ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

AERODYNAMIC SHAPE OPTIMIZATION OF ROTOR AIRFOILS VIA A GENETIC ALGORITHM

M. Sc. Thesis by
A. Taylan KÖKSAL, B. Sc.

Department: Space Engineering

Programme: Space Engineering

MAY 2003

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

AERODYNAMIC SHAPE OPTIMIZATION OF ROTOR AIRFOILS VIA A GENETIC ALGORITHM

M. Sc. Thesis by
A. Taylan KÖKSAL, B. Sc.
(511001204)

Date of submission: 5 May 2003

Date of defence examination: 30 May 2003

Supervisor (Chairman): Prof. R. Alsan MERİÇ

Members of the Examining Committee: Prof. A. Rüstem ARSLAN

Assist. Prof. Hayri ACAR

MAY 2003

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMA İLE ROTOR KANAT PROFİLLERİNİN AERODİNAMİK ŞEKİL OPTİMİZASYONU

YÜKSEK LİSANS TEZİ

Müh. A Taylan KÖKSAL

(511001204)

Tezin Enstitüye Verildiği Tarih: 5 Mayıs 2003
Tezin Savunulduğu Tarih: 30 Mayıs 2003

Tez Danışmanı: Prof. Dr. R Alsan Meriç
Diğer Jüri Üyeleri: Prof. Dr. A Rüstem Arslan
Yar. Doç. Dr. Hayri ACAR

MAYIS 2003

PREFACE

First of all, I would like to thank to my supervisor Prof.. R. Alsan Meriç for his support and encouragement throughout this study. I'd also like to thank to my family, friends and especially to my beloved one for their emotional support in my times of need.

May, 2003 A. Taylan KÖKSAL

CONTENTS

PREFACE iii
LIST OF TABLES vi
LIST OF FIGURES vii
LIST OF SYMBOLS viii
ÖZET ix
ABSTRACT x

1. INTRODUCTION 1
1.1. Introduction and Related Work 1

2. AIRFOIL GEOMETRY AND AERODYNAMICS 3
2.1. Airfoil Geometry 3
2.2. Airfoil Aerodynamics 4
2.2.1. Lift, Drag and Moment on Airfoils 4
2.2.2. Airfoil Pressure Distributions 6
2.3. Kutta-Joukowski Theorem 8

3. POTENTIAL FLOW 9
3.1. Potential Flow 9
3.2. Simple Solutions of Laplace Equations 11
3.2.1 Freestream Potential 12
3.2.2. Source/Sink Potential 13
3.2.3. Point Vortex Potential 14

4. SMITH&HESS PANEL METHOD 16
4.1. Panel Methods 16
4.2. Kutta Condition 17
4.3. Smith&Hess Panel Method 18
4.3.1 Introduction 18
4.3.2. Implementation 21

5. OPTIMIZATION AND GENETIC ALGORITHMS 27
5.1. Optimizaiton Overview 27
5.2. Airfoil Optimization 27
5.3. Genetic Algorithms 28
5.3.1 Overview of Genetic Algorithms 28
5.3.2. A Brief History of Genetic Algorithms 30
5.3.3 Biological Background 30

5.3.4. Structure of Genetic Algorithms 31
5.3.4.1 Encoding 32
5.3.4.2 Initial Population 33
5.3.4.3 Evaluation 34
5.3.4.4 Reproduction 34
5.3.4.5 Selection 34
5.3.4.6 Crossover and Mutation 35
5.3.4.7 Elitism 37
5.3.5. Micro Genetic Algorithms 37
5.3.6. Parameters of Genetic Algorithms 38

6. B-SPLINE CURVES **40**
6.1. Introduction 40
6.2. Formulation of a cubic B-Spline 40

7. PROBLEM FORMULATION AND RESULTS **43**
7.1. Approach 43
7.2. Airfoil Surface Representation 45
7.3. Aerodynamic Analysis Validation 46
7.4. Genetic Algorithm Organization 47
7.5. Results and Discussion 48
7.6. Conclusion 52

REFERENCES **54**

CURRICULUM VITAE **57**

LIST OF TABLES

Page No

Table 7.1 : Best airfoil properties from several generations and comparisons with NACA 0012 aerodynamic coefficients 53

Table 7.2 : Aerodynamic properties of the best design (NACA0012)..... 54

Table 7.3 : Aerodynamic properties of the best design (VR7)..... 56

LIST OF FIGURES

		Page No
Figure 1.1	: Aeroacoustic best airfoil	3
Figure 1.2	: Aerodynamic best airfoil	3
Figure 1.3	: Compromise airfoil	3
Figure 1.4	: Compromise airfoil pressure profiles	3
Figure 2.1	: The basic parts of an airfoil	4
Figure 2.2	: Lift and drag on an airfoil	6
Figure 2.3	: Pressure distributions on an airfoil	8
Figure 2.4	: Generation of lift	9
Figure 3.1	: Uniform flow	14
Figure 3.2	: Source flow	14
Figure 3.3	: Vortex flow	16
Figure 4.1	: Trailing edge velocities	18
Figure 4.2	: Airfoil analysis nomenclature for panel methods	20
Figure 4.3	: Local Panel Coordinate System	22
Figure 4.4	: Geometric Interpretation of Source and Vortex Induced Velocities	24
Figure 5.1	: The structure of a genetic algorithm	30
Figure 6.1	: B-Spline and Control Points	42
Figure 6.2	: Successive B-Splines Joined Together	43
Figure 7.1	: Limits of Control Points Representing Airfoil Surface	46
Figure 7.2	: Spline control points and representative surface of NACA 0012	47
Figure 7.3	: Comparison of CL Values	48
Figure 7.4	: Comparison of CM Values	48
Figure 7.5	: Convergence History of the Best Run	49
Figure 7.6	: An airfoil shape from generation-0 that violates the CM constraint	50
Figure 7.7	: The best airfoil in generation-10	50
Figure 7.8	: The best airfoil from generation-50	51
Figure 7.9	: The best airfoil from generation-200	51
Figure 7.10	: Final Design - The best airfoil from generation-1000	52
Figure 7.11	: Pressure distributions of the final airfoil	52
Figure 7.12	: Final Design - The best airfoil from generation-1000	54
Figure 7.13	: Pressure distributions of the final airfoil	54
Figure 7.14	: Final Design - The best airfoil from generation-1000	55
Figure 7.15	: Pressure distributions of the final airfoil	56

: Pressure distributions of the final airfoil ……..
3
3
3
3
4
6
8
9
14
l4
16
18
20
22

24
30
42
43
46

47
48
48
49

50
50
51
51
52
52
54
54
55
56

LIST OF SYMBOLS

ρ_∞	: Density of the fluid
V_∞	: Freestream velocity
c	: Chord length
q_∞	: Freestream dynamic pressure
S	: Characteristic body area
C_L	: Lift coefficient
C_D	: Drag coefficient
C_M	: Moment coefficient
p	: Local pressure
p_∞	: Freestream pressure
C_P	: Pressure coefficient
F'	: Force per unit depth
Γ	: Circulation
V	: Velocity
u	: Velocity in x direction
v	: Velocity in y direction
ϕ	: Velocity potential
n	: Unit normal
ψ	: Streamfunction
α	: Angle of attack
q	: Source strength
γ	: Vortex strength
r, θ	: Polar coordinates of x and y
r	: distance between panels
N_{pop}	: population size for genetic algorithm

GENETİK ALGORİTMA İLE ROTOR KANAT PROFİLLERİNİN AERODİNAMİK ŞEKİL OPTİMİZASYONU

ÖZET

Bu çalışmada, genetik algoritma metodu, bir helikopter kanat profilinin aerodinamik özellikleri göz önünde tutularak yapılacak bir dizayn problemini çözmek için kullanılmıştır. Bir mikro genetik algoritma, aerodinamik analiz amacıyla kullanılan Smith&Hess panel metoduyla bir arada çalıştırılmıştır. Kullanılan mikro genetik algoritma modeli "binary tournament selection", "uniform crossover" ve elitizm yöntemleriyle çalışmaktadır. Kanat profilini temsil eden B-spline eğrisinin 20 kontrol noktası genetik algoritma optimizasyonu için dizayn parametrelerini oluşturmaktadır. Bu dizayn parametreleri her üç uçuş durumu için kaldırma katsayısını maksimize etmek amacıyla optimize edilmiştir. Bu optimizasyon probleminde kısıtlar moment katsayıları için konmuştur. Genetik algoritmanın bu amaçlara erişmek amacı ile sürekli olarak sıradışı kanat profilleri bulduğu saptanmıştır. Moment katsayıları üzerindeki kısıtlar bulunan kanat profilinin NACA 0012 profilinden daha iyi aerodinamik özellikler sergilemesi göz önünde tutularak amaç fonksiyonuna yerleştirilmiştir. Gerçektende, bulunan optimum profilin NACA 0012 profilinden daha fazla moment oluşturmadığı ve verilen çeşitli hücum açıları için daha yüksek kaldırma sağladığı gözlenmiştir. Sonuç olarak, genetik algoritma yöntemi kanat profili dizaynında ümit verici sonuçlar ortaya çıkarmıştır. Fakat helikopter kanat profili dizaynı bir çok farklı mühendislik disiplinine bağımlı olduğu için, bu alanda disiplinler-arası bir şekil optimizasyonuna ihtiyaç vardır.

AERODYNAMIC SHAPE OPTIMIZATION OF ROTOR AIRFOILS VIA A GENETIC ALGORITHM

ABSTRACT

In this work, the genetic algorithm method has been used to solve a helicopter rotor airfoil design problem that addresses aerodynamic concerns. A micro genetic algorithm with binary tournament selection, uniform crossover and with an elitism operator is used in conjunction with Smith&Hess panel method aerodynamic code. The genetic algorithm operated on 20 variables which constituted the control points of a B-spline curve representing the airfoil surface. These design variables were optimized to maximize lift coefficient at three flight conditions with a constraint on moment coefficient. It was found that the genetic algorithm could consistently design a non-traditional airfoil to achieve its objectives. The constraints are embedded in the fitness function in such a way that the resulting designs exhibit characteristics more favorable than the famous NACA 0012 airfoil. Indeed, it was found that the resulting optimal design has better lift coefficients than the NACA 0012 airfoil at various values of angle of attack while not exceeding the moment coefficients of NACA 0012 airfoil in any flow condition. As a conclusion, genetic algorithm appears to be quite promising in airfoil shape design, but since the helicopter rotor airfoil design is dependent on the effects of various disciplines, an interdisciplinary design optimization should be applied.

1. INTRODUCTION

1.1 Introduction and Related Work

This work discusses the application of a genetic algorithm to an airfoil design problem in the consideration of aerodynamic concerns.

The Genetic Algorithm is a computational version of natural selection and reproduction observed in biological populations [1]. Because an analogy can be made between survival-of-the-fittest and optimization, different forms of the genetic algorithms have been applied to engineering design and optimization. Recently, these applications have also included rotorcraft design problems.

A genetic algorithm for the shape design of airfoils is preferred in this work due to several disadvantages that other design methods suffer while the genetic algorithms don't.

Direct airfoil design perturbs an initial airfoil shape to improve the performance of the airfoil. Although successful, this approach generally produces airfoils deviating only slightly from the initial design. Calculus-based search methods especially encounter this limitation they find the nearest local optimum to the original design. Airfoil features like trailing-edge tabs, droop-snoots, and complex camber would be difficult to discover using a traditional method that perturbs a known shape [2].

In contrast, inverse airfoil design produces an airfoil whose pressure distribution matches a desired distribution. The inverse approach risks defining a distribution that no physical shape can produce [2].

As one of its central features for engineering design applications, the GA begins its search from a randomly generated population of designs. This feature allows optimal designs to be conducted without the need of a starting design, which is different from most optimization methods. For rotorcraft airfoil design, this offers great promise since most previous airfoil design studies have been conducted using traditional optimization approaches [3]. Although airfoil design has been a popular

application recently, most of these efforts attempted to solve an inverse problem [4,5]. Of those that attempt the direct problem [6], many perturb an initial shape, thus limiting the chances of finding an untraditional airfoil shapes [2]. Further GA applications have started to explore multiobjective and multidisciplinary problems [2,7].

By defining a design space that uses the upper surface and lower surface coordinates as design variables, the GA may find airfoil designs that include features that can not be acquired using traditional optimization methods. In fact, the resulting airfoil may be significantly different from those found using these methods. However this approach has the drawback of requiring an aerodynamic analysis that can predict the performance of very poor shapes, as well as good shapes.

As a conclusion, a genetic algorithm methodology is developed to generate a family of two-dimensional airfoil designs that address aerodynamic concerns to solve a helicopter rotor airfoil design problem. The genetic algorithm code [8] was used in conjunction with a panel method code. The genetic algorithm operated on 20 design variables which constituted the control points for a spline representing the airfoil surface.

A very similar approach to rotorcraft airfoil design problems has been taken previously [2]. In this work a parallel genetic algorithm methodology was developed to generate a family of two-dimensional airfoil designs that address rotorcraft aerodynamic and aeroacoustic concerns [2]. The multiple objectives of this work were to minimize the drag and overall noise of the airfoil. Constraints were placed on lift coefficient, moment coefficient, and boundary layer convergence. The aerodynamic analysis code XFOIL provided pressure and shear distributions in addition to lift and drag predictions. The aeroacoustic analysis code, WOPWOP, provided thickness and loading noise predictions. The final set of designs generated by this methodology is shown in Figure 1.1, Figure1.2 and Figure 1.3. The pressure distributions of the final compromise airfoil for three flight conditions are presented in Figure 1.4.

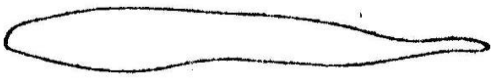

Figure 1.1 Aeroacoustic best airfoil

Figure 1.2 Aerodynamic best airfoil

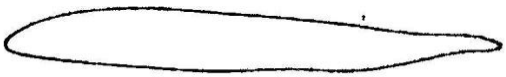

Figure 1.3 Compromise airfoil

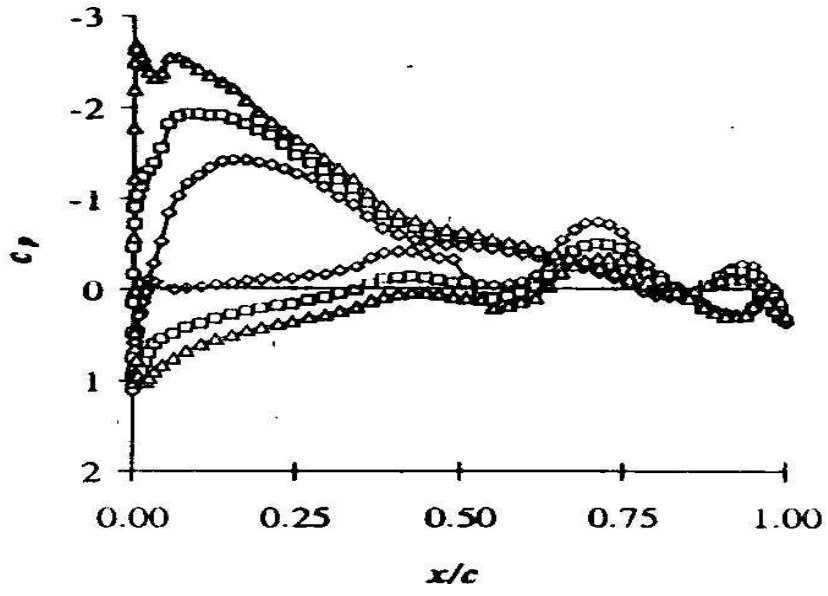

Figure 1.4 Compromise airfoil pressure profiles: ◊, flow condition 1; □, flow condition 2; ○, flow condition 3.

2. AIRFOIL GEOMETRY AND AERODYNAMICS

2.1 Airfoil Geometry

Airfoil geometry can be characterized by the coordinates of the upper and lower surface. It is often summarized by a few parameters such as: maximum thickness, maximum camber, position of max thickness, position of max camber, and nose radius. One can generate a reasonable airfoil section given these parameters.

The basic parts of an airfoil are shown in Figure 2.1, with their definitions below

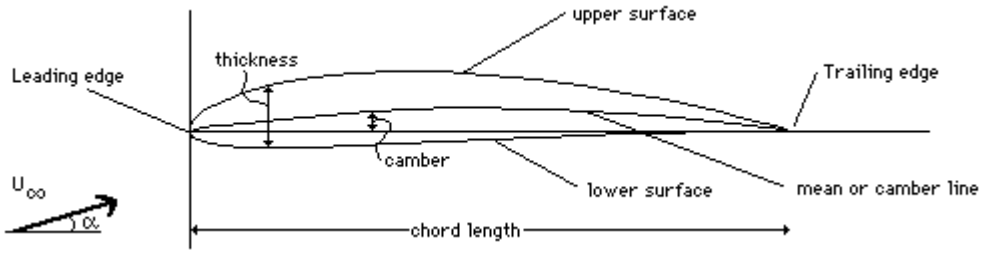

Figure 2.1 The basic parts of an airfoil

- The *chord line* is a straight line connecting the leading and trailing edges of the airfoil.
- The *chord* is the length of the chord line from leading edge to trailing edge and is the characteristic longitudinal dimension of the airfoil.
- The *mean camber line* is a line drawn halfway between the upper and lower surfaces. The chord line connects the ends of the mean camber line.
- The shape of the mean camber is important in determining the aerodynamic characteristics of an airfoil section. *Maximum camber* (displacement of the

mean camber line from the chord line) and the location of maximum camber help to define the shape of the mean camber line. These quantities are expressed as fractions or percentages of the basic chord dimension.

- Thickness and thickness distribution of the profile are important properties of an airfoil section. The *maximum thickness* and its *location* help define the airfoil shape and are expressed as a percentage of the chord.
- The *leading edge radius* of the airfoil is the radius of curvature given the leading edge shape.

2.2 Airfoil Aerodynamics

2.2.1 Lift, Drag and Moment On Airfoils

The aerodynamic forces and moments acting on an airfoil are due to two basic sources:

- Pressure distribution over the body surface
- Shear stress distribution over the body surface

The net effect of the pressure and shear stress distributions integrated over the complete body surface is a resultant aerodynamic force R and moment M on the body.[9]

An airfoil's aerodynamic force may be separated into lift and drag components. This force intersects with its chord line at a point designated as its center of pressure. The lift, drag, and center of pressure for a cambered airfoil vary as its angle of attack is changed. No aerodynamic moments (the tendency of an airfoil to turn about its center of gravity) are present at the center of pressure because the line of action of the aerodynamic force passes through this point. If one has the airfoil mounted at some fixed point along the chord, for example, a quarter of a chord length behind the leading edge, the moment is not zero unless the resultant aerodynamic force is zero or the point corresponds to the center of pressure. The moment about the quarter-chord point is generally a function of the angle of attack.

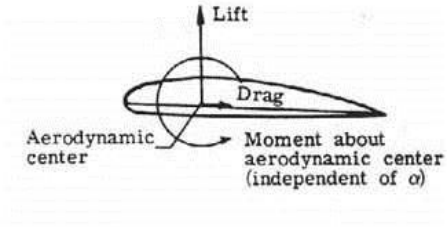

Figure 2.2 Lift and drag on an airfoil

In the flow of real fluid about a body, the aerodynamic resistance depends on the size, shape, and attitude of the body (its direction with respect to the airflow—angle of attack); the properties of the fluid, e.g., its density and pressure; and the relative velocity between the body and the fluid (air). To illustrate, consider the lift force, defined as the aerodynamic reaction perpendicular to the direction of the airflow. Lift depends on size, shape, attitude, fluid properties, and velocity. For an ideal fluid, the fluid properties (except for density) do not influence the lift force. For a real fluid, however, viscosity, elasticity (the reciprocal of compressibility), and turbulent properties are also important. In addition to the shape and attitude of the body, the surface roughness has an effect on the force. Furthermore, the effects of attitude and shape of a body are lumped together into the factor called K. Then,

$$Lift = \rho_\infty \times V_\infty^{\,2} \times S \times K \tag{2.1}$$

Since we are interested in two-dimensional shapes, we should replace the area S, with the chord length, c.

$$S = c \tag{2.2}$$

The dynamic pressure of airflow was previously defined as $q_\infty = \frac{1}{2}\rho_\infty V_\infty^{\,2}$, so if a value of 1/2 is included in equation (2.1) and the value of K is doubled to keep the equation the same, 2K may be replaced by C_L. Finally;

$$Lift = C_L q_\infty c \tag{2.3}$$

C_L is known as the coefficient of lift. The equation states simply that aerodynamic lift is determined by a coefficient of lift times the free-stream dynamic pressure, q_∞, times the characteristic body area S.

$$C_L = \frac{Lift}{q_\infty c} \tag{2.4}$$

The aerodynamic drag is the aerodynamic resistance parallel to the free-stream direction (the direction of the airflow). Similar equations can be obtained for the drag coefficient, namely,

$$C_D = \frac{Drag}{q_\infty c} \tag{2.5}$$

The *moment* acting on a body is a measure of the body's tendency to turn about its center of gravity. This moment represents the resultant aerodynamic force times a moment distance. A similar derivation may be applied to the moment equation as used for the lift and drag equations such that,

$$C_M = \frac{Moment}{q_\infty c^2} \tag{2.6}$$

where C_M is the coefficient of moment and "Moment" is the measured moment per unit length acting on the airfoil (whether at the quarter-chord point or at the aerodynamic center or any other point desired).

2.2.2 Airfoil Pressure Distributions

The aerodynamic performance of airfoil sections can be studied by reference to the distribution of pressure over the airfoil. This distribution is usually expressed in terms of the pressure coefficient:

$$C_P = \frac{p - p_\infty}{q_\infty} \tag{2.7}$$

C_P is the difference between local static pressure and freestream static pressure, nondimensionalized by the freestream dynamic pressure. To see airfoil pressure distribution C_P is plotted versus x/c which varies from 0 at the leading edge to 1.0 at the trailing edge. C_P is plotted "upside-down" with negative values higher on the plot so that the upper surface of a conventional lifting airfoil corresponds to the upper curve. Various parts of the pressure distribution are depicted in Figure 2.3.

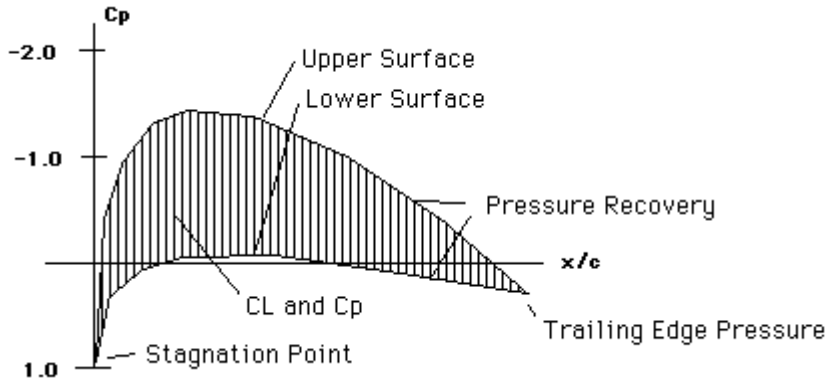

Figure 2.3 Pressure distributions on an airfoil

When the chord is taken as 1 unit, the section lift coefficient is related to the C_p by:

$$C_L = \int_0^1 C_{pl} - C_{pu} dx \tag{2.8}$$

In other words it is the area between the curves of upper and lower surface pressure coefficients.

2.3 Kutta-Joukowski Theorem

An isolated two-dimensional airfoil in an incompressible inviscid flow feels a force per unit depth of

$$F' = \rho_\infty V_\infty \times \Gamma \tag{2.9}$$

Since the lift and drag forces are defined as the forces in the directions normal and parallel to the free stream, a direct consequence is that

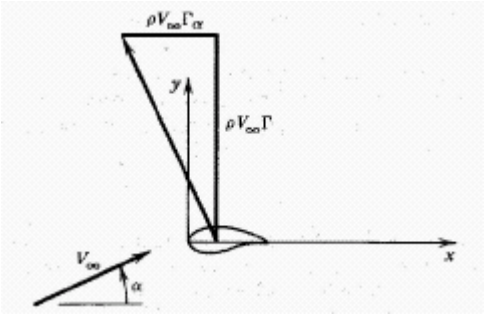

Figure 2.4 Generation of lift

$$L' = \rho V_\infty \Gamma \tag{2.10}$$
$$D' = 0$$

The Kutta-Joukowski theorem is simply an alternate way of expressing the consequences of the surface pressure distribution since lift is caused by the net imbalance of the surface pressure distribution, and circulation is simply a defined quantity determined from the same pressures.

3. POTENTIAL FLOW

3.1 Potential Flow

For incompressible flows, the continuity equation, that is the conservation of mass is given by;

$$\nabla \bullet V = 0 \tag{3.1}$$

An inviscid, incompressible fluid is also called an ideal fluid, or perfect fluid. In a velocity field, the curl of the velocity defines the vorticity. Since an ideal incompressible fluid with zero vorticity at an instant will never generate vorticity, it is mathematically consistent to consider a fluid with

$$\nabla \times V = 0 \tag{3.2}$$

everywhere and at all times. Such fluid flows are called irrotational flows. This implies that the fluid elements have no angular velocity, their motion through space is a pure translation. [9] The irrotationality condition for a two dimensional flow is ;

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0 \tag{3.3}$$

These flows are often called potential flows also, because the velocity for an irrotational flow can be written as the gradient of a scalar potential;

$$\nabla \times (\nabla \phi) = 0 \tag{3.4}$$

This equation states that for an irrotational flow, there exists a scalar function ϕ such that the velocity given is given by the gradient of ϕ which is denoted as the velocity potential.

For two dimensional flows, the velocity components , u and v can be expressed as,

$$u = \frac{\partial \phi}{\partial x} \quad , \quad v = \frac{\partial \phi}{\partial y} \tag{3.5}$$

Substituting the velocity components, u and v, for two dimensional flows, the equations for continuity and irrotationality would then become,

$$\nabla^2 \phi = 0 \tag{3.6}$$

In the above equation, the operator,

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{3.7}$$

is termed the Laplacian operator, after Pierre de Laplace (1749-1827).

Since the normal component of the fluid velocity has to be zero at a solid surface at rest, the appropriate boundary condition is

$$n \bullet \nabla \phi = 0 \tag{3.8}$$

on a static solid boundary with unit normal n.

Consequently, the Laplace equation is the governing equation for the solution of the problems of this inviscid, incompressible fluid. The assumption made in solving Laplace's equation is that the flows satisfy the equations for continuity and irrotationality.

3.2 Simple Solutions of Laplace Equations

Potential Flow is an idealized method of modeling flow. If a flow is inviscid and incompressible then a velocity potential function ϕ can be defined such that

$$u = \frac{\partial \phi}{\partial x} \quad , \quad v = \frac{\partial \phi}{\partial y} \tag{3.9}$$

Substituting these expressions for horizontal (u), and vertical (v) velocity into the governing continuity equation for a flow produces a Laplace equation the solution of which is relatively simple.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial x} = 0 \qquad \text{becomes} \qquad \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0 \tag{3.10}$$

In two-dimensional flow, stream function (ψ) can be defined as a measure of the volume flow rate of fluid between a pair of streamlines. Streamlines are defined by joining a continuous line of points in the flow field by following the local velocity vectors.

Streamlines have a constant value of stream function since all the flow must be parallel to the streamlines. No flow crosses a streamline. For two-dimensional, inviscid, incompressible flow, continuity makes the local product of distance between streamlines and velocity a constant. Thus the velocities in a flow field can also be found by differentiating stream function with respect to the flow field coordinates y and x.

$$\frac{\partial \psi}{\partial y} = u \qquad \text{and} \qquad \frac{\partial \psi}{\partial x} = -v \tag{3.11}$$

If the two dimensional flow is irrotational ($\Gamma = 0$) then stream function can also be used to define a governing Laplace equation.

$$\gamma = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0 \qquad \text{becomes} \quad \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{3.12}$$

Velocity potential (ϕ) and stream function (ψ) are orthogonal functions defining ideal, inviscid incompressible and irrotational flow in two dimensions.

Simple stream functions or velocity potential functions can be found which are exact solutions for the above Laplace equations.

One way to obtain solutions to Laplace's equation (subject to the appropriate boundary conditions) is to exploit its linear nature and the principle of superposition.

Superposition

If $\phi_1, \phi_2, \phi_3, \phi_4$ is each a solution of, $\nabla^2 \phi = 0$

then $A\phi_1 + B\phi_2 + C\phi_3 + \mathrm{K} + X\phi_n$ is also a solution of $\nabla^2 \phi = 0$, where A, B... X are constants.

Thus, the solution for a complex problem can be expressed as the sum of solutions of several simpler problems.

Obviously, the principle of superposition can be used to add up an arbitrary number of *elementary solutions* to Laplace's equation. This method is rather powerful and will be used in the remainder of this lecture to construct the solution of the flow about an arbitrarily shaped lifting airfoil.

There are three types of elementary solutions. With a *large* number of each of these three kinds of solutions we will be able to construct the flow about rather general airfoils at arbitrary angles of attack.

3.2.1 Free Stream Potential

The potential function and stream function for a *free stream* of magnitude V_∞, aligned with the *x-a*xis is given by

$$\phi_\infty = V_\infty x \qquad \psi_\infty = V_\infty y \tag{3.13}$$

Taking the gradient of this potential we see that the resulting velocity field is given by

$$u(x,y) = V_\infty \tag{3.14a}$$

$$v(x,y) = 0 \tag{3.14b}$$

That is, the velocity is uniform everywhere in the domain. The potential function can be *rotated* at an arbitrary angle α so that

$$\phi_\infty = V_\infty(\cos\alpha x + \sin\alpha y). \tag{3.15}$$

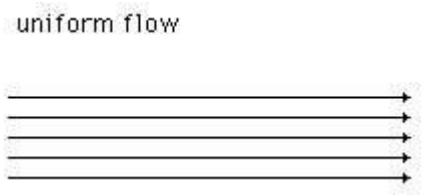

Figure 3.1 Uniform flow

3.2.2 Source/Sink Potential

A source/sink that expels/absorbs an amount of fluid volume/unit time $= \pm q$ can be constructed from the following potential;

$$\phi_S = \frac{\pm q}{2\pi}\ln(r) \tag{3.16}$$

and the stream function;

$$\psi_S = \frac{\pm q}{2\pi}\theta \tag{3.17}$$

where $x = r.\cos(\theta)$ $y = r.\sin(\theta)$ and q is source strength.

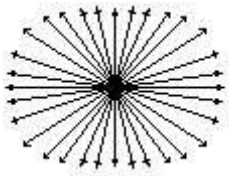

Figure 3.2 Source flow

The resulting velocity components are;

$$u(x,y) = \frac{\pm q}{2\pi}\frac{x}{x^2+y^2} \tag{3.18a}$$

$$v(x,y) = \frac{\pm q}{2\pi}\frac{y}{x^2+y^2} \tag{3.18b}$$

3.2.3. Point Vortex Potential

$$\phi_V = -\frac{\gamma}{2\pi}\theta \qquad \psi_V = \frac{\gamma}{2\pi}\ln(r) \tag{3.19}$$

where γ is the circulation strength of the vortex. γ is defined positive if the induced circular flow is clockwise. θ is the angle measured in polar coordinates from some arbitrary origin radial line. Taking the gradient of this function, we see that the velocity field of a vortex is given by

$$u(x,y) = \frac{\pm \gamma}{2\pi}\frac{y}{x^2+y^2} \tag{3.20a}$$

$$v(x,y) = \frac{\pm \gamma}{2\pi}\frac{x}{x^2+y^2} \tag{3.20a}$$

and has streamlines that are concentric circles centered about the location of the point vortex. The circulation around *any* contour that encloses the point vortex is constant and equal to γ. Furthermore, the flow *outside* of the point vortex is fully irrotational. All of the vorticity in this flow is contained at the singular location of the point vortex. Notice also that the form of the potential for a point vortex is rather similar to that of a source/sink, with the substitutions $x \rightarrow y$, and $y \rightarrow -x$ in the appropriate formula for the velocity field: the velocity fields are perpendicular to each other (and so are the equipotential lines).

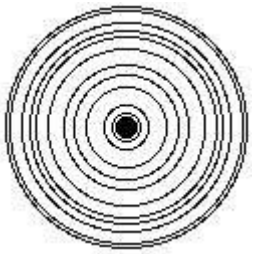

Figure 3.3 Vortex Flow

A wide range of flows and flow around objects can be built up using these simple solutions. The more complex potential flows can be constructed by the superposition of exact solutions or numerical approximation techniques can be used with this governing equation to solve a large range of aerodynamic problems.

4. SMITH&HESS PANEL METHOD

4.1 Panel Methods

Panel methods are widely used in the aerospace and automotive industry and are effectively boundary-element methods for computational fluid dynamics problems. These methods employ the surface of the body over which fluid is flowing to be used as the computational domain rather than using the whole region in which the body is embedded. This is not only computationally more efficient than a finite difference method, for example, but also allows more complicated body shapes to be studied than would be tractable if the body were embedded in a regular mesh.

Panel methods have been extensively investigated and successfully applied in the aeronautic industry for more than 30 years, and at present there are a number of production codes available that perform computations of flows around very complex geometries. The method is based on the formulation of linear potential flow around a solid body in terms of an integral equation over the surface for the velocity or potential. The surface integral is approximated by panel elements on which a singularity distribution is assumed to exist. The numerical solution of the potential flow is rendered into a matrix linear system, where the singularity distributions over the panels are the unknowns. The attractiveness of this formulation lies in the fact that the solution can be obtained from the surface problem, and it is not necessary to model the complete three-dimensional flow field. Thus, panel methods reduce the dimension of the problem, from three to two dimensions, or from two dimensions to a one-dimensional problem. Other advantages are the capability of obtaining flow solutions about completely arbitrary configurations and the fact that the predictions by panel methods have proven to agree well with experiment in a wide range of flow conditions [10].

General aspects relating to the implementation of panel methods were discussed, categorizing them in four distinct procedures, as follows:

1. Geometry definition and generation of the surface panel elements.
2. Calculation of the coefficient matrix (velocity induced by the panels on each other's control points).
3. Solution of the resulting system of linear equations for the singularity strengths.
4. Calculation of the flow field parameters of interest.

One of the important advantages of panel methods is their ability to analyze the flow around complex configurations or "arbitrary bodies", as is commonly expressed. Other computational methods for fluid dynamics, *e.g.*, finite elements and finite volume, require a grid on the fluid domain which is not easily adapted to complex body geometries. Consequently, panel methods have found massive applications in industry, and practically all major aeronautic companies have a proprietary panel code or have customized a public-domain panel method. [11]

The main interest is the possibility of simulating flows around complete aircraft, or significant parts of an aircraft.

4.2 Kutta Condition

Kutta condition states that the pressure above and below the airfoil trailing edge must be equal, and that the flow must smoothly leave the trailing edge in the same direction at the upper and lower edge. Consider the figure below:

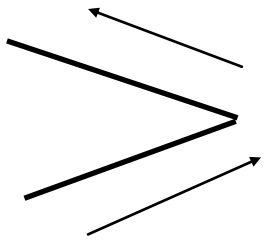

Figure 4.1 Trailing Edge Velocities

From this sketch we see that pressure will be equal, and the flow will leave the trailing edge smoothly, if, and only if

$$\gamma_{upper} = V_{upper} \qquad \gamma_{lower} = V_{lower} \tag{4.1}$$

4.3 Smith & Hess Panel Method

4.3.1 Introduction

There are many choices as to how to formulate a panel method (singularity solutions, variation within a panel, singularity strength and distribution, etc.) The simplest and first truly practical method was due to Hess and Smith, Douglas Aircraft, 1966. The Hess-Smith technique combines source panels and vortices for a single-element, lifting airfoil in incompressible flow. It is based on a distribution of sources and vortices on the surface of the geometry. In their method

$$\phi = \phi_\infty + \phi_S + \phi_V \tag{4.2}$$

where, ϕ is the total potential function and its three components are the potentials corresponding to the free stream, the source distribution, and the vortex distribution. These last two distributions have potentially locally varying strengths $q(s)$ and $\gamma(s)$, where s is an arc-length coordinate which spans the complete surface of the airfoil in any way desired.

The potentials created by the distribution of sources/sinks and vortices are given by:

$$\phi(s) = \int \frac{q(s)}{2\pi} \ln r ds \tag{4.3}$$

$$\phi(v) = -\int \frac{\gamma(s)}{2\pi} \theta ds \tag{4.4}$$

where the various quantities are defined in the Figure below, where s is the distance measured along the surface and (r,θ) are polar coordinates of the 'field point' (x,y), relative to the point on the surface whose location is indicated by s.

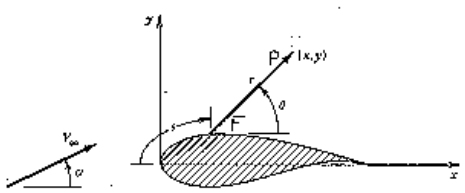

Figure 4.2: Airfoil Analysis Nomenclature for Panel Methods

In these formula, the integration is to be carried out along the complete surface of the airfoil. Using the superposition principle, any such distribution of sources/sinks and vortices satisfies Laplace's equation, but we will need to find conditions for $q(s)$ and $\gamma(s)$ such that the flow tangency boundary condition and the Kutta condition are satisfied.

In theory, we could:

- Use the source strength distribution to satisfy flow tangency and the vortex distribution to satisfy the Kutta condition.
- Use arbitrary combinations of both sources/sinks and vortices to satisfy both boundary conditions simultaneously.

Hess and Smith made the following valid simplification; take the vortex strength to be constant over the whole airfoil and use the Kutta condition to fix its value, while allowing the source strength to vary from panel to panel so that, together with the constant vortex distribution, the flow tangency boundary condition is satisfied everywhere.

In order to solve the equation, consisting challenging integrals to evaluate, simplification is done by selecting a number of points, N, on the body contour, called nodes. The nodes are then connected with straight lines, which become the panels, as shown in figure 4.2. Figure 4.2 illustrates the representation of a smooth surface by a series of line segments. The numbering system starts at the lower surface trailing edge and proceeds forward, around the leading edge and rearward to the upper surface trailing edge. N+1 points are used to define N panels.

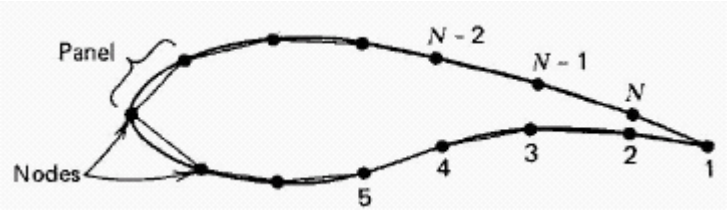

Figure 4.2 Definition of Nodes and Panels

we can discretize Equation 4.2 in the following way:

$$\phi = V_\infty(\cos\alpha x + \sin\alpha y) + \left[\sum_{j=1}^{N}\int_{panel\,j}\frac{q(s)}{2\pi}\ln r - \frac{\gamma}{2\pi}\theta\right]ds \qquad (4.5)$$

Since Equation 4.5 involves integrations over each discrete panel on the surface of the airfoil, the variation of source and vortex strength within each of the panels must be parameterized. Since the vortex strength was considered to be a constant, the attention should be on the source strength distribution within each panel. This is the major approximation of the panel method. However, it can be seen how the importance of this approximation should decrease as the number of panels, $N \to \infty$ (of course this will increase the cost of the computation considerably, so there are more efficient alternatives.)

Hess and Smith decided to take the simplest possible approximation, that is, to take the source strength to be constant on each of the panels

$$q(s) = q(i) \quad on\ panel\ i, \qquad i = 1,.....,N \qquad (4.6)$$

Therefore, we have $N+1$ unknowns to solve for in our problem: the N panel source strengths q_i and the constant vortex strength γ. Consequently, we will need $N + 1$ independent equations which can be obtained by formulating the flow tangency boundary condition at each of the N panels, and by enforcing the Kutta condition. The solution of the problem will require the inversion of a matrix of size $(N+1)\times(N+1)$.

In the Smith&Hess method, the flow tangency boundary condition is imposed on the points located at the midpoint of each of the panels. Although this approach suffers from a slight alteration of the surface geometry, it is easy to implement and yields fairly accurate results for a reasonable number of panels. This location is also

used for the imposition of the Kutta condition (on the last panels on upper and lower surfaces of the airfoil, assuming that their midpoints remain at equal distances from the trailing edge as the number of panels is increased).

4.3.2 Implementation

Consider the ith panel to be located between the ith nodes, with its orientation to the x-axis given by

$$\sin\theta_i = \frac{y_{i+1} - y_i}{l_i} \tag{4.7}$$

$$\cos\theta_i = \frac{x_{i+1} - x_i}{l_i} \tag{4.8}$$

where l_i is the length of the panel under consideration. The normal and tangential vectors to this panel, are then given by

$$\hat{n}_i = -\sin\theta_i \hat{i} + \cos\theta_i \hat{j} \tag{4.9}$$

$$\hat{t}_i = -\cos\theta_i \hat{i} + \sin\theta_i \hat{j} \tag{4.10}$$

The tangential vector is oriented in the direction from node i to node i+1, while the normal vector, if the airfoil is traversed clockwise, points into the fluid.

Letting the ith panel be defined as the one between the ith and (i +1)th nodes, and its inclination to the x-axis be , as shown in Figure 4.3.

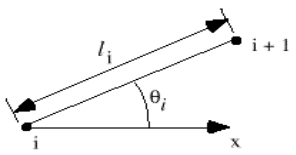

Figure 4.3. Local Panel Coordinate System

Furthermore, the coordinates of the midpoint of the panel are given by

$$\bar{x}_i = \frac{x_{i+1} + x_i}{2} \tag{4.11}$$

$$\bar{y}_i = \frac{y_{i+1} + y_i}{2} \tag{4.12}$$

and the velocity components at these midpoints are given by

$$u_i = u(\bar{x}_i, \bar{y}_i) \tag{4.13}$$

$$v_i = v(\bar{x}_i, \bar{y}_i) \tag{4.14}$$

The flow tangency boundary condition can then be simply written as $(\vec{u}.\vec{n}) = 0$, or, for each panel

$$-u_i \sin\theta_i + v_i \cos\theta_i = 0 \quad for\ i = 1, \ldots, N \tag{4.15}$$

while the Kutta condition is simply given by

$$u_1 \sin\theta_1 + v_1 \cos\theta_1 = -u_N \sin\theta_N - v_N \cos\theta_N \tag{4.16}$$

where the negative signs are due to the fact that the tangential vectors at the first and last panels have nearly opposite directions.

Now, the velocity at the midpoint of each panel can be computed by superposition of the contributions of all sources and vortices located at the midpoint of *every* panel (including itself). Since the velocity induced by the source or vortex on a panel is proportional to the source or vortex strength in that panel, q_i and γ can be pulled out of the integral in Equation 4.5 to yield

$$u_i = V_\infty \cos\alpha + \sum_{j=1}^{N} q_j u_{sij} + \gamma \sum_{j=1}^{N} u_{vij} \tag{4.17}$$

$$v_i = V_\infty \sin\alpha + \sum_{j=1}^{N} q_j u_{sij} + \gamma \sum_{j=1}^{N} u_{vij} \tag{4.18}$$

where u_{sij}; v_{sij} are the velocity components at the midpoint of panel i induced by a source of unit strength at the midpoint of panel j. A similar interpretation can be found for u_{vij}, v_{vij}. In a coordinate system tangential and normal to the panel, we can

perform the integrals in Equation 4.5 by noticing that the local velocity components can be expanded into absolute ones according to the following transformation:

$$u = u^* \cos\theta_j - v^* \sin\theta_j \tag{4.19}$$

$$v = u^* \sin\theta_j + v^* \cos\theta_j \tag{4.20}$$

Now, the local velocity components at the midpoint of the *i*th panel due to a unit-strength source distribution on this *j*th panel can be written as

$$u^*_{sij} = \frac{1}{2\pi}\int_0^{l_j} \frac{x^* - t}{(x^* - t)^2 + y^{*2}} dt \tag{4.21}$$

$$v^*_{sij} = \frac{1}{2\pi}\int_0^{l_j} \frac{y^*}{(x^* - t)^2 + y^{*2}} dt \tag{4.22}$$

where (x^*, y^*) are the coordinates of the midpoint of panel *i* in the local coordinate system of panel *j*. Carrying out the integrals in Equation 4.21 we find that

$$u^*_{sij} = -\frac{1}{2\pi}\ln\left[\left(x^* - t\right) + y^{*2}\right]^{\frac{1}{2}} \Big|_{t=0}^{t=l_j} \tag{4.23}$$

$$v^*_{sij} = -\frac{1}{2\pi}\tan^{-1}\frac{y^*}{x^* - t} \Big|_{t=0}^{t=l_j} \tag{4.24}$$

These results have a simple geometric interpretation that can be discerned by looking at the Figure 4.4. One can say that

$$u^*_{sij} = -\frac{1}{2\pi}\ln\frac{r_{ij+1}}{r_{ij}} \Big|_{t=0}^{t=l_j} \tag{4.25}$$

$$v^*_{sij} = \frac{v_1 - v_0}{2\pi} = \frac{\beta_{ij}}{2\pi} \tag{4.26}$$

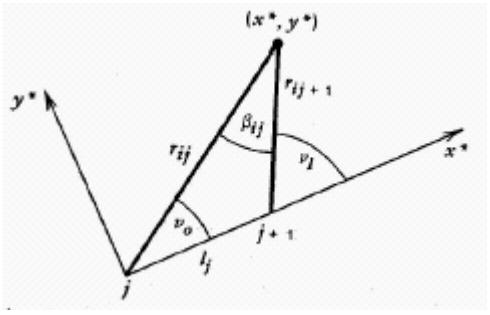

Figure 4.4 Geometric Interpretation of Source and Vortex Induced Velocities

r_{ij} is the distance from the midpoint of panel i to the jth node, while β_{ij} is the angle subtended by the jth panel at the midpoint of panel i. Notice that $u^*_{sii} = 0$, but the value of $v^*_{sii} = 0$ is not so clear. When the point of interest approaches the midpoint of the panel from the *outside* of the airfoil, this angle, $\beta_{ii} \to \pi$. However, when the midpoint of the panel is approached from the *inside* of the airfoil, $\beta_{ii} \to -\pi$. Since we are interested in the flow *outside* of the airfoil only, we will always take $\beta_{ii} \to \pi$.

Similarly, for the velocity field induced by the vortex on panel j at the midpoint of panel i we can simply see that

$$u^*_{sij} = \frac{1}{2\pi}\int_0^{l_j} \frac{x^* - t}{(x^* - t)^2 + y^{*2}} dt = \frac{\beta_{ij}}{2\pi} \tag{4.27}$$

$$v^*_{sij} = \frac{1}{2\pi}\int_0^{l_j} \frac{y^*}{(x^* - t)^2 + y^{*2}} dt = \frac{1}{2\pi}\ln\frac{r_{ij}+1}{r_{ij}} \tag{4.28}$$

and finally, the flow tangency boundary condition, using Equation 4.17 and 4.18, and undoing the local coordinate transformation of Equation 4.19 and 4.20 can be written as

$$\sum_{j=1}^{N} A_{ij} q_j + A_{iN+1}\gamma = b_i \tag{4.29}$$

where

$$A_{ij} = -u_{sij}\sin\theta_i + v_{sij}\cos\theta_i \tag{4.30}$$

which yields

$$2\pi A_{ij} = \sin(\theta_i - \theta_j)\ln\frac{r_{ij}+1}{r_{ij}} + \cos(\theta_i - \theta_j)\beta_{ij} \tag{4.32}$$

Similarly for the vortex strength coefficient

$$2\pi A_{iN+1} = \sum_{j=1}^{N}\cos(\theta_i - \theta_j)\ln\frac{r_{ij}+1}{r_{ij}} - \sin(\theta_i - \theta_j)\beta_{ij} \tag{4.33}$$

The right hand side of this matrix equation is given by

$$b_i = V_\infty \sin(\theta_i - \alpha) \tag{4.34}$$

The flow tangency boundary condition gives us N equations. We need an additional one provided by the Kutta condition in order to obtain a system that can be solved. According to Equation 4

$$\sum_{j=2}^{N} A_{N+1,j} q_j + A_{N+1,N+1} \gamma = b_{N+1} \tag{4.35}$$

After similar manipulations we find that

$$b_{N+1} = -V_\infty \cos(\theta_1 - \alpha) - V_\infty \cos(\theta_N - \alpha) \tag{4.36}$$

These various expressions set up a matrix problem of the kind;

$$Ax = b$$

where the matrix A is of size $(N+1)\times(N+1)$. This system can be sketched as follows:

$$\begin{bmatrix} A_{11} & \dots & A_{1i} & \dots & A_{1N} & A_{1,N+1} \\ \vdots & & \vdots & & \vdots & \vdots \\ A_{i1} & \dots & A_{ii} & \dots & A_{iN} & A_{i,N+1} \\ \vdots & & \vdots & & \vdots & \vdots \\ A_{N1} & \dots & A_{Ni} & \dots & A_{NN} & A_{N,N+1} \\ A_{N+1,1} & \dots & A_{N+1,i} & \dots & A_{N+1,N} & A_{N+1,N+1} \end{bmatrix} \begin{bmatrix} q_1 \\ \vdots \\ q_i \\ \vdots \\ q_N \\ \gamma \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_i \\ \vdots \\ b_N \\ b_{N+1} \end{bmatrix}$$

Notice that the cost of inversion of a full matrix such as this one is $O(N+1)^3$, so that, as the number of panels increases without bounds, the cost of solving the panel problem increases rapidly. This is usually not a problem for *two-dimensional* flows, but becomes a serious problem in *three-dimensional* flows where the number of panels are much higher.

Finally, once we have solved the system for the unknowns of the problem, it is easy to construct the tangential velocity at the midpoint of each panel according to the following formula

$$V_{ti} = V_\infty \cos(\theta_i - \alpha) + \sum_{j=1}^{N} \frac{q_{ij}}{2\pi} \left[\sin(\theta_i - \theta_j)\beta_{ij} - \cos(\theta_i - \theta_j) \ln \frac{r_{ij+1}}{r_{ij}} \right]$$

$$+ \frac{\gamma}{2\pi} \sum_{j=1}^{N} \left[\sin(\theta_i - \theta_j) \ln \frac{r_{ij+1}}{r_{ij}} + \cos(\theta_i - \theta_j)\beta_{ij} \right] \qquad (4.37)$$

And knowing the tangential velocity component, we can compute the pressure coefficient (no approximation since $V_{ni} = 0$) at the midpoint of each panel according to the following formula

$$C_p(\bar{x}_i, \bar{y}_i) = \frac{V_{ti}^{\,2}}{V_\infty^{\,2}} \qquad (4.38)$$

from which the force and moment coefficients can be computed assuming that this value of *Cp* is constant over each panel and by performing the discrete sum

5. OPTIMIZATION AND GENETIC ALGORITHMS

5.1 Optimization Overview

Optimization techniques are used to find a set of design parameters, $x = \{x_1, x_2, ..., x_n\}$, that can in some way be defined as optimal. In a simple case this might be the minimization or maximization of some system characteristic that is dependent on x. In a more advanced formulation the objective function, f(x), to be minimized or maximized, might be subject to constraints in the form of equality constraints,

$G_i(x) = 0 \quad i = 1, ..., m_e$; inequality constraints

$G_i(x) \le 0 \quad i = m_e + 1, ..., m$ and/or parameter bounds, x_l, x_u

A general problem description is stated as

Minimize $f(x)$

$x \in \Re^n$

subject to

$G_i(x) = 0 \quad i = 1, ..., m_e$

$G_i(x) \le 0 \quad i = m_e + 1, ..., m$

$x_l \le x \le x_u$

where x is the vector of design parameters ($x \in \Re^n$), $f(x)$ is the objective function that returns a scalar value, and the vector function $G(x)$ returns the values of the equality and inequality constraints evaluated at x.

An efficient and accurate solution to this problem depends not only on the size of the problem in terms of the number of constraints and design variables but also on characteristics of the objective function and constraints.

5.2 Airfoil Optimization

In the design of airfoils typical targets include prescribed pressure or velocity distributions, lift range, maximum lift, minimal drag, shock-free suction side in transonic flow and type of stall at subsonic speeds, under geometrical constraints that may include one or more of the following: thickness ratio, maximum camber, leading edge radius, trailing edge angle, or even the whole geometry itself defined by coordinates of control points.

5.3 Genetic Algorithms

5.3.1 Overview of Genetic Algorithms

Evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation [12].

Evolutionary algorithms model natural processes, such as selection, recombination, mutation, migration, locality and neighborhood. Evolutionary algorithms work on populations of individuals instead of single solutions. In this way the search is performed in a parallel manner. Figure 1 shows the structure of a simple genetic algorithm:

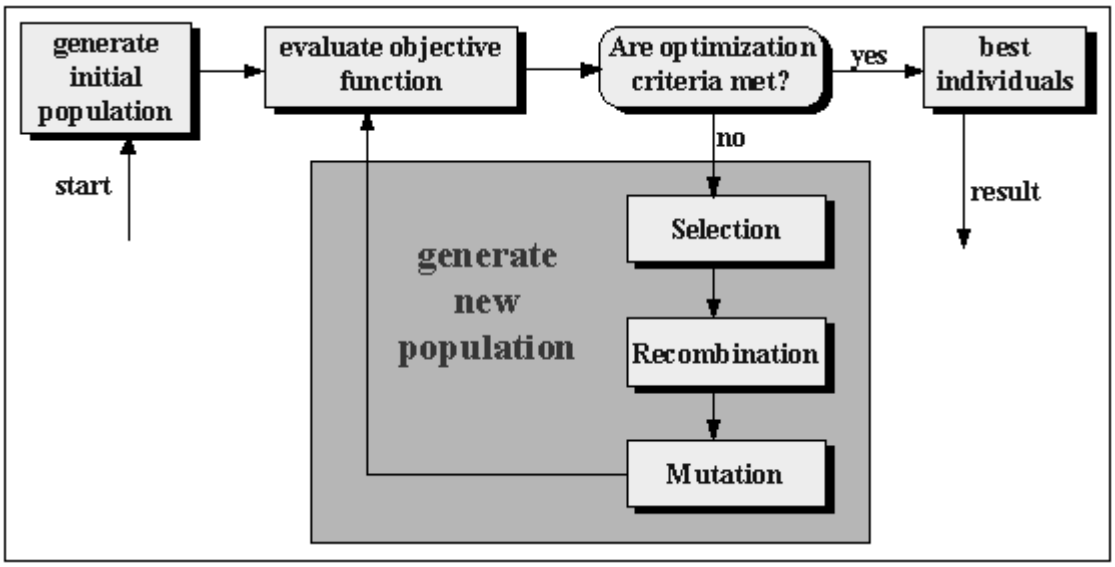

Figure 5.1 The structure of a genetic algorithm

At the beginning of the computation a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals. The first/initial generation is produced.

If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

Such a single population evolutionary algorithm is powerful and performs well on a broad class of problems. However, better results can be obtained by introducing many populations, called subpopulations. Every subpopulation evolves for a few generations isolated (like the single population evolutionary algorithm) before one or more individuals are exchanged between the subpopulations.

The Multi-population evolutionary algorithm models the evolution of a species in a way more similar to nature than the single population evolutionary algorithm.

From the above discussion, it can be seen that evolutionary algorithms differ substantially from more traditional search and optimization methods. The most significant differences are:

- Evolutionary algorithms search a population of points in parallel, not a single point.

- Evolutionary algorithms do not require derivative information or other auxiliary knowledge; only the objective function and corresponding fitness levels influence the directions of search.
- Evolutionary algorithms use probabilistic transition rules, not deterministic ones.
- Evolutionary algorithms are generally more straightforward to apply
- Evolutionary algorithms can provide a number of potential solutions to a given problem. The final choice is left to the user. (Thus, in cases where the particular problem does not have one individual solution, for example a family of pareto-optimal solutions, as in the case of multiobjective optimization and scheduling problems, then the evolutionary algorithm is potentially useful for identifying these alternative solutions simultaneously [12].

5.3.2 A Brief History of Genetic Algorithms

Genetic algorithms originated from the studies of cellular automata, conducted by John Holland and his colleagues at the University of Michigan. Holland's book [15], published in 1975, is generally acknowledged as the beginning of the research of genetic algorithms. Until the early 1980s, the research in genetic algorithms was mainly theoretical, with few real applications. This period is marked by ample work with fixed length binary representation in the domain of function optimization by, among others, De Jong and Holstein. Holstein's work provides a careful and detailed analysis of the effect that different selection and mating strategies have on the performance of a genetic algorithm. De Jong's work attempted to capture the features of the adaptive mechanisms in the family of genetic algorithms that constitute a robust search procedure [14].

From the early 1980s the community of genetic algorithms has experienced an abundance of applications which spread across a large range of disciplines. Each and every additional application gave a new perspective to the theory. Furthermore, in the process of improving performance as much as possible via tuning and specializing the genetic algorithm operators, new and important findings regarding the generality, robustness and applicability of genetic algorithms became available.

Following the last couple of years of furious development of genetic algorithms in the sciences, engineering and the business world, these algorithms in various guises have now been successfully applied to optimization problems, scheduling, data fitting and clustering, trend spotting and path finding.

5.3.3 Biological Background

All living organisms consist of cells. In each cell there is the same set of chromosomes. Chromosomes are strings of DNA and serve as a model for the whole organism. A chromosome consists of genes, blocks of DNA. Each gene encodes a particular protein. Basically, it can be said that each gene encodes a trait, for example color of eyes. Possible settings for a trait (e.g. blue, brown) are called alleles. Each gene has its own position in the chromosome. This position is called locus [13].

Complete set of genetic material (all chromosomes) is called genome. Particular set of genes in genome is called genotype. The genotype is with later development after birth base for the organism's phenotype, its physical and mental characteristics, such as eye color, intelligence etc.

During reproduction, recombination (or crossover) first occurs. Genes from parents combine to form a whole new chromosome. The newly created offspring can then be mutated. Mutation means that the elements of DNA are a bit changed. This changes are mainly caused by errors in copying genes from parents.

The fitness of an organism is measured by success of the organism in its life (survival).

5.3.4 Structure of Genetic Algorithms

Genetic algorithms are inspired by Darwin's theory of evolution. Solution to a problem solved by genetic algorithms uses an evolutionary process.

Algorithm begins with a set of solutions which are represented by chromosomes, called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are then selected to form new solutions (offspring) are selected according to their fitness - the more suitable they are the more chances they have to reproduce.

This is repeated until some condition, for example number of populations or improvement of the best solution, is satisfied.

Outline of the Basic Genetic Algorithm

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in the new population
4. **[Replace]** Use new generated population for a further run of the algorithm
5. **[Test]** If the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** Go to step **2**

5.3.4.1 Encoding

The first step in genetic algorithm is to "translate" the real problem into "biological terms". Format of chromosome is called encoding. A chromosome should in some way contain information about solution that it represents.

Each chromosome is represented by a binary string. Each bit in the string can represent some characteristics of the solution. There are many other ways of encoding which depends mainly on the solved problem. There are four commonly used encoding methods: binary encoding, permutation encoding, direct value encoding and tree encoding. Binary encoding is the most common one, mainly because the first research of GA used this type of encoding and because of its relative

simplicity. In binary encoding, every chromosome is a string of bits - 0 or 1. Binary encoding gives many possible chromosomes even with a small number of alleles. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation

Chromosome A	1011001011001010111100101
Chromosome B	1111111000001100000011111

Example of chromosomes with binary encoding

Permutation encoding can be used in "ordering problems", such as traveling salesman problem or task ordering problem. In permutation encoding, every chromosome is a string of numbers, which represents number in a sequence. For example:

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Example of chromosomes with permutation encoding

Direct value encoding can be used in problems where some complicated values such as real numbers are used. Use of binary encoding for this type of problems would be very difficult. In value encoding, every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or chars to some complicated objects

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Example of chromosomes with value encoding

Tree encoding is used mainly for evolving programs or expressions, for genetic programming. In tree encoding every chromosome is a tree of some objects, such as functions or commands in programming language.

5.3.4.2 Initial Population

A genetic algorithm starts with a population of strings to be able to generate successive populations of strings afterwards. The initialization is usually done randomly. This means, with binary strings for example, that every allele is set to 0 or 1, with each value having a chance of 50 % to occur.

5.3.4.3 Evaluation

After every generated population, every individual of the population must be evaluated to be able to distinguish between good and bad individuals. This is done by mapping the objective function to a 'fitness function'. The individuals are selected according to their fitness values and some random operators.

5.3.4.4 Reproduction

The reproduction operator allows individual strings to be copied for possible inclusion in the next generation. The chance that a string will be copied is based on the string's fitness value, calculated from a fitness function. For each generation, the reproduction operator chooses strings that are placed into a mating pool, which is used as the basis for creating the next generation.

There are many different types of reproduction operators. One always selects the fittest and discards the worst, statistically selecting the rest of the mating pool from the remainder of the population. These operators will perform better than others depending on the problem domain being explored.

Since describing all these operators is beyond the scope of this thesis, only the operators that are used in this work are presented.

5.3.4.5 Selection

Selection determines, which individuals are chosen for mating (recombination) and how many offspring each selected individual produces. In selection the individuals producing offspring are chosen. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own objective value and the objective value of all other individuals in the selection pool.

There are many methods in selecting the best chromosomes. Examples are roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others.

In tournament selection , a number Tour of individuals is chosen randomly from the population and the best individual from this group is selected as parent. This process is repeated as often as individuals to choose. These selected parents produce uniform at random offspring. The parameter for tournament selection is the tournament size Tour. Tour takes values ranging from 2 - N (number of individuals in population)

5.3.4.6 Crossover and Mutation

Once the mating pool is created, the next operator in the GA's arsenal comes into play. Remember that crossover in biological terms refers to the blending of chromosomes from the parents to produce new chromosomes for the offspring. The analogy carries over to crossover in genetic algorithms.

Depends on the nature of the problem and encoding type, implementation of crossover and mutation may vary greatly. The following are some general ideas on crossover and mutation for binary encoding

Crossover

Crossover operates on selected genes from parent chromosomes and creates new offspring. The simplest way how to do that is to choose randomly some crossover point and copy everything before this point from the first parent and then copy everything after the crossover point from the other parent.

Crossover can be illustrated as follows: (| is the crossover point):

Chromosome 1	**11011 | 00100110110**
Chromosome 2	11011 | 11000011110
Offspring 1	**11011** | 11000011110
Offspring 2	11011 | **00100110110**

There are other ways how to make crossover, for example we can choose more crossover points. Crossover can be quite complicated and depends mainly on

the encoding of chromosomes. Specific crossover made for a specific problem can improve performance of the genetic algorithm.

There are four often used methods for binary encoded crossover: single-point, two-point, uniform and arithmetic crossovers.

Single-point crossover:

One crossover point is selected. The binary string from beginning of parent 1 to its crossover point is copied to the new offspring on the same positions. The rest (from the same crossover point of parent 2 to its tail) is copied to the new offspring on the same positions. Here's an example

11001|011+1101|**111** = **11001111**

Two-point crossover:

Two crossover points are selected. The binary string from beginning of parent 1 to its first crossover point and the binary string from its second crossover point to its end are copied to the new offspring. The rest (the first crossover point of parent 2 to its second crossover point) is copied to the new offspring in the same fashion. Here's an example

11|0010|**11** + 11|**0111**|11 = **11011111**

Uniform crossover:

Bits are randomly copied from the first or from the second parent to the new offspring. Here's an example

1**10**0**10**11 + 1**10**111**01** = **11011111**

Mutation

After a crossover is performed, mutation takes place. Mutation is intended to prevent falling of all solutions in the population into a local optimum of the solved problem. Mutation operation randomly changes the offspring resulted from crossover. In case of binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1. Mutation can be then illustrated as follows:

Original offspring 1	1101**1**111000011110
Original offspring 2	1101100**0**100110110
Mutated offspring 1	1100**0**111000011110
Mutated offspring 2	1101101**1**00110110

The technique of mutation (as well as crossover) depends mainly on the encoding of chromosomes. For example when we are encoding permutations, mutation could be performed as an exchange of two genes.

5.3.4.7 Elitism

It's obvious that generating populations from only from two parents may cause you to loose the best chromosome from the last population. This is true, and so elitism is often used. This means, that at least one of a generation's best solution is copied without changes to a new population, so the best solution can survive to the succeeding generation. When creating a new population by crossover and mutation, we have a big chance, that we will loose the best chromosome.

Elitism is the name of the method that first copies the best chromosome (or few best chromosomes) to the new population. Elitism can rapidly increase the performance of GA, because it prevents a loss of the best found solution.

5.3.5 Micro Genetic Algorithms

The term micro-genetic algorithm (micro-GA) refers to a small-population genetic algorithm with reinitialization. The idea was suggested by some theoretical results obtained by Goldberg, according to which a population size of 3 was sufficient to converge, regardless of the chromosomic length. The process suggested by Goldberg, was to start with a small randomly generated population, then apply to it the genetic operators until reaching convergence (e.g., when all the individuals have their genotypes either identical or very similar), and then to generate a new population by transferring the best individuals of the converged population to the new one. The remaining individuals would be randomly generated [15].

In micro-GA firstly the convergence of the population is checked. To do this the number of different bits from the best member in the population is checked. If in the population only %5 of the number of the bits are different then it is concluded that the GA has been converged. After convergence test, Micro-GA is started again with the best individual of the previous population while the rest of the new population is filled with new randomly generated parents. During each cycle, the micro-GA undergoes conventional genetic operators: tournament selection, uniform crossover, uniform mutation, and elitism.

5.3.6. Parameters of Genetic Algorithm

Crossover probability:

It defines how often crossover will be performed. If there is no crossover, offspring are exact copies of parents. If there is crossover, offspring are made from parts of both parent's chromosome. If crossover probability is 100%, then all offspring are made by crossover. If it is 0%, whole new generation is made from exact copies of chromosomes from old population

Crossover is made in hope that new chromosomes will contain good parts of old chromosomes and therefore the new chromosomes will be better. However, it is good to leave some part of old population survive to next generation.

Crossover rate should be high generally, about 80%-95% (However some results show that for some problems crossover rate about 60% is the best.)

Mutation probability:

It defines how often parts of chromosome will be mutated. If there is no mutation, offspring are generated immediately after crossover (or directly copied) without any change. If mutation is performed, one or more parts of a chromosome are changed. If mutation probability is 100%, whole chromosome is changed, if it is 0%, nothing is changed.

Mutation generally prevents the GA from falling into local extremes. Mutation should not occur very often, because then GA will in fact change to random search.

Mutation rate should be very low. Best rates seems to be about 0.5%-1%

Population size:

It defines how many chromosomes are in population (in one generation). If there are too few chromosomes, GA have few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to use very large populations because it does not solve the problem faster than moderate sized populations.

It may be surprising, that very big population size usually does not improve performance of GA (in the sense of speed of finding solution). Good population size is about 20-30, however sometimes sizes 50-100 are reported as the best. Some research also shows, that the best population size depends on the size of encoded string (chromosomes). It means that if you have chromosomes with 32 bits, the population should be higher than for chromosomes with 16 bits.

6. B-SPLINE CURVES

6.1 Introduction

B-splines are widely used in computer graphics and computer aided design. B-splines are often used in numerically and differentiate that are defined only through a set of data points. B-splines are not really interpolating splines, because the curves do not normally pass through all of the points. B-splines have the important property of staying within the polygon determined by the given points. In addition b-splines have a nice geometric property in that in changing one of the points we only change one portion of the curve, a local effect. And finally, in contrast to other interpolating splines the points were given are not data points, but are more likely control points that we select to determine the shape of the curve we are working on.

6.2 Formulation of The Cubic B-Spline

B-spline curves can be of any degree, but since a cubic B-spline curve is used in this work, the concentration is on them for the formulation.

We begin with the description by stating the formula for a cubic B-spline in terms of parametric equations whose parameter is u.

Given the points $p_i = (x_i, y_i), \quad i = 0,1,...,n$, the cubic B-spline for the interval $(p_i, p_{i+1}), \quad i = 1,2,...,n-1,$ is

$B_i(u) = \sum_{k=-1}^{2} b_k p_{i+k}, \qquad$ where

$$b_{-1} = \frac{(1-u)^3}{6}, \tag{6.1}$$

$$b_0 = \frac{u^3}{2} - u^2 + \frac{2}{3},$$

$$b_1 = -\frac{u^3}{2} + \frac{u^2}{2} + \frac{u}{2} + \frac{1}{6},$$

$$b_2 = \frac{u^3}{6}, \qquad 0 \le u \le 1.$$

As before p_i refers to the point (x_i, y_i); it is a two-component vector. The coefficients, the b_k's, serve as a basis and do not change as we move from one set of points to the next. They can be considered weighting factors applied to the coordinates of a set of four points. The weighted sum, as u varies from 0 to 1, generates the B-spline curve.

If we write out the equations for x and y from Equation (6.1), we get

$$x_i(u) = \frac{1}{6}(1-u)^3 x_{i-1} + \frac{1}{6}(3u^3 - 6u^2 + 4)x_i + \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)x_{i+1} + \frac{1}{6}u^3 x_{i+2};$$

(6.2)

$$y_i(u) = \frac{1}{6}(1-u)^3 y_{i-1} + \frac{1}{6}(3u^3 - 6u^2 + 4)y_i + \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1)y_{i+1} + \frac{1}{6}u^3 y_{i+2};$$

$x_i(u)$ and $y_i(u)$ are functions and x_i, y_i are components of the point p. The u-cubics act as weighting factors on the coordinates of the four successive points to generate the curve.

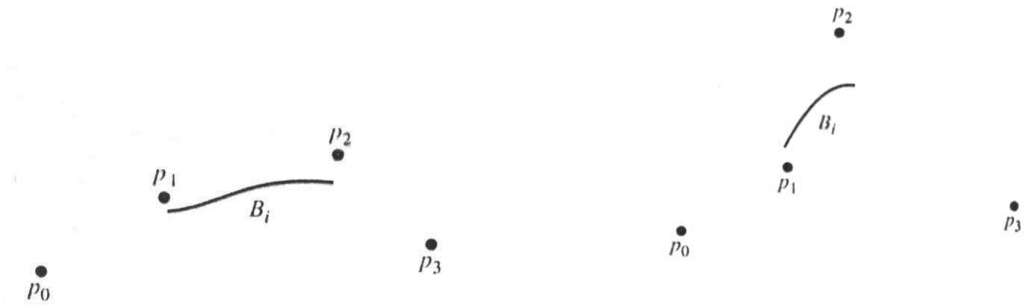

Figure 6.1 B-spline and control points

Because a set of four points is required to generate only a portion of the B-spline, that associated with two inner points, it must be considered how to get the B-spline for more than four points as well as how to extend the curve into the region outside of the middle pair. A method, marching along one point at a time, forming new sets of four.

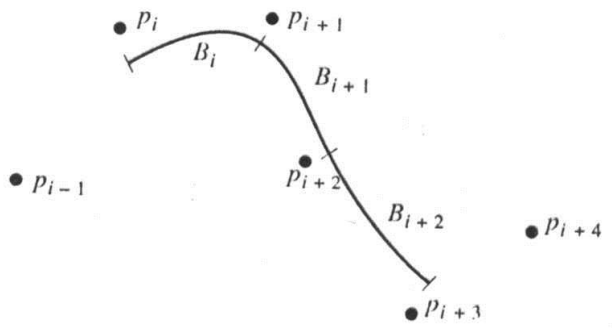

Figure 6.2 Successive B-Splines Joined Together

The conditions that we want to impose on the B-spline are ; continuity of the curve and its first and second derivatives. It turns out that the equations for weighting factors are such that these requirements are met.

Finally generation of the ends of the joined B-spline. If we have points from p_0 to p_n, we already can construct B-splines B_1 through B_{n-2}. We need B_0 and B_{n-1}. To overcome these problem we add fictitious points $p_{-2}, p_{-1}, p_{n+1}, p_{n+2}$ such that

$$p_{-2} = p_{-1} = p_0 \tag{6.3}$$

$$p_{n+1} = p_{n+2} = p_n$$

[16]

7. PROBLEM FORMULATION AND RESULTS

7.1 Approach

As mentioned before, this work tries to find an optimal shape of an airfoil considering aerodynamic concerns. Thus, the objective function and the constraints should be described in such a way that the requirements are met.

In order to keep a rotor trimmed in forward flight, the blade's angle of attack on the retreating side must be high enough to make up for its low velocity through the air. At some forward speed, the required angle of attack reaches the stall angle of airfoil and effectively limits the ability to fly any faster. The maximum would be increased by replacing the airfoil either with one that could go to a higher angle of attack, or with one that generates more lift at a specified angle of attack [17]. Therefore, a high lift coefficient, C_l is a desirable characteristic for any airfoil. Since the solver used in this work employs an inviscid panel method, we will not be able to determine the max lift coefficient, $C_{l_{\max}}$, which is the max lift obtained before stall. But instead we'll try to find an optimal airfoil shape with maximum lift for a number of given, α, angle of attack values. These values of α are chosen in such a way that they are all lower then the stall angle of the famous airfoil NACA 0012. Under these considerations, objective function, which we try to maximize, is stated as;

$$\text{maximize} \quad \zeta = \sum_{i=1}^{N} C_{l_i}$$

where N is the number of chosen α's for design. The number of α values used in this work are N=3, and their values are selected to be;

$\alpha_1 = 1.5$ i=1

$\alpha_2 = 5.5$ i=2

$\alpha_3 = 9.0$ i=3

,respectively. In other words, we have three flight conditions for the airfoil shape to be optimized.

Compared to the wing of an airplane, the blade of a helicopter is long and flexible. Any aerodynamic pitching moment generated by the airfoil will tend to twist the blade and produce oscillating loads in the control system. This nose-down pitching moment may cause aircraft to dive out of control [17]. Consequently, the moment coefficient , C_M, plays an important role in the design of rotor airfoils. Therefore, we will impose a constraint on the C_M into the objective function in order to overcome this problem. The constraints ensure that, for each flight condition, the airfoil generated by the GA must maintain a C_M smaller in magnitude than that of the airfoil NACA 0012. The constraints are imposed via an exterior penalty function for each of the design conditions. Because GA performs its search using only a fitness value must reflect the objective function value and any constraint violations. Finally the fitness function takes the form;

$$\text{maximize} \qquad \zeta = \sum_{i=1}^{3} C_{l_i} - P_i$$

where

$$P_i = \begin{cases} 100 \ , & |C_{M_i}| \geq |C_{M_i 0012}| \qquad i = 1,2,3 \\ 0 \ , & |C_{M_i}| \leq |C_{M_i 0012}| \qquad i = 1,2,3 \end{cases}$$

To keep the problem tractable, limits must be imposed on the position of each spline control point. Excessively limiting these parameters removes potentially beneficial designs from the search space, whereas excessive freedom wastes computing effort on very irregular designs. Figure 7.1 displays the limits used in this problem description. This search space still allows features like trailing-edge reflex and large amounts of camber while eliminating pointed leading edges and other unsolvable features.

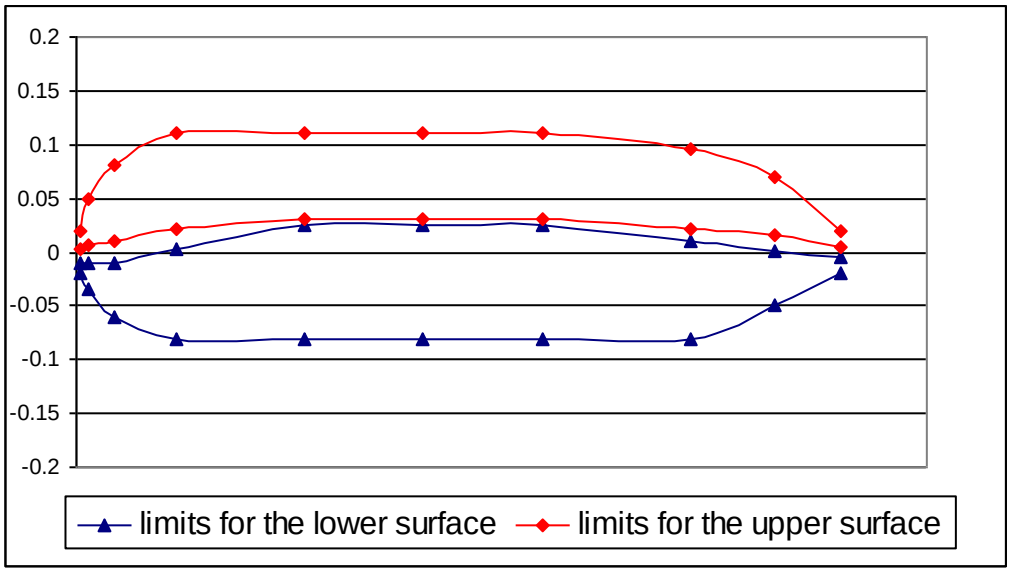

Figure 7.1 Limits of Control Points Representing Airfoil Surface

7.2 Airfoil Surface Representation

A third order B-spline with 22 control points described airfoil surfaces for this application. To define an airfoil, the chordwise coordinates of the B-spline control points remain fixed along a unit chord. Ordinate locations of the control points of the B-spline are used as design variables. Control points are arranged in such an order that they start from trailing edge, go along the lower surface round the leading edge to the upper surface and then back to trailing edge. Ten variables represent the upper surface and ten additional represented the lower surface. The control points at the trailing edge and the leading edge are fixed, so the remaining 20 points are the design variables for this problem. This approach gives a rational description of an airfoil for input into the panel code, without limiting shapes to those described by specific equations (like the NACA 4-series airfoils.)

Compared to other methods, a B-spline produces a smoother representation of irregular airfoils that are encountered by the GA, particularly in the initial randomly generated population. The smoothing effect of the B-spline allows some analysis to be conducted, even on irregular shapes [2].

For the representation of the airfoils a B-spline Curve Generator code is used. A sample representation of the NACA 0012 airfoil generated by the code is in

Figure 7.2. The figure shows that even with the limited resolution of data, the spline is able to closely represent the airfoil.

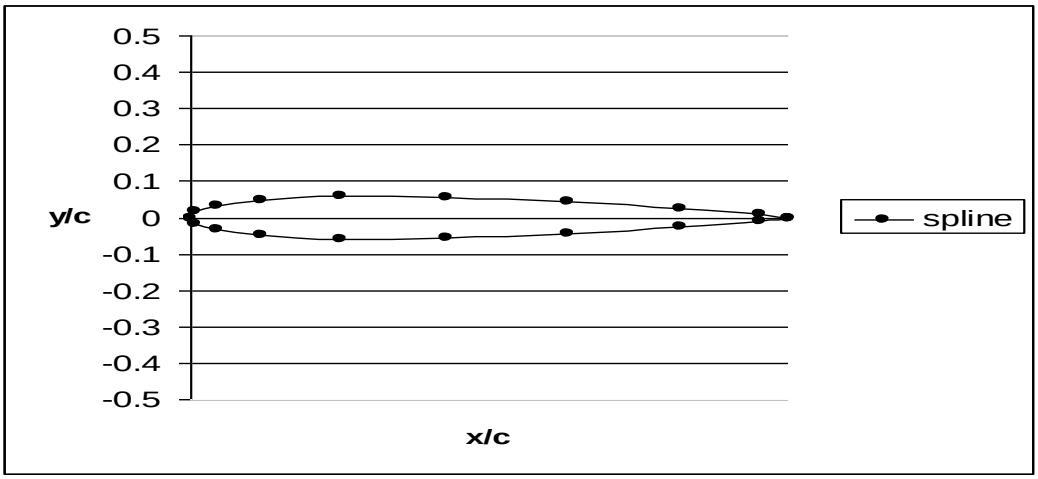

Figure 7.2 Spline control points and representative surface of NACA 0012

7.3 Aerodynamic Analysis Validation

An inviscid panel method called Smith&Hess method provided the aerodynamic analysis tool for this work. Using this allowed a wide range of shapes to be evaluated and provided reasonable estimates of lift and pitching moment coefficients.

Aerodynamic properties are calculated using 150 panels distributed over the airfoil surface represented by the B-splines. To test the accuracy of the code, the values of C_L and C_M for NACA 0012 are compared with the results from another solver. This solver is a well-established airfoil analysis code called XFOIL [18]. Figures 7.3 and 7.4 show the comparison between the values acquired from Smith&Hess panel code and XFOIL code.

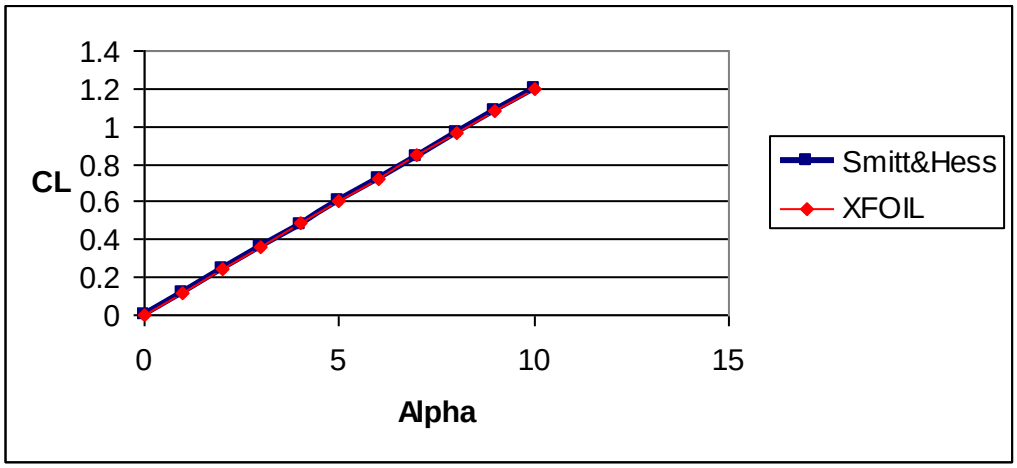

Figure 7.3 Comparison of CL values

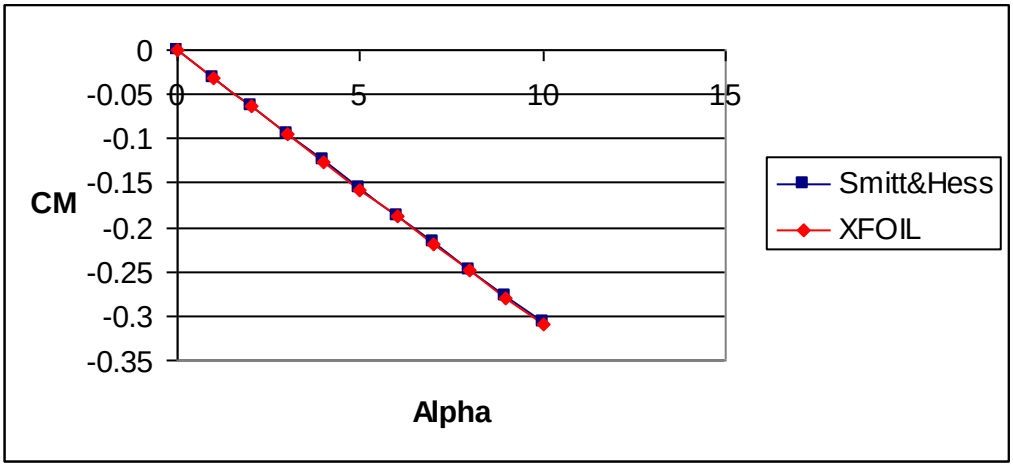

Figure 7.4 Comparison of CM values

As can be seen from the figures above, the values of CL and CM are very close to each other. Note that the moment coefficient is computed on the leading edge in both methods.

7.4 Genetic Algorithm Organization

The genetic algorithm used in this work [8] is a serial micro-GA with a binary tournament selection routine, uniform crossover and an elitism operator. Design variables are coded to binary chromosomes for use in the GA with a binary encoding scheme, so all variables are actually discretized.

The micro-GA algorithm is used since it requires considerably less function evaluations than the traditional GA. This is because the micro-GA works with a population size of 5 while traditional GA has a population size of 50.

There are 20 design variables for optimization and limits for each of the variable is shown previously in Figure 7.1. Four are used as the allele to represent the y/c location of each control point. The resolution can be any value; a finer resolution requires more bits in the chromosome. Since geometry definition and encoding procedure specifies the gene length, a chromosome length of 80 is required.

A crossover probability of %50 and a mutation probability of %2 is used in this work. The population size is determined to be $N_{pop} = 5$ since micro-GA works with small population sizes. The initial population is randomly generated. The algorithm employs an elitism operator, so the best individual in each generation is stored. The convergence criteria for this algorithm is not supplied, but the maximum number of generations is supplied. The maximum number of generations is determined to be $maxgen = 1000$.

7.5 Results and Discussion

Since the GA is a guided search method with some random elements, the same input conditions could possibly result in a different optimal solution. However, these solutions will likely be similar.

Figure 7.5 shows a plot of convergence history of the best run.

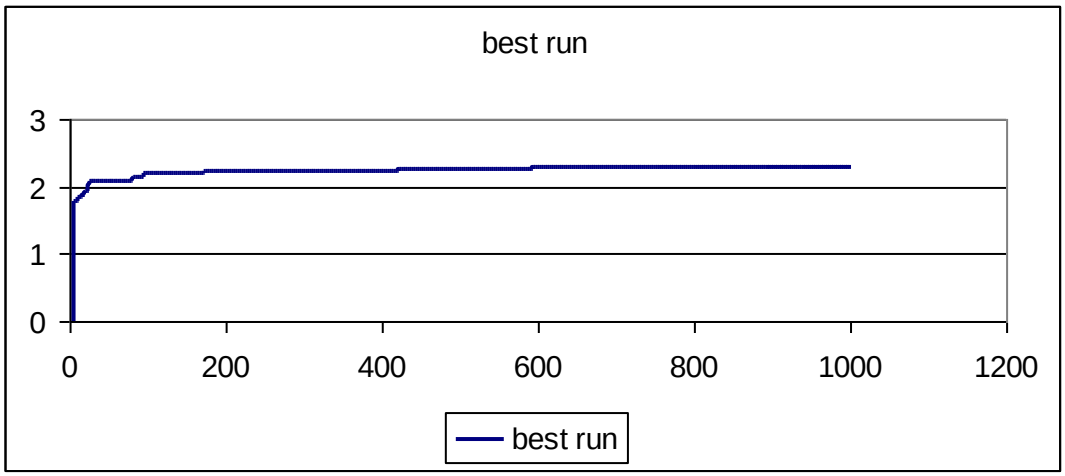

Figure 7.5 Convergence history of the best run

During the process of airfoil evolution, the GA mates individuals with high fitness values in an attempt to maximize the fitness function. The designs move from the initial, randomly generated population through a series of shapes that could be

considered highly unconventional airfoils. As the generations advance, the airfoils become more reasonable and eventually a feasible design is obtained.

Since a certain constraint for the moment coefficients is imposed, the GA eliminates the shapes that violate this constraint in the first few generations. Then the algorithm finds an initial shape that would be fine tuned by the next generations.. An example shape that was generated in generation-0 is shown in figure 7.6.

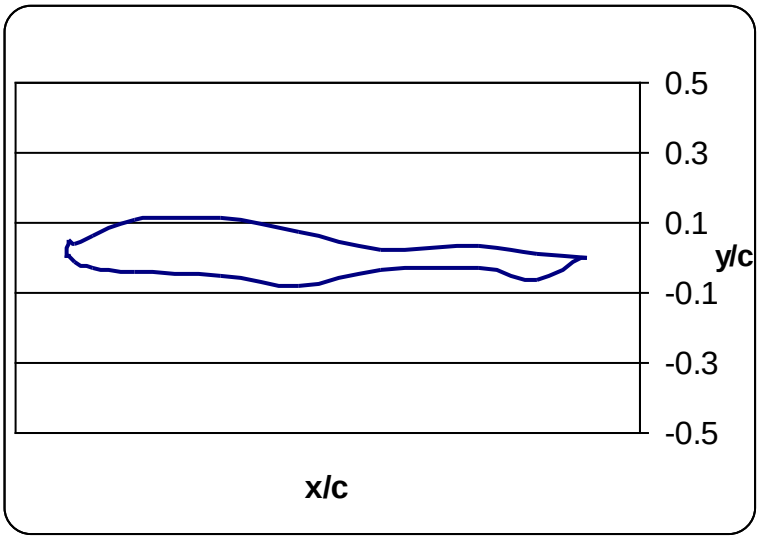

Figure 7.6 An airfoil shape from generation-0 that violates the CM constraint

After the 10. generation the GA finds a shape that doesn't violate CM constraint. In Figure 7.7 the best airfoil shape from generation-10 is shown.

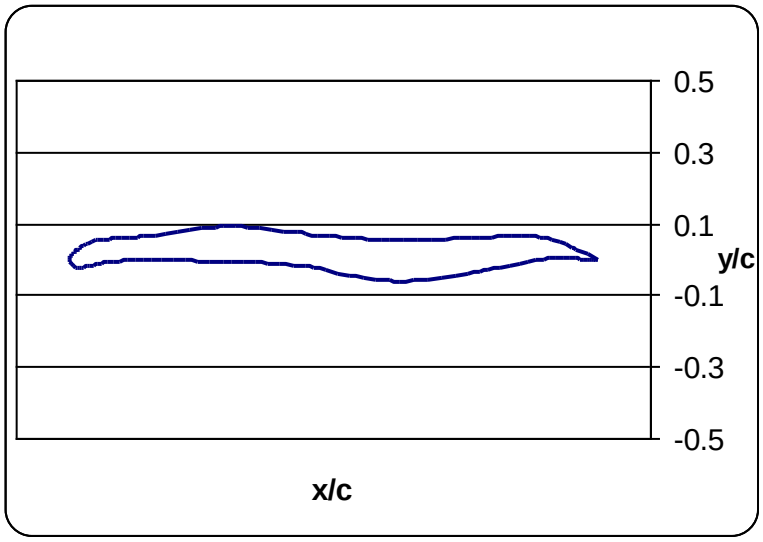

Figure 7.7 The best airfoil in generation-10

To give a felling about the finetuning process the best airfoil shape in generation-50 is shown in Figure 7.8.

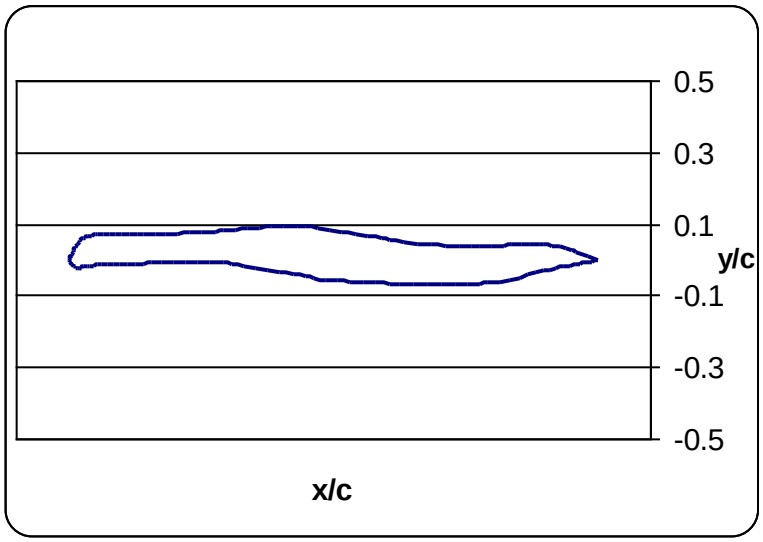

Figure 7.8 The best airfoil from generation-50

After generation-10, GA starts to finetune the shape and finally after 200 generations a reasonable airfoil shape is found. In Figure 7.9 the design which was generated after 200 iterations is shown. To improve the shape the algorithm is computed to higher number of iterations and finally after 1000 iterations the optimum shape is found. This airfoil shape is the optimum solution to our problem and it is shown in Figure 7.10.

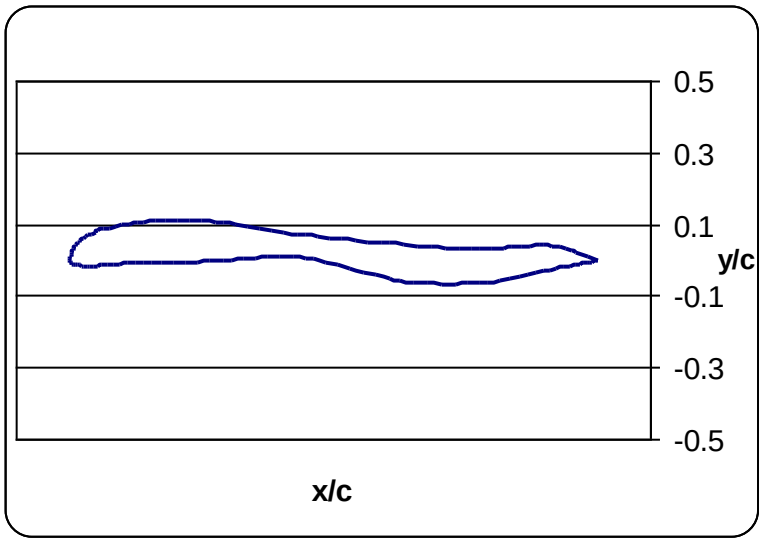

Figure 7.9 The best airfoil from generation-200

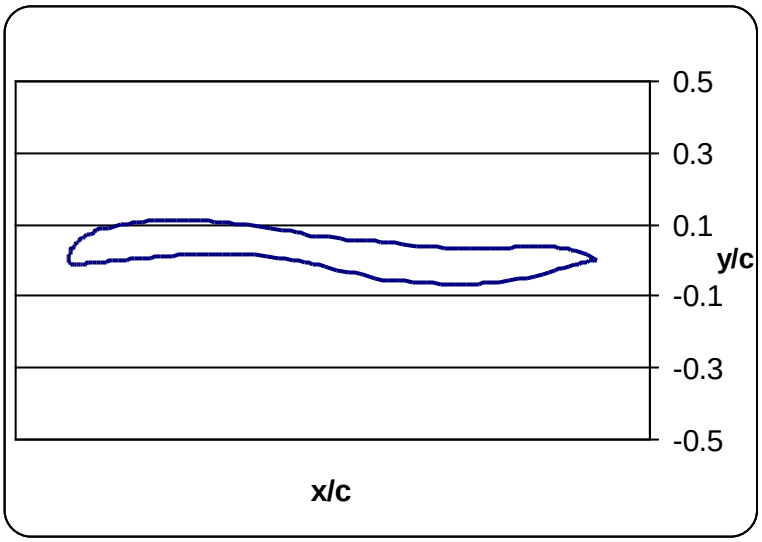

Figure 7.10 Final Design - The best airfoil from generation-1000

The following figure shows the pressure distributions for the final airfoil shape. The Cp values for α=9° is plotted against the chordwise locations (x/c).

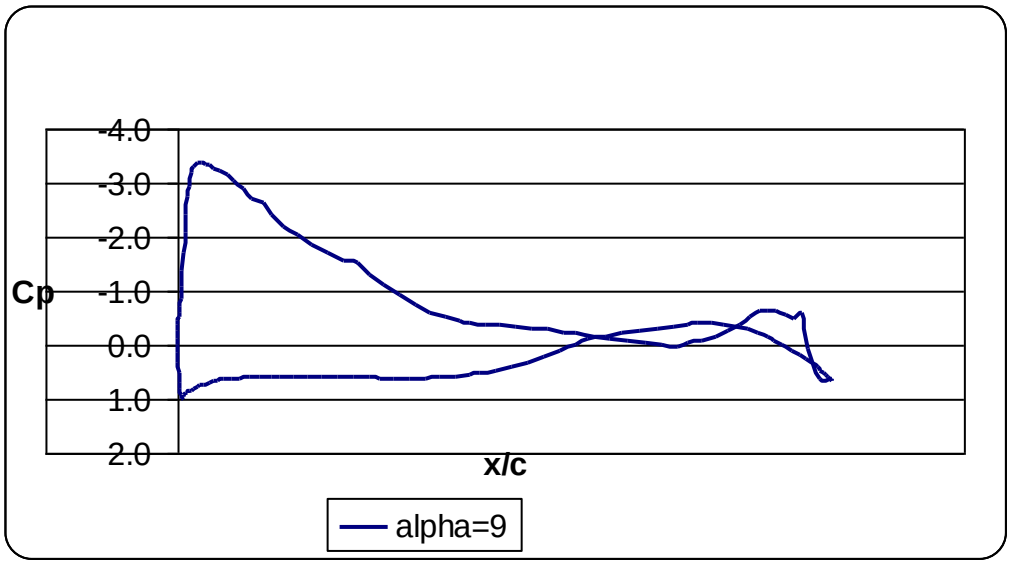

Figure 7.11 Pressure Distributions of the final airfoil shape for $\alpha = 9.0^{o}$

Table 7.1 presents how the GA maximizes the fitness function during evolutionary process. Fitness function values of best airfoils from several generations are shown with their corresponding C_L and C_M values.

generation	alpha	CL	CL0012	CM	CM0012	fitness value
	1.5	0.769502	0.18063	-0.329613	-0.04695	
0	5.5	1.259317	0.66136	-0.464148	-0.17117	**-296.28797**
	9	1.683211	1.07941	-0.57769	-0.27721	
	1.5	0.126761	0.18063	-0.013507	-0.04695	
10	5.5	0.741768	0.66136	-0.151648	-0.17117	**2.048296**
	9	1.179767	1.07941	-0.273775	-0.27721	
	1.5	0.210525	0.18063	-0.013507	-0.04695	
30	5.5	0.712021	0.66136	-0.154369	-0.17117	**2.07039**
	9	1.147844	1.07941	-0.275831	-0.27721	
	1.5	0.237607	0.18063	-0.01042	-0.04695	
100	5.5	0.741768	0.66136	-0.151648	-0.17117	**2.159142**
	9	1.179767	1.07941	-0.273775	-0.27721	
	1.5	0.26635	0.18063	-0.018901	-0.04695	
200	5.5	0.763371	0.66136	-0.156448	-0.17117	**2.224448**
	9	1.194727	1.07941	-0.275439	-0.27721	
	1.5	0.271448	0.18063	-0.01777	-0.04695	
500	5.5	0.767867	0.66136	-0.155624	-0.17117	**2.23798795**
	9	1.198674	1.07941	-0.27495	-0.27721	
	1.5	0.290319	0.18063	-0.0197	-0.04695	
1000	5.5	0.785952	0.66136	-0.156637	-0.17117	**2.29243134**
	9	1.216161	1.07941	-0.275211	-0.27721	

Table 7.1. Best airfoil properties from several generations and comparisons with NACA 0012 aerodynamic coefficients

In Table 7.1 the aerodynamic characteristics of resulting optimal design which was generated by genetic algorithms are compared to those of the NACA 0012 airfoil. As seen above the GA managed to generate an optimum airfoil having better aerodynamic properties than NACA 0012 airfoil.

To evaluate the accuracy of the algorithm, another test case is implemented. In this case the constraints on the moment coefficients are adapted from those of VR7 airfoil with 0° tab while basic the structure of the cost function remains the same. The same conditions used for the previous run are applied. After 1000

iterations the algorithm creates a final design which appear to have better aerodynamic properties than the VR7 airfoil as shown in Table 7.2. The final design and its pressure distributions for α =9° are shown in figures 7.12 and 7.13 respectively.

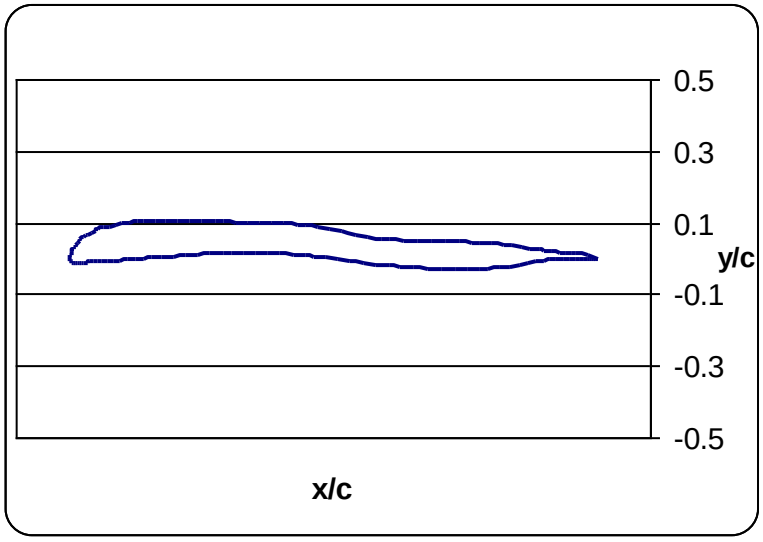

Figure 7.52 Final Design - The best airfoil from generation-1000

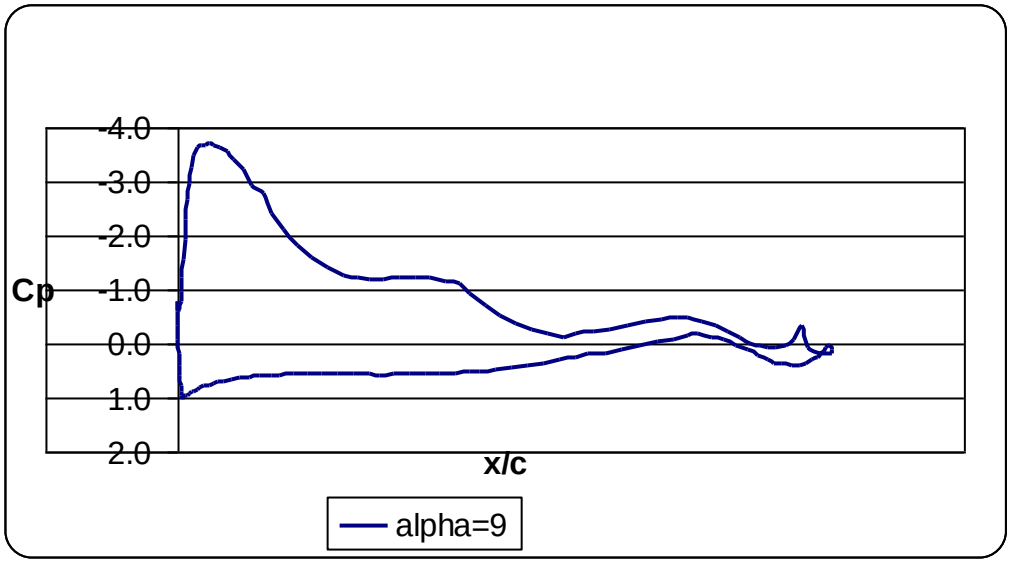

Figure 7.13 Pressure Distributions of the final airfoil shape for $\alpha = 9.0^{o}$

generation	alpha	CL	CLVR7	CM	CMVR7	fitness value
	1.5	0.521645	0.4725	-0.1516666	-0.1642	
1000	5.5	1.003042	0.9467	-0.278873	-0.2862	**2.94483**
	9	1.420147	1.3581	-0.388449	-0.3905	

Table 7.2 Aerodynamic properties of the best design compared to those of VR7 airfoil

The second approach to the problem is to formulate it with a different cost function. This time the moment coefficients take an active part in the objective function instead of being an unbreakable constraint. Therefore the moment coefficients are embedded in the cost function via an exterior penalty function. Below is the fitness function of this case;

$$\text{maximize} \quad \xi = \sum_{i=3}^{N-f_c} \frac{C_{Li}}{C_{Li_{0012}}} - R * P_i$$

where $P_i = \max(0, g_i)$

and $g_i = \frac{|C_M|_i}{|C_{M_{0012}}|_i} - 1$

Here R is the penalty coefficient and it is determined to be R=0.5 due to results from several runs. The other conditions are just the same as the previous cases. After 1000 iterations, the algorithm generates the optimum shape of the airfoil which is shown in Figure 7.14. The pressure distributions are presented in Figure 7.15.

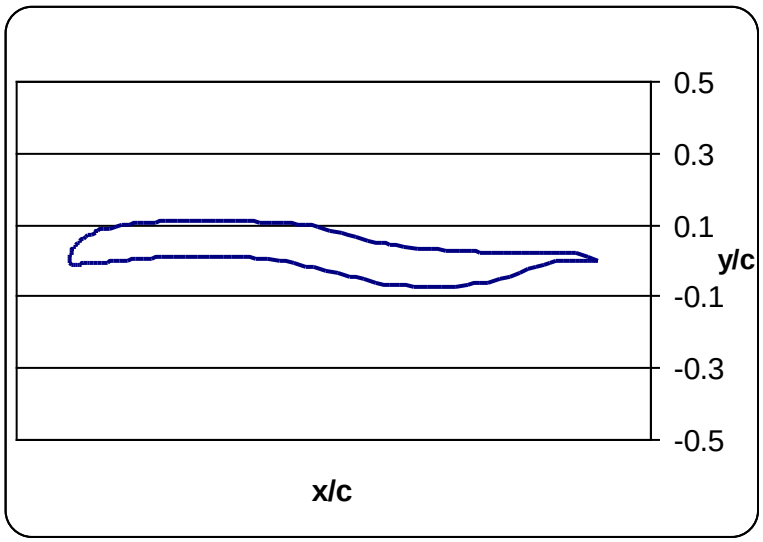

Figure 7.14 Final Design - The best airfoil from generation-1000

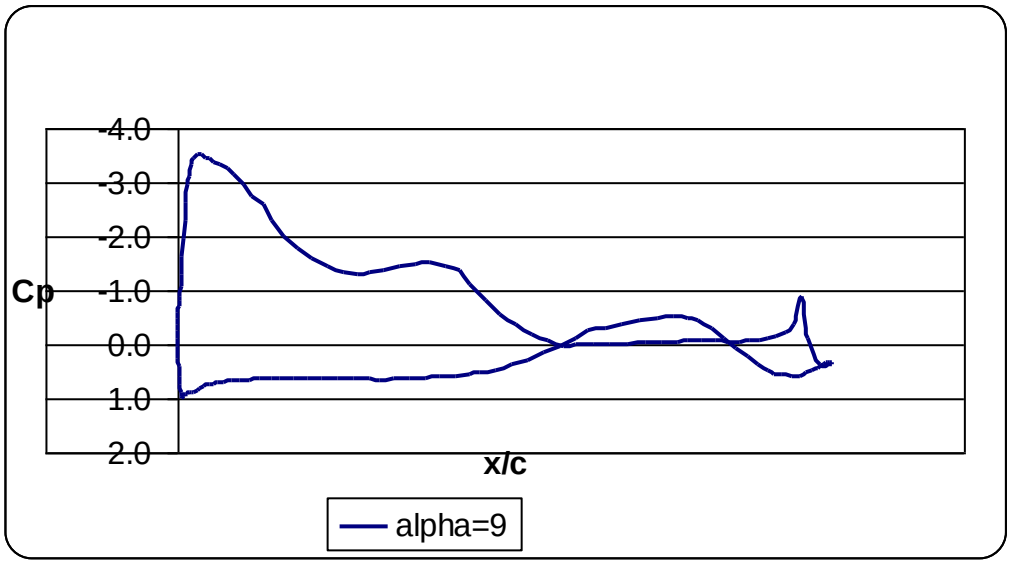

Figure 7.15 Pressure Distributions of the final airfoil shape for $\alpha = 9.0^{o}$

generation	alpha	CL	CL0012	CM	CM0012	fitness value
	1.5	0.398052	0.18063	-0.068952	-0.04695	
1000	5.5	0.888615	0.66136	-0.20197	-0.17117	**1.1458750**
	9	1.314202	1.07941	-0.316931	-0.27721	

Table 7.3 Aerodynamic properties of the best design compared to those of NACA 0012 airfoil

7.6 Conclusion

The genetic algorithm method for airfoil shape design has been shown to exhibit some advantages over traditional design techniques, most notably, the ability to generate designs that would not have otherwise been considered. This is because of the algorithm's population-based search and global-optimization behavior. Based on the prediction of the Smith&Hess panel code, the resulting non-traditional shapes appear to have good aerodynamic performance.

One another fact is the efficiency of the aerodynamic prediction tool determines the efficiency of the optimization. Although Smitt&Hess's low-order panel code appears to have successfully analyzed the non-traditional shapes, a more robust aerodynamic analysis using higher-order panel methods may provide better optimization. In other words, given increased computational capabilities, the genetic algorithm would be able to generate designs with a higher level of confidence in the aerodynamic predictions.

Another important issue is the absence of non-aero-dynamic constraints in the airfoil design problem. Since the helicopter engineering problems include the important effects of interdisciplinary dependencies, such as aerodynamic, structural, aeroacoustic, and dynamic, they must be formulated around the use of multi-disciplinary design optimization techniques [19]. In spite of the recent developments in rotorcraft multi-disciplinary design, more work has to be performed in this field [3].

REFERENCES

[1] **Goldberg, D.E.**, 1989. Genetic algorithms in Search, Optimization and Machine Learning, Addison Wesley Longman, Reading, MA,

[2] **Jones, B.R, Crossley, W.A. and Lyrintzis, A.S.**, 2000. Aerodynamic and Aeroacoustic Optimization of Rotorcraft Airfoils via a Parallel Genetic Algorithm, *Journal of Aircraft*, **37,** 1088-1096

[3] **Celi, R.**, 1999. Recent Applications of Design Optimization to Rotorcraft - A Survey, *Journal of Aircraft*, **36**, 176-189

[4] **Obayashi, S. and Takanashi, S.**, 1996. Genetic Optimization of Target Pressure Distributions for Inverse Design Methods, *AIAA Journal*, **34**, 881-886

[5] **Obayashi, S., Jeong, S. and Matsuo, Y.**, 1997. New Blunt Trailing Edge Airfoil Design by Inverse Optimization Method, *Journal of Aircraft*, **34**, 255-257

[6] **Quagliarella, D. and Della Cioppa, A.**, 1995. Genetic Algorithms Applied to the Aerodynamic Design of Transonic airfoils, *Journal of Aircraft*, **32**, 889-981

[7] **Lee, J. and Hajela, P.**, 1996. Parallel Genetic Algorithm Implementation in Multidisciplinary Rotor Blade Design, *Journal of Aircraft*, **33**, 962-970

[8] **Carroll, D.L.,** 2001. FORTRAN Genetic Algorithm (GA) Driver, http://cuaerospace.com/carroll/ga.html

[9] **Anderson, J.D,** 1991. Fundamentals of Aerodynamics, McGraw-Hill, New York

[10] **Hess, L.,** 1990. Panel Methods in Computational Fluid Dynamics, *Annual Review of Fluid Mechanics*, **22**, 255-274

[11] **Lorena A.B**, 1999. Panel Methods for the Computation of Flows around Arbitrary Bodies, Ae-100 Research, California Institute of Technology

[12] Genetic and Evolutionary Algorithm Toolbox for use with Matlab http://www.geatbx.com/docu/algindex.html

[13] Introduction to Genetic Algorithms with Java Applets http://cs.felk.cvut.cz/~xobitko/ga/

[14] An introduction to Genetic Algorithms http://www.cs.qub.ac.uk/~M.Sullivan/ga/ga4.html

[15] **Holland, J.H**, 1992. Adaptation in Natural and Artificial Systems. Cambridge: MIT Press, 1992.

[16] **Gerald, C.F. and Wheatley, P.O,** 1999. Applied Numerical Analysis, Addison Wesley, Reading

[17] **Prouty, R.W,** 1995. Helicopter Performance, Stability and Control, Krieger Publishing Company, Florida

[18] **Drela, M,** 1989. XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils, Proceedings of The Third International conference on Aerodynamics at Low Reynolds Numbers, New York, 1-12

[19] **Straub, F. K., Callahan, C. B. and Culp, J. D.,** 1992. Rotor Design Optimization Using a Multidisciplinary Approach, *Structural Optimization*, 5, 70-75

CURRICULUM VITAE

A. Taylan Köksal was born in Ankara on August 28, 1977. He graduated from Beşiktaş Atatürk A. High School in 1995. He obtained his B.Sc. degree from Istanbul Technical University, Department of Astronautical Engineering, in 2000. Same year he started M.Sc. studies in Astronautical Engineering at Institute of Science and Technology of ITU and he has been working as a research assistant in the Faculty of Aeronautics and Astronautics since 2000.