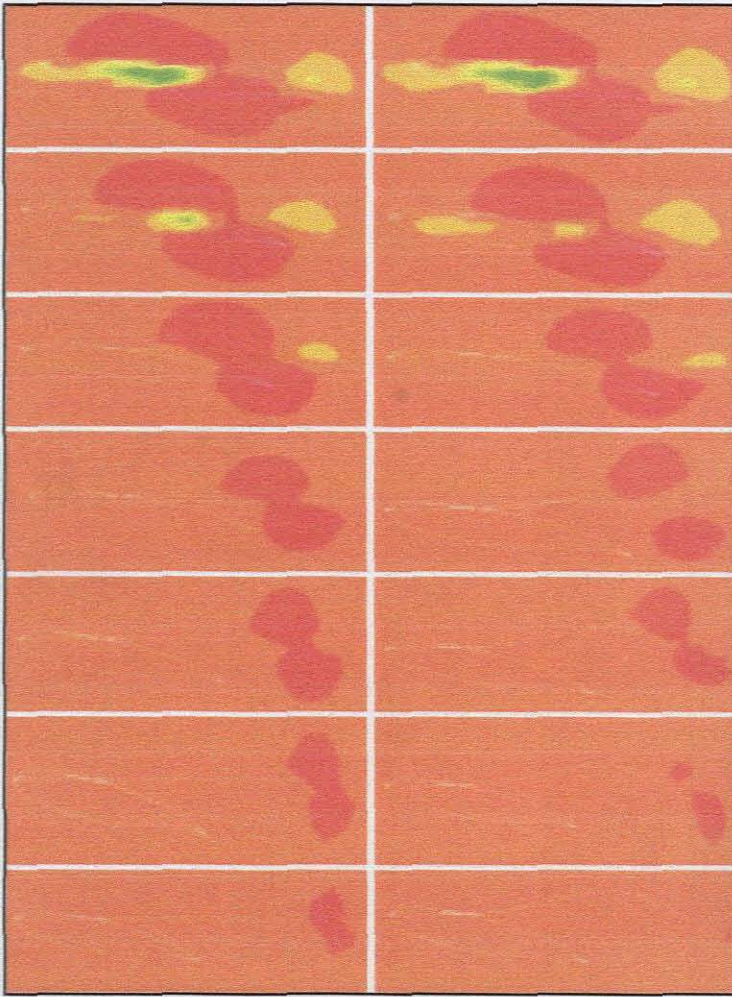


Figure 3.16: Comparison 1 of air flow speed between straight wing and bending wing (The wing in Figure 3.11)

Fig. 3.16 shows a velocity field comparison between a conventional straight wing and the conformable wing concept presented in fig. 3.11. The red parts in the graphics show the high velocity flow regions. According to the results, the velocity flow field around the wing root is similar for both wings. Other parts however show differences. The isosurfaces along the length of the bending wing show that the regions of high velocity move to its trailing edge. This means that a lower velocity is created at the lower surface of this bending wing, which means that an increase in pressure has occurred. This means that lift has improved.

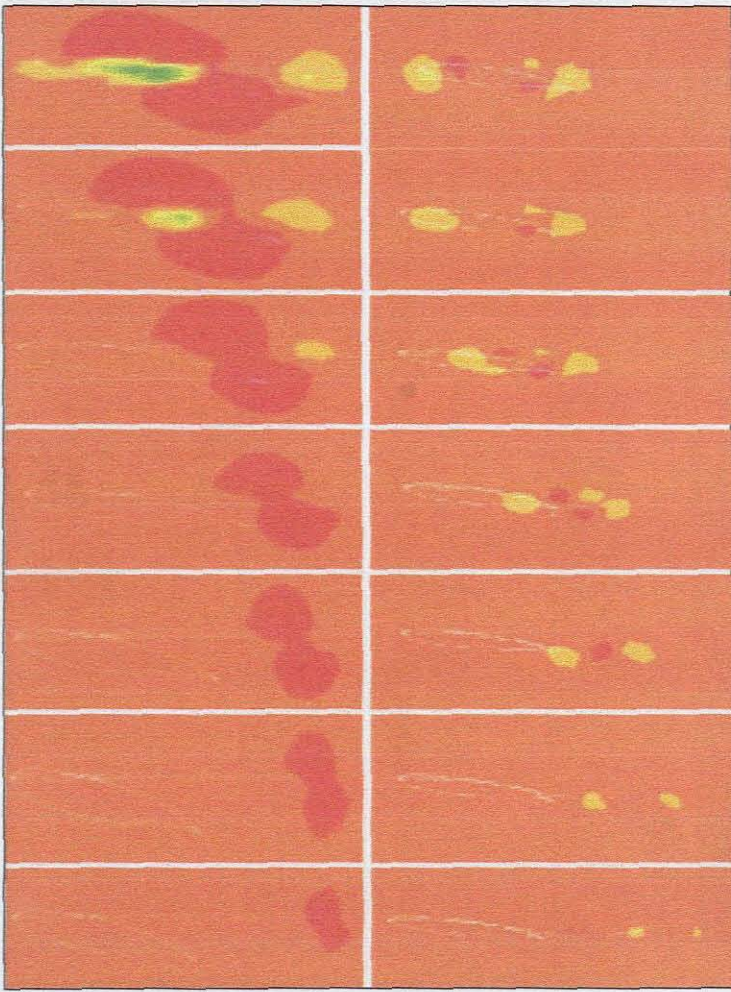


Figure 3.17: Comparison 2 of air flow speed between straight wing and bending wing (The wing in Figure 3.12)

The comparison presented in fig. 3.17 is that of the velocity flow-field around a conventional straight wing and the conformable wing concept given in fig. 3.12. Again the red regions show high velocity flow fields. The result of the conformable wing concept shows that lower velocity regions exist at its leading and trailing edges. This then shows that higher pressures exist at these positions. The isosurfaces at different sections along the length of this bending wing shows that the velocity profile remains the same.

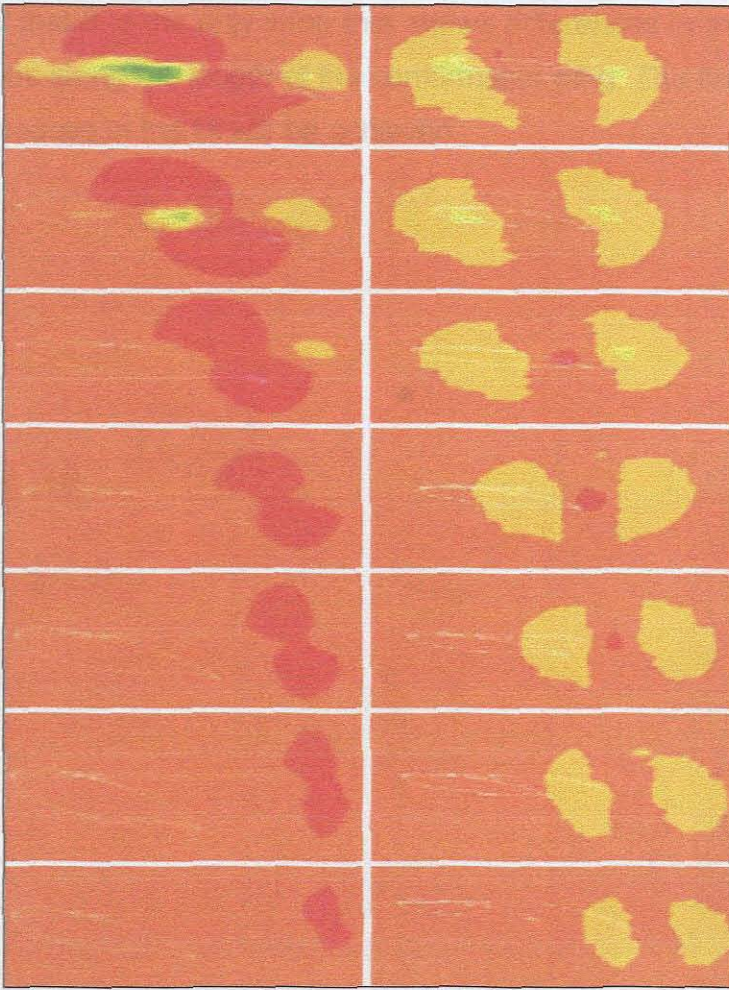


Figure 3.18: Comparison 3 of air flow speed between straight wing and bending wing (The wing in Figure 3.13)

The third comparison as shown in fig. 3.18 shows the velocity flow field results of a conventional straight wing and the conformable wing concept presented in fig. 3.13. This conformable wing concept shows similar velocity flow regions to that of the conformable wing concept presented in fig. 3.16. It typically shows that this conformable wing concept will have the same velocity profiles at different sections along its length.

The comparison of the velocity flow regions for the three conformable wing concepts shows that the first concept shows the most promising results. This is

due to the fact that it is the only wing that shows differences in the flow fields of different sections along the length of the conformable wing. The first concept thus shows that increases in the lift will be achieved.

Chapter 4

The Manufacture of the Conformable Wing

This Chapter introduces a model for the conformable wing. The computer model for the wing that was developed in Chapter 3 is used here as input to the Laminated Object Manufacturing rapid prototyping machine and prototypes of the model are created. These prototypes were used as “masters” or negatives in the vacuum casting process to develop silicon molds that would then be used to manufacture the plastic components for the conformable concept wing. The entire manufacturing procedure is described in this Chapter.

4.1. The LOM Process

Laminated Object Manufacturing (LOM) Rapid Prototyping is based upon the subtraction of materials as successive layers are formed [19] Fig. 4.1 shows the LOM machine that is housed at the Rapid Prototyping Center in the Department

of Mechanical Engineering (Bellville Campus) at the Cape Peninsula University of Technology. This is in contrast to other rapid prototype technologies, which are based upon the addition of layers.



Figure 4.1: LOM machine

In this system the model is formed from successive layers of heat bonded sheet material, typically paper [19]. The sliced CAD data is used to control a laser that cuts the perimeter (only) of each slice in the sheet material. A typical LOM machine is illustrated in figure 4.1 above and as a schematic in figure 4.2.

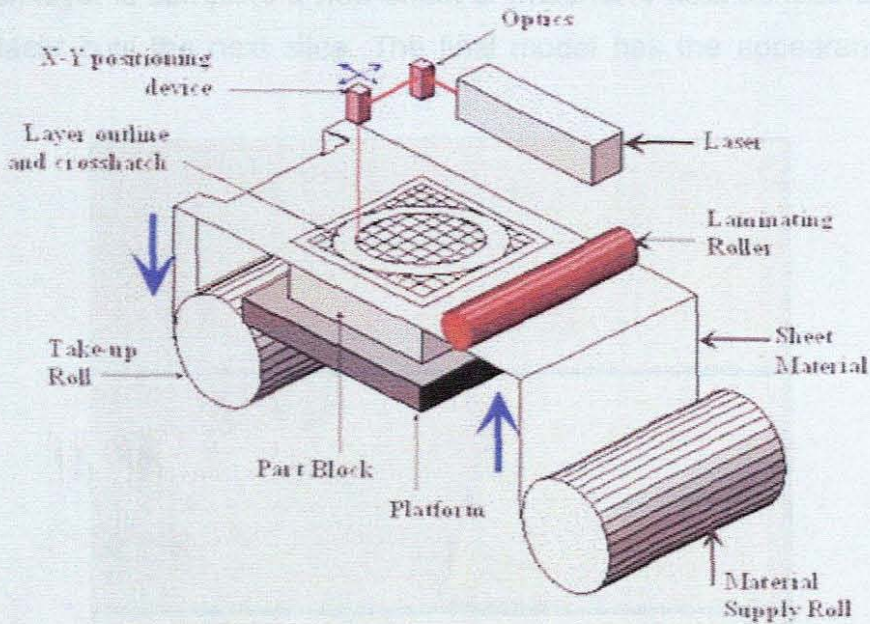


Figure 4.2: A typical LOM machine [19]

The waste material around the slice is left in place to support the next layer of the model but to assist with subsequent removal, it is cross-hatched by the laser to form blocks.

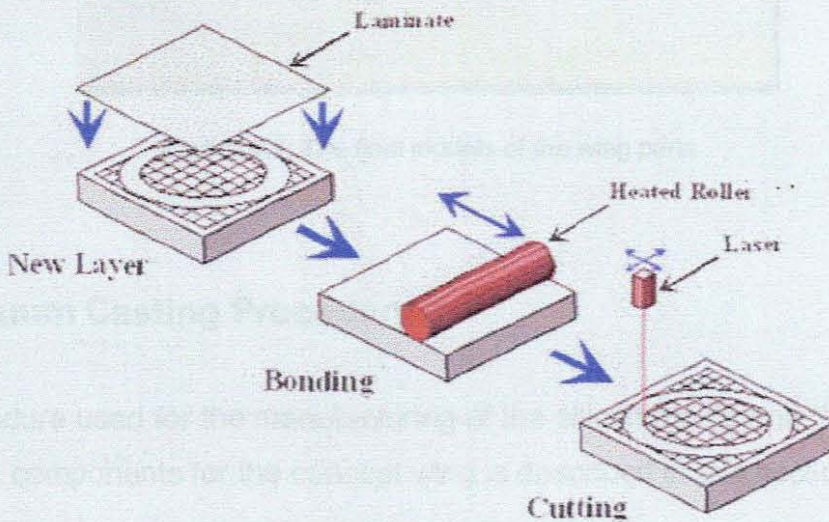


Figure 4.3: The procedure of RP in LOM [19]

Once each layer is complete a new sheet of material is heat bonded on top of it and the laser cuts the next slice. The final model has the appearance of soft wood.

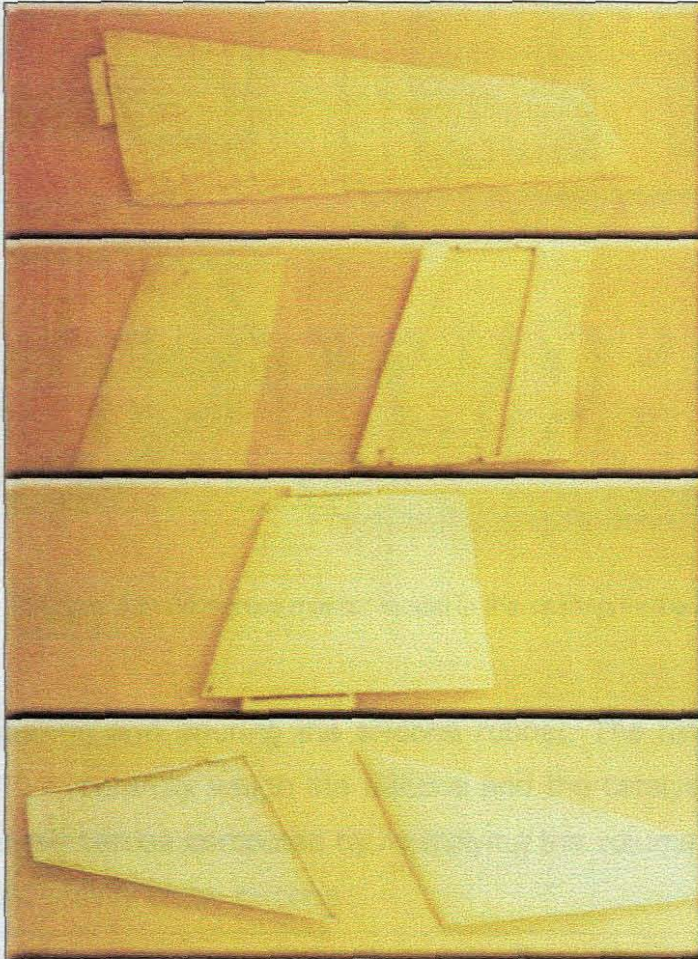


Figure 4.4: The final models of the wing parts

4.2. Vacuum Casting Procedure

The procedure used for the manufacturing of the silicon molds and the casting of the plastic components for the concept wing is described in this section.

1. Carefully clean the model with alcohol and create the parting line.
2. Fixing the master model in the casting frame.



Figure 4.5: Fixing the master model in the casting frame

3. Mixing, degassing and pouring the silicone rubber. The ratio of Silicone to Catalyst is 10:1; carefully weigh the silicone and the catalyst. The weight of silicone required can be computed by multiplying the volume of the mold box by the specific gravity of the silicone.



Figure 4.6: Mixing, degassing and pouring the silicone rubber [7]



Figure 4.7: Vacuum casting machine

Once the silicone and the catalyst are well mixed, place the container inside the Protomix System and turn ON the vacuum pump. The mixture will start bubbling and rising. It may be necessary to open and then close the vacuum vents in order to avoid overflowing. When the bubbles collapse and the mixture stops rising, the de-gassing should be continued for about 10 minutes. The total time for this step should not exceed 20 minutes.

Silicone should be carefully poured around the model. Pouring directly over the model and onto the venting lines should be avoided. When all the silicone is poured, the mold box should be placed inside the Protomix System under vacuum for about 45 minutes. During this time, most of the air bubbles will move to the surface.

4. Opening the mold

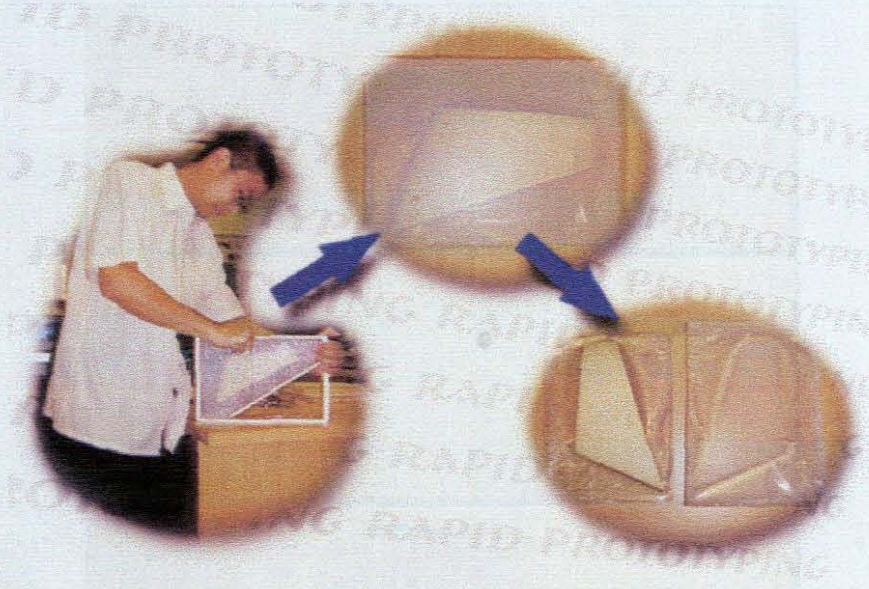


Figure 4.8: Opening the mold

5. Casting and heating the mold

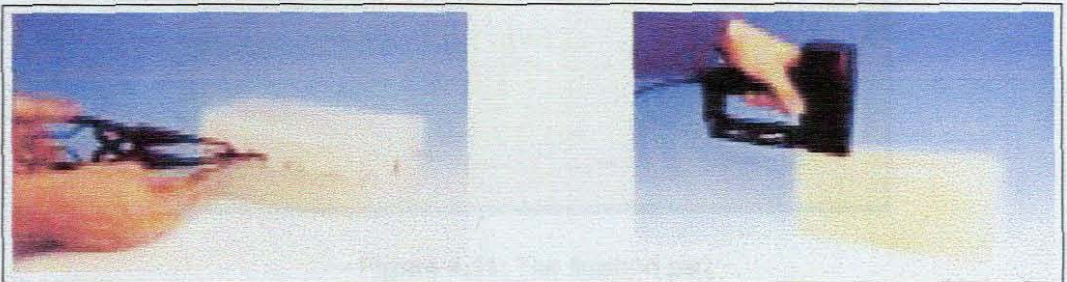


Figure 4.9: Casting and heating the mold [7]

6. Casting a part



Figure 4.10: Casting a part [7]

7. Removing the finished part

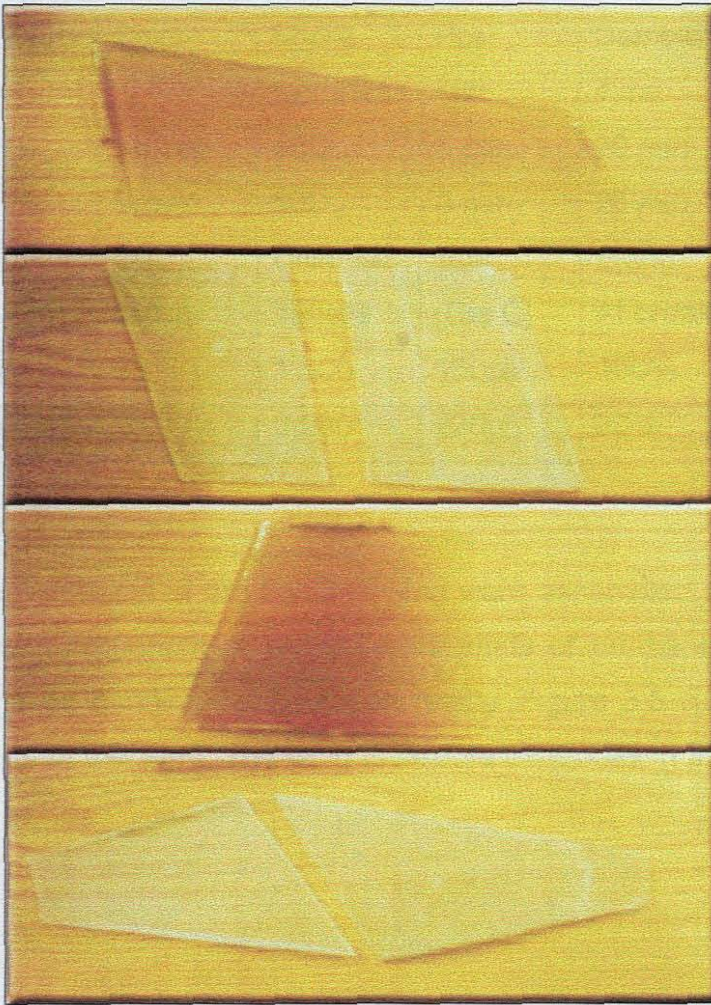


Figure 4.11: The finished part

The materials used in this research come from SMOOTH-ON, these kinds of Semi-Rigid urethane casting resins offer excellent impact resistance.

The technical overview of the thermo-plastic resins used in this manufacturing procedure is given in Table 4-1.

Table 4-1: The technical overview

Product	MixRatio (pdv)	Color	Shore Hardness	Pot Lift	Demold Time	Mixed Viscosity
SC 45D	1A:1B	Amber	45D	5 min.	20 min.	500 cps
SC 60D	1A:1B	Brown	60D	5 min.	20 min.	1500 cps
Product	G/CC	Cu In./Lb.	Ultimate Tensile	Tear (Die C)	Elong. At break	Shrink In./in.
SC 45D	1.08	26	1,500	150pli	100%	.002
SC 60D	1.05	26.4	2,200	220pli	30%	.015

The different components of the concept wing were assembled as shown in Fig 4.11; the different properties of two different kinds of urethane casting resins make the optimized bending configuration reality. Figure 3.5 clearly shows the simulation results of the wing. The wing is flexible especially in the section of the wing tip.

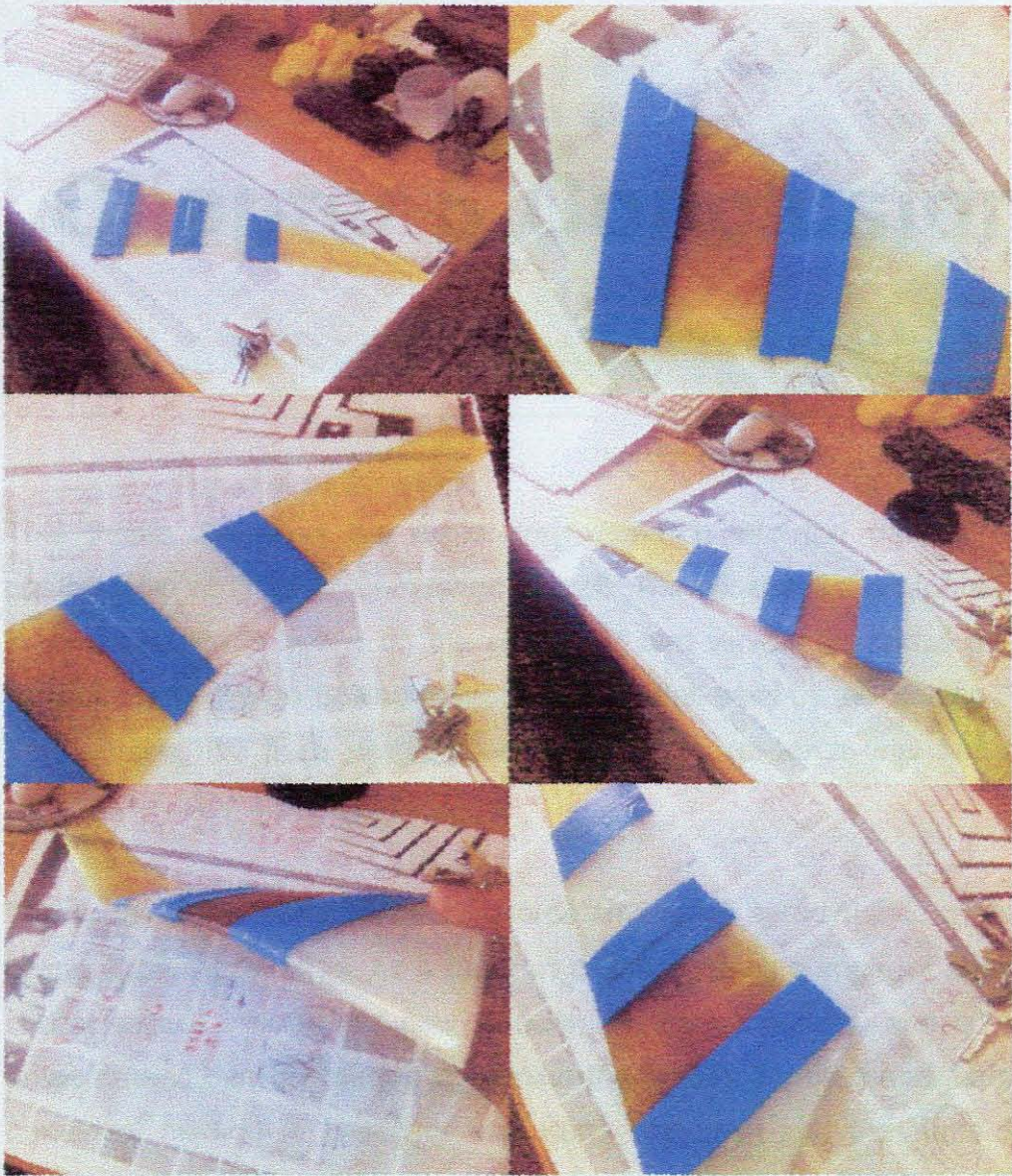


Figure 4.12: The assembling

Chapter 5 Conclusions

This thesis described the exploratory research into the development of a conformable wing for an unmanned aircraft. The concept for conformability used in this research is based on the shapes of bird wings. The airfoil used in this research is based on Boeing F supercritical airfoil used by the Boeing Company. Furthermore, the finite wing shape used in this research was based on the Boeing-747 wing shape [23].

The vortex-panel method was used to determine the aerodynamic characteristics of the supercritical airfoil as a function of the geometric angle of attack. These characteristics include the flow velocity, circulation density, and coefficient of pressure and can be used to determine the airfoil's lift, drag and moment. This numerical method is based on the Kutta Condition and the Kutta-Joukowski theorem. To facilitate the ease of these computations, a Python program was developed to perform the calculations. This program was developed for the intention of determining the aerodynamic characteristics of any body that experience lift when placed in a low speed flow field. This program will form the foundation for all further research in the conformable wing project. The Python program was validated with numerical results found in literature for the NACA 2412 airfoil. A very good agreement between the Python program results and that found in literature was achieved (see Table 2.1).

Seven different wing concepts for conformability based on the natural shapes of

bird wings were developed. The wing concepts are all modeled with rigid (aluminum) and flexible (rubber) parts to facilitate conformability. An initial set of finite element analyses was performed to determine possible deformed shapes for the seven different wing concepts. The results of these analyses were compared and the concept with the best displacement and rotation was selected. Further finite element analysis was performed on an initially straight wing and three deformed wing shapes to study the differences in velocity and pressure distributions over these wings. From these results, the deformed wing that would provide the best lift characteristics was selected.

A possible manufacturing procedure for the conformable wing was introduced. The three dimensional computer models for the wing were used as input for the Laminated Object Manufacturing rapid prototyping machine and prototypes of the model were created. These prototypes were used as negatives in the vacuum casting process to develop silicon molds that would be used to manufacture the plastic components for the conformable wing. The components were assembled and the entire wing was described (see Fig. 4.11). SMA wires were set in the silicon molds for the flexible components of the wing and the thermo-plastic resin was cast around them. The test for conformability was not conducted since it fell out of the scope of this research.

The work presented in this thesis does provide an entry level simulation tool for the design of conformable airfoils and finite wings. Furthermore it shows positive results for the lift of one conformable wing shape using the MSC suite of finite element software packages.

Future research following up the manufacture of a fully conformable wing, should involve Experimental investigations aimed at testing the conformal properties of the wing. Furthermore, wind tunnel experiments should be conducted to validate the simulation results.

Appendix A

Python Codes used for get the shape information NACA series aerofoil

```

from wxPython.lib.evntmgr import eventManager
from wxPython.wx import *
import random
import string

ID_MODIFY_MENU = wxNewId()
ID_FILE_MENU = wxNewId()
ID_BUTTON1 = wxNewId()
ID_BUTTON2 = wxNewId()
USE_BUFFERED_DC = 1
ID_EXIT_MENU = wxNewId()
BMP_ID = wxNewId()
PRINT_ID = wxNewId()
RBOX1 = wxNewId()
RBOX2 = wxNewId()
count=0

def yt(x,t):
    s=(t/0.2)*(0.2969*pow(x,0.5)-0.126*x-0.3537*pow(x,2)+0.2843*pow(x,3)-0.1015*pow(x,4))
    return s

def yc(x,m,p):
    if x<=p:
        s=(m/pow(p,2))*(2*p*x-pow(x,2))
    else:
        s=(m/pow(1-p,2))*((1-2*p)+2*p*x-pow(x,2))
    return s

class wxBufferedWindow(wxWindow):
    def __init__(self, parent, id,
                 pos = (320,50),

```

```
        size = (450,300),
        style = wxSIMPLE_BORDER ):
wxWindow.__init__(self, parent, id, pos, size, style)

EVT_PAINT(self, self.OnPaint)
EVT_SIZE(self, self.OnSize)
# OnSize called to make sure the buffer is initialized.
# This might result in OnSize getting called twice on some
# platforms at initialization, but little harm done.
self.OnSize(None)

def Draw(self,dc):
    ## just here as a place holder.
    ## This method should be over-ridden when sub-classed
    pass

def OnPaint(self, event):
    # All that is needed here is to draw the buffer to screen
    if USE_BUFFERED_DC:
        dc = wxBufferedPaintDC(self, self._Buffer)
    else:
        dc = wxPaintDC(self)
        dc.DrawBitmap(self._Buffer,0,0)

def OnSize(self,event):
    # The Buffer init is done here, to make sure the buffer is always
    # the same size as the Window
    self.Width, self.Height = self.GetClientSizeTuple()

    # Make new off screen bitmap: this bitmap will always have the
    # current drawing in it, so it can be used to save the image to
    # a file, or whatever.
    self._Buffer = wxEmptyBitmap(self.Width, self.Height)
    self.UpdateDrawing()

def SaveToFile(self,FileName,FileType):
    ## This will save the contents of the buffer
    ## to the specified file. See the wxWindows docs for
    ## wxBitmap::SaveFile for the details
    self._Buffer.SaveFile(FileName,FileType)

def PrintFile(frame, nb, log):
    self._Buffer.PrintFile(FileName,FileType)
```

```

def UpdateDrawing(self):
    if USE_BUFFERED_DC:
        dc = wxBufferedDC(wxClientDC(self), self._Buffer)
        self.Draw(dc)
    else:
        print "updating the drawing"
        # update the buffer
        dc = wxMemoryDC()
        dc.SelectObject(self._Buffer)

        self.Draw(dc)
        # update the screen
        #wxClientDC(self).Blit(0, 0, self.Width, self.Height, dc, 0, 0)

class DrawPanel( wxPanel ):
    def __init__(self,parent):
        wxPanel.__init__(self,parent,-1)

class DrawWindow(wxBufferedWindow):
    def __init__(self, parent, id = -1):
        ## Any data the Draw() function needs must be initialized before
        ## calling wxBufferedWindow.__init__, as it will call the Draw
        ## function.

        wxBufferedWindow.__init__(self, parent, id)

    def Draw(self, dc):
        dc.BeginDrawing()
        dc.SetBackground( wxBrush(wxColor(TestFrame.R.GetValue()),\
                                     TestFrame.G.GetValue()),\
                          TestFrame.B.GetValue()))
        dc.Clear() # make sure you clear the bitmap!

        # Here's the actual drawing code.
        dc.SetPen( wxPen("Black",2) )
        xc1=10
        yc1=100
        yt1=100
        yt11=100
        m=TestFrame.M
        t=TestFrame.T
        p=TestFrame.P
        ch=TestFrame.ch2
        if count==0:

```

```

    pass
else:
    dc.SetPen( wxPen(TestFrame.F,1) )
    for i in range(1,101):
        if ch==1:
            xc2=xc1+4
            yc2=-yc(i/100.0,m,p)*600+100
            yt2=yc2+(yt(i/100.0,t))*600/2
            yt22=yc2-(yt(i/100.0,t))*600/2

            if TestFrame.cb1==1:
                dc.DrawCircle(xc2,yt2,3)
                dc.DrawCircle(xc2,yt22,3)
            if TestFrame.cb2==1:
                dc.DrawLine( xc1,yc1,xc2,yc2 )
            if TestFrame.cb3==1:
                dc.DrawLine( xc1,yt1,xc2,yt2 )
                dc.DrawLine( xc1,yt11,xc2,yt22 )
            xc1=xc2
            yc1=yc2
            yt1=yt2
            yt11=yt22
    dc.EndDrawing()

class TestFrame(wxFrame):
    pp=0
    kk=1
    def __init__(self):
        wxFrame.__init__(self, NULL, -1, "Xing Zhe",
            wxDefaultPosition,
            wxSize(800,500), wxFRAME_TOOL_WINDOW )
        self.Centre(wxBOTH)
        panel=DrawPanel( self )
        panel.SetBackgroundColour('#eeeeee')
        self.panel=panel

# Set up the MenuBar
MenuBar = wxMenuBar()
file_menu = wxMenu()
file_menu.Append(ID_EXIT_MENU, "E&xittAlt-F4","Terminate the program")
EVT_MENU(self, ID_EXIT_MENU, self.OnQuit)
MenuBar.Append(file_menu, "&File")

draw_menu = wxMenu()

```

```

draw_menu.Append(BMP_ID, '&Save Drawing\tAlt-I,')
EVT_MENU(self, BMP_ID, self.SaveToFile)
MenuBar.Append(draw_menu, "&Output")

print_menu = wxMenu()
print_menu.Append(PRINT_ID, '&Print Drawing\tAlt-P,')
EVT_MENU(self, PRINT_ID, self.PrintFile)
MenuBar.Append(print_menu, "&Print")

self.SetMenuBar(MenuBar)

```

Set up labels and label_inputs

```

sizer = wxBoxSizer(wxVERTICAL)
rb = wxStaticBox(panel, RBOX1, "NACA 4 Series Airfoil Number",
                (20,20), (270,250),
                wxRA_SPECIFY_COLS, "Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

sizer = wxBoxSizer(wxVERTICAL)
rb = wxStaticBox(panel, RBOX1, "Setting For The Figure",
                (20,280), (270,180),
                wxRA_SPECIFY_COLS, "Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

rb = wxStaticBox(panel, RBOX1, "Front Color",
                (150,300), (100,50),
                wxRA_SPECIFY_COLS, "Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

rb = wxStaticBox(panel, RBOX1, "Number",
                (40,40), (100,50),
                wxRA_SPECIFY_COLS, "Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

rb = wxStaticBox(panel, RBOX1, "Clickable",

```

```

        (40,100), (220,150),
        wxRA_SPECIFY_COLS,"Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

rb = wxStaticBox(panel, RBOX1, "NACA 4 Digit Airfoil Plot",
        (300,20), (490,440),
        wxRA_SPECIFY_COLS,"Xing Zhe")
#rb.SetBackgroundColour(wxBLUE)
#rb.SetLabel("wxRadioBox")
sizer.Add(rb, 0, wxALL, 20)

self.SetSizer(sizer)
#spin-----
wxStaticText(panel, -1, "Thickness divide chord %.",
        wxPoint(110, 150))
sc1 = wxSpinCtrl(panel, -1, "", wxPoint(60, 150), wxSize(40, -1))
self.sc1=sc1
sc1.SetRange(1,100)
sc1.SetValue(15)

wxStaticText(panel, -1, "The maximum camber %.",
        wxPoint(110, 180))
sc2 = wxSpinCtrl(panel, -1, "", wxPoint(60, 180), wxSize(40, -1))
self.sc2=sc2
sc2.SetRange(1,100)
sc2.SetValue(2)

wxStaticText(panel, -1, "The maximum camber position.",
        wxPoint(110, 210))
sc3 = wxSpinCtrl(panel, -1, "", wxPoint(60, 210), wxSize(40, -1))
self.sc3=sc3
sc3.SetRange(1,100)
sc3.SetValue(40)
#checkbox-----
cid = wxNewId()
cb1 = wxCheckBox(panel, cid, " Draw Circles", wxPoint(350, 360), wxSize(150, 20),
wxNO_BORDER)
cb1.SetBackgroundColour("")
self.cb1=cb1

cb2 = wxCheckBox(panel, cid+1, " Draw Camber Axes", wxPoint(350, 390), wxSize(150, 20),
wxNO_BORDER)

```

```

self.cb2=cb2
cb2.SetValue(True)
cb3 = wxCheckBox(panel, cID+2, " Draw Thickness Lines", wxPoint(350, 420), wxSize(150, 20),
wxNO_BORDER)
self.cb3=cb3

b = wxButton(panel, 10, "Restore View", wxPoint(650, 360))
EVT_BUTTON(self, 10, self.OnSubmit)
#List
sampleList1 = ['Red','White','Black','Green','Yellow','Blue','purple','pink','brown']
self.ch1 = wxChoice(panel, 40 ,(170, 320),choices = sampleList1)
self.ch1.SetSelection(2)
EVT_CHOICE(self, 40, self.EvtChoice)

sampleList2 = ['2415','4415']
EVT_CHOICE(self, 40, self.EvtChoice)

sampleList2 = ['2415','4415']
ch2 = wxChoice(panel, 40 ,(65, 60),choices = sampleList2)
self.ch2=ch2
self.ch2.SetSelection(1)
#Slider
#self.count = 0
wxStaticText(panel, -1, "Red:", wxPoint(30, 370))
self.slider_red = wxSlider(panel, 255, 255, 0, 255, wxPoint(80, 370),
wxSize(200, -1),
wxSL_HORIZONTAL | wxSL_LABELS )
wxStaticText(panel, -1, "Green:", wxPoint(30, 430))
self.slider_green = wxSlider(panel, 255, 255, 0, 255, wxPoint(80, 430),
wxSize(200, -1),
wxSL_HORIZONTAL | wxSL_LABELS )
wxStaticText(panel, -1, "Blue:", wxPoint(30, 400))
self.slider_blue = wxSlider(panel, 255, 255, 0, 255, wxPoint(80, 400),
wxSize(200, -1),
wxSL_HORIZONTAL | wxSL_LABELS )

self.__class__.R=self.slider_red
self.__class__.G=self.slider_green
self.__class__.B=self.slider_blue
self.__class__.F='Red'
self.__class__.cb1=self.cb1.IsChecked()
self.__class__.cb2=self.cb2.IsChecked()
self.__class__.cb3=self.cb3.IsChecked()
self.__class__.T=float(self.sc1.GetValue())/100

```

```
self.__class__.M=float(self.sc2.GetValue())/100
self.__class__.P=float(self.sc3.GetValue())/100
self.__class__.ch2=self.ch2.GetSelection()
self.Window = DrawWindow(self)

EVT_COMMAND_SCROLL_THUMBTRACK(self,-1,self.NewDrawing)
```

```
def EvtChoice(self, event):
    self.__class__.F=event.GetString()
    self.Window.UpdateDrawing()
```

```
def OnQuit(self,event):
    self.Close(true)
```

```
def NewDrawing(self,event):
    #self.Window.DrawData = self.MakeNewData()
    self.Window.UpdateDrawing()
```

```
def SaveToFile(self,event):
    dlg = wxFileDialog(self, "Choose a file name to save the image as a PNG to",
                      defaultDir = "",
                      defaultFile = "",
                      wildcard = "*.png",
                      style = wxSAVE)
    if dlg.ShowModal() == wxID_OK:
        print self.Window
        self.Window.SaveToFile(dlg.GetPath(),wxBITMAP_TYPE_PNG)
    dlg.Destroy()
```

```
def PrintFile(frame, nb):
    data = wxPrintDialogData()
    print 'sdfsdfs'
    data.EnableSelection(True)
    data.EnablePrintToFile(True)
    data.EnablePageNumbers(True)
    data.SetMinPage(1)
    data.SetMaxPage(5)
    data.SetAllPages(True)

    dlg = wxPrintDialog(frame, data)
    if dlg.ShowModal() == wxID_OK:
        data = dlg.GetPrintDialogData()
        log.WriteText('GetAllPages: %d\n' % data.GetAllPages())
    dlg.Destroy()
```

```
def OnSubmit(self, event):
    global count
    count=count+1
    self.__class__.cb1=self.cb1.IsChecked()
    self.__class__.cb2=self.cb2.IsChecked()
    self.__class__.cb3=self.cb3.IsChecked()
    self.__class__.T=float(self.sc1.GetValue())/100
    self.__class__.M=float(self.sc2.GetValue())/100
    self.__class__.P=float(self.sc3.GetValue())/100
    self.__class__.ch2=self.ch2.GetSelection()
    #print self.ch2.GetSelection()
    self.Window.UpdateDrawing()

def runTest(frame, nb, log):
    dlg = wxMessageDialog(frame, 'You input values have exceed elasticity limit.\nYours Zee',\
        'Z.XING', wxOK | wxICON_INFORMATION)
        #wxYES_NO | wxNO_DEFAULT | wxCANCEL | wxICON_INFORMATION)
    dlg.ShowModal()
    dlg.Destroy()

class DemoApp(wxApp):
    def OnInit(self):
        wxInitAllImageHandlers() # called so a PNG can be saved
        frame = TestFrame()
        frame.Show(true)

        ## initialize a drawing
        ## It doesn't seem like this should be here, but the Frame does
        ## not get sized until Show() is called, so it doesn't work if
        ## it is put in the __init__ method.
        frame.NewDrawing(None)
        self.SetTopWindow(frame)
        return true

if __name__ == "__main__":
    app = DemoApp(0)
    app.MainLoop()
```

Appendix B

Python Codes used for get pressure coefficient of NACA series aerofoil and Boeing F aerofoil

```

from numarray import *

import numarray.linear_algebra as la

from math import *

#M=20

#Dimension of NACA 2412-----
#XB=array([1,0.933,0.750,0.5,0.25,0.067,0,0.067,0.25,0.5,0.75,0.933,1])
#YB=array([0,-0.005,-0.017,-0.033,-0.042,-0.033,0,0.045,0.076,0.072,0.044,0.013,0])

#Dimension of Boeing f-----
""XB=array([1.0,0.98,0.95,0.9,0.7,0.45,0.22,0.1,0.012,0.004,0,0.004,0.012,0.1,0.22,0.45,0.7,0.9,0.95,0.9
8,1])
YB=array([0.00095,0.001491,0.00334,0.00388,-0.01768,-0.04338,-0.04445,-0.03739,-0.01688,-
0.01062,0,0.01061,0.01624,0.03617,0.04788,0.056341,0.048471,0.02199,0.01183,0.00534,0.00095])

ALPHA=10*pi/180

X=zeros((M),type=Float32)

Y=zeros((M),type=Float32)

S=zeros((M),type=Float32)

V=zeros((M),type=Float32)

CP=zeros((M),type=Float32)

```

```

THETA=zeros((M),type=Float32)
RHS=zeros((M+1),type=Float32)
GAMA=zeros((M+1),type=Float32)
CN1=array([0.0,0.0])
CN1=resize(CN1,(M,M))
CN2=array([0.0,0.0])
CN2=resize(CN2,(M,M))
CT1=array([0.0,0.0])
CT1=resize(CT1,(M,M))
CT2=array([0.0,0.0])
CT2=resize(CT2,(M,M))
AN=array([0.0,0.0])
AN=resize(AN,(M+1,M+1))
AT=array([0.0,0.0])
AT=resize(AT,(M+1,M+1))

"""def DETERM(ARRAY,N):"""
"""ARRAY=array([0.0,0.0])
    ARRAY=resize(ARRAY,(M+1,M+1))"""
"""A=array([0.0,0.0])
    A=resize(A,(N,N))
    for i in range(0,N):
        for j in range(0,N):
            A[i][j]=ARRAY[i][j]
    L=0
    while(L<N):
        K=L+1
        for i in range(K,N):
            RATIO=A[i][L]/A[L][L]
            for j in range(K,N):

```

```
A[i][j]=A[i][j]-A[L][j]*RATIO
```

```
L=L+1
```

```
DETERM=1.0
```

```
for L in range(0,N):
```

```
    DETERM=DETERM*A[L][L]"
```

```
""DETERM=la.determinant(ARRAY)
```

```
return DETERM
```

#For a given system of linear equations, Cramer's Rule is a handy way to solve the whole system of equations.

```
def CRAMER(C,A,X,N):
```

```
    DENOM=DETERM(C,N)
```

```
    CC=array([0.0,0.0])
```

```
    CC=resize(CC,(N+1,N+1))
```

```
    for K in range(0,N+1):
```

```
        for i in range(0,N+1):
```

```
            for j in range(0,N+1):
```

```
                CC[i][j]=C[i][j]
```

```
            for i in range(0,N+1):
```

```
                CC[i][K]=A[i]
```

```
            X[K]=DETERM(CC,N)/DENOM
```

```
    return X
```

```
for i in range(0,M):
```

```
    X[i]=0.5*(XB[i]+XB[i+1])
```

```
    Y[i]=0.5*(YB[i]+YB[i+1])
```

```
    S[i]=sqrt((XB[i+1]-XB[i])**2+(YB[i+1]-YB[i])**2)
```

```
    THETA[i]=atan2((YB[i+1]-YB[i]),(XB[i+1]-XB[i]))
```

```
    RHS[i]=sin(THETA[i]-ALPHA)
```

```
for i in range(0,M):
```

for j in range(0,M):

if i==j:

$$CN1[i][j]=1.0$$

$$CN2[i][j]=1.0$$

$$CT1[i][j]=0.5*\pi$$

$$CT2[i][j]=0.5*\pi$$

else:

$$A=(X[i]-XB[j])*cos(THETA[j])-(Y[i]-YB[j])*sin(THETA[j])$$

$$B=(X[i]-XB[j])**2+(Y[i]-YB[j])**2$$

$$C=sin(THETA[i]-THETA[j])$$

$$D=cos(THETA[i]-THETA[j])$$

$$E=(X[i]-XB[j])*sin(THETA[j])-(Y[i]-YB[j])*cos(THETA[j])$$

$$F=log(1.0+S[j]*(S[j]+2*A)/B)$$

$$G=atan2((E*S[j]),(B+A*S[j]))$$

$$P=(X[i]-XB[j])*sin(THETA[i]-2*THETA[j])+(Y[i]-YB[j])*cos(THETA[i]-2*THETA[j])$$

$$Q=(X[i]-XB[j])*cos(THETA[i]-2*THETA[j])-(Y[i]-YB[j])*sin(THETA[i]-2*THETA[j])$$

$$CN2[i][j]=D+0.5*Q/F/S[j]-(A*C+D*E)*G/S[j]$$

$$CN1[i][j]=0.5*D*F+C*G-CN2[i][j]$$

$$CT2[i][j]=C+0.5*P/F/S[j]+(A*D-C*E)*G/S[j]$$

$$CT1[i][j]=0.5*C*F-D*G-CT2[i][j]$$

for i in range(0,M):

$$AN[i][0]=CN1[i][0]$$

$$AN[i][M]=CN2[i][M-1]$$

$$AT[i][0]=CT1[i][0]$$

$$AT[i][M]=CT2[i][M-1]$$

for j in range(1,M):

$$AN[i][j]=CN1[i][j]+CN2[i][j-1]$$

$$AT[i][j]=CT1[i][j]+CT2[i][j-1]$$

$$AN[M][0]=1.0$$

$$AN[M][M]=1.0$$

```
for j in range(1,M):
    AN[M][j]=0.0
RHS[M]=0.0

GAMA=CRAMER(AN,RHS,GAMA,M)

#print GAMA
#GAMA=array([-0.0823,-0.1403,-0.1422,-0.1413,-0.1334,-
0.0981,0.217,0.2785,0.2401,0.2098,0.1843,0.1578,0.0823])

for i in range(0,M):
    V[i]=cos(THETA[i]-ALPHA)
    for j in range(0,M+1):
        V[i]=V[i]+AT[i][j]*GAMA[j]
    CP[i]=1.0-V[i]**2

print CP

output_file=open("Boeing_10.dat","w")

for i in range(0,M):
    output_file.writelines(str(XB[i])+','+str(CP[i])+'\n')

output_file.close() "
```

Appendix C

Python Codes Used for Get Lift Coefficient and Induce Drag Coefficient for Finite Wing Model

```
from numarray import *

import numarray.linear_algebra as la

from math import *

""output_file=open("Boeing_10.dat","w")

for i in range(0,M):

    output_file.writelines(str(XB[i])+','+str(CP[i])+'\n')

output_file.close() ""

# The parameters of the wing model can be found in last sector of chapter2-----

k=18

AR=7.173445297

LAMBDA=0.239986468

RHO=1.226

TWIST=0

V=250.0

AREA=0.1872

THETA=zeros((k),type=Float32)
```

C=zeros((k),type=Float32)

Y=zeros((k),type=Float32)

A=zeros((k),type=Float32)

M=zeros((k),type=Float32)

ALABS=zeros((k),type=Float32)

ALIND=zeros((k),type=Float32)

CDIND=zeros((k),type=Float32)

CL=zeros((k),type=Float32)

CLm4=zeros((k),type=Float32)

CLo=zeros((k),type=Float32)

CL1=zeros((k),type=Float32)

CL2=zeros((k),type=Float32)

CL3=zeros((k),type=Float32)

CL4=zeros((k),type=Float32)

CL5=zeros((k),type=Float32)

CL6=zeros((k),type=Float32)

CLA=zeros((k),type=Float32)

CLB=zeros((k),type=Float32)

D=array([0.0,0.0])

D=resize(D,(k,k))

ALm4=4.0+2

ALo=0.0+2

AL1=3.0+2

AL2=6.0+2

AL3=7.0+2

AL4=8.0+2

AL5=9.0+2

AL6=10.0+2

```
def DETERM(ARRAY,N):
```

```
    DETERM=la.determinant(ARRAY)
```

```
    return DETERM
```

#For a given system of linear equations, Cramer's Rule is a handy way to solve the whole system of equations.

```
def CRAMER(C,A,X,N):
```

```
    DENOM=DETERM(C,N)
```

```
    CC=array([0.0,0.0])
```

```
    CC=resize(CC,(N,N))
```

```
    for K in range(0,N):
```

```
        for i in range(0,N):
```

```
            for j in range(0,N):
```

```
                CC[i][j]=C[i][j]
```

```
            for i in range(0,N):
```

```
                CC[i][K]=A[i]
```

```
            X[K]=DETERM(CC,N)/DENOM
```

```
    return X
```

```
for j in range(0,k):
```

```
    THETA[j]=pi*(j+1)/(2.0*k)
```

```
    Y[j]=cos(THETA[j])
```

```
    C[j]=1.0-(1.0-LAMBDA)*cos(THETA[j])
```

```
for j in range(0,k):
```

```
    D1=1.0/C[j]
```

```
    D2=pi/(AR*(1.0+LAMBDA)*sin(THETA[j]))
```

```
for n in range(0,k):
```

```
    l=2*n+1
```

```
    D[j][n]=(D1+D2*l)*sin(l*THETA[j])
```

```
#AL1
```

```
for j in range(0,k):
```

```
ALABS[j]= (AL1+TWIST*cos(THETA[j]))*pi/180
```

```
A=CRAMER(D,ALABS,A,k)
```

```
CLW1=pi**2*A[0]/(1.0+LAMBDA)
```

```
for j in range(0,k):
```

```
    SUM=0.0
```

```
    for n in range(0,k):
```

```
        SUM=SUM+A[n]*sin((2*n+1)*THETA[j])
```

```
    CL1[j]=2*pi/C[j]*SUM
```

```
#AL2-----
```

```
for j in range(0,k):
```

```
    ALABS[j]= (AL2+TWIST*cos(THETA[j]))*pi/180
```

```
A=CRAMER(D,ALABS,A,k)
```

```
CLW2=pi**2*A[0]/(1.0+LAMBDA)
```

```
for j in range(0,k):
```

```
    SUM=0.0
```

```
    for n in range(0,k):
```

```
        SUM=SUM+A[n]*sin((2*n+1)*THETA[j])
```

```
    CL2[j]=2*pi/C[j]*SUM
```

```
#AL3-----
```

```
for j in range(0,k):
```

```
    ALABS[j]= (AL3+TWIST*cos(THETA[j]))*pi/180
```

```
A=CRAMER(D,ALABS,A,k)
```

```
CLW3=pi**2*A[0]/(1.0+LAMBDA)
```

```
for j in range(0,k):
```

```
    SUM=0.0
```

```
    for n in range(0,k):
```

```
        SUM=SUM+A[n]*sin((2*n+1)*THETA[j])
```

$$CL3[j]=2*\pi/C[j]*SUM$$

#AL4

for j in range(0,k):

$$ALABS[j]=(AL4+TWIST*\cos(THETA[j]))*\pi/180$$

A=CRAMER(D,ALABS,A,k)

$$CLW4=\pi**2*A[0]/(1.0+LAMBDA)$$

for j in range(0,k):

SUM=0.0

for n in range(0,k):

$$SUM=SUM+A[n]*\sin((2*n+1)*THETA[j])$$

$$CL4[j]=2*\pi/C[j]*SUM$$

#AL4

for j in range(0,k):

$$ALABS[j]=(AL4+TWIST*\cos(THETA[j]))*\pi/180$$

A=CRAMER(D,ALABS,A,k)

$$CLW4=\pi**2*A[0]/(1.0+LAMBDA)$$

for j in range(0,k):

SUM=0.0

for n in range(0,k):

$$SUM=SUM+A[n]*\sin((2*n+1)*THETA[j])$$

$$CL4[j]=2*\pi/C[j]*SUM$$

#AL5

for j in range(0,k):

$$ALABS[j]=(AL5+TWIST*\cos(THETA[j]))*\pi/180$$

A=CRAMER(D,ALABS,A,k)

$$CLW5=\pi**2*A[0]/(1.0+LAMBDA)$$

for j in range(0,k):

SUM=0.0

for n in range(0,k):

SUM=SUM+A[n]*sin((2*n+1)*THETA[j])

CL5[j]=2*pi/C[j]*SUM

#AL6

for j in range(0,k):

ALABS[j]=(AL6+TWIST*cos(THETA[j]))*pi/180

A=CRAMER(D,ALABS,A,k)

CLW6=pi**2*A[0]/(1.0+LAMBDA)

for j in range(0,k):

SUM=0.0

for n in range(0,k):

SUM=SUM+A[n]*sin((2*n+1)*THETA[j])

CL6[j]=2*pi/C[j]*SUM

#ALo

for j in range(0,k):

ALABS[j]=(ALo+TWIST*cos(THETA[j]))*pi/180

A=CRAMER(D,ALABS,A,k)

CLW0=pi**2*A[0]/(1.0+LAMBDA)

for j in range(0,k):

SUM=0.0

for n in range(0,k):

SUM=SUM+A[n]*sin((2*n+1)*THETA[j])

CLo[j]=2*pi/C[j]*SUM

#AL6

for j in range(0,k):

ALABS[j]=(ALm4+TWIST*cos(THETA[j]))*pi/180

A=CRAMER(D,ALABS,A,k)

```

CLWm4=pi**2*A[0]/(1.0+LAMBDA)

for j in range(0,k):
    SUM=0.0
    for n in range(0,k):
        SUM=SUM+A[n]*sin((2*n+1)*THETA[j])
    CLm4[j]=2*pi/C[j]*SUM

#print CL1,CLW1
#print CL2,CLW2
#print CL3,CLW3
#print CL4,CLW4
#print CL5,CLW5
#print CL6,CLW6

"output_file=open("Wing_m4.dat","w")
for i in range(0,k):
    output_file.writelines(str(CLm4[i])+'\n')
output_file.writelines('\n'+str(CLWm4)+'\n')
output_file.close()"

CLWF=0.5*RHO*(V*1000.0/3600.0)**2*AREA*CLW4
ALF=AL1+(AL2-AL1)*(CLWF-CLW1)/(CLW2-CLW1)

for j in range(0,k):
    CL[j]=CLB[j]+CLA[j]*CLWF

for j in range(0,k):
    ALABS[j]=(ALF+TWIST*cos(THETA[j]))*pi/180.0

A=CRAMER(D,ALABS,A,k)

for j in range(0,k):
    SUM=0.0
    for n in range(0,k):
        SUM=SUM+A[n]*sin((2*n+1)*THETA[j])

```

```

CL[j]=2.0*pi/C[j]*SUM
for j in range(0,k):
    SUM=0.0
    for n in range(0,k):
        l=2*n+1
        SUM=SUM+l*A[n]*sin(l*THETA[j])/sin(THETA[j])
    ALIND[j]=-p/(AR*(1.0+LAMBDA))*SUM
    CDIND[j]=-CL[j]*ALIND[j]
    M[j]=CL[j]/ALABS[j]
    ALIND[j]=ALIND[j]*180.0/pi

SUM=0.0
for n in range(0,k):
    SUM=SUM+(2*n+1)*A[n]**2
CDINDW=pi**3/(AR*(1.0+LAMBDA)**2)*SUM
SUM=0.0
KM1=k-1
for j in range(0,KM1):
    SUM=SUM+(M[j]*C[j]+M[j+1]*C[j+1])*(Y[j]-Y[j+1])
MBAR=SUM/(1.0+LAMBDA)
ALABSW=CLWF/MBAR*180.0/pi

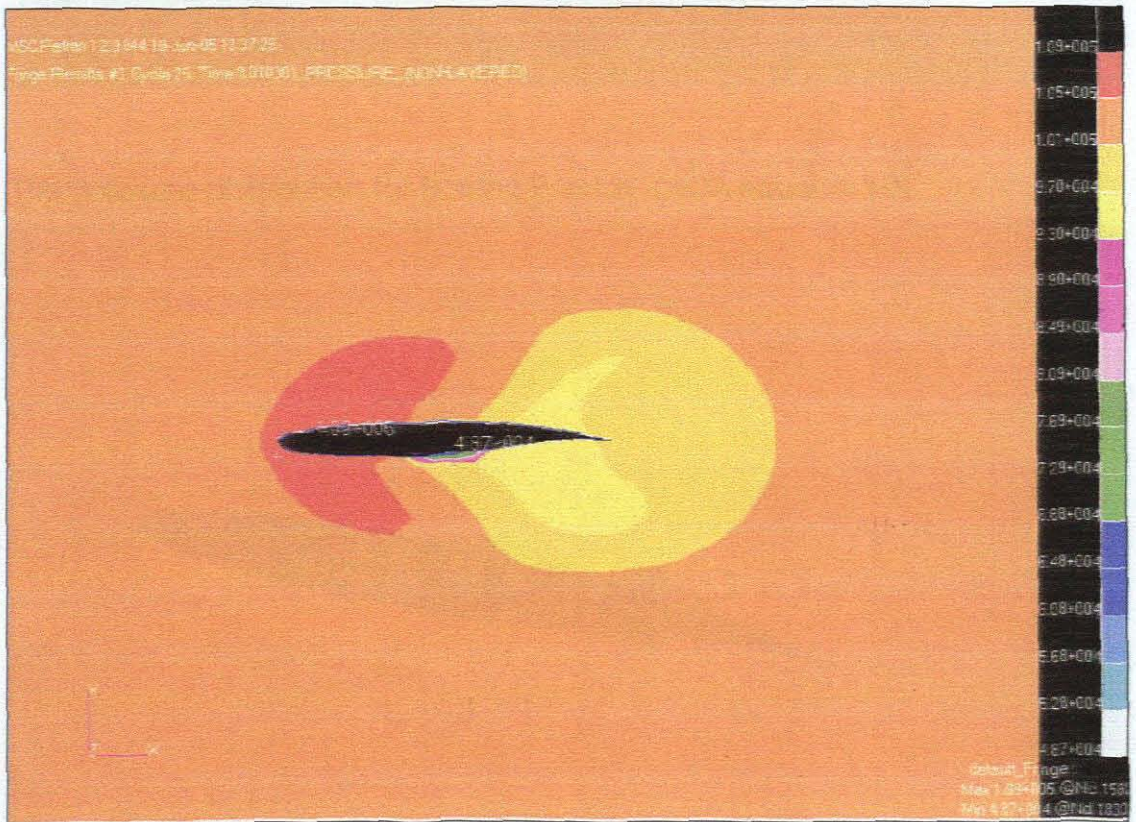
print ALF

```


Appendix D

Some Graphics Come From MSC.Dytran Analyses

The pressure distribution for Boeing F airfoil. (AOA equal to 0)



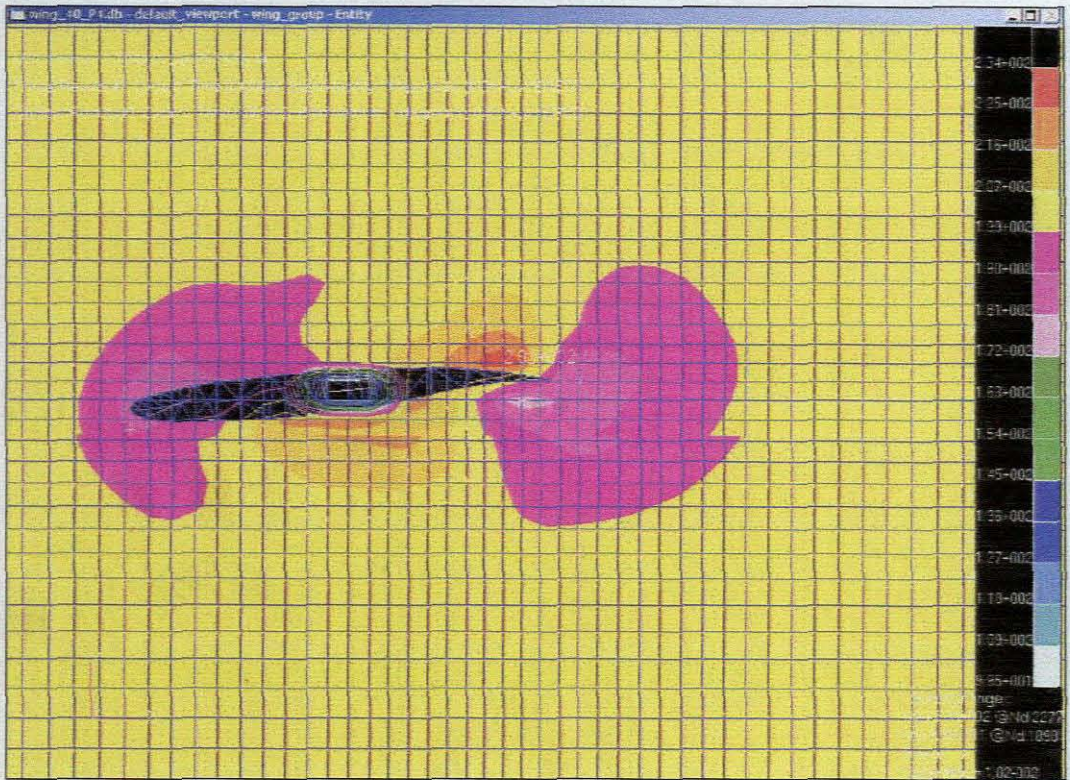
The pressure distribution for Boeing F airfoil. (AOA equal to 8)



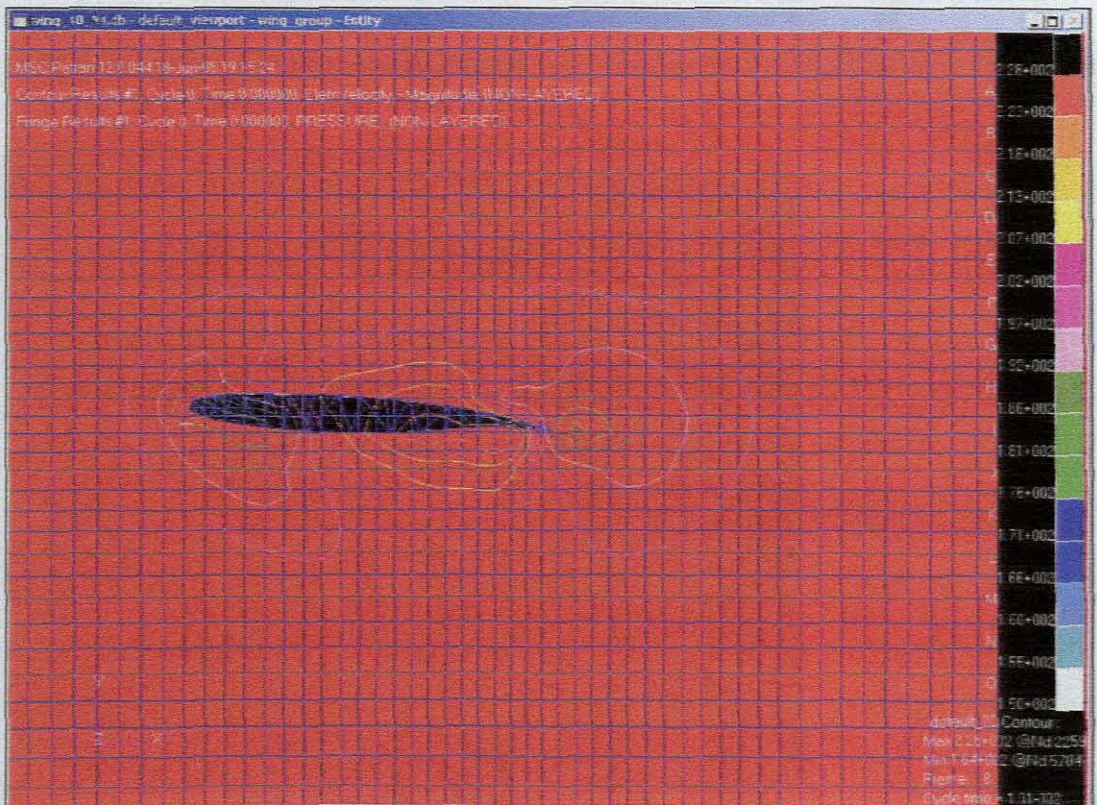
The pressure distribution for Boeing F airfoil. (AOA equal to 12)



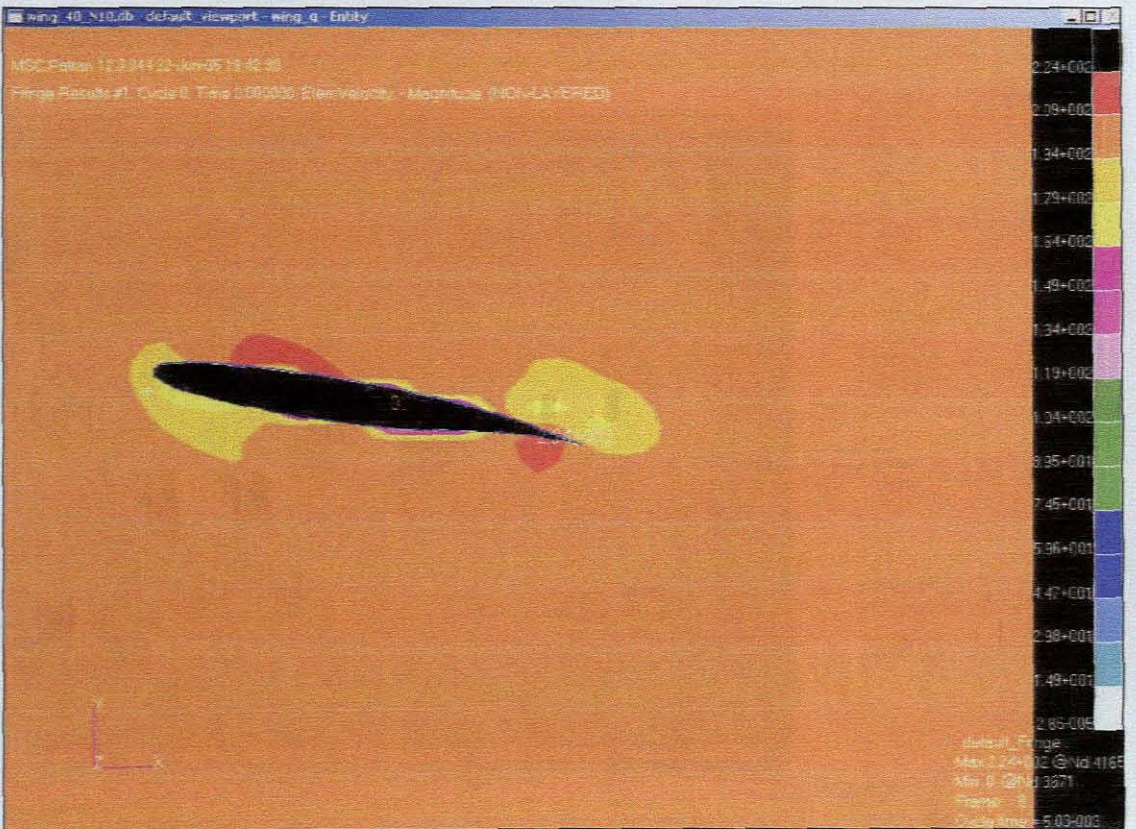
The pressure distribution for Boeing F airfoil. (AOA equal to -4)



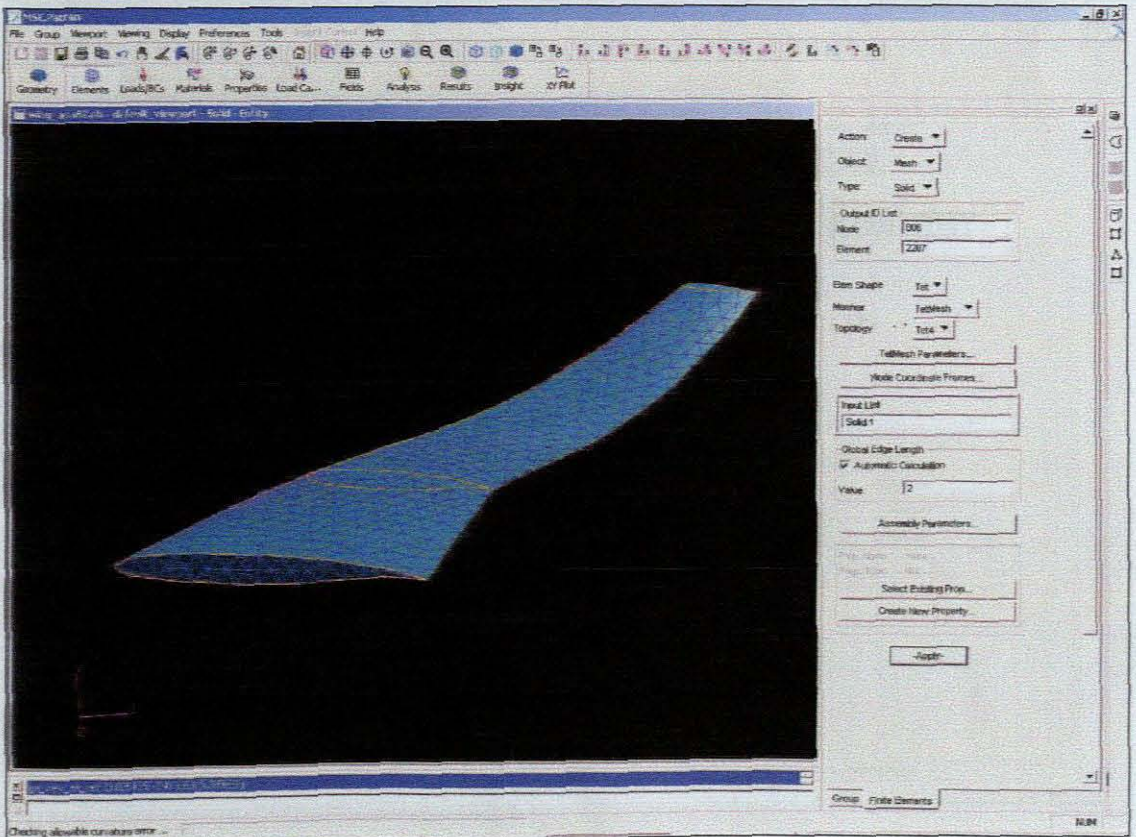
The pressure and velocity distribution for Boeing F airfoil. (AOA equal to 4)



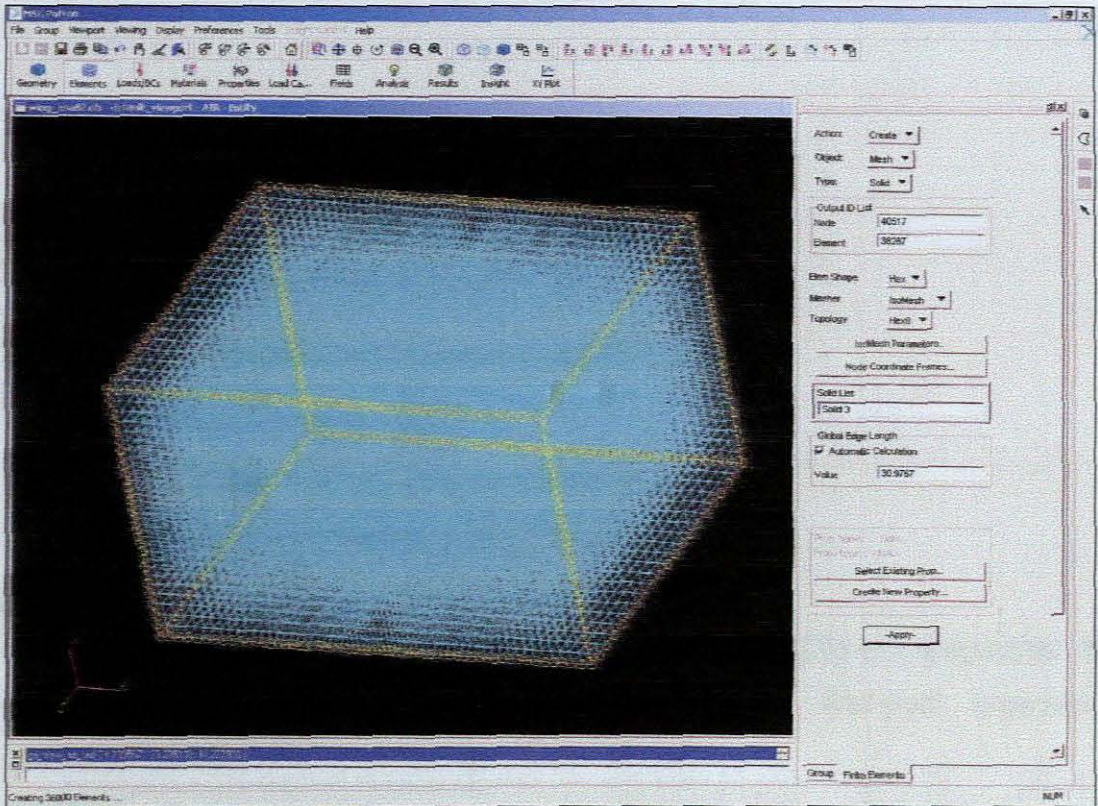
The velocity distribution for Boeing F airfoil. (AOA equal to 10)



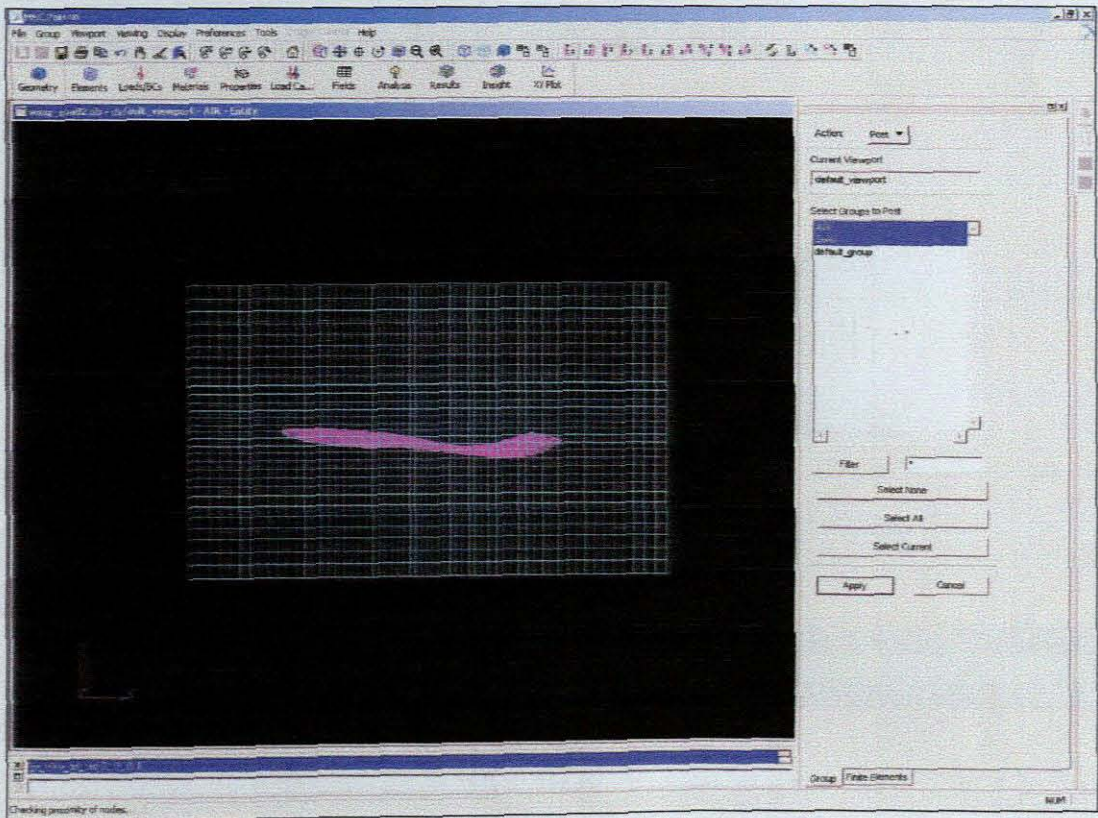
Rigid solid wing structure:



Euler mesh:



The coupling:



Reference

- [1] Marcos Kuhns, *The evolution of airplane wing*, 2000.
- [2] Luke Penrod, Diana Talley, Jeff Froyd, Rita Caso, Dimitris Lagoudas, and Terry Kohutek. *Integrating "smart" materials into a first-year engineering curriculum*, 2002.
- [3] *Biocompatibility evaluation of nickel-titanium shape memory metal alloy*.
<http://herkules.oulu.fi/sbn9514252217/html/x317.html>
- [4] *Airfoil Design*. November 24, 2003.
- [5] MIT's OpenCourseWare. <http://www.cocw.net/mit/index.htm>
- [6] William Palm , *Rapid Prototyping Primer*, May 1998.
<http://www.me.psu.edu/lamancusa/rapidpro/primer/chapter2.htm>
- [7] *Vacuum casting*.
http://modellfertigung.mw.tu-dresden.de/Ergaenzung/zpp/verfahre.nsf/77f0bdf742d17db3c125698500222b15/3bea8614419c9b6ac125698500228be8?OpenDocument&ExpandSection=1#_Section1

- [8] Justin K. Streleca, Dimitris C. Lagoudasa. *Fabrication and Testing of a Shape Memory Alloy Actuated Reconfigurable Wing*. Aerospace Engineering Department, Texas A&M University, College Station, 2001.
- [9] J T Oh, H C Park and W. Hwang, *Active shape control of a double-plate structures using piezoceramics and SMA wires*, Department of Mechanical Engineering, Pohang University of Science and Technology. 22 May 2001.
- [10] Edward V. White. *Smart Structures and Smart Structures and Systems*, Boeing - Phantom Works St. Louis, MO, 2000.
- [11] Arnold M. Kuethe Chuen-Yen Chow, *Foundations of Aerodynamics Bases of Aerodynamic Design*, John Wiley & Sons, Inc, 2001.
- [12] http://www.cs.ualberta.ca/~database/MEMS/sma_mems/flap.html
- [13] <http://www.smaterial.com/SMA/sma.html>
- [14] <http://aerodyn.org/aero.html>
- [15] Luke Penrod, Diana Talley, Jeff Froyd, Rita Caso, Dimitris Lagoudas, and Terry Kohutek, *Integrating "smart" materials into a first-year engineering curriculum, 2001*
- [16] Justin Figueroa November 24, 2003, *Airfoil Design*, 2003.
- [17] *Design and Analysis of Low Reynolds Number Airfoils*, 4 December 2002.
- [18] Emilie J. Siochi, John B. Anders, Jr., David E. Cox, Dawn C. Jegley, Robert L. Fox, and Stephen J. Katzberg, *Year 2000 Report of Findings From a Six-Month Survey*, Langley Research Center, Hampton, Virginia, 2000.

[19] <http://www.mor-tech.com/rprototypinglomm.htm>

[20] <http://www.nasq.com/afdb/insert-category-e.phtml>

[21] Edward V. White, *Smart Structures and Smart Structures and Systems*, Boeing - Phantom Works St. Louis, MO, 2000.

[22] *Finite Element Analysis*.

http://www.stress.com/oilgas/fin_elem_analysis.htm

[23] *Boeing 747 Long-Range Jetliner*.

<http://www.aerospaceweb.org/aircraft/jetliner/b747/>

[24] *Blended Winglet Technology*.

<http://www.aviationpartners.com/otherprograms/gulfstream.html>