Old Dominion University Research Foundation

DEPARTMENT OF MECHANICAL ENGINEERING & MECHANICS
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

AN ANALYTICAL APPROACH TO GRID SENSITIVITY ANALYSIS FOR NACA FOUR-DIGIT WING SECTIONS

[illegible]

By

I. Sadrehaghighi, Graduate Research Assistant

and

S.N. Tiwari, Principal Investigator

P69

Progress Report
For the period ended December 1991

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Cooperative Agreement NCC1-68
Dr. Robert E. Smith Jr., Technical Monitor
ACD-Computer Applications Branch

April 1992

DEPARTMENT OF MECHANICAL ENGINEERING & MECHANICS
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

AN ANALYTICAL APPROACH TO GRID SENSITIVITY ANALYSIS FOR NACA FOUR-DIGIT WING SECTIONS

By

I. Sadrehaghighi, Graduate Research Assistant

and

S.N. Tiwari, Principal Investigator

Progress Report
For the period ended December 1991

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Cooperative Agreement NCC1-68
Dr. Robert E. Smith Jr., Technical Monitor
ACD-Computer Applications Branch

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369

April 1992

FOREWORD

This is the progress report on the research project " Numerical Solutions of Three-Dimensional Navier-Stokes Equations for Closed-Bluff Bodies". Within the guidelines of the project, special attention was directed toward research activities in the area of "Grid Sensitivity in Airplane Design." The period of performance of this specific research was January 1, 1991 through December 31, 1991. This work was supported by the NASA Langley Research Center through Cooperate Agreement NCC1-68. The cooperate agreement was monitored by Dr. Robert E. Smith Jr. of Analysis and Computation Division (Computer Applications Branch), NASA Langley Research Center, Mail Stop 125.

ABSTRACT

An Analytical Approach To Grid Sensitivity Analysis For NACA Four-Digit Wing Sections

Ideen Sadrehaghighi[*]
Surendra N. Tiwari[†]

Sensitivity analysis in Computational Fluid Dynamics with emphasis on grids and surface parameterization is described. An interactive algebraic grid-generation technique is employed to generate C-type grids around NACA four-digit wing sections. An analytical procedure is developed for calculating grid sensitivity with respect to design parameters of a wing section. A comparison of the sensitivity with that obtained using a finite-difference approach is made. Grid sensitivity with respect to grid parameters, such as grid-stretching coefficients, are also investigated. Using the resultant grid sensitivity, aerodynamic sensitivity is obtained using the compressible two-dimensional thin-layer Navier-Stokes equations.

[*] Graduate Research Assistant

[†] Eminent Professor

TABLE OF CONTENTS

NOMENCLATURE

$\mathbf{Q}^*$	= vector of field variables
$\mathbf{X}^*$	= vector of physical coordinates
$\mathbf{R}$	= steady-state residual
$\mathbf{P}$	= vector of control parameters
$\mathbf{X}_\mathrm{D}$	= vector of design parameters
$\mathbf{U}$	= horizontal interpolant
$\mathbf{V}$	= normal interpolant
x, y, z	= physical coordinates
x_1, y_1	= airfoil surface coordinates
r	= surface grid distribution function
s	= far-field boundary grid distribution function
t	= grid stretching function
$R(r)$	= surface orthogonality function
$S(s)$	= far-field boundary orthogonality function
K_1, K_2	= orthogonality parameters
$\bar{\mathbf{k}}$	= stretching parameter
$\mathbf{T}$	= maximum thickness parameter
$\mathbf{M}$	= maximum camber parameter
$\mathbf{C}$	= camber location parameter
C_P	= pressure coefficient
C_D	= drag coefficient

C_L	= lift coefficient
C_f	= skin friction coefficient
C_X, C_Y	= force coefficients in x and y directions
$\bar{x}, \bar{y}_c$	= camber line coordinates
$y_{\mathbf{T}}$	= section thickness distribution
P_i	= surface pressure
u, v, w	= velocity components in x , y and z directions
ρ	= density
e	= energy per unit volume
M_∞	= free-stream Mach number
ρ_∞	= free-stream density
U_∞	= free-stream velocity
Re_∞	= free-stream Reynolds number
ξ, η, ζ	= computational coordinates
$\alpha_I^n(\xi, \mathbf{P}_0^\xi)$	= horizontal blending function
$\beta_J^m(\eta, \mathbf{P}_0^\eta)$	= normal blending function
θ	= surface slope
α	= angle of attack
τ_i	= surface skin friction

1. INTRODUCTION

Over the past several years, Computational Fluid Dynamics (CFD) has rapidly evolved. This has been the result of the immense advances in computational algorithms [1] and the development of supercomputers. These innovations have had a major impact on obtaining numerical flow simulations about complex geometries. On current supercomputers, viscous-compressible flow simulations about aircraft configuration can require several hours per steady-state solution [2]. Such large amounts of computational time are acceptable for proof-of-concept studies and selective analysis, but they are not acceptable for optimization and design. With advent of the next generation of parallel supercomputers [3], airplane design and optimization using nonlinear CFD (Euler and Navier-Stokes equations) should become routine. For an individual component, such as a wing, it is now reasonable to consider design and optimization in conjunction with nonlinear CFD [4].

An essential element in design and optimization is acquiring the sensitivity of functions of CFD solutions with respect to control parameters. For aerodynamic surfaces, the control parameters specify the shapes of the surfaces. This affects the surface grid and the field grid which, in turn, affects the flow- field solution. There are two basic components in obtaining aerodynamic sensitivity. They are: (1) obtaining the sensitivity of the governing equations with respect to the state variables; and (2) obtaining the sensitivity of the grid with respect to the defining parameters. The sensitivity of the state variables with respect to the defining parameters are described by a linear-algebraic relation [5]. This study concentrates on the grid sensitivity and parameterization of aerodynamic surfaces.

The simplest method for calculating grid sensitivity is based upon finite-difference approximation. In this approach, a design parameter is perturbed from the nominal value, a new grid is obtained, and the difference between the new and the old grid is used to obtain the grid sensitivity derivatives. This direct, or brute force technique, has the disadvantages of being computationally intensive. It is the objective of the analytical approach to avoid the time consuming and costly numerical differentiation. In addition, the analytic derivatives are exact instead of approximate.

Taylor et al. [6] set the stage for the development of the technique used in this study. For a steady-state solution of the Euler or Navier-Stokes equations, the sensitivity of a function of the solution with respect to the control parameters is to be found. The problem includes the determination of sensitivity of the grid with respect to control parameters. For grid generation, algebraic transfinite interpolation [7] is ideally suited for the study of parameterization. Parameters are sub-grouped according to their purposes (grid spacing control and surface shape control). The objective is to cast the surface parameterization in terms of design parameters rather than geometric variables. Geometric parameterization has only local control and consequently requires large number of parameters to define a grid. A specialized application of transfinite interpolation [8] is cast in terms of design parameters for a class of wing sections and wings. The grid sensitivity of this system is discussed.

2. ALGEBRAIC GRID GENERATION

2.1 Basic Formulation

Structured algebraic grid generation techniques can be thought of as transformations from a rectangular computational domain to an arbitrarily-shaped physical domain [7]. The transformations are governed by the vector of control parameters **P**. That is,

$$\mathbf{X}(\xi,\eta,\zeta,\mathbf{P}) = \left\{ \; x(\xi,\eta,\zeta,\mathbf{P}) \quad y(\xi,\eta,\zeta,\mathbf{P}) \quad z(\xi,\eta,\zeta,\mathbf{P}) \; \right\}^T \tag{2.1}$$

where

$$0 \leq \xi \leq 1, \;\; 0 \leq \eta \leq 1, \;\; and \;\; 0 \leq \zeta \leq 1.$$

A discrete subset of the vector-valued function $\mathbf{X}(\xi_i,\eta_j,\zeta_k,\mathbf{P}) \equiv \mathbf{X}\{\; x \; y \; z \;\}^T_{i,j,k} \equiv \mathbf{X}^*$ is a structured grid for $\xi_i = \frac{i-1}{L-1}, \eta_j = \frac{j-1}{M-1}, \zeta_k = \frac{k-1}{N-1}$, where $i = 1,2,3\cdots,L, \;\; j = 1,2,3,\cdots,M$ and $k = 1,2,3,\cdots,N$.

The dominant algebraic approach for grid generation is the *Transfinite Interpolation* scheme. The general methodology was first described by Gordon [9] , and there have been numerous variations applied to particular problems. The methodology can be presented as recursive formulas composed of univariate interpolations [10] or as the Boolean sum of univariate interpolations [7]. Here, we follow the Boolean sum representation but; for brevity, we restrict the process to two dimensions and omit some of the details that can be found in Ref. [7]. Also, to be consistent with the example considered below, the parameterization is restricted to functions and first derivatives at the boundaries ($\xi = 0, 1$) and ($\eta = 0, 1$) and control in the interpolation functions. The transformation is

$$\mathbf{X}(\xi, \eta, \mathbf{P}) = \mathbf{U} \oplus \mathbf{V} = \mathbf{U} + \mathbf{V} - \mathbf{UV} \tag{2.2}$$

where

$$\mathbf{U} = \sum_{I=1}^{2} \sum_{n=0}^{1} \alpha_I^n(\xi, \mathbf{P}_0^\xi) \frac{\partial^n \mathbf{X}(\xi_I, \eta, \mathbf{P}_I^\eta)}{\partial \xi^n} \tag{2.3}$$

and

$$\mathbf{V} = \sum_{J=1}^{2} \sum_{m=0}^{1} \beta_J^m(\eta, \mathbf{P}_0^\eta) \frac{\partial^m \mathbf{X}(\xi, \eta_J, \mathbf{P}_J^\xi)}{\partial \eta^m}. \tag{2.4}$$

The term **UV** (not expanded here) is the tenser product of the two univariate interpolations. The boundary curves and their derivatives ($\frac{\partial^n \mathbf{X}(\xi_I, \eta, \mathbf{P}_I^\eta)}{\partial \eta^n}$ and $\frac{\partial^m \mathbf{X}(\xi, \eta_J, \mathbf{P}_J^\xi)}{\partial \eta^m} I, J =$ $1, 2 \; m, n = 0, 1$) are blended into the interior of the physical domain by the interpolation functions $\alpha_I(\xi, \mathbf{P}_0^\xi)$ and $\beta_J(\eta, \mathbf{P}_0^\eta)$. The boundary grid, the derivatives at the boundary grid and the spacing between points are governed by the parameters $\left\{\mathbf{P}_J^\xi \; \mathbf{P}_I^\eta\right\}^T$. The interpolation functions are controlled with the parameters $\left\{\mathbf{P}_0^\xi \; \mathbf{P}_0^\eta\right\}^T$. The entire set of control parameters can be thought of as a vector $\left\{\mathbf{P}_0^\xi \; \mathbf{P}_0^\eta \; \mathbf{P}_J^\xi \; \mathbf{P}_I^\eta\right\}^T \equiv \mathbf{P}$.

2.2 Grid Algorithm

An interactive univariate version of Eq.(2.2) using only the normal interpolant **V** is developed. This, known as *Hermite Cubic Interpolation*, matches both the function and its first derivative at two boundaries. An analytical approximation of the physical coordinates for a class of wing sections can be expressed as

$$\begin{aligned} x = {} & x_1(r, \mathbf{P}_1^\xi)\beta_1^0(t, \mathbf{P}_0^\eta) + R(r)\frac{\partial x_1(r, \mathbf{P}_1^\xi)}{\partial t}\beta_1^1(t, \mathbf{P}_0^\eta) \\ & + x_2(s, \mathbf{P}_2^\xi)\beta_2^0(t, \mathbf{P}_0^\eta) + S(s)\frac{\partial x_2(s, \mathbf{P}_2^\xi)}{\partial t}\beta_2^1(t, \mathbf{P}_0^\eta) \end{aligned} \tag{2.5}$$

$$y = y_1(r, \mathbf{P}_1^\xi)\beta_1^0(t, \mathbf{P}_0^\eta) + R(r)\frac{\partial y_1(r, \mathbf{P}_1^\xi)}{\partial t}\beta_1^1(t, \mathbf{P}_0^\eta)$$
$$+ y_2(s, \mathbf{P}_2^\xi)\beta_2^0(t, \mathbf{P}_0^\eta) + S(s)\frac{\partial y_2(s, \mathbf{P}_2^\xi)}{\partial t}\beta_2^1(t, \mathbf{P}_0^\eta) \qquad (2.6)$$

where

$$\beta_1^0(t, \mathbf{P}_0^\eta) = 2t^3 - 3t^2 + 1,$$
$$\beta_1^1(t, \mathbf{P}_0^\eta) = t^3 - 2t^2 + t,$$
$$\beta_2^0(t, \mathbf{P}_0^\eta) = -2t^3 + 3t^2,$$
$$\beta_2^1(t, \mathbf{P}_0^\eta) = t^3 - t$$

and

$$0 \leq t \leq 1.$$

Five functions $r = f_1(\xi)$, $s = f_2(\xi)$, $R(r) = K_1 f_3(\xi)$, $S(s) = K_2 f_4(\xi)$ and $t = f_5(\eta)$ and their implied defining parameters control the grid spacing on the boundaries and the interior grid. Functions r and s define the grid spacing for lower and upper boundaries respectively, while $R(r)$ and $S(s)$ specify the magnitude of orthogonality along those boundaries. The parameter t defines the grid distribution for the connecting curves between the two boundary. The quantities K_1 and K_2 are parameters that scale the magnitude of the orthogonality at the boundaries. Increasing K_1 and K_2 extends the orthogonality of the grid into the interior domain. Excessively large values of K_1 and K_2 can also cause the grid lines to intersect themselves, which is not a desirable phenomena.

A discrete uniform distribution of the computational coordinate, can be mapped into an arbitrary distribution of the physical coordinate, using the specified control function. The essence of mapping, is that the abscissa corresponds to the percentage of grid points and the ordinate corresponds to a particular control function which, in turn, relates to the geometric definition of the physical domain. The control function, can be either a specified analytical function, or for more general purposes, a cubic- spline function. For example, the function

$$f_5(\eta) = \frac{e^{K\eta} - 1}{e^{K} - 1} \tag{2.7}$$

would concentrate grid points close to the bottom or the top boundary depending on the magnitude and the sign of the constant K. Figure 2.1 shows a unit square used as a control domain for grid spacing. Figure 2.2 presents the parametric representation of the boundaries and the cubic connecting function of Eqs.(2.5) and (2.6). Appendix A provides a complete listing of the FORTRAN source module for this type of grid generation algorithm.

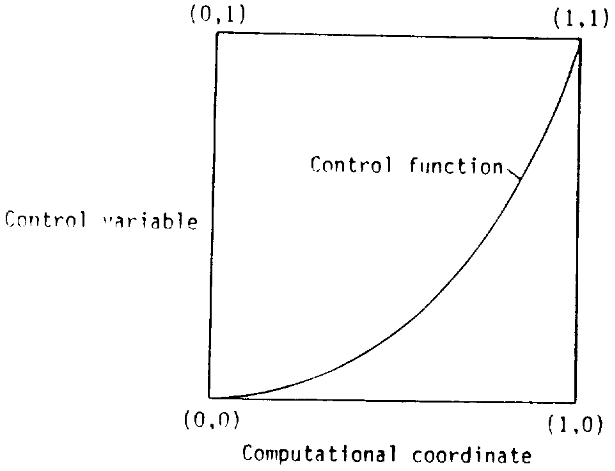

Fig. 2.1 Unit square control domain

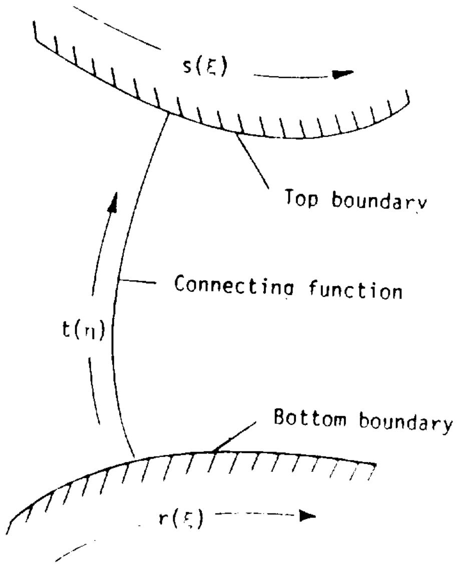

Fig. 2.2 Cubic connecting function

3. PROBLEM STATEMENT AND METHOD OF SOLUTION

3.1 Theoretical Formulation

Let $\mathbf{Q}$ be a solution of the Euler or Navier-Stokes equations on the domain $\mathbf{X}$ and $\mathbf{Q}^*$ be a discrete solution on the grid $\mathbf{X}^*$ where

$$\mathbf{Q}^* = \left\{ \rho \quad \rho u \quad \rho v \quad \rho w \quad \rho e \right\}^T_{i,j,k} \qquad \mathbf{X}^* = \left\{ x \quad y \quad z \right\}^T_{i,j,k} \tag{3.1}$$

and

$$\frac{\partial \mathbf{Q}}{\partial t} = \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*) = 0. \tag{3.2}$$

Here, $\mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*)$ is the residual of a steady-state solution as $t \to \infty$. Let $\mathbf{P}$ be a vector of parameters that controls the grid $\mathbf{X}^*$ such that $\mathbf{X}^* = \mathbf{X}(\xi_i, \eta_j, \zeta_k, \mathbf{P})$ where ξ, η and ζ are computational coordinates [7]. The numerical sensitivity of a function $F(\mathbf{Q}(\mathbf{X}))$ with respect to the control parameters is

$$F_p(\mathbf{Q}^*(\mathbf{X}^*)) = \left\{ \frac{\partial F(\mathbf{Q}^*(\mathbf{X}^*))}{\partial \mathbf{P}} \right\} = \left\{ \frac{\partial F(\mathbf{Q}^*(\mathbf{X}^*))}{\partial \mathbf{Q}^*} \right\} \left\{ \frac{\partial \mathbf{Q}^*(\mathbf{X}^*)}{\partial \mathbf{P}} \right\}. \tag{3.3}$$

The fundamental sensitivity equation containing $\left\{ \frac{\partial \mathbf{Q}^*(\mathbf{X}^*)}{\partial \mathbf{P}} \right\}$ and described by Taylor et al. [6] is

$$\left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*)}{\partial \mathbf{Q}^*(\mathbf{X}^*)} \right] \left\{ \frac{\partial \mathbf{Q}^*(\mathbf{X}^*)}{\partial \mathbf{P}} \right\} + \left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*)}{\partial \mathbf{X}^*} \right] \left\{ \frac{\partial \mathbf{X}^*}{\partial \mathbf{P}} \right\} = 0. \tag{3.4}$$

It is important to notice that Eq.(3.4) is a set of linear, algebraic equations , and the matrices $\left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*)}{\partial \mathbf{Q}^*(\mathbf{X}^*)} \right]$ and $\left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*), \mathbf{X}^*)}{\partial \mathbf{X}^*} \right]$ are well understood [6]. The quantitiy $\left\{\frac{\partial \mathbf{Q}^{*}(\mathbf{X}^{*})}{\partial \mathbf{P}}\right\}$ is the solution to Eq.(3.4) given the sensitivity of the grid with respect to the parameters. Therefore, the grid sensitivity problem is described by

$$\left\{\frac{\partial \mathbf{X}^{*}}{\partial \mathbf{P}} \equiv \mathbf{GridSensitivity}\right\}.$$

3.2 Surface Parameterization

In Eq.(2.2), the parameterization is on the boundaries and in the interpolation functions. The most general parameterization of the boundaries would be to specify every grid point $\mathbf{X}^{*}_{I,J}$ (i.e., each boundary grid point is a parameter). This conceivably could be desirable for the boundaries corresponding to an airplane surface to allow a design procedure to have the greatest possible flexibility. This, however, is impractical from a computational point of view. A compromise is to specify the knots of a spline function or the control polygon for a Bézier function. Even with this compromise, it could require hundreds of parameters for a wing. Here we propose a quasi-analytical parameterization in terms of design variables. For instance, a class of wing sections is specified by two camber-line parameters and a thickness distribution parameter; a wing is specified by several wing sections; and the wing surface is interpolated from the sections. In this manner, an airplane component can be specified by tens of parameters instead of hundreds or housands of parameters. The disadvantage is that a great deal of generality is not available, but the generality is a moot point if computational capacity cannot accommodate it. For design and optimization with CFD, at the present time, it is advocated here to use a small number of design parameters for boundary definition.

3.3 Grid Sensitivity

As it is stated in the introduction, the simplest way to obtain grid sensitivity with respect to the parameterization is to vary the parameters and finite difference the results. This is computationally expensive compared to analytically differentiating Eq.(2.2). Therefore, we propose the latter. Grid sensitivity can be used for grid adaptation, or it can be used for boundary design. For adaptation, the grid sensitivity with respect to those parameters that control the grid spacing and the shape of the field grid away from fixed boundaries are desired. The sensitivity is used to improve some grid-quality function of the solution. For design and optimization the sensitivity of the grid with respect to the parameters that define the shape of a boundary is desired. The sensitivity is used to improve a design function of the solution.

4. WING-SECTION EXAMPLE

4.1 Wing-Section Parameterization

Much research has been devoted to the development and representation of wing sections. The NACA four-digit wing sections are examined for grid-generation parameterization. Families of wing sections are described by combining a mean line and a thickness distribution. The resultant expressions possess the necessary features that suit the problem, mainly the concise description of a wing section in terms of several design parameters. Reference 11 provides the general equations which define a mean line and a thickness distribution about the mean line. The design parameters are: $\mathbf{T} \equiv$ the maximum thickness, $\mathbf{M} \equiv$ the maximum ordinate of the mean line or camber, and $\mathbf{C} \equiv$ chordwise position of maximum ordinate. The numbering system for NACA four-digit wing-section is based on the geometry of the section. The first and second integers represent **M** and **C** respectively, while the third and fourth integers represent **T**. Symmetrical sections are designated by zeros for the first two integers, as in the case of NACA 0012 wing-section. Figure 4.1 provides a schematic of the section definition. The ξ-coordinate is first mapped into the chord line $\bar{x} = \bar{x}(r) = \bar{x}(f_1(\xi))$ forward and then reversed to cover both the top and bottom of the section. The mean line equation is

$$\bar{y}_c(\bar{x}) = \frac{\mathbf{M}}{\mathbf{C}^2}(2\mathbf{C}\bar{x} - \bar{x}^2)\ , \quad \bar{x} \le \mathbf{C} \tag{4.1}$$

$$\bar{y}_c(\bar{x}) = \mathbf{M}\frac{(1 - 2\mathbf{C} + 2\mathbf{C}\bar{x} - \bar{x}^2)}{(1 - \mathbf{C})^2}, \quad \bar{x} \ge \mathbf{C}. \tag{4.2}$$

The section thickness is given by

$$y_{\mathbf{T}}(\bar{x}) = \frac{\mathbf{T}}{0.2}(0.2969\bar{x}^{\frac{1}{2}} - 0.126\bar{x} - 0.3516\bar{x}^2 + 0.2843\bar{x}^3 - 0.1015\bar{x}^4). \tag{4.3}$$

The section coordinates are

$$x_1(r, \mathbf{P}_1^{\xi}) = \bar{x} \qquad y_1(r, \mathbf{P}_1^{\xi}) = \bar{y}_c(\bar{x}) \pm y_{\mathbf{T}}(\bar{x}). \tag{4.4}$$

Figures 4.2 and 4.3 show sample grids for NACA 0012 and NACA 8512 section using this procedure. The orthogonality at the far-field boundary is ignored. For solid boundary, the orthognality is obtained using the components of unit normal vector at the surface

$$\frac{\partial x_1(r, \mathbf{P}_1^{\xi})}{\partial t} = \mp sin\theta \qquad \frac{\partial y_1(r, \mathbf{P}_1^{\xi})}{\partial t} = \pm cos\theta \qquad \theta = tan^{-1}\left(\frac{\partial y_1(r, \mathbf{P}_1^{\xi})}{\partial x_1(r, \mathbf{P}_1^{\xi})}\right). \tag{4.5}$$

Figure 4.4 shows a wing-surface grid derived from three differently-specified sections in the spanwise direction. The surface grid results from the distribution function$f_1(\xi)$, and interpolation of the design parameters for the three wing sections in the spanwise direction. In addition to the three design parameters for each wing section, it is necessary to specify their relative chord lengths and positions. The additional design parameters can be: leading edge sweep, trailing edge sweep, dihedral angle, reference chord length, and section-span locations.

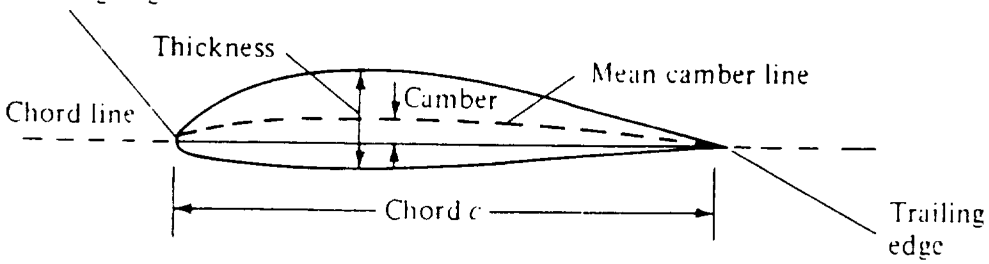

Fig. 4.1 Schematic of wing-section

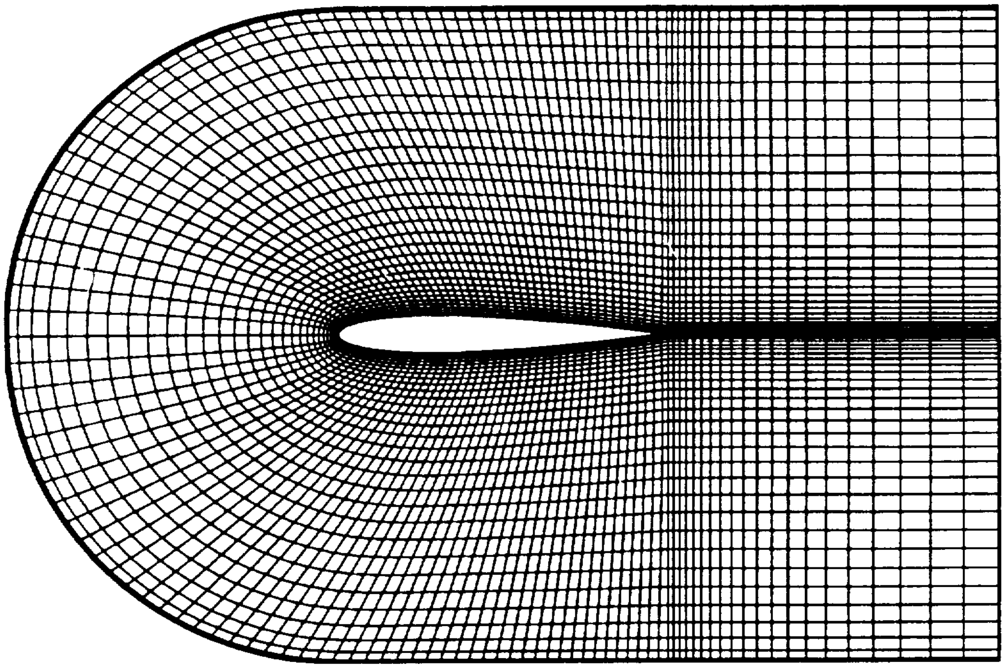

Fig. 4.2 Example grid for NACA 0012 wing-section

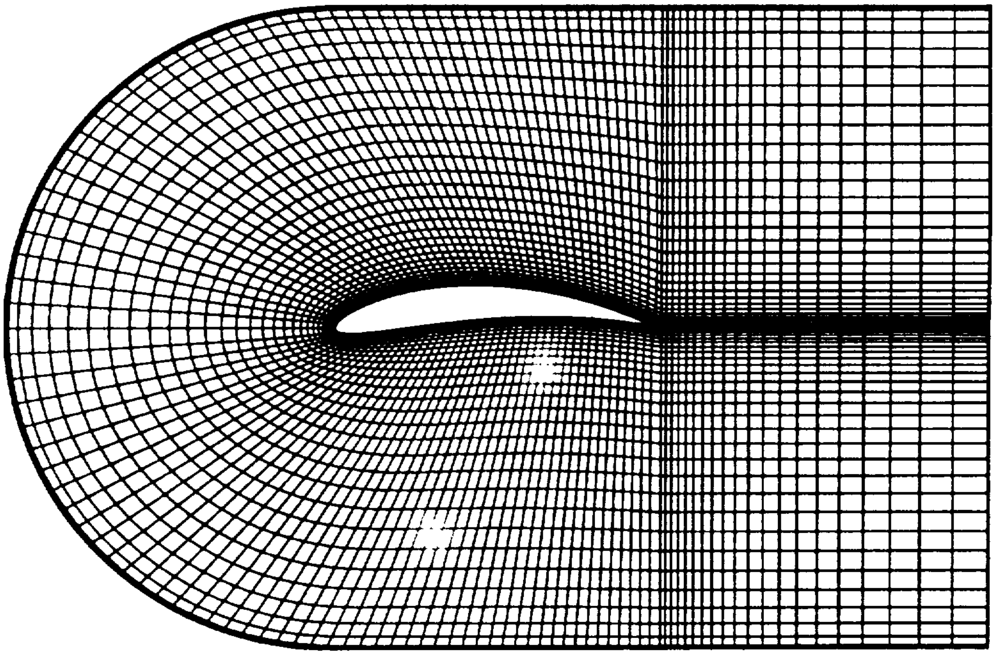

Fig. 4.3 Example grid for NACA 8512 wing-section

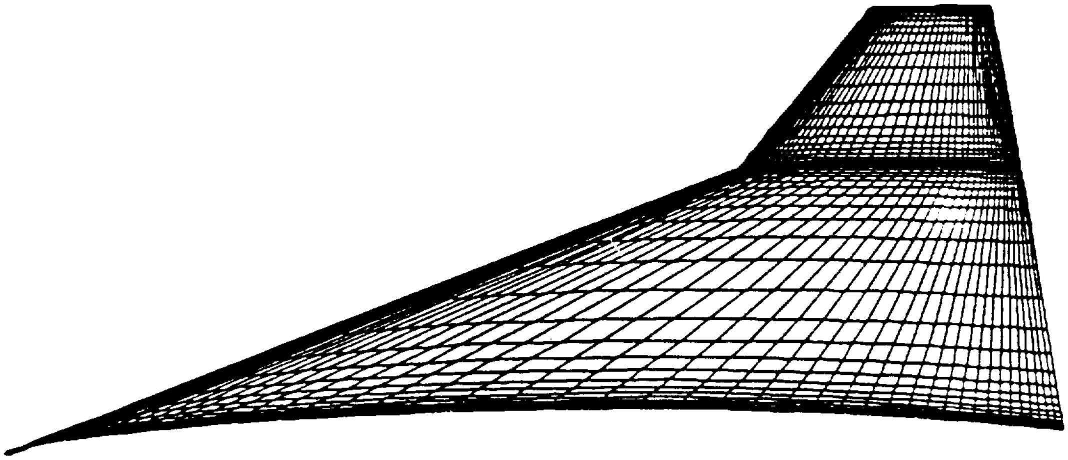

Fig. 4.4 Example wing surface grid (top view)

Appendix B provides the source module for generating the surface of NACA four-digit wing-sections.

4.2 Grid Sensitivity With Respect To Control Parameters

There are two types of control parameters involved in this analysis. First, there are the design parameters (**T**, **M**, and **C**) which specify the shape of the primary boundary and secondly, there are the parameters that define the other boundaries and the the spacing between grid points. Here we express, in part, the sensitivity of the grid with respect to the design parameter vector $\mathbf{X_D}$, and with $\bar{\mathbf{k}}$ a stretching parameter in the interpolation functions related by $f_5(\eta, \bar{\mathbf{k}}))$. The grid sensitivity with respect to design parameters at the outer boundary has been ignored. Also, due to zero orthogonality at the outer boundary, a direct differentiation of Eqs.(2.5) and (2.6) with respect to $\mathbf{X_D}$ yields to

$$\frac{\partial x}{\partial \mathbf{X_D}} = \frac{\partial x}{\partial x_1}\frac{\partial x_1(r, \mathbf{P}_1^{\xi})}{\partial \mathbf{X_D}} + \frac{\partial x}{\partial x_1'}\frac{\partial x_1'(r, \mathbf{P}_1^{\xi})}{\partial \mathbf{X_D}} \tag{4.6}$$

$$\frac{\partial y}{\partial \mathbf{X_D}} = \frac{\partial y}{\partial y'_1}\frac{\partial y'_1(r,\mathbf{P}^\xi_1)}{\partial \mathbf{X_D}} + \frac{\partial y}{\partial y_1}\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial \mathbf{X_D}} \tag{1.7}$$

where

$$\mathbf{X_D} = (\mathbf{T},\mathbf{M},\mathbf{C}) \tag{1.8}$$

$$\frac{\partial x}{\partial x_1} = \frac{\partial y}{\partial y_1} = \beta^0_1(t,\mathbf{P}^\eta_0) \qquad \frac{\partial x}{\partial x'_1} = \frac{\partial y}{\partial y'_1} = R(r)\beta^1_1(t,\mathbf{P}^\eta_0). \tag{1.9}$$

The prime indicates differentiation with respect to t and can be substituded from Eq.(4.5). Since $x_1(r,\mathbf{P}^\xi_1)$ is independent of design parameters $\mathbf{X_D}$, then

$$\frac{\partial x_1(r,\mathbf{P}^\xi_1)}{\partial \mathbf{X_D}} = 0.0. \tag{1.10}$$

The x coordinate sensitivity, Eq.(4.6), can now be reduced to

$$\frac{\partial x}{\partial \mathbf{X_D}} = R(r)\beta^1_1(t,\mathbf{P}^\eta_0)\frac{\partial(\mp sin\theta)}{\partial \mathbf{X_D}} \qquad \theta = tan^{-1}\left(\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}\right). \tag{1.11}$$

Using the relation

$$\frac{\partial}{\partial \mathbf{X_D}}tan^{-1}u = \frac{1}{1+u^2}\frac{\partial u}{\partial \mathbf{X_D}} \tag{1.12}$$

the x coordinate sensitivity becomes

$$\frac{\partial x}{\partial \mathbf{X_D}} = \mp R(r)\beta^1_1(t,\mathbf{P}^\eta_0)cos\theta\frac{1}{1+\left(\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}\right)^2}\frac{\partial}{\partial \mathbf{X_D}}\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}. \tag{1.13}$$

The term $\frac{\partial}{\partial \mathbf{X_D}}\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}$ can be evaluated by direct differentiation of Eq.(4.4). The y coordinate sensitivity with respect to design papameters can be obtained using similar procedure. Equation (4.7) can be modified to

$$\frac{\partial y}{\partial \mathbf{X_D}} = \beta^0_1(t,\mathbf{P}^\eta_0)\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial \mathbf{X_D}} \mp R(r)\beta^1_1(t,\mathbf{P}^\eta_0)sin\theta\frac{1}{1+\left(\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}\right)^2}\frac{\partial}{\partial \mathbf{X_D}}\frac{\partial y_1(r,\mathbf{P}^\xi_1)}{\partial x_1(r,\mathbf{P}^\xi_1)}. \tag{1.14}$$

All terms at the right hand side of Eqs.(4.13) and (4.14) can be evaluated explicitly due to analytical parameterization of the surface for this particular example. The grid sensitivity with respect to the stretching parameter $\bar{\mathbf{k}}$ are

$$\frac{\partial x}{\partial \bar{\mathbf{k}}} = x_1(r,\mathbf{P}^\xi_1)\frac{\partial \beta^0_1(t,\mathbf{P}^\eta_0)}{\partial t}\frac{\partial t}{\partial \bar{\mathbf{k}}} + R(r)\frac{\partial x_1(r,\mathbf{P}^\xi_1)}{\partial \eta}\frac{\partial \beta^1_1(t,\mathbf{P}^\eta_0)}{\partial t}\frac{\partial t}{\partial \bar{\mathbf{k}}}$$

$$+ x_2(r, \mathbf{P}_2^{\xi}) \frac{\partial \beta_2^0(t, \mathbf{P}_0^{\eta})}{\partial t} \frac{\partial t}{\partial \bar{\mathbf{k}}} + S(s) \frac{\partial x_2(s, \mathbf{P}_2^{\xi})}{\partial \eta} \frac{\partial \beta_2^1(t, \mathbf{P}_0^{\eta})}{\partial t} \frac{\partial t}{\partial \bar{\mathbf{k}}} \tag{4.15}$$

where

$$\frac{\partial \beta_1^0(t, \mathbf{P}_0^{\eta})}{\partial t} = 6t^2 - 6t \qquad \frac{\partial \beta_1^1(t, \mathbf{P}_0^{\eta})}{\partial t} = 3t^2 - 4t + 1$$

$$\frac{\partial \beta_2^0(t, \mathbf{P}_0^{\eta})}{\partial t} = -6t^2 + 6t \qquad \frac{\partial \beta_2^2(t, \mathbf{P}_0^{\eta})}{\partial t} = 3t^2 - 1.$$

An example of the stretching function is

$$t = \frac{e^{\bar{\mathbf{k}}\eta} - 1}{e^{\bar{\mathbf{k}}} - 1} \tag{4.16}$$

$$\frac{\partial t}{\partial \bar{\mathbf{k}}} = \frac{(e^{\bar{\mathbf{k}}} - 1)\eta e^{\bar{\mathbf{k}}\eta} - (e^{\bar{\mathbf{k}}\eta} - 1)e^{\bar{\mathbf{k}}}}{(e^{\bar{\mathbf{k}}} - 1)^2}. \tag{4.17}$$

Similar developments can be extended to other grid control parameters such as the distribution of grid point around the wing section and magnitude of orthogonality at the boundaries. Appendix C provides the source module for grid sensitivity of NACA four-digit wing-sections with respect to design parameters.

4.3 Flow Sensitivity With Respect To Control Parameters

The flow sensitivity coefficient $\left\{\frac{\partial \mathbf{Q}^*(\mathbf{X}^*)}{\partial \mathbf{P}}\right\}$ can now be evaluated using the fundamental sensitivity equation, Eq.(3.4). The Jacobian matrices $\left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*),\mathbf{X}^*)}{\partial \mathbf{Q}^*(\mathbf{X}^*)}\right]$ and $\left[\frac{\partial \mathbf{R}(\mathbf{Q}^*(\mathbf{X}^*),\mathbf{X}^*)}{\partial \mathbf{X}^*}\right]$ are obtained using an implicit time integration of the 2-D thin-layer Navier-Stokes equations [10]. These equations are solved in their conservation form using an upwind cell-centered finite-volume formulation. A detailed description of the procedure is found in Refs. [12-17] and is not repeated here. A third-order accurate upwind biased inviscid flux balance is used in both streamwise and normal directions. The finite-volume equivalent of second-order accurate central differences is used for viscous terms. For a typical design analysis of an airfoil, the flow sensitivity coefficient $\left\{\frac{\partial \mathbf{Q}^{\bullet}(\mathbf{X}^{\bullet})}{\partial \mathbf{P}}\right\}$ provides far more information than needed. In most cases, the sensitivity of aerodynamic forces on the surface, such as lift and drag coefficients, are sought. For such analysis, only a small subset of the flow sensitivity coefficient $\left\{\frac{\partial \mathbf{Q}^{\bullet}(\mathbf{X}^{\bullet})}{\partial \mathbf{P}}\right\}$ (i.e. surface properties) is needed since the lift and drag coefficients can be expressed as

$$C_L = C_Y cos\alpha - C_X sin\alpha \tag{4.18}$$

$$C_D = C_Y sin\alpha + C_X cos\alpha \tag{4.19}$$

where α is the flow angle of atack. The quantities C_X and C_Y are the total force coefficents along x and y directions respectively and can be expressed as

$$C_X = \sum_{i=1}^{NE} C_{p_i}(y_{i+1} - y_i) + C_{f_i}(x_{i+1} - x_i) \tag{4.20}$$

$$C_Y = \sum_{i=1}^{NE} C_{p_i}(x_{i+1} - x_i) + C_{f_i}(y_{i+1} - y_i) \tag{4.21}$$

where C_{pi} and C_{f_i} are pressure and skin friction coefficients respectively

$$C_{p_i} = \frac{P_i}{\frac{1}{2}\rho_\infty U_\infty^2} \qquad\qquad C_{f_i} = \frac{\tau_i}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{4.22}$$

and NE represents total number of bondary cells along airfoil surface. The terms P_i and τ_i are pressure and shear stress associated with boundary cell i and the quantity $\frac{1}{2}\rho_\infty U_\infty^2$ is known as dynamic pressure of the free stream . Finally, the drag and lift sensitivity coefficients with respect to $\mathbf{X_D}$ are obtained by differentiating Eqs.(4.18) and (4.19) as

$$\frac{\partial C_L}{\partial \mathbf{X_D}} = \frac{\partial C_Y}{\partial \mathbf{X_D}} cos\alpha - \frac{\partial C_X}{\partial \mathbf{X_D}} sin\alpha \tag{4.23}$$

$$\frac{\partial C_L}{\partial \mathbf{X_D}} = \frac{\partial C_Y}{\partial \mathbf{X_D}} cos\alpha - \frac{\partial C_X}{\partial \mathbf{X_D}} sin\alpha. \tag{4.24}$$

5. RESULTS AND DISCUSSION

5.1 NACA 0012 Airfoil Test Case

5.1.1 Grid Sensitivity

The first test case considered is the NACA 0012 symmetrical airfoil. The previously obtained grid, as shown in Fig. 4.2, is considered for grid sensitivity analysis. The grid sensitivity with respect to the vector of design parameters $\mathbf{X_D}$, are obtained using Eqs. (4.13) and (4.14). The maximum thickness **T** is the only design parameter for this case.

Figure 5.1 shows the contour levels of the y-coordinate sensitivity with respect to the thickness parameter, **T**. The highest contour levels are, understandably, located in the vicinity of the chordwise location for the maximum thickness of the wing section. For a NACA four-digit wing section, this is positioned about 0.3 of the chord length from the leading edge [11]. The positive and negative contour levels corresponding to the upper and lower surfaces are the direct consequence of Eq.(4.4) and the second term on the right hand side of Eq.(4.7). The sensitivity levels decrease when approaching the far-field boundary due to diminishing effects of the interpolation function $\beta_1^0(t, \mathbf{P}_0^7)$. The first term on the right hand side of Eq.(4.7) is responsible for the sensitivity effects due to orthogonality on the surface, and it is directly proportional to the magnitude of the orthogonality vector K_1. The wake region is not sufficiently affected by any of the design parameters, and no major sensitivity gradient should be expected there.

Figure 5.2 demonstrates the contour levels of the x-coordinate sensitivity with respect to thickness parameter, **T**. An interesting observation can be made here regarding the contour levels adjacent to the surface. Unlike the y-coordinate sensitivity, the x-coordinate sensitivity has its lowest value on the surface. This can be traced back to Eq.(4.4), which indicates that x-coordinates on the surface are basically independent of the design parameters. The only remaining factor is the second term on the right hand side of Eq.(4.6) which is the effect of the surface orthogonality vector. There are some negative pockets of contour levels on the forward section and some corresponding positive pockets on the rear section. The dividing line between these pockets is located near the location of the maximum thickness (i.e., 0.3 of chord from leading edge). A simple conclusion from Fig. 5.2 is that by increasing the thickness parameter, **T** points on the forward section will move to the left, while at the same time, points at rear section will move to the right.

For comparison purposes, the grid sensitivity for this case is obtained using the finite difference approach. The design parameter(i.e. **T** for this case) are perturbed, one at a time, and a new grid is obtained using Eqs.(2.5) and (2.6). The sensitivity is then computed using a central difference approximation and the results are presented in Figs. 5.3 and 5.4 . A side by side comparison of both results indicates good agreement between the two approaches.

5.1.2 Flow Sensitivity

The second phase of the problem is obtaining the flow sensitivity coefficients using the previously obtained grid sensitivity coefficients. In order to achieve this, according to Eq. (3.4), a converged flow field solution about a fixed design point should be obtained. The computation is performed on a C-type grid composed of 141 points in the streamwise direction and 31 points in the normal direction. It is apparent that such a coarse grid is inadequate for capturing the full physics of the viscous flow over an airfoil. Therefore, it should be understood that the main objective here is not to produce a highly accurate flow field solution rather than to demonstrate the feasibility of the approach.

A free stream Mach number of $M_\infty = 0.8$, Reynolds number $Re_\infty = 10^6$, and angle of attack $\alpha = 0°$ is used. Figures 5.5 and 5.6 demonstrate the pressure and Mach number contours of the converged solution. Figure 5.7 shows the surface pressure coefficient C_p, where the lift and drag coefficient for this particular example are $C_L = 1.53\text{x}10^{-8}, C_D = 4.82\text{x}10^{-2}$. The sensitivities of the aerodynamic forces, such as drag and lift coefficients with respect to thickness parameter **T**, are obtained utilizing Eqs.(4.18-4.24). The corresponding results are presented in Table 5.1. Again, for comparison purposes, a finite difference approximation has been implemented to validate the results. A nominal perturbation of 10^{-3} for design parameter **T** has been chosen and the corresponding results are included in Table 5.1. The good agreement between the two sets of numbers verifies the accuracy of the approach.

Another important goal of using sensitivity analysis, apart from optimization, is the approximation analysis. An approximate version of Eq.(3.4) can be used to predict the steady-state solution changes which occur in response to geometric shape changes. Such a method is valid as long as the changes in geometric shape (i.e., design parameter) are small. Figure 5.8 shows the non-linear relation between drag coefficient and thickness parameter **T** verifying the above argument.

5.2 NACA 8512 Airfoil Test Case

5.2.1 Grid Sensitivity

The second test case considered is the NACA 8512 cambered airfoil. Again, the previously obtained grid, as shown in Fig. 4.3, is considered for grid sensitivity analysis. Figures 5.9 and 5.10 show the y and x-coordinate sensitivity with respect to parameter **T** respectively. Their characteristics are similar to the previous symmetrical airfoil case; hence, detailed description of their behavior is omitted here.

Figure 5.11 represents the y-coordinate sensitivity with respect to camber, **M**. It appears that the highest sensitivity contour levels are located at the chordwise location of camber, **C** (i.e., 0.5 of chord length). The contour levels decrease toward the far-field boundary, again as a consequence of interpolation function. However, unlike Fig. 5.9 here they possess positive values on both upper and lower surfaces. Consequently, an increase in camber, **M**, shifts all the points upward. Again, minimum activity can be detected in the wake region. Figure 5.12 shows the x-coordinate sensitivity contours with respect to camber, **M**. Here, as in Fig. 5.10, the sensitivities are minimum on the surface of the wing-section. There is a small gradient on the forward section, but by far, the strongest gradient is in the rearward section due to orthogonality effects.

Figure 5.13 illustrates the y-coordinate sensitivity with respect to camber location, **C**. A dividing line between positive and negative contour levels appears near the chordwise position of the camber. Like previous cases, there is no significant activity in the wake region. The result indicates that a positive change of **C** will cause the movement of points downward on the forward section, while at the same time, the points on the rear section will respond by moving upward. Figure 5.14 illustrates the x-coordinate sensitivity with respect to camber location **C**. The two major features are attributed to chordwise location of the camber and the orthogonality effects on the tail section. It is interesting to notice that the sensitivity level for camber location is considerably less than the other two design parameters.

Similar developments can be extended to other grid control parameters such as the distribution of grid point around the wing section and magnitude of orthogonality at the boundaries. For example, the grid sensitivity with respect to stretching parameter $\bar{\mathbf{k}}$, using Eqs.(4.15-4.17), is obtained and the results are presented on Figs. 5.15 and 5.16.

5.2.2 Flow Sensitivity

Using free stream conditions of $M_\infty = 0.8$, $Re_\infty = 10^6$, and $\alpha = 0°$, a converged flow field solution is obtained. As in previous case, a C-type grid of 141x31 is used. Figures 5.17 and 5.18 illustrate the pressure and Mach number contours. Figure 5.19 shows the surface pressure coefficient C_p, where lift and drag coefficients are $C_L = 0.106$, and $C_D = 0.0738$. The aerodynamic sensitivity coefficients with respect to vector of design parameters $\mathbf{X_D}$, are obtained and presented in Table 5.2. A comparison with finite difference validates the accuracy of the approach.

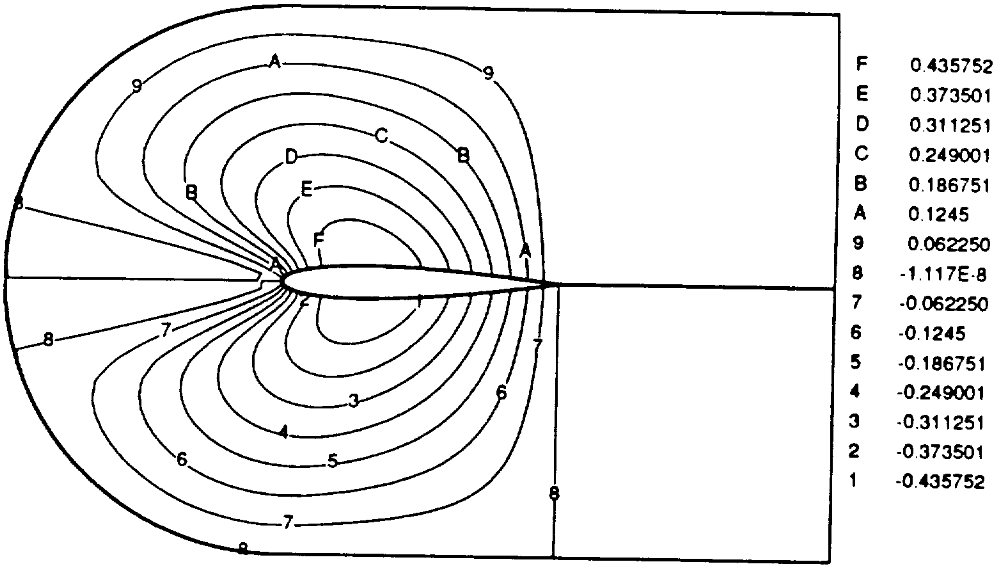

Fig. 5.1 Y-coordinate sensitivity with respect to thickness **T** (Analytical)

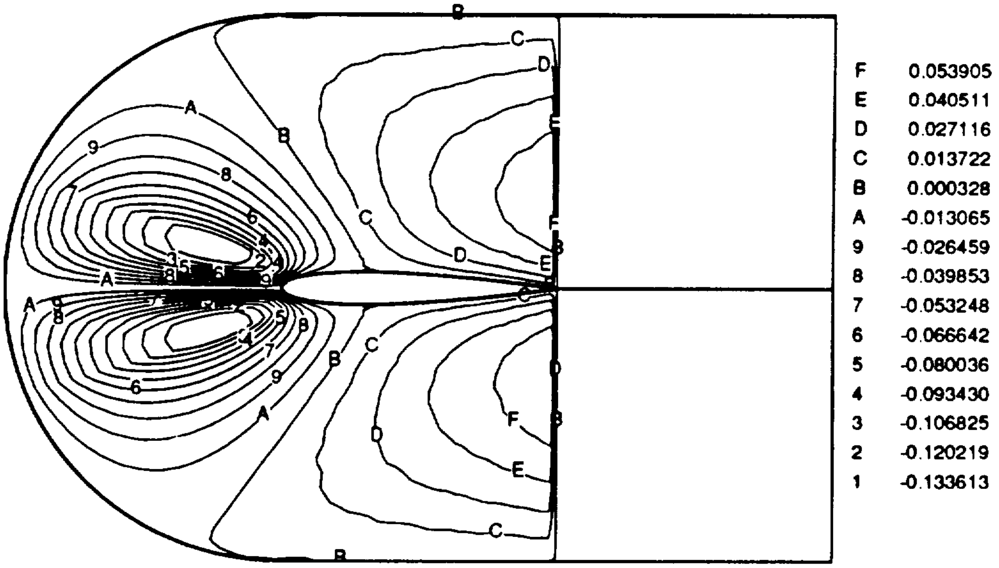

Fig. 5.2 X-coordinate sensitivity with respect to thickness **T** (Analytical)

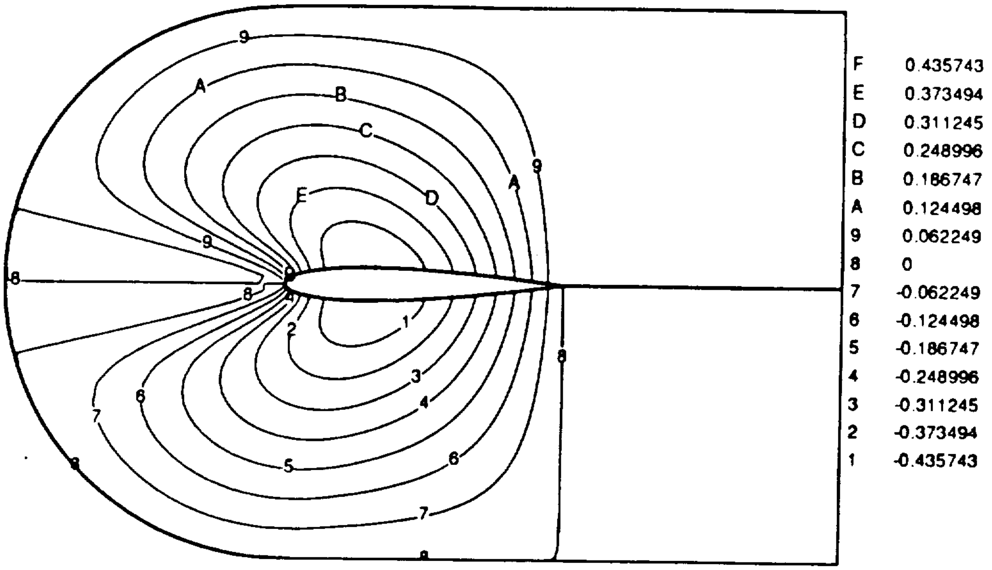

Fig. 5.3 Y-coordinate sensitivity with respect to thickness **T** (Finite-Difference)

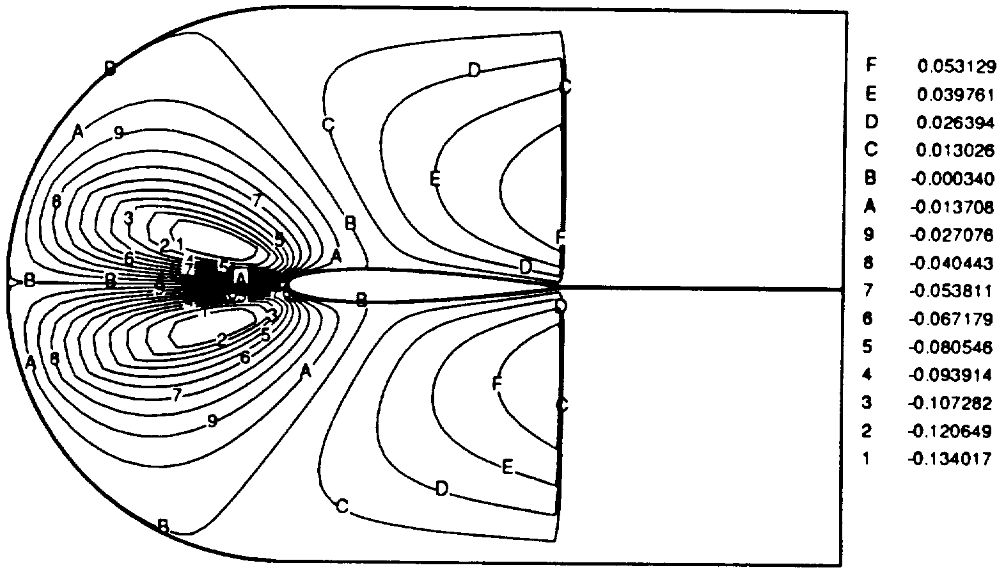

Fig. 5.4 X-coordinate sensitivity with respect to thickness **T** (Finite-Difference)

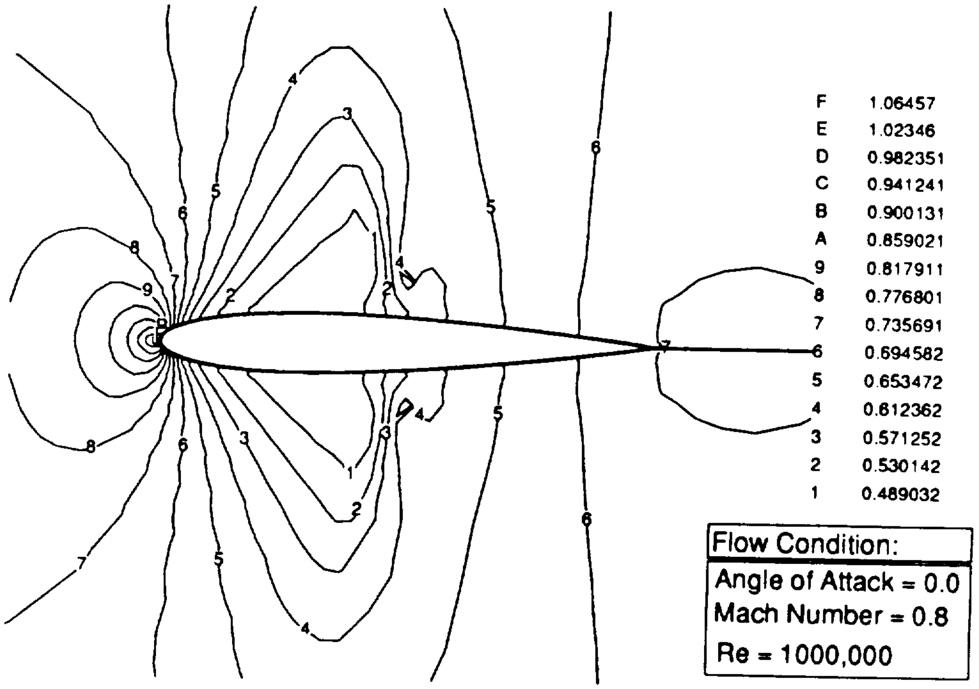

Fig. 5.5 Pressure contours for NACA 0012 wing-section

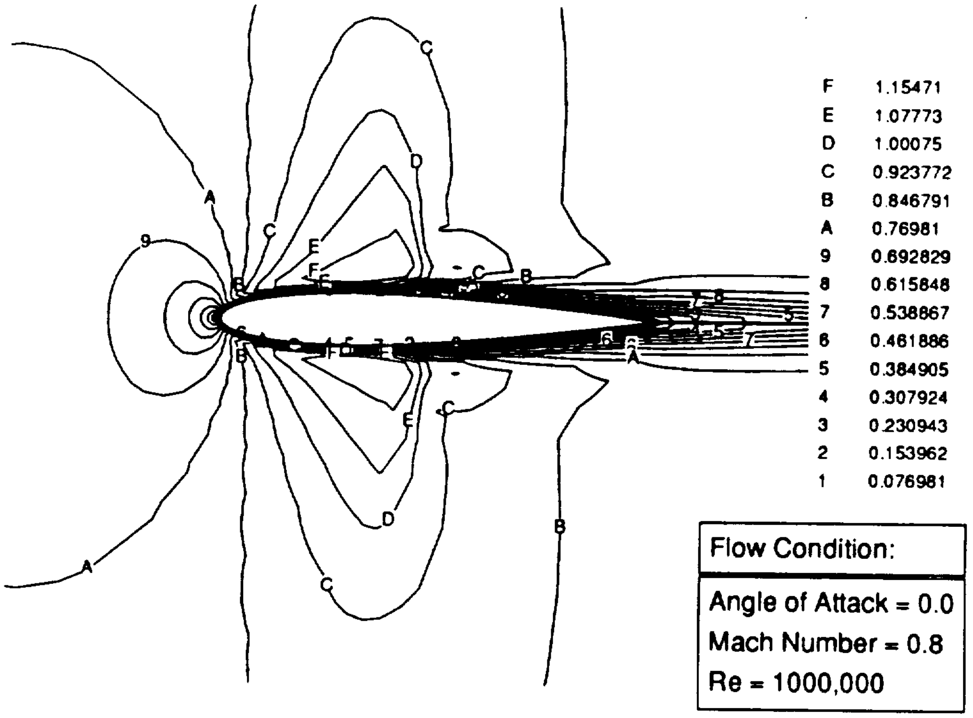

Fig. 5.6 Mach number contours for NACA 0012 wing-section

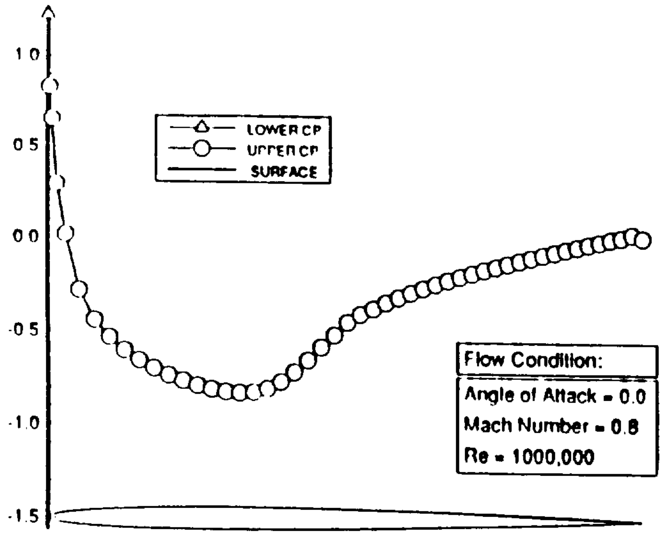

Fig. 5.7 Surface pressure coefficient for NACA 0012 wing-section

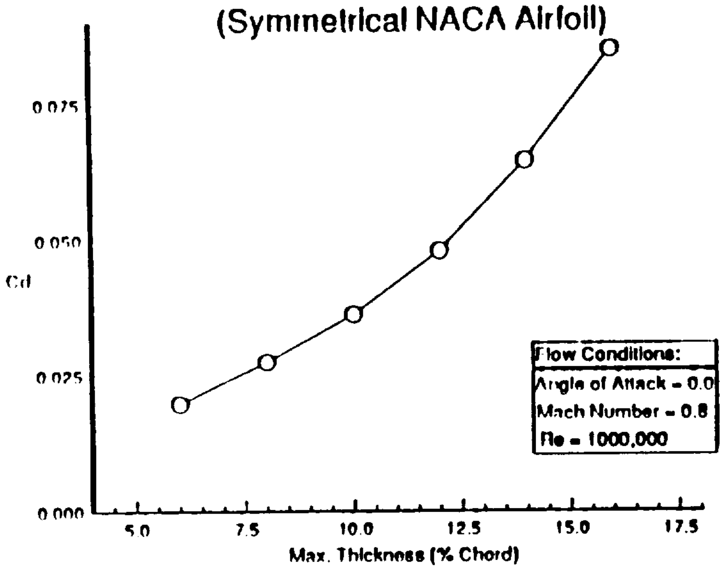

Fig. 5.8 Drag coefficient Vs Maximum Thickness

Table 5.1 Lift and Drag sensitivities with respect to design parameter T

NACA0012	Direct Approach		Finite Difference	
Design Parameters	$\frac{\partial C_L}{\partial X_D}$	$\frac{\partial C_D}{\partial X_D}$	$\frac{\partial C_L}{\partial X_D}$	$\frac{\partial C_D}{\partial X_D}$
X_D=Thickness	$-2.68\text{x}10^{-6}$	0.709	$-4.75\text{x}10^{-6}$	0.707

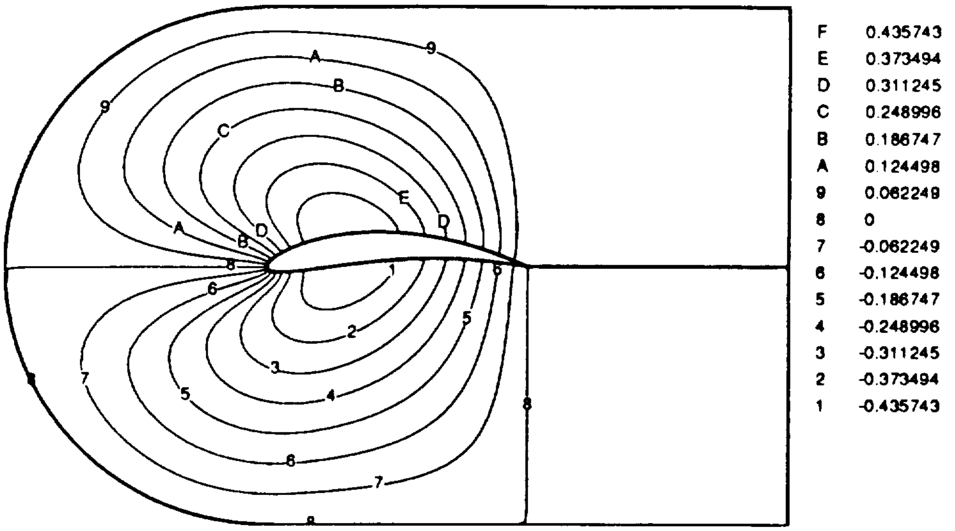

Fig. 5.9 Y-coordinate sensitivity with respect to thickness **T** (Analytical)

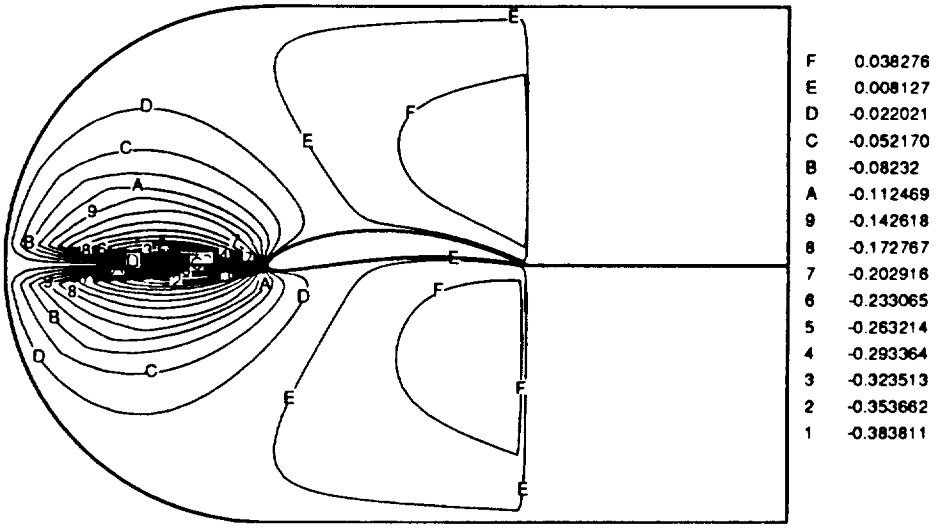

Fig. 5.10 X-coordinate sensitivity with respect to thickness **T** (Analytical)

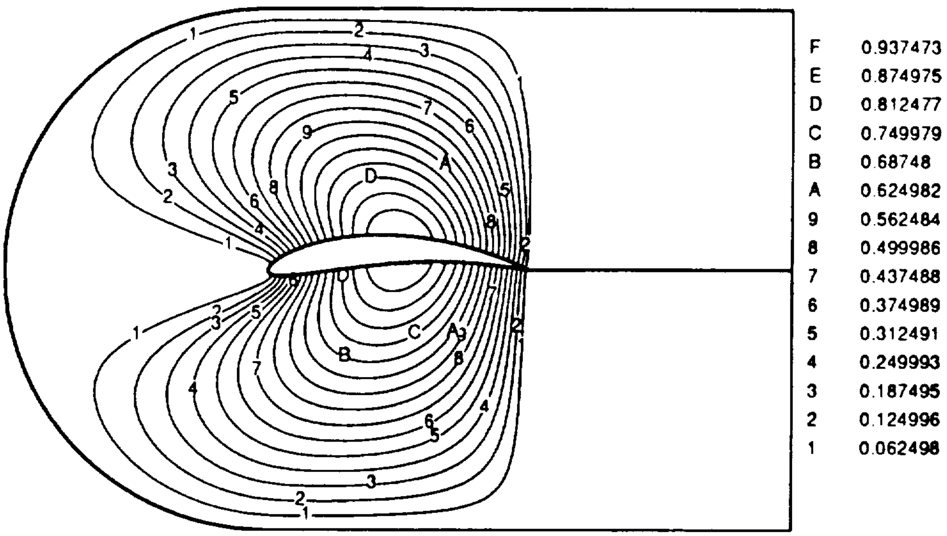

Fig. 5.11 Y-coordinate sensitivity with respect to camber M (Analytical)

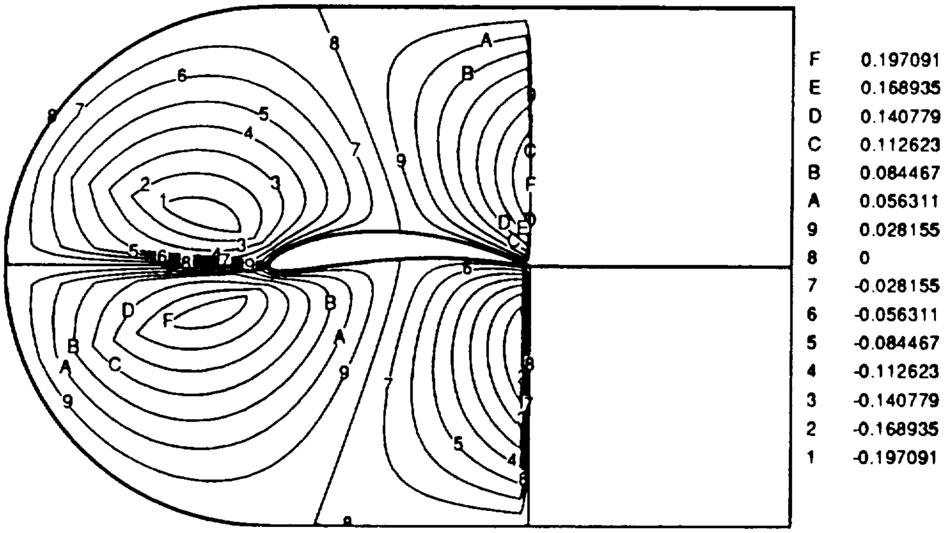

Fig. 5.12 X-coordinate sensitivity with respect to camber M (Analytical)

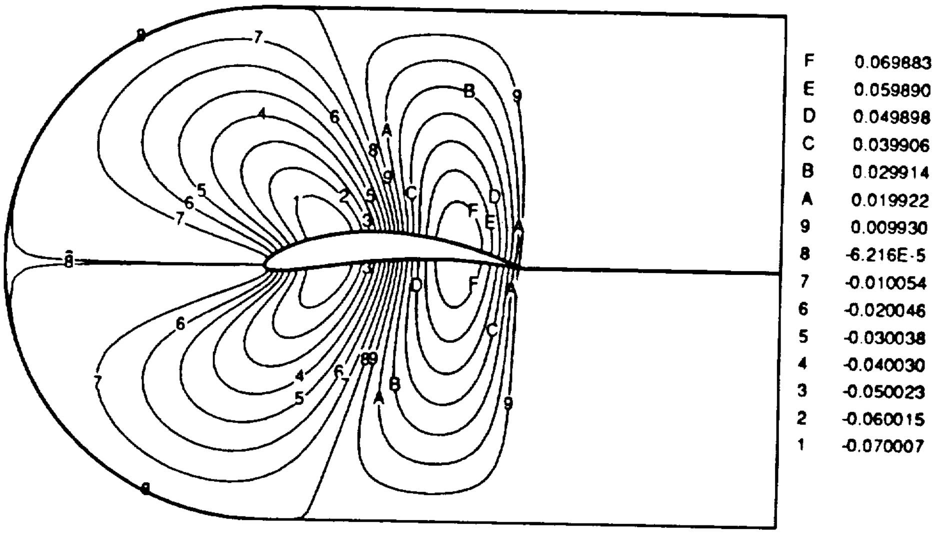

Fig. 5.13 Y-coordinate sensitivity with respect to camber location C (Analytical)

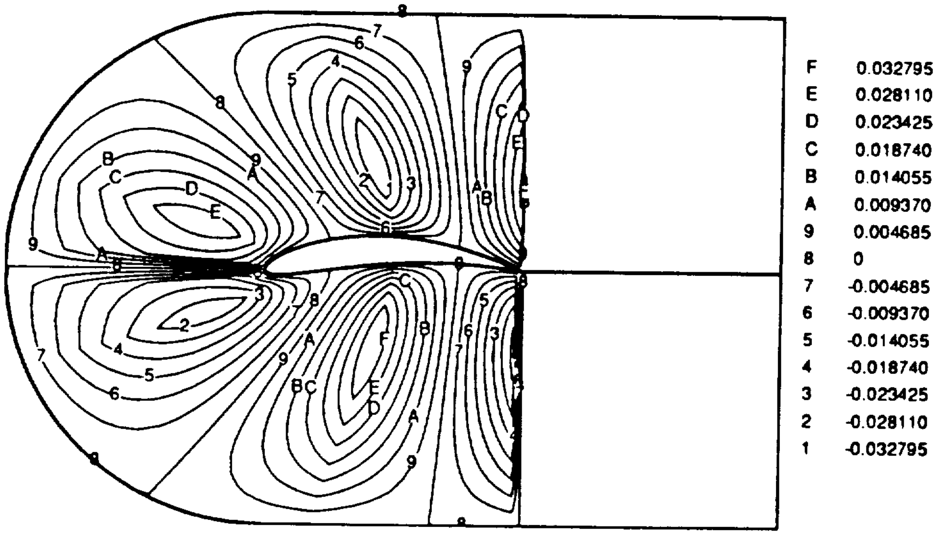

Fig. 5.14 X-coordinate sensitivity with respect to camber location C (Analytical)

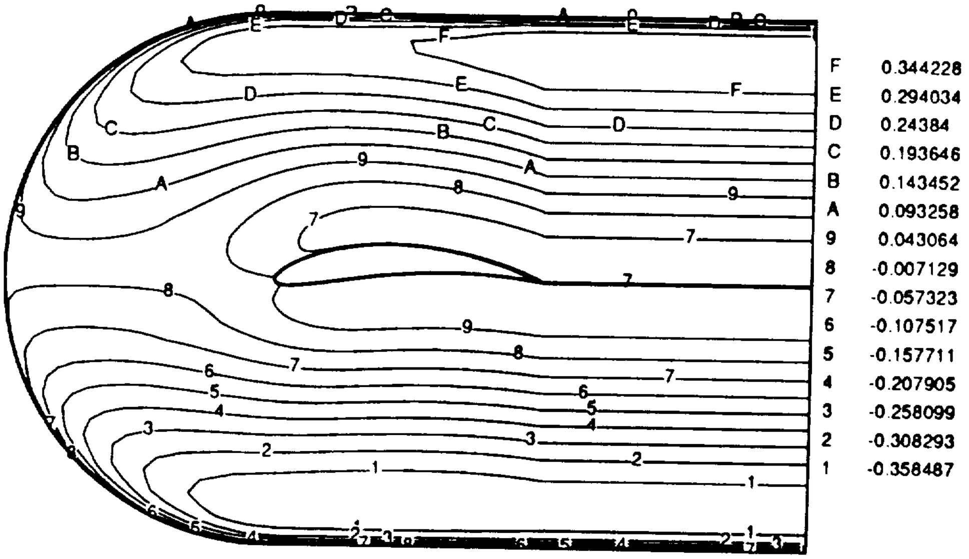

Fig. 5.15 Y-coordinate sensitivity with respect to stretching parameter $\bar{k}$

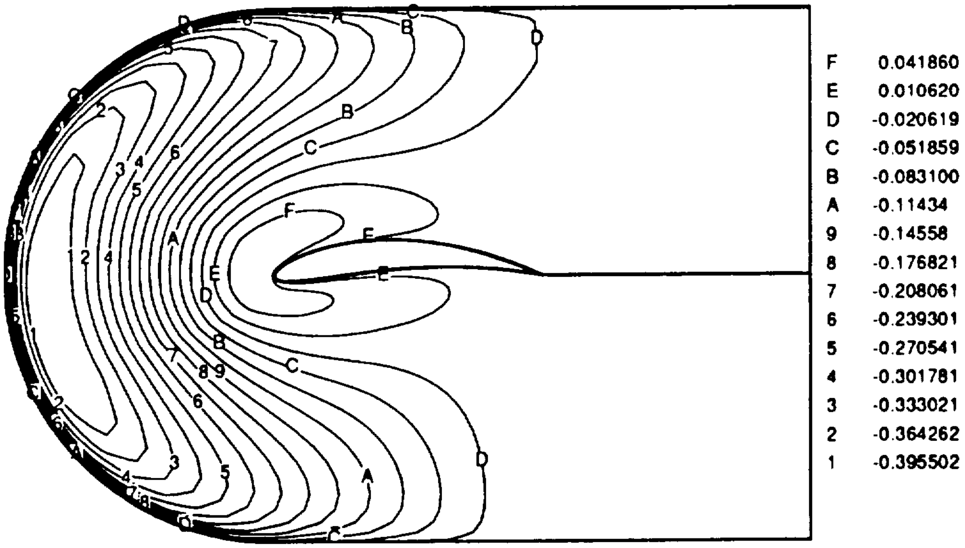

Fig. 5.16 X-coordinate sensitivity with respect to stretching parameter $\bar{k}$

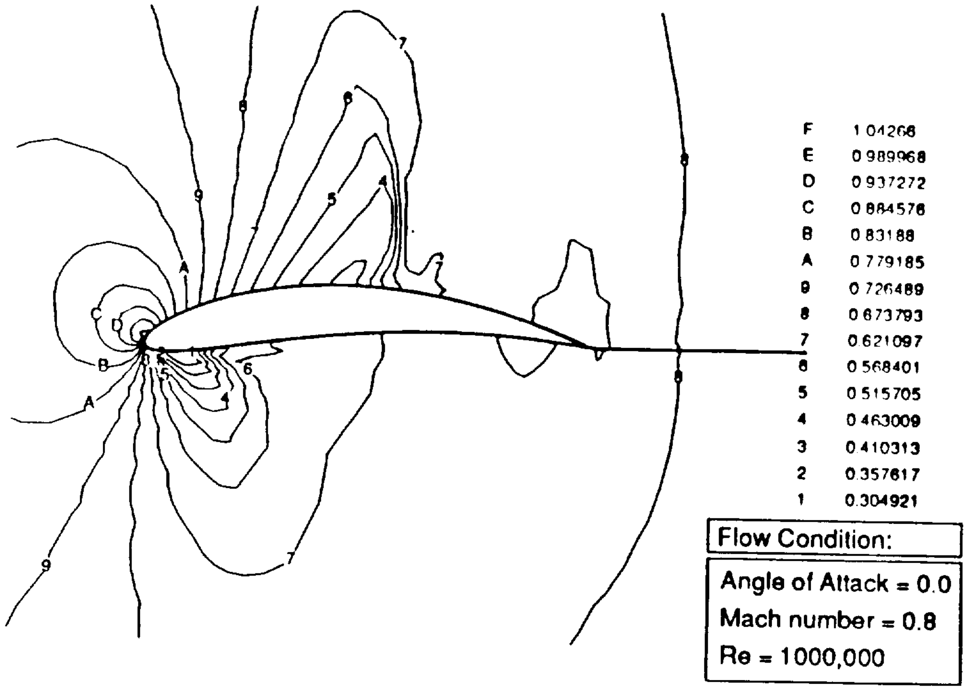

Fig. 5.17 Pressure contours for NACA 8512 wing-section

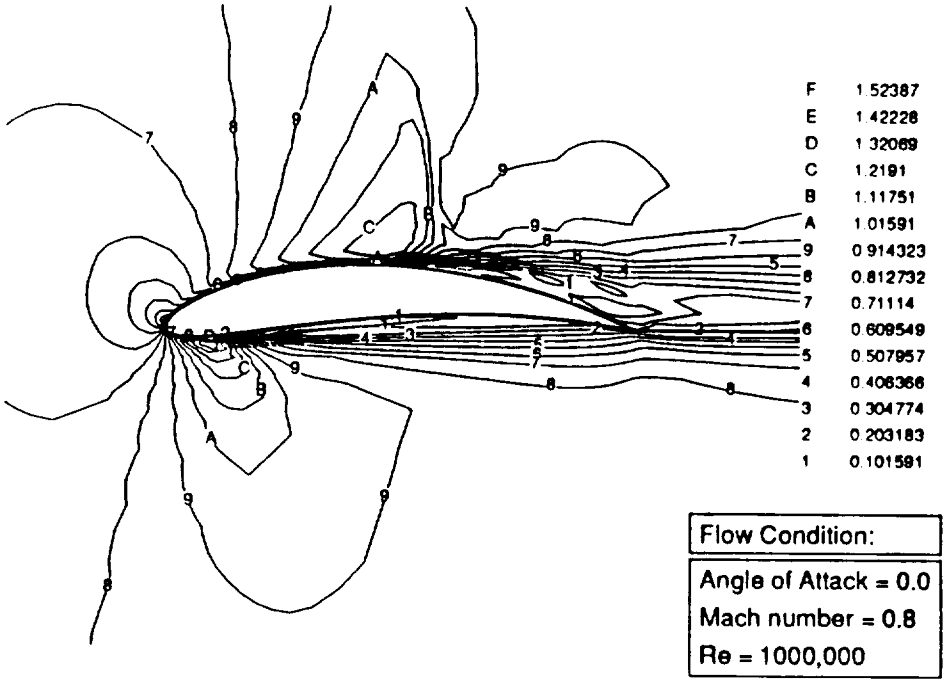

Fig. 5.18 Mach number contours for NACA 8512 wing-section

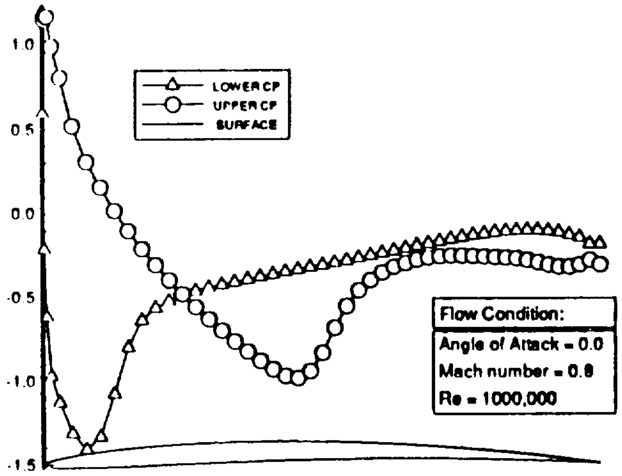

Fig. 5.19 Surface pressure coefficient for NACA 8512 wing-section

Table 5.2 Lift and Drag sensitivities with respect to vector of design parameters $\mathbf{X_D}$

NACA 8512	Direct Approach		Finite Difference	
Design Parameters	$\frac{\partial C_L}{\partial X_D}$	$\frac{\partial C_D}{\partial X_D}$	$\frac{\partial C_L}{\partial X_D}$	$\frac{\partial C_D}{\partial X_D}$
X_D = Thickness	-1.058	0.410	-1.058	0.410
X_D = Camber	7.840	0.322	7.838	0.322
X_D = Camber location	-8.34x10^{-3}	-4.31x10^{-3}	-8.5x10^{-3}	-4.3x10^{-3}

6. CONCLUSION

The objective of this study has been to demonstrate an approach for obtaining grid sensitivity which can be used in aerodynamic design and optimization. It is shown that grid sensitivity is an essential ingredient in the calculation of aerodynamic sensitivity. The main supposition is that a grid is defined algebraically in terms of parameters and computational coordinates. Therefore, coordinates of the grid and derivatives of the coordinates with respect to the parameters (grid sensitivity) are computed directly as functions of the parameters and uniformly-discrete values of the computational coordinates. A subset of the parameters defines the shape of the grid boundaries which corresponds to the aerodynamic surfaces of interest. It is recommended that the aerodynamic surfaces be parameterized in terms of design parameters which have global control. As compared to a geometric parameterization, this drastically reduces the number of parameters. However, it limits the design flexibility. In addition to the aerodynamic surface parameters, the sensitivity with respect to parameters that define other boundaries, such as the far-field boundary or the spacing of grid points, is available for analysis or grid adaptation.

The algebraic grid-generation scheme and NACA wing sections presented here are intended to demonstrate the elements involved in obtaining grid sensitivity from an algebraic grid generation process. It is evident that each grid generation formulation would require considerable analytical differentiation. This implies that a symbolic manipulator, which directly produces computer code for derivative evaluation, should be considered. Also, there are trade-offs between analytical differentiation and finite-difference differentiation. It may be feasible to obtain some of the derivatives by finite differences.

It is implied that airplane surfaces should be parameterized in terms of design variables. This is not simple or feasible for geometrically-complex airplanes in advanced stages of design. However, design parameterization is feasible during conceptional and preliminary design. The parameterization, which is the development of analytical formulas for part or all of a surface is critical for satisfactory results. It is always possible to create a geometric parameterization of a surface (collection of points or derivatives that define surface patches), but geometric parameterizations have very local sensitivities and a complete aerodynamic surface can require a large number of parameters for its definition.

As a compromise between totally analytic parameterization of surfaces and geometric parameterization, a hybrid approach is advocated. In a hybrid approach, certain sections or skeletal parts of a surface are specified analytically and interpolation formulas are used for the remainder of the surface. This is employed for the wing example described herein.

REFRENCES

1. Elbanna, H., and Carlson, L., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," AIAA Paper 89-0532, January 1989.

2. Huband, G. W., Shang, J. S., and Aftosmis, M. J., "Numerical Simulation of an F-16A at Angle of Attack," *Journal of Aircraft*, Vol. 27, pp. 886-892, October 1990.

3. Corcoran, E., "Calculating Reality," *Scientific American*, Vol. 23, pp. 101-109, January 1990.

4. Smith, R. E., and Sadrehaghighi, I., "Grid Sensitivity in Airplane Design," Proceedings of the fourth International Syposium on Computational Fluid - Dynamics, Vol. 1, September 9-12, 1991, Davis, California, pp. 1071-1076.

5. Sobieszczanski-Sobieski, J. "The Case for Aerodynamic Sensitivity Analysis," Paper presented to NASA/VPI SSY Symposium on Sensitivity Analysis in Engineering, September 25-26, 1986.

6. Taylor, A. C., III, Hou, G. W., and Korivi, V. M., "Sensitivity Analysis Applied to the Euler Equations : A Feasibility Study with Emphasis on Variation of Geometric Shape," AIAA Paper 91-0173, January 1991.

7. Eiseman, P. R., and Smith, R. E., "Applications of Algebric Grid Generation , Applications of Mesh Generation to Complex 3-D Configurations," AGARD-CP-464, pp. 4-1-12, 1989.

8. Smith, R. E., and Wiese, M. R., "Interactive Algebraic Grid-Generation Technique," NASA Technical Paper 2533, March 1986.

9. Gordon, W. N., and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Application to Mesh Generation," *Journal of Numerical Methods for Engineers*, Vol. 7, pp 461-477, May 1973.

10. Smith, R. E., and Eriksson, L. E., "Algebraic Grid Generation," *Comp. Meth. Appl. Mech. Eng.*, Vol. 64, pp 285-300 , 1987.

11. Abbott, I. H., and Von Doenhoff, A. E.,Theory of Wing Sections, Dover, NewYork.. 1959.

12. Taylor, A. C., III, Hou, G. W., and Korivi, V. M., " Sensitivity Analysis, and Design Optimization For Internal and External Viscous Flows," AIAA Paper 91-3083, September 1991.

13. Taylor, A. C., III, Hou, G. W., and Korivi, V. M., " An Efficient Method For Estimating Steady-State Numerical Solutions to the Euler Equations," AIAA Paper 91-1680.

14. Taylor, A. C., III, Hou, G. W., and Korivi, V. M., " A Methodology for Determining Aerodynamic Sensitivity Derivatives With Respect to Variation of Geometric Shape," Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, April 8-10, Baltimore, MD, AIAA Paper 91-1101, April 1991.

15. Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations,"Recent Advances in Computational Fluid-Dynamics, (ed. O. Baysal), ASME-FED Vol. 103, 11th Winter Annual Meeting, November, 1990, pp. 191-202.

16. Baysal, O., and Eleshaky, M. E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," AIAA Paper 91-0471, January 1991.

17. Korivi, V. M., " Sensitivity Analysis Applied to the Euler Equations," M.S. Thesis , Old Dominion University, Norfolk, VA, June 1991.

18. Yates, E. C.,Jr., "Aerodynamic Sensitivities from Subsonic,Sonic, and Supersonic Unsteady, Nonplanar Lifting-Surface Theory,"NASA TM-100502 , September 1987.

APPENDIX A

FORTRAN LISTING FOR GRID GENERATION
ALGORITHM : HERMITE

```
      SUBROUTINE HERMITE1(XS,YS,NI,NJ,ALFA,XGRD,YGRD,PROBID,DOBATCH)
      PARAMETER(NGRID=3)
$INCLUDE tbgg1.inc
$INCLUDE tbgg5.inc
      DIMENSION XS(NGDIM,*), YS(NGDIM,*),T(NGDIM),XGRD(NGDIM,NGDIM)
     1          ,YGRD(NGDIM,NGDIM),ALFA(NGDIM),S1(NGDIM,NGDIM)
     2          ,XLEFT(NGDIM),YLEFT(NGDIM),XRIGHT(NGDIM),YRIGHT(NGDIM)
     3          ,RI(NGDIM),RO(NGDIM),R(NGDIM),S(NGDIM),SK(NGDIM)
     4          ,X(NGDIM),Y(NGDIM),XVIEW(NGDIM,NGDIM),YVIEW(NGDIM,NGDIM)
      COMMON/PQ/PS(MGDIM),QS(MGDIM),tS(MGDIM,MGDIM)
      COMMON/DXYDETA/DXBDETA(NGDIM),DYBDETA(NGDIM)
      COMMON/JAYS/J1,J2,J3
      common/design/cm,p,th,rr,wlength,chord
      CHARACTER PROBID*(*),PR(4)*80
      SAVE RO,XLEFT,YLEFT,XRIGHT,YRIGHT
      LOGICAL REDRAW,DONE,REDIST,DOBATCH,FIRST
C
C
C
C-----------------------------------------------------------------------
C  Does basic grid calculations, draws grid, and provides for user
C  modifications in interactive loop.
C-----------------------------------------------------------------------
C
C
c     REDIST = .FALSE.
      REDIST = .TRUE.
c     DONE   = .FALSE.
      DONE   = .TRUE.
C
C ............ Magnitude Of Orthogonality Vector
C
C
               WRITE(*,*)
               WRITE(*,*)' Magnitude of Normal Derivatives ?'
               READ(*,*) SKB
C
                 DO 299 I = 1 , NI
                    SK(I) = SKB
 299             CONTINUE
C
C
 99   CONTINUE
C
C                     Distribution For Stretching Variable T
C              Uses Linear Distribution For Initial Trial and Arc-length
C                        Disribution For Final Trial
C
C
      IF(REDIST.AND.(.NOT.DONE)) THEN
C
        DONE = .TRUE.
C
           IF(DOBATCH) THEN
               CALL BATCH(NJ,RO,5)
                   ELSE
               CALL DISTXI(NJ,RO,5)
           ENDIF
C
               DO 100 I = 1 , NI
```

```
C
                   DO 199 J = 1 , NGDIM
                      X(J) = XGRD(I,J)
                      Y(J) = YGRD(I,J)
 199               CONTINUE
C
                   CALL ARC(X,Y,R,NGDIM,RMAX)
C
                      DO 101 J = 1 , NGDIM
C
                          RI(J) = R(J) / RMAX
                          IF(RI(NGDIM).NE.1.0)RI(NGDIM)=1.0
C
 101                  CONTINUE
C
                      CALL INTERPO(RI,T,NGDIM,RO,S,NJ)
C
                            DO 102 J = 1 , NJ
C
                                S1(I,J) = S(J)
                                tS(I,J) = S1(I,J)
C
 102                        CONTINUE
C
 100            CONTINUE
C
        ELSEIF((.NOT.REDIST).AND.(.NOT.DONE)) THEN
C
              REDIST = .TRUE.
C
C
              DO 201 I = 1 , NI
C
                  DO 201 J = 1 , NGDIM
C
                     T(J) = FLOAT(J-1)/FLOAT(NGDIM-1)
                     S1(I,J) = T(J)
                     tS(I,J) = S1(I,J)
C
 201  CONTINUE
C
C ......... Base Line Distribution
C
        ELSEIF(REDIST.AND.DONE) THEN
C
           IF(DOBATCH) THEN
               CALL BATCH(NJ,RO,5)
                   ELSE
               CALL DISTXI(NJ,RO,5)
           ENDIF
C
           DO 107 I = 1 , NI
           DO 107 J = 1 , NJ
              S1(I,J) = RO(J)
              tS(I,J) = S1(I,J)
 107       CONTINUE
      ENDIF
C
      IF(MGDIM.NE.NGDIM) THEN
         WRITE(*,*)' >>>> tbgg1.inc & tbgg5.inc Should Have the
```

```
     1                            Same Dimension'
      STOP
      ENDIF
C
C         ------------------------------------------------
C          Hermite Interpolation Function (Eq.3 of manual)
C         ------------------------------------------------
C
      SKT = 0.0
C
      DO 240 I=1,NI
C
         XB = XS(I,1)
         YB = YS(I,1)
         XT = XS(I,2)
         YT = YS(I,2)
C
C ........... Orthogonality For Foil
C
         IF(I.GE.1.AND.I.LE.J2) THEN
C
               DXDETA = - SIN(ALFA(I))
               DYDETA = + COS(ALFA(I))
C
         ELSEIF(I.GT.J2.AND.I.LE.NI)THEN
C
               DXDETA = + SIN(ALFA(I))
               DYDETA = - COS(ALFA(I))
C
         ENDIF
C
C
            DXBDETA(I)=DXDETA
            DYBDETA(I)=DYDETA
C
C
         IF(I.EQ.J2-4)SK(I) = SKB * 0.85
         IF(I.EQ.J2-3)SK(I) = SKB * 0.75
         IF(I.EQ.J2-2)SK(I) = SKB * 0.65
         IF(I.EQ.J2-1)SK(I) = SKB * 0.55
         IF(I.EQ.J2)  SK(I) = SKB * 0.45
         IF(I.EQ.J2+1)SK(I) = SKB * 0.55
         IF(I.EQ.J2+2)SK(I) = SKB * 0.65
         IF(I.EQ.J2+3)SK(I) = SKB * 0.75
         IF(I.EQ.J2+4)SK(I) = SKB * 0.85
C
         PS(I) = SK(I)
         QS(I) = SKT
C
       IF(.NOT.DONE)THEN
C
            N=NGDIM
C
               ELSE
C
            N=NJ
C
        ENDIF
C
            DO 240 J = 1 , N
```

```
C
                    CALL BLENDF(S1(I,J),F1,F2,F3,F4)
C
            XGRD(I,J) = XB*F1 + XT*F2 + PS(I) * DXDETA * F3
            YGRD(I,J) = YB*F1 + YT*F2 + PS(I) * DYDETA * F3
C
 240   CONTINUE
C
      IF(.NOT.DONE) GOTO 99
C
C
      RETURN
      END
```

APPENDIX B

FORTRAN LISTING FOR NACA FOUR-DIGIT SURFACE GENERATION : NACA

```
      SUBROUTINE NACA(IS,XS,YS,NLI,ALFA,DOBATCH,CHEAT)
$INCLUDE tbgg1.inc
      PARAMETER(NGRID=3)
      DIMENSION XS(NGDIM,4),YS(NGDIM,4),NLI(4),ALFA(NGDIM)
     1          ,XOU(NGDIM),YOU(NGDIM),XOL(NGDIM),YOL(NGDIM)
     2          ,YC(NGDIM),XCI(NGDIM),YT(NGDIM),DYTDXCI(NGDIM)
     3          ,DYCDXCI(NGDIM),ETA(NGDIM)
      DIMENSION XI(NGDIM),YI(NGDIM),XO(NGDIM),YO(NGDIM),RI(NGDIM)
     1         ,RO(NGDIM),ROBAR(NGDIM),X(NGDIM),XX(NGDIM)
     2         ,XU(NGDIM),YU(NGDIM),XL(NGDIM),YL(NGDIM),XDIST(NGDIM)
      COMMON /A/A1,A2,A3,A4,A5
      COMMON /DESIGN/CM,P,TH,R,WLENGTH,CHORD
      COMMON/TANGENT/ALFAU(NGDIM),ALFAL(NGDIM),DYUDX(NGDIM),DYLDX(NGDIM),NFOIL
      COMMON/OLD/XUS(NGDIM),YUS(NGDIM),XLS(NGDIM),YLS(NGDIM)
      LOGICAL REDIST(3),DOBATCH,CHEAT
      DATA PI,CK/3.14159,7./
C
C
C
C
C
C  ===============================================================
C
C        >>>>  NACA FOUR DIGIT AIRFOIL SECTION  <<<<
C
C      COMPUTES AN AIRFOIL SECTION ANALITICALLY USING THE
C        RELATIONS GIVEN IN 'THEORY OF WING SECTIONS'
C
C
C        WRITTEN BY :  IDEEN  SADREHAGHIGHI
C                      MECHANICAL ENGINEERING & MECHANICS
C                      OLD DOMINION UNIVERSITY
C                      7-12-1991
C  ===============================================================
C
C
C
C --------- INPUT PARAMETERS
C
      CHORD = 1.0
      WRITE(*,*)'              ==============================='
      WRITE(*,*)'              >>>>>> AIRFOIL GEOMETRY <<<<<<'
      WRITE(*,*)'              ==============================='
      WRITE(*,*)
      WRITE(*,*)'NUMBER OF POINTS IN XI-DIRECTION?'
      READ(*,*)NI
      NFOIL=NI
      WRITE(*,*)'NUMBER OF POINTS IN ETA-DIRECTION?'
      READ(*,*)NJ
      WRITE(*,*)'OUTER BOUNDARY LOCATION (CHORD LENGTH)?'
      READ(*,*)R
      R=CHORD*R
      YLENGTH=R
      PERCENT=CHORD/100.
      TENTH=CHORD/10.
      WRITE(*,*)'MAXIMUM ORDINATE OF MEAN LINE OR CAMBER (% CHORD) ?'
      READ(*,*) CM
      CM=CM*PERCENT
      WRITE(*,*)' CHORDWISE POSITION OF CAMBER (./CHORD)?'
      READ(*,*) P
      P=P*TENTH
      WRITE(*,*)' MAXIMUM THICKNESS OF AIRFOIL ( % CHORD) ?'
      READ(*,*) TH
      TH=TH*PERCENT
C
C-----------------------------------------------------------------
C
C     AIRFOIL NOMENCLATURE:
C
C     CM ............... CAMBER
C     P ................ CAMBER LOCATION ALONG CHORD
C     TH ............... MAXIMUM THICKNESS
C     A ................ THICKNESS DISTRIBUTION COEFFICIENTS
C     XCI , YC ......... MEAN LINE COORDINATES
C     CHORD ............ CHORD LENGTH
C     WLENGTH .......... WAKE LENGTH
C
C-----------------------------------------------------------------
C
C
C --------- THICKNESS DISTRIBUTION
C
           A1=0.29648
           A2=-0.12642
           A3=-0.35202
           A4=0.28388
           A5=-0.10192
C
C
C .........INITIAL  DISTRIBUTION IN XI-DIRECTION FOR LOWER BOUNDARY
C
      DO 5 I=1,NGDIM
         C = FLOAT(I-1)/FLOAT(NGDIM-1)
         X(I) = (EXP(CK*C)-1.)/(EXP(CK)-1.)
         XCI(I) = X(I) * CHORD
 5    CONTINUE
C
C .........GET INITIAL FOIL (UPPER & LOWER)
C
      CALL FOIL(XCI,XUS,YUS,ALFAU,DYUDX,NGDIM,1,CHEAT,999)
      CALL FOIL(XCI,XLS,YLS,ALFAL,DYLDX,NGDIM,2,CHEAT,999)
C
C --------- OUTER BOUNDARY
C
      XOUTER = CHORD+R
C
C
C ......... INITIAL DISTRIBUTION IN XI-DIRECTION FOR OUTER BOUNDARY
C
        DO 35 I=1,NI
C
        C = FLOAT(I-1)/FLOAT(NI-1)
        XX(I) = (EXP(C*CK)-1.)/(EXP(CK)-1.)
C
 35     CONTINUE
C
        DO 36 I=1,NI
           XOL(I) = XX(I) * XOUTER
C
           XOU(I)=XOL(I)
```

```
             IF(XOL(I).LE.R) THEN
                YOU(I)=SQRT(R*R-(XOL(I)-R)*(XOL(I)-R))
                YOL(I)=-YOU(I)
                   ELSE
                YOU(I)=YLENGTH
                YOL(I)=-YLENGTH
             END IF
36      CONTINUE
C
C ----------- OUTPUT TO HERMITE
C
C   NLI(J) = REPRESENTS NUMBER OF POINTS IN EACH BOUNDARY
C   J=1 ........  SOLID BOUNDARY
C   J=2 ........  OUTER BOUNDARY
C   J=3 ........  RIGHT BOUNDARY
C   J=4 ........  LEFT BOUNDARY
C

      NLI(1) = NI*2
      NLI(2) = NI*2
      NLI(3) = NJ
      NLI(4) = NJ
C
C ..................... BOTTOM BOUNDARY DISTRIBUTION (ARC-LENGTH)
C
      J=1
C
c     REDIST(J) = .TRUE.
      REDIST(J) = .FALSE.
C
      IF(REDIST(J)) THEN
C
C ................INTERPOLATE  UPPER PORTION
C
         CALL ARC(XUS,YUS,RI,NGDIM,RMAX)
         IF(DOBATCH) THEN
            CALL BATCH(NI,ROBAR,1)
               ELSE
            CALL DISTXI(NI,ROBAR,IS)
         ENDIF
            DO 100 I=1,NI
                RO(I) = ROBAR(I) * RMAX
 100        CONTINUE
         CALL INTERPO(RI,XUS,NGDIM,RO,XDIST,NI)
C
C ............ GET FINAL FOIL (UPPER)
C
         CALL FOIL(XDIST,XU,YU,ALFAU,DYUDX,NI,1,CHEAT,999)
C
C ................INTERPOLATE  LOWER PORTION
C
         CALL ARC(XLS,YLS,RI,NGDIM,RMAX)
         IF(DOBATCH) THEN
            CALL BATCH(NI,ROBAR,2)
               ELSE
            CALL DISTXI(NI,ROBAR,IS)
         ENDIF
            DO 101 I=1,NI
                RO(I) = ROBAR(I) * RMAX
 101        CONTINUE
```

```
         CALL INTERPO(RI,XLS,NGDIM,RO,XDIST,NI)
C
C ............. GET FINAL FOIL (LOWER)
C
         CALL FOIL(XDIST,XL,YL,ALFAL,DYLDX,NI,2,CHEAT,999)
C
      ELSE
C
C  ............... Base Line Distribution
C
         IF(DOBATCH) THEN
           CALL BATCH(NI,ROBAR,1)
             ELSE
           CALL DISTXI(NI,ROBAR,IS)
         ENDIF
         CALL FOIL(ROBAR,XU,YU,ALFAU,DYUDX,NI,1,CHEAT,999)
         IF(DOBATCH) THEN
           CALL BATCH(NI,ROBAR,2)
             ELSE
           CALL DISTXI(NI,ROBAR,IS)
         ENDIF
         CALL FOIL(ROBAR,XL,YL,ALFAL,DYLDX,NI,2,CHEAT,999)
      ENDIF
C
C .......... SHIFT X-COORDINATES TO THE RIGHT FOR C-TYPE GRID
C
      DO 30 I=1,NI
         XU(I) = XU(I) + R
         XL(I) = XL(I) + R
 30   CONTINUE
C
C ................ ASSEMBLE COUNTER-CLOCKWISE
C
      DO 40 I=1,NLI(J)
         IF(I.LE.NI) THEN
            XS(I,J)=XU(NI-I+1)
            YS(I,J)=YU(NI-I+1)
            ALFA(I)=ALFAU(NI-I+1)
            ALFA(NI)=PI/2.
               ELSE
            XS(I,J)=XL(I-NI)
            YS(I,J)=YL(I-NI)
            ALFA(I)=ALFAL(I-NI)
         END IF
40    CONTINUE
C
C ................ CHECK DOUBLE POINTS FOR BOTTOM BOUNDARY
C
      DO 46 I=1,NLI(J)-1
         IF(XS(I,J).EQ.XS(I+1,J).AND.YS(I,J).EQ.YS(I+1,J)) THEN
             K=I+1
             XS(K,J)=XS(K+1,J)
             YS(K,J)=YS(K+1,J)
             ALFA(K)=ALFA(K+1)
                IF(I.EQ.NLI(J)-1)NLI(J)=NLI(J)-1
         END IF
46    CONTINUE
C
C ---------- OUTER BOUNDARY
C
```

```
      J=2
C
      REDIST(J) = .TRUE.
      DO 45 I=1,NLI(J)
         IF(I.LE.NI) THEN
            XI(I)=XOU(NI-I+1)
            YI(I)=YOU(NI-I+1)
               ELSE
            XI(I)=XOL(I-NI)
            YI(I)=YOL(I-NI)
         END IF
45    CONTINUE
C
C ----------- CHECK FOR DOUBLE POINTS (SYMMETRY LINE)
C
      DO 48 I=1,NLI(J)-1
         IF(XI(I).EQ.XI(I+1).AND.YI(I).EQ.YI(I+1)) THEN
             K=I+1
             XI(K)=XI(K+1)
             YI(K)=YI(K+1)
             IF(I.EQ.NLI(J)-1) NLI(J)=NLI(J)-1
         END IF
48    CONTINUE
C
      IF(REDIST(J)) THEN
         CALL ARC(XI,YI,RI,NLI(J),RMAX)
         IF(DOBATCH) THEN
             CALL BATCH(NLI(J),ROBAR,6)
                 ELSE
             CALL DISTXI(NLI(J),ROBAR,IS)
         ENDIF
            DO 103 I=1,NLI(J)
                RO(I) = ROBAR(I) * RMAX
 103        CONTINUE
         CALL INTERPO(RI,XI,NLI(J),RO,XO,NLI(J))
         CALL INTERPO(RI,YI,NLI(J),RO,YO,NLI(J))
      ENDIF
C
      DO 201 I=1,NLI(J)
         IF(REDIST(J))THEN
            XS(I,J) = XO(I)
            YS(I,J) = YO(I)
               ELSEIF(.NOT.REDIST(J))THEN
            XS(I,J) = XI(I)
            YS(I,J) = YI(I)
         ENDIF
 201  CONTINUE
C
C ....... ETA DISTIBUTION
C
      DO 51 J = 1 , NJ
         ETA(J) = FLOAT(J-1)/FLOAT(NJ-1)
 51   CONTINUE
C
C ----------- RIGHT BOUNDARY
C
      J=3
      DO 49 I=1,NJ
      XS(I,J)=CHORD+R
      YS(I,J)=ETA(I)*R
 49   CONTINUE
C
C ----------- LEFT BOUNDARY
C
      J=4
      DO 50 I=1,NJ
      XS(I,J)=CHORD+R
      YS(I,J)=-ETA(I)*R
 50   CONTINUE
C
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE FOIL(XCI,X,Y,ANGLE,DYDX,N,II,CHEAT,IGRID)
$INCLUDE tbgg1.inc
      DIMENSION  XCI(NGDIM),X(NGDIM),Y(NGDIM),YC(NGDIM),YT(NGDIM),XD(3)
     1          ,DYCDXCI(NGDIM),DYTDXCI(NGDIM),ANGLE(NGDIM),DYDX(NGDIM)
      COMMON /A/A1,A2,A3,A4,A5
      COMMON /DESIGN/CM,P,TH,R,WLENGTH,CHORD
      COMMON /DELTAXD/TH1,CM1,P1,TH3,CM3,P3
      LOGICAL CHEAT
      DATA PI,CK/3.14159,3./
C
C
C
      XD(1) = TH
      XD(2) = CM
      XD(3) = P
C
      IF(CHEAT)THEN
        IF(IGRID.EQ.1)THEN
           XD(1) = TH1
           XD(2) = CM1
           XD(3) = P1
        ELSEIF(IGRID.EQ.3)THEN
           XD(1) = TH3
           XD(2) = CM3
           XD(3) = P3
        ENDIF
      ENDIF
C
      DO 10 I=1,N
C
          IF(XD(3).NE.0.0.AND.XCI(I).LE.XD(3)) THEN
C
            YC(I)=(XD(2)/(XD(3)*XD(3)))*(2.*XD(3)*XCI(I)-XCI(I)*XCI(I))
            DYCDXCI(I)=(XD(2)/(XD(3)*XD(3)))*(2.*XD(3)-2.*XCI(I))
C
            ELSEIF(XD(3).NE.0.0.AND.XCI(I).GT.XD(3))THEN
C
            YC(I)=(XD(2)/((1.-XD(3))*(1.-XD(3))))*
     1                     (1.-2.*XD(3)+2.*XD(3)*XCI(I)-XCI(I)*XCI(I))
            DYCDXCI(I)=(XD(2)/((1.-XD(3))*(1.-XD(3))))*(2.*XD(3)-2.*XCI(I))
C
            ELSEIF(XD(3).EQ.0.0)THEN
C
              YC(I)=0.0
              DYCDXCI(I) = 0.0
C
          END IF
10    CONTINUE
C
C
      DO 15 I=1,N
C
        IF(XCI(I).LE.CHORD) THEN
C
           YT(I)=(XD(1)/0.2)*
     1       (A1*SQRT(XCI(I))+A2*XCI(I)+A3*XCI(I)*XCI(I)
     2       +A4*XCI(I)*XCI(I)*XCI(I)+A5*XCI(I)*XCI(I)*XCI(I)*XCI(I))
C
           DYTDXCI(I)=(XD(1)/0.2)*(A1*(0.5/SQRT(XCI(I)))+A2+2.*A3*XCI(I)
     1            +3.*A4*XCI(I)*XCI(I)+4.*A5*XCI(I)*XCI(I)*XCI(I))
C
                      ELSE
C
                    YT(I)=0.0
                    DYTDXCI(I) =0.0
C
        END IF
15    CONTINUE
C
C
C ----------- SURFACE COORDINATES (EQS. 5,6)
C
      DO 25 I = 1,N
C
         X(I) = XCI(I)
         IF(II.EQ.1) THEN
           Y(I) = YC(I) + YT(I)
           DYDX(I)  = DYCDXCI(I) + DYTDXCI(I)
           ANGLE(I)  = ATAN(DYDX(I))
           ANGLE(1)  = PI/2
             ELSEIF(II.EQ.2)THEN
           Y(I) = YC(I) - YT(I)
           DYDX(I)  = DYCDXCI(I) - DYTDXCI(I)
           ANGLE(I)  = ATAN(DYDX(I))
           ANGLE(1)  = PI/2
             ELSE
           WRITE(*,*)' Trouble in FOIL'
           STOP
         ENDIF
C
25    CONTINUE
C
      RETURN
      END
```

APPENDIX C

FORTRAN LISTING OF NACA FOUR-DIGIT GRID SENSITIVITY : SENSIT

```
      SUBROUTINE DXYDXYD(XX,YY,NI,NJ,ICUR,FLAG,ISENS,SRAN)
      PARAMETER(ICHOICE=8)
$INCLUDE tbgg1.inc
$INCLUDE tbgg5.inc
      DIMENSION XX(NGDIM,NGDIM),YY(NGDIM,NGDIM)
      COMMON/XYDXYD/ DXDTH(NGDIM,NGDIM),DYDTH(NGDIM,NGDIM),
     1               DXDCM(NGDIM,NGDIM),DYDCM(NGDIM,NGDIM),
     2               DXDP(NGDIM,NGDIM),DYDP(NGDIM,NGDIM)
      COMMON /DXYBDES/DXBDCM(NGDIM),DYBDCM(NGDIM),DXBDTH(NGDIM),
     1          DYBDTH(NGDIM),DXBDP(NGDIM),DYBDP(NGDIM),DXBEDTH(NGDIM),
     2          DYBEDTH(NGDIM),DXBEDCM(NGDIM),DYBEDCM(NGDIM),DXBEDP(NGDIM),
     3          DYBEDP(NGDIM)
      COMMON/PQ/P(MGDIM),Q(MGDIM),T(MGDIM,MGDIM)
      COMMON/JAYS/J1,J2,J3
      LOGICAL FLAG(ICHOICE)
C
C
C
C
C         ---------------------------------------------------
C          GRID SENSITIVITY WITH RESPECT TO DESIGN VARIABLES
C
C             WRITTEN BY : IDEEN  SADREHAGHIGHI
C         ---------------------------------------------------
C
C ................... Get Bottom Boundary Sensitivity
C
           CALL DXYBDD(XX,YY,NI,NJ,ICUR,SRAN,CHEAT)
C
C ...... Sensitivity Without Arc-length Distribution For Normal Direction
C
      DO 10 J = 1 , NJ
C
         DO 10 I = 1 , NI
C
            CALL BLENDF(T(I,J),ALFA1,ALFA2,ALFA3,ALFA4)
C
C
            DXDTH(I,J)  = ALFA1 * DXBDTH(I)  + P(I)  * ALFA3 * DXBEDTH(I)
            DYDTH(I,J)  = ALFA1 * DYBDTH(I)  + P(I)  * ALFA3 * DYBEDTH(I)
            DXDCM(I,J)  = ALFA1 * DXBDCM(I)  + P(I)  * ALFA3 * DXBEDCM(I)
            DYDCM(I,J)  = ALFA1 * DYBDCM(I)  + P(I)  * ALFA3 * DYBEDCM(I)
            DXDP  (I,J) = ALFA1 * DXBDP(I)   + P(I)  * ALFA3 * DXBEDP( I)
            DYDP  (I,J) = ALFA1 * DYBDP(I)   + P(I)  * ALFA3 * DYBEDP( I)
C
 10   CONTINUE
C
      FLAG(ISENS) = .TRUE.
C
      RETURN
      END
C
C*******************************************************************
C
      SUBROUTINE DXYBDD(XX,YY,NI,NJ,ICUR,SRAN,CHEAT)
      PARAMETER(ICHOICE=8)
$INCLUDE tbgg1.inc
      DIMENSION XX(NGDIM,NGDIM),YY(NGDIM,NGDIM),
     1          DXBUDTH(NGDIM),DXBUDCM(NGDIM),DXBUDP(NGDIM),
     2          DYBUDTH(NGDIM),DYBUDCM(NGDIM),DYBUDP(NGDIM),
     3           DXBLDTH(NGDIM),DXBLDCM(NGDIM),DXBLDP(NGDIM),
     4           DYBLDTH(NGDIM),DYBLDCM(NGDIM),DYBLDP(NGDIM),
     5           DXUEDTH(NGDIM),DXUEDCM(NGDIM),DXUEDP(NGDIM),
     6           DYUEDTH(NGDIM),DYUEDCM(NGDIM),DYUEDP(NGDIM),
     7           DXLEDTH(NGDIM),DXLEDCM(NGDIM),DXLEDP(NGDIM),
     8           DYLEDTH(NGDIM),DYLEDCM(NGDIM),DYLEDP(NGDIM),
     9           XU(NGDIM),XL(NGDIM),YU(NGDIM),YL(NGDIM)
      COMMON /DXYBDES/DXBDCM(NGDIM),DYBDCM(NGDIM),DXBDTH(NGDIM),
     1         DYBDTH(NGDIM),DXBDP(NGDIM),DYBDP(NGDIM),DXBEDTH(NGDIM),
     2         DYBEDTH(NGDIM),DXBEDCM(NGDIM),DYBEDCM(NGDIM),DXBEDP(NGDIM),
     3         DYBEDP(NGDIM)
      COMMON/JAYS/J1,J2,J3
      COMMON/TANGENT/THETAU(NGDIM),THETAL(NGDIM),DYUDX(NGDIM),DYLDX(NGDIM),NFOIL
      COMMON /A/A1,A2,A3,A4,A5
      COMMON /DESIGN/CM,P,TH,R,WLENGTH,CHORD
      COMMON /TEMP1/XNEW(NGDIM),YNEW(NGDIM),RNEW(NGDIM)
      LOGICAL CHEAT
C
C
C
C    ================================================================
C     COMPUTES ANALITICALLY THE DERIVATIVE OF BOUNDARY COORDINATES
C            WITH RESPECT TO DESIGN VARIABLES DXB/DXD
C
C                <<<  DESIGN VARIBLES >>>
C
C                    CM ....... CAMBER
C                    P ........ LOCATION OF CAMBER
C                    TH ....... MAX. THICKNESS
C
C    ================================================================
C
C
C
C
C
C
      IUPPER = J2
      ILOWER = NI - J2 + 1
      IF(IUPPER.NE.ILOWER)THEN
        WRITE(*,*)' ERROR FROM DXYBDD'
        STOP
      ENDIF
C
      DO 1 I = 1 , IUPPER
         XU(I) = XX(J2-I+1,1) - R
         YU(I) = YY(J2-I+1,1)
 1    CONTINUE
C
      DO 2 I = 1 , ILOWER
         XL(I) = XX(I+J2-1,1) - R
         YL(I) = YY(I+J2-1,1)
 2    CONTINUE
C
C
C           ----------------------------------------------------
C             BOUNDARY DESIGN SENSITIVITY DERIVATIVES OF AIRFOIL
C           ----------------------------------------------------
C
C
```

```
C
      IDUMMY = J2 - J1
      DO 16 I = 1,IUPPER
C
      IF(I.LE.IDUMMY) THEN
C
C ---------------------- CAMBERED AIRFOIL ------------------------
C
C .................... FORWARD OF MAX. CAMBER
C
         IF(P.NE.0.0.AND.XU(I).LE.P)THEN
C
            YC = (CM/(P*P))*(2.*P*XU(I)-XU(I)*XU(I))
            DYCDCM = (1./(P*P))*(2.*P*XU(I)-XU(I)*XU(I))
            DYCDP  = (2.*CM/(P*P))*(-XU(I)+XU(I)*XU(I)/P)
            D2YCDCM= (1./(P*P))*(2.*P-2.*XU(I))
            D2YCDP = (2.*CM/(P*P))*(2.*XU(I)/P-1.)
C
         ELSEIF(P.NE.0.0.AND.XU(I).GT.P)THEN
C
C .................... AFT OF MAX. CAMBER
C
         YC = (CM/((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XU(I)-
     .                  XU(I)*XU(I))
         DYCDCM = (1./((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XU(I)
     .                   -XU(I)*XU(I))
         FACTOR1 = CM/((1.-P)*(1.-P))
         FACTOR2 = 2./(1.-P)
         FACTOR3 = 1.-2.*P+2.*P*XU(I)-XU(I)*XU(I)
         DYCDP  = FACTOR1 * (FACTOR2 * FACTOR3 - 2. + 2.*XU(I))
         D2YCDCM= (1./((1.-P)*(1.-P)))*(2.*P-2.*XU(I))
         D2YCDP = (2.*CM/((1.-P)*(1.-P)))*(2.*((P-XU(I))/(1.-P))+1.)
C
        ELSEIF(CM.EQ.0.0.AND.P.EQ.0.0) THEN
C
C -------------------- SYMMETRICAL AIRFOIL ------------------------
C
                         YC      = 0.0
                         DYCDCM = 0.0
                         DYCDP  = 0.0
                         D2YCDCM= 0.0
                         D2YCDP = 0.0
C
         ENDIF
C
C
         D2YTDTH = (1./0.2)*(0.5*A1/SQRT(XU(I))+A2
     .    +2.*A3*XU(I)+3.*A4*XU(I)*XU(I)+4.*A5*XU(I)*XU(I)*XU(I))
C
         FACTOR = A1*SQRT(XU(I))+A2*XU(I)+A3*XU(I)*XU(I)
     .    +A4*XU(I)*XU(I)*XU(I)+A5*XU(I)*XU(I)*XU(I)*XU(I)
C
              YT = (TH/0.2)*FACTOR
C
              DYDTH = YT/TH
              DYBUDTH(I) = DYDTH
              DYBUDCM(I) = DYCDCM
              DYBUDP(I)  = DYCDP
C
C .............. EVALUATING DXETA/DTH , DYETA/DTH
```

```
C
C
               TIMESU      = COS(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DXUEDTH(I)  = + D2YTDTH * - TIMESU
C
               TIMESU      = SIN(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DYUEDTH(I)  = + D2YTDTH * - TIMESU
C
C .............. EVALUATING DXETA/DCM , DYETA/DCM
C
               TIMESU      = COS(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DXUEDCM(I)  = + D2YCDCM * - TIMESU
C
               TIMESU      = SIN(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DYUEDCM(I)  = + D2YCDCM * - TIMESU
C
C .............. EVALUATING DXETA/DP , DYETA/DP
C
               TIMESU      = COS(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DXUEDP(I)   = + D2YCDP * - TIMESU
C
               TIMESU      = SIN(THETAU(I))/(1.+(DYUDX(I)*DYUDX(I)))
               DYUEDP(I)   = + D2YCDP * - TIMESU
C
C ............... Singularity At Nose .... Slope dy/dx = Infinite
C
               IF(I.EQ.1)THEN
                 DXUEDTH(I) = 0.0
                 DYUEDTH(I) = 0.0
                 DXUEDCM(I) = 0.0
                 DYUEDCM(I) = 0.0
                 DXUEDP (I) = 0.0
                 DYUEDP (I) = 0.0
               ENDIF
C
            ELSE
C
C ......... SET BOUNDARY SENSITIVITY DERIVATIVES TO ZERO IN WAKE REGION
C
                  DXBUDCM(I) = 0.0
                  DYBUDCM(I) = 0.0
                  DXBUDP (I) = 0.0
                  DYBUDP (I) = 0.0
                  DXBUDTH(I) = 0.0
                  DYBUDTH(I) = 0.0
                  DXUEDTH(I) = 0.0
                  DYUEDTH(I) = 0.0
                  DXUEDCM(I) = 0.0
                  DYUEDCM(I) = 0.0
                  DXUEDP (I) = 0.0
                  DYUEDP (I) = 0.0
C
      ENDIF
C
16    CONTINUE
C
C
      IL = J2 - J1 + 1
C
C ......... GET X-BOUNDARY COORDINATE SENSITIVITY FOR UPPER WAKE REGION
```

```
C        ....... DXB/DXD = DXB/DR * DR/DXD .........
C
      CALL DXBDXD(1,IL,XU,YU,DXBUDCM,DXBUDTH,DXBUDP,CHEAT)
C
      IDUMMY = J3 - J2
      DO 17 I = 1,ILOWER
C
      IF(I.LE.IDUMMY) THEN
C
C ---------------------- CAMBERED AIRFOIL ------------------------
C
C .................... FORWARD OF MAX. CAMBER
C
         IF(P.NE.0.0.AND.XL(I).LE.P)THEN
            YC = (CM/(P*P))*(2.*P*XL(I)-XL(I)*XL(I))
            DYCDCM = (1./(P*P))*(2.*P*XL(I)-XL(I)*XL(I))
            DYCDP  = (2.*CM/(P*P))*(-XL(I)+XL(I)*XL(I)/P)
            D2YCDCM= (1./(P*P))*(2.*P-2.*XL(I))
            D2YCDP = (2.*CM/(P*P))*(2.*XL(I)/P-1.)
C
         ELSEIF(P.NE.0.0.AND.XL(I).GT.P)THEN
C
C .................... AFT OF MAX. CAMBER
C
         YC = (CM/((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XL(I)-
     .                  XL(I)*XL(I))
         DYCDCM = (1./((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XL(I)
     .                   -XL(I)*XL(I))
         FACTOR1 = CM/((1.-P)*(1.-P))
         FACTOR2 = 2./(1.-P)
         FACTOR3 = 1.-2.*P+2.*P*XL(I)-XL(I)*XL(I)
         DYCDP   = FACTOR1 * (FACTOR2 * FACTOR3 - 2. + 2.*XL(I))
         D2YCDCM= (1./((1.-P)*(1.-P)))*(2.*P-2.*XL(I))
         D2YCDP = (2.*CM/((1.-P)*(1.-P)))*(2.*((P-XL(I))/(1.-P))+1.)
C
         ELSEIF(CM.EQ.0.0.AND.P.EQ.0.0) THEN
C
C -------------------- SYMMETRICAL AIRFOIL ----------------------
C
                          YC      = 0.0
                          DYCDCM = 0.0
                          DYCDP  = 0.0
                          D2YCDCM= 0.0
                          D2YCDP = 0.0
C
         ENDIF
C
C
         D2YTDTH = (1./0.2)*(0.5*A1/SQRT(XL(I))+A2
     .           +2.*A3*XL(I)+3.*A4*XL(I)*XL(I)+4.*A5*XL(I)*XL(I)*XL(I))
C
         FACTOR = A1*SQRT(XL(I))+A2*XL(I)+A3*XL(I)*XL(I)
     .    +A4*XL(I)*XL(I)*XL(I)+A5*XL(I)*XL(I)*XL(I)*XL(I)
C
              YT = (TH/0.2)*FACTOR
C
              DYDTH = YT/TH
              DYBLDTH(I) = - DYDTH
              DYBLDCM(I) = DYCDCM
              DYBLDP(I)  = DYCDP
```

```
C
C .............. EVALUATING DXETA/DTH , DYETA/DTH
C
C
               TIMESL      = COS(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DXLEDTH(I) = - D2YTDTH * + TIMESL
C
               TIMESL      = SIN(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DYLEDTH(I) = - D2YTDTH * + TIMESL
C
C .............. EVALUATING DXETA/DCM , DYETA/DCM
C
               TIMESL      = COS(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DXLEDCM(I) = + D2YCDCM * + TIMESL
C
               TIMESL      = SIN(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DYLEDCM(I) = + D2YCDCM * + TIMESL
C
C .............. EVALUATING DXETA/DP , DYETA/DP
C
               TIMESL      = COS(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DXLEDP(I)   = + D2YCDP * + TIMESL
C
               TIMESL      = SIN(THETAL(I))/(1.+(DYLDX(I)*DYLDX(I)))
               DYLEDP(I)   = + D2YCDP * + TIMESL
C
C ............... Singularity At Nose .... Slope dy/dx = Infinite
C
               IF(I.EQ.1)THEN
                 DXLEDTH(I) = 0.0
                 DYLEDTH(I) = 0.0
                 DXLEDCM(I) = 0.0
                 DYLEDCM(I) = 0.0
                 DXLEDP  (I) = 0.0
                 DYLEDP  (I) = 0.0
               ENDIF
C
             ELSE
C
C ........ SET SENSITIVITY DERIVATIVES TO ZERO IN WAKE REGION
C
                  DXBLDCM(I) = 0.0
                  DYBLDCM(I) = 0.0
                  DXBLDP  (I) = 0.0
                  DYBLDP  (I) = 0.0
                  DXBLDTH(I) = 0.0
                  DYBLDTH(I) = 0.0
                  DXLEDTH(I) = 0.0
                  DYLEDTH(I) = 0.0
                  DXLEDCM(I) = 0.0
                  DYLEDCM(I) = 0.0
                  DXLEDP  (I) = 0.0
                  DYLEDP  (I) = 0.0
C
      ENDIF
C
17    CONTINUE
C
C ......... GET X-BOUNDARY COORDINATE SENSITIVITY FOR LOWER WAKE REGION
C        ....... DXB/DXD = DXB/DR * DR/DXD .........
```

```
C
      CALL DXBDXD(2,IL,XL,YL,DXBLDCM,DXBLDTH,DXBLDP,CHEAT)
C
      DO 50 I = 1,IUPPER
C
          DXBDTH(IUPPER - I + 1) = DXBUDTH(I)
          DYBDTH(IUPPER - I + 1) = DYBUDTH(I)
          DXBDCM(IUPPER - I + 1) = DXBUDCM(I)
          DYBDCM(IUPPER - I + 1) = DYBUDCM(I)
          DXBDP  (IUPPER - I + 1) = DXBUDP  (I)
          DYBDP  (IUPPER - I + 1) = DYBUDP  (I)
          DXBEDTH(IUPPER- I + 1) = DXUEDTH(I)
          DYBEDTH(IUPPER- I + 1) = DYUEDTH(I)
          DXBEDCM(IUPPER- I + 1) = DXUEDCM(I)
          DYBEDCM(IUPPER- I + 1) = DYUEDCM(I)
          DXBEDP(IUPPER - I + 1) = DXUEDP  (I)
          DYBEDP(IUPPER - I + 1) = DYUEDP  (I)
C
50    CONTINUE
C
      DO 60 I = 1,ILOWER
C
          DXBDTH(IUPPER - 1 + I) = DXBLDTH(I)
          DYBDTH(IUPPER - 1 + I) = DYBLDTH(I)
          DXBDCM(IUPPER - 1 + I) = DXBLDCM(I)
          DYBDCM(IUPPER - 1 + I) = DYBLDCM(I)
          DXBDP  (IUPPER - 1 + I) = DXBLDP  (I)
          DYBDP  (IUPPER - 1 + I) = DYBLDP  (I)
          DXBEDTH(IUPPER- 1 + I) = DXLEDTH(I)
          DYBEDTH(IUPPER- 1 + I) = DYLEDTH(I)
          DXBEDCM(IUPPER- 1 + I) = DXLEDCM(I)
          DYBEDCM(IUPPER- 1 + I) = DYLEDCM(I)
          DXBEDP(IUPPER - 1 + I) = DXLEDP  (I)
          DYBEDP(IUPPER - 1 + I) = DYLEDP  (I)
C
60    CONTINUE
C
      RETURN
      END
C
C***********************************************************************
C
      SUBROUTINE DXBDXD(II,IL,XNEW,YNEW,DXBDCM,DXBDTH,DXBDP,CHEAT)
$INCLUDE tbgg1.inc
      DIMENSION XNEW(NGDIM),YNEW(NGDIM),DXBDCM(NGDIM),DXBDTH(NGDIM)
     1         ,DXBDP(NGDIM),ROLD(NGDIM),RNEW(NGDIM),XOLD(NGDIM)
     2         ,YOLD(NGDIM),DYDTH(NGDIM),DYDCM(NGDIM),DYDP(NGDIM)
     3         ,DRDCM(NGDIM),DRDTH(NGDIM),DRDP(NGDIM),DRDCMN(NGDIM)
     4         ,DRDTHN(NGDIM),DRDPN(NGDIM)
      COMMON/OLD/XUS(NGDIM),YUS(NGDIM),XLS(NGDIM),YLS(NGDIM)
      COMMON/TANGENT/THETAU(NGDIM),THETAL(NGDIM),DYUDX(NGDIM),DYLDX(NGDIM),NFOIL
      COMMON/A/A1,A2,A3,A4,A5
      COMMON /DESIGN/CM,P,TH,R,WLENGTH,CHORD
      COMMON /DELTAXD/TH1,CM1,P1,TH3,CM3,P3
      COMMON/DUMMY/X(NGDIM),Y(NGDIM),X1(NGDIM),X3(NGDIM),RO(NGDIM),RI(NGDIM),
     1             XDIST(NGDIM),DUM1(NGDIM),DUM2(NGDIM),XCI(NGDIM),ROBAR(NGDIM)
      LOGICAL CHEAT
      DATA CK,DELTA/4.0,0.0001/
C
C
```

```
C
      goto 999
      DO 10 I = 1 , NGDIM
C
         IF(II.EQ.1)THEN
C
             XOLD(I)=XUS(I)
             YOLD(I)=YUS(I)
C
           ELSE
C
             XOLD(I)=XLS(I)
             YOLD(I)=YLS(I)
C
         ENDIF
C
 10     CONTINUE
C
C
C
      DO 15 I = 1,NGDIM
C
       IF(XOLD(I).LE.CHORD)THEN
C
C --------------------- CAMBERED AIRFOIL -----------------------
C
C ................... FORWARD OF MAX. CAMBER
C
         IF(P.NE.0.0.AND.XOLD(I).LE.P)THEN
            YC = (CM/(P*P))*(2.*P*XOLD(I)-XOLD(I)*XOLD(I))
            DYCDCM = (1./(P*P))*(2.*P*XOLD(I)-XOLD(I)*XOLD(I))
            DYCDP  = (2.*CM/(P*P))*(-XOLD(I)+XOLD(I)*XOLD(I)/P)
C
            ELSEIF(P.NE.0.0.AND.XOLD(I).GT.P)THEN
C
C ................... AFT OF MAX. CAMBER
C
         YC = (CM/((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XOLD(I)-
     .                  XOLD(I)*XOLD(I))
         DYCDCM = (1./((1.-P)*(1.-P)))*(1.-2.*P+2.*P*XOLD(I)
     .                   -XOLD(I)*XOLD(I))
         FACTOR1 = CM/((1.-P)*(1.-P))
         FACTOR2 = 2./(1.-P)
         FACTOR3 = 1.-2.*P+2.*P*XOLD(I)-XOLD(I)*XOLD(I)
         DYCDP  = FACTOR1 * (FACTOR2 * FACTOR3 - 2. + 2.*XOLD(I))
C
           ELSEIF(CM.EQ.0.0.AND.P.EQ.0.0) THEN
C
C -------------------- SYMMETRICAL AIRFOIL ----------------------
C
                         YC     = 0.0
                         DYCDCM = 0.0
                         DYCDP  = 0.0
C
         ENDIF
C
           ELSE
C
             WRITE(*,*)'ERROR FROM DXBDXD..... XOLD IS OUT OF DOMAIN'
             STOP
```

```
C
         ENDIF
C
C
         FACTOR = A1*SQRT(XOLD(I))+A2*XOLD(I)+A3*XOLD(I)*XOLD(I)
     .   +A4*XOLD(I)*XOLD(I)*XOLD(I)+A5*XOLD(I)*XOLD(I)*XOLD(I)*XOLD(I)
C
              YT = (TH/0.2)*FACTOR
C
              IF(II.EQ.2)YT=-YT
C
              DYDTH(I) = YT/TH
              DYDCM(I) = DYCDCM
              DYDP(I)  = DYCDP
C
 15   CONTINUE
C
C
      CALL ARC(XOLD,YOLD,ROLD,NGDIM,RMAX)
      CALL ARC(XNEW,YNEW,RNEW,IL   ,RMAX)
C
C
      DRDCM(1)=0.0
      DRDTH(1)=0.0
      DRDP (1)=0.0
C
      DO 20 I = 2 , NGDIM
C
       FACTOR   = (YOLD(I) - YOLD(I-1))/(ROLD(I)-ROLD(I-1))
C
           DRDCM(I) = FACTOR * (DYDCM(I)-DYDCM(I-1)) + DRDCM(I-1)
           DRDTH(I) = FACTOR * (DYDTH(I)-DYDTH(I-1)) + DRDTH(I-1)
           DRDP (I) = FACTOR * (DYDP (I)-DYDP (I-1)) + DRDP (I-1)
C
 20   CONTINUE
C
C
      CALL INTERPO(ROLD,DRDCM,NGDIM,RNEW,DRDCMN,IL)
      CALL INTERPO(ROLD,DRDTH,NGDIM,RNEW,DRDTHN,IL)
      CALL INTERPO(ROLD,DRDP ,NGDIM,RNEW,DRDPN ,IL)
C
C
      DO 25 I = 1 , IL
C
        IF(I.EQ.1) THEN
             DXBDR = (XNEW(I+1)-XNEW(I))/(RNEW(I+1)-RNEW(I))
        ELSEIF(I.EQ.IL)THEN
             DXBDR = (XNEW(I)-XNEW(I-1))/(RNEW(I)-RNEW(I-1))
        ELSE
C
        DXBDR = (XNEW(I+1)-XNEW(I-1))/(RNEW(I+1)-RNEW(I-1))
C
        ENDIF
C
        IF(II.EQ.1)DYDX=DYUDX(I)
        IF(II.EQ.2)DYDX=DYLDX(I)
        FACTOR=SQRT(1. + DYDX*DYDX)
        DXBDR=1./FACTOR
C
        DXBDCM(I) = DXBDR * DRDCMN(I)
        DXBDTH(I) = DXBDR * DRDTHN(I)
        DXBDP (I) = DXBDR * DRDPN (I)
C
 25   CONTINUE
C
999   CONTINUE
      goto 1099
      IF(CHEAT) THEN
C
           TH1 = TH + DELTA
           TH3 = TH - DELTA
           IF(CM.NE.0.0)THEN
           CM1 = CM + DELTA
           CM3 = CM - DELTA
           ELSE
           CM1=0.0
           CM3=0.0
           ENDIF
           IF(P.NE.0.0)THEN
           P1  = P  + DELTA
           P3  = P  - DELTA
           ELSE
           P1=0.0
           P3=0.0
           ENDIF
C
         IGRID = 1
 1000    CONTINUE
C
         DO 35 I=1,NGDIM
C
            C = FLOAT(I-1)/FLOAT(NGDIM-1)
            XCI(I) = ((EXP(CK*C)-1.)/(EXP(CK)-1.))*CHORD
C
 35      CONTINUE
C
         CALL FOIL(XCI,X,Y,DUM1,DUM2,NGDIM,II,CHEAT,IGRID)
         CALL ARC(X,Y,RI,NGDIM,RMAX)
         CALL BATCH(IL,ROBAR,1)
C
         DO 45 I=1,IL
            RO(I)=ROBAR(I)*RMAX
 45      CONTINUE
C
         CALL INTERPO(RI,X,NGDIM,RO,XDIST,IL)
         CALL FOIL(XDIST,X,Y,DUM1,DUM2,IL,II,CHEAT,IGRID)
C
            IF(IGRID.EQ.1)THEN
                 DO 65 I=1,IL
                 X1(I) = X(I)
 65              CONTINUE
                 IGRID = 3
                 GOTO 1000
            ELSEIF(IGRID.EQ.3)THEN
                 DO 85 I=1,IL
                 X3(I) = X(I)
 85              CONTINUE
            ENDIF
C
C
```

```
C     ........ Finite Differencing
C
         DELTATH = TH1 - TH3
         DELTACM = CM1 - CM3
         DELTAP  = P1  - P3
C
         DO 75 I=1,IL
            DXBDTH(I) = (X1(I) - X3(I))/(TH1 - TH3)
            IF(DELTACM.NE.0.0)THEN
            DXBDCM(I) = (X1(I) -X3(I))/(CM1 - CM3)
            ELSE
            DXBDCM(I) = 0.0
            ENDIF
            IF(DELTAP.NE.0.0)THEN
            DXBDP(I) = (X1(I) - X3(I))/(P1 - P3)
            ELSE
            DXBDP(I) = 0.0
            ENDIF
 75      CONTINUE
C
      ENDIF
C
 1099 continue
      do 2000 I=1,IL
         DXBDTH(I) = 0.0
         DXBDCM(I) = 0.0
         DXBDP (I) = 0.0
 2000 CONTINUE
      RETURN
      END
```