5-8-2004

Finite Element Analysis and Sensitivity Analysis for the Potential Equation

Marco G F Capozzi

FINITE ELEMENT ANALYSIS AND SENSITIVITY ANALYSIS FOR THE POTENTIAL EQUATION

By

Marco G.F. Capozzi

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Aerospace Engineering
in the Bagley College of Engineering

Mississippi State, Mississippi

May 2004

FINITE ELEMENT ANALYSIS AND SENSITIVITY ANALYSIS FOR THE POTENTIAL EQUATION

By

Marco G.F. Capozzi

Approved:

____
James C. Newman III
Associate Professor of Aerospace
engineering
(Director of Thesis)

____
Pasquale Cinnella
Professor of Aerospace
Engineering
(Committee Member)

____
J. Mark Janus
Associate Professor of Aerospace
Engineering
(Committee Member)

____
Greg W. Burgreen
Associate Research Professor
Engineering Research Center
(Committee Member)

____
A. Wayne Bennett
Dean of the Bagley College
of Engineering

____
Pasquale Cinnella
(Graduate Director of the
Department of Aerospace
Engineering)

Name: Marco G.F. Capozzi

Date of Degree: May , 2004

Institution: Mississippi State University

Major Field: Aerospace Engineering

Major Professor: Dr. James C. Newman III

Title of Study: FINITE ELEMENT ANALYSIS AND SENSITIVITY ANALYSIS FOR THE POTENTIAL EQUATION

Pages in Study: 63

Candidate for Degree of Master of Science

A finite element solver has been developed for performing analysis and sensitivity analysis with Poisson's equation. An application of Poisson's equation in fluid dynamics is that of potential flow, in which case Poisson's equation reduces to Laplace's equation. The stiffness matrix and sensitivity of the stiffness matrix are evaluated by direct integration, as opposed to numerical integration. This allows less computational effort and minimizes the sources of computational errors. The capability of evaluating sensitivity derivatives has been added in order to perform design sensitivity analysis of non-lifting airfoils. The discrete-direct approach to sensitivity analysis is utilized in the current work. The potential flow equations and the sensitivity equations are computed by using a preconditioned conjugate gradient method. This method greatly reduces the time required to perform analysis, and the subsequent design optimization. Airfoil shape is updated at each design iteration by using a Bezier-Berstein surface parameterization. The unstrucured grid is adapted considering the mesh as a system of inteconnected springs. Numerical solutions from the flow solver are compared with analytical results obtained for a Joukowsky airfoil. Sensitivity derivatives are validated

using carefully determined central finite difference values. The developed software is then used to perform inverse design of a NACA 0012 and a multi-element airfoil.

DEDICATION

I'd like to dedicate this work to all those people who have supported me in this work.

ACKNOWLEDGMENTS

I would like to express my appreciation to my major professor, Dr. James C. Newman III, who has provided valuable moral support, patience and friendship through the many years that we have known each other. I also would like to thank Dr. Pasquale Cinnella, who has allowed me to make this wonderful experience here, at ERC, and has also helped me many times. I'd like to remember all the people without whom I could never have been able to finish what I started two years ago: in particular, Phil, Gazi, Amalia and Raavi. In addition, I would like to thank the Engineering Research Center for providing financial support and facilities for this research effort. Last, but not least, a special thanks goes to Dr. M. Napolitano, from Politecnico di Bari, who has allowed me to come here at Mississippi State University.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Identifiers:

k	Iteration Number
m	Number of Nodes per Element
n_x	x Component of the Normal
n_y	y Component of the Normal
Q	Vector of Secondary Variables
DV	Design Variables
FD	Finite Difference
FEM	Finite Element Method
GS	Gauss-Seidel
MEA	Multielement Airfoil
NDV	Number of Design Variables
N_i	Interpolation Function
PCG	Preconditioned Conjugate Gradient
α	Jacobian Component
β	Jacobian Component
γ	Jacobian Component
η	Design Variable
λ	Eigenvalue
ω	Relaxation Factor
Γ^e	Element Boundary Length

Fluid Dynamic Variables:

p	Fluid Pressure
p_∞	Far Field Fluid Pressure
u	Fluid Velocity Vector along x
v	Fluid Velocity Vector along y
w	Fluid Velocity Vector along z
$\mathbf{v}_\infty$	Far Field Fluid Velocity
A	Area of a Cell
V	Volume of a Cell
C_P	Coefficient of Pressure
ρ	Fluid Density
Φ	Fluid potential

Tensor Operators:

$\nabla\cdot()$	Divergence
$\nabla()$	Gradient
∇x()	Curl
I	Identity Matrix

CHAPTER I

INTRODUCTION

1.1 Motivation

Design sensitivity analysis investigates methodologies to calculate the derivatives of system output with respect to the system input. Sensitivity analysis has become a mature field, but in the past, due to the lack of sufficient computer power, it has been difficult to perform such a task on a large scale problem. Based on the information provided by sensitivity derivatives, this type of analysis may be particularly useful for complex systems (e.g., ships, aircrafts and automotive vehicles).

Engineering systems of practical interest are usually characterized by complex interactions occurring between various disciplines. Sometimes it is possible to decouple the problem into the separate disciplines, some other times it is impossible or it may lead to misleading results. In those cases in which decoupling is not possible, sensitivity analysis represents a powerful tool, since it allows the designer to quantify the impact of each discipline on the overall performance of the system. In multidisciplinary systems, sensitivity analysis and design has been termed Multidisciplinary Design Optimization. It provides the sensitivity of the design with respect to any set of independent variables (regardless of which discipline the variable originates from), so that one can effectively analyze and interpret the response of the system performance to these variations.

The motivation of the current work is to develop an efficient analysis and sensitivity analysis procedure to be used for the aforementioned purposes. As previously noted, large scale problems that utilize high-fidelity analysis (e.g., Reynolds' Averaged Navier-Stokes

Equations) become prohibitively expensive. However, lower-fidelity analysis offers an efficient computational alternative that may lead to a qualitatively better understanding of the problem, may be used to perform numerous design or parametric studies, or gauge the impact of various disciplines in multidisciplinary systems.

1.2 Brief Survey of Sensitivity Analysis and Optimization

Sensitivity analysis and optimization has been used by engineers for several centuries. In mechanics, most of the fundamental solutions or equations are derived from energy principles, which are nothing more than using variational methods to arrive at a stationary point. However, these solutions may only be easily found analytically for regular geometries or highly simplified analysis.

With the advent of digital computer more realistic geometry and high-fidelity analysis could be investigated. Sensitivity analysis for these systems then required the development of supplementary numerical techniques, and from which the continuous (variational) and the discrete formulations were born. Additionally, these formulations could be cast into both adjoint and direct approaches.

The first discipline to embrace numerical analysis, as well as sensitivity analysis and design optimization, was solid and structural mechanics. Analysis of structures was adopted as routine, primarily by the aeronautical industry, in the early mid-50's [3]. By the late-50's and early-60's, sensitivity analysis and optimization of structural components became common place [4], [5]. Aerodynamic design began to emerge in the late-60's, [6] [7], however, it was not until the mid-70's that its use with low-fidelity models became routine [8] [9]. Moreover, these low-fidelity models primarily focused on isolated airfoils and wings, and used finite-difference evaluation of derivative information. Then, in the mid-80's, Sobieski [10] challenged the aeronautics community to include shape sensitivity analysis of the geometry with the use of high-fidelity models. This plea ignited intense studies and research, which still continue today.

Detailed reviews of sensitivity analysis and design for structural systems has been reported in [11] [12] and for aeronautical systems in [7] [13]. The reader is urged to seek these sources for a complete survey of research activities in these disciplines.

1.3 Objectives of the Present Work

The primary objective of the present work is to develop an efficient analysis and sensitivity analysis method for Poisson's equation. Poisson's equation has physical significance in numerous disciplines. For example, Poisson's equation can be used to model the irrotational flow of an ideal fluid, conductive heat transfer, ground water flow, torsion of constant cross-sectional members, transverse deflection of elastic membranes, electrostatics and magneto statics.

The current work uses the irrotational flow of an ideal fluid to demonstrate and validate the analysis, sensitivity analysis, and design capabilities developed. For the irrotational flow of an ideal fluid (potential flow) the dependent variable can represent either the stream function or the velocity potential. The velocity potential formulation was chosen to facilitate the three-dimensional analysis. Furthermore, sensitivity analysis for this system of equations is performed via a discrete-direct formulation.[1]

The analysis and sensitivity analysis of Poisson's equation is accomplished by the finite element method. This discretization results in a symmetric, positive-definite system of equations, which is solved for the dependent variable. Since this system is symmetric and positive-definite, direct methods such as Cholesky LU decomposition may be used for solving the algebraic equations. However, the memory requirements of direct methods become prohibitive for large scale problems. Hence, in current work, these equations are solved via iterative techniques such as Jacobi iteration, Gauss-Seidel, and preconditioned conjugate gradient.

The secondary objective of the present work was to demonstrate that with the developed analysis and sensitivity analysis capabilities, design optimization was possible.

In the present study, design optimization consisted of the inverse design of surface geometry, whereby a target pressure distribution was matched. The objective function is thus the sum of the squares of the difference between computed and target pressure; which represents a least-squares function. A modified Gauss-Newton (Levemberg-Marquardt) method was used for numerical optimization.

CHAPTER II

FUNDAMENTAL EQUATIONS AND DISCRETIZATION METHOD

2.1 Fluid Dynamics Equations

As previously mentioned, Poisson's equation has application in many disciplines. In the current work, the use of Poisson's equation as it pertains to incompressible, irrotational flow of an ideal fluid will be used for demonstration and validation.

The fundamental equations of fluid dynamics are based on the conservation of mass, momentum, and energy. The complete set of equations are referred to as the Navier-Stokes equations. Often many assumptions are made that will reduce the complexity of this system of equations.These assumption are used to lessen the computational expense and, thus, provide a qualitative analysis of a given problem.

First, neglecting the viscous effects will lead to the elimination of the viscous stress tensor. The resulting equations from this assumption are referred to as the Euler Equations. Further, if it is assumed that the flow may be considered as incompressible, then the density remains constant both spatially and temporally. Additionally, assuming that the flow is irrotational ($\nabla x(\mathbf{v}) = 0$) yields the potential flow equations.

Introducing a function Φ, called the velocity potential, the fluid velocities in the coordinate directions may be expressed in terms of Φ as:

$$\begin{aligned} u &= -\frac{\partial \Phi}{\partial x}, \\ v &= -\frac{\partial \Phi}{\partial y}, \end{aligned} \tag{2.1}$$

$$w = -\frac{\partial \Phi}{\partial z}.$$

Or, in a more compact form as:

$$\mathbf{v} = -\nabla \boldsymbol{\Phi}. \tag{2.2}$$

This definition satisfies the condition of irrotationality ($\nabla x(\mathbf{v}) = 0$) exactly. Under the assumption of incompressibility, the continuity (conservation of mass) equation reduces to:

$$\nabla \cdot \mathbf{v} = 0. \tag{2.3}$$

Substitution of Eq. 2.2 into 2.3 yields

$$\nabla^2 \Phi = 0. \tag{2.4}$$

Hence, for potential flow the source term in Poisson's equation is identically zero, and the result is typically referred to as Laplace's equation.

To evaluate this boundary value equation, conditions must be imposed on the boundaries of the domain. The boundary conditions may be of Dirichlet, Neumann, or Robin (mixed) type. For a Dirichlet condition the value of the dependent variable (velocity potential) must be specified. Neumann boundary conditions consist of specifying the gradient of the dependent variable on the boundary, which for potential flow will represent the velocity normal to the domain. It is customary to use a coefficient, termed pressure coefficient, that is defined as follows.

$$C_P = \frac{p - p_\infty}{q_\infty}, \tag{2.5}$$

where p is the pressure at a given point, p_∞ is the pressure in the far field and q_∞ is given by:

$$q_\infty = \frac{1}{2}\varrho v^2.$$

An equivalent form of Eq. 2.5 is given by:

$$C_P = 1 - \frac{v}{v_\infty}^2, \tag{2.6}$$

where v is the velocity at a given point and v_∞ is the far field velocity. The pressure coefficient is equal one at the stagnation point where v=0 whilst is zero in the far field.

2.2 Finite Element Method

As noted, physical phenomena are typically governed by differential equations. While the derivation of the governing equation is straightforward, the solution of these equations may be extremely difficult, if not impossible, by classical means. This difficulty has lead to the development of many numerical techniques to solve these sets of equations.

The three most predominant numerical techniques are the finite difference, finite volume, and the finite element methods. The current work utilizes the finite element method, which discretize the problem domain into a finite number of smaller domains (called elements). An advantage of the finite element method over the other two possibilities is, for the given problem, the ease of coding. In the FEM, the variation of the dependent variable over each of the elements is assumed and appropriate shape or interpolation functions are derived. By enforcing continuity of the solution at the boundaries of the elements, algebraic relations among the undetermined coefficients may be obtained.

Modeling physical phenomena begins with the determination of a mathematical description of the problem at hand, for example the Navier-Stokes equations. In many cases these models are represented by partial differential equations. A partial differential equation for a given function $u(x, y, \cdots)$ defined over a domain Ω with boundary Γ may be expressed by the form

$$F(x, y, \cdots, u, u_x, \cdots, u_{xx}, u_{xy}, \cdots) = 0, \tag{2.7}$$

where F is a function of the independent variables $x, y, \cdots$,of the function u, and of a finite number of its partial derivatives. Typically the dependent variables represent physical properties of the models, for example energy, momentum, mass, etc. Assuming that these approximations are accurate and that the model equations represent the underlying physical processes, these functions will closely model the behavior of physical systems.

2.2.1 Finite Element Model for the Potential Equation

The first step in the finite element method is to obtain an approximate function that represents the solution within an element. This function must satisfy the following properties [17]:

1. The approximate solution should be continuous over the element, and differentiable, as required by the weak form.

2. It should be a complete polynomial, i.e., include all lower-order terms up to the highest order used.

3. It should be an interpolant of the primary variables at the nodes of the finite element.

Completeness requires that the element interpolation function is capable of exactly representing a linear polynomial.[14] These shape functions (usually identified by the notation N_i) usually have the property of convexity, described by

$$\sum_{i=1}^{m} N_i = 1. \tag{2.8}$$

Using these shape functions, the value of the function within an element is described by a linear combination of nodal values and the shape functions. For example, the function value u is approximated as

$$\tilde{u} = \sum_{i=1}^{m} u_i N_i. \tag{2.9}$$

Consider the Laplace equation:

$$-\frac{\partial}{\partial x}\frac{\partial u}{\partial x} - \frac{\partial}{\partial y}\frac{\partial v}{\partial y} - \frac{\partial}{\partial z}\frac{\partial w}{\partial z} = 0 \text{ in } \Omega. \tag{2.10}$$

In order to build the Finite Element Method (FEM) for the given equation, the first step is to multiply Eq. 2.10 with a weighting function v and then to integrate the resulting equation over a generic element domain Ω^e:

$$0 = \int_{\Omega^e} v\left[-\frac{\partial}{\partial x}\frac{\partial u}{\partial x} - \frac{\partial}{\partial y}\frac{\partial v}{\partial y} - \frac{\partial}{\partial z}\frac{\partial w}{\partial z}\right] dxdydz. \tag{2.11}$$

Function v is required to be at least once differentiable with respect to x and y. The next step is to distribute the differentiation between u and v. In order to do so, integration by parts of Eq. 2.11 is required. Given two functions F and v, recall that:

$$\frac{\partial(vF)}{\partial x_i} = v\frac{\partial F}{\partial x_i} + F\frac{\partial v}{\partial x_i}, \tag{2.12}$$

or:

$$-v\frac{\partial F}{\partial x_i} = F\frac{\partial v}{\partial x_i} - \frac{\partial(vF)}{\partial x_i}. \tag{2.13}$$

Moreover, according to the divergence theorem:

$$\int_{\Omega^e} \frac{\partial}{\partial x_i}(vF)dxdy = \int_{\Gamma^e} wFn_x ds, \tag{2.14}$$

where n_x represents the x component of the normal. The same relation holds for the y and z direction. If $F = \frac{\partial u}{\partial x_i}$, then by using Eq. 2.13 and Eq. 2.14, equation (2.11) leads to:

$$\begin{aligned} 0 = \int_{\Omega^e} \left[\frac{\partial v}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\frac{\partial v}{\partial y} + \frac{\partial v}{\partial z}\frac{\partial w}{\partial z} \right] dxdydz \\ - \int_{\Gamma^e} \left[w \left(n_x \left(\frac{\partial u}{\partial x} \right) + n_y \left(\frac{\partial u}{\partial y} \right) + n_z \left(\frac{\partial w}{\partial z} \right) \right) \right] ds. \end{aligned} \tag{2.15}$$

The quantity

$$q = n_x \left(\frac{\partial u}{\partial x} \right) + n_y \left(\frac{\partial u}{\partial y} \right) + n_z \left(\frac{\partial w}{\partial z} \right) \tag{2.16}$$

arising from Eq. 2.15 is called the *secondary variable.* More details will be given later. Assuming a finite element interpolation of the form:

$$u = \sum_{j=1}^{n} u_i N_j^e(x, y, z) \tag{2.17}$$

over the element Ω^e, substituting $v = N_i$ into 2.15 and performing some algebraic manipulation, the finite element equation is obtained:

$$[K^e]\{u^e\} = \{Q^e\}, \tag{2.18}$$

where the stiffness matrix $[K^e]$ is given by:

$$K_{ij}^e = \int_{\Omega^e} \left[\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z} \right] dxdydz, \tag{2.19}$$

whilst u^e is a vector of unknown and Q^e is the vector of secondary variables. The dependent variable of the problem is called the *primary variable.Secondary variables* are the coefficients of the interpolation function and its derivatives in the boundary expressions. Note that since the interpolation functions are linear, their derivatives will be constant, therefore the element is called Constant Strain Tetrahedra (actually this definition belongs to the mechanics of materials). In the 2D case, one must simply eliminate all the terms containing a z-coordinate and w, thus Eq. 2.15 becomes:

$$0 = \int_{\Omega^e} \left[-\frac{\partial v}{\partial x}\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\frac{\partial v}{\partial y} \right] dxdy - \int_{\Gamma^e} \left[w \left(n_x \left(\frac{\partial u}{\partial x} \right) + n_y \left(\frac{\partial v}{\partial y} \right) \right) \right] ds. \tag{2.20}$$

Therefore, Eq. 2.20, considered Eq. 2.19, assumes the form:

$$K_{ij}^e = \int_{\Omega^e} \left[\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \right] dxdy. \tag{2.21}$$

Moreover, Eq. 2.17 reduces to:

$$u = \sum_{j=1}^{n} u_i N_j^e(x, y). \tag{2.22}$$

In this case we obtain a constant strain triangle. Up to this point, integrations over each element have occurred, but an integration over the entire domain is necessary to obtain a solution. There are two possibilities for performing the integration over each element. The first, analytical integration, was used in the present work and consists in the analytical evaluation of Eq. 2.21. The second, numerical integration, requires the use of Gauss-Legendre quadrature. This method transforms the function from physical to natural coordinates, then integration is performed in natural coordinates. By doing so, evaluation of the Jacobian is required per each element. This method leads to inevitable numerical errors. Direct integration, whenever possible, leads to a better result. Due to the algebraic complexity of the equation arising from analytically integrating Eq. 2.21, analytical integration is only suitable for basic elements like bars, linear triangles, linear tetrahedra. For more complex elements, numerical integration is necessary. In order to assemble the global equations over the entire domain, the following conditions must be satisfied:

- Continuity of primary variables at the connecting nodes;
- Balance of secondary variables at the connecting nodes.

Continuity of primary variables ensures the uniqueness of the solution at a node. Balance of secondary variables guarantees the equilibrium of point sources at the node connecting several elements.

2.3 Element Library

In the construction of the element interpolation (shape) functions, the three aforementioned requirements of continuity, completeness and interpolation in section 2.2.1 must be satisfied. The current work utilizes triangular elements in two-dimension and tetrahedral elements in three-dimensions. These elements were chosen over other element types due to their inherent ability to discretize complex geometries easily and efficiently. An advantageous feature of the finite element method is its ability to increase the order of spatial accuracy by either reducing the characteristic length/size of the elements (called h-refinement) or increasing the order of the interpolation function (called p-refinement). Triangular and tetrahedral elements were used in the current work because are easily adaptively refined. Moreover, these are the simplest geometric entities possessing area and volume respectively (tetrahedron has, obviously, both). Linear interpolation functions were chosen for sake of simplicity. More details on convergence analysis will be given in the next chapters.

2.3.1 Triangular Element

The interpolation functions for a triangular element are easier to derive in a non dimensional area coordinate system. Given a general triangular element in Cartesian coordinates, shown in Fig. 2.1, area coordinates L_i defined based on the area ratios given by

$$L_1 = \frac{Area(P23)}{Area(123)} \tag{2.23}$$

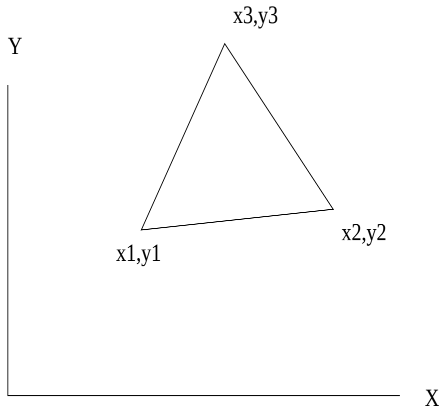

Figure 2.1: Triangular Element.

and illustrated in Fig. 2.2. The approximation of the geometry may be expressed similar to the approximation of the dependent variable as

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned} \tag{2.24}$$

The last expression stems from the constraint that the three sub-element areas must sum to the total area of the triangle. Additionally, this expression ensures that the property of convexity is satisfied. The three equations above may be solved for the areas coordinates, as

$$A = det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}; \tag{2.25}$$

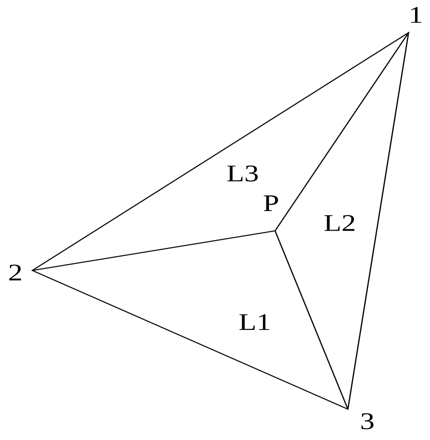

Figure 2.2: Triangular Element Area Coordinates.

which represents the area of the triangle and and the coefficients a,b,c may be expressed as

$$a_i = x_j y_k - x_k y_j, \tag{2.26}$$

$$b_i = y_j - y_k, \tag{2.27}$$

$$c_i = x_k - x_j. \tag{2.28}$$

In the above, i,j,k= 1,2,3, and cyclic permutation will uniquely specify all coefficients. The interpolation (shape) functions then become simply the area coordinates

$$N_1 = L_1; \qquad N_2 = L_2; \qquad N_3 = L_3. \tag{2.29}$$

It can be clearly seen that the interpolation functions have the following properties:

$$\begin{gathered} N_i^e(x_j^e, y_j^e) = \delta_{ij}; \qquad (i,j = 1,2,3); \\ \sum_{i=1}^{3} N_i^e = 1, \qquad \sum_{i=1}^{3} \frac{\partial N_i^e}{\partial x} = 0, \qquad \sum_{i=1}^{3} \frac{\partial N_i^e}{\partial y} = 0. \end{gathered} \tag{2.30}$$

which satisfies the aforementioned requirements for convergence. The element stiffness matrix for a linear triangular element, using the definition given in Eq. 2.25 and 2.30, may be obtained by differentiation to yield

$$\begin{gathered}\frac{\partial N_i^e}{\partial x} = \frac{b_i}{2A}, \\ \frac{\partial N_i^e}{\partial y} = \frac{c_i}{2A},\end{gathered} \tag{2.31}$$

where A is the element area computed from Eq. 2.25. On substitution of Eq. 2.31 into 2.21, the element stiffness matrix will assume the form:

$$K_{ij}^e = \frac{1}{2A}(b_i^e b_j^e + c_i^e c_j^e). \tag{2.32}$$

As seen, the element stiffness matrix for the linear triangular element is symmetric, and will be symmetric for any order triangle [17]. Moreover, the stiffness matrix is positive definite, but not necessarily diagonal dominant. The latter property may lead to stability problems in the solution process.

2.3.2 Tetrahedral Element

In a tetrahedral element, the state of displacement of a point is defined by three displacement components, u,v, and w, in the directions of the three coordinates x,y,z. This displacement field may be written as

$$\mathbf{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix}. \tag{2.33}$$

For a four node tetrahedral element, as shown in Fig. 2.3, a linear variation may be defined as

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z. \tag{2.34}$$

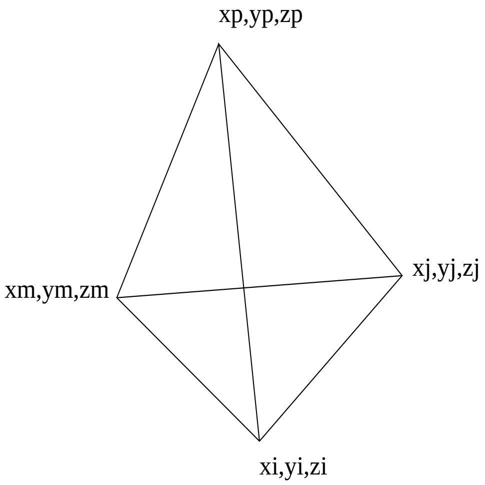

Figure 2.3: Tetrahedra Element.

By using the four conditions on displacement at the nodes, four equations of the following type may be written

$$u = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i + \alpha_4 z_i, \tag{2.35}$$

from which the coefficients α_i may be evaluated. Consider now a generic tetrahedron (see Fig. 2.3). It is possible to show [18] that the volume of the tetrahedral cell is given by:

$$6V \quad = det \begin{pmatrix} 1 & x_i & y_i & z_i \\ 1 & x_j & y_j & z_j \\ 1 & x_m & y_m & z_m \\ 1 & x_p & y_p & z_p \end{pmatrix}, \tag{2.36}$$

where V represents the tetrahedral volume. Moreover, it is possible to relate the displacements field over the tetrahedral element to the nodal displacements as

$$\begin{aligned} u = \frac{1}{6V}\Big[&(a_i + b_i x + c_i y + d_i z)u_i + (a_j + b_j x + c_j y + d_j z)u_j \\ &(a_m + b_m x + c_m y + d_m z)u_m + (a_p + b_p x + c_p y + d_p z)u_p\Big] \end{aligned} \tag{2.37}$$

where the coefficients a,b,c,d are:

$$a_i = det\begin{pmatrix} x_i & y_i & z_i \\ x_j & y_j & z_j \\ x_m & y_m & z_m \end{pmatrix}; \qquad b_i = -det\begin{pmatrix} 1 & y_i & z_i \\ 1 & y_j & z_j \\ 1 & y_m & z_m \end{pmatrix}; \tag{2.38}$$

$$c_i = det\begin{pmatrix} 1 & x_i & z_i \\ 1 & x_j & z_j \\ 1 & x_m & z_m \end{pmatrix}; \qquad d_i = -det\begin{pmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{pmatrix} \tag{2.39}$$

The coefficients for j,m and p are obtained by cyclic permutation. Similarly to the area coordinates defined for triangular elements, the above is equivalent to a volume system for tetrahedral elements.

The interpolation functions may be expressed in terms of the coefficients, as follows:

$$N_i \quad = \quad \frac{a_i + b_i x + c_i y + d_i z}{6V}. \tag{2.40}$$

Differentiating Eq. 2.40 with respect to the coordinate directions, and on substituti into Eq. 2.32 yields the element stiffness matrix for a linear tetrahedral element

$$K_{ij}^e = \frac{1}{6V}\begin{pmatrix} \frac{(b_1^2+c_1^2+d_1^2)}{6} & \frac{(b_1b_2+c_1c_2+d_1d_2)}{6} & \frac{(b_1b_3+c_1c_3+d_1d_3)}{6} & \frac{(b_1b_4+c_1c_4+d_1d_4)}{6} \\ \ldots & \frac{(b_2^2+c_2^2+d_2^2)}{6} & \frac{(b_2b_3+c_2c_3+d_2d_3)}{6} & \frac{(b_2b_4+c_2c_4+d_2d_4)}{6} \\ \ldots & \ldots & \frac{(b_3^2+c_3^2+d_3^2)}{6} & \frac{(b_3b_4+c_3c_4+d_3d_4)}{6} \\ \ldots & \ldots & \ldots & \frac{(b_4^2+c_4^2+d_4^2)}{6} \end{pmatrix} \tag{2.41}$$

Once again, the element matrix is symmetric.

CHAPTER III

SENSITIVITY ANALYSIS FOR THE FUNDAMENTAL EQUATIONS

3.1 Continuous and Discrete Approaches

In order to perform a sensitivity analysis, sensitivity derivatives are needed. Two approaches are possible to calculate them. The continuous approach calculates the derivatives directly, based upon the continuous governing equations, by using the method of material derivatives or generalized calculus of variations. In this method, the equations are differentiated *before* discretization.

The discrete approach takes analytical derivatives of the discretize equations with respect to shape variables, namely the co-ordinates of the grid points. In this method, differentiation is performed *after* discretization. [1]. The present study will utilize this procedure in order to evaluate sensitivities.

For discrete aerodynamic shape sensitivity analysis, the objective function and constraints may be expressed as:

$$F = F(\overline{u}, \overline{X}, \eta_k) \tag{3.1}$$

and

$$C_j = C_j(\overline{u}, \overline{X}, \eta_k) \tag{3.2}$$

Here $\overline{u}$ is the state vector on which the objective or constraint is defined. The equations that define $\overline{u}$ are called the state equations. At this point, one of two discrete formulations may be used to determine sensitivity derivatives; the direct differentiation method and the adjoint differentiation method [1]. The gradient of the function, ∇F, may be evaluated numerically. There are many tools available in order to evaluate sensitivity derivatives. One may use ADIFOR (or ADIC), but this means dealing with long and unreadable codes, or one may try to evaluate sensitivities analytically, but this is not always possible [1]. Another way is to evaluate derivatives numerically. A Finite-Difference (FD) approximation of the derivative yields

$$\frac{df}{dx} \approx \frac{f(x+h) - f(x-h)}{2h}. \tag{3.3}$$

This expression for the derivative has a truncation error of $O(h^2)$. The advantage of FD is that any existing code may be used without modification. The disadvantages are:

- computational time required;
- possible inaccuracy of the derivatives.

The second disadvantage is due to the truncation, cancellation and subtractive errors. The choice of the correct step size h is vital, though it is not known *a priori*. Moreover, it may change from point to point and/or from one design variable to the next one. One of the most effective means for evaluating sensitivity derivatives is the Complex Taylor Series Expansion (CTSE) [1]. In the CTSE method a series expansion is still performed, but using complex numbers as follows:

$$f(x+ih) = f(x) + ih\frac{df}{dx} - \frac{h^2}{2}\frac{d^2 f}{dx^2} - i\frac{h^3}{6}\frac{d^3 f}{dx^3} + \frac{h^4}{24}\frac{d^4 f}{dx^4} \tag{3.4}$$

Solving this equation for the imaginary part of the function yields

$$\frac{df}{dx} \approx \frac{Im[f(x+ih)]}{h} \tag{3.5}$$

This expression still has a truncation error $O(h^2)$. Thus, by using CTSE, both the function and its derivative are obtained, without subtractive error. The main advantages of CTSE are:

- very little modification of the original code is required;
- the derivative is not sensitive to the step size selection, and requires step sizes that avoid excessive truncation error, without regard to the cancellation error.

Therefore,CTSE was chosen in order to evaluate the grid sensitivity $\frac{\partial X}{\partial \eta}$, where $X(\eta_k)$ represents the grid.

3.1.1 Direct and Adjoint Methods

The direct differentiation method establishes sensitivity of state variables by directly differentiating the discretized matrix equation with respect to the design variables. Call F the function to be differentiated. Then:

$$\nabla F = \frac{\partial F}{\partial \overline{\eta}_k} + \frac{\partial F}{\partial \overline{X}} \frac{\partial X}{\partial \overline{\eta}_k} + \frac{\partial F}{\partial \overline{u}} \frac{\partial \overline{u}}{\partial \overline{\eta}_k} \tag{3.6}$$

The sensitivity of the state vector is obtained from differentiating the state equation $\overline{w}(\overline{X}, \overline{u}(\overline{X})) = 0$:

$$\frac{\partial \overline{w}}{\partial \overline{\eta}_k} + \frac{\partial \overline{w}}{\partial \overline{X}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} + \frac{\partial \overline{w}}{\partial \overline{u}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} = 0, \tag{3.7}$$

or:

$$\frac{\partial \overline{w}}{\partial \overline{u}} \frac{\partial \overline{u}}{\partial \overline{\eta}_k} = -\frac{\partial \overline{w}}{\partial \overline{X}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} - \frac{\partial \overline{w}}{\partial \overline{X}}. \tag{3.8}$$

This equation needs to be solved for $\frac{\partial \overline{u}}{\partial \overline{\eta}_k}$.

The adjoint variable formulation of the sensitivity derivatives makes use of an adjoint variable, in order to evaluate the sensitivities. The mathematical formulation of the problem can be easily derived form Eq. 3.6 as follows. First augment $F(\overline{\eta_k})$:

$$F(\overline{\eta_k}) = F(\overline{\eta_k}) + \lambda^T \overline{w}, \tag{3.9}$$

where the adjoint vector λ remains to be defined. On differentiation, Eq. 3.9 yields

$$\nabla F = \frac{\partial F}{\partial \overline{\eta}_k} + \frac{\partial F}{\partial \overline{X}} \frac{\partial X}{\partial \overline{\eta}_k} + \frac{\partial F}{\partial \overline{u}} \frac{\partial \overline{u}}{\partial \overline{\eta}_k} + \lambda^T \left[\frac{\partial \overline{w}}{\partial \overline{\eta}_k} + \frac{\partial \overline{w}}{\partial \overline{X}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} + \frac{\partial \overline{w}}{\partial \overline{u}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} \right], \tag{3.10}$$

or, rearranging:

$$\nabla F = \frac{\partial F}{\partial \overline{\eta}_k} + \frac{\partial F}{\partial \overline{X}} \frac{\partial X}{\partial \overline{\eta}_k} + \lambda^T \frac{\partial \overline{w}}{\partial \overline{\eta}_k} + \lambda^T \frac{\partial \overline{w}}{\partial \overline{X}} \frac{\partial \overline{X}}{\partial \overline{\eta}_k} + \frac{\partial \overline{w}}{\partial \overline{X}} + \left[\lambda^T \frac{\partial \overline{w}}{\partial \overline{u}} + \frac{\partial F}{\partial \overline{u}} \right] \frac{\partial \overline{u}}{\partial \overline{X}}. \tag{3.11}$$

In order to eliminate the dependency of the gradient on $\frac{\partial \overline{u}}{\partial \overline{X}}$, it is useful to define the adjoint vector as:

$$\lambda^T \frac{\partial \overline{w}}{\partial \overline{u}} + \frac{\partial F}{\partial \overline{u}} = 0; \tag{3.12}$$

Therefore:

$$\left[\frac{\partial \overline{w}}{\partial \overline{u}} \right]^T \lambda = -\frac{\partial F}{\partial \overline{u}} \tag{3.13}$$

Note that all the derived sensitivity equations are linear in terms of state variable derivatives or adjoint variables. This makes the problem easier to handle. The equations related to the direct method must be solved for each of the independent shape variables, whereas those derived for the adjoint method need to be solved for each of the functions whose sensitivities are sought. Suppose there are M design variables and N functions: by using the direct approach, one has to solve Eq. 3.6 M times, whilst, by using the adjoint one, one needs to solve Eq. 3.13 N times. Therefore, the best choice will be based on the number of function evaluations needed to obtain the sensitivity sought. In this work, since there are only a few design variables (the Bezier-Bernstein control points), it is apparent that the adjoint method works better. However, the code developed makes use of the direct method. Since, as aforementioned, many physical phenomena are described by Poisson's equation, the code can be used to handle this class of problems. By using the adjoint method it is necessary to modify the code in order to analyze different phenomena. By using the direct approach, this inconvenience is bypassed.

3.2 Mathematical Model

3.2.1 Linearization of Finite Element Method

Recall that the Laplace's equation, in terms of FEM method, is recast in the following form:

$$[K]\{\Phi\} = \{Q\} \tag{3.14}$$

In order to get the derivative of the potential Φ, one needs to evaluate the derivative of eq. 3.14 with respect to η. This leads to:

$$\left[\frac{\partial K}{\partial \eta}\right]\{\Phi\} + [K]\frac{\partial \Phi}{\partial \eta} = \{\frac{\partial Q}{\partial \eta}\}. \tag{3.15}$$

Equation 3.15, since in the present case the term $\{\frac{\partial Q}{\partial \eta}\}$ is always zero, can be rearranged in the form:

$$\left[\frac{\partial K}{\partial \eta}\right]\{\Phi\} = -[K]\frac{\partial \Phi}{\partial \eta} \tag{3.16}$$

This system needs to be solved in order to evaluate the sensitivity of the equations with respect to the set of chosen design variables.

3.2.2 Sensitivity of the Stiffness Matrix

Consider Eq. 3.14. Given a design variable η, sensitivity of the stiffness matrix with respect to η can be computed in the following way. Consider the 2D case.

$$K_{ij} = \frac{1}{4Area}(\beta_i\beta_j - \gamma_i\gamma_j) \tag{3.17}$$

where:

$$\begin{aligned} \beta_i &= y(j) - y(k) \\ \gamma_i &= x(k) - x(j) \end{aligned} \tag{3.18}$$

and i,j,k are cyclic indexes.Moreover:

$$Area = \frac{1}{2}[J] \tag{3.19}$$

where J is the Jacobian whose expression is given by:

$$[J] = [x(i) - x(k)][y(j) - y(k)] - [x(j) - x(k)][y(x) - y(k)] \tag{3.20}$$

Now, by setting:

$$t = \beta_i\beta_j - \gamma_i\gamma_j \tag{3.21}$$

it is possible to rewrite K as

$$K_{ij} = \frac{t}{2J}. \tag{3.22}$$

Therefore, the derivative of K with respect to β is given by:

$$\frac{\partial K_{ij}}{\partial \eta} = \frac{1}{2}\frac{\frac{\partial t}{\partial \eta}J - t\frac{\partial J}{\partial \eta}}{J^2} \tag{3.23}$$

Then:

$$\frac{\partial t}{\partial \eta} = \frac{\partial}{\partial \eta}[\beta_i\beta_j - \gamma_i\gamma_j] \tag{3.24}$$

or:

$$\frac{\partial t}{\partial \eta} = \left[\frac{\partial \beta_i}{\partial \eta}\beta_j + \frac{\partial \beta_j}{\partial \eta}\beta_i - \frac{\partial \gamma_i}{\partial \eta}\gamma_j - \frac{\partial \gamma_j}{\partial \eta}\gamma_i\right]. \tag{3.25}$$

In eq. 3.25 the derivatives with respect to β are given by:

$$\frac{\partial \beta_i}{\partial \eta} = \frac{\partial y_j}{\partial \eta} - \frac{\partial y_k}{\partial \eta} \tag{3.26}$$

$$\frac{\partial \gamma_i}{\partial \eta} = \frac{\partial x_k}{\partial \eta} - \frac{\partial x_j}{\partial \eta} \tag{3.27}$$

Again, the indexes i,j,k are cyclic in nature. Concerning the Jacobian, Eq. 3.20 yields:

$$\begin{aligned}\frac{\partial J}{\partial \eta} = &\left(\frac{\partial x_i}{\partial \eta} - \frac{\partial x_k}{\partial \eta}\right)(y_j - y_k) + (x_i - x_k)\left(\frac{\partial y_j}{\partial \eta} - \frac{\partial y_k}{\partial \eta}\right) - \\ &\left(\frac{\partial x_j}{\partial \eta} - \frac{\partial x_k}{\partial \eta}\right)(y_i - y_k) + (x_j - x_k)\left(\frac{\partial y_i}{\partial \eta} - \frac{\partial y_k}{\partial \eta}\right)\end{aligned} \tag{3.28}$$

By substituting 3.28, 3.25, 3.20 and 3.21 into Eq. 3.23, the final expression for the derivative of the stiffness matrix is obtained.

CHAPTER IV

ITERATIVE SOLUTION METHODS.

The system of equations resulting from a finite element discretization of Poisson's equation is symmetric and positive-definite. The symmetric, positive-definiteness of the system affords the use of direct solution methods, such as Cholesky decomposition. These direct methods are very memory intensive and not amenable for large-scale three-dimensional problems. However, for analyses that have a large number of right-hand-side vectors, direct methods may offer significant CPU time savings; provided the stiffness matrix may be assumed to be constant.

In the current work, only one right-hand-side will be associated with a given analysis, however, the sensitivity analysis will have numerous right-hand-side vectors. From one design iteration to the next the stiffness matrix, due to the sensitivity analysis operated on the matrix itself, will change and, thus, a new factorization would be required for direct methods resulting in a considerable loss of time. For this reason, and for the intended use in large-scale three dimensional problems, the solution to the resulting system of equations will be performed with iterative methods. The iterative solution methods utilized for this study are Jacobi point iteration, Gauss-Seidel (GS), and preconditioned conjugate gradient.

The following system of equations will be assumed in discussing the various solution methods:

$$[K]\{\Phi\} = \{Q\}, \tag{4.1}$$

4.1 Jacobi Point Iterative Solver

Jacobi point iteration and GS iteration only differ in the means by which the solution vector is updated. Jacobi iteration updates the solution vector using the values of this vector at the previous iteration. Jacobi point iteration solver can be seen as a variation of GS. If the unknown on the right hand side of Eq. 4.3 are updated only after each iteration through the entire field, *Jacobi iteration* is obtained.

Therefore, the point Jacobi iteration update formula for Eq. 4.1 will be:

$$\Phi_i^{k+1} = {K_{ii}}^{-1}\left(Q_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} K_{ij}\Phi_j^k \right), \tag{4.2}$$

where k represents the iteration number.

The system of equations studied in the current work are positive-definite, but this does not guarantee diagonal dominance. Lack of diagonal dominance may cause this method to be slow to converge or even diverge. This fact is evident in the two-dimensional test cases used to verify the developed software. Due to the lack of diagonal dominance, Jacobi point iteration solver has showed instability, halting after 14000 iterations.

4.2 Gauss-Seidel Solver

Gauss-Seidel is one of the most efficient point iterative solution methods. The update for each solution vector always uses the most recent value of this vector. Additionally, a relaxation factor may be used to over-relax the solution and thereby accelerate the convergence of an already convergent system or under-relax the system to make a non-convergent system converge. When the system lacks diagonal dominance, under relaxation is required. The Gauss-Seidel iteration update formula for Eq. 4.1 may be written as:

$$\Phi_i^{new} = K_{ii}{}^{-1}\left(Q_i - \sum_{j=1}^{i-1} K_{ij}\Phi_j^{k+1} - \sum_{j=i+1}^{n} K_{ij}\Phi_j^{k}\right) \tag{4.3}$$

$$\Phi_i^{k+1} = \omega\Phi_i^{new} + (1-\omega)\Phi_i^{old} \tag{4.4}$$

where ω is the relaxation factor. Values of $0 \leq \omega \leq 1$ represent under-relaxation, and $1 \leq \omega \leq 2$ represents over-relaxation.

The iteration is continued until the change in the change in the solution vector falls below a pre-specified convergence tolerance ϵ as:

$$\frac{\|\Phi^{(k+1)} - \Phi^{(k)}\|}{\|\Phi^{(k+1)}\|} < \epsilon. \tag{4.5}$$

Here $\|\cdot\|$ means the Euclidean norm of a quantity. The required number of iteration to converge the system to the prescribed tolerance is dependent on the initial guess and the condition number of the coefficient matrix. To converge the system of equations for a non-lifting airfoil required over 27900 iterations, with an under-relaxation factor of 0.7. Since the intended use of the current software is to demonstrate design optimization, a more efficient method must be selected. It can be seen that the Gauss-Seidel method offers improvement of the point Jacobi method. However, this improvement was not deemed adequate for design optimization studies because the optimization process is an iterative method by itself. If, say, g iterations are required for the optimization process and k are required by the GS solver, the number of computation required in order to achieve convergence will be gk. It is apparent that, for large three dimensional geometries, this number will be excessively high.

4.3 Preconditioned Conjugate Gradient Solver

The Conjugate Gradient method seeks the solution to Eq. 4.1 by minimizing the potential function:

$$\Pi = \frac{1}{2}\Phi^T K\Phi - \Phi^T Q. \tag{4.6}$$

The task is to find Φ^{k+1} such that: $\Pi^{k+1} < \Pi^k$. In order to obtain this, a vector p^1, p^2, $\ldots$, p^s is used to calculate the potential. The algorithm [20]is stated below.

- Choose a starting vector $\Phi^{(0)}$
- Calculate the residual $r^{(1)}$ = Q - K$\Phi^{(0)}$. If $r^{(1)}$=0, quit.
- $p^{(1)}$=$r^{(1)}$
- DO WHILE convergence is reached:

$$\begin{aligned}
\alpha_k &= \frac{r^{(k)^T} r^{(k)}}{p^{(k)^T} K p^{(k)}} \\
\Phi^{(k+1)} &= \Phi^{(k)} + \alpha_k p^{(k)} \\
r^{(k+1)} &= r^{(k)} - \alpha_k K p^{(k)} \\
\beta_k &= \frac{r^{(k+1)^T} r^{(k+1)}}{r^{(k)^T}} \\
p^{(k+1)} &= r^{(k+1)} + \beta_k p^{(k)}.
\end{aligned} \tag{4.7}$$

Iterations will be performed until convergence tolerance on $r^{(k+1)}$ is reached. It is possible to show that the convergence to the solution Φ can be reached in a finite number of iterations. This means that if there exist a solution to the system of equations, it will be reached. The problem is that the number of iterations required, N, may be large. The number of iterations depends on the condition number of the matrix K. The condition number is defined as:

$$cond(K) = \left| \frac{\lambda_n}{\lambda_1} \right|, \tag{4.8}$$

where λ_1 is the smallest eigenvalue of K, λ_n the largest. The larger the condition number, the slower the convergence, so it is necessary to accelerate the solution process. One possibility is given by the use of pre-conditioning. The basic idea of the pre-conditioning is that instead of solving Eq. 4.1,the following is solved:

$$\overline{K}\overline{\Phi} = \overline{Q}, \tag{4.9}$$

where:

$$\begin{aligned} \overline{\Phi} &= C_Q \Phi, \\ \overline{Q} &= C_L^{-1} Q. \end{aligned} \tag{4.10}$$

The non singular matrix $K_P = C_L C_Q$ is called the pre-conditioner. The matrix $\overline{K} = [K]{K_P}^{-1}$ can have a much improved condition number [20]. The present work makes use of a Jacobi pre-conditioner, which consists in the inverse the stiffness matrix diagonal. Figure Fig. 4.3 shows the tremendous improvement in terms of speed obtained by using apreconditioned conjugate gradient (PCG) solver instead of a Gauss-Seidel. The PCG solver took only 161 iterations to solve the same problem where Gauss-Seidel required 29700 iterations. The time required to solve the same problem was, respectively, 4 minutes for the Jacobi solver and 0.5 sec for the PCG solver.

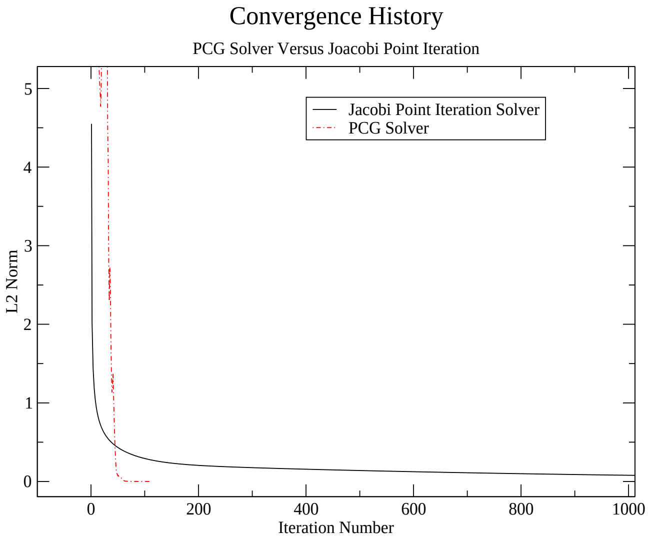

Figure 4.1: PCG Solver Versus Gauss-Seidel Solver Convergence History for a NACA 0012 non Lifting Airfoil.

CHAPTER V

SHAPE DESIGN OPTIMIZATION

5.1 Design Optimization Problem Formulation

The purpose of computational design optimization is to aid the designer in rationally searching for the best design out of many possible. The main goal of the present work is to investigate the shape sensitivities of aerodynamics quantities. Usually it is possible to recast a problem in terms of a number of functions to be optimized, and a number of constraints that act on these functions. The gradients of the objective functions are called *sensitivity derivatives*. If:

- F is the function to be optimized;
- $\overline{X}$ is the computational mesh over which the PDE is discretized;
- η_k is the vector of design variables

then, the optimization problem is of the form

$$\begin{array}{ll} \text{minimize:} & F(\overline{X}, \eta_i) \\ \text{Subject to:} & C_j(\overline{X}, \eta_i) \leq 0 \\ \quad\text{where:} & \eta_i^L \leq \eta_i \leq \eta_i^U \end{array}$$

where $C_j(X)$ is the inequality constraint, η_i^L represents the lower bound on each variable, whilst η_i^U is the upper one.

The goal of this work is to analyze the sensitivities of aerodynamic quantities. Clearly these quantities (e.g., Φ, C_P) are implicit functions of many other quantities. The source of implicit dependency is called the *state vector*, which is defined by the *state equations*.

5.2 Geometry and Grid Representation

5.2.1 Bezier-Bernstein Surface Parameterization

There are many available methods to represent a given geometry. As aforementioned, *Bezier-Bernstein* curves were used. Any point on a Bezier curve segment may be expressed by a parametric function:

$$c(u) = \sum_{i=0}^{N} b_i B_{i,N}(u) \qquad u \in [0,1]. \tag{5.1}$$

Here b_i represents the N+1 vertices called Bezier control points, and the blended functions $B_{i,N}(u)$ are given by the N^{th}-degree Bernstein polynomials:

$$B_{i,N}(u) = \frac{N!u^i(1-u)^{N-i}}{i!(N-i)!}. \tag{5.2}$$

Here u is the normalized computational arclength along the curve. Bezier-Bernstein curves satisfy the following properties:

- the curve must pass through the first and last control point;
- the tangent to the curve at each point may be controlled or specified if necessary.

The design variables are the NDV control points, which permits parameterization of the airfoil surface. Figure 5.1 illustrates an example of Bezier-Bernstein surface parameterization.

5.2.2 Mesh Movement Strategy

In order to move the mesh to the changing design surface, the method developed by Batina [2] was used. In this approach, the grid is considered as a system of interconnecting springs. By doing so, each grid element edge is represented as a tension spring. The stiffness of this string is assumed to be inversely proportional to the length

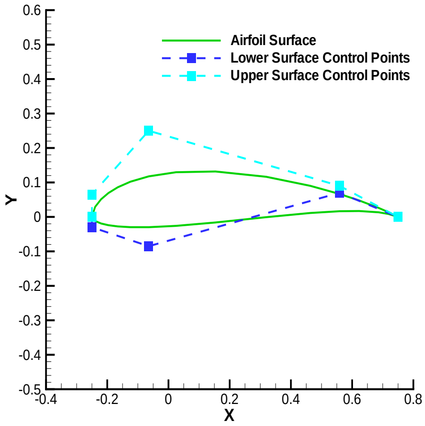

Figure 5.1: Bezier-Bernstein surface parameterization.

of its edge and may be written as:

$$s_{ij} = \frac{1}{[(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{p}{2}}}, \tag{5.3}$$

where p is the parameter used to control the stiffness of the spring. Obviously, in the 3D case, the term $(z_i - z_j)^2$ must be added to 5.3. The resulting set of linear equations are solved for the displacements of each node using a Jacobi iteration:

$$\Delta X_j^{n+1} = \frac{\sum_i s_{ij} \Delta X_i^n}{\sum_i s_{ij}}, \tag{5.4}$$

where i is summed over all edges connected to node j. The positions of interior nodes are then updated using the determined displacement. The advantage of the iterative method is that it does not require a large amount of memory.

5.2.3 Grid Sensitivity

Grid sensitivity to the design variables must be evaluated to perform shape sensitivity analysis. In the current work, grid sensitivity is evaluated via the Complex Taylor Series Expansion method [1].

5.3 Optimization Techniques

The optimization process is made up of four parts. First, FEM solver iteratively solves for the potential flow in the entire flow field. Second, once the potential Φ and the pressure coefficient are known, evaluation of their derivatives is performed by direct-discrete sensitivity analysis: $\frac{\partial \Phi}{\partial \eta}$, $\frac{\partial C_P}{\partial \eta}$. The third step is to run the optimizer. This subroutine, given a number of design variables and the sensitivities of C_P and Φ, will evaluate the updated design variable. As fourth step, a Bezier-Bernstein curve is used to generate, the updated airfoil shape at each design cycle by simply moving the control points. Note that only a few points are required in order to generate the airfoil shape. Once the new grid has been generated, a new iteration will be performed until convergence is reached. The result will be the optimized airfoil shape. Typically, there are two different way to approach an airfoil design:

- Maximization or Minimization of a specified aerodynamic quantity;
- Matching a pre specified aerodynamic quantity (inverse design).

The code developed in this work allows to perform the inverse design. In order to validate the sensitivities of C_P, a perturbation study was carried out by perturbing one node and comparing the fD sensitivity against those obtained by the sensitivity analysis. This validation is presented in a subsequent section.

5.3.1 Levemberg-Marquardt for Least-Squares

In a large number of problems, the function to be minimized is a sum of squares of non linear functions:

$$f(x) = \frac{1}{2}\sum_{i=1}^{m} f_i(x)^2 \tag{5.5}$$

This problem is better known as the *least-squares* problem. Equation 5.5 can be minimized by using any of the general unconstrained methods [15], but in most circumstances it is worthwhile to use methods specifically designed for the least-squares problems. Suppose one wants to minimize *f(x).* given m functions and n design vairables, denote the Jacobian of the function as J(x) the $m\ x\ n$, and let $G_i(x)$ be the Hessian matrix of $f_i(x)$. Then

$$G(x) = J(x)^T f(x), \tag{5.6}$$

$$G(x) = J(x)^T J(x) + Q(x), \tag{5.7}$$

where Q(x)=$\sum_{i=1}^{m} f_i(x) G_i(x)$. from 5.7 it is evident to note that the Hessian consists of two elements. One is of first order, the other is of second order.

The first order term $(J(x)^T J(x))$ will dominate the second order $(Q(x))$ in most cases (e.g., when residual are very small). This is the basic assumption of all least-squares methods. Let x_k denote the current estimate of the solution; a quantity subscripted by k will denote that quantity evaluated at x_k. The *Levenberg-Marquardt* method search direction is defined as the solution of the equations

$$(J_k^T J_k + \lambda_k I) p_k = -J_k^T f_k \tag{5.8}$$

where λ_k is a non-negative scalar. As regards to p_k, it is possible to show [15] that, for some scalar Δ related to λ_k, it is a solution of the constrained subproblem

$$\text{minimize} \qquad \frac{1}{2} \| J_k p + f_k \| \tag{5.9}$$

$$\text{subject to} \qquad \|p\| \leq \Delta.$$

A "good" value of λ_k or Δ must be chosen in order to ensure convergence. There are two limiting cases:

- λ_k is zero, then p_k is the Gauss-Newton direction;

- $\lambda_k \to \infty, \|p_k\| \to 0$, then the method degenerates into the steepest descent method.

Furthermore, for the Gauss-Newton or Newton's method, the design variable is updated as

$$\eta^{k+1} = \eta^k + p_k \tag{5.10}$$

It may be seen that the optimal step length is unity, and the one-dimensional search is not required. However, for the Levenberg-Marquardt method this is only the case when λ_k is identically zero, otherwise a one-dimensional search should be performed. Due to the inexpensive cost of the analysis and sensitivity analysis, the step length is chosen as unity. This results in the need for more design cycles to reach the optimum, but in a greatly simplified algorithm.

CHAPTER VI

RESULTS AND DISCUSSION

6.1 Validation of Potential Solver

The potential flow code must be validated. In order to validate the code, a comparison between the numerical results obtained with the analytical solution for a Joukowsky airfoil was made. Then, a convergence analysis was carried out in order to make sure that the code would achieve the correct result as the grid size $h \to 0$. In the FEM method it is possible to show [17] that the error is given by:

$$e = ch^p, \tag{6.1}$$

where c is a constant, h is the grid size and p is the order of the interpolation function. So, in order to achieve the best possible value, either p can be increased or h decreased. In this study, the order of p was kept constant.

The validation test case selected for presentation was the Joukowsky airfoil. The grid refinement study used three different meshes in order to validate the code. As the mesh is refined, the result converges to the analytical solution. The first test (Fig. 6.1) is a coarse grid made up of 3760 elements and 1942 nodes: 250 iterations were required for convergence. Figure 6.2 illustrates the C_P distribution around the airfoil. The second grid shown in Fig.6.3 was made up of 5826 elements and 3000 nodes. In order to achieve convergence, 323 iterations were required. Figure 6.4 shows C_P distribution around the airfoil. The final mesh, shown in figure 6.5 contains 9156 elements and 4715 nodes. It took 420 iterations in order to reach a convergent solution. Again, the following Fig.

6.6 represents C_P contours past the airfoil. Figure 6.10 represents the analytical result. Next there follows a comparison between the analitycal C_P for the Joukowsky airfoil and the computational. Figure 6.7, 6.8 and 6.9 represent the initial, a refined and the finer grid results respectively, while Fig. 6.10 reports the analytical C_P distribution over the airfoil. It can be seen that resultsd are identical.

6.2 Validation of Potential Sensitivity Analysis Solver

Next, the potential sensitivity analysis solver was validated. In order to achieve this task, a perturbation study was performed to verify that the optimal step size was found for the finite-difference sensitivity derivative. A NACA 0012 grid was used for the test cases. Figure 6.11 illustrates the region near the airfoil. The entire grid was made up of 898 elements and 478 nodes. As first, validation of the code was performed. Then, the results obtained from the direct-discrete solver were compared with those obtained from the finite-difference in order to validate the code. Figure 6.12 shows the values at one node (node 21) on the airfoil surface. It is possible to see that analytical and FD results shows good agreement when the perturbation h used for the FD code is in the range: $(10^{-4}, 10^{-10})$. If h is less than 10^{-4}, the FD scheme doesn't reach sufficient accuracy mainly due to the truncation error. If h is greater than 10^{-10}, the effects of cancellation error become relevant and, as one can tell from Fig. 6.12, the FD solution becomes unstable and hence inaccurate. The comparison between the analytical and the FD Φ sensitivity is showed in Fig. 6.13 and 6.14. Figures Fig. 6.17 and Fig. 6.18 show the comparison between FD and analytical values for C_P sensitivity.

6.3 Inverse Design Optimization Studies

Inverse Design Optimization was carried out on a NACA 0012 airfoil and a Multi-Element airfoil.

6.3.1 Target C_P Inverse Design: NACA 0012

The first picture (Fig. 6.19) represents the initial NACA 0012 airfoil C_P distribution while Fig. 6.20 depicts the same quantity for the final airfoil, after 450 design cycles. Figures Fig. 6.21 and Fig. 6.22 show the sensitivity of Φ and C_P for the final airfoil. The initial geometry shows a symmetric pressure distribution around the airfoil whilst, after 450 iterations, the pressure distribution has been substantially modified, as one can from figure Fig. 6.20. Table Tab. 6.1 reports the convergence history for all the 6 design variables. Figure 6.25 depicts the target C_P distribution versus computed C_P for the (deformed) NACA 0012 airfoil. Figures Fig. 6.26 and Fig. 6.27 represent how C_P changes during the optimization process both for the upper and lower airfoil surface. Figures 6.21, 6.22, 6.23 and Fig. 6.24 report respectively the optimization history for two design variables located on the upper surface and two design variables on the lower one. Figure 6.28 illustrates the deformed mesh and Fig. 6.29 shows the change in the airfoil shape at 1, 50, 100, 300 and 450 design iterations.

6.3.2 Target C_P Inverse Design: Multi-element Airfoil

Here the inverse design analysis is carried out for a multi-element airfoil (MEA). Note that the airfoil is made up of 4 elements: slat, main wing, vane and flap, but the inverse design process only affects the main wing. Figure Fig. 6.30 represents the initial mesh while Fig. Fig. 6.31 is the final deformed airfoil. The initial and final C_P distribution is reported in Fig. 6.32 and Fig. 6.33 respectively. Figures Fig. 6.34 and Fig. 6.35 illustrate the sensitivity of C_P and Φ for the MEA airfoil. Figures Fig. 6.36, Fig. 6.37, Fig. 6.38, Fig. 6.39, show the convergence history for Design Variables 1,3,4,6 respectively. At last, figure Fig. 6.40 reports the airfoil shapes at 1, 500, 1000, 2000 and 2500 iterations.

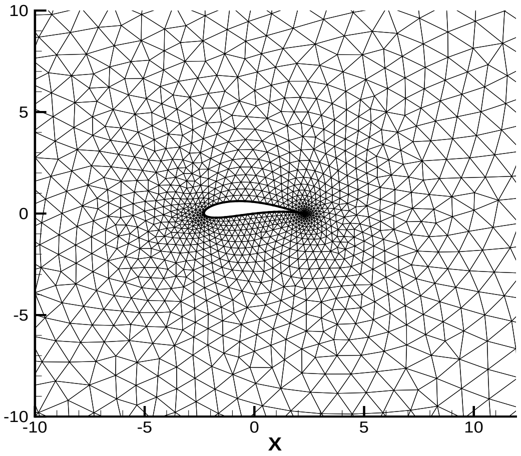

Figure 6.1: Joukowsky Airfoil, Coarse Grid.

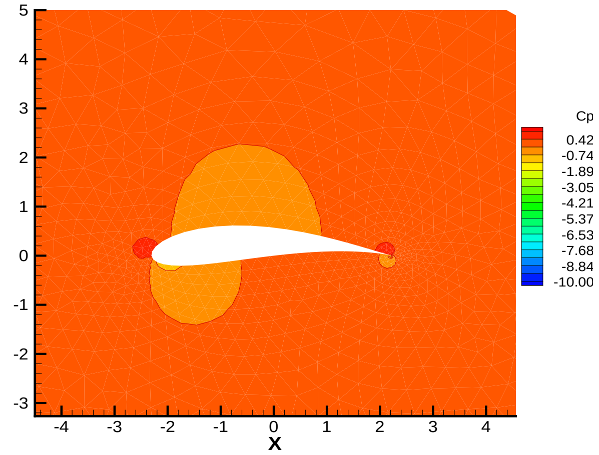

Figure 6.2: Joukowsky Airfoil, C_P Distribution on Coarse Grid.

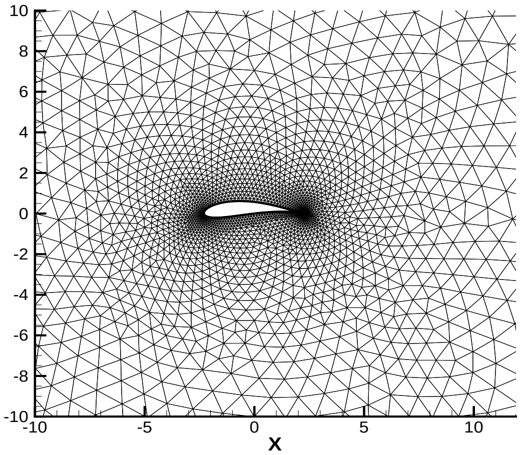

Figure 6.3: Joukowsky Airfoil, Refined Grid.

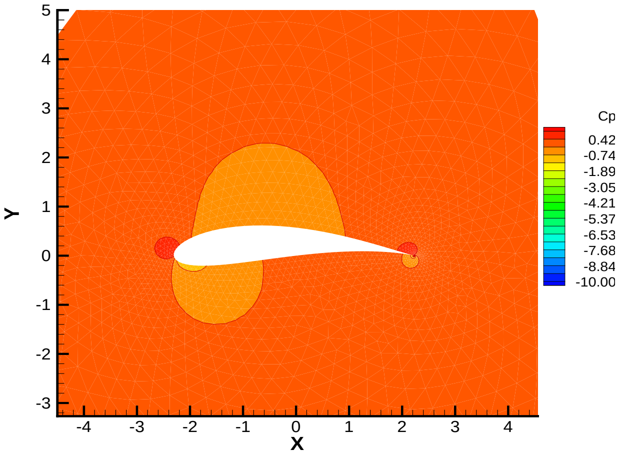

Figure 6.4: Joukowsky Airfoil, C_P Distribution on Refined Grid.

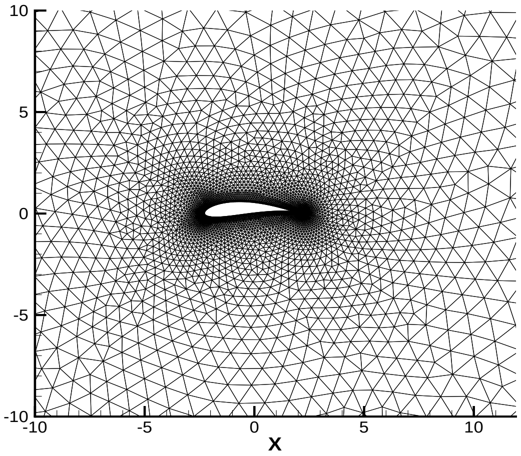

Figure 6.5: Joukowsky Airfoil, Fine Grid.

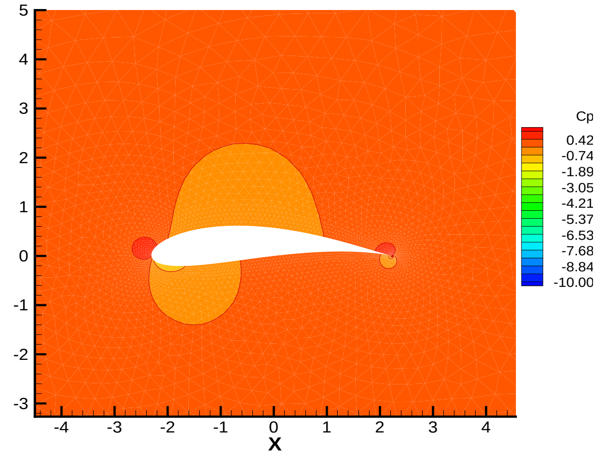

Figure 6.6: Joukowsky Airfoil C_P Distribution on Fine Grid.

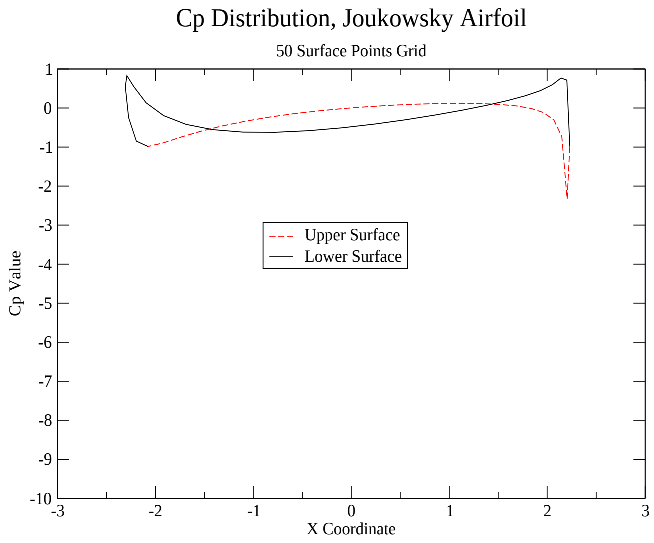

Figure 6.7: Joukowsky Airfoil, C_P on Coarse Grid.

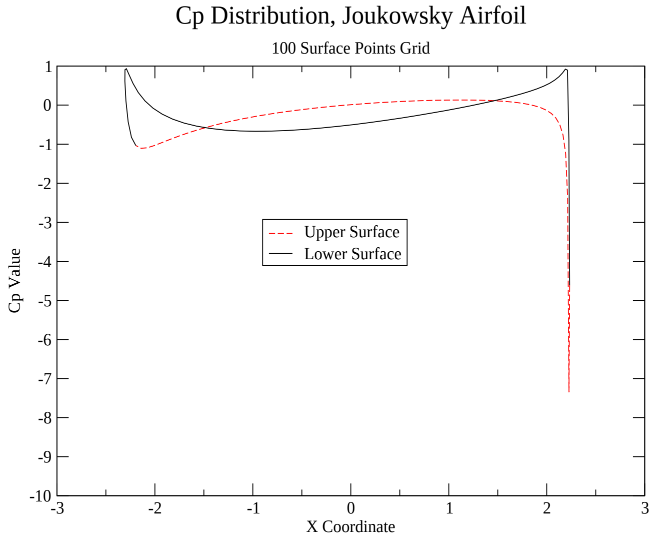

Figure 6.8: Joukowsky Airfoil, C_P Distribution on Refined Grid.

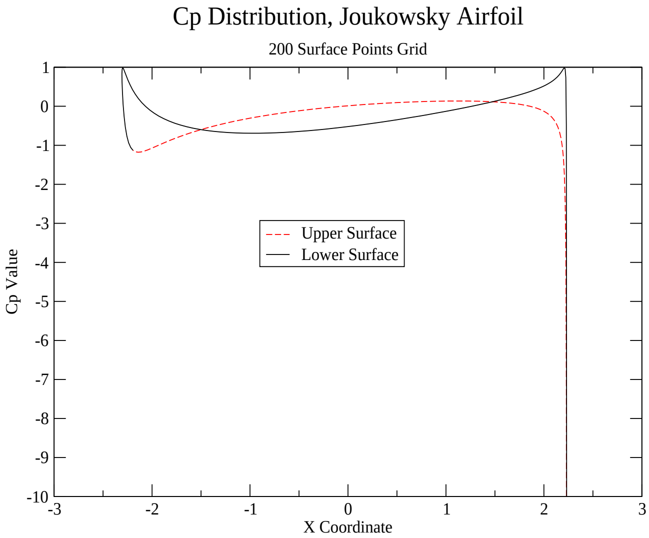

Figure 6.9: Joukowsky Airfoil, C_P on Distribution on Fine Grid.

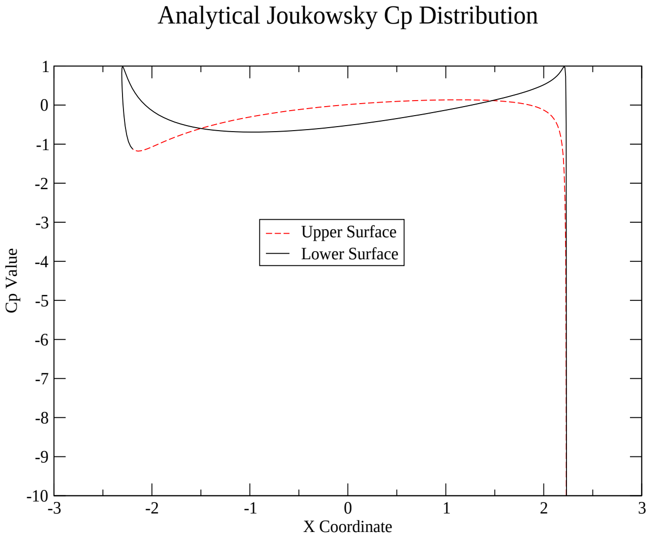

Figure 6.10: Joukowsky Airfoil, Analytical C_P Distribution.

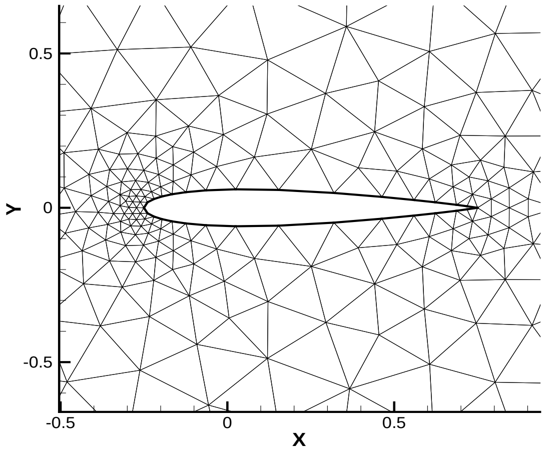

Figure 6.11: NACA 0012 Grid Close-Up.

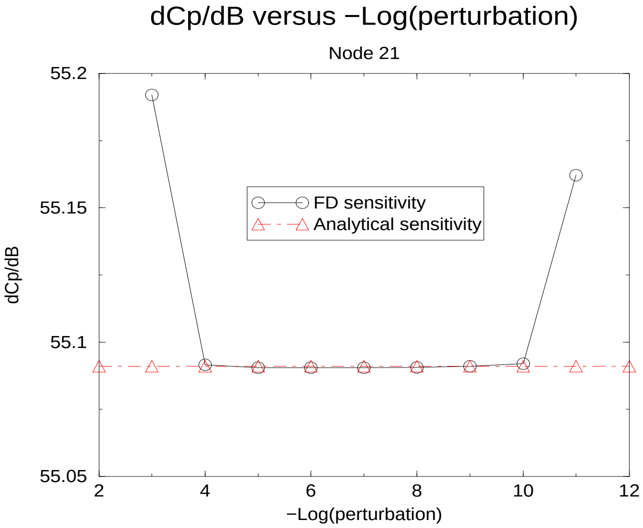

Figure 6.12: NACA 0012 Analytical Versus Finite Difference Sensitivities Comparison.

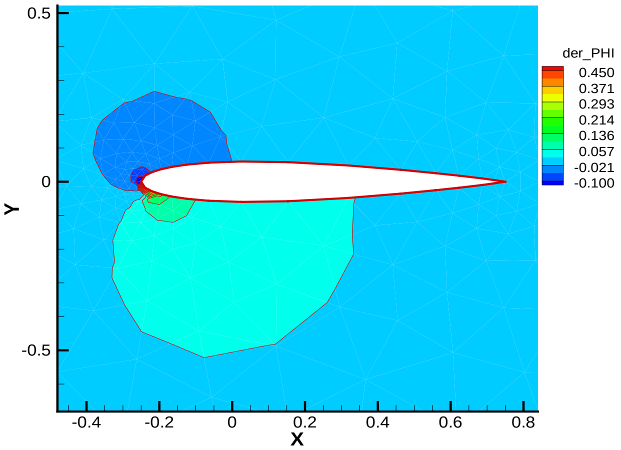

Figure 6.13: NACA 0012 FD Sensitivity of Φ at Node 21.

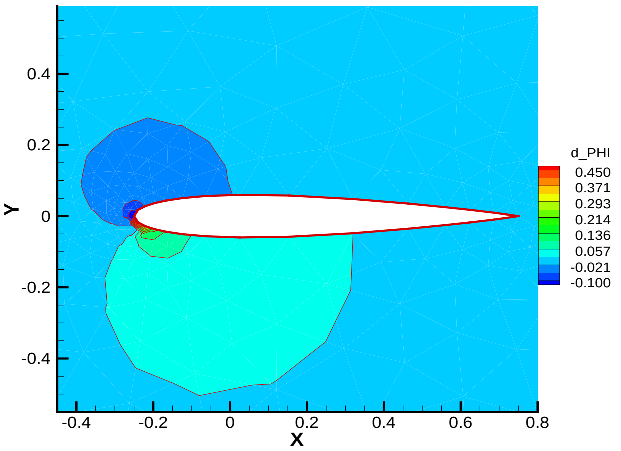

Figure 6.14: NACA 0012 Analytical Sensitivity of Φ at Node 21.

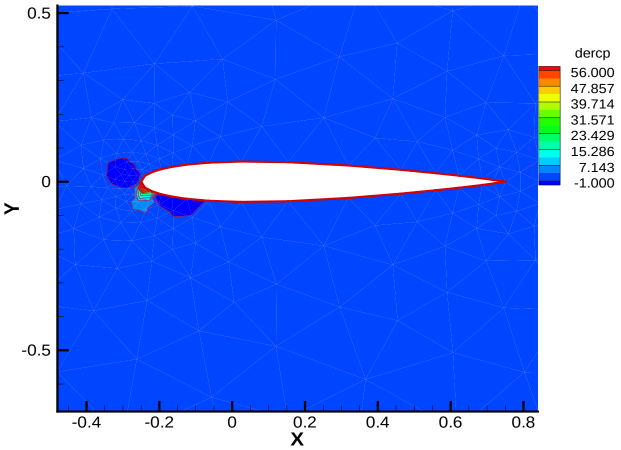

Figure 6.15: NACA 0012 FD Sensitivity of C_P at Node 21.

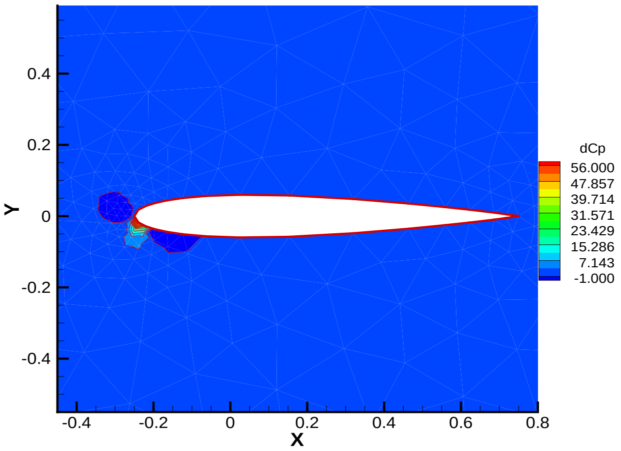

Figure 6.16: NACA 0012 Analytical Sensitivity of C_P at Node 21.

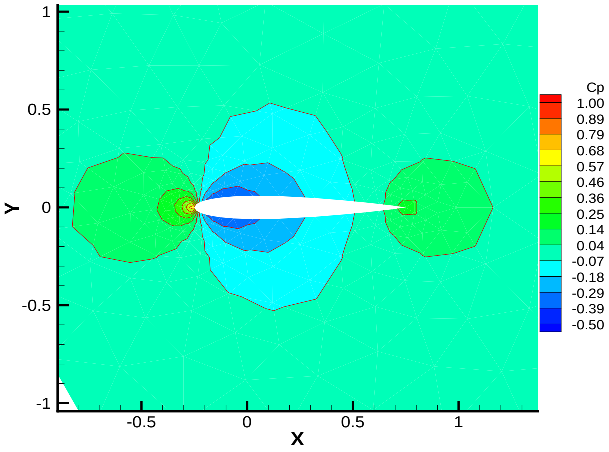

Figure 6.17: NACA 0012 Initial C_P Distribution.

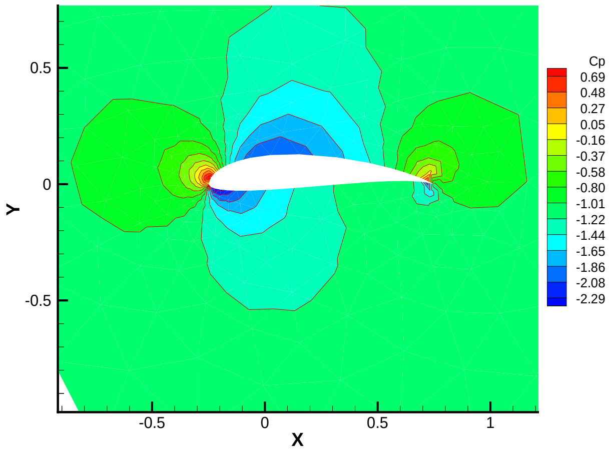

Figure 6.18: Deformed Airfoil Final C_P Distribution.

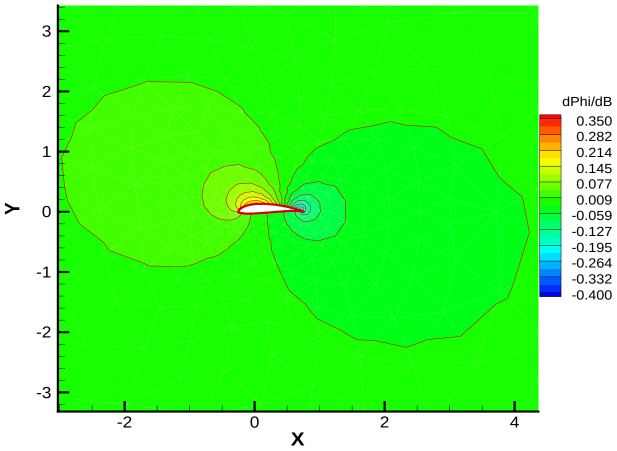

Figure 6.19: NACA 0012 initial Φ Sensitivity.

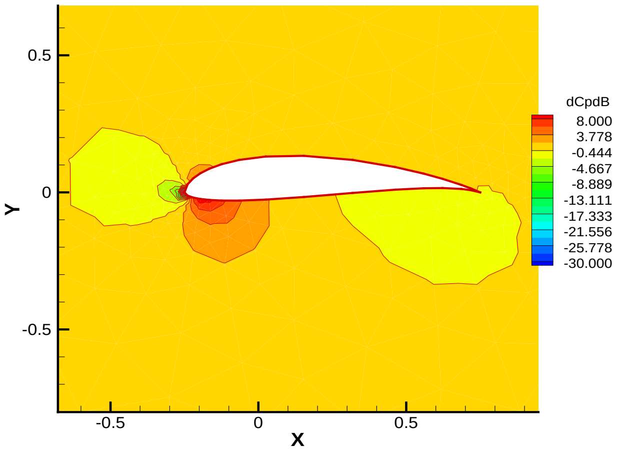

Figure 6.20: Deformed Airfoil Final C_P Sensitivity.

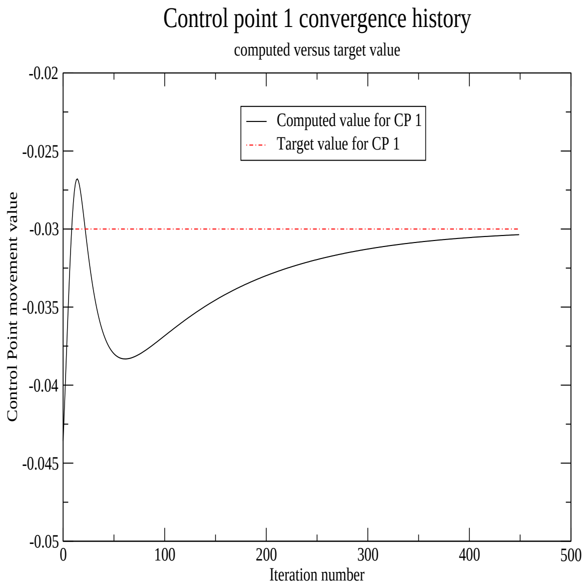

Figure 6.21: NACA 0012 Convergence History for DV 1.

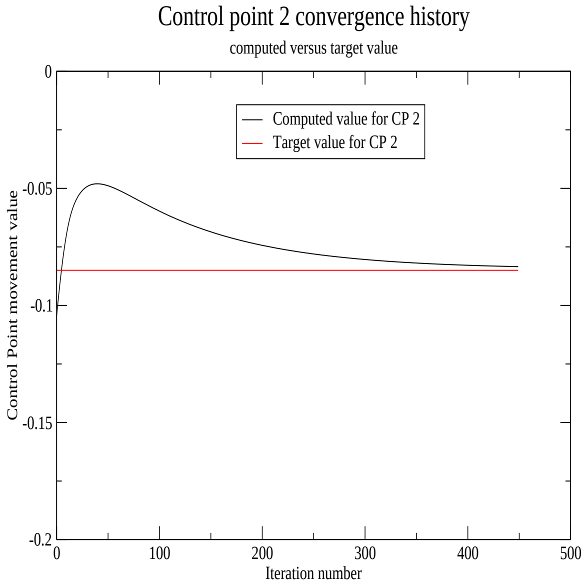

Figure 6.22: NACA 0012 Convergence History for DV 2.

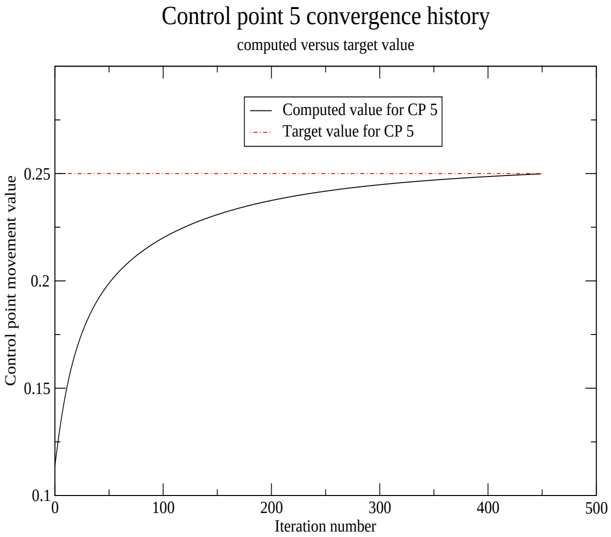

Figure 6.23: NACA 0012 Convergence History for DV 5.

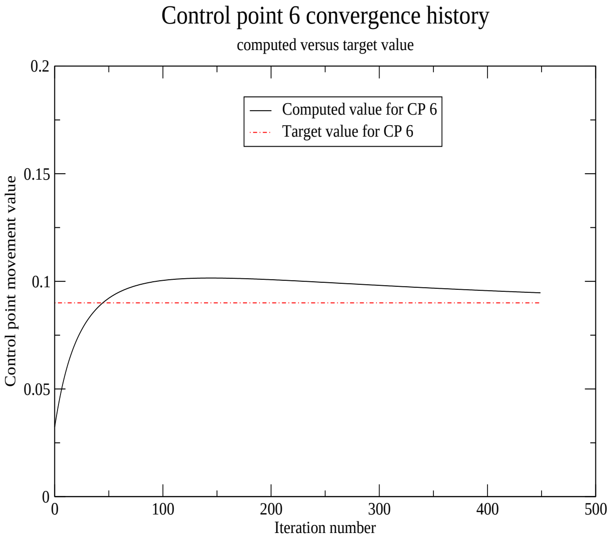

Figure 6.24: NACA 0012 Convergence History for DV 6.

Table 6.1: Comparison among initial, final and target value for NACA 0012 design variables.

DV	initial value	final value	target value
1	$-4.52417930000000024E-2$	$-3.03599612381197D-002$	-0.03000000000000000
2	-0.10904652200000001	$-8.34369843001496D-002$	$-8.5000000000000000E-2$
3	$-2.90710690000000015E-2$	$6.55115617904011D-002$	$7.0000000000000000E-2$
4	$4.52417930000000024E-2$	$6.55261555036117D-002$	$6.5000000000000000E-2$
5	0.10904652200000001	0.249851957384880	0.25000000000000000
6	$2.90710690000000015E-2$	$9.46798362304384D-002$	$9.0000000000000000E-2$

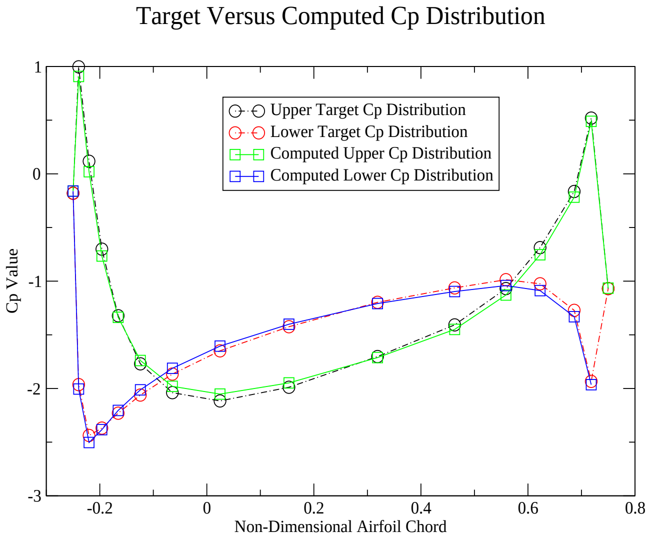

Figure 6.25: Target C_P Distribution Versus Computed C_P distribution for the Deformed Airfoil.

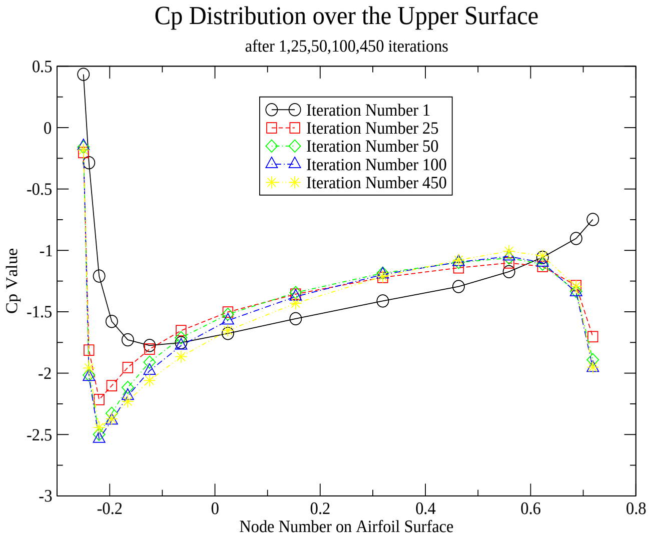

Figure 6.26: Deformed Airfoil C_P variation on upper surface.

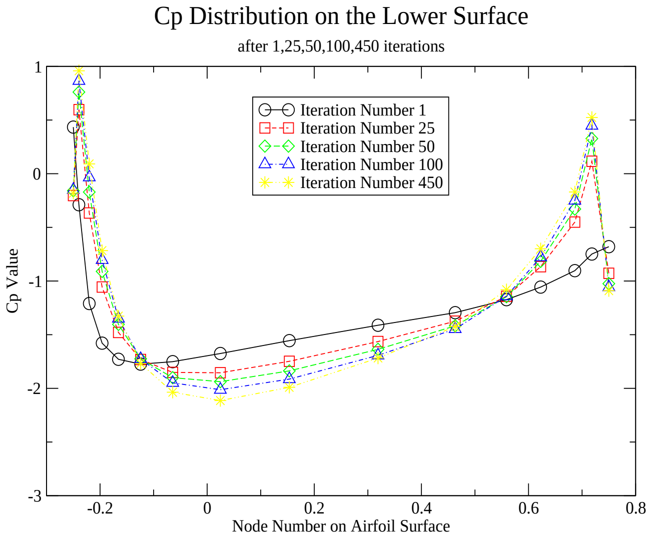

Figure 6.27: Deformed Airfoil C_P variation on lower surface.

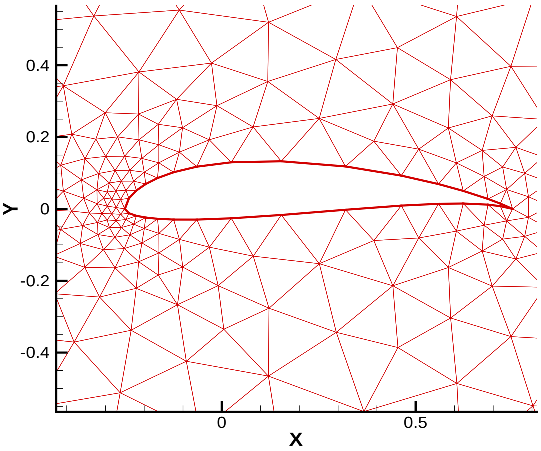

Figure 6.28: NACA 0012 Deformed Mesh Close-Up.

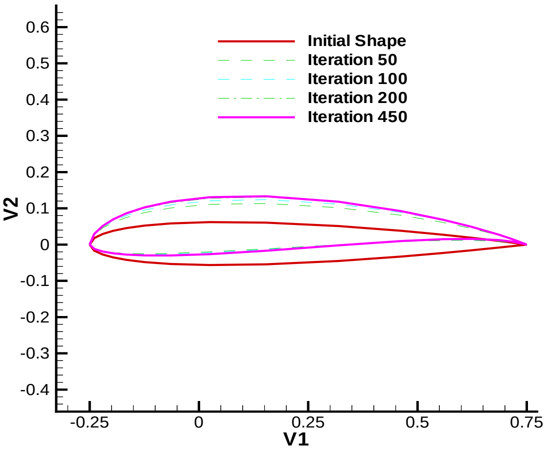

Figure 6.29: NACA 0012 design history.

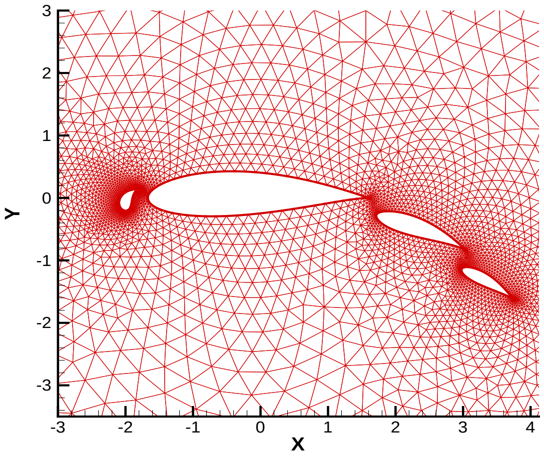

Figure 6.30: MEA Initial Computational Mesh.

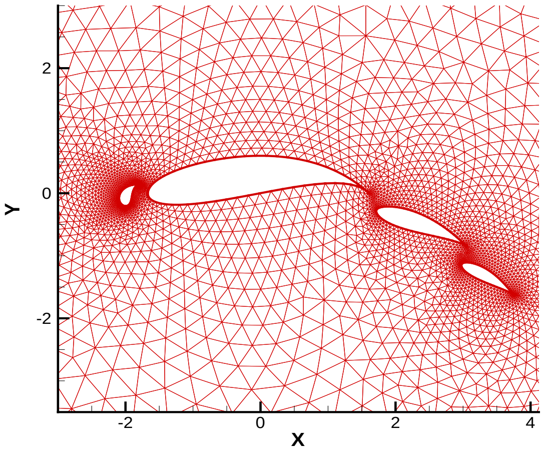

Figure 6.31: MEA Final Deformed Mesh.

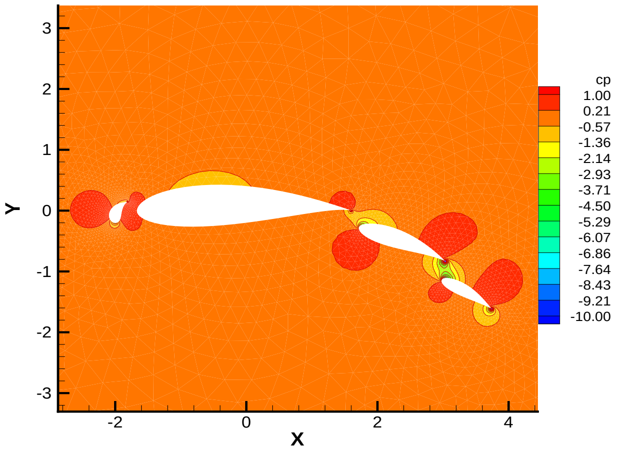

Figure 6.32: MEA Initial Cp Distribution.

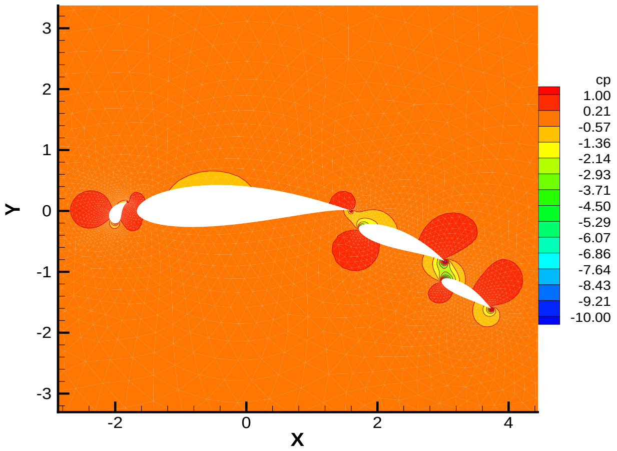

Figure 6.33: MEA final Cp Distribution.

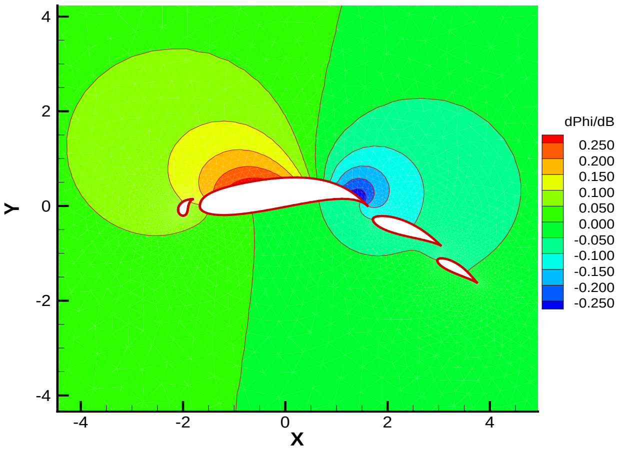

Figure 6.34: MEA Φ sensitivity.

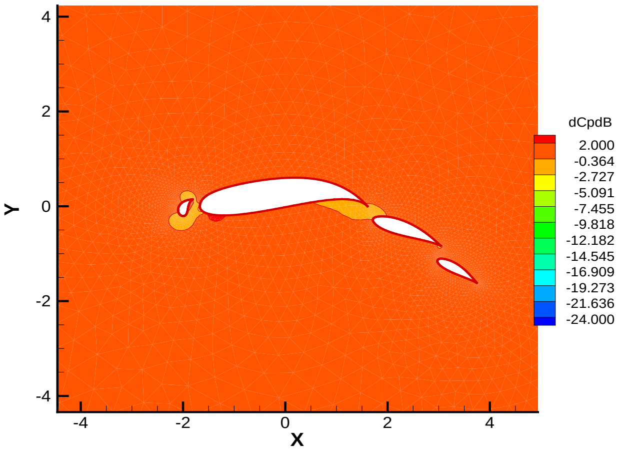

Figure 6.35: MEA C_P sensitivity.

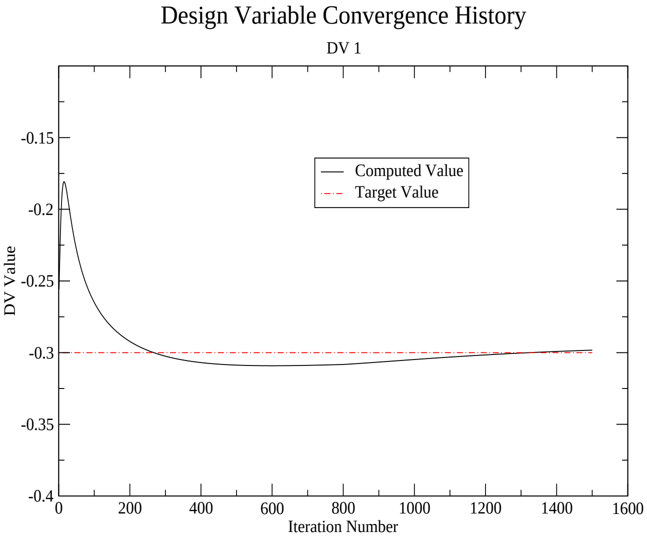

Figure 6.36: MEA Convergence History for DV 1.

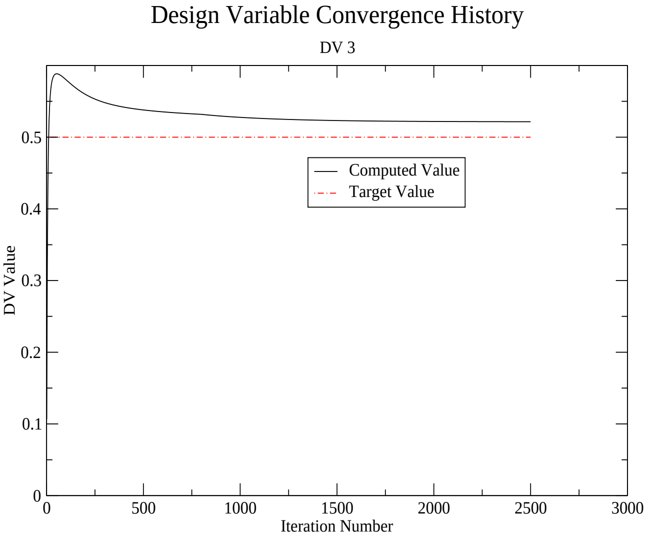

Figure 6.37: MEA Convergence History for DV 3.

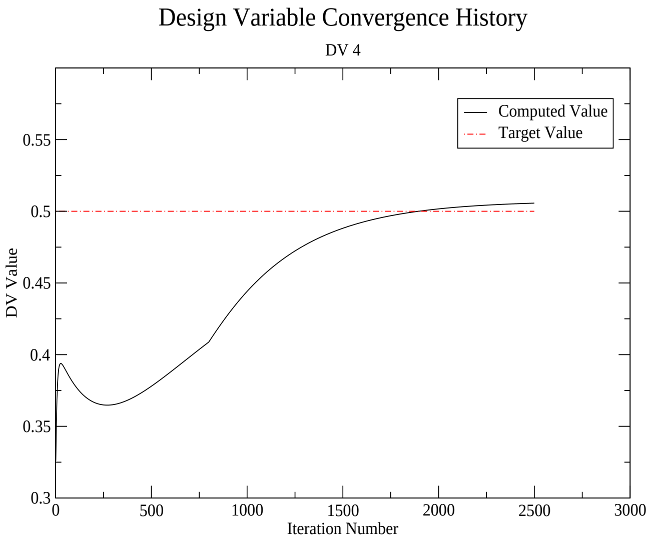

Figure 6.38: MEA Convergence History for DV 4.

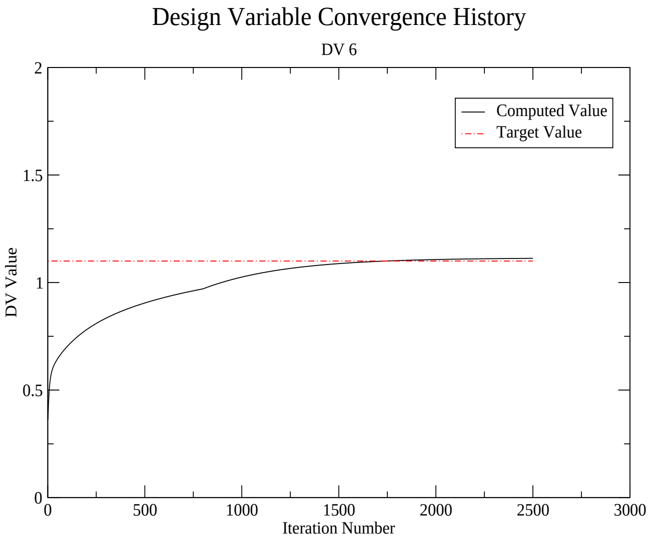

Figure 6.39: MEA Convergence History for DV 6.

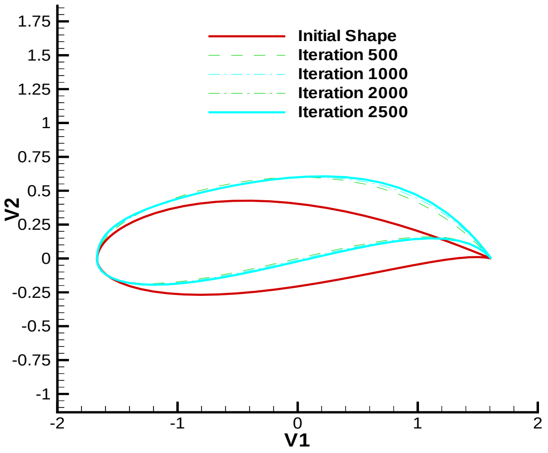

Figure 6.40: Airfoil Shapes at Different Design Cycles for MEA Airfoil.

CHAPTER VII

SUMMARY AND RECOMMENDATIONS.

A 2D finite element solver has been developed for performing analysis and sensitivity analysis with Poisson's equation. An application of Poisson's equation in fluid dynamics is that of potential flow, in which case Poisson's equation reduces to Laplace's equation. Capability of evaluating sensitivity derivatives has been added in order to perform design sensitivity analysis of non-lifting airfoils. Test cases have shown that as the grid size increases, the number of iterations required to reach convergence increases consistently, however so does the accuracy.

A 3D code has also been developed. Future development of the code will require the capability of aerodynamic design for lifting cases. This can be easily implemented by forcing the Kutta condition at the trailing edge of the airfoil and then updating the potential via a Dirichlet boundary condition along a branch cut from the trailing edge to the far field. The 3D code still requires validation. It is worth noting that due to the large grid size required for complex three dimensional geometries, a parallelization of the code is strongly suggested in order to perform the inverse design in a reasonable time. In fact, most of the time is spent during the optimization process because of the iterative nature of the optimization itself. Thus, parallelization of the code would strongly improve the optimization process in such a large files. Additionally, it is possible to use the code to perform optimization analysis in every field in which Poisson's equation has application.

REFERENCES

[1] Newman, J.C. III, Anderson, W.K., Whitfield, D.L., *”Multidisciplinary Sensitivity Derivatives Using Complex Variables,”* Mississippi State University Report MSSU-COE-ERC-98-08, 1998.

[2] Batina, J.T., *”Unstready Euler Airfoil Solutions Using Unstructured Dynamic Meshes,”* AIAA Paper 89-0115, 1989.

[3] Shanley, F.R., *Weight-Strength Analysis of Aircraft Structures.* McGraw-Hill, 1952.

[4] Norris, C.H., et al., *Structural Design for Dynamic Loads.* McGraw-Hill, 1959.

[5] Schimt, L.A., *”Structural Design by Systematic Synthesis,” proceedings of the* 2^{nd} *Conference on Electronic Computation,* 1960, pp 105-122.

[6] Schimt, L.A., and Thornton, W.A., *”Synthesis of an Airfoil at Supersonic Mach Number,”* NASA CR-144, 1965.

[7] Taylor III, A.C., Newman, P.A., Hou, G.J.W., and Jones, H.E., *”Recent Advances in Steady Compressible Aerodynamics,”* IMA Workshop on Flow Control, Nov. 1992.

[8] Hicks, R.M., Murman, E.M., and Vanderplaats, G.N., *”An assessment of Airfoil Design by Numerical Optimization,”* NASA TM-3092, July 1994.

[9] Hicks, R.M., and Henne, P.A., *”Wing Design by Numerical Optimization”* AIAA Paper 77-1247, 1977.

[10] Sobieski, J., *”The Case for Aerodynamic Sensitivity Analysis,”* NASA/VPI and SU Symposium on Sensitivity Analysis in Engineering, Sept 25-26, 1986.

[11] Andradotti, S., *”A Review of Simulation Optimization Techniques”* Proceedings of 1998 Winter Simulation Conference (Eds: D.J. Medeires, E.F. Watson, and M.S. Manivannan, 1998, pp. 151-158)

[12] Barthelemy, J.F., and Haftka, R.T., *”Approximation Concepts for Optimum Structural Design,” Structural Optimization*, Vol.5, 1993, pp. 129-144.

[13] Newman III, J.C., Taylor II, A.C., Barnwell, R.W., Newman, P.A., and Hou, G.J.-W, *”Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations, ”* J. Aircraft, Vol.36, No.1, Jan 1999, pp. 87-96.

[14] Thomas J.R. Hughes, *The Finite Element Method.* Dover publications, 2000.

[15] P.E. Gill, W. Murray, M.H. Wright, *Practical optimization.* Academic Press, 1982.

[16] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods.* Springer-Verlag, 1995.

[17] J.N. Reddy, *Introduction to the of Finite Elements method.* John Wiley and Sons, 1976.

[18] R.Taylor, O.C.Zienkiewicz, *The Finite Element Method,5th edition.* Butterworth-Heinemann, 2000.

[19] K. Huebner and E. Thornton and T. Byrom, *The Finite Element Method for Engineers.* John Wiley and Sons, 1995.

[20] Klaus-Jurgen Bathe, *Finite Element Procedures.* Prentice Hall, 1995.

[21] John C. Tannehill, Dale A. Anderson, Richard H. Pletcher, *Computational Fluid Mechanics And Heat Transfer.* Taylor and Francis, 1997.

[22] Garret N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design: With Applications.* VR and D, Inc., 1984.

[23] Dhatt G., *The Finite Element Displayed.* John Wiley and Sons, 1984.

[24] J.C. Newman III, *Integrated Multidisciplinary Design Optimization Using Discrete Sensitivity Analysis for Geometrically Complex Aeroelastic Configurations.* PhD Dissertation, Virginia Polytechnic Institute and State University, 1997.

[25] E.A. Luke, *A Rule-based Specification System for Computational Fluid Dynamics.* PhD Dissertation, Mississippi State University, 1999