AERODYNAMIC DESIGN SENSITIVITIES ON AN UNSTRUCTURED MESH USING THE NAVIER-STOKES EQUATIONS AND A DISCRETE ADJOINT FORMULATION

By

Eric J. Nielsen

A DISSERTATION SUBMITTED TO THE FACULTY OF

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

AEROSPACE ENGINEERING

APPROVED:

Bernard Grossman, Chair

W. Kyle Anderson

Robert W. Walters

Joseph A. Schetz

William H. Mason

December 1998

Blacksburg, Virginia

Keywords: Adjoint, Aerodynamics, Design, Navier-Stokes, Sensitivity, Unstructured

Copyright 1998, Eric J. Nielsen

AERODYNAMIC DESIGN SENSITIVITIES ON AN UNSTRUCTURED MESH USING THE NAVIER-STOKES EQUATIONS AND A DISCRETE ADJOINT FORMULATION

Eric J. Nielsen

(ABSTRACT)

A discrete adjoint method is developed and demonstrated for aerodynamic design optimization on unstructured grids. The governing equations are the three-dimensional Reynolds-averaged Navier-Stokes equations coupled with a one-equation turbulence model. A discussion of the numerical implementation of the flow and adjoint equations is presented. Both compressible and incompressible solvers are differentiated, and the accuracy of the sensitivity derivatives is verified by comparing with gradients obtained using finite differences and a complex-variable approach. Several simplifying approximations to the complete linearization of the residual are also presented. A first-order approximation to the dependent variables is implemented in the adjoint and design equations, and the effect of a “frozen” eddy viscosity and neglecting mesh sensitivity terms is also examined. The resulting derivatives from these approximations are all shown to be inaccurate and often of incorrect sign. However, a partially-converged adjoint solution is shown to be sufficient for computing accurate sensitivity derivatives, yielding a potentially large cost savings in the design process. The convergence rate of the adjoint solver is compared to that of the flow solver. For inviscid adjoint solutions, the cost is roughly one to four times that of a flow solution, whereas for turbulent computations, this ratio can reach as high as ten. Sample optimizations are performed for inviscid and turbulent transonic flows over an ONERA M6 wing, and drag reductions are demonstrated.

For my Mom and Dad

Acknowledgments

First and foremost, I would like to thank Dr. W. Kyle Anderson for his time and effort over the past six years. He has spent countless hours supporting my education, whether through teaching me fundamentals, or simply helping to track down coding problems from time to time. Without his willingness to take time out for my questions and problems, this work would surely not have been a success. In addition, I would like to express my appreciation to Dr. Chris Rumsey for introducing me to the world of computational fluid dynamics and providing me with my first basic lessons in the field. My academic advisors, Drs. Bernard Grossman and Robert Walters, have been helpful in a number of ways, and I owe them my thanks as well. Finally, I would like to thank the head of the Aerodynamic and Acoustic Methods Branch at NASA Langley Research Center, Dr. James Thomas. I am grateful for his financial support as well as simply allowing me to work alongside experts in the field on a daily basis ever since my undergraduate years. The experience and encouragement he has provided have given me an invaluable foundation for a professional career.

Table of Contents

List of Tables

List of Figures

Nomenclature

A	Coefficient matrix
a	Speed of sound
B	Coefficient for Spalding's boundary layer profile
b	Right hand side vector
$C*$	Sutherland's constant
C_l, C_d	Lift and drag coefficients
C_l^*, C_d^*	Target lift and drag coefficients
$\bar{C}_{l_{trans}}$	Ratio of lift derivatives for translation
c_{t_3}, c_{t_4}	Constants for turbulence model
c_{b_1}, c_{b_2}, c_{v_1}	Constants for turbulence model
c_{w_1}, c_{w_2}, c_{w_3}	Constants for turbulence model
$\boldsymbol{D}$	Vector of design variables
d	Distance to the nearest surface
E	Total energy per unit volume
$\vec{\boldsymbol{F}}_i$	Inviscid flux vector
$\vec{\boldsymbol{F}}_v$	Viscous flux vector
f	Cost function
$\boldsymbol{f}_v$, $\boldsymbol{g}_v$, $\boldsymbol{h}_v$	Components of viscous fluxes
f_{v_1}, f_{v_2}	Functions for turbulence model
f_w, f_{t_1}, f_{t_2}	Functions for turbulence model
g	Function for turbulence model
H_k	Upper Hessenberg matrix
h	Step size for finite difference, also entries in H_k
$\boldsymbol{I}$	Identity matrix
$\hat{\boldsymbol{i}}$, $\hat{\boldsymbol{j}}$, $\hat{\boldsymbol{k}}$	Unit vectors in Cartesian coordinate directions

i	$\sqrt{-1}$
K_{ij}	Spring constant for mesh movement
k	Number of search directions
L	Lagrangian function
l	Length of face
M	Mach number
$\hat{\boldsymbol{n}}$	Outward-pointing normal to control volume
Pr	Prandtl number
Pr_t	Turbulent Prandtl number
p	Pressure
$\boldsymbol{Q}$	Vector of cell-averaged conserved variables
q_x, q_y, q_z	Components of heat flux
$\boldsymbol{R}$	Residual for a control volume
Re	Reynolds number
r	Function for turbulence model, also residual vector
S	Magnitude of vorticity
T	Temperature
t	Time
u, v, w	Cartesian components of velocity
V	Volume of control volume
v	Search direction
$\boldsymbol{X}$	Grid-point locations
x	Solution vector
x, y, z	Cartesian coordinate directions

Greek

α	Angle of attack
β	Artificial compressibility parameter, also norm of initial residual
Γ	Boundary of control volume
γ	Ratio of specific heats
κ	Constant for turbulence model, also coefficient for Spalding's boundary layer profile
Λ	Vector of Lagrange multipliers
μ	Laminar viscosity
μ_t	Turbulent viscosity
υ	μ/ρ
υ_t	μ_t/ρ

υ	Dependent variable for turbulence model
ρ	Density
σ	Constant for turbulence model
τ_{ij}	Components of shear stress
χ	Function for turbulence model
ω_1, ω_2	Weights for lift and drag in cost function

Subscripts

∞	Free-stream quantity

Chapter ONE

Introduction

As computing capabilities have continued to advance in recent years, the field of computational fluid dynamics (CFD) has received an increasing amount of attention as an everyday tool. Traditionally, CFD has served largely as an analysis tool during the design process. With the tremendous growth in today's computing power however, designers are looking for CFD to play a larger role in the initial development of new concepts.

Early attempts at design using CFD were based on simplified physical models such as panel methods and linear theory.[1] Where gradient-based optimization algorithms were employed, aerodynamic sensitivity information was calculated using simple finite-difference techniques.[1-3] This method of computing design sensitivities requires virtually no modifications to the existing analysis code.

Unfortunately, many flows that interest the designer contain complex flow physics not able to be captured using such simplified models. For example, high-lift applications may contain highly non-linear regions where more accurate models are required. To compound the problem, the finite-difference technique becomes prohibitively expensive for solutions governed by the Euler or Navier-Stokes equations on meshes containing large numbers of grid points. For a system involving n design variables, $2n$ flow solutions are required to compute the sensitivity information using a central-difference approach. The choice of the step size is an additional drawback. If the step size is too large, truncation errors will degrade the accuracy of the derivatives. Alternatively, if a step size is chosen that is excessively small, subtractive cancellation becomes an issue.

A more efficient and accurate means of computing sensitivity information could prove very beneficial for such applications.

Two methods that have recently been used by researchers to calculate aerodynamic sensitivities are known as the direct differentiation and adjoint approaches. In these methods, the governing equations are differentiated and the resulting equations are used to compute the required gradients. Both techniques may be applied at either the continuous or discrete level. If the governing equations are differentiated prior to discretization, the method is termed a continuous formulation. Here, the resulting derivatives are consistent within truncation error. In a discrete approach, the governing equations are differentiated following the discretization process, and the resulting derivatives are consistent with the flow solver regardless of the mesh size.

The direct differentiation method yields the largest amount of sensitivity information however it requires the solution of a large linear system of equations for each design variable. The solution of each system of equations yields the sensitivity of the flow field variables to the design variable in question. Once these sensitivities are known, gradients of a specified cost function or constraint may be determined with relatively little effort. Therefore, the direct differentiation approach is appropriate when there are few design variables and many cost functions or constraints. Applications of this technique can be found in Refs. 4-10.

Another approach that yields the same sensitivity information as the direct differentiation technique is based on the use of complex variables. This method was originally introduced in Refs. 11 and 12 for a scalar function of a single variable, and resurfaced in a review by Squire and Trapp.[13] It has recently been applied to compute aerodynamic and aerostructural sensitivity derivatives using inviscid flow[14] and has also been extended for turbulent flows.[15] The technique has the advantage of being able to compute highly accurate derivative information without incurring subtractive cancellation errors, nor requiring the differentiation of any source code. The technique is a promising one, however because of extra memory requirements and computational time, fully conclusive work remains to be done in order to assess the efficiency of this technique. Regardless of the efficiency, this approach provides an excellent method for verifying linearizations.

For problems containing many design variables and a single (or few) cost function(s), it is beneficial to employ an adjoint formulation. In this approach, an intermediate vector of Lagrange multipliers, or costate variables, is introduced. These costate variables are determined through the solution of a single linear system of equations. The desired design sensitivities may then be computed as a single matrix-vector product.

Some of the earliest applications of adjoint techniques to aerodynamic design problems can be found in the work of Pironneau.[16,17] In Ref. 16, optimality conditions are derived for a minimum-drag two-dimensional body in a viscous-dominated flow. The method was further developed in Ref. 17 to include flows at higher Reynolds numbers, and Glowinski and Pironneau verified computationally in Ref. 18 that the method could indeed be used for aerodynamic design purposes.

Many researchers have since worked towards applying the adjoint approach to increasingly more complex problems. Angrand uses a discrete adjoint technique in a finite element framework to perform design examples for two-dimensional nozzle and airfoil flows governed by compressible and incompressible potential flow.[19] In Ref. 20, Jameson describes a continuous adjoint formulation using coordinate mappings in two and three dimensions for flows governed by the compressible potential and Euler equations. Frank and Shubin examine a one-dimensional Euler flow through a nozzle in Ref. 21, where they compare a finite-difference approach with a discrete adjoint method. Beux and Dervieux[22] derive a continuous adjoint approach for internal two-dimensional Euler-based problems, then treat the numerical problem using a discrete formulation. Iollo and Salas[23] solve similar problems using the continuous approach. Baysal and Eleshaky examine a two-dimensional supersonic nozzle flow with discrete adjoint and direct methods based on the Euler equations.[4] In Ref. 24, Cabuk and Modi maximize the pressure rise through a diffuser using a continuous adjoint technique for the two-dimensional incompressible laminar Navier-Stokes equations on structured grids. Huan and Modi employ a similar formulation in an effort to minimize drag on a two-dimensional body.[25] In Ref. 26, Reuther *et al* describe the use of a continuous adjoint method on multiblock structured grids to design the wing of a business jet. Anderson and Bonhaus develop a discrete adjoint technique on unstructured meshes for two-dimensional turbulent flows in Ref. 27. Elliott and Peraire also use a discrete adjoint formulation for laminar problems in both two and three dimensions on unstructured meshes.[28] In Ref. 29, Jameson demonstrates a continuous adjoint approach on structured grids for turbulent flows in three dimensions. Here, the eddy viscosity is considered constant and the turbulence model is not differentiated. Soemarwoto has implemented a similar approach on structured grids for two-dimensional applications.[30]

In recent years, software tools have been devised which greatly automate the development of design codes. In these approaches, the existing analysis code is differentiated automatically in a line-by-line fashion to generate new source code which can be used for computing sensitivity information. This technique can either be implemented in “forward” mode (analogous to the direct differentiation technique), or in “reverse” mode (equivalent to an adjoint approach). Many researchers have applied the “forward-mode” of automatic differentiation to design problems; representative examples can be found in Refs. 7, 9, and 31. Refs. 32 and 33 describe implementations using a “reverse-mode” of automatic differentiation. Although the use of automatic differentiation can reduce code development time by orders of magnitude, whether or not the resulting code is as efficient or as understandable as hand-differentiated code is subject to debate. By using a manual approach, the developer can employ knowledge of the analysis code and its solution procedure, enabling potential benefits in both CPU as well as memory utilization. However, the drawback to the manual approach is the considerable amount of time necessary to achieve accurate linearizations.

The current work largely follows the efforts of Anderson *et al.*[27,34] In Ref. 34, a continuous adjoint formulation was used to compute sensitivity derivatives based on the two-dimensional laminar Navier-Stokes equations. Since mesh sensitivity terms do not arise in this approach, the method can lead to derivatives that are inconsistent with the flow solver for problems such as translation and rotation of elements. In Ref. 27, a discrete formulation is applied that yields highly accurate sensitivity information for turbulent flows in two dimensions.

The primary goal of the current work is to develop aerodynamic design optimization capabilities for three-dimensional applications on unstructured meshes. More specifically, a discrete adjoint approach will be described for computing aerodynamic sensitivity derivatives based on the Reynolds-averaged Navier-Stokes equations, using both compressible and incompressible formulations. A one-equation turbulence model is tightly coupled in the solution process to ensure fully consistent derivatives. Linearizations are continuously verified using finite-differences and the complex-variable approach mentioned above. The use of unstructured meshes coupled with an efficient surface parameterization scheme will allow arbitrarily complex geometries to be handled with relative ease. The motivation to implement the technique within both compressible and incompressible frameworks stems from the desire to handle a number of wide-ranging applications outside the aerospace spectrum.

Chapter TWO

The Governing Equations

The governing flow equations used for the current study are the three-dimensional Reynolds-averaged Navier-Stokes equations. To compute the eddy viscosity, a one-equation turbulence model is used. The compressible form of these equations is presented in the following sections. The derivation of the discrete adjoint and design equations is also shown. The solution of the adjoint equation yields an intermediate vector of costate variables and the design equation gives the sensitivity derivatives necessary for optimization.

2.1 The Navier-Stokes Equations

The three-dimensional Reynolds-averaged Navier-Stokes equations in non-dimensional form are given as:[35]

$$V\frac{\partial \boldsymbol{Q}}{\partial t} + \oint_{\Gamma}(\vec{\boldsymbol{F}}_i \cdot \hat{\boldsymbol{n}})d\Gamma - \oint_{\Gamma}(\vec{\boldsymbol{F}}_v \cdot \hat{\boldsymbol{n}})d\Gamma \;=\; 0 \qquad (2.1)$$

where $\hat{\boldsymbol{n}}$ is the outward-pointing normal of the control volume boundary and the vector of cell-averaged conserved variables, $\boldsymbol{Q}$, and inviscid and viscous flux vectors $\vec{\boldsymbol{F}}_i$ and $\vec{\boldsymbol{F}}_v$, are given by

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} \tag{2.2}$$

$$\vec{\boldsymbol{F}}_i = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E + p)u \end{bmatrix} \hat{\boldsymbol{i}} + \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ (E + p)v \end{bmatrix} \hat{\boldsymbol{j}} + \begin{bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ (E + p)w \end{bmatrix} \hat{\boldsymbol{k}} \tag{2.3}$$

$$\vec{\boldsymbol{F}}_v = \boldsymbol{f}_v \hat{\boldsymbol{i}} + \boldsymbol{g}_v \hat{\boldsymbol{j}} + \boldsymbol{h}_v \hat{\boldsymbol{k}} \tag{2.4}$$

where

$$\boldsymbol{f}_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} - q_x \end{bmatrix} \tag{2.5}$$

$$\boldsymbol{g}_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{zy} - q_y \end{bmatrix} \tag{2.6}$$

$$\boldsymbol{h}_v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} - q_z \end{bmatrix} \tag{2.7}$$

The shear stress and heat conduction terms are given by

$$\tau_{xx} = (\mu + \mu_t)\frac{M_\infty}{Re}\frac{2}{3}[2u_x - (v_y + w_z)] \tag{2.8}$$

$$\tau_{yy} = (\mu + \mu_t)\frac{M_\infty}{Re}\frac{2}{3}[2v_y - (u_x + w_z)] \tag{2.9}$$

$$\tau_{zz} = (\mu + \mu_t)\frac{M_\infty}{Re}\frac{2}{3}[2w_z - (u_x + v_y)] \tag{2.10}$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t)\frac{M_\infty}{Re}(u_y + v_x) \tag{2.11}$$

$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t)\frac{M_\infty}{Re}(u_z + w_x) \tag{2.12}$$

$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t)\frac{M_\infty}{Re}(v_z + w_y) \tag{2.13}$$

$$q_x = \frac{-M_\infty}{Re(\gamma - 1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial a^2}{\partial x} \tag{2.14}$$

$$q_y = \frac{-M_\infty}{Re(\gamma - 1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial a^2}{\partial y} \tag{2.15}$$

$$q_z = \frac{-M_\infty}{Re(\gamma - 1)}\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\partial a^2}{\partial z} \tag{2.16}$$

Here, the governing equations have been reduced to non-dimensional form using the free-stream values of density, speed of sound, temperature, thermal conductivity, laminar viscosity and a reference length L. The equations are closed with the equation of state for a perfect gas

$$p = (\gamma - 1)\left[E - \rho\frac{(u^2 + v^2 + w^2)}{2}\right] \tag{2.17}$$

and the laminar viscosity is determined through Sutherland's Law:[35]

$$\mu = \frac{\hat{\mu}}{\hat{\mu}_\infty} = \frac{(1 + C^*)(\hat{T}/\hat{T}_\infty)^{3/2}}{\hat{T}/\hat{T}_\infty + C^*} \tag{2.18}$$

where $C^* = 198.6/460.0$ is Sutherland's constant divided by a free-stream reference temperature, which is assumed to be $460°R$.

The incompressible solvers are based on an artificial compressibility formulation.[36] In this approach, a pseudo-time derivative of pressure is added to the continuity equation, which allows the continuity equation to be advanced in a time-marching manner, much the same as the momentum equations.[37] The system of equations takes the same form as Eq. 2.1, with the conserved variables and fluxes redefined as

$$\boldsymbol{Q} = \begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix} \tag{2.19}$$

$$\vec{\boldsymbol{F}}_i = \begin{bmatrix} \beta u \\ u^2 + p \\ uv \\ uw \end{bmatrix} \hat{\boldsymbol{i}} + \begin{bmatrix} \beta v \\ vu \\ v^2 + p \\ vw \end{bmatrix} \hat{\boldsymbol{j}} + \begin{bmatrix} \beta w \\ wu \\ wv \\ w^2 + p \end{bmatrix} \hat{\boldsymbol{k}} \tag{2.20}$$

$$\vec{\boldsymbol{F}}_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{bmatrix} \hat{\boldsymbol{i}} + \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \end{bmatrix} \hat{\boldsymbol{j}} + \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \end{bmatrix} \hat{\boldsymbol{k}} \tag{2.21}$$

Here, β is the artificial compressibility parameter, and the shear stresses are defined as

$$\tau_{xx} = (\mu + \mu_t)\frac{2}{Re}u_x \tag{2.22}$$

$$\tau_{yy} = (\mu + \mu_t)\frac{2}{Re}v_y \tag{2.23}$$

$$\tau_{zz} = (\mu + \mu_t)\frac{2}{Re}w_z \tag{2.24}$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t)\frac{1}{Re}(u_y + v_x) \tag{2.25}$$

$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t)\frac{1}{Re}(u_z + w_x) \tag{2.26}$$

$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t)\frac{1}{Re}(v_z + w_y) \tag{2.27}$$

As can be seen, the form of the incompressible equations is very similar to that of the compressible equations. This allows the time integration and solution of the linear system at each time step to be performed in the same manner as described for the compressible flow solver.

2.2 Turbulence Model

For the current study, the turbulence model of Spalart and Allmaras[38] is used, non-dimensionalized using the same quantities employed above. This model has been chosen for several reasons. Experience has shown it to be very robust and that it yields good results for transonic and high-lift applications.[38-41] The model is also representative of the form of typical convection-diffusion-type multiequation turbulence models. The model is given as

$$\frac{D\tilde{\upsilon}}{Dt} = \frac{M_\infty}{\sigma Re}\left\{\nabla\cdot[(\upsilon + (1+c_{b_2})\tilde{\upsilon})\nabla\tilde{\upsilon}] - c_{b_2}\tilde{\upsilon}\nabla^2\tilde{\upsilon}\right\} \tag{2.28}$$

$$-\frac{M_\infty}{Re}\left(c_{w_1} f_w - \frac{c_{b_1}}{\kappa^2} f_{t_2}\right)\left(\frac{\tilde{\upsilon}}{d}\right)^2 + c_{b_1}(1-f_{t_2})\tilde{S}\tilde{\upsilon} + \frac{Re}{M_\infty} f_{t_1}\Delta U^2$$

where

$$f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3} \tag{2.29}$$

$$\chi = \frac{\tilde{\upsilon}}{\upsilon} \tag{2.30}$$

$$\tilde{S} = S + \frac{M_\infty}{Re}\frac{\tilde{\upsilon}}{\kappa^2 d^2} f_{v_2} \tag{2.31}$$

$$f_{t_2} = c_{t_3} e^{-c_{t_4}\chi^2} \tag{2.32}$$

and

$$f_{v_2} = 1 - \frac{\chi}{1 + \chi f_{v_1}} \tag{2.33}$$

In these equations, S is the magnitude of the vorticity, and d is the distance to the nearest wall. The function f_w is given as

$$f_w = g\left(\frac{1 + c_{w_3}^6}{g^6 + c_{w_3}}\right)^{1/6} \tag{2.34}$$

where

$$g = r + c_{w_2}(r^6 - r) \tag{2.35}$$

and

$$r = \frac{M_\infty}{Re}\frac{\tilde{\upsilon}}{\tilde{S}\kappa^2 d^2} \tag{2.36}$$

The last term in Eq. 2.28 is used when specifying the transition location. Because the computations in the present work are all assumed to be fully turbulent, this term is not used. Therefore, the definition of f_{t_1}, which is associated with this term, is not given. After Eq. 2.28 is solved for $\tilde{\upsilon}$, the eddy viscosity is computed as $\mu_t = \rho\upsilon_t = \rho\tilde{\upsilon} f_{v_1}$.

2.3 Derivation of the Discrete Adjoint and Design Equations

In a gradient-based approach to an optimization problem, one seeks to minimize some cost function through a series of updates to a vector of design variables, subject to a set of constraints. These updates rely on gradients of the cost function which are used to determine the search direction that will yield an improvement in the cost function. The ability to accurately compute gradients is therefore crucial to such an approach.

In the adjoint approach for design optimization, a cost function is defined and augmented with the flow equations as constraints to form a Lagrangian given by

$$L(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{X}, \Lambda) = f(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{X}) + \Lambda^{\boldsymbol{T}}\boldsymbol{R}(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{X}) \tag{2.37}$$

Here, $f(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{X})$ is the cost function to be minimized and $\boldsymbol{D}$ is a vector of design variables. The vector of Lagrange multipliers (also known as costate variables) is denoted by Λ, and $\boldsymbol{R}$ is the residual of the discretized steady-state flow equations. The vector $\boldsymbol{Q}$ contains the conserved variables and $\boldsymbol{X}$ represents the computational grid. Although not explicitly denoted in Eq. 2.37, both $\boldsymbol{Q}$ and $\boldsymbol{X}$ are functions of the design variables. If a design variable is changed, there is a corresponding change in $\boldsymbol{Q}$ in order to satisfy the flow equations. Similarly, if a shape design variable is altered, the mesh is typically modified to conform to the new surface and to maintain a smooth variation of the mesh points in the interior. Since it is required that the steady-state flow equations be satisfied ($\boldsymbol{R}(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{X}) = 0$), the addition of the residual does not change the cost function. However, with the addition of this term, the derivatives reflect the fact that the flow constraint must be satisfied.

Differentiating Eq. 2.37 with respect to the design variables yields

$$\frac{\partial L}{\partial \boldsymbol{D}} = \left\{\frac{\partial f}{\partial \boldsymbol{D}} + \left[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{D}}\right]^T \frac{\partial f}{\partial \boldsymbol{X}}\right\} + \left[\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{D}}\right]^T \left\{\frac{\partial f}{\partial \boldsymbol{Q}} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right]^T \Lambda\right\} + \left\{\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}}\right]^T + \left[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{D}}\right]^T \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]^T\right\}\Lambda \qquad (2.38)$$

Because Λ is arbitrary, the terms multiplied by $\left[\frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{D}}\right]^T$ may be eliminated using the following equation

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right]^T \Lambda = -\left(\frac{\partial f}{\partial \boldsymbol{Q}}\right) \qquad (2.39)$$

Equation 2.39 is a linear system which represents the discrete adjoint equation for the optimization problem. After the flow equations have been solved for $\boldsymbol{Q}$, the adjoint equation can be solved for the unknown vector of Lagrange multipliers Λ. The remaining terms in Eq. 2.38 can be used to evaluate the sensitivity derivatives as follows:

$$\frac{\partial L}{\partial \boldsymbol{D}} = \left\{\frac{\partial f}{\partial \boldsymbol{D}} + \left[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{D}}\right]^T \frac{\partial f}{\partial \boldsymbol{X}}\right\} + \left\{\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{D}}\right]^T + \left[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{D}}\right]^T \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}}\right]^T\right\}\Lambda \qquad (2.40)$$

After the solution for the costate variables is obtained using Eq. 2.39, the vector containing all of the desired sensitivities can be evaluated as a single matrix-vector product between the linearization of the residual with respect to the design variables and the costate variables, given by the second term on the right hand side of Eq. 2.40. In the current implementation, this matrix-vector product is performed a single row at a time in order to minimize memory requirements.

Chapter THREE

Numerical Solution of the Governing Equations

In this section, the numerical solution of the governing equations is discussed and several representative test cases are shown to establish the accuracy of the flow solvers in both two and three dimensions. The solution procedure used to treat the adjoint and design equations is also presented.

3.1 Numerical Solution of the Flow Equations and Turbulence Model

The flow solvers used in the current work are described at length in Refs. 37, 42, and 43, and are known as FUN2D/3D. These solvers have been chosen based on their ability to accurately compute a wide range of flows, which will be demonstrated in a subsequent section. The codes use an implicit, upwind, finite-volume discretization, in which the dependent variables are stored at the mesh vertices. For both compressible and incompressible flows, the inviscid fluxes at cell interfaces are computed using the upwind scheme of Roe.[44] For compressible flows, the upwind flux formulations of van Leer[45] or Osher[46] can also be utilized. Viscous fluxes are formed using an approach equivalent to a central-difference Galerkin procedure. Temporal discretization is performed using a backward-Euler time-stepping scheme, and multigrid acceleration can be used for the two-dimensional codes.

The implicit scheme employed by the flow solver results in a linear system of equations at each time step that takes the form $Ax = b$:

$$\left\{\frac{V}{\Delta t}\boldsymbol{I} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right]\right\}\Delta^n Q = -R(Q^n) \tag{3.1}$$

where

$$\Delta^n Q = Q^{n+1} - Q^n \tag{3.2}$$

The method used to solve this linear system of equations is a point-iterative scheme described in Ref. 42. Here, the solution is obtained by a relaxation scheme in which $\Delta^n Q$ is obtained through a sequence of iterates $\Delta^i Q$ which converge to $\Delta^n Q$. In this scheme, the grid points are updated so that the odd- and even-numbered points are solved for alternately. In this way, a Gauss-Seidel type of scheme is obtained which can be written as

$$[D]^n\{\Delta^{i+1}Q\} = [\{R\}^n - [O]^n\{\Delta^l Q\}] \tag{3.3}$$

where $\Delta^l Q$ is the most recent value of ΔQ and will be at subiteration level $l = i$ or $l = i+1$ depending on whether the current node being updated is even or odd, and $[D]^n$ and $[O]^n$ represent the diagonal and off-diagonal blocks, respectively, of the matrix A.

The turbulence model is solved separately from the flow equations at each time step, using a backward-Euler time-stepping scheme. The resulting linear system is solved using the same point-iterative scheme employed for the flow equations. The turbulence model is integrated all the way to the wall without the use of wall functions.

3.2 Some Representative Test Cases

The flow solvers have been utilized extensively for a number of test cases and are employed by many groups in industry as well as academia. Some representative cases are included here for demonstration purposes. First, the results for laminar flow over a flat plate computed using FUN2D are compared with the Blasius solution. The flow past a three-element airfoil is also examined using the compressible and incompressible versions of the two-dimensional code. To demonstrate the accuracy of the compressible version of FUN3D, two turbulent flow fields around an ONERA M6 wing are examined, in addition to the transonic flow over a configuration known as Lockheed Wing C,[47] which has separation across the outboard third of the wing. Finally, a wing with a partial-span flap is examined using the incompressible code.

In each test case, the mesh size has been chosen based on both previous experience and the desire to limit turnaround time. For two-dimensional flows, mesh sizes are chosen based on extensive experience for transonic as well as multielement airfoil computations. For example, in Ref. 39, grid convergence studies are shown for three-element airfoil configurations in which the effects of the grid on pressures, velocity profiles, and Reynolds stresses are examined. In addition, studies to determine the effects of the minimum wall spacing have been conducted over a wide range of angles of attack. The meshes for three-dimensional computations are constructed using experience gained from two-dimensional flow. However, the number of points has been decreased in order to reduce turnaround time. As a result, the computations do not represent grid-converged solutions.

3.2.1 Laminar Flow Over a Flat Plate

The two-dimensional incompressible version of the code is used to compute the laminar flow over a flat plate. The mesh contains 16,641 nodes (with 64 points on the surface) with a uniform spacing at the wall of 0.0017 of the length of the plate and has been formed from a stretched rectangular mesh by cutting across the diagonals. A comparison of the computed skin friction with the Blasius solution[35] is shown in Fig. 3.1. The agreement is seen to be very good except in the immediate vicinity of the leading edge singularity. Profiles of the u- and v-velocity components are also included in Fig. 3.1. These profiles have been obtained at a location on the plate where the Reynolds number is approximately 1500. With the spacing at the wall given above, there are about 15 points in the boundary layer. The results for a Blasius profile are also shown, and it can be seen that the agreement is excellent.

3.2.2 Flow Past a Three-Element Airfoil

In Ref. 37 the flow over an Advanced Energy Efficient Transport three-element airfoil has been examined at freestream Mach numbers of 0.15 and 0.26, and a Reynolds number of 9 million. Several representative test cases are shown here to demonstrate the applicability of the codes to these types of flows. The mesh used for these computations is shown in Fig. 3.2 and has been generated using the method of Ref. 37. The mesh contains 70,686 nodes, with 472, 768, and 288 points on the slat, main element, and flap, respectively. The minimum spacing at the wall is 2×10^{-6} based on a reference chord of the airfoil with the elements retracted, and the stretching factor used to generate the volume mesh is 1.15. The geometry is shown in Fig. 3.3, along with the pressure distributions calculated using the compressible version of FUN2D at $0.0°$, $8.0°$, and $16.0°$ angle of attack. The experimental data from Ref. 48 for the three cases is also shown. It can be seen from the figure that as the angle of attack is increased, compressibility effects begin to appear, particularly in the nose regions of the slat and main element. The computations successfully predict these trends, and generally exhibit good agreement for each of the cases. Although not shown, the pressure distributions computed using the incompressible solver agree very closely with the solutions obtained with the compressible code at $M_\infty = 0.15$. Fig. 3.4 shows the lift coefficient as a function of angle of attack for the computations and the experiment. Here, results from the incompressible version of FUN2D are plotted along with results using the compressible version at the higher Mach number. The experiment shows an increase in lift for the main element as the Mach number is increased, and the computations exhibit the same feature. At a $22.0°$ angle of attack, the flow at the higher Mach number has exceeded maximum lift, and the computations successfully predict this trend as well. Finally, Fig. 3.5 shows a plot of u^+ versus y^+ for four chordwise stations along the upper surface of the main element. It can be seen that at each station the computed boundary layer in the near-wall region all collapse to a single curve which closely follows the analytical prediction of Spalding given in Ref. 35, with $\kappa = 0.41$ and $B = 5.2$ as suggested in Ref. 49.

3.2.3 ONERA M6 in Subsonic Flow

The results shown are for flow over an ONERA M6 configuration with only small regions of supersonic flow near the leading edge. The unstructured surface mesh is shown in Fig. 3.6, and contains 9,129 nodes on the wing surface. The entire mesh consists of 359,536 nodes and 2,074,955 tetrahedra with a minimum spacing at the wall of 2×10^{-6} of the mean aerodynamic chord, and has been generated using the method of Ref. 50. The flow conditions are a Mach number of 0.699, a $3.06°$ angle of attack, and a Reynolds number of 11.78 million based on the mean aerodynamic chord. These conditions correspond to those used in the experiments discussed in Ref. 51. Computed pressure distributions at various spanwise locations are shown in Fig. 3.7, along with the experimental results. Also shown are the results obtained from a widely-used structured grid code, CFL3D.[52] The structured grid is 193 x 49 x 33, and contains 312,081 points. As can be seen, the agreement between the two codes and the experiment is generally good, with some slight discrepancies in the suction peak region at the outboard stations. However, a close inspection of the nose regions of each surface mesh reveals that the two geometries are slightly inconsistent, as shown in Fig. 3.8. The source of this inconsistency is unknown. The convergence history for the computation is included as Fig. 3.9, along with the histories of two additional cases discussed in subsequent sections. As seen, the residual is reduced three orders of magnitude and the final lift value is obtained in 300 iterations.

3.2.4 ONERA M6 in Transonic Flow

The next computation is for transonic flow over the same geometry using the mesh described above. In this case, the free-stream Mach number is 0.84, the angle of attack is 3.06°, and the Reynolds number is 11.78 million. This case has a shock emanating from the leading edge as well as a normal shock further aft. These shocks coalesce into a single shock at approximately the $\eta = 0.80$ spanwise station. Figure 3.9 shows the convergence history, while pressure distributions are plotted in Fig. 3.10 for the experiment as well as for FUN3D and CFL3D. The computed results agree well with experiment as well as with each other, although the shocks at the outboard stations are slightly sharper with CFL3D than with FUN3D due to higher resolution.

3.2.5 Lockheed Wing C Configuration

For this case, flow over the Lockheed Wing C configuration[47] is examined. This is a transonic case with separation occurring aft of the shock over the outboard third of the wing. The mesh for this computation consists of 523,785 nodes and 3,097,641 tetrahedra and a spacing at the wall of 2×10^{-6} of the mean aerodynamic chord. The surface mesh is shown in Fig. 3.11, and contains 9,730 nodes on the wing surface. This mesh has been generated using the technique described in Ref. 53. Note that although the wing is a no-slip surface, the symmetry plane does not include the spacing at the wall necessary for viscous computations. The transition from the viscous surface to the symmetry plane is done over the last inboard cell and should not affect the overall solution. The free-stream Mach number is 0.8833 and the angle of attack is 4.5°. The Reynolds number based on the mean aerodynamic chord is 10 million. The convergence of the solution is included in Fig. 3.9, and the pressure distributions for the converged solution at various spanwise locations are shown in Fig. 3.12. Also shown are results obtained through the use of CFL3D on a mesh containing 578,641 points. As in the previous example, the computations are in fairly good agreement with experiment and agree quite well with one another.

3.2.6 Incompressible Flow Over a Wing with a Partial-Span Flap

In order to demonstrate results obtained using the three-dimensional incompressible code, results are shown from Ref. 37. For this case, flow over a wing with a partial-span flap is examined. Here, the angle of attack is 10° and the Reynolds number is 3.7 million. The experimental results are described in Ref. 54, and were obtained at a Mach number of 0.2. The surface mesh containing 14,924 points on both wing surfaces is shown in Fig. 3.13, and the computed pressure distribution is compared with the experimental results in Fig 3.14. Results obtained using CFL3D are also shown. The unstructured mesh for this computation has been generated using the method of Ref. 50 and consists of 549,176 nodes, whereas the structured mesh contains approximately 2.25 million nodes. The agreement is generally good, although the suction peaks are slightly underpredicted in the FUN3D computations. As noted in Ref. 37, grid refinement and accounting for compressibility effects in the leading edge region may resolve these discrepancies.

3.3 Numerical Solution of the Adjoint Equation

The adjoint equation given in Eq. 2.39 represents a linear set of equations for the costate variables Λ. To solve this system of equations, a technique known as the Generalized Minimal Residual (GMRES) method[55] is employed. This solver has been chosen based on its ability to handle large linear systems in a robust manner. The basic outline of the structure of the GMRES algorithm is shown below and is taken directly from Ref. 55:

To solve the system $Ax = b$:

1. Start: Choose an initial guess, x_0, for the solution of the linear system and compute

$$r_0 = b - Ax_0 \text{ and } v_1 = \frac{r_0}{\|r_0\|}.$$

2. Iterate: For $j = 1, 2, \ldots, k$ until satisfied do:

$$h_{i,j} = (Av_j, v_i) \qquad i = 1, 2, \ldots, j$$

$$\hat{v}_{j+1} = Av_j - \sum_{i=1}^{j} h_{i,j} v_i$$

$$h_{j+1,j} = \|\hat{v}_{j+1}\|$$

and $v_{j+1} = \dfrac{\hat{v}_{j+1}}{h_{j+1,j}}$.

3. Form the approximate solution $x_k = x_0 + V_k y_k$, where y_k minimizes the functional

   $J(y) = \left\| \beta e_1 - \overline{H}_k y \right\|$.

The symbol e_1 represents the first column of the $(k+1) \times (k+1)$ identity matrix, and $\overline{H}_k$ is an upper-Hessenberg matrix with an additional row whose only nonzero element is $h_{k+1,k}$ in the $(k+1, k)$ position. The value of β is simply the norm of the initial residual $\beta \equiv \|r_0\|$. The current implementation is adapted from Ref. 56.

Although the adjoint system of equations can be solved directly using GMRES as outlined above, a time-like derivative is added and the solution is obtained by marching in time, much like the flow solver:

$$\left\{ \frac{V}{\Delta t} \boldsymbol{I} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}} \right]^T \right\} \Delta^n \Lambda = -\frac{\partial f}{\partial \boldsymbol{Q}} - \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}} \right]^T \Lambda^n \tag{3.4}$$

where

$$\Lambda^{n+1} = \Lambda^n + \Delta^n \Lambda \tag{3.5}$$

The time term can be used to increase the diagonal dominance for cases in which GMRES alone would tend to stall. With the addition of the time term, the solution is obtained through a sequence of linear systems where the right hand side is continuously changed. It has been found through numerical experiments that this ultimately results in a more robust adjoint solver.

Due to the large amount of code resulting from the linearization of the viscous terms and the turbulence model, these contributions are stored in the present implementation. Because the stencil for the inviscid contributions is larger, the linearization of these terms is recomputed at each step to avoid the need for extra storage and data structure.

To precondition the linear system, an incomplete LU decomposition of the matrix obtained from a first-order accurate discretization is used. The preconditioning is applied on the left and no fill-in is allowed (ILU[0]).[57] (Right preconditioning has also been implemented with no noticeable differences.) Numerical experiments using this preconditioner have shown that some cases are slow to converge. An alternate means of preconditioning that has often been found useful is to employ a point-iterative scheme similar to that used for the flow equations. This technique allows for continual improvement in preconditioning the first-order system but is only effective when the time step is small and the matrix is diagonally dominant.

In the present work, the differentiation of both the flow equations and the turbulence model is accomplished by “hand differentiating” the code, and individual sections are continuously verified using finite differences and a complex-variable approach. As mentioned earlier, automated tools for performing code differentiation have recently been developed; however, these techniques are considered to be in their infancy and the efficiency of the resulting code is questionable. This is the primary motivation for pursuing a development based on hand differentiation of the flow solvers.

For obtaining the solution of the adjoint equations, the turbulence model is tightly coupled during the solution process, whereas it is solved separately during the flow analysis. During development, various treatments of the turbulence model have been studied and it has been found that the close coupling of the turbulence model is required in order to obtain sensitivity derivatives consistent with those obtained using finite differences. This will be illustrated in a subsequent section.

Chapter FOUR

Grid Generation and Mesh Movement Strategy

For all of the two-dimensional computations to be shown, the meshes have been generated using the method described in Ref. 53. In three dimensions, the methods of Refs. 50 and 53 are utilized. Both techniques employ an advancing front type of methodology and generate good quality grids for both inviscid and viscous calculations.

4.1 Surface Parameterization

The technique used to parameterize the surface geometry relies on a scheme recently introduced by Samareh.[58] The method utilizes a free-form deformation technique similar to that used in the motion picture industry for animating digital images. Here, a B-spline net is used to describe the changes in the geometry, rather than the geometry itself. The net is placed around the baseline mesh and the changes at each point on the surface are obtained by interpolation from the changes in the B-spline net. The control points in the net may be used directly as the design variables, or they may be further grouped into design variables such as camber, thickness, and twist. This parameterization technique has been chosen for its ability to handle arbitrary geometries and because the mesh generation process does not depend on a prior parameterization of the geometry. This allows meshes which have been previously generated solely for analysis to be utilized for design purposes.

4.2 Mesh Movement Procedure

When the design process requires modifications to the surface geometry, the computational mesh must be deformed to reflect the changes. For inviscid flows, the mesh movement strategy is based on the spring analogy described in Ref. 59. The edges of the mesh are treated as tension-springs, and the following equation is solved using a Jacobi iteration process:

$$\sum_{j \in N_i} K_{ij}(\Delta \boldsymbol{x}_i - \Delta \boldsymbol{x}_j) = 0 \tag{4.1}$$

Here $\Delta \boldsymbol{x}_i$ and $\Delta \boldsymbol{x}_j$ represent the change in the coordinates of nodes i and j from the initial mesh to the desired mesh. The spring constants K_{ij} are assumed to be l_{ij}^{-2}, where l is the length of the edge connecting node i to node j. Since this technique may result in crossed grid lines, the required shift of the surface coordinates is decomposed into a series of smaller movements (usually around 10), and Eq. 4.1 is relaxed for each change in the surface. This strategy has been found to work well for Euler-based designs, provided the initial mesh does not contain stretched or deformed cells.

For viscous meshes, the method described above is not adequate and can easily lead to crossing mesh lines and negative volumes. For these cases, the nodes near viscous surfaces are shifted by interpolating the changes in the coordinates at the boundaries of the nearest surface triangle or edge. For a graphical illustration of this process, see Fig. 4.1. As surface points A and B are relocated to positions A' and B', respectively, field point X is shifted to position X' such that segment XX' is equal and parallel to CC'. This technique is blended with a smoothing procedure so that away from the highly stretched cells near the surface, the mesh movement reverts to that of the procedure described above for inviscid meshes. Further details can be found in Ref. 34.

As the surface is deformed during the design process, the procedure described above yields a corresponding change in the interior mesh points as well. The effect of the changing grid is reflected through the mesh sensitivity terms given by $\partial \boldsymbol{X} / \partial \boldsymbol{D}$ in Eq. 2.40. The computation of these terms is achieved by differentiating the mesh movement process described above.

4.3 Some Observations

For most two-dimensional applications, the procedure described provides a very robust mesh movement algorithm. Large deformations of the surface geometry can be accommodated while still maintaining high-quality meshing within the interior of the domain. To illustrate this effect, a test is performed where a viscous grid for a NACA 0012 airfoil as shown in Fig. 4.2 is examined. The surface is described parametrically using a third-order B-spline with 29 control points,[27] and several control points around the airfoil are perturbed in the vertical direction as shown in Fig. 4.3 so as to significantly alter the surface geometry. The mesh movement process is applied, and the resulting mesh is shown in Fig. 4.4, with a close-up of the nose region shown in Fig. 4.5. It can be seen that the procedure yields a valid mesh even in the boundary layer region, where the highly stretched cells have been modified successfully.

One instance in which this process suffers, however, is in the presence of multi-element configurations such as the one depicted in Fig. 4.6. There is a tendency to open “gaps” in the mesh between elements when the elements are allowed to translate away from one another. A related problem occurs when the elements move closer together in which case there is a “jamming” together of mesh points. These difficulties are simply due to the fact that no additional mesh points are inserted or removed during the process so that as elements shift in relation to one another, voids can be created. This has not had a detrimental effect on the flow solver and can be remedied by periodically regenerating the mesh.

In three dimensions, the current mesh movement procedure is inadequate when large changes in the geometry are required. In these cases, negative cell volumes have been observed to occur around the edges of the planform. Therefore, in this work, changes in the geometric design variables have been limited to only a few percent of the chord. Further research is required to develop a more reliable methodology for large geometric changes in three dimensions.

Chapter FIVE

Consistency of Linearization

During code development, great care has been taken at each step to ensure that the derivatives are consistent with those obtained using finite differences. In this section, the accuracy of the resulting derivatives is verified for both compressible and incompressible flow. Comparisons are made between derivatives computed using finite differences with those obtained using the adjoint method. When computing derivatives using finite differences, central-difference formulas are used with a step size of 1×10^{-5}, and the flow solver is converged to machine accuracy. Derivatives computed with the adjoint approach are also compared with results from a relatively new complex-variable approach. All of the results shown below are for turbulent flows although the consistency of derivatives has been verified for inviscid and laminar flows as well. Though not the focus of the current work, the accuracy of the two-dimensional compressible linearizations is included here for use in a subsequent discussion.

5.1 Methods Used to Verify Linearization

Two different methods are used to verify the hand-differentiated codes. The first is the commonly-used finite-difference technique. In addition to this approach, an alternative method for computing sensitivity derivatives based on the use of complex variables is employed. The complex-variable approach was originally suggested in Refs. 11 and 12 for determining derivatives of complicated non-linear functions of a single variable, and has recently been reviewed in Ref. 13.

An in-depth study of the method as applied to aerodynamic simulation codes is not presented here; for such investigations, the reader is referred to Refs. 14 and 15.

Consider the central-difference approximation f' to the derivative of a function f:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \tag{5.1}$$

This expression for f' is widely used to verify linearizations. Eq. 5.1 has a truncation error of $O(h^2)$, and it has been well documented that such approximations to f' may be dependent on the choice of the step size h for highly non-linear functions. A small step size is usually desired in order to keep the truncation error low. However, as h gets smaller in magnitude, subtractive cancellation tends to dominate the computation because the function is only evaluated to machine precision. Thus the expression in Eq. 5.1 can often yield approximations for f' that are highly dependent on the step size and may even be highly inaccurate.

Alternatively, the derivative of a real-valued function may be obtained through the simple use of complex variables as discussed in Ref. 13. Consider the Taylor series expansion of f using a complex step size ih:

$$f(x+ih) \;=\; f(x) + ihf'(x) - \frac{h^2}{2}f''(x) - \frac{ih^3}{6}f'''(x) + \frac{h^4}{24}f^{iv}(x) + \ldots \tag{5.2}$$

Solving for the derivative of f yields

$$f'(x) \approx \frac{Im[f(x+ih)]}{h} \tag{5.3}$$

Note that this expression has a truncation error of $O(h^2)$, just as Eq. 5.1. However, the subtraction of two quantities has been eliminated, therefore the issue of subtractive cancellation error need not be of concern when employing Eq. 5.3, and true second-order accuracy can be obtained.

The significance of Eq. 5.3 lies in the fact that very little coding effort is required to implement the complex variable strategy – not a single line of code needs to be differentiated. To incorporate the complex-variable approach into an existing flow solver, all that is required is that the floating point variables be declared as complex and a complex perturbation is given to the design variable of interest. The resulting flow solver is then run and the derivatives of any function dependent on the flow solution is determined by examining the complex part divided by the step size (see Eq. 5.3). The primary drawback is that roughly twice the amount of memory is required, and numerical experiments have shown that the CPU time needed for a function evaluation is increased approximately by a factor of three.

5.2 Linearization of Two-Dimensional Compressible Solver

The two-dimensional compressible version of the flow solver has been previously differentiated as described in Ref. 27. The results are repeated here to support a discussion to be presented in an upcoming section.

For demonstrating the consistency of the derivatives obtained using the adjoint formulation with those obtained using finite differences, two test cases are considered. The first case is a 2-element airfoil at a free-stream Mach number of 0.25, an angle of attack of 1°, and a Reynolds number of 9 million based on the chord of the airfoil. The geometry has been chosen arbitrarily and is that given in Ref. 60.

The mesh used for this test has 4,901 nodes and is shown in Fig. 5.1. The geometry of each airfoil is described with a third-order B-spline. The derivatives of the lift and drag coefficients with respect to the vertical and horizontal positions of four shape design variables have been obtained. The locations of the design variables are indicated by the solid circles shown in Fig. 5.2. As seen in the figure, two of these design variables are located on the main airfoil and two are located on the flap. For each element, one design variable is located on the upper surface near the nose of the airfoil and one is located near the rear. A comparison of derivatives of the lift and drag coefficients with respect to changes in the vertical position of these design variables is shown in Tables 5.1 and 5.2, while Tables 5.3 and 5.4 compare the derivatives of the lift and drag coefficient with respect to x- and y-translation of the flap. Note that this required two solutions of the adjoint equation — one for lift and one for drag. As seen, the derivatives obtained with the adjoint approach are in very good agreement with the finite-difference derivatives for all cases. Although not shown, similar accuracy is obtained for the derivatives with respect to horizontal changes in the control points.

In order to further demonstrate the accuracy of the differentiation, a case containing transonic flow is examined. An RAE 2822 airfoil is used at an angle of attack of 2.81°, a Mach number of 0.75, and a Reynolds number of 6.2 million. The mesh contains 14,127 nodes and the spacing at the wall is 1×10^{-5}. The computed pressure distribution is shown in Fig. 5.3 along with the corresponding experimental data.[61] For this case, a strong shock is present on the upper surface which separates the flow immediately downstream. The locations of the three design variables are shown by the filled circles in Fig. 5.4. The corresponding sensitivity derivatives for the lift coefficient with respect to a vertical movement of the control points are listed in Table 5.5 along with those for Mach number and angle of attack. The agreement with finite differences is very good. Numerical experiments using different step sizes for the point downstream of the shock have shown that finite differences for this control point are somewhat sensitive to the perturbation level. For example, step sizes of 1×10^{-6} and 5×10^{-5} result in finite-difference derivatives of 1.9187 and 1.9150 respectively. The complex-variable approach yields a derivative of 1.9219, which agrees with the adjoint result to five significant digits.

5.3 Linearization of Three-Dimensional Compressible Solver

To verify the accuracy of the derivatives in three dimensions, a similar experiment is conducted. For this case, an ONERA M6 wing[51] has been parameterized using 46 design variables describing the planform, twist, shear, thickness, and camber. The design variables are depicted in Fig. 5.5 where twist (about the quarter-chord) and wing shear have been parameterized at five spanwise locations. The thickness and camber have also been parameterized using the six locations shown in the figure. The design variables describing the planform are not shown in the figure nor are thickness and camber design variables along the leading and trailing edges. The mesh used for these tests contains 16,391 nodes and 90,892 tetrahedra and is shown in Fig. 5.6. The flow conditions are an angle of attack of 2°, a Reynolds number of 5 million based on the mean aerodynamic chord, and a Mach number of 0.3. In this test, the cost function is a linear combination of the lift and drag coefficients so that only one adjoint solution is required. The derivatives of the cost with respect to the angle of attack and the Mach number as well as the derivatives with respect to four of the shape parameterization variables are shown in Table 5.6. As can be seen, the consistency between the derivatives obtained with the adjoint formulation and finite differences is excellent. Additional derivatives for the design variables depicted in Fig. 5.5 have also been verified with comparable accuracy.

Derivatives obtained using the linearization of the compressible version of FUN3D are now compared with results from the complex-variable approach. For this test, derivatives of the lift coefficient for turbulent flow are computed on the same ONERA M6 mesh as described above, using the same free-stream conditions. Results for this experiment are shown in Table 5.7, where the design variables correspond to those shown in Fig. 5.5. Note that the adjoint and complex-variable results exhibit excellent agreement, yielding identical derivatives for several of the design variables and differing at most in the sixth decimal place. However, the finite-difference results show some discrepancies attributable to subtractive cancellation error.

5.4 Linearization of Three-Dimensional Incompressible Solver

The accuracy of the incompressible linearizations is verified in a similar fashion using the same mesh as above. The flow conditions are an angle of attack of 3° and a Reynolds number of 5 million. The shape design variables are the same as those used in verifying the sensitivity derivatives for the compressible solver. Results are presented in Table 5.8, and the linearizations are shown to be highly accurate.

Chapter SIX

Linearization Approximations

Due to the complexity in achieving accurate linearizations for use in Eqs. 2.39 and 2.40, one may consider the use of simplifying assumptions. Clearly, a great deal of effort can be avoided if certain terms may be neglected or replaced with simpler approximations without seriously compromising the accuracy of the results. The previous sections have established the accuracy of the derivatives obtained from the adjoint formulation using a consistent linearization of the flow solvers. This section will examine the accuracy of the derivatives obtained using several natural approximations. These numerical experiments are conducted in two dimensions using the test case and flow conditions used for Table 5.1. A discussion of each of the approximations is given below and some representative derivatives for vertical changes in the design variables are shown in Table 6.1.

6.1 First-Order Adjoint Solution

For second-order accurate schemes, the complete linearization of the inviscid contribution to the residual requires information from mesh points beyond the immediately adjacent nodes. This requirement arises from having to form gradients of the dependent variables at the nodes in order to extrapolate them to the faces of each control volume. This large stencil makes an exact linearization quite tedious. However, if the fluxes are formed using only nearest-neighbor information, the amount of coding involved is drastically reduced, as differentiation of the gradient terms are no longer required. This corresponds to using a first-order accurate scheme for the convective terms and results in a linear system that is easier to solve when using the ILU(0) preconditioner, since the bandwidth of the coefficient matrix is reduced significantly.

In Table 6.1, derivatives obtained using a first-order linearization of the convective terms are compared with those obtained from the linearization of the higher order residual. For these results, the first-order approximation is made in evaluating both Eqns. 2.39 and 2.40. Using this approximation, the derivative of the lift with respect to a vertical shift of the design variable towards the rear of the flap is within 8% of the correct value. However, the derivatives obtained by ignoring the higher order terms are generally highly inaccurate and several are of incorrect sign. The derivatives of incorrect sign would most certainly have an adverse effect on an optimization process, especially near a minimum.

6.2 “Frozen” Turbulence Model

An accurate linearization of the turbulence model can be difficult to obtain. As seen from the equations given in Section 2.2, there are many terms and additional functions that must be properly differentiated. These terms exhibit complex dependency on both the flow variables as well as mesh-dependent quantities such as the distance to the wall. By assuming that the turbulence model is “frozen”, a significant reduction in the required level of effort may be obtained. This approach has been previously used in Refs. 29 and 30 for structured grid applications to airfoils and wings. In these references, successful optimizations have been performed although the accuracy of the derivatives has not been explicitly demonstrated. To implement this approach in the current framework, the adjoint and design equations are solved as in a laminar calculation, but with the eddy viscosity included as $\mu = \mu + \mu_t$.

Results obtained by making the assumption of a constant eddy viscosity are listed in Table 6.1 for derivatives with respect to vertical shifts in the control point locations. While the computed sensitivities show a large amount of error when compared to finite differences, the current results all exhibit the correct sign. However, for derivatives associated with horizontal changes in these same design variables, several are of incorrect sign. For example, the finite-difference derivative obtained by perturbing point A in the horizontal direction is -0.18060 whereas the current approximations to the linearizations yield 0.30328. For this same design variable, use of the complete linearization yields a derivative of -0.18052 which is less than 0.05 percent different from the finite-difference result.

A similar technique that can be used to simplify the implementation is to neglect the contributions from the turbulence model in Eq. 2.40. This assumption would eliminate the need to linearize the turbulence model with respect to the design variables and is primarily motivated from numerical experiments in which it has been observed that the costate variable associated with the turbulence model is much smaller than the others and decays very rapidly away from the body. To illustrate this effect, the variation of the five costate variables with the distance from the wall at $x/c = 0.2$ is plotted in Fig. 6.1. It can be seen that the first four variables are considerably larger in magnitude than the fifth costate variable, which corresponds to the turbulence model. Although not shown, numerical experiments using this approximation indicate that the resulting accuracy of the derivatives is poor with many derivatives of incorrect sign. This is due to the relatively large magnitude of the $\partial R / \partial D$ term multiplying the costate variable corresponding to the turbulence model.

6.3 Extent of Mesh Sensitivities

For each design variable, the evaluation of Eq. 2.40 requires the same amount of work as a matrix-vector product of the costate variables with the linearization of the residual with respect to the design variables. This also includes computation of the mesh sensitivities for each design variable. For large numbers of design variables and mesh points, this can potentially represent a significant expense due to the complicated linearization of the residual.   Also, mesh sensitivities can be expensive to compute because the mesh movement process must essentially be repeated for each design variable. If the mesh is analytically defined and moved, this computation may be negligible. However, when the mesh movement procedure is expensive, the cost of the adjoint approach is not independent of the number of design variables.

Because nodes further away from the body are subjected to more moderate changes than those in the immediate vicinity of the surface, it may be possible to neglect terms in Eq. 2.40 that are sufficiently far from the body. This could help to reduce the cost of evaluating Eq. 2.40 by avoiding the need to include terms from every mesh point in the field.

To investigate the effects of neglecting terms farther from the surface, a region around the surfaces of the airfoil is defined by first “tagging” the nodes on the surface and then identifying nodes that lie within a set number of grid layers adjacent to the surface. Figure 6.2 shows an example where three grid layers have been tagged. Outside of this region, the mesh sensitivities are set to zero to emulate the effect of neglecting all the contributions outside of the tagged region.

The sensitivity derivatives for the lift coefficient with respect to vertical and horizontal translations of the flap are computed for a varying number of grid layers and the results are shown in Fig. 6.3. Here, $\bar{C}_{l_{trans}}$ is the ratio of the approximate derivative to the derivative obtained by including the mesh sensitivities at every grid point in the domain. In this figure, the curve labeled n/n_{tot} is the ratio of the number of nodes where mesh sensitivities are employed to the total number of nodes in the mesh. It should be noted that examining a single derivative may not be representative of the behavior of the rest of the derivatives and an accurate computation of this derivative does not guarantee accuracy for the remaining derivatives. However, inaccuracy of this derivative demonstrates that neglecting the full effects of the mesh sensitivities may have an adverse effect on other derivatives as well.

As seen in the figure, the influence of the mesh sensitivities gradually decays away from the surface. Accurate results are obtained when the number of mesh layers is greater than approximately 15. At this point, about half the total number of points in the mesh are included in the layers so that a factor of two savings could be realized when evaluating Eq. 2.40. When many design variables are present, neglecting some of the mesh sensitivities could lead to a substantial savings in computer time. However, for the present study, the computer time required for evaluating Eq. 2.40 does not dominate the overall optimization process so this strategy is not used.

6.4 Level of Convergence Required for the Adjoint Solution

Here, the level of convergence required of the adjoint solution in order to achieve acceptable accuracy in the results is examined. All of the results shown to this point have been obtained using an adjoint solution converged to machine accuracy. If it is possible to obtain sufficient precision through a partially-converged adjoint solution, a substantial CPU savings may be realized.

For this test, the flow solver is converged to machine accuracy and the derivatives presented for the multielement airfoil in Section 5.2 are recomputed using partially-converged solutions of the adjoint system. Figure 6.4 shows the percent error in each of the derivatives as the adjoint solution is converged towards machine accuracy. The error is a measure of the difference between the approximate result and the results that are listed in Tables 5.1 and 5.3 for the adjoint method. It can be seen that after a two order-of-magnitude decrease in the residual of the adjoint system, the derivatives are all within 1% of their final values. After three orders of magnitude, the results are generally on the order of 0.01% in error. However, even with only a one order-of-magnitude reduction, all of the derivatives are within 6% of their final values. Note that the curve corresponding to point D appears flat initially because this derivative does not require a drop in the residual of the adjoint system in order to obtain a result within 0.01% of its final value. Based on this simple test, it appears that for practical design purposes it may be possible to employ sensitivities computed using partially-converged adjoint solutions. This could have a significant impact on the amount of time spent in evaluating sensitivity information.

Chapter SEVEN

Relative Efficiency of the Adjoint Solver

Timing results are presented for the compressible flow and adjoint solvers for inviscid and turbulent flow on several meshes with varying levels of refinement. For both inviscid and viscous flows, results are shown for subsonic and transonic Mach numbers. The flow conditions for the subsonic results are a Mach number of 0.5 and a 2° angle of attack. For the transonic results, a Mach number of 0.84 and an angle of attack of 3.06° are used. The Reynolds number is 5 million based on the mean aerodynamic chord for each of the viscous runs. All of the calculations are run using Roe's scheme for the convective terms. The flow solver is run using the point-iterative scheme described in Section 3.1, while the adjoint solver utilizes GMRES with 45 search directions and the ILU(0) preconditioner as discussed in Section 3.3. The CFL number for the flow solver is ramped linearly from 10 to 200 over 50 iterations, while the adjoint solver employs a constant CFL number of 10,000. If a computation of the adjoint fails to successfully converge, these values are lowered appropriately. For the inviscid computations, mesh sizes of 22,677, 53,963, and 357,900 nodes are used. These meshes contain 3,856, 9,872, and 39,588 nodes, respectively, on the wing surface. Meshes containing 62,360, 129,541, and 359,536 nodes are utilized for the viscous runs. The number of points on the wing surface for each of these meshes is 3,396, 4,142, and 9,129, respectively. For each of the computations, a single 250 MHz R10000 processor is used.

7.1 Subsonic Inviscid Flow

The convergence history as a function of CPU time is shown in Fig. 7.1 for the coarse mesh. It can be seen that the adjoint solution converges at a rate of roughly one-third that of the flow solver for this case. Similar results are shown in Fig. 7.2 for the medium mesh, where the adjoint solver converges at approximately the same rate as the flow solver after overcoming a slight initial increase in the residual. After the residual is reduced six orders of magnitude from its initial value, the convergence exhibits an oscillatory behavior. The reason for this is unclear, and it has been observed in other cases as well. Fig. 7.3 shows results for the same flow conditions on the fine mesh. Here, the adjoint solution converges slightly faster than the flow solver for the first six orders of magnitude of reduction in the residual. However, at this point in the convergence history, the solution stalls prior to achieving machine accuracy.

7.2 Transonic Inviscid Flow

Results are shown in Fig. 7.4 for the coarse mesh, where the adjoint solver is shown to converge at a rate of roughly one-fourth that of the flow solver. Fig. 7.5 shows results for the same conditions on the medium mesh, where the convergence rate of the adjoint solver can be seen to be just slightly slower than that of the flow solver. For this case, the adjoint solver exhibits an oscillation in the latter stages of convergence. As in the subsonic results, it is unknown why this occurs. Finally, this computation is also performed on the fine mesh, and results are presented in Fig. 7.6. Although the convergence rate of the adjoint solver appears to be marginally faster than that of the flow solver, the residual for the adjoint system stalls after converging six orders of magnitude.

7.3 Subsonic Turbulent Flow

Similar convergence histories are now examined for viscous flow. Fig. 7.7 shows the results for the flow and adjoint solvers on the coarse mesh. Both solvers converge at approximately the same rate. However, the adjoint solution shows a tendency to oscillate after reducing the residual by five orders of magnitude. Results for the medium mesh are shown in Fig. 7.8. For this mesh, the adjoint solution requires two to three times longer than the flow solution to converge two orders of magnitude and appears to stall at this level. Finally, convergence histories for the fine mesh are plotted in Fig. 7.9. The adjoint solution exhibits a convergence rate similar to that of the flow solver in the initial stages, but eventually slows down after two orders of magnitude.

7.4 Transonic Turbulent Flow

The final set of results is for transonic turbulent flow. The convergence histories for the coarse mesh are shown in Fig. 7.10. Here, the adjoint solution converges similar to the flow solver, but begins to oscillate in the latter stages of convergence. Fig. 7.11 shows the convergence rates for the medium mesh. As for the subsonic results on this mesh, the rate of convergence of the adjoint system is sluggish – approximately eight to ten times slower than the flow solver. For the fine mesh results depicted in Fig. 7.12, the adjoint solver converges at a rate of approximately six times that of the flow solver.

7.5 Some Observations

As has been shown, an inviscid adjoint solution can require as much as four times the cost of a flow solution under the same conditions. This ratio can reach as high as eight to ten for turbulent flows. The adjoint solver has also shown a tendency to stall or oscillate prior to achieving machine accuracy in some cases. However, in all cases where oscillations occur, the residual has been decreased to near machine zero. Based on results from Section 6.4, the resulting derivatives should still be highly accurate.

Chapter EIGHT

Sample Optimizations

In this section, the optimization framework is described, and sample optimizations are presented. These cases serve to demonstrate the functionality of the design methodology and are not intended to be realistic studies.

8.1 Cost Functions

For the current study, the cost function is composed of a linear combination of the lift and drag coefficients:

$$f = \omega_1 (C_l - C_l^*)^2 + \omega_2 (C_d - C_d^*)^2 \tag{8.1}$$

The drag can be minimized while maintaining a specified lift by adjusting the weights associated with each term in Eq. 8.1 so that neither term dominates the other. The current method for choosing the initial weights is to simply set the ratio of ω_2 to ω_1 to be equal to the ratio of the lift to the drag:

$$\frac{\omega_2}{\omega_1} = \frac{C_l}{C_d} \tag{8.2}$$

During the design process, these weights may require adjustment. However, this avoids the need to solve separate adjoint equations for lift and drag. Since the optimizer requires the gradients of any constraints on the design, an explicit lift constraint would necessitate an additional adjoint solution to determine its gradient.

8.2 Design Variables

For both incompressible and compressible flows, the angle of attack can be utilized as a design variable. For compressible flows, the free-stream Mach number can also be specified as a design parameter. The Mach number and angle of attack are useful as design variables in such problems as correlating wind tunnel data with computations. When the shape is evolving, the surface is parameterized using the technique implemented by Samareh[58] and discussed in Section 4.1.

8.3 Optimization Technique

The optimization technique used in all of the results to follow is the quasi-Newton method of Davidon-Fletcher-Powell.[62,63] The current implementation of this technique, referred to as KSOPT, allows for multipoint optimization as well as both equality and inequality constraints.[64] For the present work, the multipoint capability is not utilized although this is an obvious future requirement.

8.4 Inviscid Drag Reduction for ONERA M6 Wing

An example optimization is conducted for inviscid flow over the ONERA M6 wing.[51] The free-stream Mach number for this case is 0.84 and the angle of attack is $3.06°$. The mesh used for this computation consists of 53,961 nodes and 287,962 tetrahedra. This mesh is extremely coarse and is not adequate for accurate computations; it serves merely as an initial demonstration for evaluating the methodology. The contours for the initial and final density distribution on the surface of the wing are shown in Fig. 8.1 with the corresponding pressure distributions shown in Fig. 8.2. The objective of the optimization is to reduce the drag while maintaining a specified lift. For this design, the angle of attack is allowed to change in addition to ten shape design variables (four twist, four camber, and two thickness). The twist variables are located at the four outboard stations in Fig. 5.5 and are allowed to increase or decrease by $1°$. The thickness and camber variables at positions 3 and 4 are also design variables as is the camber at positions 5 and 6. Each of these is allowed to change by 2 percent of the span. After ten design cycles, none of the design variables have reached their limits, and the drag has been reduced from 0.0182 to 0.0167 while the lift has been maintained. Exact timings for optimizations are not available, but an estimate of three days on a Silicon Graphics Origin 2000 with R10000/250MHz processors can be made. (It should be noted that the flow solver utilizes four processors although the parallelization is achieved through compiler directives which yields only about fifty percent scalability for the current computations. The adjoint solver currently utilizes only a single processor.) The pressure distribution shown in Fig. 8.2 indicates that the shock has weakened at all of the spanwise stations. As can be seen from the geometric cross-sections in Fig. 8.3, the optimizer has reduced the wing thickness at the inboard stations, and the forward portion of the lower surface has been modified to help achieve the lift constraint. In addition, the camber near the wing tip has been increased substantially.

8.5 Turbulent ONERA M6 Wing Redesign

A transonic wing design has been conducted using an ONERA M6 mesh consisting of 62,360 nodes and 355,814 tetrahedra. As in the previous example, the mesh size has been chosen solely to enable a demonstration of the implementation. The flow is assumed to be fully turbulent at a Mach number of 0.84, an angle of attack of 3.06°, and a Reynolds number of 5 million. Density contours for the initial flow field are shown in the top portion of Fig. 8.4. The weakness of the normal shock is in large part due to the coarseness of the mesh; further refinement would yield a shock structure similar to that shown in the initial flow field in Fig. 8.1.

The objective of the design is to reduce the drag while holding the lift constant. For this case, thickness and camber have been allowed to vary at two chordwise stations located at the mid-span of the wing. These design variables have been allowed to change up to 1 percent of the span of the wing. The angle of attack is also allowed to vary in order to maintain the original lift coefficient, giving a total of five design variables for this test case.

After ten design cycles, none of the design variables are against their bounds, and the drag coefficient is reduced from 0.0200 to 0.0184. As in the previous case, an exact timing for the computations is not known, although approximately six days were required. Clearly, efficient parallel implementations of both the flow and adjoint solvers would reduce this time significantly. For example, using only six processors, the time could be reduced to a single day. Density contours for the final design are shown in the lower portion of Fig. 8.4. It is apparent from the increased spacing between the contours that the strength of the shock at the midchord location is somewhat weaker for the final design which accounts for the lower drag. This can be seen in Fig. 8.5, where the pressure distribution is shown for the initial and final geometries. As stated above, the shock is not well-resolved due to the coarseness of the mesh (see e.g. Fig. 3.10), however the optimizer has successfully altered the configuration. Geometric cross-sections are shown in Fig. 8.6, where it can be seen that the optimizer has induced changes similar to the inviscid design case discussed in the preceding section. The wing thickness at each spanwise location has been reduced, and the lower surface has been modified in the leading edge region.

Chapter NINE

Summary and Concluding Remarks

Compressible and incompressible versions of a three-dimensional unstructured mesh Reynolds-averaged Navier-Stokes flow solver have been differentiated and the resulting derivatives have been verified by comparisons with finite differences and a complex-variable approach. In this implementation, the turbulence model is fully coupled with the flow equations in order to achieve this consistency. The accuracy demonstrated in the current work represents the first time that such an approach has been successfully implemented.

The accuracy of a number of simplifying approximations to the linearizations of the residual have been examined. A first-order approximation to the dependent variables in both the adjoint and design equations has been investigated. The effects of a “frozen” eddy viscosity and the ramifications of neglecting some mesh sensitivity terms were also examined. It has been found that none of the approximations yielded derivatives of acceptable accuracy and were often of incorrect sign. However, numerical experiments indicate that an incomplete convergence of the adjoint system often yields sufficiently accurate derivatives, thereby significantly lowering the time required for computing sensitivity information.

The convergence rate of the adjoint solver relative to the flow solver has been examined. Inviscid adjoint solutions typically require one to four times the cost of a flow solution, while for turbulent adjoint computations, this ratio can reach as high as eight to ten. Numerical experiments have shown that the adjoint solver can stall before converging the solution to machine accuracy, particularly for viscous cases. A possible remedy for this phenomenon would be to include the complete higher-order linearization in the preconditioning step, or to employ a simple form of mesh sequencing to obtain better approximations to the solution through the use of coarser meshes.

An efficient surface parameterization based on a free-form deformation technique has been utilized and the resulting codes have been integrated with an optimization package. Lastly, sample optimizations have been shown for inviscid and turbulent flow over an ONERA M6 wing. Drag reductions have been demonstrated by reducing shock strengths across the span of the wing.

In order for large scale optimization to become routine, the benefits of parallel architectures should be exploited. Although the flow solver has been parallelized using compiler directives, the parallel efficiency is under 50 percent. Clearly, parallel versions of the codes will have an immediate impact on the ability to design realistic configurations on fine meshes, and this effort is currently underway (see e.g. Ref. 65).

Further development of mesh movement strategies which enable large changes in the geometry are needed. Finally, another area that requires future work is the incorporation of multipoint optimization capability for designing geometries that perform well at off-design conditions.

References

[1] Hicks, R.M., and Henne, P.A., “Wing Design by Numerical Optimization,” *J. of Aircraft*, Vol. 15, 1978, pp. 407-412.

[2] Joh, C.-Y., Grossman, B., and Haftka, R.T., “Design Optimization of Transonic Airfoils,” *Eng. Opt.*, Vol. 21, 1993, pp. 1-20.

[3] Vanderplaats, G.N., Hicks, R.N., and Murman, E.M., “Application of Numerical Optimization Techniques to Airfoil Design,” NASA Conference on Aerodynamic Analysis Requiring Advanced Computers, Langley Research Center, Virginia, NASA SP-347, Part II, March 1975.

[4] Baysal, O., and Eleshaky, M.E., “Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations,” *J. of Fluids Engineering*, Vol. 113, 1991, pp. 681-688.

[5] Borggaard, J.T., Burns, J., Cliff, E.M., and Gunzburger, M.D., “Sensitivity Calculations for a 2-D Inviscid Supersonic Forebody Problem,” *Identification and Control Systems Governed by Partial Differential Equations*, SIAM Publications, Philadelphia, 1993, pp. 14-24.

[6] Burgreen, G.W., and Baysal, O., “Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods,” AIAA Paper No. 93-3322, 1993.

[7] Hou, G. J.-W., Maroju, V., Taylor, A.C., and Korivi, V.M., “Transonic Turbulent Airfoil Design Optimization with Automatic Differentiation in Incremental Iterative Forms,” AIAA Paper No. 95-1692, 1995.

[8] Newman, J.C., and Taylor, A.C., “Three-Dimensional Aerodynamic Shape Sensitivity Analysis and Design Optimization Using the Euler Equations on Unstructured Grids,” AIAA Paper No. 96-2464, 1996.

[9] Sherman, L.L., Taylor, A.C., Green, L.L., Newman, P.A., Hou, G.J.-W., and Korivi, V.M., “First- and Second-Order Aerodynamic Sensitivity Derivatives via Automatic Differentia-

tion with Incremental Iterative Methods," AIAA Paper No. 94-4262, 1994.

[10] Young, D.P., Huffman, W.P., Melvin, R.G., Bieterman, M.B., Hilmes, C.L., and Johnson, F.T., "Inexactness and Global Convergence in Design Optimization," AIAA Paper No. 94-4386, 1994.

[11] Lyness, J.N., "Numerical Algorithms Based on the Theory of Complex Variables," Proc. ACM 22nd Nat. Conf., Thomas Book Co., Washington, D.C., 1967, pp. 124-134.

[12] Lyness, J.N., and Moler, C.B., "Numerical Differentiation of Analytic Functions," *SIAM J. Numer. Anal*., Vol. 4, 1967, pp. 202-210.

[13] Squire, W., and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Rev.*, Vol. 10, No. 1, March 1998, pp. 110-112.

[14] Newman, J.C., Anderson, W.K., and Whitfield, D.L., "Multidisciplinary Sensitivity Derivatives Using Complex Variables," Mississippi State University Report No. MSSU-COE-ERC-98-08, 1998.

[15] Anderson, W.K., Whitfield, D.L., Newman III, J.C., and Nielsen, E.J., "Sensitivity Analysis for the Navier-Stokes Equations on Unstructured Meshes Using Complex Variables," in preparation.

[16] Pironneau, O., "On Optimum Profiles in Stokes Flow," *J. Fluid Mech.*, Vol. 59, 1973, pp. 117-128.

[17] Pironneau, O., "On Optimum Design in Fluid Mechanics," *J. Fluid Mech.*, Vol. 64, 1974, pp. 97-110.

[18] Glowinski, R., and Pironneau, O., "On the Numerical Computation of the Minimum-Drag Profile in Laminar Flow," *J. Fluid Mech.*, Vol. 72, 1975, pp. 385-389.

[19] Angrand, F., "Optimum Design for Potential Flows," *Int. J. Num. Methods in Fluids*, Vol. 3, 1983, pp. 265-282.

[20] Jameson, A., "Aerodynamic Design Via Control Theory," *J. Scientific Computing*, Vol. 3, 1988, pp. 233-260.

[21] Frank, P.D., and Shubin, G.R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem," *J. Comp. Phys.*, Vol. 98, 1992, pp. 74-89.

[22] Beux, F., and Dervieux, A., "Exact-Gradient Shape Optimization of a 2-D Euler Flow," *Finite Elements in Analysis and Design*, Vol. 12, 1992, pp. 281-302.

[23] Iollo, A., and Salas, M.D., "Contribution to the Optimal Shape Design of Two-Dimensional

Internal Flows with Embedded Shocks," ICASE Report No. 95-20, 1995.

[24] Cabuk, H., and Modi, V., "Optimum Plane Diffusers in Laminar Flow," *J. Fluid Mech.*, Vol. 237, 1992, pp. 373-393.

[25] Huan, J., and Modi, V., "Optimum Design of Minimum Drag Bodies in Incompressible Laminar Flow Using a Control Theory Approach," *Inverse Problems in Engineering,* Vol. 1, 1994, pp. 1-25.

[26] Reuther, J., Jameson, A., Farmer, J., Martinelli, L., and Saunders, D., "Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA Paper No. 96-0094, 1996.

[27] Anderson, W.K., and Bonhaus, D.L., "Aerodynamic Design on Unstructured Grids for Turbulent Flows," NASA TM 112867, June 1997.

[28] Elliott, J., and Peraire, J., "Progress Towards a 3D Aerodynamic Shape Optimization Tool for the Compressible, High-Re Navier-Stokes Equations Discretized on Unstructured Meshes," AIAA Paper No. 98-2897, 1998.

[29] Jameson, A., Alonso, J.J., Reuther, J., Martinelli, L., and Vassberg, J.C., "Aerodynamic Shape Optimization Techniques Based on Control Theory," AIAA Paper No. 98-2538, 1998.

[30] Soemarwoto, B., "Multipoint Aerodynamic Design by Optimization," Ph.D. Thesis, Delft University of Technology, 1996.

[31] Biedron, R.T., Samareh, J.A., and Green, L.L., "Parallel Computation of Sensitivity Derivatives with Application to Aerodynamic Optimization of a Wing," Presented at HPCCP/CAS Workshop '98, NASA Ames Research Center, August 25-27, 1998.

[32] Mohammadi, B., "Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence," AIAA Paper No. 97-0099, 1997.

[33] Carle, A., Fagan, M., and Green, L.L., "Preliminary Results from the Application of Automated Adjoint Code Generation to CFL3D," AIAA Paper No. 98-4807, 1998.

[34] Anderson, W.K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper No. 97-0643, 1997.

[35] White, F.M., Viscous Fluid Flow, McGraw-Hill, New York, 1974.

[36] Chorin, A.J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *J. Comp. Phys.*, Vol. 2, 1967, pp. 12-26.

[37] Anderson, W.K., Rausch, R.D., and Bonhaus, D.L., "Implicit/Multigrid Algorithms for In-

compressible Turbulent Flows on Unstructured Grids," *J. Comp. Phys.*, Vol. 128, 1996, pp. 391-408.

[38] Spalart, P.R., and Allmaras, S.R., "A One-Equation Turbulence Model for Aerodynamic Flows," *La Recherche Aerospatiale*, Vol. 1, 1994, pp. 5-21.

[39] Anderson, W.K., and Bonhaus, D.L., "Navier-Stokes Computations and Experimental Comparisons for Multielement Airfoil Configurations," AIAA Paper No. 93-0645, 1993.

[40] Rumsey, C.L., Gatski, T.B., Ying, S.X., and Bertelrud, A., "Prediction of High-Lift Flows Using Turbulent Closure Models," AIAA Paper No. 97-2260, 1997.

[41] Potter, R.C., van Dam, C.P., and Hardin, J.D., "Viscous-Flow Analysis of a Subsonic Transport High-Lift System Including Comparisons with Flight-Measured Results," AIAA Paper No. 95-0043, 1995.

[42] Anderson, W.K., and Bonhaus, D.L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No.1, 1994, pp. 1-21.

[43] Bonhaus, D.L., "An Upwind Multigrid Method for Solving Viscous Flows on Unstructured Triangular Meshes," M.S. Thesis, George Washington University, 1993.

[44] Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *J. Comp. Phys.*, Vol. 43, No. 2, 1981, pp. 357-372.

[45] Van Leer, B., "Flux Vector Splitting for the Euler Equations," Lecture Notes in Physics, Vol. 170, 1982, pp. 501-512.

[46] Chakravarthy, S.R., and Osher, S., "Numerical Experiments with the Osher Upwind Scheme for the Euler Equations," *AIAA Journal*, Vol. 21, No. 9, 1983, pp. 1241-1248.

[47] Hinson, B., and Burdges, K., "Acquisition and Application of Transonic Wing and Far-Field Test Data for Three-Dimensional Computational Method Evaluation, Volume II — Appendix B, Experimental Data," Air Force Office of Scientific Research, AFOSR-TR-80-0422, March 1980.

[48] Lin, J.C., and Dominick, C.J., "Optimization of an Advanced Design Three-Element Airfoil at High Reynolds Numbers," AIAA Paper No. 95-1858, 1995.

[49] Nakayama, A., "Flowfield Survey Around High-Lift Airfoil Model LB 546," MDC Report No. J4827, February 1987.

[50] Pirzadeh, S., "Viscous Unstructured Three-Dimensional Grids by the Advancing-Layers Method," AIAA Paper No. 94-0417, 1994.

[51] Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA-M6 Wing at Tran-

sonic Mach Numbers," Experimental Data Base for Computer Program Assessment, AGARD-AR-138, May 1979, pp. B1-1-B1-44.

[52] Krist, S.L., Biedron, R.T., and Rumsey, C.L., "CFL3D Users Manual (Version 5.0)," NASA TM-1998-208444, 1998.

[53] Marcum, D.L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper No. 95-0212, 1995.

[54] Storms, B.L., Takahashi, T.T., and Ross, J.C., "Aerodynamic Influence of a Finite-Span Flap on a Simple Wing," SAE Paper 951977, 1995.

[55] Saad, Y., and Schultz, M.H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, 1986, pp. 856-869.

[56] Wigton, L.B., Young, D.P., and Yu, N.J., "GMRES Acceleration of Computational Fluid Dynamics Codes," AIAA Paper No. 85-1494, 1985.

[57] Hackbusch, W., Iterative Solution of Large Sparse Systems of Equations, Springer-Verlag, New York, 1994.

[58] Samareh, J., "Geometry Modeling and Grid Generation for Design and Optimization," ICASE/LaRC/NSF/ARO Workshop on Computational Aerosciences in the 21st Century, April 22-24, 1998.

[59] Venkatakrishnan, V., and Mavriplis, D.J., "Implicit Method for the Computation of Unsteady Flows on Unstructured Grids," *J. Comp. Phys.*, Vol. 127, 1996, pp. 380-397.

[60] Williams, B.R., "An Exact Test Case for the Plane Potential Flow About Two Adjacent Lifting Aerofoils," R. & M. No. 3717, British Aeronautical Research Counci, 1973.

[61] Cook, P., McDonald, M., and Firmin, M., "Airfoil RAE 2822—Pressure Distributions and Boundary Layer Wake Measurement," AGARD AR-138, Paper A6, 1979.

[62] Davidon, W.C., "Variable Metric Method for Minimization," AEC Research and Development Report, ANL-5990, 1959.

[63] Fletcher, R., and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization," *The Computer Journal*, Vol. 6, July 1963, pp. 163-168.

[64] Wrenn, G.A., "An Indirect Method for Numerical Optimization Using the Kresselmeier-Steinhauser Function," NASA CR 4220, March 1989.

[65] Kaushik, D.K., Keyes, D.E., and Smith, B.F., "On the Interaction of Architecture and Algorithm in the Domain-Based Parallelization of an Unstructured Grid Incompressible Flow

Code," in Proceedings of the 10th Intl. Conf. on Domain Decomposition Methods, J. Mandel et al., eds., AMS, pp. 311-319.

Tables

Table 5.1. Accuracy of two-dimensional derivatives for lift coefficient

	Finite Difference	Adjoint	% Diff.
M_∞	0.30965	0.30962	-0.010
α	0.12285	0.12282	-0.024
Point A	-1.0952	-1.0952	0.000
Point B	0.57480	0.57480	0.000
Point C	-2.1368	-2.1366	-0.009
Point D	0.76215	0.76215	0.000

Table 5.2. Accuracy of two-dimensional derivatives for drag coefficient

	Finite Difference	Adjoint	% Diff.
M_∞	-0.05029	-0.05029	0.000
α	0.00843	0.00843	0.000
Point A	0.21925	0.21925	0.000
Point B	-0.03489	-0.03489	0.000
Point C	0.17007	0.17007	0.000
Point D	0.06447	0.06448	0.016

Table 5.3. Accuracy of two-dimensional derivatives of lift coefficient for flap translation

	Finite Difference	Adjoint	% Diff.
x-translation	1.4226	1.4232	0.042
y-translation	-6.8991	-6.8990	-0.001

Table 5.4. Accuracy of two-dimensional derivatives of drag coefficient for flap translation

	Finite Difference	Adjoint	% Diff.
x-translation	0.02710	0.02716	0.221
y-translation	-0.24164	-0.24163	-0.004

Table 5.5. Sensitivity derivatives for lift coefficient for RAE 2822 airfoil

	Finite Difference	Adjoint	% Diff.
M_∞	-3.0546	-3.0546	0.000
α	5.7614	5.7615	0.002
Point A	7.9826	7.9814	-0.015
Point B	1.9247	1.9219	-0.145
Point C	1.3283	1.3283	0.000

Table 5.6. Three-dimensional compressible derivatives

	Finite Difference	Adjoint	% Diff.
Mach Number	0.00960	0.00959	-0.104
α	-0.03243	-0.03243	0.000
Twist #3	0.00965	0.00965	0.000
Shear #3	-0.04275	-0.04277	0.047
Thickness #3	-0.04011	-0.04012	0.025
Camber #4	-1.3174	-1.3174	0.000

Table 5.7. Derivatives of lift coefficient computed using various methods

Design Variable	Finite Difference	Adjoint	Complex Variables
Camber #4	1.409643	1.409592	1.409592
Thickness #3	0.041174	0.041195	0.041194
Twist #3	-0.010392	-0.010372	-0.010372
Shear #3	0.045804	0.045844	0.045844

Table 5.8. Three-dimensional incompressible derivatives

	Finite Difference	Adjoint	% Diff.
α	-0.00307	-0.00307	0.000
Twist #3	0.00922	0.00922	0.000
Shear #3	-0.02648	-0.02648	0.000
Thickness #3	-0.04399	-0.04400	0.023
Camber #4	-1.2594	-1.2594	0.000

Table 6.1. Sensitivity derivatives for lift coefficient using various approximations (see Fig. 5.2)

	Finite Difference	Adjoint	% Diff.
Exact Linearization			
Point A	-1.0952	-1.0952	0.000
Point B	0.57480	0.57480	0.000
Point C	-2.1368	-2.1366	-0.009
Point D	0.76215	0.76215	0.000
x-translation (flap)	1.4226	1.4232	0.042
y-translation (flap)	-6.8991	-6.8990	-0.001
First-Order Adjoint			
Point A	-1.0952	0.34633	-132
Point B	0.57480	0.47104	-18.1
Point C	-2.1368	-0.10590	-95.0
Point D	0.76215	0.70373	-7.67
x-translation (flap)	1.4226	-0.68201	-148
y-translation (flap)	-6.8991	0.14801	-102
Constant μ_t			
Point A	-1.0952	-1.6844	53.8
Point B	0.57480	0.37262	-35.2
Point C	-2.1368	-2.2888	7.11
Point D	0.76215	0.57459	-24.6
x-translation (flap)	1.4226	1.1909	-16.3
y-translation (flap)	-6.8991	-7.6163	10.4

Figures

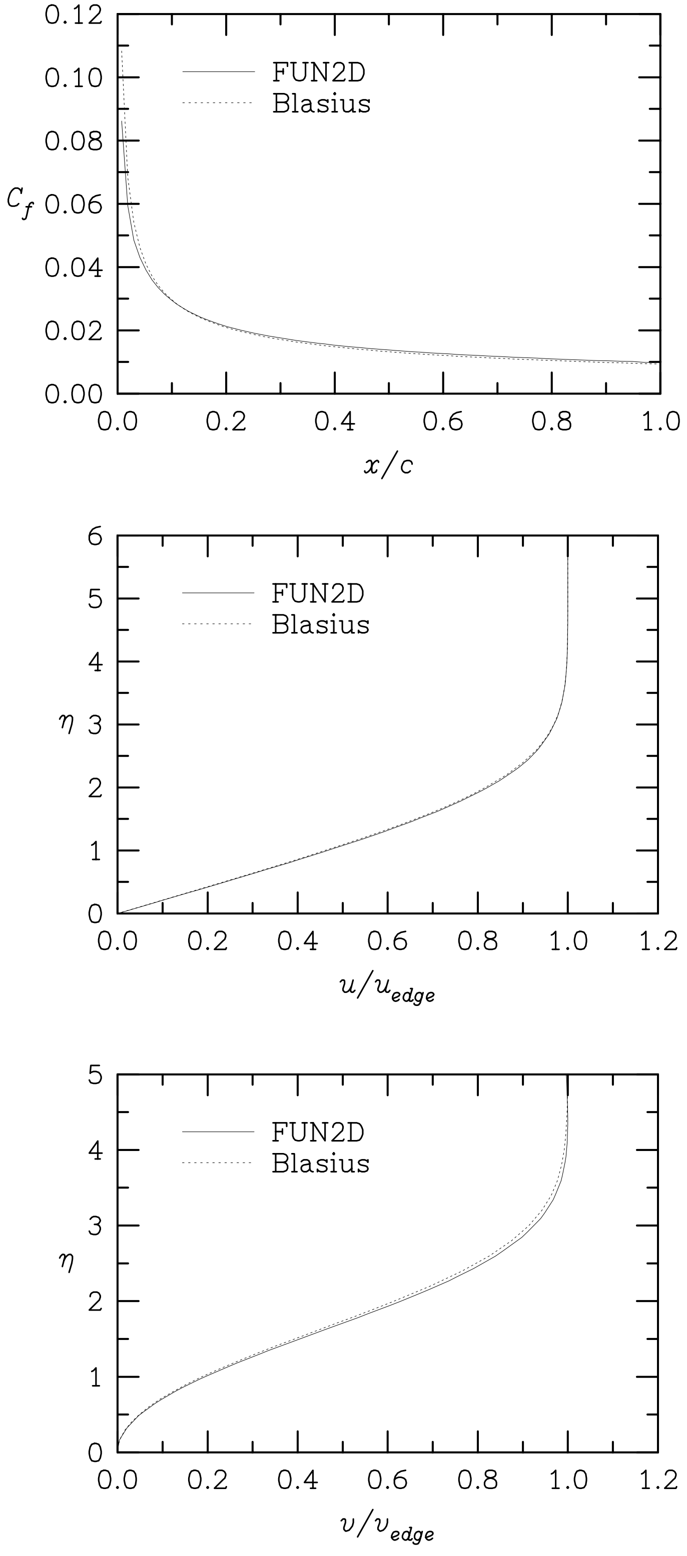

Figure 3.1. Comparison of skin friction and velocity profiles for flat plate.

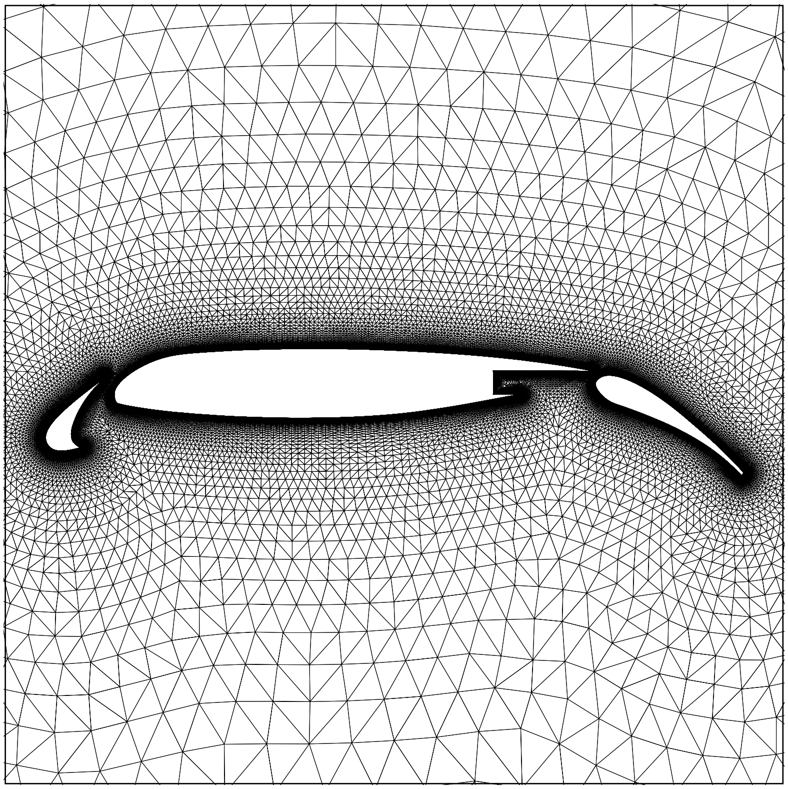

Figure 3.2. Grid used for AEET airfoil computations.

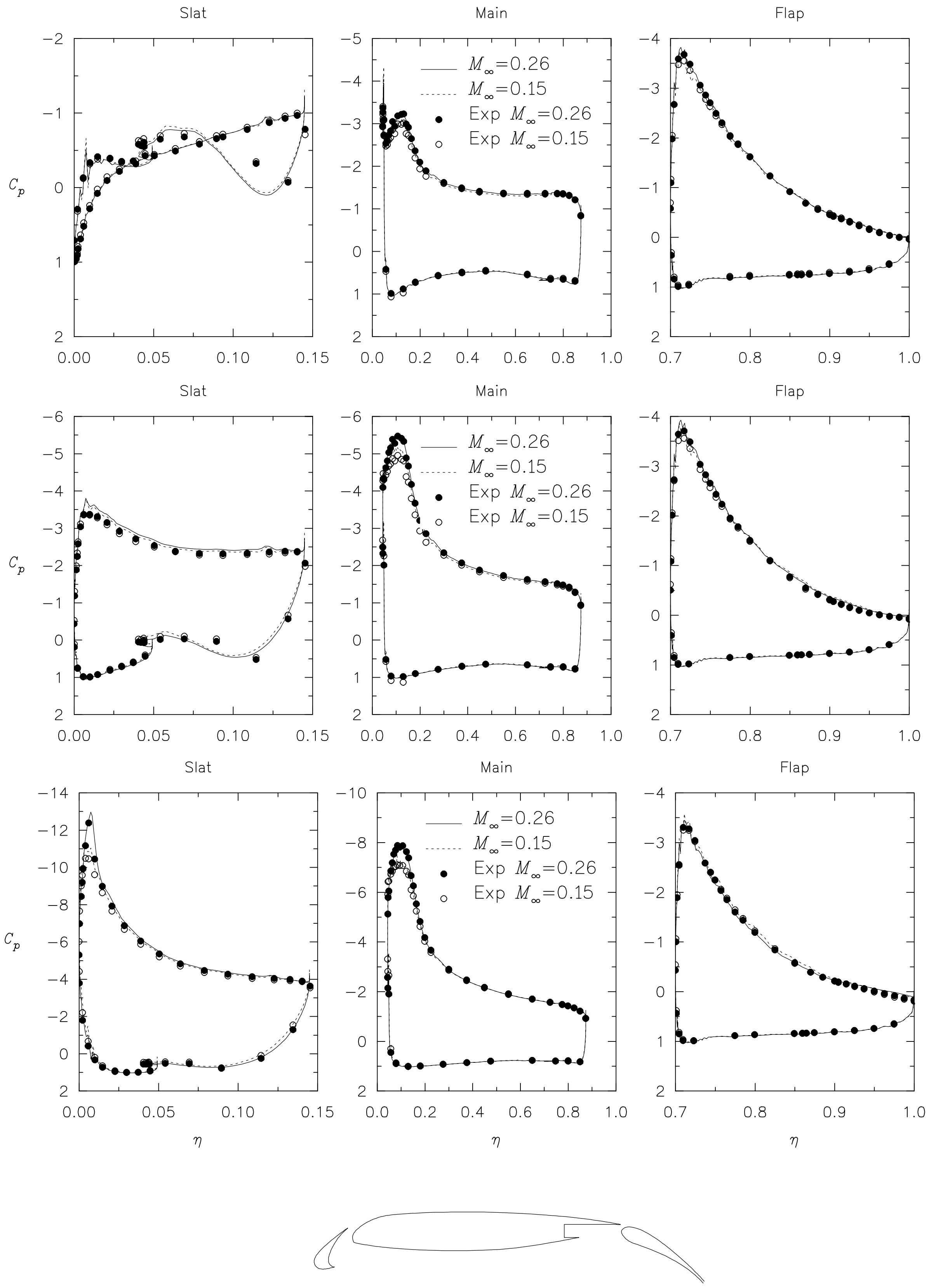

Figure 3.3. Pressure distributions for AEET airfoil.

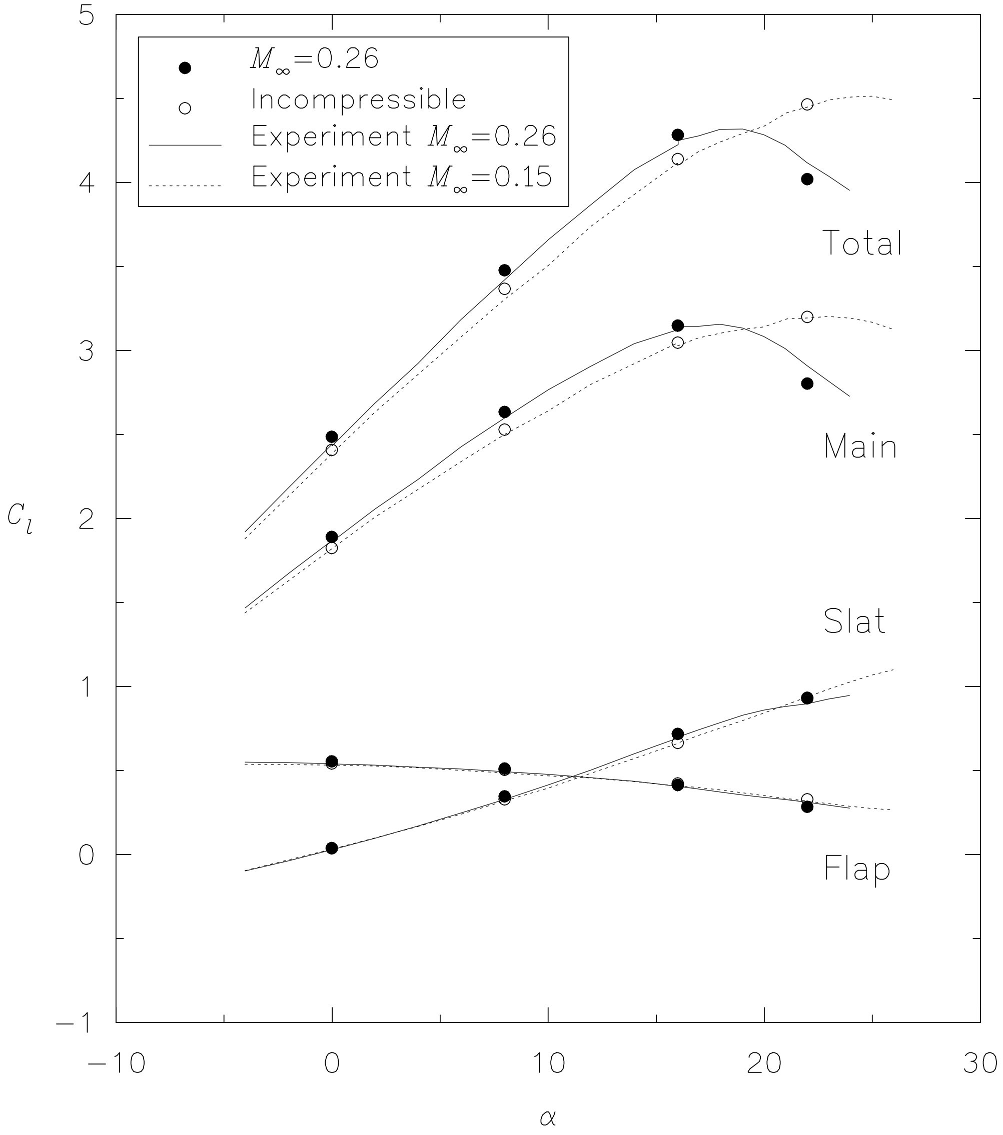

Figure 3.4. Lift versus angle of attack for computations and experiment.

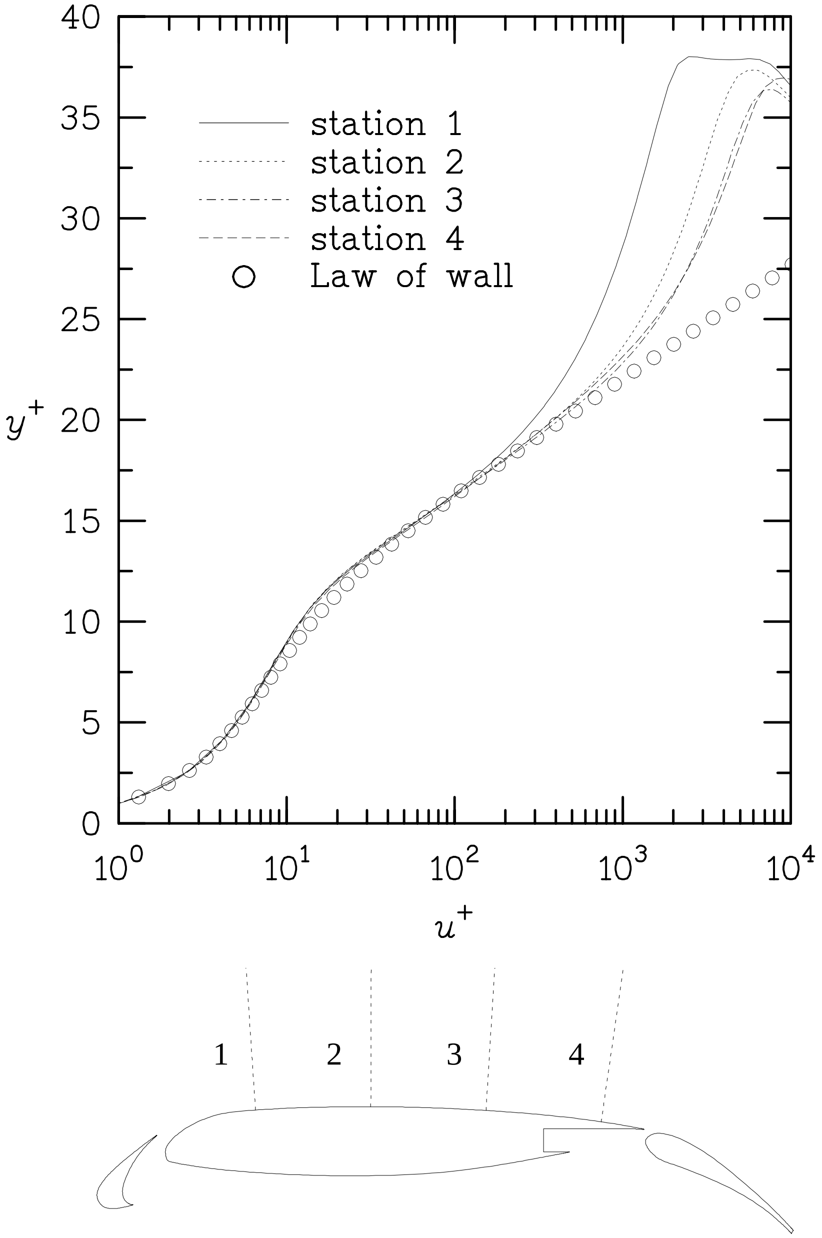

Figure 3.5. u^+ vs y^+ for the AEET airfoil.

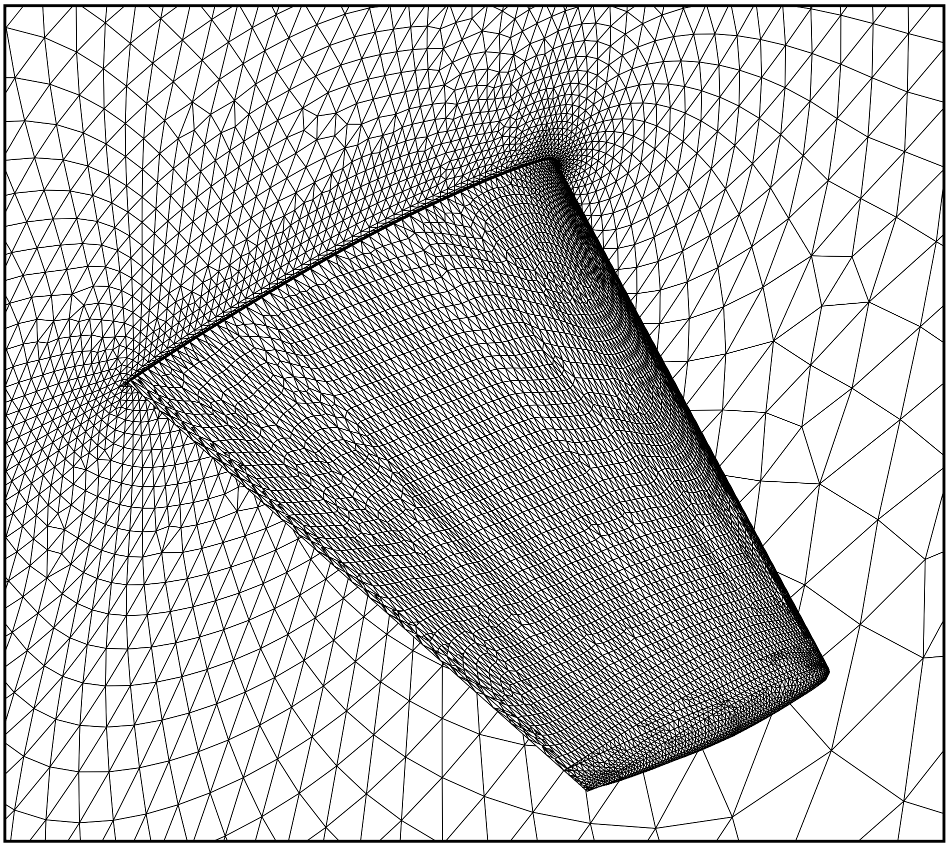

Figure 3.6. View of unstructured surface mesh used for ONERA M6 test case.

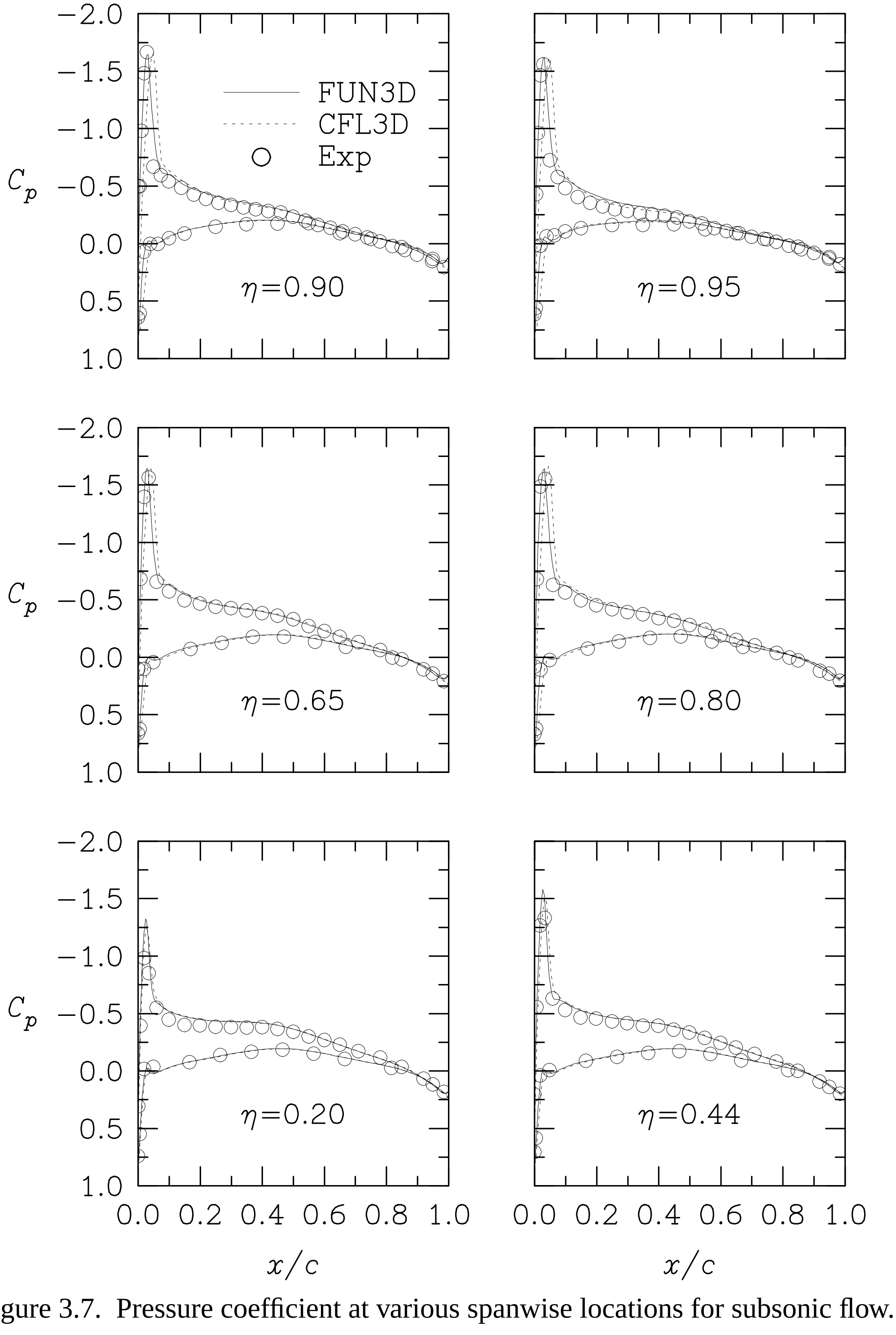

Figure 3.7. Pressure coefficient at various spanwise locations for subsonic flow.

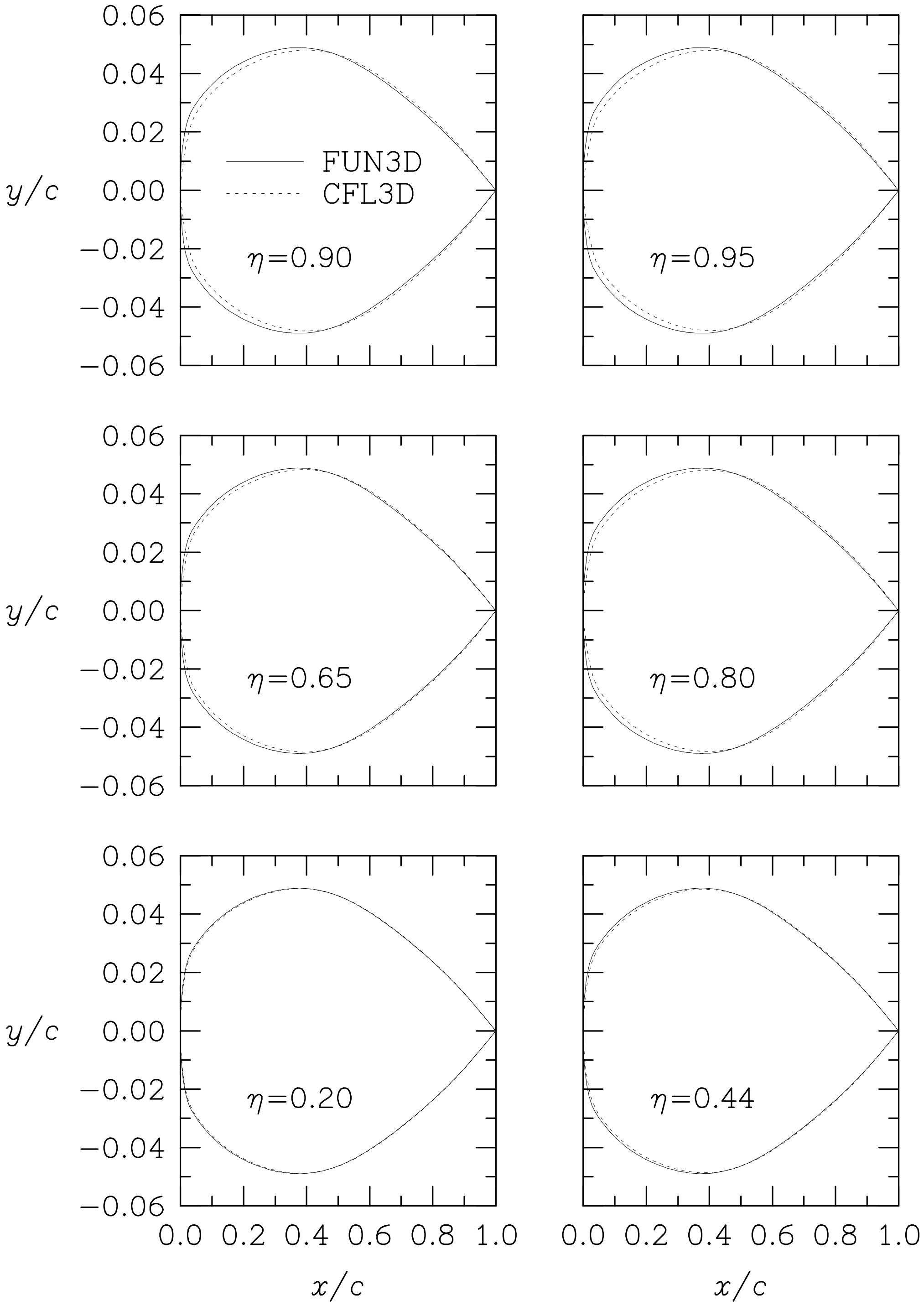

Figure 3.8. Geometric discrepancies for the structured and unstructured meshes.

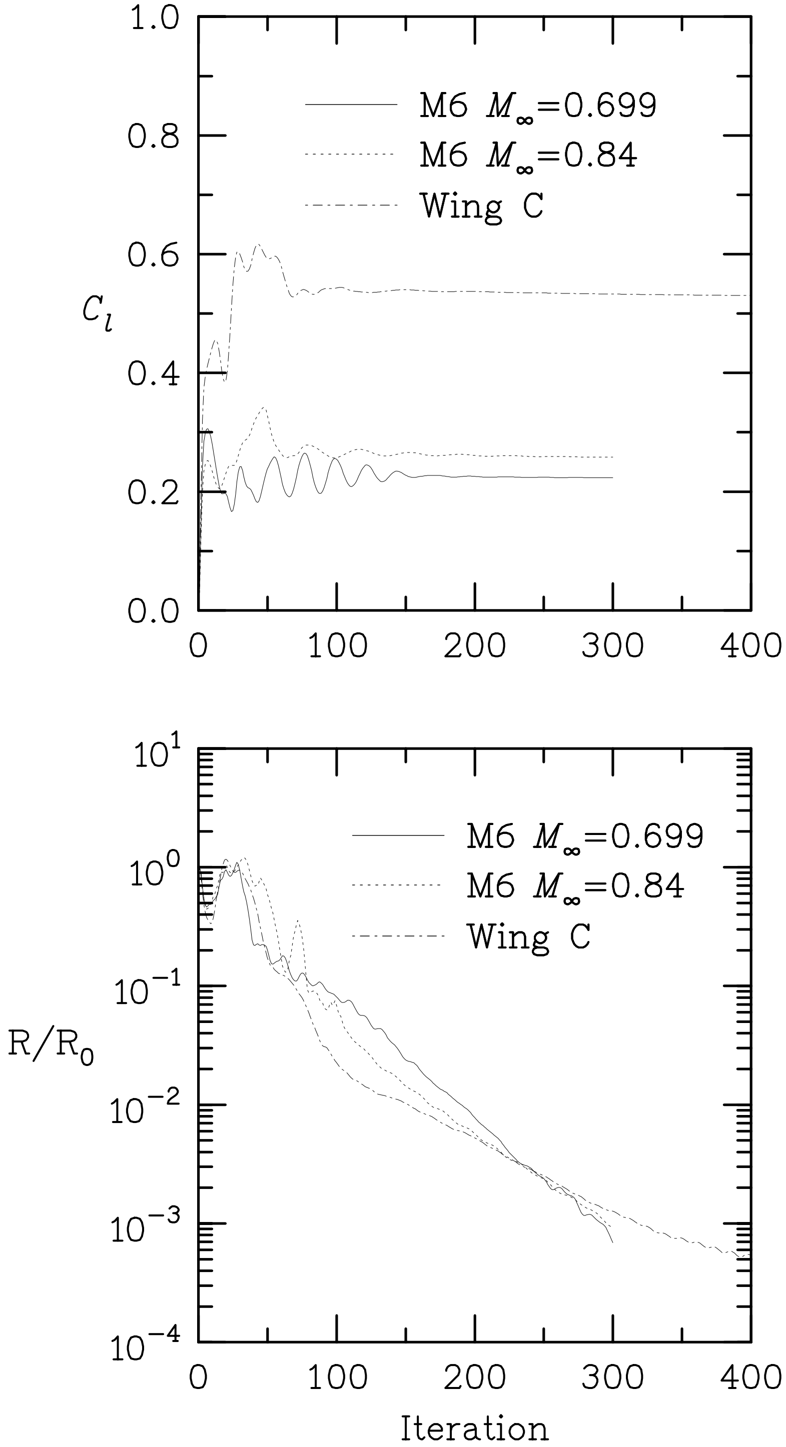

Figure 3.9. Convergence histories for the ONERA M6 and Lockheed Wing C cases.

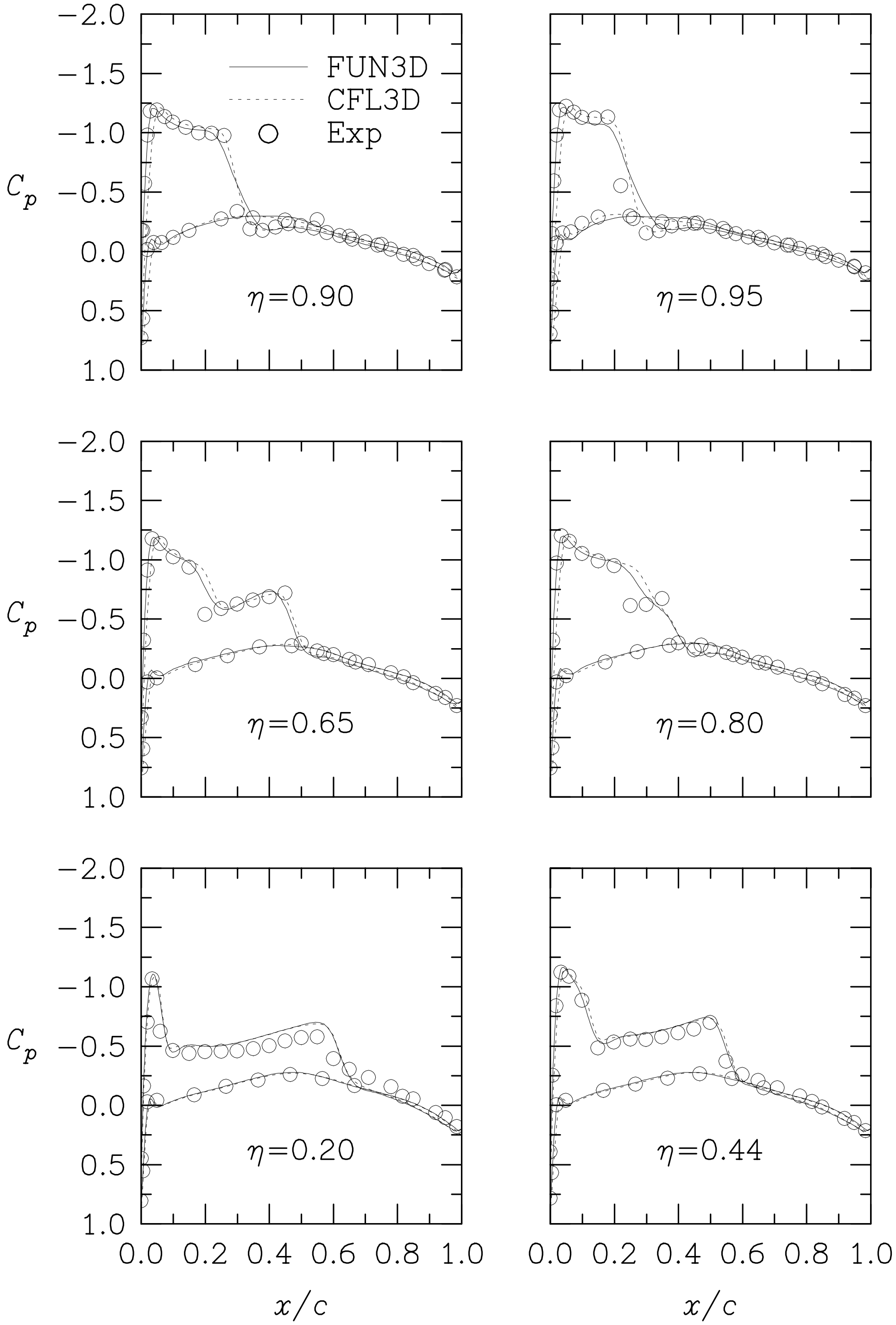

Figure 3.10. Pressure coefficient at various spanwise locations for transonic flow.

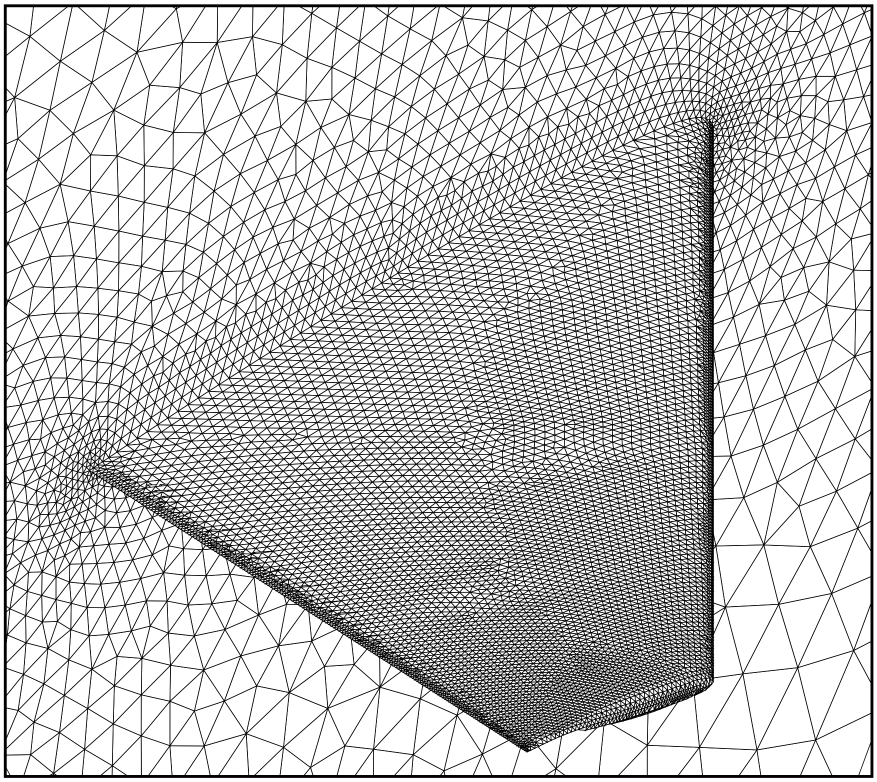

Figure 3.11. Surface mesh for the Lockheed Wing C configuration.

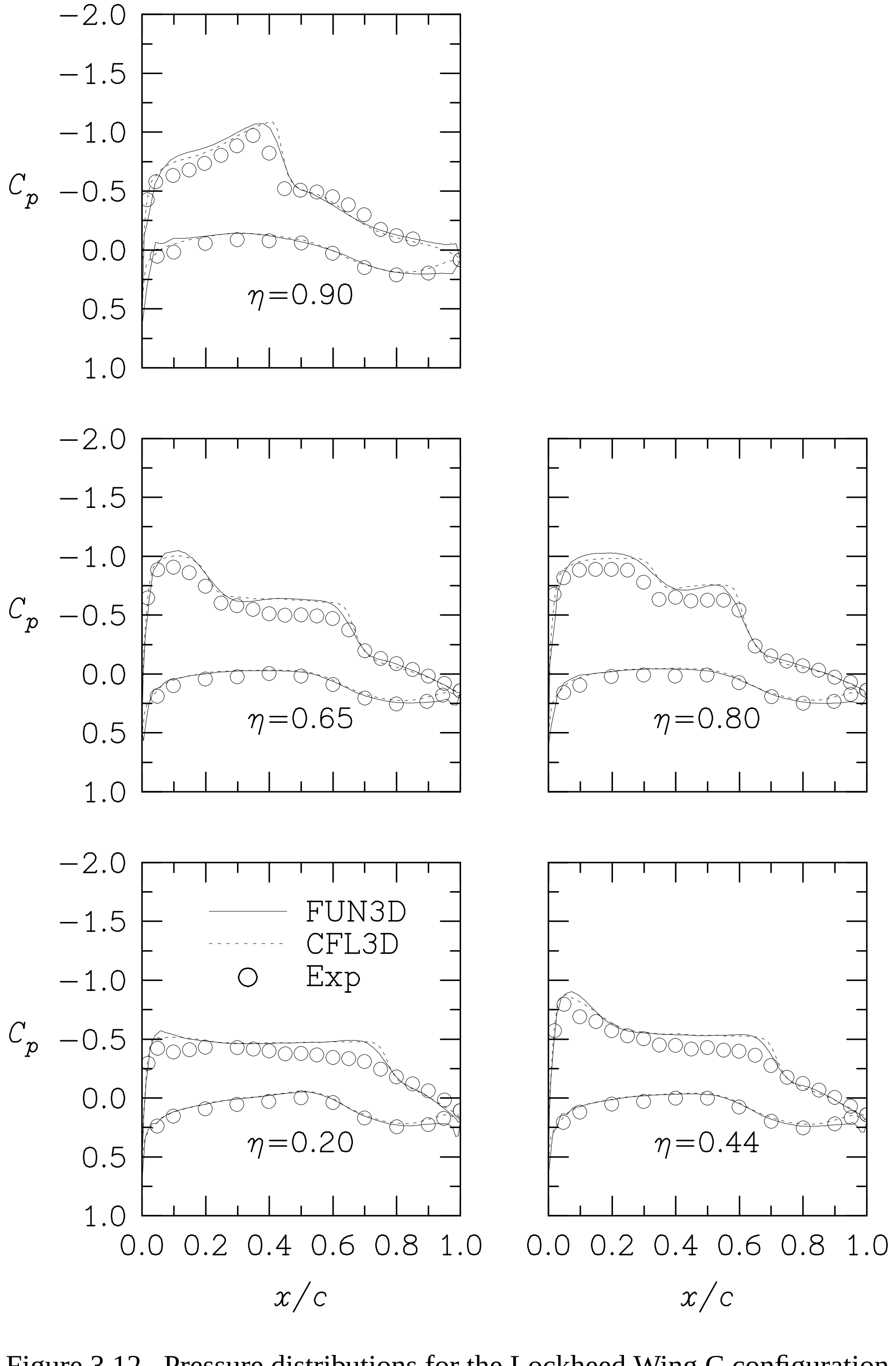

Figure 3.12. Pressure distributions for the Lockheed Wing C configuration.

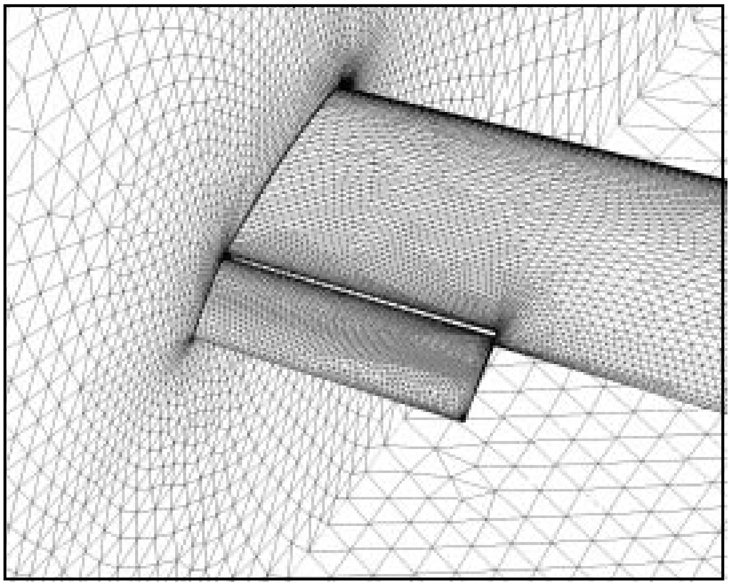

Figure 3.13. Surface mesh for partial-span flap computation.

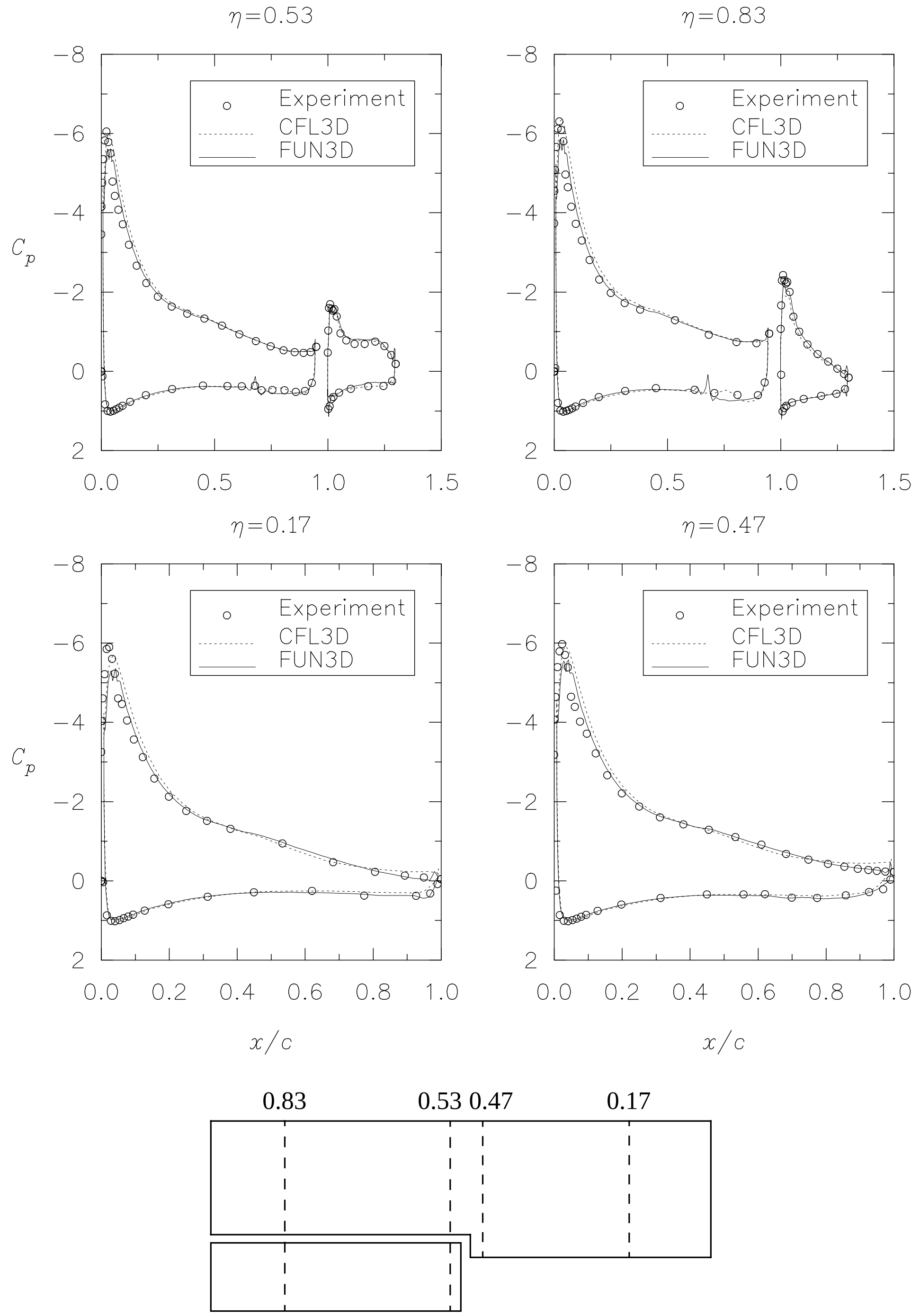

Figure 3.14. Partial-span flap results.

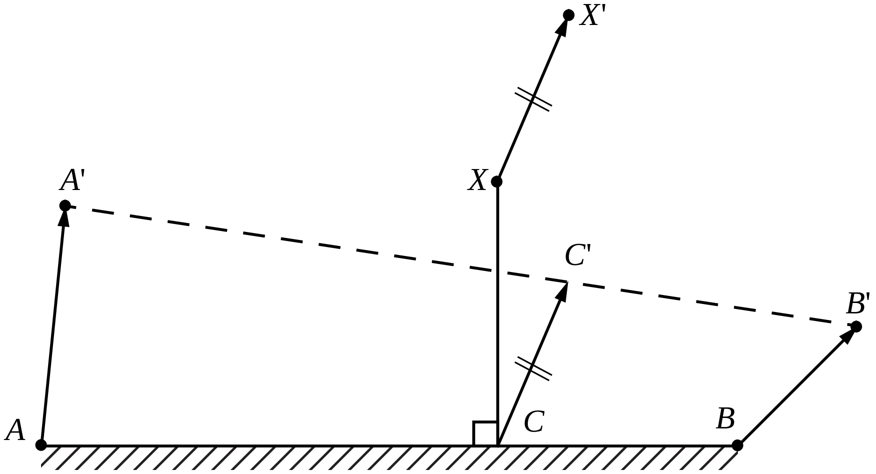

Figure 4.1. Illustration of viscous mesh movement procedure in the boundary layer region.

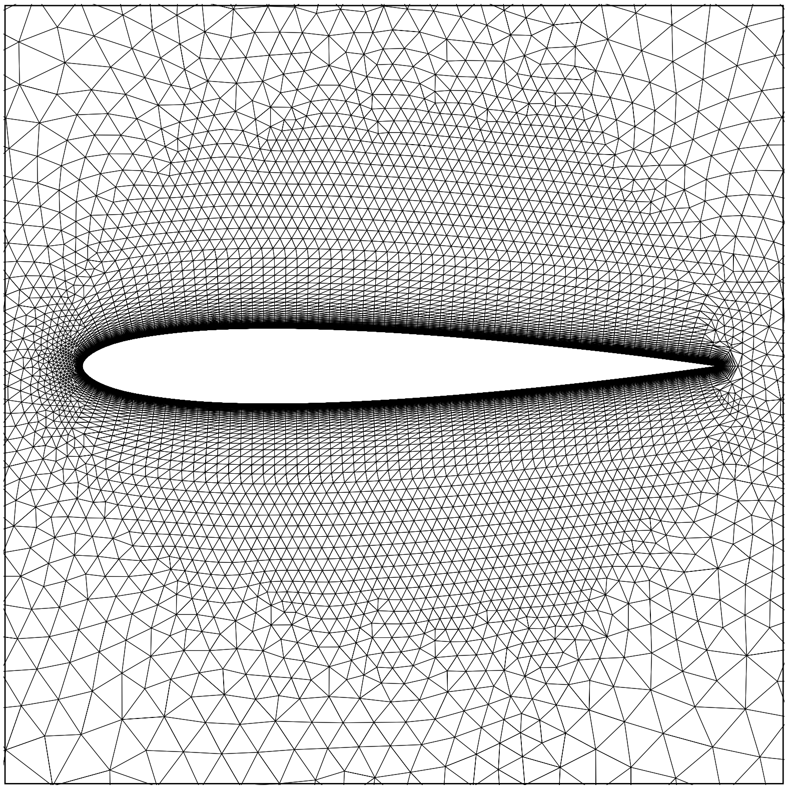

Figure 4.2. Baseline viscous mesh for NACA 0012 airfoil.

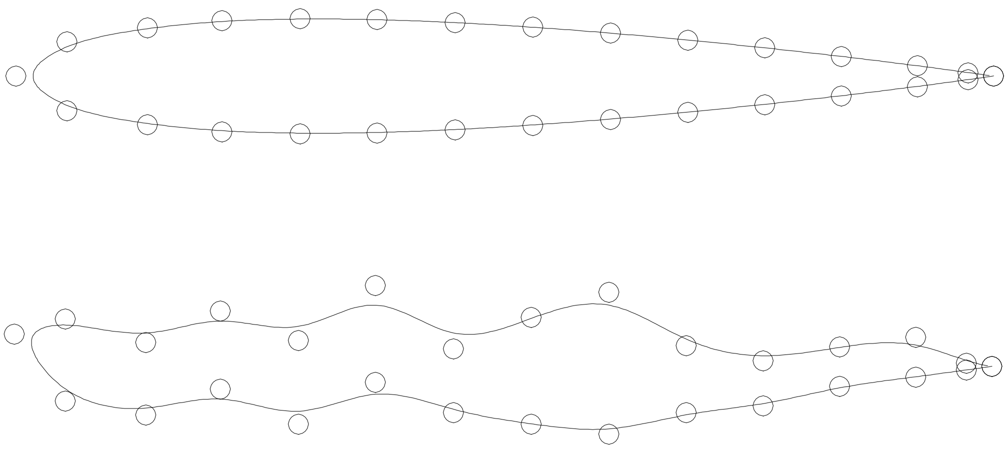

Figure 4.3. Control point locations for NACA 0012 airfoil before and after perturbations.

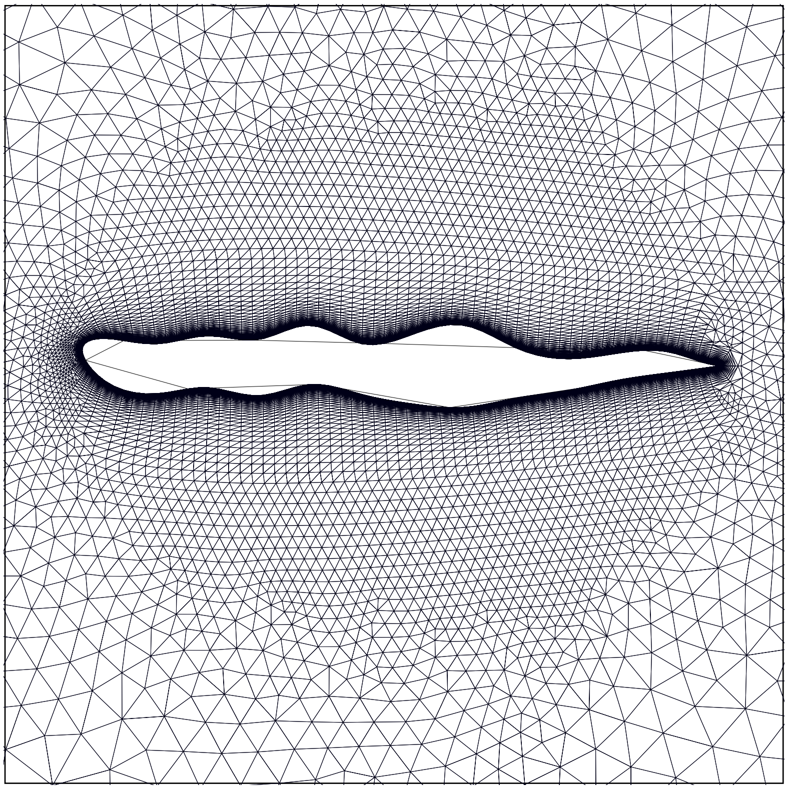

Figure 4.4. Perturbed viscous mesh for NACA 0012 airfoil.

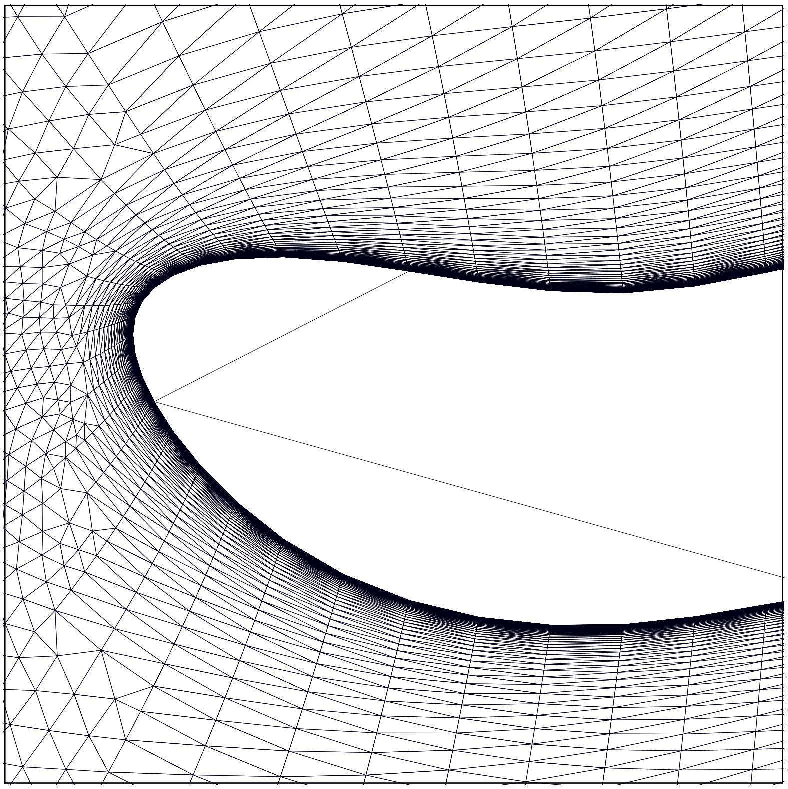

Figure 4.5. Close-up of nose region for perturbed mesh.

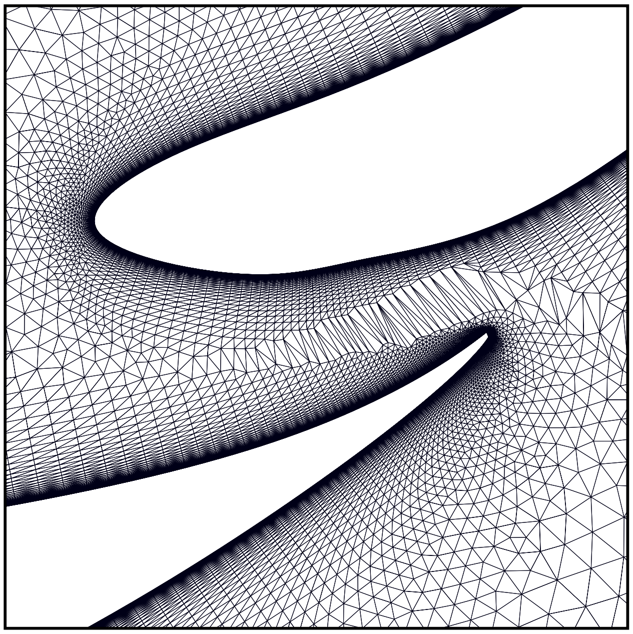

Figure 4.6. Region between elements where “gaps” in mesh can occur during translation.

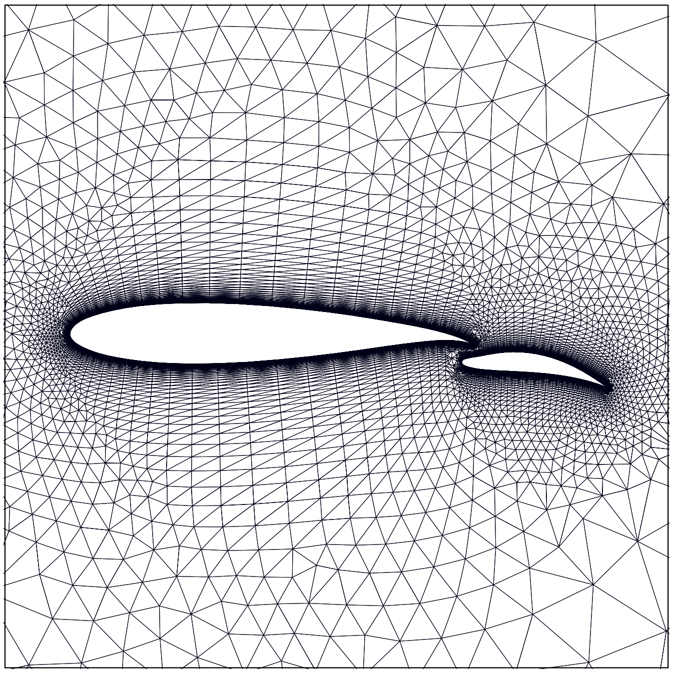

Figure 5.1. Mesh used in assessment of two-dimensional design sensitivities.

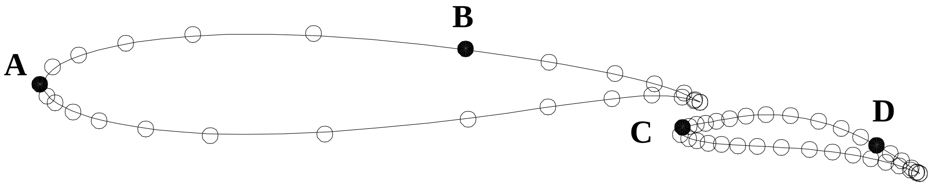

Figure 5.2. Location of design variables for 2-element airfoil.

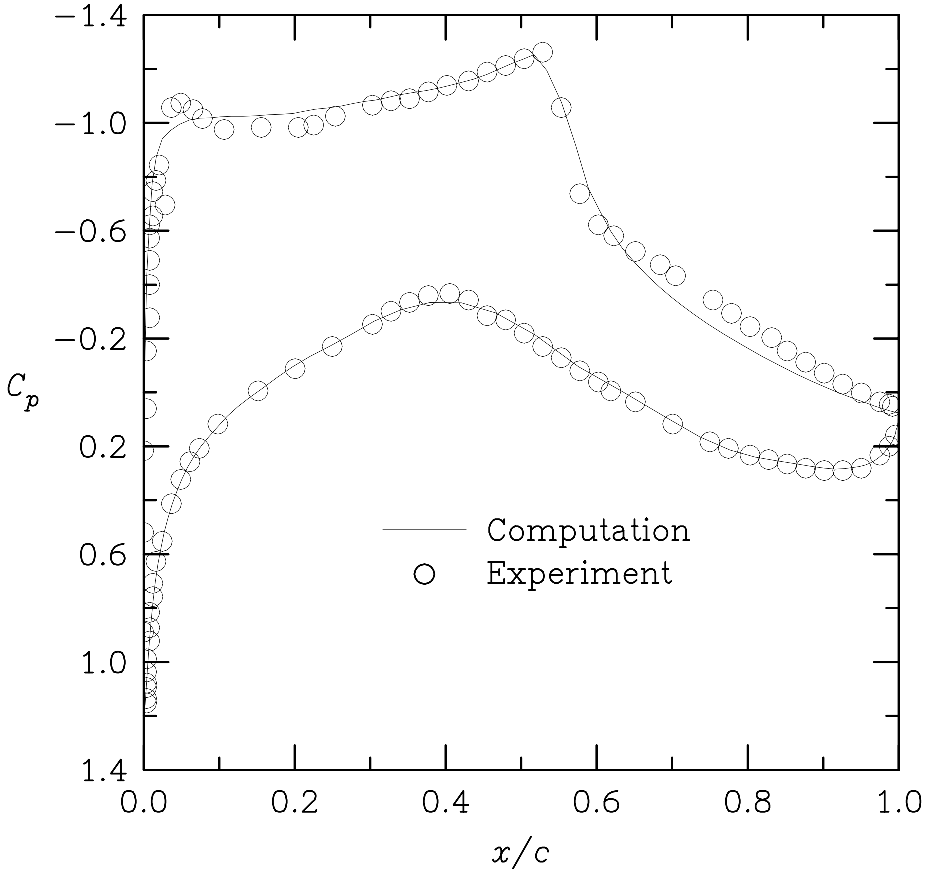

Figure 5.3. Pressure distribution for transonic RAE 2822 airfoil.

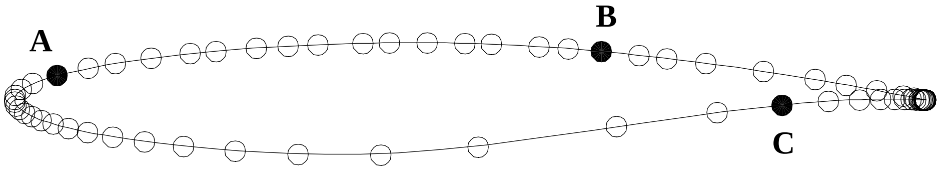

Figure 5.4. Location of design variables for RAE 2822 airfoil.

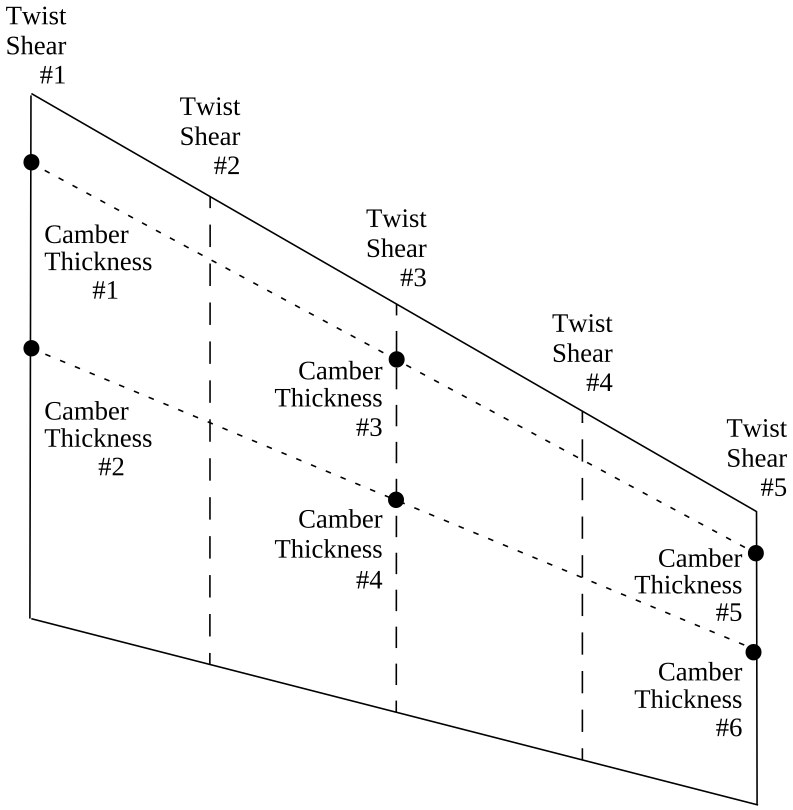

Figure 5.5. Location of design variables for ONERA M6 wing.

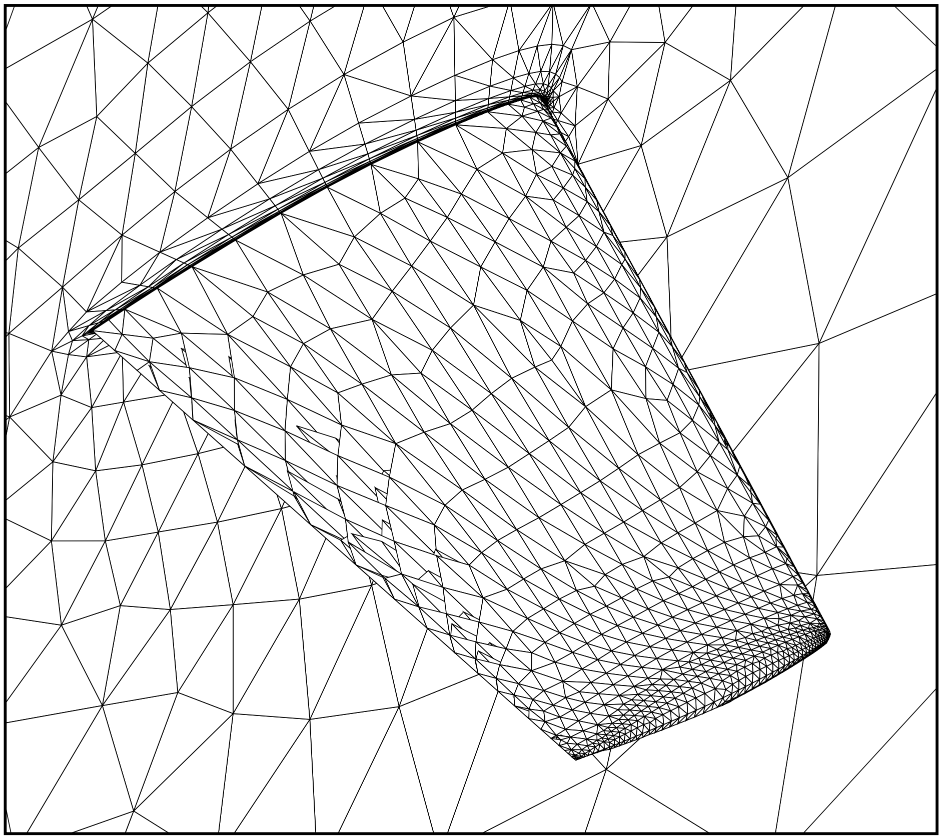

Figure 5.6. Grid used for assessment of three-dimensional design sensitivities.

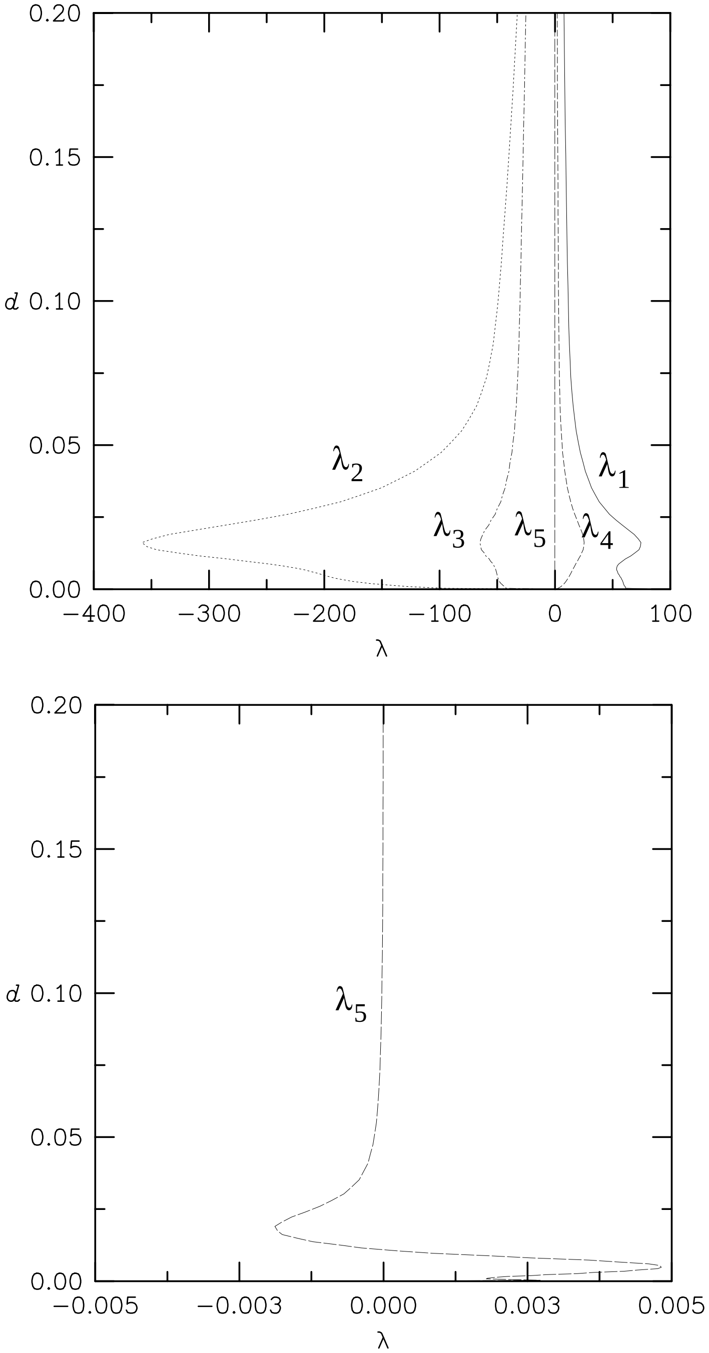

Figure 6.1. Variations of the costate variables with distance from the surface ($x/c = 0.2$).

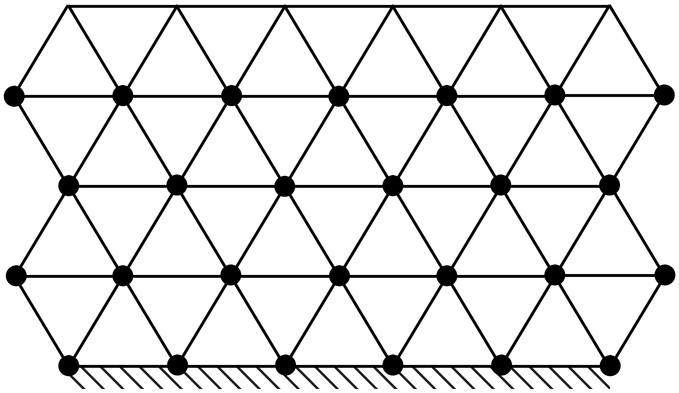

Figure 6.2. Example of three “tagged” grid layers.

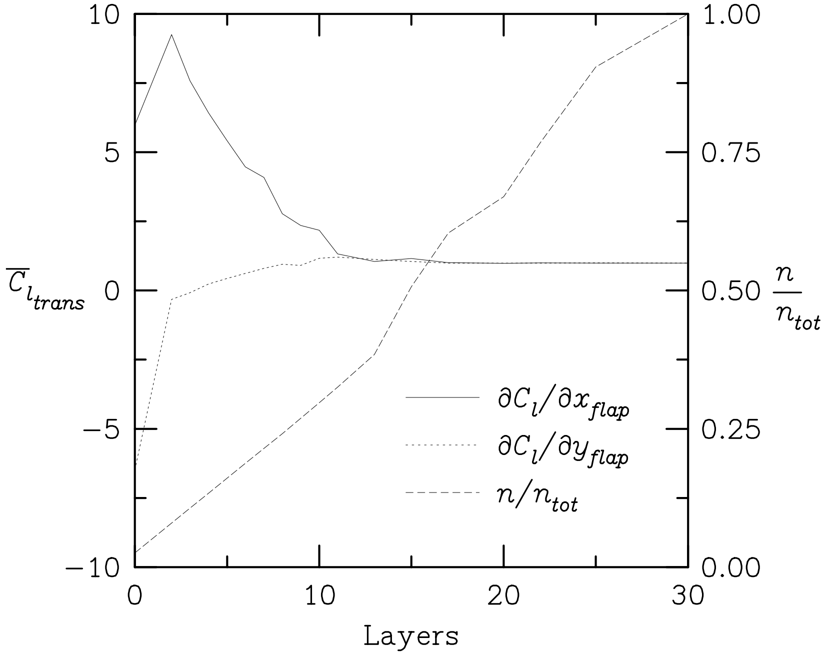

Figure 6.3. Extent of mesh sensitivity terms required for translation sensitivity accuracy.

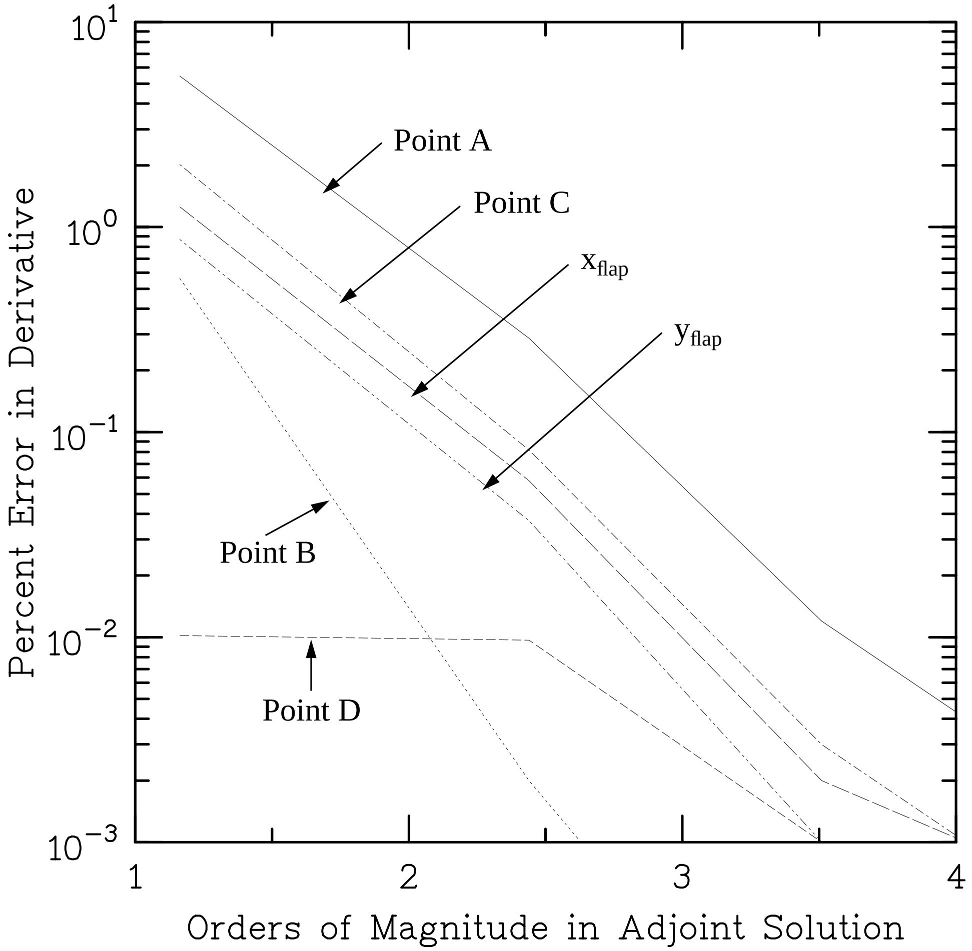

Figure 6.4. Convergence level of the adjoint solution necessary for derivative accuracy.

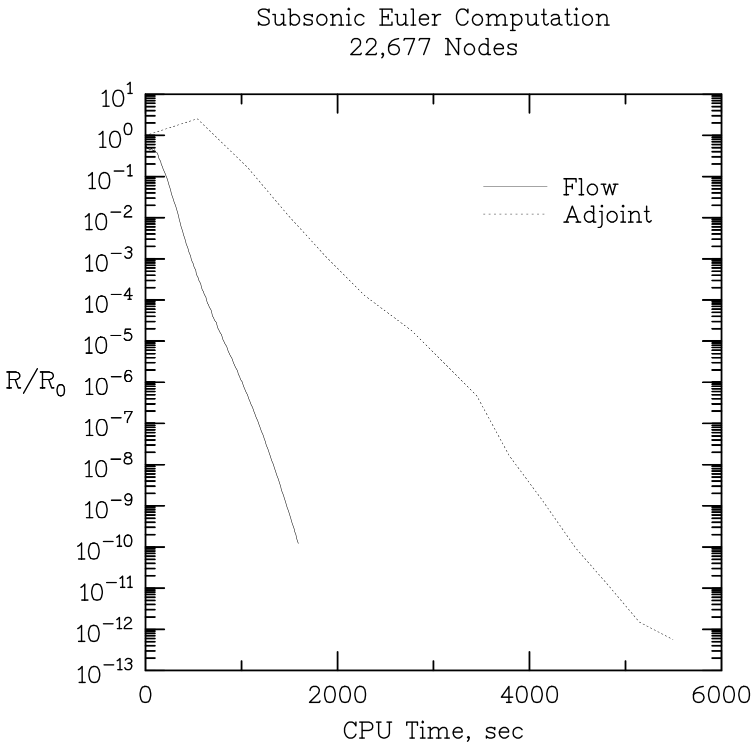

Figure 7.1. Convergence histories for subsonic inviscid flow on coarse mesh.

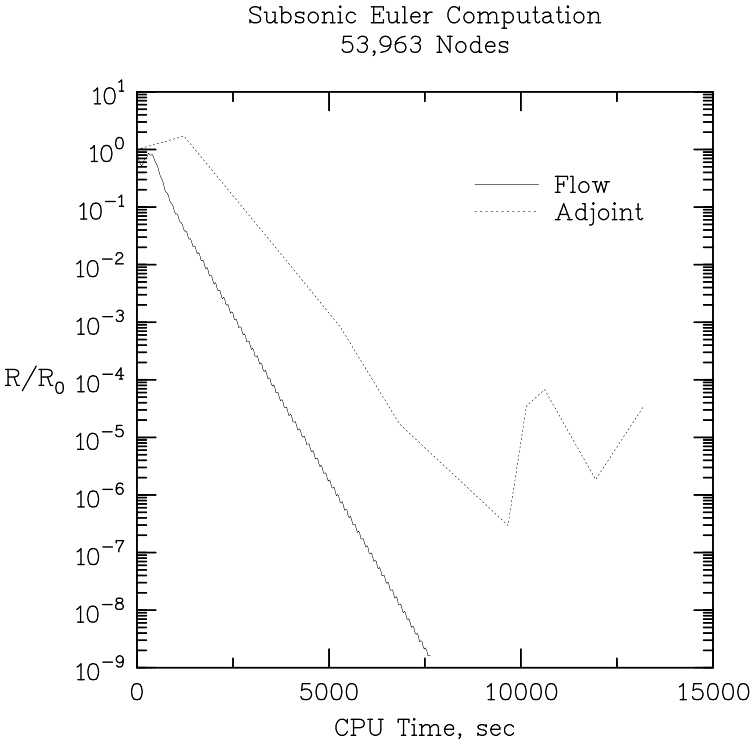

Figure 7.2. Convergence histories for subsonic inviscid flow on medium mesh.

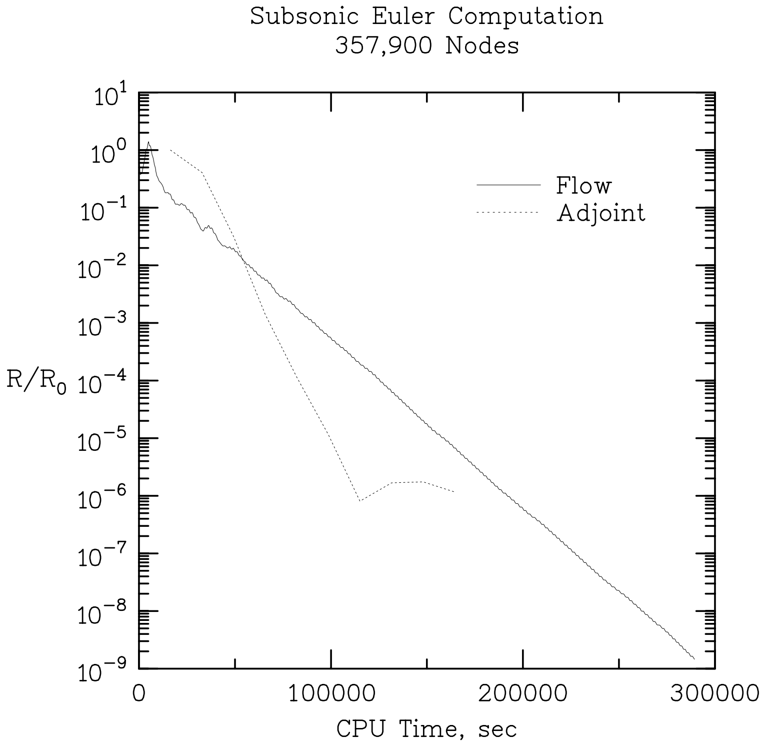

Figure 7.3. Convergence histories for subsonic inviscid flow on fine mesh.

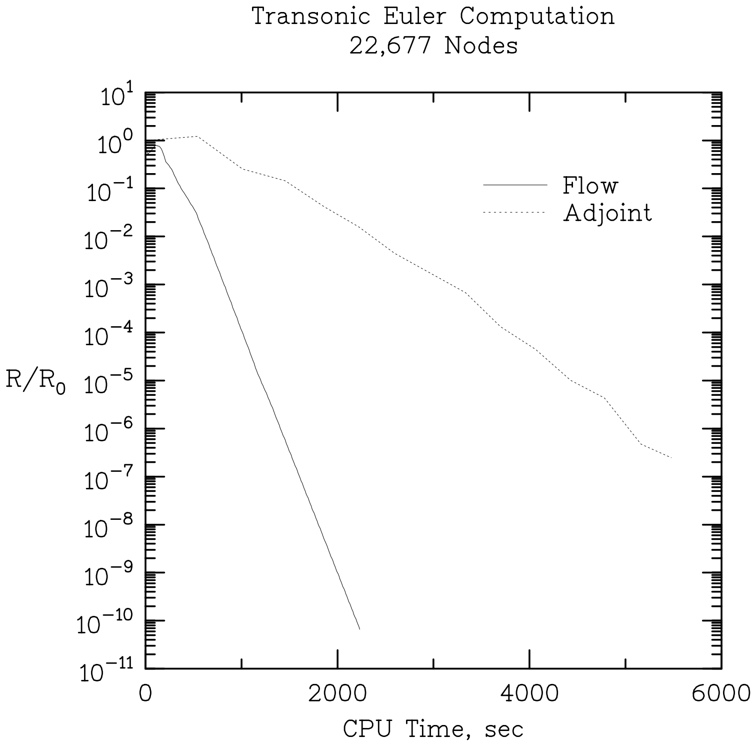

Figure 7.4. Convergence histories for transonic inviscid flow on coarse mesh.

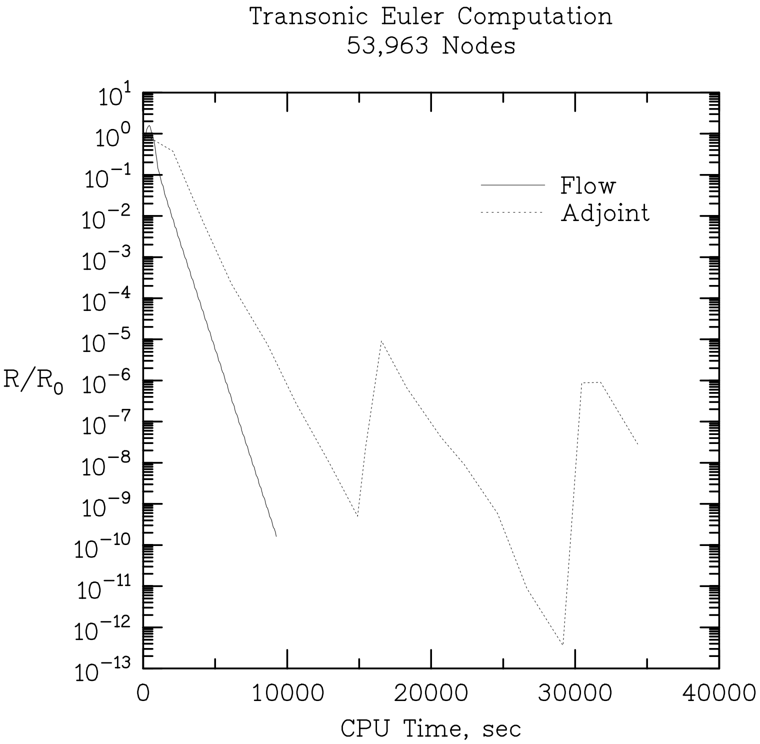

Figure 7.5. Convergence histories for transonic inviscid flow on medium mesh.

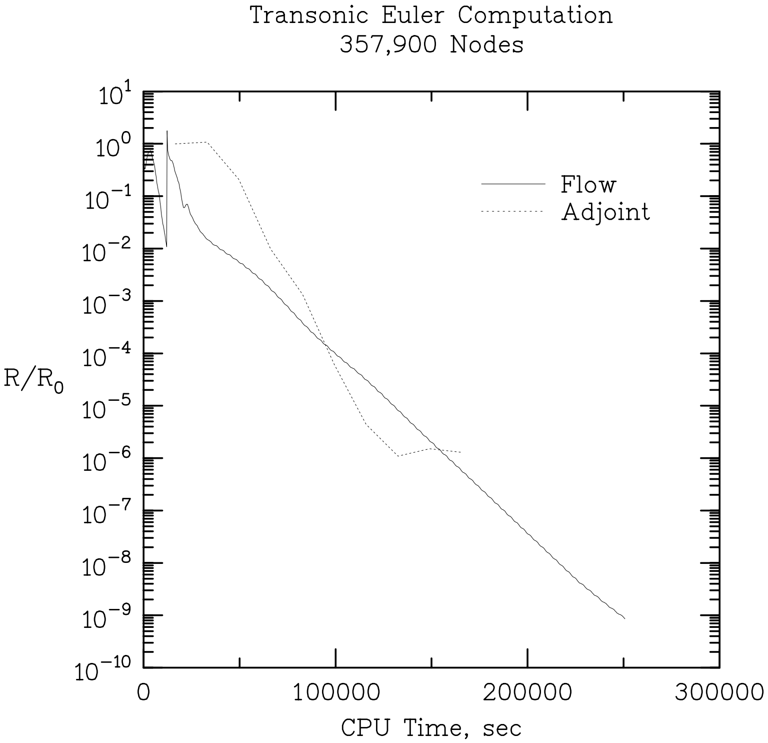

Figure 7.6. Convergence histories for transonic inviscid flow on fine mesh.

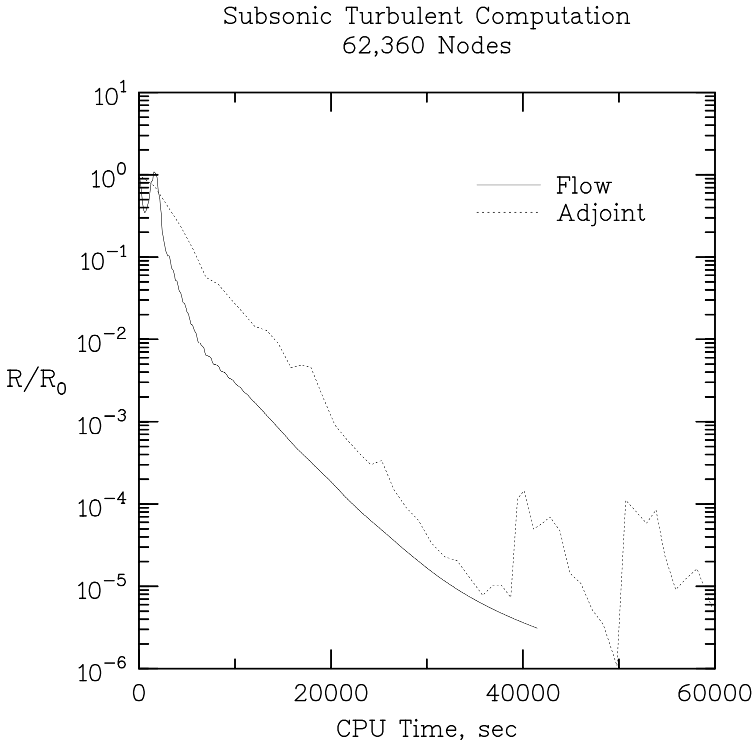

Figure 7.7. Convergence histories for subsonic turbulent flow on coarse mesh.

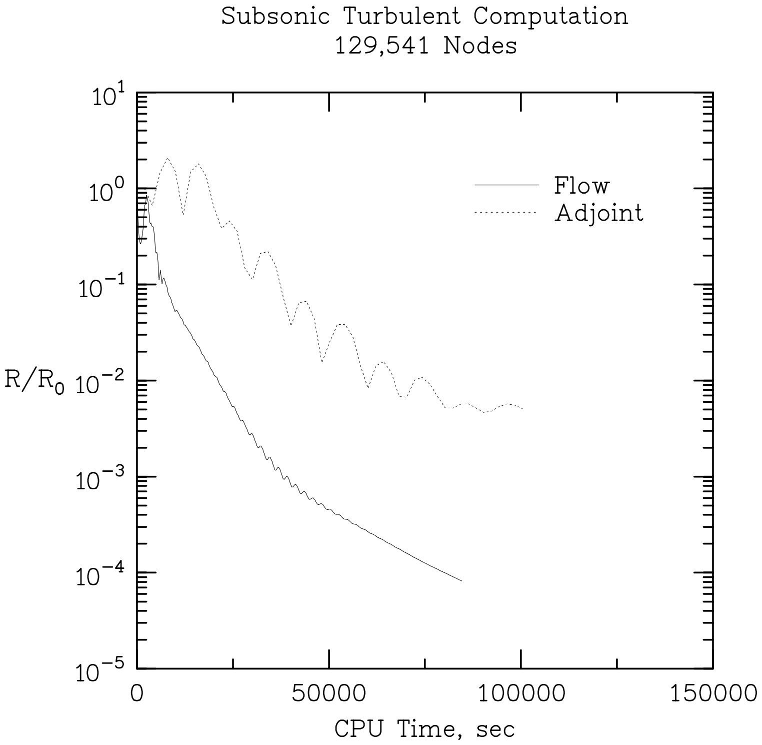

Figure 7.8. Convergence histories for subsonic turbulent flow on medium mesh.

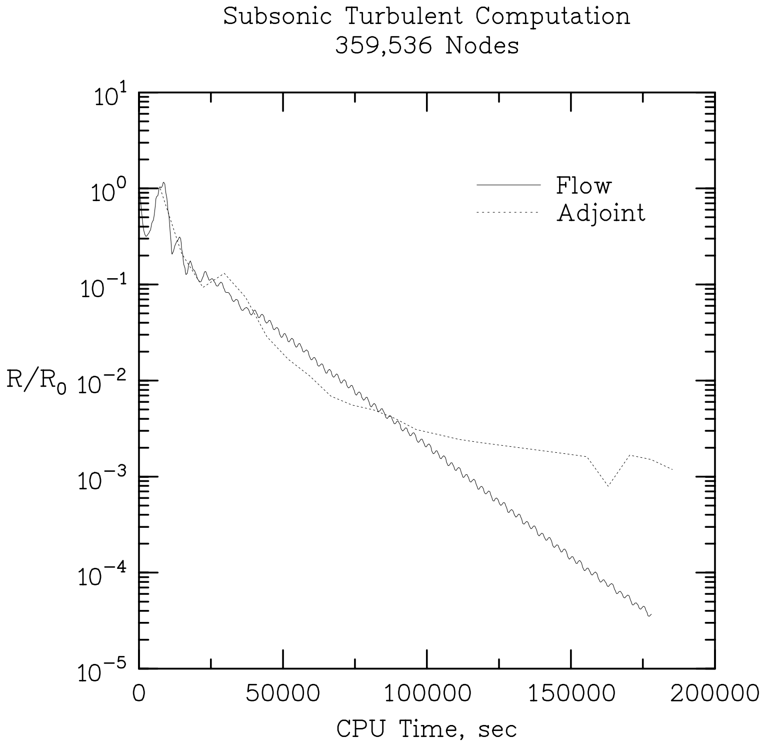

Figure 7.9. Convergence histories for subsonic turbulent flow on fine mesh.

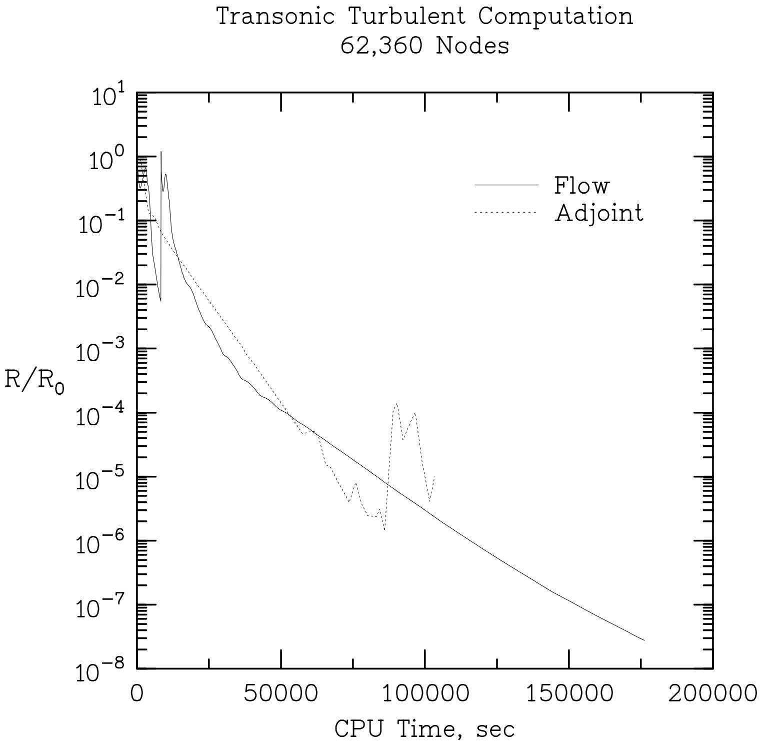

Figure 7.10. Convergence histories for transonic turbulent flow on coarse mesh.

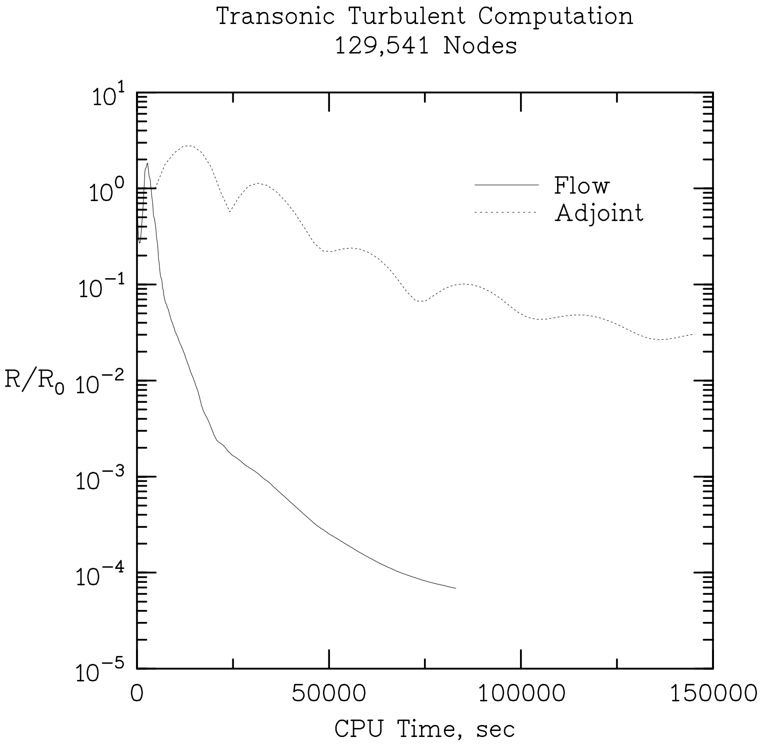

Figure 7.11. Convergence histories for transonic turbulent flow on medium mesh.

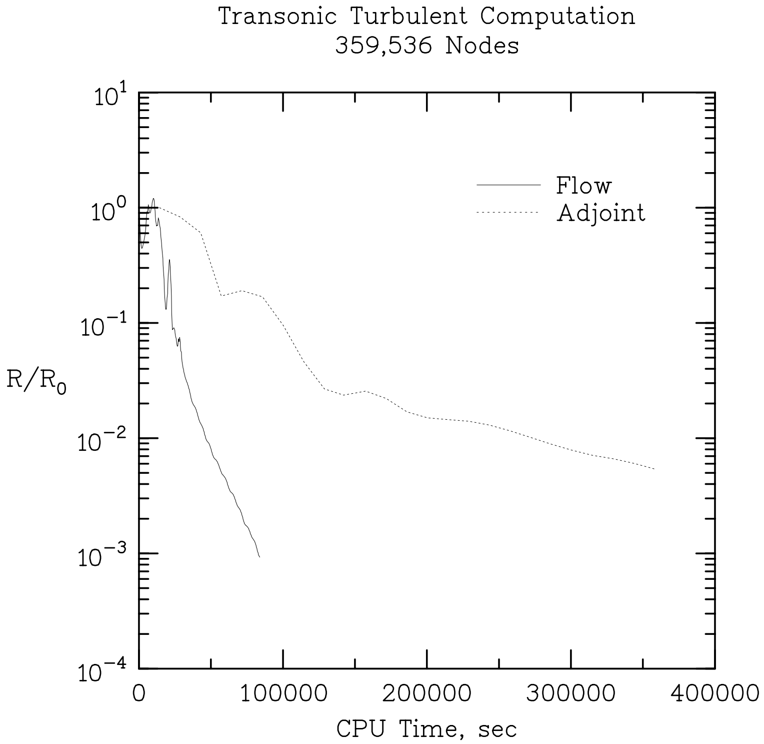

Figure 7.12. Convergence histories for transonic turbulent flow on fine mesh.

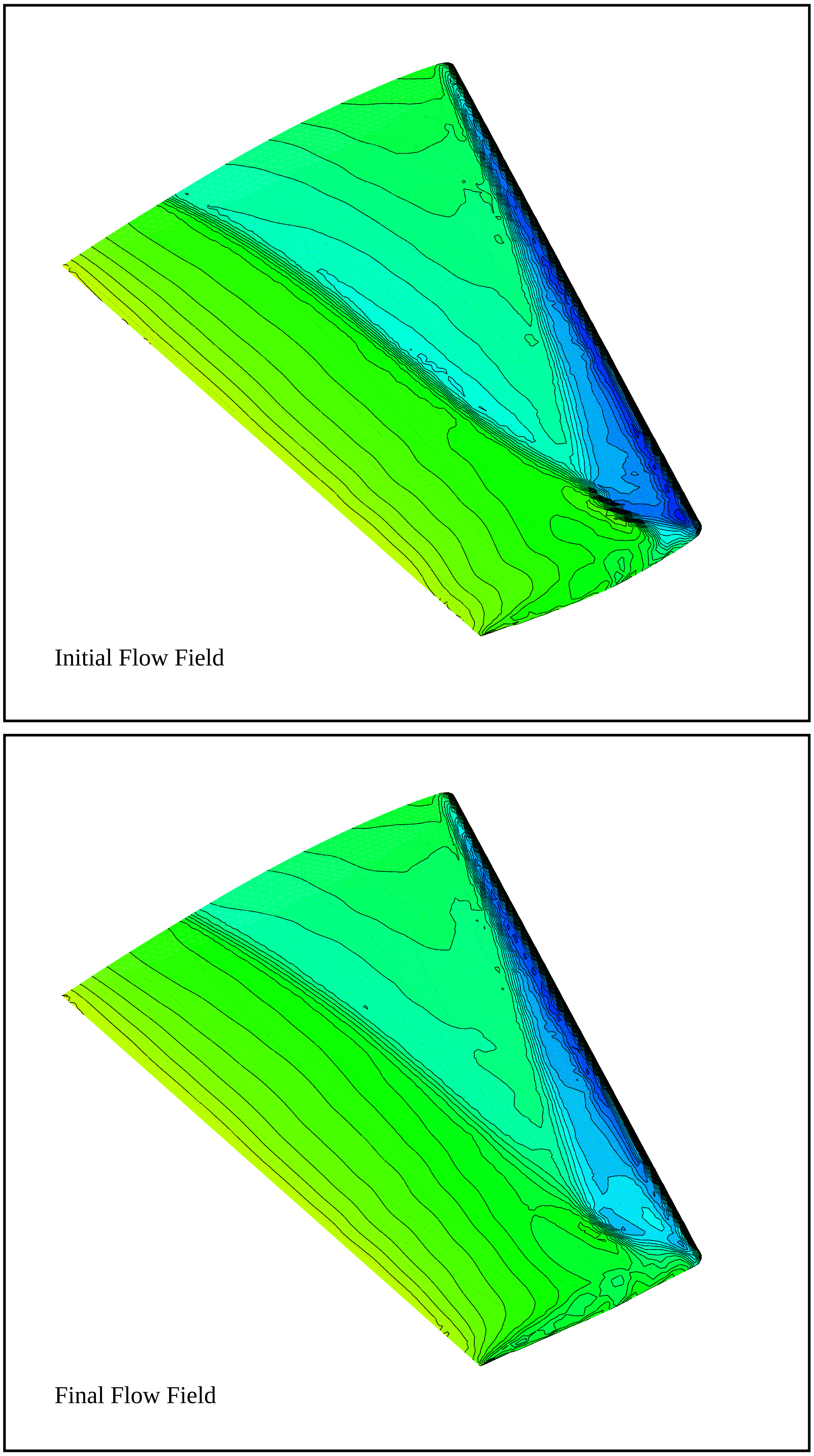

Figure 8.1. Initial and final density contours for inviscid wing design.

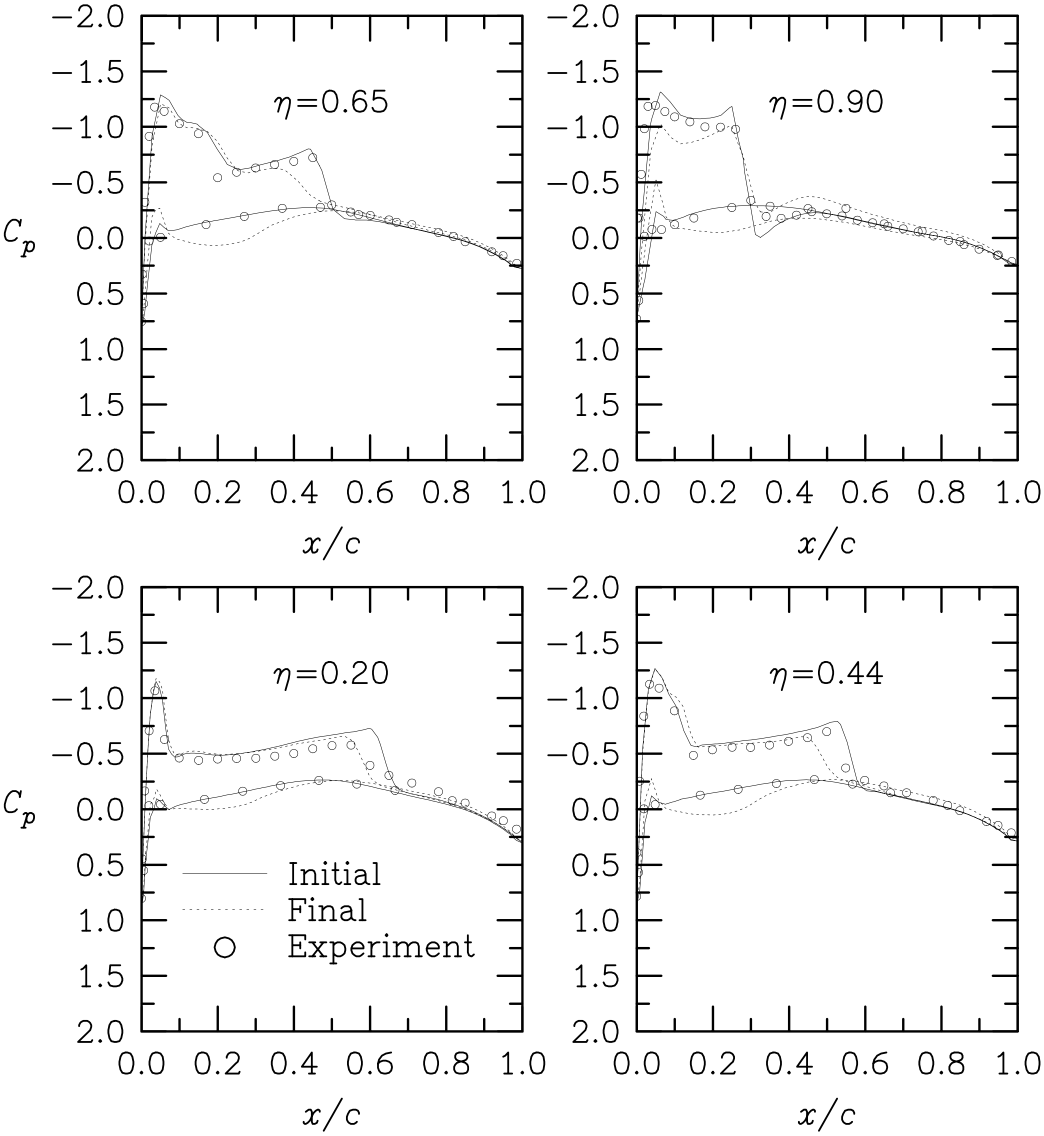

Figure 8.2. Initial and final pressure distributions for inviscid wing design.

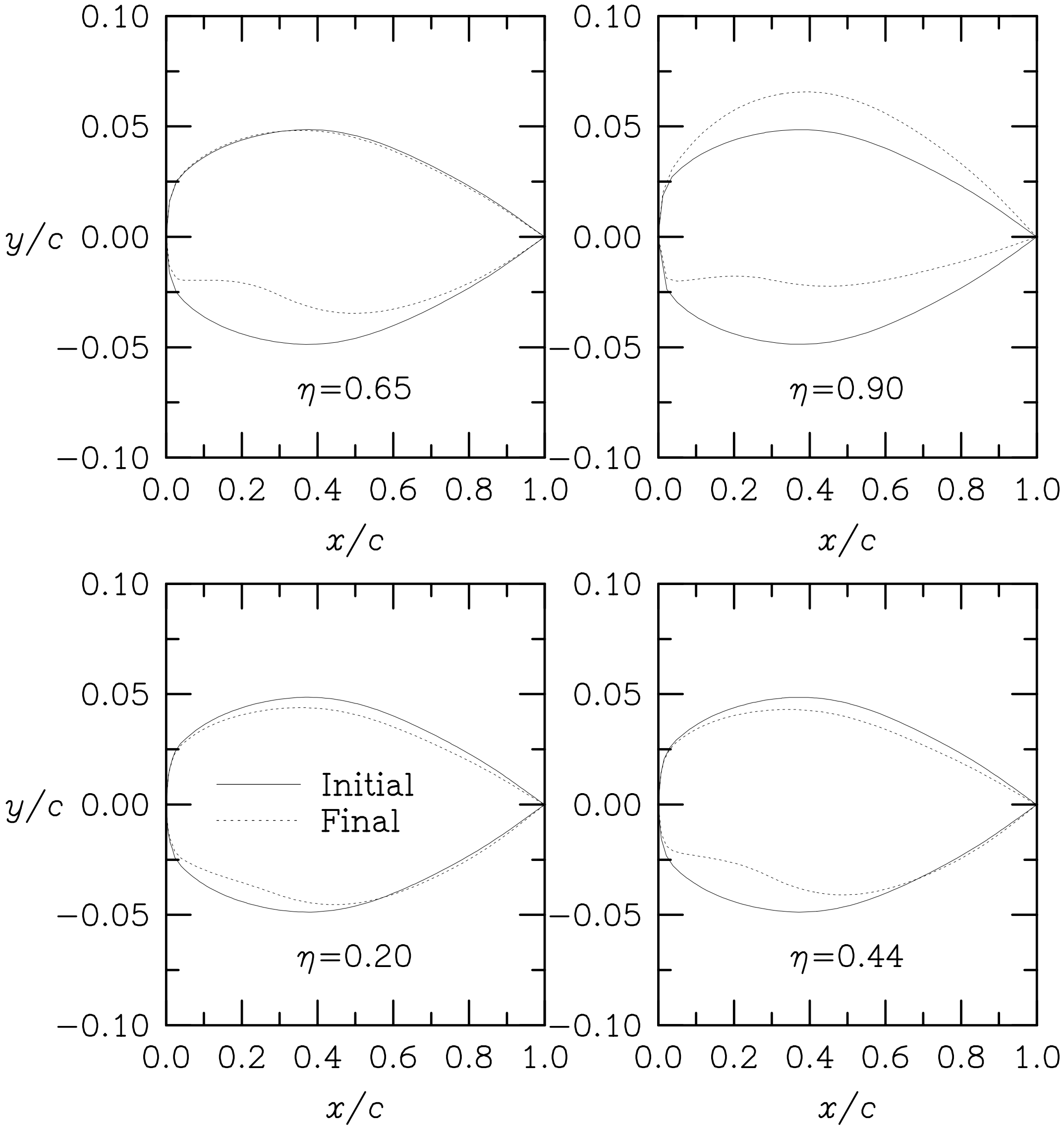

Figure 8.3. Initial and final geometries for inviscid wing design.

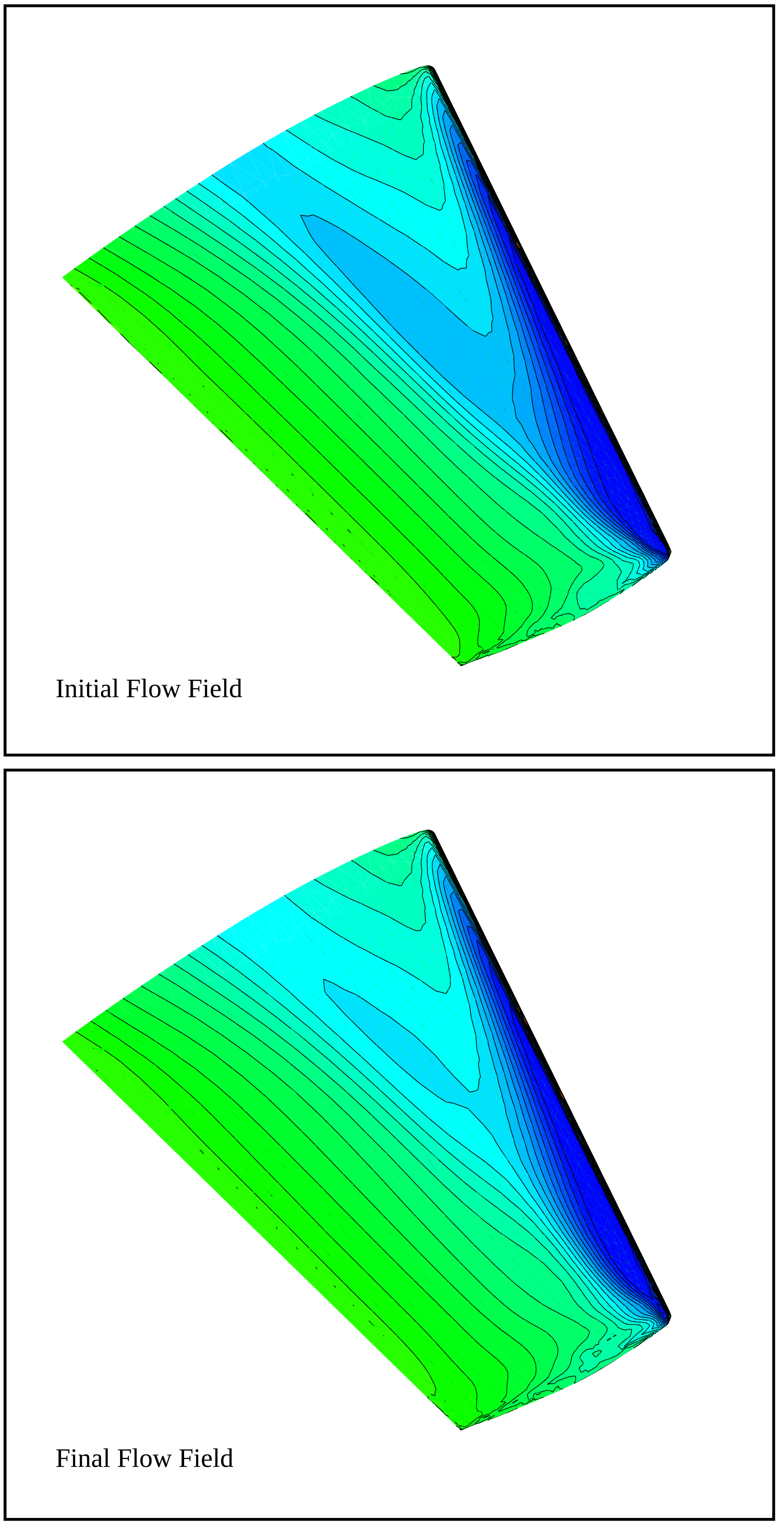

Figure 8.4. Initial and final density contours for turbulent wing design.

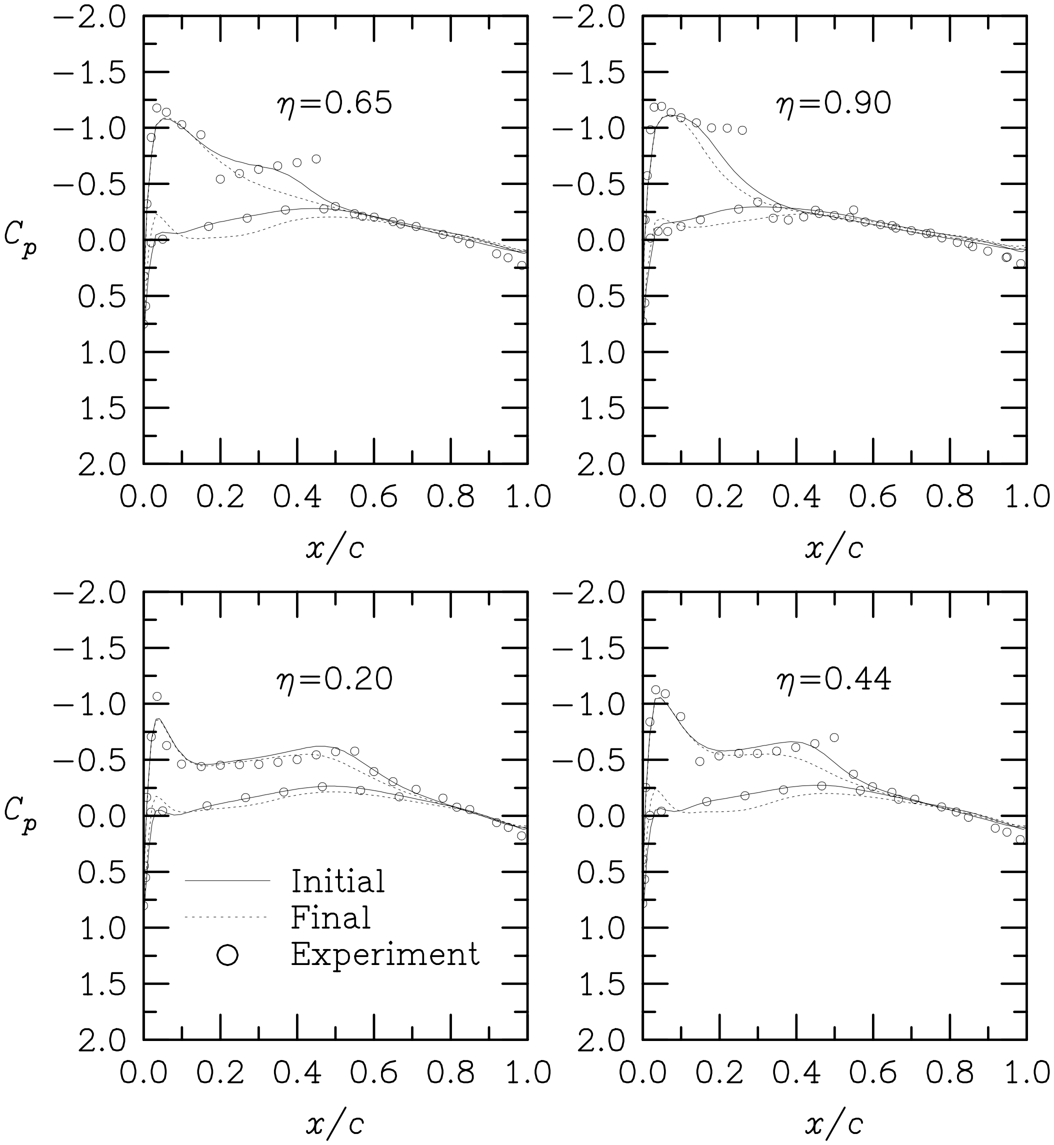

Figure 8.5. Initial and final pressure distributions for turbulent wing design.

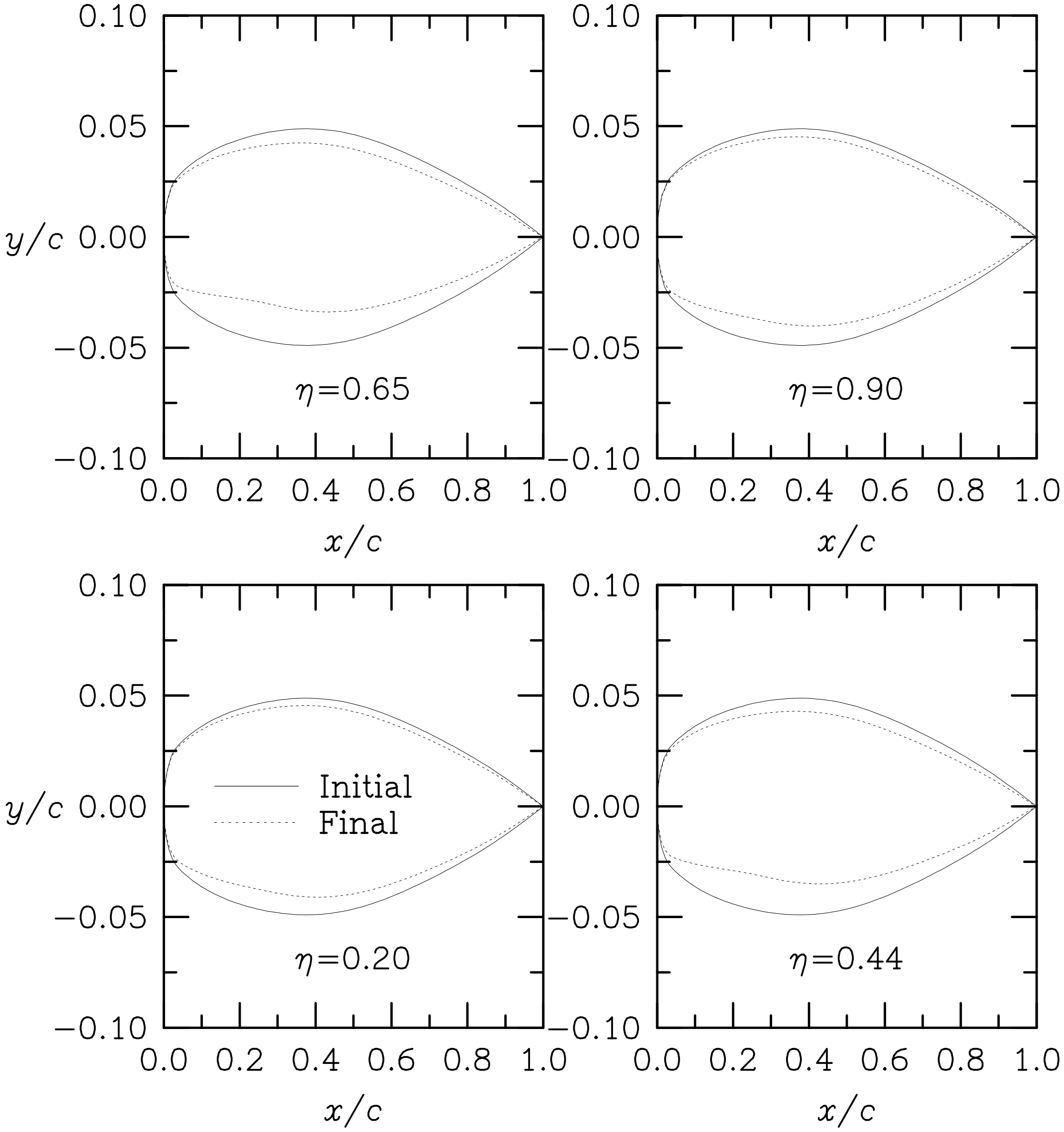

Figure 8.6. Initial and final geometries for turbulent wing design.

Vita

Eric John Nielsen was born on October 16th, 1972 at Scott AFB in Illinois. He grew up in a number of different places as the military relocated his family frequently. Eric finished high school in June 1990 at the age of seventeen in Tabb, Virginia. In the fall of that year, he enrolled as a freshman in the engineering program at Virginia Tech in Blacksburg, Virginia. Eric spent three summers working on Los Angeles-class submarines while employed by Newport News Shipbuilding as an undergraduate, and graduated Cum Laude with a dual degree in Aerospace and Ocean Engineering and a minor in Mathematics from Virginia Tech in May 1994. He continued with his education at Virginia Tech that fall, entering the graduate program in Aerospace Engineering. As a young graduate student, Eric spent two summers with the Aerodynamic and Acoustic Methods branch at NASA Langley Research Center through the Langley Research Summer Scholars program, and the branch continued to support his graduate work through a research grant. Eric Nielsen completed his doctorate in Aerospace Engineering in December 1998, and plans to continue his research at Langley through a post-doctoral position with the National Research Council. In his free time, he enjoys travelling to Metallica events and overseeing a large chapter of the bands' fan club.