Spring 1997

Aerodynamic Gradient-Based Optimization Using Computational Fluid Dynamics and Discrete Sensitivities for Practical Problems

Mohagna Jayendrarai Pandya
Old Dominion University

AERODYNAMIC GRADIENT-BASED OPTIMIZATION USING COMPUTATIONAL FLUID DYNAMICS AND DISCRETE SENSITIVITIES FOR PRACTICAL PROBLEMS

by

Mohagna Jayendrarai Pandya
B.E., January 1986, Saurashtra University, Rajkot, India
M.Tech., January 1988, Indian Institute of Technology, Kanpur, India

A Dissertation submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

AEROSPACE ENGINEERING

OLD DOMINION UNIVERSITY
May 1997

Approved by:

Oktay Baysal (Director)

Colin P. Britcher (Member)

Osama A. Kandil (Member)

David E. Keyes (Member)

ABSTRACT

AERODYNAMIC GRADIENT-BASED OPTIMIZATION USING COMPUTATIONAL FLUID DYNAMICS AND DISCRETE SENSITIVITIES FOR PRACTICAL PROBLEMS

Mohagna Jayendrarai Pandya
Old Dominion University, 1997

Director: Dr. Oktay Baysal

A gradient-based shape optimization methodology based on quasi-analytical sensitivities has been developed for practical three-dimensional aerodynamic applications. The flow analysis has been rendered by a fully implicit, finite-volume formulation of the Euler and Thin Layer Navier-Stokes (TLNS) equations. The flow equations and aerodynamic sensitivity equation have been solved using an alternating-direction-implicit (ADI) algorithm for memory efficiency. A wing geometry model based on space-surface and planform parameterization has been utilized. The present methodology and its components have been tested via several comparisons.

Initially, the inviscid flow analysis for a wing has been compared with those obtained using an unfactored, Preconditioned Conjugate Gradient (PCG) approach, and an independent Computational Fluid Dynamics (CFD) code which has been extensively validated. Then, the viscous laminar flow analysis for a wing has been compared with that obtained using again the extensively validated CFD code. Next, the sensitivities computed with the present method have been compared with those obtained using the finite-difference and the PCG approaches. Effects of convergence tolerance on the flowfield sensitivities have been shown. Also, effects of grid size and

viscosity on the flow analysis, sensitivity analysis and the shape optimization have been established.

Despite the expected increase in the computational time, the results indicate that shape optimization problems, which require large numbers of grid points, can be resolved with a gradient-based approach.

The new procedure has been demonstrated in the design of a cranked arrow wing at Mach 2.4, with coarse and fine grid based computations performed with Euler and TLNS equations. The influence of the initial constraints on the geometry and aerodynamics of the optimized shape has been explored. Various final shapes generated for an identical initial problem formulation but with different optimization path options (coarse or fine grid, Euler or TLNS) have been aerodynamically evaluated via a common fine grid TLNS based analysis. The efficacy of these design options has been evaluated by comparing net performance improvement in tandem with the CPU time requirements.

Results show that fluid dynamic and sensitivity analyses using ADI compare well with the PCG method and CFL3D code. The ADI method reduces the memory storage but increases the computing time as compared to the PCG method. It is demonstrated that the inherent larger size of optimization problems can be accommodated by using the ADI method. The presently developed optimization procedure is capable of learning aerodynamic lessons during the evolution of optimized shapes. Initial constraints conditions show significant bearing on the optimization results. Results demonstrate that to produce an aerodynamically efficient design, it is imperative to include viscous physics in the optimization procedure with proper resolution. However, if CPU time constraints do not permit this option, it is advantageous to incorporate inadequately resolved viscous flow physics in lieu of properly resolved inviscid flow physics.

Based upon the present results, it is recommended that to better utilize computational resources, a number of viscous coarse grid cases using the PCG (preferably) or ADI method, should initially be explored to improve optimization problem definition, design space and initial shape. Optimized shapes should be analyzed using high fidelity (viscous fine grid resolution) flow analysis to evaluate their true performance potential. Subsequently, a viscous fine grid-based shape optimization should be conducted, using ADI method, to accurately obtain the final optimized shape.

DEDICATION

To my parents
Sarla J. Pandya and Jayendrarai B. Pandya

ACKNOWLEDGMENTS

I wish to express deep appreciation and gratitude to my advisor Dr. Oktay Baysal for providing his invaluable guidance, encouragement and support during the entire course of this study. I wish to very sincerely thank my guidance and dissertation committee members Dr. Colin P. Britcher, Dr. Osama A. Kandil and Dr. David E. Keyes for their valuable time and suggestions. I also wish to extend sincere thanks to my Graduate Program Director, Dr. Norman F. Knight for his valuable time in reviewing this dissertation.

I would like to recognize the previous works of Dr. Greg W. Burgreen and Dr. Mohamed E. Eleshaky from which I learned a lot. Special thanks to my friends Dr. K. P. Singh, Cem C. Item, Ratikant Mohanty and Dr. Guan-Wei Yen for their moral support and help during various stages of this study.

Finally, I wish to thank all of my family members, especially my brother Dr. Jagdip J. Pandya, relatives and close friends, whose prayers, inspiration, and love served as catalyst for me in setting high standards and striving to pursue them. I also thank my wife Archita for her understanding and patient support throughout this study.

This research was supported by the NASA Langley Research Center under Grant No. NCC-1-211. The technical monitor was Dr. James L. Thomas. All computations for the present study were performed on the CRAY 2 and CRAY-YMP supercomputers at NASA Langley Research Center.

Mohagna J. Pandya

March 1997

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

a	speed of sound
A	area of airfoil section
ADI	alternating directions implicit
aoa	angle of attack
C_D	coefficient of drag
C_L	coefficient of lift
C_L/C_D	lift-to-drag ratio
c, chdscal	chord and its distribution
C_p	coefficient of pressure
CPU	central processor unit
D	vector of design variables
e_t	total energy per unit volume
f	shape functions
F	objective function
$\hat{F}, \hat{G}, \hat{H}$	inviscid flux vectors in curvilinear coordinates
G	constraints
$\hat{H}_v$	viscous flux vector in curvilinear coordinate
I	identity matrix
J	Jacobian matrix
M	Mach number, vector of metric terms
MW	millions of 64-bit words
n	normal direction
$NCON$	total number of constraints
$NCON_f$	number of flow related constraints
NDV	number of design variables
p	static pressure
P	Bezier control points
P_r	Prandtl number
Q	vector of conserved flow variables
Q'	sensitivity of Q to design variable
R	vector of residuals
R_e	Reynolds number
$R^{\pm}$	Riemann invariants
s	arc length

$\vec{S}$	search direction vector
`spn`	half-span length
t	time
T	static temperature
t, `thkscal`	thickness and its distribution
TLNS	thin layer Navier-Stokes
`tranx, tranz`	coordinates of the wing leading edge
`twst`	geometric twist
u,v,w	velocity components in Cartesian coordinates
u,v	nondimensional arc lengths in Bezier-Bernstein framework
U,V,W	contravariant velocity components
V	wing volume
X	vector of volume-grid coordinates
x,y,z	Cartesian coordinates

Greek Symbols

α	angle of attack
γ	ratio of specific heats
δ	central difference operator, scaling factor
Δ	forward difference operator
ϵ	very small number to prevent division by zero
θ	included angle at the trailing edge
ϑ	tolerance, order of accuracy
κ, φ	parameters which determine nature and order of accuracy of spatial differencing
λ	second coefficient of viscosity, adjoint-variable vector
Λ	sweep angle
μ	coefficient of dynamic viscosity
ξ,η,ζ	generalized curvilinear coordinate directions
ρ	density
τ	time coordinate in computational domain
ω	relaxation parameter
$\sum$	summation

Other Symbols

∂	partial derivative
∇	gradient, backward difference operator
$\in$	within a set of values

Overlines

$\sim$	approximate value

Subscripts

b	body surface
crank	mid half-span location
L	nondimensional characteristic length
LE	wing leading edge
max	maximum value
min	minimum value
N	conditions in the normal direction for a swept leading edge wing
TE	wing trailing edge
∞	freestream value
0	value at zero lift

Superscripts

n	time level
*	dimensional quantity
*,**	intermediate solution during a time step advancement in ADI
+,−	conditions at the right and left of the cell interface, positive and negative fluxes
-1	inverse

Chapter 1

INTRODUCTION

Aviation and aerodynamics have developed rapidly since their humble success in 1903. Two world wars and numerous regional wars have spurred the advances in the airplane for military and civilian purposes.

For several decades now, swept wings and jet propulsion have been the workhorse of military and civilian sectors of aviation [1]. The trend in commercial aviation to fly higher and faster has resulted in the development of the 707 class of Conventional Take-Off and Landing (CTOL) transport and its subsequent derivatives. As a logical extension of this trend, endeavors are being made to make supersonic transport viable and routine. After an abrupt halt in the mission to realize such a goal in the 1970's, due to economic and environmental concerns, today there is a resurgence of interest in the development of such an aircraft, which is designated as the Supersonic Transport (SST) or High Speed Civil Transport (HSCT).

Development of such an aircraft poses many technological challenges, as it needs to address concerns related to ozone depletion, sonic booms, sideline noise, and wake-vortex hazards while being economically viable. It is recognized that innovative concepts for drag reduction have to be incorporated through aerodynamic design, so that the consequently lower aircraft weight can provide benefits in terms of reduced sonic boom, sideline noise pollution, and initial and direct operating costs [2].

[1]The journal model for this dissertation is the AIAA *Journal of Aircraft*.

The impact of improving lift-to-drag ratio, L/D, on take-off gross weight is shown by Masuda and Yoshida [3] by performing calculations for a 300-seat airplane with a 5500-NM range and a cruise speed of Mach 2.5. It is shown in this study that the take-off gross weight is reduced by 130,000 lbs, with cascading environmental and economical benefits, if the lift-to-drag ratio is improved from 8 to 9. It should be mentioned here that a first generation SST, the Concorde, operates at a cruise lift-to-drag ratio of 7.

It is against the backdrop of such exigencies that "optimization," which is concerned with achieving the best outcome of a given operation while satisfying certain restrictions, assumes paramount importance for modern aircraft design. Optimization is implicit in the thinking of most aircraft designers who themselves assume the role of "optimizer" by resorting to parametric analysis or trade-off studies. In this method, a range of values of each of a number of parameters, as guided by judgment of the designer, is analyzed while temporarily holding the remaining parameters fixed. This approach to optimization is unwieldy and inefficient for higher-dimensional design space, provides less certainty of obtaining an optimum, and requires a great deal of expertise to produce optimum configurations [4]. There are other automated design methods such as indirect methods, inverse design techniques, and direct design techniques. These automated design techniques can reduce overall engineering effort and calendar time for developing aircraft components and configurations that have favorable aerodynamic performance or aerodynamic interference characteristics [5].

The calculus of variations is typical of indirect methods. In this approach, results can occasionally be obtained in closed-form expressions, which represents solutions to a parameterized class of problems. However, application of the calculus of variations to the optimization of complex systems, such as an aircraft configuration can be very difficult. In the case of inverse design, the geometry of the object, compatible with the prescribed features of the flowfield, is solved for. Direct methods

generally consist of procedures involving design by numerical optimization. For the aerodynamic design of an aircraft, a flow analysis program is coupled with a numerical optimization algorithm to create a design tool [6]. Aerodynamic quantities such as lift, drag, and pitching moment are computed by the CFD algorithm for a certain configuration and are used in defining an objective function to be minimized by the optimizer [7]. Synergistic advances in CFD algorithms and computer hardware have made this technique very attractive for aircraft configuration optimization.

Currently, this technique is being vigorously pursued for the development of Multidisciplinary Design Optimization (MDO) as advocated by Sobieski [8]. In essence, MDO procedures facilitate effective communication between various subsystems and disciplines to arrive at a mutually optimal solution. Aircraft wing design involving several disciplines, such as aerodynamics, structures, controls and materials, is an example of MDO. This procedure is a reflection of the modern trend of "concurrent engineering" in industry, which implies incorporating as many relevant disciplines as practicable in an early stage of the design process in order to reduce costly feedback.

For a design problem of the HSCT where design requirements of each discipline are conflicting, the need to follow this procedure cannot be overemphasized [9]. Sobieski's [8] formulation for MDO is based on derivatives of each individual discipline's output functions with respect to its input functions. These are known as sensitivity derivatives and the reliability of MDO procedures critically hinges on the reliability of sensitivity derivatives from individual disciplines. The aerodynamic discipline among all, is perhaps one of the more challenging as it is fraught with several ramifications such as attendant nonlinearity, large scale, and asymmetry.

As is evident from the discussion, efficient wing design is a fundamental aspect of any transport aircraft design and has a crucial impact on the viability of

the supersonic transport aircraft. Therefore, wing aerodynamics is the focal point of the present study.

1.1 Literature Review

In this section, literature on sensitivity analysis is reviewed and underlying concepts are highlighted. Subsequently, popular aerodynamic design optimization methods are surveyed identifying their origin, concepts and applications.

1.1.1 Sensitivity Analysis

Following Refs. [10, 11], the sensitivity of a system can be defined as the derivatives of the functions (objective function, constraints), that characterize the system with respect to variables (design variables) that govern the system. It helps the engineer to identify the parameters that influence the system most strongly. Sensitivity analysis methodology for structural mechanics has been in existence for a long time. However, emergence of sensitivity analysis as a fruitful tool in the aerodynamics arena is relatively new and enjoys ascendency as a research interest following Sobieski's [12] appeal to the CFD community for the extension of its capability to include sensitivity analysis to facilitate MDO. Flow sensitivities can also be used to determine perturbed flows efficiently, using Taylor series approximations [11]. Lastly, flow sensitivities are indispensable components of any gradient-based aerodynamic design optimization technique.

Sensitivities are computed from the CFD codes by either of the two approaches, namely, finite differencing and analytical approaches. Use of finite differencing is traditional due to its conceptual simplicity. In this approach, sensitivity derivatives are obtained by differencing two flow solutions of incrementally different design variables. The most appealing feature of this approach is that it obviates the need for analysis code modification. However, this approach is computationally expensive

as additional flow analyses need to be performed for sensitivity derivatives with respect to each design variable. Besides this, the quantum of design variable perturbation has a direct bearing on the accuracy of sensitivities vis-a-vis truncation error and condition error [10]. Thus, there is an element of uncertainty in determining the appropriate sizes of perturbations. In the past, efforts have been made to improve finite-difference sensitivity calculation [13]. Recently for example, two methods have been presented [14] to improve the accuracy and reduce the computational cost in evaluating finite-difference sensitivities for aerodynamic shape optimization. The first method is a modified finite difference approach that improves the accuracy of sensitivities at a reduced cost using an improved iteration strategy. The second method finds the optimum perturbation size based on an asymptotic error formula. Despite these improvements, the inherent cost disadvantage of the finite difference approach becomes more apparent as higher order methods based on Euler and Navier-Stokes equations begin to occupy the centerstage of aerodynamic analysis and the purview of sensitivity analysis is expanded from that involving sensitivity calculation for a simple object, such as an airfoil defined by a few design variables, to that for a three-dimensional object, such as a wing, a wing-body, or a whole aircraft, governed by several design variables. These deficiencies of the finite difference method has led to the quest for, and emergence of, methods such as analytical approaches for the determination of sensitivities.

Analytical approaches for the computation of sensitivities can be categorized as continuous and discrete [15]. The underlying difference between the two approaches is in the preferred sequence of differentiation and discretization. In the continuous approach, the governing equations of fluid flow are differentiated starting with their original form prior to discretization whereas in the discrete approach, a discretized form of the governing equations is used for the differentiation with respect to the

design variables. The continuous approach is also known as the variational approach [16], or adjoint approach [17], or control theory approach [18].

Application of the continuous approach was pioneered by Lions [19] and Pironneau [20], who used variational methods of optimal control to obtain optimality conditions for minimum drag problems of fluid mechanics using incompressible Navier-Stokes equations. Angrand [21] computed optimal shapes of airfoils and nozzles with potential flow equations using a continuous formulation. Yates [22] incorporated a variational sensitivity analysis method based on the linearized lifting surface theory and calculated the rates of change of lifting pressures with respect to general changes in aircraft geometry. Cabuk and Modi [23] and Huan and Modi [24] followed continuous approach for the design of minimum drag bodies in incompressible laminar flow. Ibrahim and Baysal [16] have used a sensitivity analysis approach based on variational method and perturbation technique. They derived the adjoint (co-state) partial differential equations and their boundary (transversality) conditions from the quasi one-dimensional Euler equations of the fluid flow.

A novel concept by Lions [19] was inspired by control theory, which was first applied to systems governed by partial differential equations and, more recently to aerodynamic sensitivities by Jameson [25]. The aerodynamic design problem is treated as a control problem in which the control function is the shape of the boundary, and consequently, the mapping function. The continuous sensitivity equations are obtained by first rewriting the potential/Euler equations in terms of the mapping function and then differentiating the resulting equations with respect to design variables [26, 18, 27].

The discrete approach for the computation of sensitivities is also known as the implicit gradient method, discrete sensitivity analysis, or quasi-analytical approach. This approach can further be divided into the direct formulation and discrete adjoint formulation. In either of these approaches, a linear sensitivity equation needs

to be solved. Since the late 1980's, there have been promising developments in the area of discrete sensitivity analysis. Elbanna and Carlson [28] developed this approach for transonic small perturbation formulation and applied it to airfoils. Subsequently, they extended the same approach to the three-dimensional, full potential equation to compute the sensitivity coefficients for wing aerodynamics in the transonic regime [29].

Baysal and Eleshaky [10] and Baysal et al. [11] presented the mathematical formulation for aerodynamic sensitivity coefficients based on compressible, two-dimensional Euler equations with first- and third-order accurate discretizations. In these works, three methods to determine sensitivity coefficients, namely the finite difference method, the direct method, and the adjoint variable method are discussed with regards to their relative accuracies and computational effort. Also, several methods to solve the system of linear algebraic equations that arises in the quasi-analytical approach are investigated with regard to their accuracies, computational time, and memory requirement. Eleshaky and Baysal [30] later extended this method to the thin-layer Navier-Stokes equations in order to account for the viscous effects and computed sensitivities for transonic airfoils. In a parallel development, advances in the discrete sensitivity analysis have also been reported by Taylor et al. [31, 32]. Hou et al. [33] derived sensitivity equations to consistently account for the linearization of the boundary conditions. Taylor et al. [34] have also extended the formulation for 2D thin-layer Navier-Stokes equations and demonstrated it for the test problems of double-throat nozzle and a NACA 4-digit airfoil. The inexactness of consistent discrete aerodynamic sensitivity derivatives arising due to omission of turbulence modeling terms from the consistent differentiation has been highlighted by Korivi et al. [35] using the test problem of transonic turbulent high Reynolds number flow over a NACA airfoil.

It is shown in Refs. [29, 10, 35] that the discrete approach is computationally more efficient than the finite difference approach and two approaches to discrete sensitivities, direct differentiation and adjoint formulation generate identical sensitivities.

As mentioned before, computation of discrete sensitivities involves solution of a very large linear system of equations. Previous efforts for the solution essentially revolved around direct matrix inversion methods which enjoyed the advantage of being computationally cheap for the multiple right-hand-side vectors. However, its applicability turned out to be tenuous as sensitivity derivatives were sought for realistic applications, such as a finely resolved 2D airfoil or 3D wing problems. It was realized that application of direct matrix inversion techniques sometimes demands prohibitively high computer memory requirements for such problems. These problems have motivated researchers to circumvent memory bottleneck issue by devising new means for large problems. Two alternatives pursued to address these problems are the application of iterative methods and domain decomposition techniques.

Burgreen and Baysal [36] examined the solution of CFD and sensitivity equations by using a preconditioned conjugate gradient (PCG)-like method, GMRES [37], and concluded that it yields significant reductions in CPU time and memory requirements over the direct inversion method. Another approach in the iterative solution of the linear systems of equation, named the incremental iterative strategy, was developed by Korivi et al. [38]. This approach was applied in Ref. [39] for the calculation of sensitivity derivatives using a three-dimensional supersonic Euler code. This approach permits the use of the identical approximate coefficient matrix operator and algorithm to solve the nonlinear flow and linear sensitivity equations while ensuring the accuracy of the sensitivity derivatives at convergence.

A domain decomposition scheme for three-dimensional aerodynamic sensitivity analysis was proposed by Eleshaky and Baysal [40, 41]. This scheme is based

on the divide-and-conquer principle. This scheme divides the computational domain into smaller and nonoverlapping subdomains, known as multiblock grids, which are solved separately. Then the final solution is constructed from the subdomain solutions. This approach reduces the high memory requirements associated with the single large domain approach and also facilitates easy structured grid generation for complex geometry problems. Sensitivity equations for the interior cells and the sensitivity equations for boundary cells that couple the subdomains are iteratively solved with a preconditioned conjugate gradient-like procedure.

Some complementary developments in the application of quasi-analytical approaches for sensitivities involve the use of mathematical symbolic manipulation or automatic differentiation techniques. The classical quasi-analytical approach entails hand differentiation of discretized residuals with respect to design variables and dependent unknowns. This approach, although very efficient, requires extra caution in handling terms such as boundary conditions, and turbulence models. This may make this approach cumbersome in some cases. In order to mitigate this difficulty, a symbolic manipulator, MACSYMA [42] was used by Elbanna and Carlson [29] to carry out the symbolic evaluation of the elements of the sensitivity equation corresponding to three-dimensional full potential equations. Use of the automatic differentiation precompiler software tool ADIFOR [43] is demonstrated by Green et al. [44], Sherman et al. [45] in various manners, and discussed by Baysal and Cordero [46]. The output of this precompiler procedure is a new differentiated source code, which upon compilation and execution computes exact numerical values of the derivatives of any specified output functions with respect to any specified input variables [43].

A comprehensive survey of various approaches to aerodynamic sensitivity analysis is presented by Taylor et al. [47]. Some research efforts have been focused on assessing the differences between the continuous and discrete approaches for the sensitivities. Frank and Shubin [48] have shown that the implicit gradient (i.e., discrete

method) is equivalent to applying the "variational" or optimal control approach to design optimization directly to the discretized analysis problem rather than to the continuous problem as is usually done. Recently, Burkardt and Gunzburger [17] have examined the relationship between the solution of the sensitivity equation derived from the discrete state equation and that derived by discretizing the sensitivity equation of the continuous state equation. It is shown in this work that the latter results were significantly inaccurate for sensitivities with respect to geometric parameters and resulted in erroneous gradient computations and breakdown of the optimization.

Sensitivity computations outside the domain of steady aerodynamics can be found in the works of Lorence and Hall [49] for the unsteady aerodynamic loads in cascades, and Barthelemy and Bergen [50], Arslan and Carlson [51], and Sorensen and Drela [52] among others for static aeroelastic applications.

1.1.2 Aerodynamic Optimization

Popular aerodynamic design methods can broadly be classified as inverse methods and direct methods. In the inverse design approach, the geometry of the aerodynamic object is solved to yield the desired flow characteristic, typically surface pressure distribution, specified by the designer.

The foundation of the inverse method was laid by Lighthill [53] for the incompressible flow over an airfoil using conformal mapping technique. Lighthill's work also illustrates the existence of constraints on the pressure distribution for the well-posedness of the design problem. Two related design methods, having restricted applications, are the hodograph design method of Bauer et al. [54] and the fictitious gas method of Sobieczky [55]. Both methods are restricted to shock-free flows. In addition, the hodograph method is applicable for only two-dimensional flows and the fictitious gas method does not provide for the specification of surface values of the parameters. One approach to the inverse design in the transonic regime involves

Dirichlet boundary condition specification and surface transpiration for the computation of the geometry as formulated by Tranen [56]. The main idea [25] is to integrate surface speed corresponding to the desired pressure distribution to obtain the surface potential. The potential flow equation is solved with the Dirichlet boundary condition, and a shape correction is determined from the calculated normal velocity through the surface. Volpe and Melnik [57] have shown the existence of constraints on the prescribed pressure distribution in the compressible regime also and proposed methods to ensure existence of a solution to the inverse problem. Extension of this method for the three-dimensional case of transonic wing design is made by Henne [58].

Another inverse method, due to Garabedian and McFadden [59], is an iterative method based on residual-correction algorithm for transonic wings. A full potential code and the pseudo-time dependent equation of the transformed wings are coupled and the surface is treated as a free boundary. Movement of the surface is governed by the auxiliary time dependent equation based upon the difference of desired and predicted velocities on the airfoil surface. This algorithm afforded flexibility to incorporate higher order analysis codes into the design procedure although it yielded designs with slower convergence and surface shape distortions. Some of these difficulties were overcome in the modified Garabedian-McFadden scheme devised by Malone et al. [5], where this algorithm is used to design airfoils using a Navier-Stokes-equation-based analysis. This approach was also followed by Birckelbaw [60] for the Navier-Stokes-equation-based airfoil design. Giles and Drela [61] have formulated an approach for inverse design using two-dimensional Euler equations and a streamline coordinate system. The design method is based on the simultaneous solution of multiple streamtubes, coupled through the position of, and pressure at, the streamline interfaces and allows for the specification of airfoil surface pressure distribution. Some inverse methods solving the Dirichlet problem required special treatment of the trailing

edge closure and methods used to resolve this produced pressure distributions far off the desired ones [62].

Inverse design methods suffer from certain drawbacks such as the requirement of having *a priori* knowledge of a desirable pressure distribution, difficulties in imposing variety of constraints, and poor off-design performance of surfaces. However, inverse design methods are generally considered computationally more efficient as compared to direct methods, whenever they are applicable.

Recently, some hybrid methods have been formulated which aim at alleviating some difficulties associated with the pure inverse design approach while retaining its computational efficiency. One such design method formulated by Campbell and Smith [63] is of the predictor/corrector type, in which differences between an input target pressure distribution and the pressure distribution calculated by the aerodynamic code are related to changes in airfoil surface curvature. This method allows specification of geometry constraints and flexibility in incorporating various codes. It has been used for Navier-Stokes-equation-based designs and multipoint design of wings and airfoils [64, 65]. This method has been extended for the incorporation of flow constraints into design by automatically modifying the target pressure distributions and demonstrated for inverse designs of airfoils, wings [66] for multiple flow and geometry constraints, isolated and installed nacelle designs [67], and natural laminar flow airfoils [68]. Another inverse method which employs control theory to make successive modifications to a given airfoil to achieve a desired pressure distribution is shown by Lewis et al. [69] for airfoil design using potential and Euler equations. Summary of some basic approaches in inverse methods can be found in Ref. [25]. A detailed review of status and trends in design methods with emphasis on inverse design methods is presented by Dulikravich [70].

The direct design approach is also known as the design by numerical optimization, in which an aerodynamic analysis technique is coupled with a numerical

optimization procedure. The numerical optimization procedure controls the geometric variables that define the shape in such a manner that the objective function is extremized. Typical examples of objective functions (although not limited to these) are drag, lift, lift-to-drag ratio, pitching moment, and the difference between the target and actual pressure distributions of the design.

An aerodynamic analysis technique, of varying degree of fidelity to the flow physics, incumbent upon the availability of state-of-art of technology and computer resources at hand, is employed repeatedly for the evaluation of the objective function. This approach can be a very powerful tool for the designer, as it renders completely automated design with the capability to include several flow and geometry constraints and perform multipoint design. However, the computational cost of this approach can be higher than the inverse design approach.

Numerical optimization techniques can be classified into non-gradient-based and gradient-based design techniques. In the non-gradient-based design techniques, the optimization algorithm merely requires the objective function and constraint values during the optimization process. On the other hand, gradient-based design techniques incorporate algorithms which also require information on the derivatives of the objective functions and constraints. This method is called the hill-climbing strategy, as the design is moved in the direction of steepest ascent during the optimization process. Good discussions on various optimization algorithms can be found in Haftka and Gurdal [15, pages 115–207].

A good example of a non-gradient optimization of aerodynamic application is given by Gregg and Misegades [71] for transonic wing optimization based on the evolution theory. Some of the algorithms used for non-gradient methods of design optimization are probabilistic search methods such as simulated annealing and genetic algorithms. Genetic algorithms are based on the mechanics of natural selection and natural genetics, and search for optima through simulated evolution [72]. In the

same work [72]. Obayashi and Tsukahara have evaluated the performance of three optimization algorithms, namely, gradient based method, simulated annealing, and genetic algorithms using a simplified airfoil design problem, and argued that genetic algorithms make the best choice for the aerodynamic optimization. It is well known that gradient-based designs often produce only a locally optimal design, as most of the design problems of practical consequence do not satisfy the conditions of convexity and differentiability for a global optimum. On the other hand, genetic algorithms can yield globally optimum designs. However, the globally optimum design is produced only after a vast number of design evaluations. The use of accurate analysis methods for the performance evaluations can only produce such a design at an overwhelming computational cost.

Until now, gradient-based methods have been predominantly used for the aerodynamic design optimization. Hicks, Murman and Vanderplaats [73] presented the first application of this approach for the transonic airfoil design. Their procedure calculated gradients by finite differencing, used a transonic small disturbance code for the analysis, and involved the use of coefficients of polynomial expression used for defining an airfoil contour as design variables. Subsequent research has led to remarkable diversity in the application of the same approach with focus on the enhancement of accuracy, efficiency and applicability of the design procedure. The basic method of [73] was soon extended for the 3D wing design by Hicks and Henne [74] using a full potential code. Further applications of full potential codes for the airfoil design can be found in Refs. [75, 76], and for the 3D wing design in Refs. [77, 7, 78].

The use of shock expansion theory for the supersonic airfoil design optimization is made by Dutt and Srikanth [79]. Application of Euler equations for airfoil design is found in Refs. [80, 81]. The use of Euler equations in wing design is demonstrated by Chang et al. [82]. Euler codes are also used for the optimization of a wing in the presence of fuselage and nacelle as illustrated in Refs. [83, 84].

Some researchers have also performed airfoil optimization by including viscous effects with inviscid/viscous coupling, using a panel method, a full potential code or an Euler code for the inviscid flowfield and boundary-layer solution method for the viscous effects [85, 86, 87]. Finally, application of Navier-Stokes equations has been demonstrated by Hager et al. [88] for airfoil optimization and by Greenman et al. [89] for a transonic wing.

All gradient-based design techniques cited previously (Refs. [73]–[89]) used the finite-difference method for the gradient calculation. However, the accuracy of a gradient-based design optimization method, and its efficiency are directly related to the accurate and efficient receipt of the gradient information on the objective and constraints; that is, the sensitivity coefficents. To this end, the deficiencies of finite difference method as mentioned previously, impede their embrace by any practical gradient-based optimization methodology.

As explained in the previous section, sensitivity coefficients can also be delivered to the optimizer via sensitivity analysis. This approach has recently become a focal point of research into gradient-based optimization methods and has made promising strides as a very viable method for the realistic design optimization problems. To this end, both of the options for the sensitivity calculations, namely, discrete and continuous formulations, have been pursued. Concurrently, attention has also been devoted to the several methods for shape definitions to obtain smooth optimized shapes economically.

Baysal and Eleshaky [6, 11] demonstrated a discrete sensitivities-based method for the optimization of a scramjet-afterbody using the Euler equations. Eleshaky and Baysal [30] demonstrated this method for the transonic airfoil using the Thin-Layer Navier-Stokes equations. In these works, surface grid points were used as design variables and direct matrix inversion methods were used for solving sensitivity equations. Subsequently, a method for performing sensitivity analysis on decomposed

computational domains was developed to mitigate the computer memory problem associated with large two- and three-dimensional problems. Feasibility of this approach was demonstrated by Eleshaky and Baysal [90] for airfoil design, and its extension and suitability for a three-dimensional multicomponent design case was highlighted in Ref. [91] by performing nacelle optimization in the presence of a flat plate wing. Other examples of optimization based on decomposed domains are provided by Lacasse and Baysal [92] for single- and two-element airfoil designs and by Item and Baysal [93] for the single-element supersonic viscous airfoil design. Improvements in the design efficiency of single-block methods were first demonstrated by Burgreen et al. [94] and Burgreen and Baysal [36] by using Bezier-Bernstein polynomial for the surface parameterization and preconditioned conjugate gradient methods for the solution of flow equations and the sensitivity equation. These improvements were demonstrated using Euler equations for the transonic airfoil and internal-external nozzle configuration. Extension of this approach for three-dimensional optimization has been shown by Burgreen and Baysal [95, 96] for a transonic transport wing and supersonic delta wings in the presence of several flow and geometric constraints. Another example of gradient-based optimization studies incorporating discrete sensitivities can be found in Korivi et al. [97]. In this work, an incremental iterative method and space marching algorithm have been employed for the Euler equations to optimize a wing assumed to be in fully supersonic flow.

Pandya and Baysal [98] have examined the efficiency and accuracy of the factored and unfactored forms of flow and sensitivity equations and performed a large-scale, arrow-wing optimization study in inviscid flow using ADI based solution methodology. Recently, this methodology has been extended by Pandya and Baysal [99] for the viscous flow condition, and applied to the arrow-wing optimization in order to study the effect of viscosity, grid size and various problem formulation strategies on the optimized shape and corresponding aerodynamics. Unstructured grid and Euler

solver based methodology is demonstrated in Refs. [100, 101]. Elliott and Peraire [100] demonstrated this method by optimizing a transonic wing/body configuration and a business jet configuration consisting of wing, body, strut, nacelle, and horizontal and vertical fins.

Another variant of gradient-based optimization which uses a continuous or variational formulation for sensitivities also merits attention. In this approach, the sensitivity equation is of the same order and character as the flow equations, and its memory requirement is no more than that required by the flow analysis methods. One example of this approach is found in Ref. [21], where airfoil design optimization is performed for potential flows. In Refs. [16, 102], variational formulation and sensitivity-equation based optimization approaches are presented for the 1D nozzle and plane forebody simulator geometries, respectively. The control-theory-based optimization formulation [25] has been implemented for the airfoil design using numerically generated grids by Reuther and Jameson [26] for the potential flow and by Jameson and Reuther [18] for the Euler equations. Recently, Reuther and Jameson [27] have extended this methodology and conducted shape optimization studies on wing and wing/body configurations using Euler equations. Its latest extension is presented by Jameson, Martinelli, and Pierce [103] for the Navier-Stokes equations and by Reuther et al. [104] for their parallel computations. Anderson and Venkatakrishnan [105] have demonstrated the continuous adjoint formulation for the design optimization on unstructured grids.

All gradient-based optimization studies surveyed herein represent a two-cycle (inner-outer) iterative procedure in which the inner iterative cycle solves the analysis problem and the outer iterative cycle governs the inner cycle and determines the optimum iteratively [106]. This is also termed the "black box" method of numerical optimization. Its principal advantage is in the independence of the analysis code from the optimization code, which allows the latter to be treated as the black

box [48]. However, such loosely coupled optimization procedures may be computationally expensive as repetitive time consuming flow analyses need to be performed. To provide an efficient alternative to this class of methods, Rizk [106], has proposed a single-cycle optimization approach in which flow equations and design variables are simultaneously updated. Euler equations and transonic airfoil optimization problems have been used for the illustration of the procedure. Further examples of tightly integrated procedures may be found in Refs. [107, 108, 109]. In these methods, code independence is lost due to tight integration of the flow analysis and the optimization methodology.

A comparative study [48] of three optimization algorithms, namely, black-box methods with finite difference and implicit-theorem-based (i.e., quasi-analytical method) gradients calculations procedures, and the all-at-once method, which simultaneously altered flow and design variables, was performed using a model one-dimensional duct design problem. This study found that the finite-difference-based algorithm was robust but computationally inefficient, whereas the all-at-once procedure was computationally cheaper but less robust. Also, the implicit-gradient-based black-box procedure was found to provide a good compromise of desirable features.

1.1.3 Delta Wing Aerodynamics

Aerodynamic efficiency of an airplane is characterized by its lift-to-drag ratio. The drag of a supersonic configuration typically consists of skin friction drag, wave drag due to thickness, and drag due to lift. Supersonic drag due to lift includes both induced drag and wave drag due to lift. As an example, typical contributions to the overall drag coefficient of an HSCT-type design are shown by Küchemann [110] in Figure 1.1, where the abscissa, s/l, is the semispan-to-length ratio.

The theoretical drag polar for any symmetrical configuration, as given by Brown and McLean [111], can be expressed as:

$$C_D = C_{D0} + \left(\partial C_D / \partial C_L{}^2\right) C_L{}^2 \tag{1.1}$$

C_{D0} represents the zero-lift drag coefficient, which consists of wave drag and friction drag and $\left(\partial C_D / \partial C_L{}^2\right)$, is the lift-dependent drag rise factor, which includes the induced drag and the wave drag due to lift. The solution of the Eq. (1.1) for maximum (L/D) gives the following simple relation:

$$(L/D)_{\max} = (1/2)\sqrt{1/\left(C_{D0}\left(\partial C_D / \partial C_L{}^2\right)\right)} \tag{1.2}$$

In order to reduce the wave drag penalty for the supersonic flight, the swept wings concept, first introduced by Busemann [112], has become an integral part of high-speed aircraft design. Rogers and Hall [113] provide a very detailed understanding of the flow development over the plane swept-back wing at about 50 degrees, as the freestream Mach number is increased from a subsonic to a supersonic value. Sweep angles of 70 to 75 degrees are necessary for Mach numbers ranging from 2.0 to 3.0. Theoretical predictions indicate that an airplane with a wing of such high sweep would have an advantage of approximately 15 to 20 percent in the L/D ratio when compared to an airplane having a lower sweep angle, say 50 degrees.

However, experimental measurements on such wings have belied these theoretical predictions. Kulfan and Sigalla [114] note that this discrepancy is attributed to the influence of viscosity which causes leading edge and trailing edge flow separations. Ghorai [115] has also investigated reasons for the discrepancy in drag due to lift reductions between experiments and linear theory based predictions. Linear theory neglects such significant influences as three-dimensional separation and subsequent vortex formation, leading-edge shock-wave detachment and the presence of shock waves on the wing surface. As explained in Ref. [115], for the subsonic-leading edge lifting wing, the attachment line of the flow is below the leading edge on the

lower surface of the wing and, thus, there is a flow around the leading edge from the lower surface. The expansion of the flow in going around the leading edge results in a high negative pressure peak and subsequent steep pressure recovery. This pressure recovery appears as a strong adverse pressure gradient to the three-dimensional boundary layer and causes it to separate from the surface. When this separation occurs, the boundary layer leaves the wing surface along a swept separation line and rolls into a region of concentrated vorticity which is swept back over the surface of the wing. The effect of this vorticity is to alter the velocity and pressure distribution over the wing. For the supersonic leading edge, adverse pressure gradients due to pressure rise across shock waves also can separate the three-dimensional boundary layer and result in the formation of concentrated vortices.

Conditions for the separated flow around the swept-wing leading edges are described by Kulfan [114], which is essentially a correlation of experimental data in terms of a condition normal to the leading edge. For uncambered wings with sharp leading edges, the boundary region that separates the conditions for which attached flow or leading-edge-separation flow exists, is expressed by

$$M_N = 0.06 + 0.013\alpha_N \tag{1.3}$$

where,

$$\alpha_N = \tan^{-1}(\tan\alpha/\cos\Lambda), \qquad M_N = M\cos\Lambda\sqrt{1+\sin^2\alpha\tan^2\Lambda}\ . \tag{1.4}$$

In a similar manner, delta-wing leeside flow classification into attached and separated flows is presented by Miller and Wood [116], and a sketch from this reference is shown in Figure 1.2. High velocities and accompanying low pressures that occur, as air flows around the leading edge from the stagnation point on the lower surface to the upper surface, can yield leading-edge thrust. However, failure to maintain attached flow at the leading edge results in the loss of leading-edge thrust. Poor lifting efficiency of flat wings, which can only realize a fraction of full theoretical leading-edge

thrust, is attributed to leading-edge separation. An improved method to predict attainable leading-edge thrust and its effect on airplane aerodynamics performance accurately has been developed by Carlson et al. [117]. Leading-edge thrust loss and flow separation is dependent on flight conditions (Mach number and Reynolds number), on wing section geometry (maximum thickness, location of maximum thickness and leading-edge radius), and on the incidence of the wing section leading-edge mean camber surface relative to the local flow.

Design of a wing with proper amount of warping (i.e., camber and twist) has been found to be very effective in suppressing the leading edge vortex. Leading edges of the appropriately cambered and twisted wings are aligned with the local flow direction and attachment line lies along the leading edge. As shown Figure 1.3 from [114], expansion over such a wing's upper surface is greatly reduced, thereby eliminating the strong adverse pressure gradient near the leading edge. Action of the reduced expansion pressure on the relatively large "shoulder" area of the airfoil ensures the thrust force for cambered airfoil sections of such a wing.

Carlson and Mann [118] have surveyed research on drag-due-to-lift minimization at supersonic speeds and presented an empirical method for the the selection of proper design lift coefficient and for the estimation of achievable aerodynamic performance. Item and Baysal [93] have demonstrated a wing section design procedure for supersonic flow in which the optimization process produced camber and twist in an automated manner, starting from a symmetrical section.

It is recognized that twist- and camber-assisted wings generally offer better performance as compared to flat wings, yet it still falls short of theoretical potential. To realize full potential of camber and twist, it should be incorporated in conjunction with the wing thickness and body effects [114], such that, (i) strong spanwise flow, (ii) extremely high suction pressures, (iii) inboard shock separation, and (iv) trailing edge shock separation, are avoided.

The concept of attached flow is used by Pittman et al. [119] for the design of wings to produce low drag due to lift at high-lift conditions. The basic idea is to generate the high levels of lift using the low pressures resulting from the upper-surface supercritical crossflow, while minimizing the drag by avoiding large pressure gradients which separate the flow, and by avoiding strong shocks which result in energy losses. The wing design with the camber and thickness modification and attached cross flow concept has been performed for the design conditions of lift coefficient of 0.4, 12-degree angle of attack, and 1.62 Mach number. Benefits of such concepts have been verified by experimental measurements.

Wood [120] has performed empirical correlation of experimental data and theoretical analysis, and developed a set of graphs to quantify the inviscid aerodynamics of supersonic delta wings. In this study, zero-lift wave-drag characteristics of delta wings with various airfoil sections are determined through nonlinear analysis. Aerodynamic characteristics of delta wings at lifting conditions have been evaluated for the effects of wing leading-edge sweep, leading-edge bluntness, and wing thickness and camber. Findings have been summarized in the form of graphs. Empirical curves have been developed for the lift-curve slope, nonlinear lift effects and maximum lift. Also, impact of various airfoil parameters, wing leading-edge sweep, and lift coefficient on the drag-due-to-lift characteristics has been shown theoretically. The graphs, which detail the aerodynamics of delta wings at both zero-lift and lifting conditions, were employed to define a preliminary wing design approach, in which both the low-lift and high-lift design criteria were combined to define a feasible design space.

Another approach, counter to the traditional three-dimensional wing camber design, is the natural flow wing design approach developed by Wood and Bauer [121], in which contours of upper and lower surfaces are independently developed, such that maximum advantage of the naturally occurring flowfield and resultant pressure distribution is obtained. In this approach, a combination of optimum thickness,

leading-edge bluntness and asymmetry modification has been employed to design a multipoint supersonic wing.

Bushnell [2] mentions that, thus far, major drag reduction efforts for HSCT-class aircraft have concentrated upon wave drag reduction. However, drag breakdown of a typical HSCT design, crudely assuming no pressure drag associated with flow separation and neglecting trim drag, is on the order of 1/3 skin friction, 1/3 wave drag (mostly volume wave drag, but including sizable wave drag due to lift) and 1/3 vortex drag. Therefore, it is argued that the supersonic aircraft drag reduction problem should be tackled from an overall viewpoint in which all the components of drag are reduced. Emerging possibilities, such as roughness minimization, transition estimation and delay for the natural laminar flow, laminar flow control by wall cooling and suction, and turbulent drag reduction methods, are discussed in Ref. [2] for viscous drag reduction. In order to reduce the vortex drag due to lift, energy/thrust extraction from the tip vortex, alteration of tip boundary conditions, such as eliminating the tip altogether by employing a ring wing or adding mass in the tip region, and non-planar vortex sheet approaches (such as winglets) are suggested. For wave drag reduction, in addition to the classical approaches such as wing sweep, area ruling, increased effective body length/thin sections and gradual compression, wave drag reduction via favorable interference is suggested. Flow separation control, rather than its avoidance, is also worth considering so that the full potential of several drag reduction techniques can be unleashed.

Kulfan [122] has investigated several favorable aerodynamic interference concepts to increase $(L/D)_{max}$. Supersonic biplane, caret wing, Nonweiler wing, and parasol wing concepts were studied. The parasol wing concept was found to yield the greatest potential aerodynamic benefits.

Finally, the dependence of $(L/D)_{max}$ on speed is given by Küchemann's empirical rule as

$$\left(\frac{L}{D}\right)_{max} = \frac{4(M+3)}{M} \quad . \tag{1.5}$$

Current target values for $(L/D)_{max}$ on the order of 10 have been fixed for the realization of an HSCT-class supersonic aircraft. Potential benefits of various drag reduction techniques have been delineated in Figure 1.4 taken from Ref. [122], which provides drag boundaries for a typical HSCT configuration.

1.2 Present Study

1.2.1 Scope and Objectives

Survey of the relevant literature on wing aerodynamics has highlighted the complex interplay of various physical phenomena and helped to recognize stringent requirements of modern aircraft design. The trend in design is for the exploitation, rather than avoidance, of the aerodynamic nonlinearities and interferences. Design practice needs to divest itself from traditional design concepts and accommodate evaluation of the vast expanse of design possibilities in a cost-effective manner. To this end, a necessity for developing an automated design methodology, with an excellent learning potential, has been observed.

A literature survey on various aerodynamic design methods and its elements has revealed that gradient-based methods, with the capability for the analytical evaluation of sensitivities, are most appropriate to fulfill all the requirements of the modern aerodynamic design highlighted previously. Also it is realized that a realistic gradient-based method capable of handling large-scale problems and producing reliable designs should be developed. This goal has provided the impetus for the present study and led to the formulation of the objectives for the present study including:

(1) To develop ADI-based algorithms to solve the resulting 3D sensitivity equations, as well as the flow equations; incorporate them into a design procedure amenable to accommodate large-scale design problems.

(2) To extend this procedure to the Navier-Stokes equations, in order to capture important physical phenomenon of practical consequence in the design optimization study, whenever needed.

(3) To establish the validity of these procedures by comparing with the existing results and formulate algorithmic strategies related to the wing design optimization study.

(4) To demonstrate these capabilities by performing design optimization on a model arrow wing problem of differing scales and differing levels of modeling accuracy. These studies were also performed to assess the the reliability and practicality of the simplified optimization studies.

1.2.2 Outline

The present study consists of seven chapters. The background, literature review, objectives, and outline of the present work have been presented in this chapter. Chapter 2 describes various aspects of the present flow analysis method, such as governing equations, boundary conditions, and solution algorithm. Chapter 3 discusses details of the sensitivity analysis formulations, solution methodology, and its relevance for the flowfield approximation. Chapter 4 elucidates the salient features of the optimization procedure, surveys various surface representation methods, and presents the surface representation method incorporated in this work. Chapters 5 and 6 present inviscid and viscous optimization results respectively. Finally, conclusions of the present investigation and recommendations for the future enhancement are given in Chapter 7.

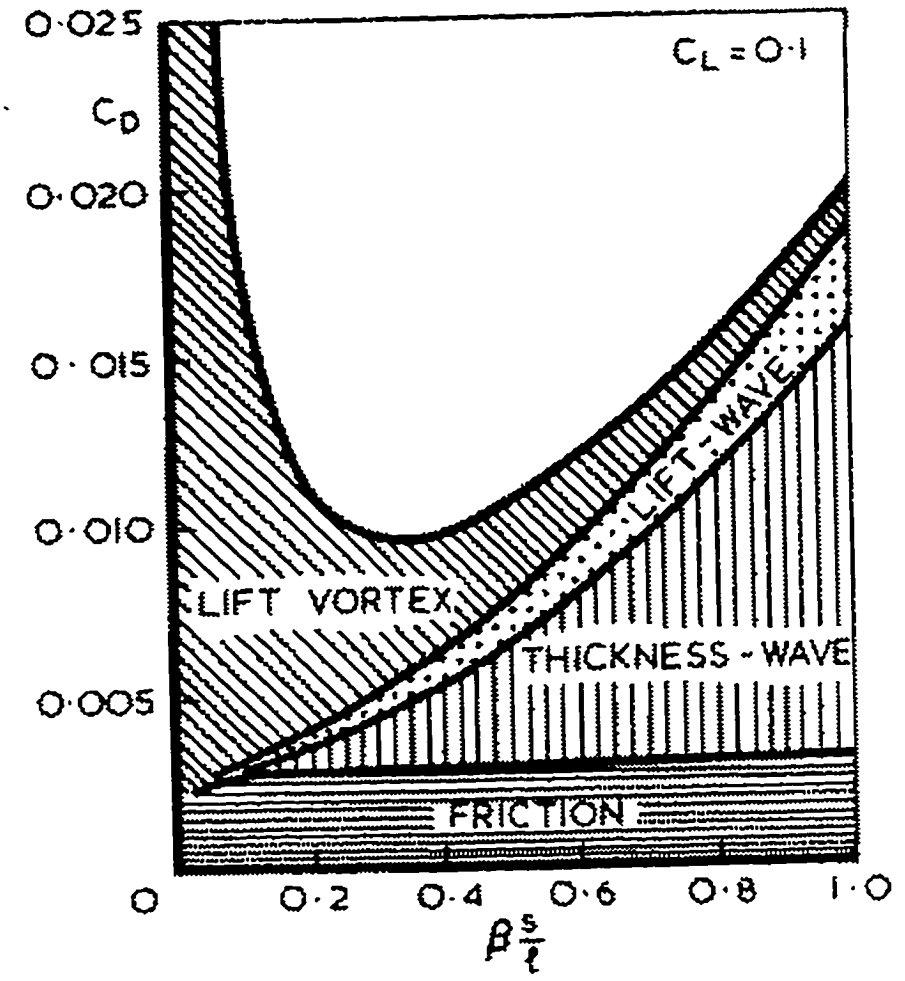


Fig. 1.1 Typical contributions to the overall drag coefficient at $M_\infty = 2$ (Ref. [110]).

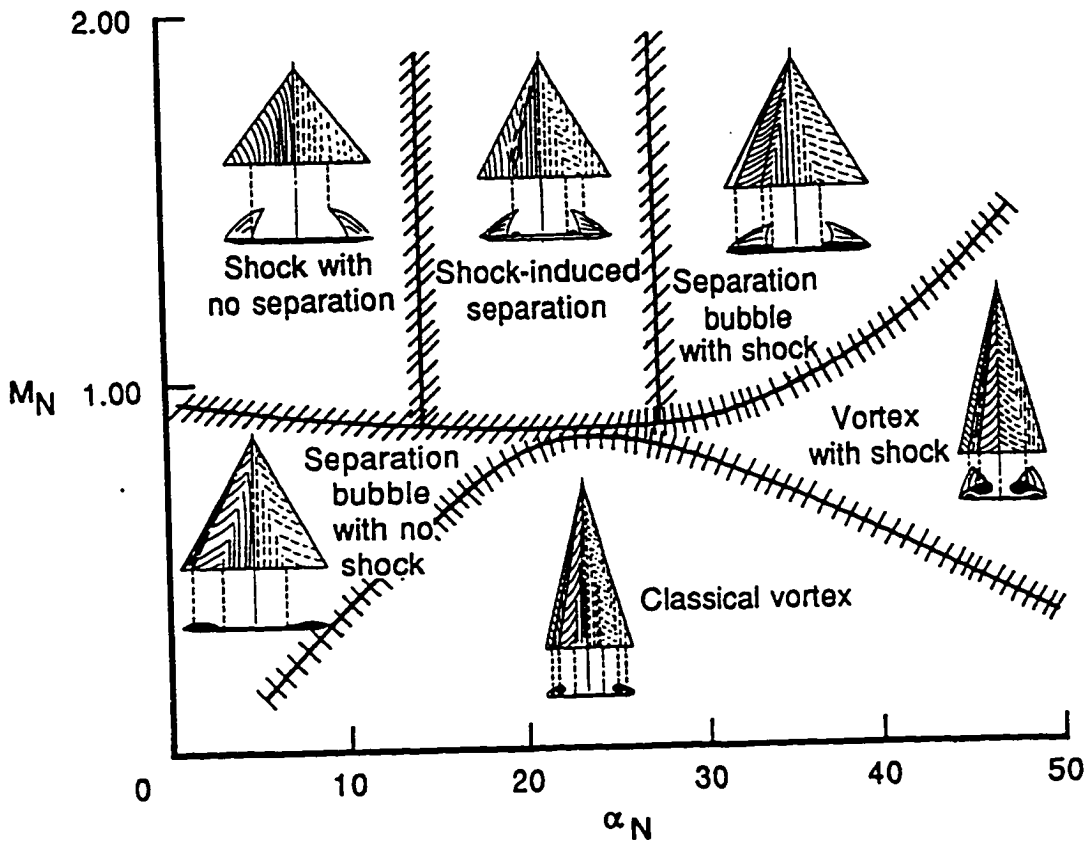


Figure 1.2 Lee-side flow classification chart for delta wing (Ref. [116]).

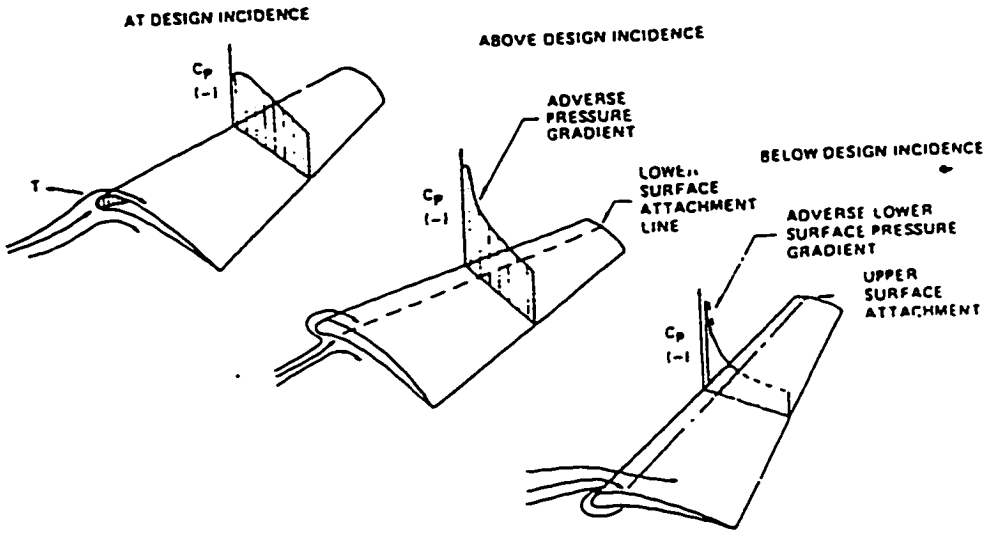


Figure 1.3 Pressure distribution on highly swept cambered wings (Ref. [114]).

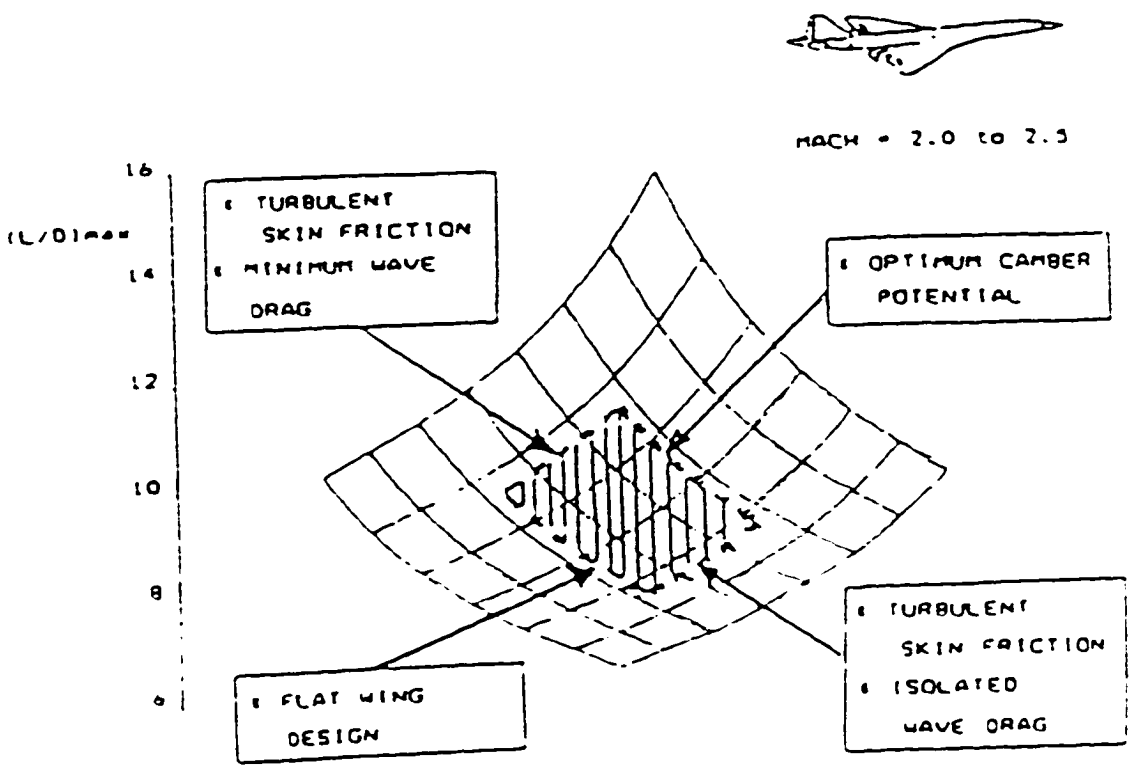


Figure 1.4 Drag boundaries for typical HSCT configuration (Ref. [122]).

Chapter 2

FLOW ANALYSIS

With the advent of faster computer hardware and very efficient and accurate solution algorithms and grid generation techniques, application of CFD has become synonymous with the flow analysis in most cases. Flow analysis is performed by solving a set of coupled partial differential equations which govern the motion of Newtonian fluid and include conservation of mass, momentum and energy. These equations, in their complete form and without any simplifying assumptions, are known as the Navier-Stokes equations.

Due to the large amount of computer time and memory required, the Navier-Stokes equations in their complete form are difficult to solve for a single flow analysis. This may be impractical for repetitive flow analyses, as in the numerical optimization procedure, even with the modern arsenal of computational tools. In order to veer around this difficulty, certain simplifying assumptions are introduced in this set of equations with emphasis on capturing essential physics of the flow. One such approach is formulated when viscosity and heat transfer are neglected from these equations, and the Euler equations are obtained. Another approach applicable for the inviscid and viscous regions is to solve the reduced Navier-Stokes equations which fall between the complete Navier-Stokes equations and the boundary-layer equations in terms of

complexity [123]. Two popular sets of equations in this category are the thin-layer Navier-Stokes equations and the parabolized Navier-Stokes equations.

The principal advantage of the former over the latter lies in its validity for broader flow regime, subsonic to supersonic, and its ability to handle streamwise flow separation. In the thin-layer Navier-Stokes equations, the viscous terms containing derivatives in the directions parallel to the body surface are neglected. This simplification reduces the computation time due to elimination of terms from the full Navier-Stokes equations. As mentioned in Ref. [124], on the same grid, the cost of evaluating the viscous terms in the thin-layer Navier-Stokes equations relative to the inviscid Euler equations is less than a 2% increase in CPU time per iteration for a typical implicit upwind code.

2.1 Governing Equations

The present study employs either the Euler or the thin-layer Navier-Stokes equations, depending on an embedded switch, as the governing equations for the flow analysis. The governing equations for the thin-layer approximations to the three-dimensional, time-dependent, compressible Navier-Stokes equations, transformed in generalized coordinates and conservation form can be expressed as [125]

$$\frac{\partial}{\partial t}(\hat{Q}) + \frac{\partial}{\partial \xi}(\hat{F}) + \frac{\partial}{\partial \eta}(\hat{G}) + \frac{\partial}{\partial \zeta}(\hat{H} - n\hat{H}_v) = 0 \tag{2.1}$$

where n is the switch for using the Euler equations,

$$n = \begin{cases} 0 & \text{Euler equations} \\ 1 & \text{Thin-layer Navier-Stokes equations} \end{cases}$$

In this equation, the terms are defined as:

$$\hat{Q} = \frac{Q}{J} = \frac{1}{J} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e_t \end{Bmatrix} \tag{2.2}$$

$$\hat{F} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ (\rho e_t + p)U \end{Bmatrix}; \quad \hat{G} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ (\rho e_t + p)V \end{Bmatrix}; \quad \hat{H} = \frac{1}{J}\begin{Bmatrix} \rho W \\ \rho W u + \zeta_x p \\ \rho W v + \zeta_y p \\ \rho W w + \zeta_z p \\ (\rho e_t + p)W \end{Bmatrix} \tag{2.3}$$

$$\hat{H}_v = \frac{M_\infty}{Re_L}\frac{\mu}{J}\begin{Bmatrix} 0 \\ \phi_1 u_\zeta + \zeta_x \phi_2 \\ \phi_1 v_\zeta + \zeta_y \phi_2 \\ \phi_1 w_\zeta + \zeta_z \phi_2 \\ \phi_1\left[\left(\frac{q^2}{2}\right)_\zeta + \frac{1}{Pr(\gamma-1)}(a^2)_\zeta\right] + W\phi_2 \end{Bmatrix} \tag{2.4}$$

where J is the Jacobian of transformation,

$$J = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} \tag{2.5}$$

U, V, and W are the contravariant velocity components,

$$\begin{aligned} U &= \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{2.6}$$

and

$$\phi_1 = \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \tag{2.7}$$

$$\phi_2 = (\zeta_x u + \zeta_y v + \zeta_z w)/3 \tag{2.8}$$

$$q^2 = u^2 + v^2 + w^2 \tag{2.9}$$

$$Re_L = \frac{\rho_\infty^* q_\infty^* L^*}{\mu_\infty^*} \ . \tag{2.10}$$

Pressure p and total energy per unit volume are related by the perfect gas state equation,

$$p = (\gamma - 1)[\rho e_t - \rho(u^2 + v^2 + w^2)/2] \ , \tag{2.11}$$

and Re_L is the freestream Reynolds number, μ is nondimensional molecular viscosity which is calculated by Sutherland's law, a is the speed of sound, and Pr is Prandtl

number. Also Stokes' hypothesis for bulk viscosity, $\lambda + \frac{2}{3}\mu = 0$, is invoked for the computations. In these equations, reference length L^*, and freestream values of the density, ρ_∞, sound speed, a_∞, and viscosity, μ_∞, are used for the nondimensionalization as

$$\rho = \frac{\rho^*}{\rho_\infty} ; u = \frac{u^*}{a_\infty} ; v = \frac{v^*}{a_\infty} ; w = \frac{w^*}{a_\infty} ; p = \frac{p^*}{\rho_\infty a_\infty^2} ; \mu = \frac{\mu^*}{\mu_\infty} \ . \tag{2.12}$$

2.2 Solution Algorithm

The integral form of the conservation laws expressed by Eq. (2.1) can be written [126] for a discrete and stationary volume as

$$\frac{\partial}{\partial t} \iiint_V \hat{Q} \, dV + \iint_S \vec{F} \cdot \hat{n} \, dS = 0 \ , \tag{2.13}$$

where volume V is bounded by surface S and $\hat{n} = n_x\hat{\imath} + n_y\hat{\jmath} + n_z\hat{k}$ is the unit normal vector pointing outward from the surface and flux vector $\vec{F}$ is given by :

$$\vec{F} = \hat{F}\,\hat{\imath} + \hat{G}\,\hat{\jmath} + (\hat{H} - \hat{H}_v)\hat{k} \ . \tag{2.14}$$

Applying this equation to each cell of the computational domain bounded by the constant ξ, η and ζ lines, a semi-discrete, finite volume formulation is obtained. For a representative cell, (i,j,k) of unit volume having $\Delta\xi = 1$, $\Delta\eta = 1$, and $\Delta\zeta = 1$, this formulation can be shown as,

$$\frac{1}{J}\left(\frac{\partial Q}{\partial t}\right)_{i,j,k} + (\hat{F}_{i+1/2,j,k} - \hat{F}_{i-1/2,j,k}) + (\hat{G}_{i,j+1/2,k} - \hat{G}_{i,j+1/2,k}) + (\hat{H}_{i,j,k+1/2} - \hat{H}_{i,j,k-1/2}) - (\hat{H}_{vi,j,k+1/2} - \hat{H}_{vi,j,k-1/2}) = 0 \ . \tag{2.15}$$

In Eq. (2.15), Q does not represent a value at the point (i,j,k), instead it is an average value over the cell volume surrounding the point. Expressing the flux terms of the Eq. (2.15) in terms of the steady-state residual, R, the following form is obtained:

$$\frac{1}{J}\left(\frac{\partial Q}{\partial t}\right) + R(Q) = 0 \tag{2.16}$$

Applying Euler implicit time integration scheme to Eq. (2.16) yields,

$$\frac{1}{J}\frac{\Delta Q}{\Delta t} + R(Q^{n+1}) = 0 \tag{2.17}$$

where, $\Delta Q = Q^{n+1} - Q^n$. Eq. (2.17) is a set of coupled nonlinear equations. In order to solve them numerically, linearization about time level n is employed, which yields,

$$\left[\frac{I}{J\Delta t} + \frac{\partial R}{\partial Q}\right]^n \{\Delta Q^n\} = -\{R^n(Q)\} \tag{2.18}$$

In Eq. (2.18), $\frac{\partial R}{\partial Q}$ is a flux Jacobian which is a large, banded, sparse matrix with 5×5 block elements for the three-dimensional problems. Rewriting Eq. (2.18) by expressing $\frac{\partial R}{\partial Q}$ in the expanded form yields:

$$\left[\frac{I}{J\Delta t} + \delta_\xi \frac{\partial \hat{F}}{\partial Q} + \delta_\eta \frac{\partial \hat{G}}{\partial Q} + \delta_\zeta \frac{\partial \hat{H}}{\partial Q} - \delta_\zeta \frac{\partial \hat{H}_v}{\partial Q}\right]^n \{\Delta Q^n\} = -\{R^n(Q)\} \tag{2.19}$$

Several approaches have been pursued to obtain steady state solution of Eq. (2.19). One approach is called Newton's method which sets $\Delta t \to \infty$ in Eq. (2.19) and solves for ΔQ by forward and backward substitutions [127]. This approach exhibits quadratic convergence but has the stipulation that the initial guess to the solution be in the domain of attraction to the final solution. Failure to satisfy this restriction results in an oscillatory behavior and eventual divergence of the solution. In order to overcome this problem, a controlled amount of relaxation can also be added [128] into the left-hand-side matrix, which is commonly known as the modified Newton's method. Due to the large memory requirements of such methods, practical application of the same is thwarted. Another approach is globalizing the nonlinear convergence through the use of artificial time-stepping, or pseudotransient continuation [129]. Pseudotranscience is a framework for approaching the steady-state Newton method through a time step that is built up from small to infinite values, unlike the time step sequence for approximate factorization schemes which have an error proportional to Δt.

Other iterative Newton methods, such as preconditioned conjugate methods, have also been formulated and applied [130, 36] which reduce the memory requirement of Newton's method and ensure higher convergence rates. Even then, for large problems, the required memory storage is rather prohibitive. This can be circumvented by resorting to a first-degree iterative scheme, where the true Jacobian matrix of the Eq. (2.19) is spatially split into more memory-manageable parts using the approximate factorization process and the ADI scheme as formulated in Ref. [131]. This approach is followed in the present study.

Approximate factorization of the matrix of Eq. (2.19) produces a sequence of three simpler operators as

$$\left[\frac{I}{J\Delta t}+\delta_\xi\frac{\partial\hat{F}}{\partial Q}\right]\left[\frac{I}{J\Delta t}\right]^{-1}\left[\frac{I}{J\Delta t}+\delta_\eta\frac{\partial\hat{G}}{\partial Q}\right]\left[\frac{I}{J\Delta t}\right]^{-1}\left[\frac{I}{J\Delta t}+\delta_\eta\frac{\partial(\hat{H}-\hat{H}_v)}{\partial Q}\right]$$
$$\{\Delta Q^n\}=-\{R^n(Q)\} \quad (2.20)$$

The solution procedure then involves, solving three one-dimensional problems as,

$$\begin{aligned}\left[\frac{I}{J\Delta t}+\delta_\xi\frac{\partial\hat{F}}{\partial Q}\right]\{\Delta Q^{n**}\} &= -\{R^n(Q)\}\\ \left[\frac{I}{J\Delta t}+\delta_\eta\frac{\partial\hat{G}}{\partial Q}\right]\{\Delta Q^{n*}\} &= \left(\frac{I}{J\Delta t}\right)\{\Delta Q^{n**}\}\\ \left[\frac{I}{J\Delta t}+\delta_\zeta\frac{\partial(\hat{H}-\hat{H}_v)}{\partial Q}\right]\{\Delta Q^{n}\} &= \left(\frac{I}{J\Delta t}\right)\{\Delta Q^{n*}\}\end{aligned} \quad (2.21)$$

and advancing the solution in time to the next step until steady state as,

$$\left\{Q^{n+1}\right\}=\{Q^n\}+\{\Delta Q^n\} \quad (2.22)$$

The rate of convergence is unaffected [128] by the order of the differencing on either side of the equation, therefore, to reduce the bandwidth and computational work, the flux Jacobian in Eq. (2.21), is typically constructed in an approximate manner by employing first-order discretization of the flux terms. Hence, each problem of Eq. (2.21) requires the inversion of a block tridiagonal matrix.

This algorithm has a factorization error of $\vartheta(\Delta t^2)$ which dominates the size of the time step and yields only linear convergence. For a steady state solution, local time stepping is employed for each cell of the computational domain which is computed for the inviscid case based on the formula,

$$\Delta t = CFL\{|U| + |V| + |W| + a(|\nabla\xi| + |\nabla\eta| + |\nabla\zeta|)\}^{-1} \tag{2.23}$$

where CFL is the Courant number which is linearly increased to its maximum value of $7 \sim 8$ in the present computations. An alternative to the approximate factorization method has been demonstrated in Ref. [132] where applicability of Krylov-accelerated domain decomposition algorithms to a classical two-dimensional Euler test problem is shown. This approach renders twin advantages of larger time steps and algorithmic parallelism.

The flux terms in the Eq. (2.19) are evaluated at each cell surface. Central schemes for the spatial differencing of the flux terms generate oscillations in the vicinity of the discontinuity and require addition of artificial viscosity terms to damp it. On the other hand, upwind methods are naturally dissipative and do not require additional dissipative terms. Also upwind schemes account for the signal propagation characteristics of the hyperbolic governing equations. In the present study, the inviscid (hyperbolic) fluxes consisting of pressure and convective terms are upwind differenced using the flux-vector splitting method of Van Leer [133] where as viscous (elliptic) flux terms are centrally differenced. Van Leer's flux vector splitting method ensures continuous differentiability of the fluxes, unlike other upwind schemes, such as Steger-Warming method and Roe's flux-difference method. This feature is very important in computing correct sensitivity derivatives and therefore, has led to the choice of Van Leer's flux-vector splitting method. In the flux splitting method for the generalized coordinates [134], Monotone Upstream-centered Schemes for Conservative Laws (MUSCL) approach is formulated where inviscid fluxes are written

conservatively as flux balances across the cell and this is followed by the flux splitting. Present study includes the procedure which follows this approach. For example, in the ξ direction:

$$\left(\frac{\partial \hat{F}}{\partial \xi}\right)_i = \hat{F}_{i+1/2} - \hat{F}_{i-1/2} \tag{2.24}$$

and interface fluxes are constructed as,

$$\hat{F}_{i+1/2} = \hat{F}(Q^{\pm}_{i+1/2}) \tag{2.25}$$

Conserved flow variables on the cell interfaces represented by $(Q^{\pm}_{i+1/2})$ are constructed by extrapolation from their cell centered values based on the MUSCL scheme. Higher-order extrapolation is accomplished with the $\phi - \kappa$ interpolating polynomials given as,

$$\begin{aligned} Q^{-}_{i+1/2} &= Q_i + \left\{\frac{\phi}{4}[(1-\kappa)\nabla + (1+\kappa)\Delta]\right\}_i \\ Q^{+}_{i+1/2} &= Q_{i+1} - \left\{\frac{\phi}{4}[(1+\kappa)\nabla + (1-\kappa)\Delta]\right\}_{i+1} \end{aligned} \tag{2.26}$$

where, $\nabla Q_i = Q_i - Q_{i-1}$, and $\Delta Q_i = Q_{i+1} - Q_i$.

For the different values of the parameters ϕ and κ, different order accurate spatial schemes are obtained. For $\phi = 0$, a first-order fully upwind scheme is obtained. For $\kappa = -1$, second-order fully upwind, and for $\kappa = 1/3$, third-order upwind-biased differencing is obtained.

In the present study, third-order upwind biased differencing is used. However, when an upwind-biased scheme is used, numerical oscillations such as overshoots and undershoots appear in the presence of large flow gradients. In order to suppress this undesirable tendency of the scheme, a flux limiter is used to switch an upwind-biased scheme to a first-order fully upwind scheme and ensure monotonic interpolation in the regions of discontinuities. The present study makes use of Van Albada limiter [135] which modifies the Eq. (2.26) as shown below.

$$
\begin{aligned}
Q^{-}_{i+1/2} &= Q_i + \left\{ \frac{S}{4}[(1-\kappa S)\nabla + (1+\kappa S)\Delta] \right\}_i \\
Q^{+}_{i+1/2} &= Q_{i+1} - \left\{ \frac{S}{4}[(1+\kappa S)\nabla + (1-\kappa S)\Delta] \right\}_{i+1}
\end{aligned}
\tag{2.27}
$$

where,

$$
S_i = \frac{2\Delta_i \nabla_i + \epsilon}{\Delta_i^2 + \nabla_i^2 + \epsilon} \tag{2.28}
$$

and ϵ is a small number (e.g. 1.0e-06), to prevent division by zero in the smooth regions. This limiter is a continuously differentiable one, and this very feature has made it the choice for the present study as it leads to the sensitivities discussed in Chapter 3.

In Van Leer's flux splitting method, the inviscid flux vectors in generalized coordinates are split as shown below for the representative ξ direction. The flux $\hat{F}$ is split depending on the local contravariant Mach number, M_ξ defined as,

$$
\begin{aligned}
M_\xi &= \frac{\bar{u}}{a} \\
\bar{u} &= \frac{U}{|\nabla \xi|}
\end{aligned}
\tag{2.29}
$$

For supersonic flow, $|M_\xi| \geq 1$,

$$
\begin{aligned}
F^+ = F \qquad F^- = 0 \qquad \text{for } |M_\xi| \geq 1 \ , \\
F^+ = 0 \qquad F^- = F \qquad \text{for } |M_\xi| < 1 \ .
\end{aligned}
\tag{2.30}
$$

For subsonic flow, $|M_\xi| < 1$,

$$
\hat{F}^{\pm} = \frac{|\nabla \xi|}{J} \begin{Bmatrix} f^{\pm}_{mass} \\ f^{\pm}_{mass}[\hat{\xi}_x(-\bar{u} \pm 2a)/\gamma + u] \\ f^{\pm}_{mass}[\hat{\xi}_y(-\bar{u} \pm 2a)/\gamma + v] \\ f^{\pm}_{mass}[\hat{\xi}_z(-\bar{u} \pm 2a)/\gamma + w] \\ f^{\pm}_{energy} \end{Bmatrix} \ , \tag{2.31}
$$

where

$$f^{\pm}_{mass} = \pm\rho a(M_\xi \pm 1)^2/4 \tag{2.32}$$

$$f^{\pm}_{energy} = \pm f^{\pm}_{mass}\left[\frac{(1-\gamma)\bar{u}^2 \pm 2(\gamma-1)\tilde{u}a + 2a^2}{(\gamma^2-1)} + \frac{u^2+v^2+w^2}{2}\right] . \tag{2.33}$$

Since the present method is finite-volume, the metric terms in the above equations are evaluated geometrically. The term, $|\nabla\xi|/J$ denotes the surface area of the cell at the interface in ξ direction, $1/J$ is the cell volume and and $\hat{\xi}_x$, $\hat{\xi}_y$, and $\hat{\xi}_z$ are the direction cosines of the cell interface, defined as

$$(\hat{\xi}_x, \hat{\xi}_y, \hat{\xi}_z) = (\xi_x, \xi_y, \xi_z)/|\nabla\xi| . \tag{2.34}$$

The split-flux differences are implemented as a flux balance across each cell. For instance, flux balance in the ξ direction at a point (i,j,k) is given by

$$\begin{aligned}\delta^-_\xi \hat{F}^+ + \delta^+_\xi \hat{F}^- &= \left[\hat{F}^+(Q^-_{i+1/2,j,k}) + \hat{F}^-(Q^+_{i+1/2,j,k})\right] - \\ &\quad \left[\hat{F}^+(Q^-_{i-1/2,j,k}) + \hat{F}^-(Q^+_{i-1/2,j,k})\right] .\end{aligned} \tag{2.35}$$

A similar expression can be obtained for the inviscid flux balance in the η and ζ directions. Spatial derivatives of viscous flux, $\hat{H}_v$ in the Eq. (2.1) are computed as flux balance across the cell interfaces, Hence,

$$\left(\frac{\partial \hat{H}_v}{\partial\zeta}\right)_{i,j,k} = (\hat{H}_v)_{i,j,k+1/2} - (\hat{H}_v)_{i,j,k-1/2} . \tag{2.36}$$

The viscous flux $\hat{H}_v$ at the interfaces is evaluated based on Eq. (2.4). To maintain second-order accuracy, its derivative terms are evaluated with one-sided differencing.

By grouping the inviscid flux terms, as represented by Eq. (2.35) for the ξ direction, for all the three coordinate directions and viscous terms represented by Eq. (2.36) together, the residual R of the Eq. (2.19) can be written as,

$$\begin{aligned}R^n_{i,j,k}(Q) &= \left[\hat{F}^+(Q^-_{i+1/2,j,k}) + \hat{F}^-(Q^+_{i+1/2,j,k})\right] - \\ &\quad \left[\hat{F}^+(Q^-_{i-1/2,j,k}) + \hat{F}^-(Q^+_{i-1/2,j,k})\right] +\end{aligned}$$

$$
\begin{aligned}
&\left[\hat{G}^{+}(Q^{-}_{i,j+1/2,k}) + \hat{G}^{-}(Q^{+}_{i,j+1/2,k})\right] - \\
&\left[\hat{G}^{+}(Q^{-}_{i,j-1/2,k}) + \hat{G}^{-}(Q^{+}_{i,j-1/2,k})\right] + \\
&\left[\hat{H}^{+}(Q^{-}_{i,j,k+1/2}) + \hat{H}^{-}(Q^{+}_{i,j,k+1/2})\right] - \\
&\left[\hat{H}^{+}(Q^{-}_{i,j,k-1/2}) + \hat{H}^{-}(Q^{+}_{i,j,k-1/2})\right] - \\
&\left[\hat{H}_{v}(Q_{i,j,k+1}, Q_{i,j,k}) - \hat{H}_{v}(Q_{i,j,k}, Q_{i,j,k-1})\right] \quad .
\end{aligned}
\tag{2.37}
$$

The algorithm described above is implemented by constructing a flow domain and discretizing it into number of finite volumes. In the present study, this is accomplished by initially generating C-H grids. For this purpose, GRIDGEN3D [136] was used to construct grids on the surface of the flow domain, and subsequently, CSCMDO [137] was used to generate the volume grid.

2.3 Initial and Boundary Conditions

Freestream conditions are specified as the initial conditions. Boundary conditions are enforced explicitly at every time step in the iteration process. For the solution of inviscid equations, flow tangency condition is imposed by setting the contravariant component of the velocity normal to the wall as zero. Pressure and density at the wall are evaluated using the zeroth order extrapolation from the interior neighbouring cell's corresponding values. Hence,

$$
W = 0 \quad ; \quad \frac{\partial p}{\partial n} = 0 \quad ; \quad \frac{\partial \rho}{\partial n} = 0 \quad . \tag{2.38}
$$

For the viscous flow computations, the no-slip boundary condition is imposed at the wall, which is also considered to be adiabatic. Pressure at the wall is again evaluated based on zeroth order extrapolation and density is calculated based on the equation of state. These can be represented as

$$
u = v = w = 0 \quad ; \quad \frac{\partial p}{\partial n} = 0 \quad ; \quad \frac{\partial T}{\partial n} = 0 \quad . \tag{2.39}
$$

Farfield boundary conditions for inviscid and viscous flows are determined using the Riemann invariants where direction of the information travel is dependent on whether the flow is subsonic or supersonic. The Riemann invariants are given by

$$R^{\pm} = U \pm \frac{2a}{\gamma - 1} \ , \tag{2.40}$$

which are constant along the characteristic lines given by $\left(\frac{\partial \xi}{\partial t}\right)^{\pm} = U \pm a$. More details on the far field boundary conditions can be found in Ref. [138].

Chapter 3

SENSITIVITY ANALYSIS

Sensitivity analysis refers to the determination of sensitivity information, which is one of the key elements of any gradient-based numerical optimization procedure. In design optimization of aerodynamic origin, sensitivity analysis is a major contributor to the computational cost of the optimization procedure. For various methods of sensitivity analysis, computational cost and complexity of sensitivity analysis are generally in inverse proportions.

Sensitivity information is typically delivered to the optimization algorithm, in terms of gradients of the objective function, F, and the flow-related constraints, G_j with respect to the design variables, D [41]. These gradients are also known as sensitivity coefficients. In aerodynamic applications, the objective functions and constraints are functionally dependent on the vector of design variables, D, and the vector of conserved flow variables, Q, which is an implicit function of design variables [41]. This functional relationship of the objective function and flow constraints can be symbolically expressed as $F(D, Q(D))$ and $G(D, Q(D))$.

3.1 Sensitivity Coefficients

As evident from the literature survey in Subsection 1.1.1, various approaches toward evaluation of sensitivity coefficients comprise of finite-difference, continuous,

and discrete methods of sensitivity analysis. This diversity in the method of sensitivity coefficients determination has been the major differentiating factor between many numerical optimization approaches currently pursued [98]. In the present study, sensitivity coefficients have been evaluated using discrete method, and the results have been compared with finite difference method for validation. Hence, focus will be shifted on these two methods for further illustration.

3.1.1 Finite Difference Approach

In this approach, derivatives of F and G_j are evaluated based upon their truncated Taylor series expansion in terms of the design variable D_i. Depending on the order of the truncated terms, various-order accurate sensitivity coefficients are obtained. Typically, first-order or second-order approximation is used in sensitivity coefficients evaluation. For a specific design variable, D_i, among the vector of design variables, the first-order forward approximation is given by,

$$\frac{\partial F}{\partial D_i} \approx \frac{F\left[D + \Delta D_i, Q\left(D + \Delta D_i\right)\right] - F\left[D, Q\left(D\right)\right]}{\Delta D_i} \tag{3.1}$$

For the second-order accurate sensitivity coefficients, central difference approximation of the derivatives can be written in a similar manner. For the NDV design variables, first-order-accurate finite-difference approximation would require $NDV + 1$ flow analyses, whereas, second-order-accurate approximation requires $2 * NDV$ flow analyses. Appeal of this method lies in its simplicity as flow analysis codes can be repeatedly used without any modifications. However, this apparent simplicity is fraught with the risk of computing erroneous coefficients as the appropriate perturbation size of design variables, ΔD, is not known *a priori*. Its use in practical aerodynamic optimization is further hampered due to its lack of computational efficiency. This is borne out of the fact that every gradient evaluation requires a number of flow analyses which scale directly with the number of design variables.

3.1.2 Discrete Sensitivity Analysis Approach

This approach is also known as the quasi-analytical approach. Functional dependence of F and G_j on the design variables, as described previously, is used to invoke the chain rule of differentiation and obtain the sensitivity coefficients,

$$\nabla F = \frac{\partial F(D,Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial F}{\partial D_i}\right)_Q + \left(\frac{\partial F}{\partial Q}\right)_D^T \cdot Q_i'\right]\hat{e}_i \ , \tag{3.2}$$

$$\nabla G_j = \frac{\partial G_j(D,Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial G_j}{\partial D_i}\right)_Q + \left(\frac{\partial G_j}{\partial Q}\right)_D^T \cdot Q_i'\right]\hat{e}_i \ , \tag{3.3}$$

where $i \in 1,\ldots,NDV$, $j = 1,\ldots,NCON_f$ and $Q_i' \equiv (\partial Q/\partial D_i)$. These gradients include the summation of an explicit and an implicit part. The terms $\left(\frac{\partial F}{\partial D}\right)_Q$, and $\left(\frac{\partial G_j}{\partial D}\right)_Q$, which constitute the explicit parts, and $\left(\frac{\partial F}{\partial Q}\right)^T$ and $\left(\frac{\partial G_j}{\partial Q}\right)^T$, which are in the implicit components, are obtained by straightforward differentiations. In order to evaluate the gradients in the above equation, computation of implicit parts is accomplished by the direct computation of Q' in the above equations. Alternatively, surrogate terms are devised for the implicit parts through adjoint variables. These variations under the discrete approach are known as the direct method and the adjoint method [6], which will be explained next in detail.

The residual of the flow equations, as expressed in the Eq. (2.38), is a function of Q, but it also has an implicit relationship with the geometric design variables. Hence in a more general form, the governing equations of three-dimensional, compressible, inviscid or viscous flow are written in the steady-state residual, R, form as

$$R = R\{Q(D), M[X(D)]\} = \vartheta(tol) \ , \tag{3.4}$$

which represents the balance of mass, momentum and energy within the convergence tolerance, $\vartheta(tol)$, of the computation. In Eq. (3.4), X represents the vector of computational grid points and M denotes the coordinate transformation metrics.

By differentiating Eq. (3.4) with respect to design variables and rearranging the linear sensitivity equation is obtained:

$$\left[\frac{\partial R}{\partial Q}\right]_D \left\{\frac{\partial Q}{\partial D}\right\} = -\left\{\frac{\partial R}{\partial D}\right\}_Q \quad . \tag{3.5}$$

Direct Method

For every design variable, Eq. (3.5) is solved for $\frac{\partial Q}{\partial D_i}$,

$$\left\{\frac{\partial Q}{\partial D_i}\right\} = -\left[\frac{\partial R}{\partial Q}\right]_D^{-1} \left\{\frac{\partial R}{\partial D_i}\right\}_Q \quad , \tag{3.6}$$

where, $i \in 1, \ldots, NDV$. The scalar product of $\frac{\partial Q}{\partial D_i}$ with the terms $\left(\frac{\partial F}{\partial Q}\right)^T$, and $\left(\frac{\partial G_j}{\partial Q}\right)^T$ renders the implicit parts of the Eqs. (3.2) and (3.3) and thereby facilitates the evaluation of the sensitivity coefficients.

Adjoint Method

In this method, Eq. (3.5) is not solved in an explicit manner as in the direct method. Instead, Eq. (3.6) is substituted in the implicit parts of Eqs. (3.2) and (3.3) for algebraic manipulation. This results in the following equations:

$$\nabla F = \frac{\partial F(D,Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial F}{\partial D_i}\right)_Q - \left(\frac{\partial F}{\partial Q}\right)_D^T \cdot \left[\frac{\partial R}{\partial Q}\right]_D^{-1} \left\{\frac{\partial R}{\partial D_i}\right\}_Q\right] \tag{3.7}$$

$$\nabla G_j = \frac{\partial G_j(D,Q)}{\partial D_i}\hat{e}_i = \left[\left(\frac{\partial G_j}{\partial D_i}\right)_Q - \left(\frac{\partial G_j}{\partial Q}\right)_D^T \cdot \left[\frac{\partial R}{\partial Q}\right]_D^{-1} \left\{\frac{\partial R}{\partial D_i}\right\}_Q\right] \quad . \tag{3.8}$$

Regrouping the implicit part terms of Eqs. (3.7), (3.8), the vectors of adjoint variables $(\lambda_F, \lambda_{G_j})$ are defined as,

$$\{\lambda_F\}^T = \left(\frac{\partial F}{\partial Q}\right)_D^T \cdot \left[\frac{\partial R}{\partial Q}\right]_D^{-1} \tag{3.9}$$

$$\left\{\lambda_{G_j}\right\}^T = \left(\frac{\partial G_j}{\partial Q}\right)_D^T \cdot \left[\frac{\partial R}{\partial Q}\right]_D^{-1} \quad . \tag{3.10}$$

Evaluation of the sensitivity coefficients entails the determination of vectors $(\lambda_F, \lambda_{G_j})$, necessary equations for which are then rendered by the rearrangement of Eqs. (3.9) and (3.10):

$$\left[\frac{\partial R}{\partial Q}\right]_D^T \cdot \{\lambda_F\} = \left\{\frac{\partial F}{\partial Q}\right\}_D \quad , \tag{3.11}$$

$$\left[\frac{\partial R}{\partial Q}\right]_D^T \cdot \left\{\lambda_{G_j}\right\} = \left\{\frac{\partial G_j}{\partial Q}\right\}_D \quad , \tag{3.12}$$

where, $j \in 1, \ldots, NCON_f$.

Comparison of Direct and Adjoint Methods

For the direct method, Eq. (3.5) needs to be solved once for every design variable. However, the adjoint variable method requires the solution of Eq. (3.11) and Eq. (3.12) once for each flow related constraint, G_j. It is emphasized that the adjoint system of Eqs. (3.11) and (3.12) does not involve any dependence on the design variable vector, D; therefore, for every element, D_i, the adjoint vectors $(\lambda_F, \lambda_{G_j})$ remain unchanged. This implies, that for NDV design variables and $NCON_f$ flow-related constraints, the adjoint method is computationally more efficient than the direct method when [10],

$$NCON_f + 1 > NDV$$

However, a significant offshoot of the direct method is the ready availability of conserved flow variable sensitivities to the design variables, Q', which are necessary in performing approximate flow analysis [10, 11, 6]. The detailed description of the approximate analysis method is deferred, but the amenability of a direct method for its use in conjunction with approximate flow analysis is highlighted here.

3.2 Sensitivity Equation

From the previous discussion, it can be seen that use of the discrete sensitivity analysis approach for the sensitivity coefficients presents the sensitivity equations

either in the form of Eq. (3.5) for the direct method or in the form of Eqs. (3.11) and (3.12) for the adjoint method. These two systems of equations are always simultaneous, linear and algebraic, regardless of the mathematical nature of the flowfield governing equations.

3.2.1 Composition of Sensitivity Equation

The left-hand side of these linear systems consist of the coefficient matrices, $\left[\frac{\partial R}{\partial Q}\right]$ or its transpose, $\left[\frac{\partial R}{\partial Q}\right]^T$. Note that $\left[\frac{\partial R}{\partial Q}\right]$ is the Jacobian of the nonlinear flow equations, which is evaluated at the steady state condition of the flowfield variables. Also, this Jacobian matrix must include consistent linearization of the boundary conditions [33]. Elements of this Jacobian matrix have been discussed in the Chapter 2.

Observing the functional dependence of the residual, R, on the metric terms from Eq. (3.4), the right-hand side of sensitivity Eq. (3.5) can be expressed as,

$$\left\{\frac{\partial R}{\partial D}\right\}_Q = \left[\frac{\partial R}{\partial M}\right]_Q \left\{\frac{\partial M}{\partial D}\right\} \quad . \tag{3.13}$$

This can be expanded based upon Eq. (2.38) for the $(i,j,k)^{\text{th}}$ point as follows:

$$\begin{aligned}
\frac{\partial R_{i,j,k}}{\partial D} = & \left[\frac{\partial \hat{F}^+\left(Q^-_{i+1/2,j,k}, M_{i+1/2,j,k}\right)}{\partial M_{i+1/2,j,k}} + \frac{\partial \hat{F}^-\left(Q^+_{i+1/2,j,k}, M_{i+1/2,j,k}\right)}{\partial M_{i+1/2,j,k}}\right]\frac{\partial M_{i+1/2,j,k}}{\partial D} \\
& - \left[\frac{\partial \hat{F}^+\left(Q^-_{i-1/2,j,k}, M_{i-1/2,j,k}\right)}{\partial M_{i-1/2,j,k}} + \frac{\partial \hat{F}^-\left(Q^+_{i-1/2,j,k}, M_{i-1/2,j,k}\right)}{\partial M_{i-1/2,j,k}}\right]\frac{\partial M_{i-1/2,j,k}}{\partial D} \\
& + \left[\frac{\partial \hat{G}^+\left(Q^-_{i,j+1/2,k}, M_{i,j+1/2,k}\right)}{\partial M_{i,j+1/2,k}} + \frac{\partial \hat{G}^-\left(Q^+_{i,j+1/2,k}, M_{i,j+1/2,k}\right)}{\partial M_{i,j+1/2,k}}\right]\frac{\partial M_{i,j+1/2,k}}{\partial D} \\
& - \left[\frac{\partial \hat{G}^+\left(Q^-_{i,j-1/2,k}, M_{i,j-1/2,k}\right)}{\partial M_{i,j-1/2,k}} + \frac{\partial \hat{G}^-\left(Q^+_{i,j-1/2,k}, M_{i,j-1/2,k}\right)}{\partial M_{i,j-1/2,k}}\right]\frac{\partial M_{i,j-1/2,k}}{\partial D} \\
& + \left[\frac{\partial \widehat{H}^+\left(Q^-_{i,j,k+1/2}, M_{i,j,k+1/2}\right)}{\partial M_{i,j,k+1/2}} + \frac{\partial \widehat{H}^-\left(Q^+_{i,j,k+1/2}, M_{i,j,k+1/2}\right)}{\partial M_{i,j,k+1/2}}\right]\frac{\partial M_{i,j,k+1/2}}{\partial D} \\
& - \left[\frac{\partial \widehat{H}^+\left(Q^-_{i,j,k-1/2}, M_{i,j,k-1/2}\right)}{\partial M_{i,j,k-1/2}} + \frac{\partial \widehat{H}^-\left(Q^+_{i,j,k-1/2}, M_{i,j,k-1/2}\right)}{\partial M_{i,j,k-1/2}}\right]\frac{\partial M_{i,j,k-1/2}}{\partial D}
\end{aligned}$$

$$- \left[\frac{\partial \widehat{H}_v\left(Q_{i,j,k+1}, Q_{i,j,k}, M_{i,j,k+1/2}\right)}{\partial M_{i,j,k+1/2}}\right] \frac{\partial M_{i,j,k+1/2}}{\partial D}$$

$$+ \left[\frac{\partial \widehat{H}_v\left(Q_{i,j,k-1}, Q_{i,j,k}, M_{i,j,k-1/2}\right)}{\partial M_{i,j,k-1/2}}\right] \frac{\partial M_{i,j,k-1/2}}{\partial D} \quad . \tag{3.14}$$

Derivatives of the split fluxes and viscous fluxes with respect to the metric terms can be obtained in a straightforward manner and details may be found in Refs. [139, 138].

Metric terms in this equation have geometric significance as they represent projected areas and computational cell volumes, which in turn are functions of the coordinates, X, of the computational grid. Influence of the geometry-related design variables, which control the body coordinates, X_b, is translated to the the field grid coordinates, X, provided that for every perturbation of design variable vector, re-gridding or grid adaptation is employed. With these considerations, the derivative of metric terms ($M_{i\pm1/2,j\pm1/2,k\pm1/2}$) with respect to the design variables can be expressed symbolically in the following manner:

$$\left\{\frac{\partial M}{\partial D}\right\} = \left[\frac{\partial M}{\partial X}\right]\left[\frac{\partial X}{\partial X_b}\right]\left\{\frac{\partial X_b}{\partial D}\right\} \quad . \tag{3.15}$$

In Eq. (3.15), terms $\frac{\partial X}{\partial X_b}$ and $\frac{\partial X_b}{\partial D}$ denote the grid sensitivities. Procedure for the analytical evaluation of grid sensitivities and metric derivatives, $\frac{\partial M}{\partial D}$, are described in Ref. [139]. Other alternatives for the evaluation of grid sensitivities include finite difference method or automatic differentiation technique in which tools, such as ADIFOR, are employed on the grid generation codes [140].

3.2.2 Solution of Sensitivity Equation

Methods for solving linear systems of algebraic equations can be classified as either direct or iterative methods. Choice of a specific method is governed by the limitations of computer memory and execution time requirements. Computer memory requirements of a direct method are generally very high, and, to alleviate this problem, sparsity of the coefficient matrix is exploited for the storage of matrix elements.

Accordingly, sparse solvers or banded solvers are employed for the solution of the sensitivity equation [41]. Both these solvers use standard Gauss elimination methods. Effect of ordering of the unknowns on the memory requirement is shown in Ref. [10]. Notwithstanding the reduction brought about by such concepts, computer memory requirement of direct methods remains prohibitive for the large two-dimensional and even for smaller three-dimensional problems. This drawback has thwarted its application for the solution of sensitivity equations.

The chief advantage of iterative methods over direct methods is their relatively low memory requirement. Iterative methods possess a distinct advantage when a single large sparse system of linear equations is solved as its convergence is also rapid. However, for the linear systems with multiple right-hand-side vectors, iterative methods are less efficient vis-a-vis direct from a CPU standpoint. This is due to the fact that in the direct method, lower-upper decomposition of the coefficient matrix is performed once and stored, which is used for the forward-backward substitution for each of the multiple right-hand-side vectors. With iterative methods, the system of equations has to be solved for each right-hand-side vector.

Iterative methods can be divided into traditional relaxation methods and conjugate-gradient-based methods. Relaxation methods require diagonal dominance of the coefficient matrices, as also demonstrated in Ref. [126]. Application of higher order accurate upwind discretization of the spatial terms divests the coefficient matrix of the diagonal dominance. Besides this, the sensitivity equation, as presented in the form of Eqs. (3.5) or (3.11) and (3.12), does not permit any mitigating measures, such as the addition of time terms as it generates inaccurate sensitivity derivatives [38]. In the conjugate-gradient-based iterative methods, such as the generalized minimum residual methods (GMRES), the diagonal dominance of the coefficient matrix is not a requirement. However, this method requires the eigenvalues of the coefficient matrix to be clustered around unity and, hence, preconditioning of the coefficient matrix is

necessary for its success [36, 41, 95]. The memory requirement of a preconditioned conjugate gradient (PCG) algorithm is low as compared to a direct inversion solver. However, this memory requirement is high enough to preclude the realistic, high grid density design of a practical 3D geometry which involves the solution of very large system of sensitivity equations [95, 96].

Another approach is to transform the linear sensitivity equation in the delta form [35, 98] such that the Eq. (3.5) or (3.11) can be written respectively as:

$$\left[\frac{\partial \hat{R}}{\partial Q}\right]_D^n \{\Delta Q'\}^n = -\left[\frac{\partial R}{\partial Q}\right]_D^n \{Q'\}^n - \left\{\frac{\partial R}{\partial D}\right\}_Q \quad . \tag{3.16}$$

$$\left[\frac{\partial \hat{R}}{\partial Q}\right]_D^{T^n} \{\Delta \lambda_F\}^n = -\left[\frac{\partial R}{\partial Q}\right]_D^{T^n} \{\lambda_F\}^n - \left\{\frac{\partial F}{\partial Q}\right\}_D \quad . \tag{3.17}$$

Here, $\partial \hat{R}/\partial Q$ is an approximation to the true Jacobian that is used as a preconditioner. Note that the delta form of the adjoint equation for the constraints can be written similarly to Eq. (3.16). Then, these equations can be solved using the PCG approach. Another preconditioning approach is to use the approximate factors of the coefficients matrix, then break the problem into a sequence of simpler problems. This is the premise of the ADI (alternating-direction-implicit) method. The trade-off in the latter approach is a slower convergence for a much reduced computer memory requirement [98]. With the addition of a relaxation term for diagonal dominance, the ADI scheme for the delta form of the sensitivity Eq. (3.16) is written as,

$$\begin{aligned}
\left[\frac{I}{\omega} + \frac{\partial \hat{F}}{\partial Q}\right]_D^n \{\Delta Q'\}^* &= -\left[\frac{\partial R}{\partial Q}\right]_D^n \{Q'\}^n - \left\{\frac{\partial R}{\partial D}\right\}_Q \quad . \\
\left[\frac{I}{\omega} + \frac{\partial \hat{G}}{\partial Q}\right]_D^n \{\Delta Q'\}^{**} &= \left(\frac{I}{\omega}\right) \{\Delta Q'\}^* \quad . \\
\left[\frac{I}{\omega} + \frac{\partial \hat{H}}{\partial Q}\right]_D^n \{\Delta Q'\}^n &= \left(\frac{I}{\omega}\right) \{\Delta Q'\}^{**} \quad .
\end{aligned} \tag{3.18}$$

The solution is iteratively obtained as,

$$\{Q'\}^{n+1} = \{Q'\}^n + \{\Delta Q'\}^n \text{ where, } n = 1, 2, \ldots \tag{3.19}$$

The relaxation parameter, ω may be set to $\Delta\tau$, where τ is a time-like variable. Other conventional options for the relaxation parameters are described in Ref. [95]. For simplicity, the coefficient matrices on the left-hand sides in Eq. (3.18) may be constructed using the first-order-accurate upwind scheme, and the consistent linearization of the boundary conditions may be neglected.

3.3 Approximate Flow Analysis

In this method, flow solution is obtained in an approximate manner rather than via the usual CFD techniques. Recall that derivatives of the flowfield solution with respect to design variables are available in the direct method of sensitivity coefficients. These derivatives at a certain design point, along with the flowfield solution at the same point, are used in the Taylor series approximation of the flowfield at the perturbed values of the design variables [10, 11, 6]. This concept is also known as the flowfield prediction method. Denoting the available flowfield solution corresponding to the design variable vector, D^m, as $\tilde{Q}(D^m)$, the flowfield solution corresponding to the perturbed state of design variable vector, $D^m + \Delta D^m$, can be approximated as follows:

$$\tilde{Q}(D^m + \Delta D^m) = \tilde{Q}(D^m) + \sum_{i=1}^{NDV} \left(\frac{\partial \tilde{Q}(D^m)}{\partial D_i^m} \right) \Delta D_i^m + \text{h.o.t.} \ , \qquad (3.20)$$

where, $m = 0, 1, 2, \ldots$ and $D^{m+1} = D^m + \Delta D^m$. In this equation, superscript m is indicative of the possibility for the hierarchical flowfield approximation. All of these successive approximate flowfield solutions are computed from the base flowfield solution which is generated from a CFD technique ($m = 0$ and $\tilde{Q} = Q$). However, due to the truncation error associated with the omission of the higher order terms in the Taylor series expansion, this flow prediction method is certainly less accurate than the fully converged CFD analysis. The magnitude of the truncation error is proportional to the size of design variable perturbations. Hence, integrity of the predicted flowfield is contingent upon the selection of the appropriate size of design

variable perturbations. To this end, it is demonstrated in Ref. [11] that the accuracy of the flowfield solution is significantly compromised when a single large perturbation of geometry is used in the above approximation. Instead, successive approximation of the flowfield with the small incremental changes enroute the largely deformed final geometry, generates the corresponding flowfield solution with higher degree of fidelity.

In the optimization procedure, design evolution toward its optimum condition requires a number of evaluations of the objective and the constraints. For the aerodynamic design optimization, this requirement necessitates repetitive computation of the flowfield. Application of a CFD technique with a finer convergence tolerance criterion, greatly increases the computational cost of design optimization. In this context, judicious use of approximate analysis with its inherent computational efficiency presents a viable alternative [11, 141].

Chapter 4

ELEMENTS OF OPTIMIZATION PROCEDURE

The aerodynamic optimization determines the values for the vector of design variables, such that extremization of the objective function, in deference to the aerodynamic and geometric constraints, of the design, is obtained. For the aerodynamic shape optimization problems, design variables bear a direct relationship, with the aerodynamic surface geometry and govern its shape. Various methods have been devised to define this relationship and the process is known as surface parameterization. Therefore, the shape optimization procedure incorporates a systematic modification of these design variables and, hence, the surface shape is modified toward the optimum conditions.

Broadly, two categories of optimization methods are formulated, namely, unconstrained and constrained methods. In the unconstrained optimization, cognizance of the constraints can also be taken by redefining the objective functions, and a typical example is the penalty function method. The constrained optimization procedure followed in the present study, is discussed next.

4.1 Constrained Optimization Problems

Mathematically, design variables can be denoted by a vector in the following form:

$$D = \{D_1, D_2, D_3, \ldots, D_{NDV}\}^T \tag{4.1}$$

where, as mentioned in the preceding chapter, NDV is the total number of design variables. Expressed mathematically, a typical constrained aerodynamic shape optimization problem is:

minimize an objective function

$$F(D, Q(D)) \tag{4.2}$$

subject to

(i) flow-related inequality constraints

$$G_j(D, Q(D)) \leq 0, \qquad j = 1, \ldots, NCON_f \ . \tag{4.3}$$

(ii) geometric inequality constraints

$$G_j(D) \leq 0, \qquad j = NOCN_f + 1, \ldots, NCON \ . \tag{4.4}$$

(iii) side constraints

$$D_i^{\text{lower}} \leq D_i \leq D_i^{\text{upper}}, \qquad i = 1, \ldots, NDV \ . \tag{4.5}$$

In aerodynamic applications, typical examples of objective functions and flow-related constraints include lift, drag, lift-to-drag ratio, pitching moment or the root mean square of the difference between a target, and an actual pressure distribution over a surface. Both the objective function and the flow-related constraints depend on the flowfield solution and the design variables. Geometric constraints depend only on the design variables. Typical examples of such constraints may include thickness, sectional area, wing volume, leading edge radius, leading and trailing edge angles,

Side constraints are enforced directly on the design and serve to prescribe the lower and upper bounds on the design variables.

Inequality constraints in Eqs. (4.3)-(4.4) divide the design space into two domains [15]; the feasible domain where all the constraints are satisfied and the infeasible domain where at least one of the constraints is violated. This is schematically shown for the case of two design variables in Figure 4.1. Satisfied constraints can also be further classified as inactive constraints, which fulfill the inequality condition easily, and the active constraints, which just meet the inequality condition of Eqs. (4.3–4.4), that is $G_j(D, Q(D)) \simeq 0$. Thus, aerodynamic optimization involves scouting the NDV-dimensional design space in the domain which is the intersection of the feasible design space and the space delineated by the upper and lower bounds of the design variables.

In the present study, the method of feasible direction is used to seek optimum solution. This method is coded by Vanderplaats [142] in the Automated Design Synthesis (ADS) program, which is used almost as a blackbox in the present study. In this method, a search direction for the optimum solution is computed such that any move along this direction does not result in the immediate violation of constraints and at the same time, yields a reduction in the objective function. Therefore, this method seeks the feasible-usable search direction. Computation of the search direction involves the use of the gradients of the objective function and constraints, known as the sensitivity coefficients. Therefore, sensitivity analysis becomes an integral part of the the optimization procedure. After the computation of the search direction, next task is to find the step length by which a design variable is moved along this direction. This task is also known as the one-dimensional search task. This process of a one-dimensional search is continued as long as a reduction of the objective function is realized and the constraint boundary is not hit. Computation of the search direction and one-dimensional moves along this direction, mark a single iteration in

the optimization process. When a constraint boundary is hit, either a new iteration is started or, if the design is inside the feasible domain, a new direction based on an unconstrained minimization technique is used. During the one-dimensional search, repetitive evaluation of the objective function and constraints is required, and this computation necessitates the availability of an efficient and accurate flow analysis capability for the optimization process.

Thus, an optimization process begins with an initial guess on the starting point for the design variables, and the design is updated through an inner-outer loop iterative procedure by updating the design variables such that,

$$D_k^m = D^{m-1} + \alpha_k \vec{S}^m \tag{4.6}$$

The index m refers to the outer loop of the optimization where the search direction is computed, and index k refers to the inner loop of the optimization process where the one-dimensional search is performed to move the design along the search direction. Several unconstrained minimization techniques can be employed for the one-dimensional search. This includes, for example, the bracketing method, polynomial interpolation, and golden section method. These automated methods, commonly used for structural optimization problems, can ensure functional minimization in a smaller number of steps, by computing larger perturbations of design variables. However, in the present study, instead of using these automated methods for the one-dimensional search, a constant step size method of design variable perturbations is used, following Burgreen and Baysal [36]. Such a strategy can be beneficial for aerodynamic optimization problems governed mostly by nonlinear physics, where several evaluations of designs, generated by means of constant and smaller perturbations of the design variables, facilitate due resolution of such a nonlinear design regime. This strategy also reduces the CPU time spent for each flow analysis, due to the proximity of the flowfield solutions corresponding to any two neighboring designs. This feature is essential for the effective use of PCG-based flow analysis during optimization cycle [36].

For the case where a constant step size is used, the instantaneous values of the design variables in Eq. (4.6) is determined by,

$$\alpha_k = \alpha_{k-1} + \Delta\alpha \tag{4.7}$$

where, the size of the increment, $\Delta\alpha$, is left to the user's judgment and numerical experimentation.

4.2 Surface Parameterization

The search for an efficient and accurate means of aerodynamic surface representation has been a topic of intense research and almost parallels the efforts to enhance the efficiency of numerical optimization procedures. Optimization of existing aerodynamic shapes is facilitated by first defining this shape in terms of a set of control points. Therefore, these control points, also known as design variables, have governing influence on the generation of the surface shape. During the optimization process, these design variables are assigned discrete values to create several geometries for their performance evaluation. Therefore, design variables hold the key to the success of the optimization procedure, as their proper selection enables the optimization process to generate and evaluate a broad range of surface geometries, and allows it to converge quickly. Additionally, the ability to represent a surface with a minimum number of design variables lowers the computational memory and time requirements of an optimization procedure. These factors serve to highlight the cardinal role played by the surface parameterization techniques in the aerodynamic design optimization arena where computational efficiency is directly linked to its feasibility for practical application.

Samareh [143] has presented an excellent summary of the different parameterization techniques and categorized them into four approaches, namely, analytical, semi-analytical, discrete, and CAD representation. Analytical methods represent a simple approach toward parameterization. Typical examples of design variables

in this approach are wing sweep, thickness ratios, twist and camber and dihedral [96, 98, 89]. These design variables help generate smooth surface. Although in this approach, a few design variables are required, its primary disadvantage is the inability to generate a very wide variety of geometries and to optimize an existing free-form surface [143]. A somewhat similar approach is Ref. [73], where a symmetrical airfoil thickness t is analytically defined in terms of a polynomial whose coefficients are the design variables.

In the semi-analytical approach, various kinds of shape functions have been devised. Two broad categories of shape functions include analytical shape functions and shape functions of aerodynamic origin. Analytical shape functions are added linearly to the initial airfoil which is described by a set of points. Thus airfoil definition is characterized by the following form:

$$y_{\text{current}} = y_{\text{initial}} + \sum_{i=1}^{NDV} \delta_i f_i \tag{4.8}$$

where δ_i constitute the design variables and denote the weights of the various shape functions which characterize geometry perturbations. Several shape functions such as Hicks-Henne functions, Wagner functions and polynomial functions have been used for numerical optimization. Their mathematical expressions and characteristic are detailed in Ref. [75]. Wagner functions permit fairly large variations in the airfoil shape but are unsuitable for the substantial camber modification and produce a wavy geometry at higher harmonics. However, Hicks-Henne functions and polynomial functions have simpler form and allow substantial change of the camberline. Hicks-Henne functions combine two classes of functions and each affects a limited region of the profile, as illustrated in Ref. [82], which uses Hicks-Henne functions. Several examples of their use can be found in literature including Ref. [74]. These functions consist of two classes, namely polynomial and sine functions [86]:

$$f = \frac{x^n(1-x)}{e^m x} \tag{4.9}$$

$$f = \sin^m [\pi x^n] \tag{4.10}$$

Several shape functions have been developed and adapted to particular geometric specification from these general formulae by suitably choosing the values of m and n.

Another category of the shape functions evoke interest due to their physical meaning and aerodynamic origin. These very features help to reduce the number of design variables and computational requirements of the aerodynamic optimization procedure. This category can further be divided into airfoil library and aerofunctions.

The airfoil library concept has been formulated in Ref. [142] where the shape functions are generated using already existing airfoils. Here the difference between the ordinates of the airfoils could serve as a shape function. Thus, during the optimization procedure, a new airfoil shape could be generated from

$$\{y\}_{\text{current}} = \{y\}_{\text{initial}} + \sum_{i=1}^{NDV} \delta_i \left(\{y\}_{\text{initial}} - \{y\}_i\right) \tag{4.11}$$

where $\{y\}$ is a vector of airfoil ordinates, and a total of NDV existing airfoils are used in conjunction with an initial airfoil for the synthesis of a new airfoil shape. This approach can account for geometric constraints by selecting airfoils that adhere to these constraints individually [86].

The concept of using aerofunctions in the optimization procedure was propounded by Aidala et al. [144]. An aerofunction shape is a wing-section ordinate perturbation that has been defined by a 2-D inverse code for desired pressure characteristics. Because of their definition, each aerofunction is a geometric function which is directly associated with a well-defined aerodynamic effect. In an optimization procedure, aerofunction shapes are added to the baseline geometry by the scaling factors, δ_i in the Eq. (4.8), which then serve as design variables. Aerofunctions are generally perceived to be very effective as only a few of them are sufficient to modify the geometry. However, because their origin is linked to an inverse code, they are defined for a specific operating condition and their generalization is difficult [75].

Various shape functions mentioned thus far under the semi-analytical category generally require a few design variables and are computationally efficient. However, they can produce non-smooth optimized shapes. Also, they are difficult to generalize for a complex geometry such as an entire aircraft.

The use of splines, which are piecewise continuous polynomials, is another approach to alleviate problems of oscillations that are generally associated with global polynomials of higher order used to fit from one boundary point to another [145, pages 288–290]. Cubic splines are the most commonly used polynomials. Use of such patched polynomials consisting of cubic polynomial and parabola for the airfoil optimization may be found, for example, in Ref. [81].

Other possibilities in this category include use of special polynomials which are orthonormal functions. Hartwich and Agrawal [146] emphasize their importance by arguing that the shape functions above are non-orthogonal and hence can cover only a limited design space and fail to yield significantly better approximation of a target shape beyond a problem-dependent number. In their study, three orthonormal functions, namely, Legendre polynomials, Chebyshev polynomials and "No Name" type polynomials have been assessed from the efficiency and accuracy standpoint, for the geometry parameterizations of airfoils and wings. Their study concluded that Chebyshev-type polynomials offer the most effective, accurate and user-independent approach toward geometry representations in aerodynamic optimization.

Another polynomial, known as Bezier polynomial also offers a viable alternative for the shape parameterization. Burgreen et al. [94] mention that using this approach, a complex shape can be parameterized with relatively small number of Bezier control points, which also remains smooth. In Ref. [95], a three-dimensional surface, that of a unit wing, has been parameterized in the Bezier-Bernstein framework by a tensor product of two one-dimensional curves as shown below,

$$X_b(u,v) = \sum_{m=0}^{M} \sum_{n=0}^{N} g_m^M(u) \cdot g_n^N(v) \cdot P_{mn} \tag{4.12}$$

where $u, v \in [0, 1]$, P_{mn} represents the position vector of the $(M+1)(N+1)$ number of Bezier control points. Bernstein basis functions of degree M and N are denoted by $g_m^M(u)$, $g_n^N(v)$. A representative formula for $g_m^M(u)$ is given by,

$$g_m^M(u) = \binom{M}{m} u^i (1-u)^{M-m} \quad , \tag{4.13}$$

where

$$\binom{M}{m} = \frac{M!}{m!\,(M-m)!} \tag{4.14}$$

is a binomial coefficient. The Mth order polynomial is defined by $(M+1)$ Bezier control points. All the Bezier control points define the derivatives, order, and shape of the curve, but only the first and last points actually lie on the curve in each direction [60]. Examples of their use for the optimization procedures can be found in the works of Huddleston and Mastin [147], Birckelbaw [60], Greff et al. [62], Burgreen et al. [94], Burgreen and Baysal [36, 95], and Venkataraman [148].

The third category in the classification by Ref. [143] is the discrete approach where discrete representation of the geometry is used. Baseline geometry is discretized and position of each grid points assume the role of design variables. Baysal et al. [11], and Eleshaky and Baysal [41] have used such an approach for the optimization of two-dimensional surfaces, such as ramp shape of scramjet-afterbody and airfoil where surface is defined by grid points coordinates and the relative slopes. This approach permits a great degree of latitude in exploring various geometries for the optimum solution. However this latitude comes with a penalty of higher computational expense due to an increase in the number of design variables, and possibility of generating a nonsmooth shape.

The fourth approach for the shape parameterization is based on the variants of B-splines which are widely used in Computer Aided Design (CAD) tools. A popular subset of B-splines for the surface approximation in this category is Non-Uniform Rational B-Splines (NURBS) surface representation. This type of parameterization

can represent a wide range of geometries and assure its continuity and smoothness. However, this representation involves a large number of design variables [143]. Details on the NURBS surface representation can be found in Refs. [149, 150].

Another new method for representing complex surface geometry is a PDE method of Bloor and Wilson [151]. The method views surface generation as a boundary value problem and produces surfaces as the solutions to fourth-order elliptic partial differential equations.

Thomas et al. [149] have used NURBS- and PDE-based representation of aerospace vehicle surfaces for exploring their feasibility in the optimization process. This concept has been used in a methodology which defines a class of airplane configurations, and directly evaluates surface grids, volume grids and grid sensitivity [152]. This methodology has also been used [150] for the parameterization of the blended-wing-body type of unconventional configuration alongwith the NURBS approximation.

In the present study, the wing shape parameterization from Burgreen and Baysal [95] is adopted. The wing used in this study is generated from a geometrically simple wing, which is unswept, untwisted, uncambered and rectangular, with both its chord and span equal to unity; hence this wing is referred to as the unit wing. To generate a variety of shapes, the geometric parameterization of a wing should allow flexibility in its sections, taper distributions, sweep, span, spanwise bending, geometric twist, and global angle of attack. In the present parameterization, as shown in Figure 4.2, each feature is implemented as an independent geometric operation in a sequential manner. Its details can be found in Ref. [95].

4.3 Grid Adaptation

The present numerical optimization procedure essentially focuses on the evaluation of several surface geometries. Each surface geometry requires a flowfield

analysis. In order to perform consistent evaluation of several perturbed surface geometries, field grids also have to be regenerated corresponding to the each new wing shape and surface grid definition. Employing grid generation program for such purposes is not a viable option within an optimization procedure because of the constraints on the computational time and implementation constraints. Hence, this process of re-gridding enroute to the optimized shape is completed using the flexible grid approach of Burgreen and Baysal [95], where the volume grid is written as a function of the body surface grid (X_b). As the surface is changed during the shape optimization, the X-coordinates, for example Ref. [75], of the new volume grid are found using the following relations:

$$x_i^{\text{new}} = x_i^{\text{old}} + [1 - v_j]\left(x_b^{\text{new}} - x_b^{\text{old}}\right) \quad ,$$

where

$$\begin{aligned} v_j &= \frac{s_j - s_2}{s_{jmax} - s_2} \quad , \\ s_j &= \sum_{i=2}^{j} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \quad . \end{aligned} \tag{4.15}$$

Using this flexible grid approach and the surface parameterization described above, grid sensitivities can be derived analytically. This obviates the need to generate the same by the finite difference method. Expressions for the analytically generated grid sensitivities that are used in the present study can be found in Ref. [94].

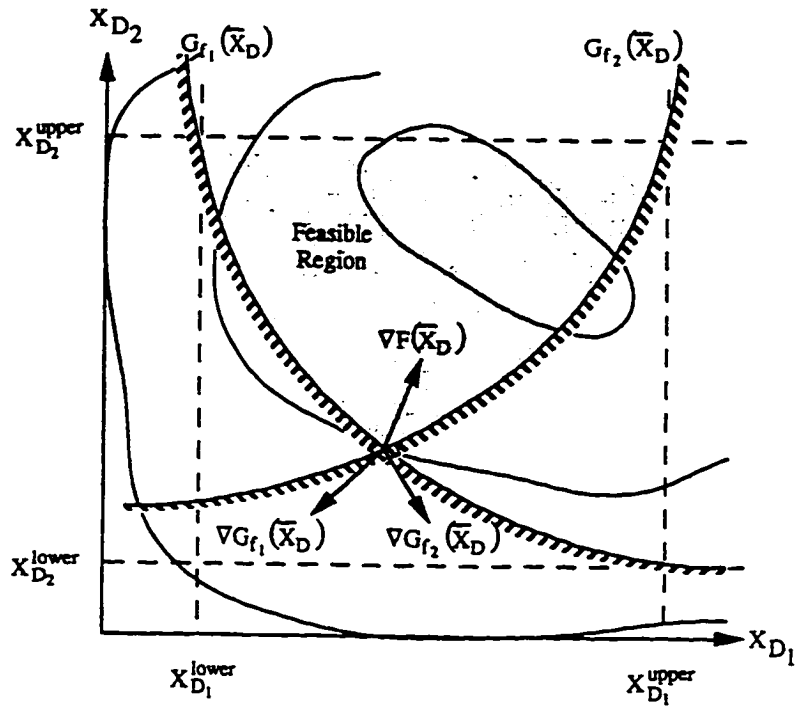


Figure 4.1 Usable-feasible design space (Ref. [92]).

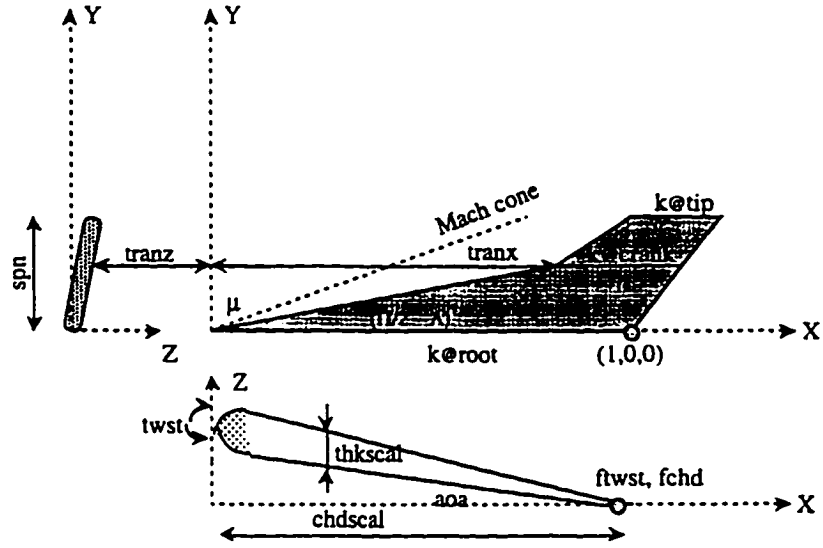


Figure 4.2 Wing Parameterization.

Chapter 5

RESULTS ON INVISCID OPTIMIZATION OF WING

From the previous discussions, it is evident that the present shape optimization methodology is an ensemble of CFD analysis, sensitivity analysis, shape parameterization, and a search/minimization procedure. Each of these elements is a crucial building block for the success of the optimization methodology. It should be recalled here that the objective of the present study has been to develop a three-dimensional design methodology that is automated and practical.

Practicality of any shape optimization method is directly related to its suitability for handling large-scale problems. One obvious origin of a large-scale problem is the necessity of solving a design optimization problem on a high grid density, to resolve the flow gradients adequately. Previous optimization studies [96, 95] have demonstrated a practical use of fully implicit methods (quasi-Newton method) within an optimization procedure. Memory requirement of such a procedure which uses a PCG algorithm is low compared to direct inversion solvers. However, this memory requirement has been found to be high enough to preclude the realistic, high grid-density design of a practical 3D geometry such as a wing or wing-body combination, although this procedure allows the realization of high convergence rates and consequent reduction of CPU times. This particular limitation has served as

the impetus to improve upon this recently developed 3D optimization methodology. This improvement has been achieved by the use of ADI-factored operators to serve as preconditioning matrices for the CFD as well as the sensitivity equations. All the computations of the present study (Chapters 5 and 6) have been performed on CRAY-YMP supercomputer at NASA Langley Research Center.

5.1 Inviscid Optimization For Large-Scale Problems

For this purpose, an arrow wing is considered to be at 3-deg angle-of-attack to an oncoming freestream flow of Mach 2.4. This wing is generated from the unit wing by linear distributions of `chdscal` and `thkscal` along the span as schematically shown in Figure 4.2. The wing's entire leading edge is subsonic. Geometry of such a wing has been described below (Table 5.1).

Selection of a particular wing is arbitrary, since the purpose of the present study is to evaluate the merits of the methodology, how it improves the shape within the definition of the problem objective and constraints, and doing so by using much less memory, which makes it a candidate for a conceivable large-scale problem. Therefore, initial or final shapes cannot be claimed to be realistic for practical application although the method that produces such shapes could be as such. To demonstrate the applicability of the method for the large-scale problems, a coarse grid (43×15×9) and a fine grid (127×43×25)[1] have been used.

Table 5.1 Geometry of initial arrow wing

Λ_{LE} and Λ_{TE} (deg)	Aspect ratio	$(t/c)_{root,mid,tip}$ (%)	θ_{tip} (deg)	Half $-$ span$/c_{root}$
72-56.3	1.01	4.80, 4.55, 4.80	6.80	0.57

[1] Dimensions in streamwise, normal and spanwise directions, respectively.

5.1.1 Validation of ADI Approach

Before embarking upon the optimization study, its major elements, namely, flow analysis and sensitivity analysis are validated by comparing their respective results with corresponding results from other codes or methods. The present ADI-based flow analysis results are compared with those from the PCG-based code. The coarse grid is used to accommodate memory-intensive PCG solutions within reasonable computer resources, as well as for the grid refinement studies.

The flowfield is computed on the coarse grid using the present ADI and PCG methods as well as CFL3D (Version 3.0) [125]. The results of these analyses are successfully compared in Figure 5.1 via their chordwise pressure coefficient distributions at three spanwise sections. The drag values of CFL3D and ADI (7.6e-03 vs 7.8e-03) match to the third digit and, for the ADI and PCG solutions, the lift and drag values match up to five significant digits (Table 5.2).

As necessary for quasi-Newton methods, the PCG method is initialized with the solution to a M_∞=2.35 flow, obtained using the ADI method. Then, the convergence to the $M_\infty = 2.4$ solution is timed for both of the methods (Table 5.2). As expected, by using ADI, the storage is reduced by a factor of about six compared to the PCG, even for this coarse grid. The initialization, apparently, is not in the domain of attraction to the root, hence, the convergence time of PCG is also unexpectedly higher.

In Figure 5.2, the chordwise pressure coefficient distributions, obtained on the coarse and fine grids with the ADI method, are presented for three sections. Although the coarse-grid flowfield appears plausible with its salient features, improvement in the solution due to grid refinement is clearly observed: with coarse-grid computations, the pressure peaks are overestimated in the root section, yet they are underestimated in the mid and tip sections. The surface pressures obtained on the fine grid (Figure 5.3) are deemed satisfactory based on the general flowfield features, such

as the crisper definition of the upper surface shock. A more conclusive evidence to an almost grid-independent solution, however, would require another level of finer grid solution which should differ only insignificantly, if at all, from the current fine-grid solution.

Next, the sensitivities are computed by the PCG and ADI methods using the direct differentiation (Eqs. (3.5) and (3.18)) on the coarse grid. The primary reason for the computations on the coarse grid is, naturally, the significant cost savings. Note that the point of this exercise is a fair comparison of the sensitivities from different methods, and not their absolute accuracy. Most of the values match to the fourth significant digit (Table 5.3). As for the efficiencies, this comparison essentially epitomizes the characteristics of both the methods. The unfactored solution procedure based on the PCG method, when in the domain of attraction to its root, is very efficient from a CPU-time standpoint. Whereas the factored solution procedure (ADI) can only achieve the linear convergence, it is very efficient from the memory viewpoint. The results shown in Table 5.3 are obtained using a convergence criterion of five orders of magnitude reduction in the L2-norm residual. As this much reduction may not always be necessary during an optimization, the effect of the order of magnitude on the efficiency and accuracy is investigated (Table 5.4). Each two orders of reduction increases the CPU time by a factor of about 1.6. Generally, the change in the values, when the residual is dropped from five orders to seven orders, is in the fifth or higher significant digit. Finally, ADI based sensitivities, obtained for the convergence criterion of three order of reduction, are compared with those by the finite difference method [10, 6] and they are found to be in excellent agreement (Table 5.5).

5.1.2 Shape Optimization

The primary thrust of the present study is to apply the gradient-based optimization technique for 3D shapes, which often require large grids. The challenge is to perform such a task within feasibility limits for computer memory, despite the fact that higher grid densities than what has been used previously [95, 96] are needed. As a demonstration, the present method is implemented to optimize a supersonic arrow wing configuration (Table 5.1 and Figure 5.3), first on a coarse grid then on a fine grid. The present problem formulation has nine design variables (`spn`, and at mid and tip sections: `thkscal`, `chdscal`, `tranx` and `twst`). The objective function is selected to be the maximization of C_L/C_D, subject to two aerodynamic and fourteen geometric and inequality constraints:

$$C_L \geq 0.11, \quad \text{and} \quad C_D \leq 0.008 \tag{5.1}$$

$$V_{\text{wing}} \geq 0.9 V_{\text{wing}}^{\text{initial}}, \qquad A_{\text{midspan}} \geq 0.6 A_{\text{midspan}}^{\text{initial}} \tag{5.2}$$

and at the root, mid, and tip sections:

$$2^\circ \leq \theta_{0.90\text{chord}} \leq 20^\circ, \qquad 2^\circ \leq \theta_{0.98\text{chord}} \leq 20^\circ \tag{5.3}$$

The volume constraint prevents the wing from becoming too thin whereas the angle constraint ensures that the trailing edge does not become too sharp or too blunt. Also, side (equality) constraints are imposed on the design variables; for example, `spn` is tightly bounded to prevent the optimizer from seeking higher lift by simply increasing the wing span. The selected values of the design variables allow for the formation of a planform break in the chord distribution. Also, the thickness, sweep and geometrical twist are allowed to have linear spanwise distributions. The airfoil section at the wing root is elected to remain unchanged. Present optimization studies use full CFD analyses for flowfield computations prior to the search direction

evaluations as well as for the one-dimensional searches. During the one-dimensional searches, the wing geometry is perturbed by $\Delta\alpha = 0.01$ (Eqs. 4.6 and 4.7) and successive flowfield analyses are performed with a convergence tolerance $\vartheta(4)$. For coarse as well as fine grid optimization cases, C_L, θ_{tip} at 90 % chord, and V constraints are violated in the final design. Additionally, for the fine grid optimization, $A_{midspan}$ constraint is also violated. The optimization results in a cranked arrow wing (Figures 5.4 and 5.5). The primary change is observed in the enlarged tip chord, the planform break, reduced thickness and the increased geometric twist. The present optimization problem formulation is such that the optimizer is lift driven (violated lift constraint and inactive drag constraint) and hence, it concentrates on lift improvement, by increasing planform area of the outboard wing (increased tip chord) and by increasing geometric twist. To compensate for the accompanying drag rise, it also reduces the wing thickness-to-chord ratios at the wing outboard. A summary of the optimized wing's geometry and aerodynamics is given in Table 5.6 (for initial wing, see Tables 5.1 and 5.2). The surface pressures of the optimized wing are presented in Figure 5.6, which may be contrasted with Figure 5.3. The objective function histories during the evolution to the optimized shape are presented in Figure 5.7. The trends appear similar for the coarse and the fine grid cases, but the values are distinctly different and the coarse grid shape converged in fewer iterations. Table 5.7 presents some of the statistics for these optimization cases. Finally, optimized wings from coarse and fine grid optimization are evaluated via the fine grid Euler-based flowfield analysis. This approach provides a common measure of merit for these optimization cases and serves to illustrate the trade-off between computational efficiency and performance gain. A summary of these results is presented in Table 5.8. It should be noted from these results that although fine grid optimization requires substantially higher CPU time, it yields aerodynamically more efficient design, as compared to coarse grid optimization. It should be highlighted that the PCG method, had it been used for this fine-grid shape optimization, would require 164 MW memory.

5.2 Summary of Results

In this chapter, ADI method's capability to accommodate inherent larger size optimization problems has been demonstrated, by optimizing a supersonic arrow wing for an inviscid flow condition, with coarse and fine grids. Fluid dynamic analysis using ADI and PCG methods and using the CFL3D code compare well. Sensitivity analysis using these two methods and the finite difference method also compare well. Hence, present computations have served to demonstrate successfully the feasibility and accuracy of PCG and ADI methods, for the high supersonic flowfield analysis and sensitivity analysis. These computations also serve to quantify the numerical efficiency characteristics (run time memory and CPU time requirements) of PCG and ADI methods. ADI method reduces memory by a factor of five but generally increases CPU time by a factor of seven, as compared to PCG method. Optimization results indicate that the design procedure explores the avenues of performance improvement within the permitted design space, following the conventional aerodynamic concepts. These results indicate superior aerodynamic performance of the fine grid optimization, albeit with a high CPU time and run time memory requirement. These results also underscore the need to introduce synergistic application of these two methods in solving a practical optimization problem.

Based upon the present computations, it is recommended that to better utilize computational resources, a number of coarse grid cases, using the PCG method, should initially be explored to improve optimization problem definition, design space and initial shape. The optimized shapes should be analyzed using a high fidelity (fine grid resolution) flow analysis to evaluate their true performance potential. Subsequently, a fine grid shape optimization should be conducted, using ADI method, to obtain the final optimized shape accurately.

In retrospect of these computations, it is felt that to minimize numerical noise in the evaluation of various designs during the one-dimensional search, the

convergence tolerance for the flowfield analysis should be decreased to $\vartheta(5)$. The stricter convergence tolerance would entail higher CPU time requirement, especially for the ADI-based optimization procedure. However, this increase can be partially compensated, if during the one-dimensional searches, the design is perturbed in larger steps ($\Delta\alpha = 0.02$). This may reduce the required number of flow analyses during one-dimensional searches. However, this strategy also makes the design space exploration somewhat coarser and therefore, its full implication on the aerodynamic performance of the optimized design should be investigated. Finally, in order to precisely investigate the role of grid density on the optimization results, present computations should be performed with three levels of grid. In order to resolve the computer memory bottlenecks associated with large-scale optimization problems, the present code should be parallelized in a distributed memory environment or SADD (Sensitivity Analysis on Decomposed Domains) scheme should be incorporated in the present procedure.

Table 5.2 Efficiency and accuracy comparisons for CFD analyses*

	PCG	ADI	
	Coarse Grid (43×15×9)	Coarse Grid (43×15×9)	Fine Grid (127×43×25)
Iterations to convergence	100	241	327
CPU time (sec)	640**	283	8138
Memory (MW)	10.61	2.25	42.11
C_L	8.4815e-02	8.4814e-02	8.7349e-02
C_D	7.8270e-03	7.8270e-03	7.6629e-03
(C_L/C_D)	10.8362	10.8362	11.40

*Computed flowfield for $M_\infty = 2.4$.
**Initialized with converged ADI solution of $M_\infty = 2.35$ flow.

Table 5.3 Comparison of quasi-analytical sensitivities*

	ADI	PCG
Iterations to convergence	112	9
CPU time (sec)	344	45
Memory (MW)	2.60	10.61
$\partial C_L/\partial(\texttt{chdscal}_{\texttt{tip}})$	0.0217	0.0217
$\partial C_L/\partial(\texttt{tranx}_{\texttt{mid}})$**	-0.0142	-0.0142
$\partial C_L/\partial(\texttt{spn})$	0.0439	0.0439
$\partial(C_L/C_D)/\partial(\texttt{tranx}_{\texttt{mid}})$	2.5124	2.5120
$\partial(C_L/C_D)/\partial(\texttt{tranx}_{\texttt{tip}})$	2.7549	2.7549
$\partial(C_L/C_D)/\partial(\texttt{spn})$	-10.4890	-10.4881

*Computed flowfield for $M_\infty = 2.4$ on 43×15×9 C-H grid; direct method with $\vartheta(5)$ convergence.

**Parameters as defined in Figure 4.2.

Table 5.4 Effect of convergence tolerance on efficiency and accuracy of sensitivities for ADI method*.

	$\vartheta(3)$	$\vartheta(5)$	$\vartheta(7)$
Iterations to convergence	65	112	163
CPU time (sec)	204	344	498
$\partial C_L/\partial(\texttt{chdscal}_{\texttt{tip}})$	0.0217	0.0217	0.0217
$\partial C_L/\partial(\texttt{tranx}_{\texttt{mid}})$	-0.0142	-0.0142	-0.0142
$\partial C_L/\partial(\texttt{spn})$	0.0439	0.0439	0.0439
$\partial(C_L/C_D)/\partial(\texttt{chdscal}_{\texttt{mid}})$	7.0183	7.0186	7.0186
$\partial(C_L/C_D)/\partial(\texttt{tranx}_{\texttt{mid}})$	2.5128	2.5124	2.5124
$\partial(C_L/C_D)/\partial(\texttt{tranx}_{\texttt{tip}})$	2.7547	2.7549	2.7549
$\partial(C_L/C_D)/\partial(\texttt{spn})$	-10.4889	-10.4890	-10.4890

*Computed flowfield for $M_\infty = 2.4$ on $43\times15\times9$ C-H grid; direct method.

Table 5.5 Comparison of ADI (quasi-analytical) and finite difference sensitivities*

	Quasi Analytical**	Finite Difference	Quasi Analytical / Finite Difference
$\partial C_L/\partial(\texttt{chdscal}_{\texttt{tip}})$	0.0217	0.0216	1.0055
$\partial C_L/\partial(\texttt{twst}_{\texttt{mid}})$	0.0124	0.0124	0.9993
$\partial C_D/\partial(\texttt{thkscal}_{\texttt{tip}})$	0.0007	0.0007	0.9956
$\partial C_D/\partial(\texttt{tranx}_{\texttt{mid}})$	-0.0031	-0.0031	1.0006
$\partial C_D/\partial(\texttt{twst}_{\texttt{mid}})$	0.0014	0.0014	1.0005
$\partial(C_L/C_D)/\partial(\texttt{chdscal}_{\texttt{tip}})$	2.3543	2.3571	0.9988
$\partial(C_L/C_D)/\partial(\texttt{twst}_{\texttt{mid}})$	-0.4108	-0.4091	1.0065
$\partial(C_L/C_D)/\partial(\texttt{spn})$	-10.4889	-10.4829	1.0006

*Computed flowfield for $M_\infty = 2.4$ on $43\times15\times9$ C-H grid.
**Direct method with $\vartheta(3)$ convergence.

Table 5.6 Aerodynamics and geometry of optimized wing

	Coarse Grid (43×15×9)	Fine Grid (127×43×25)
C_L	9.5675e-02	9.9113e-02
C_D	7.8516e-03	7.4940e-03
(C_L/C_D)	12.1855	13.2257
Λ_{LE} (deg)	71.9-67.0	71.9-67.6
Aspect Ratio	0.8701	0.9214
$(t/c)_{root,mid,tip}$ (%)	4.71, 3.07, 1.44	4.71, 2.32, 1.51
$(\texttt{twst})_{root,mid,tip}$ (deg)	0.00, 0.50, 0.12	0.00, 0.50, 0.24
$\%A_{mid}^{initial}$	68.7	52.0
$\%V_{wing}^{initial}$	88.6	80.4

Table 5.7 Statistics from optimization cases*

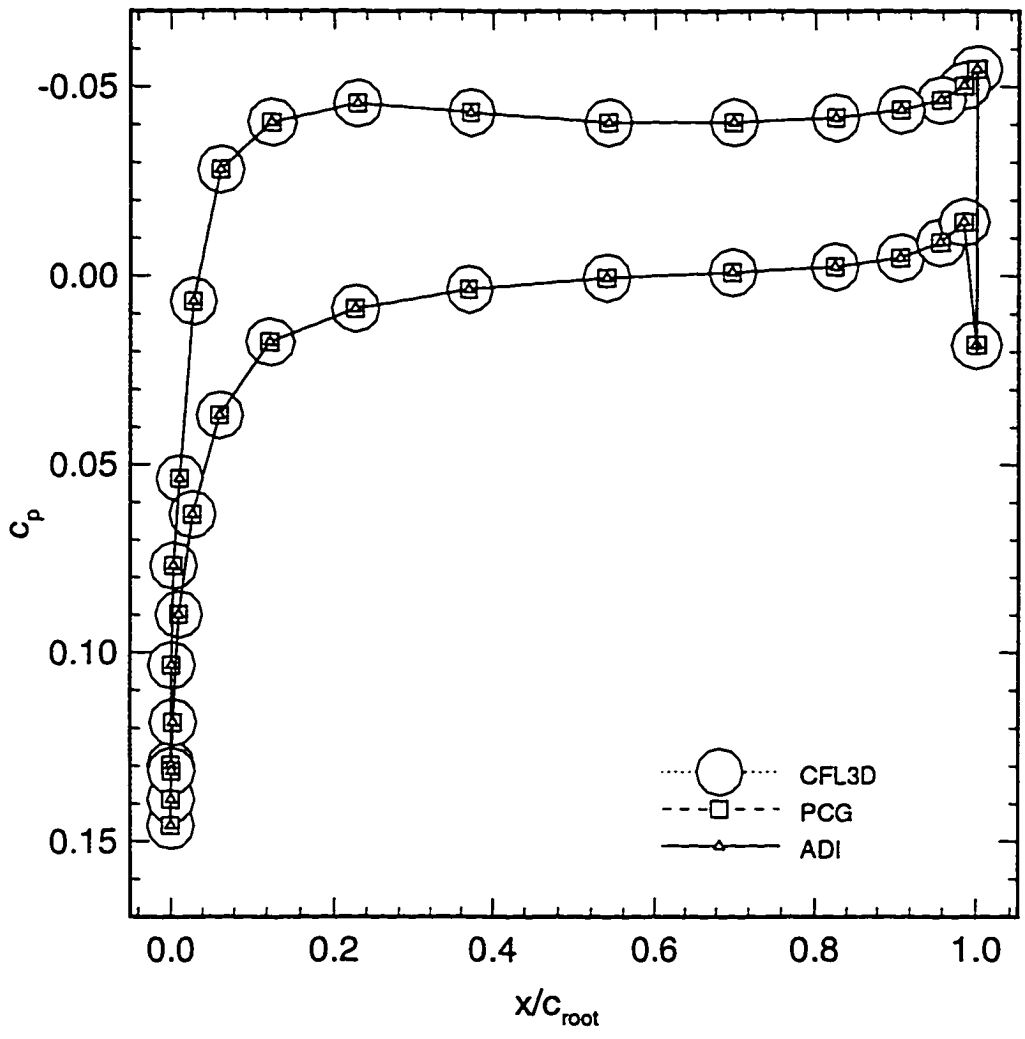
	Coarse Grid (43×15×9)	Fine Grid (127×43×25)
Lift change (%)	12.8	13.5
Drag change (%)	-1.26	-2.22
(C_L/C_D) change (%)	16.1	12.5
1-D searches	69	88
Gradient evaluations	5	9
Memory (MW)	2.93	48.37
CPU time (hr)	0.53	38.1

*Direct sensitivity method with $\vartheta(3)$ convergence.

Table 5.8 Aerodynamic performance and efficiency comparisons of optimized wings*

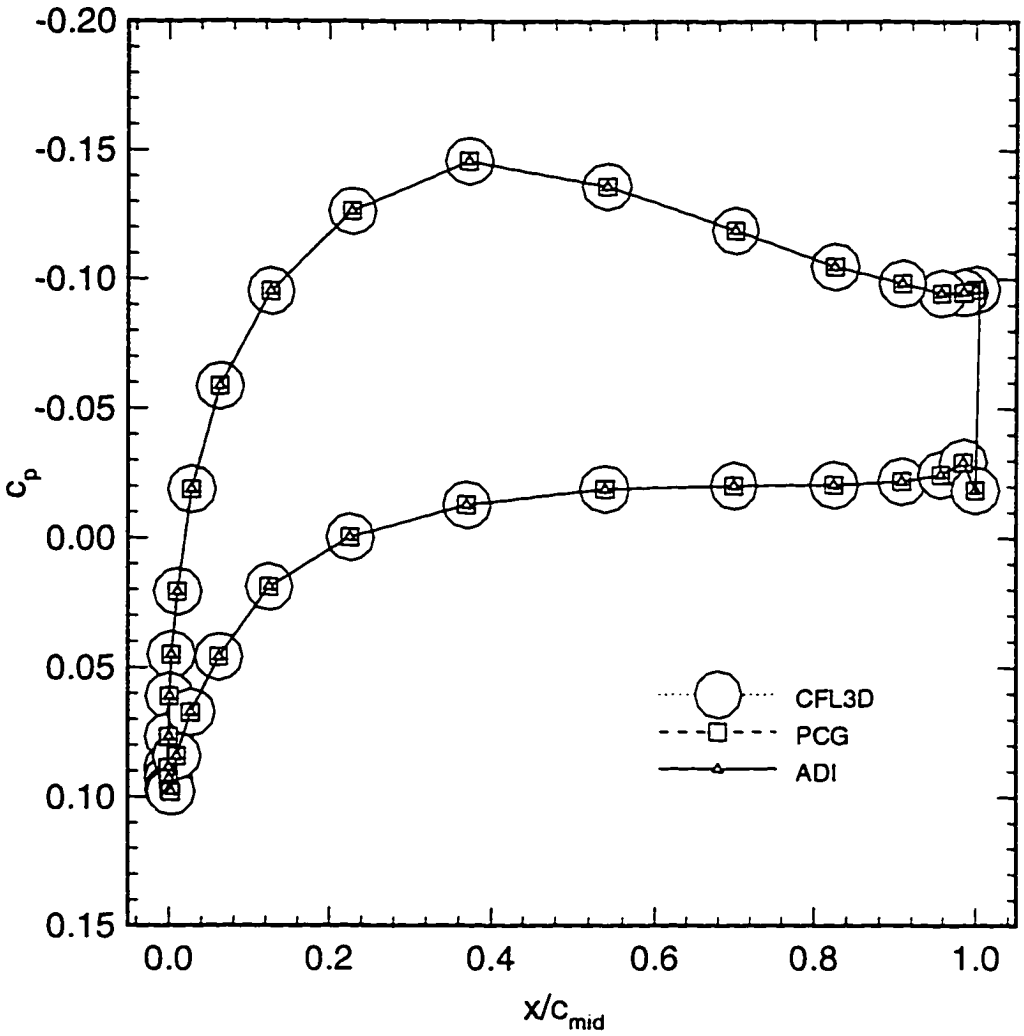
	Coarse Grid (43×15×9)	Fine Grid (127×43×25)
C_L	9.7674e-02	9.1113e-02
C_D	7.7455e-03	7.4940e-03
(C_L/C_D)	12.6104	13.2257
CPU time** (hr)	2.79	38.1

*Euler flowfield analysis for $M_\infty = 2.4$ on fine grid (127×43×25 points).
**Includes respective optimization and current flowfield.



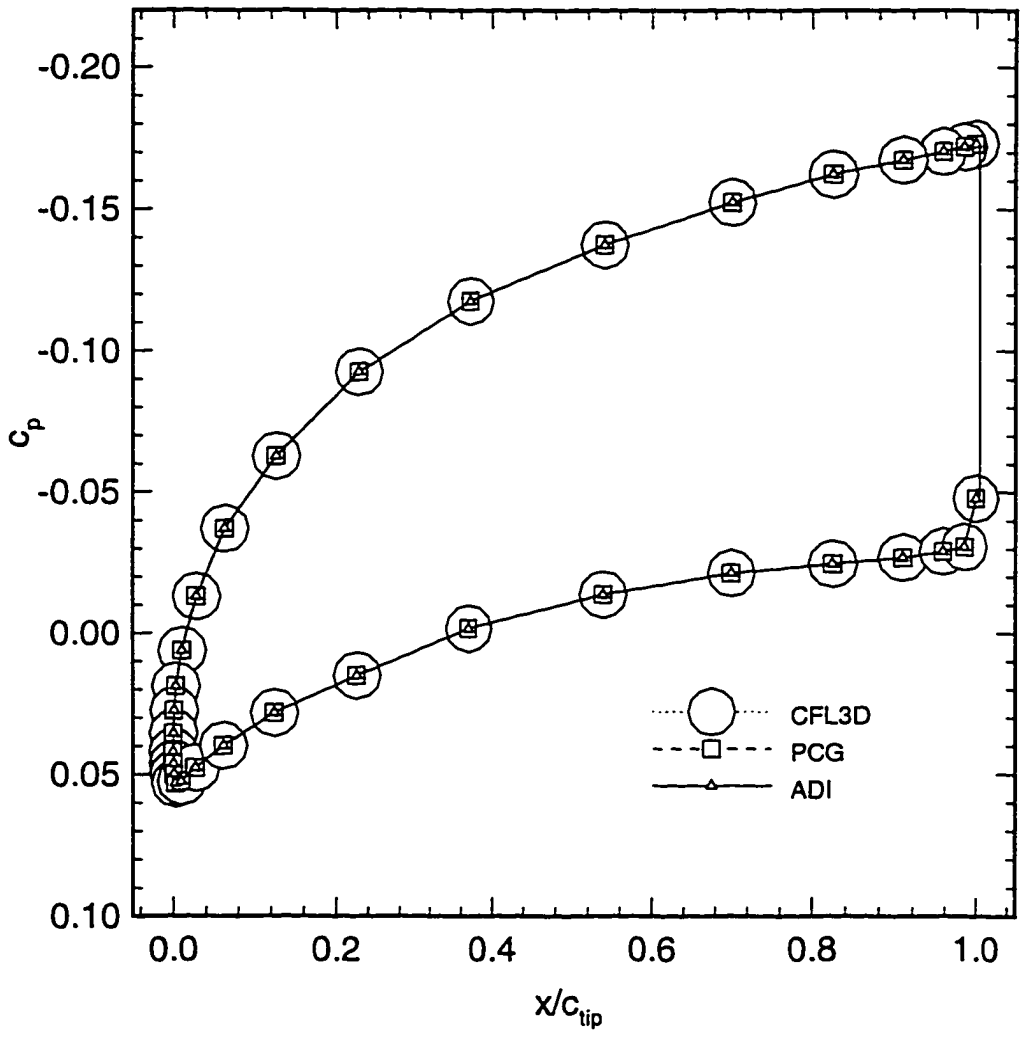
Root Section

Figure 5.1 Comparison of chordwise pressure coefficients on coarse grid (43×15×9). M=2.4, aoa=3-deg.



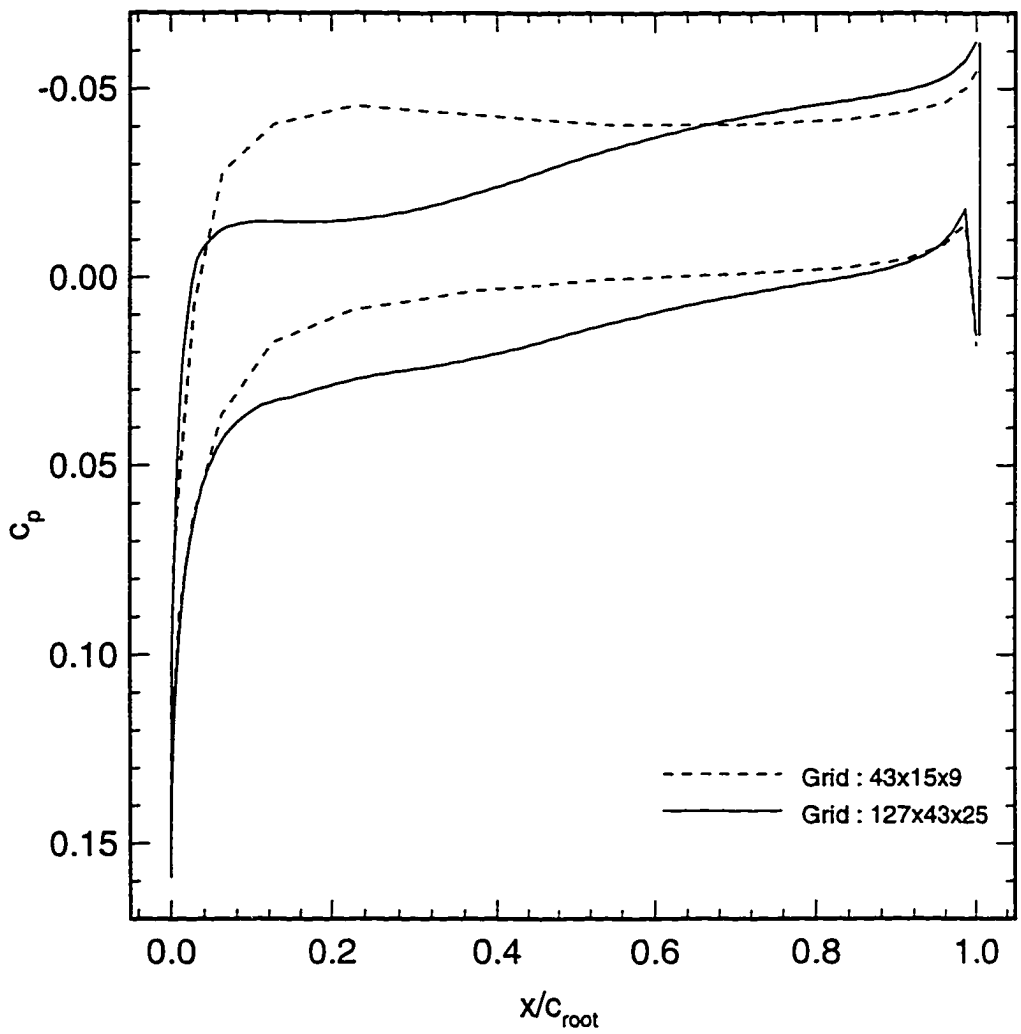
Mid Section

Figure 5.1 Continued.



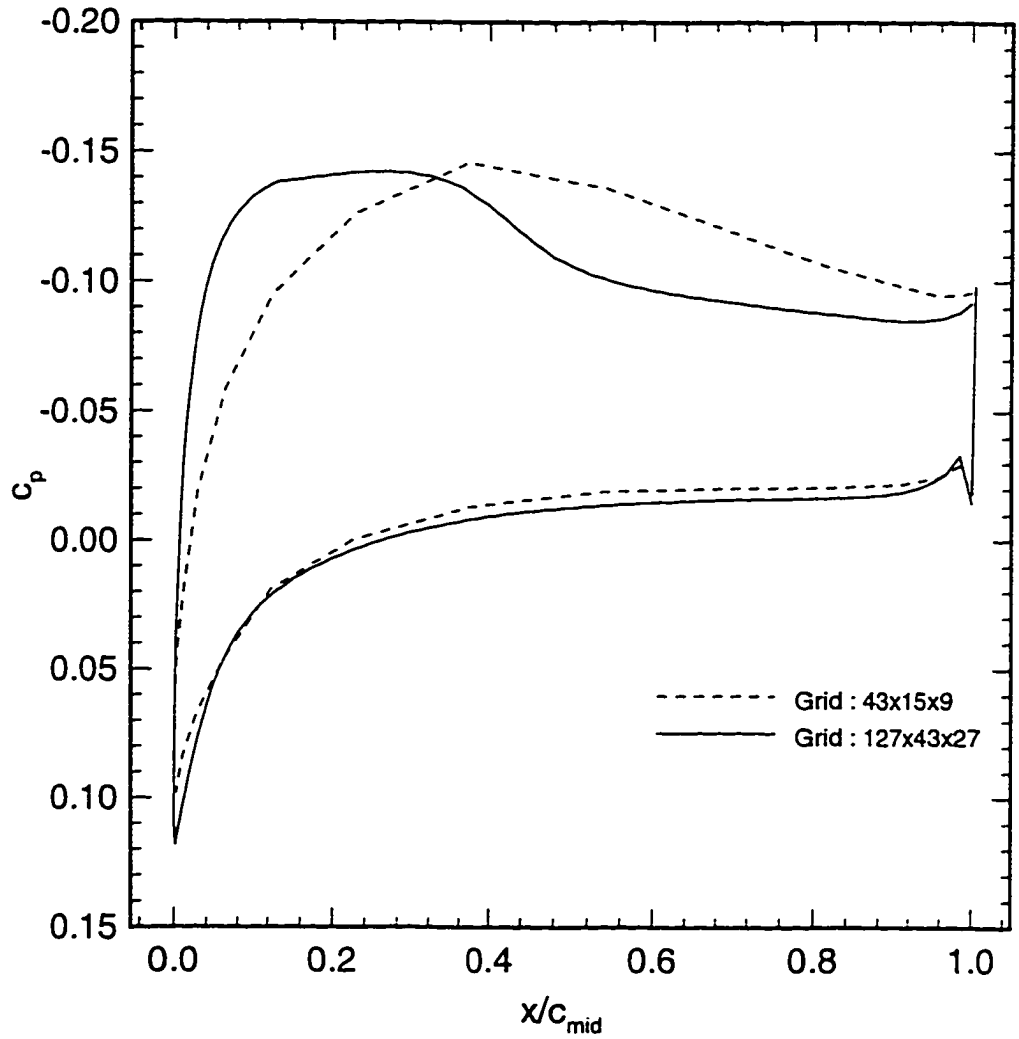
Tip Section

Figure 5.1 Concluded.



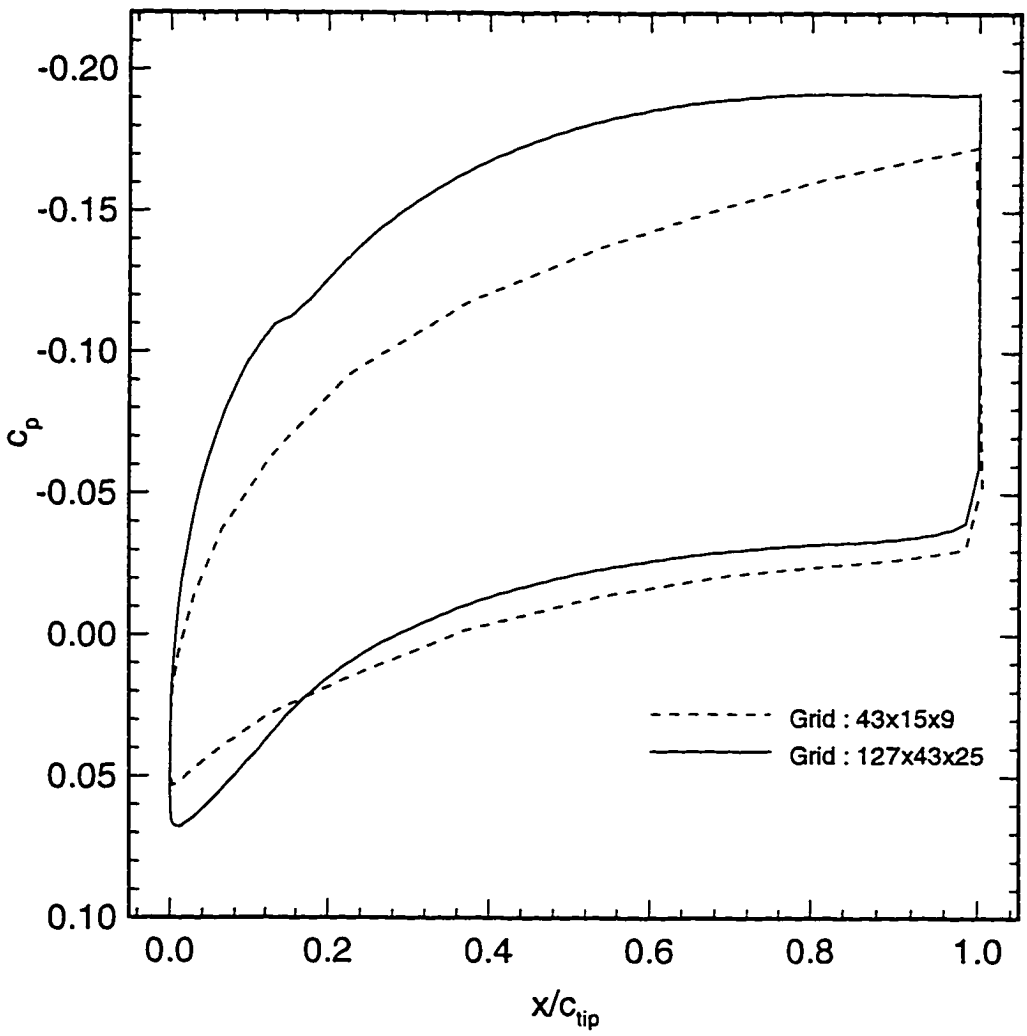
Root Section

Figure 5.2 Comparison of chordwise pressure coefficient distributions by ADI method on fine and coarse grids. M=2.4, aoa=3-deg.



Mid Section

Figure 5.2 Continued.



Tip Section

Figure 5.2 Concluded.

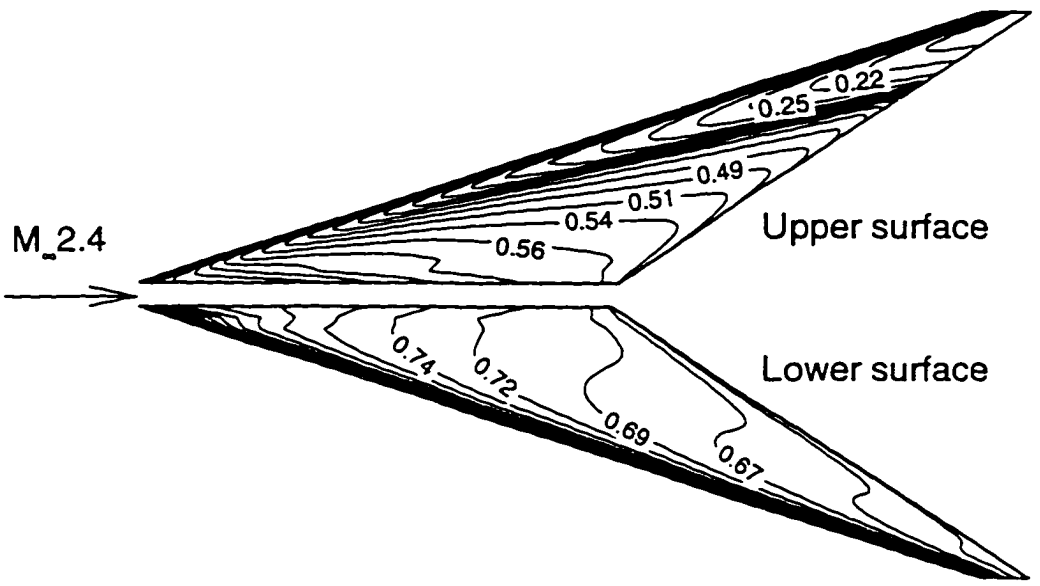
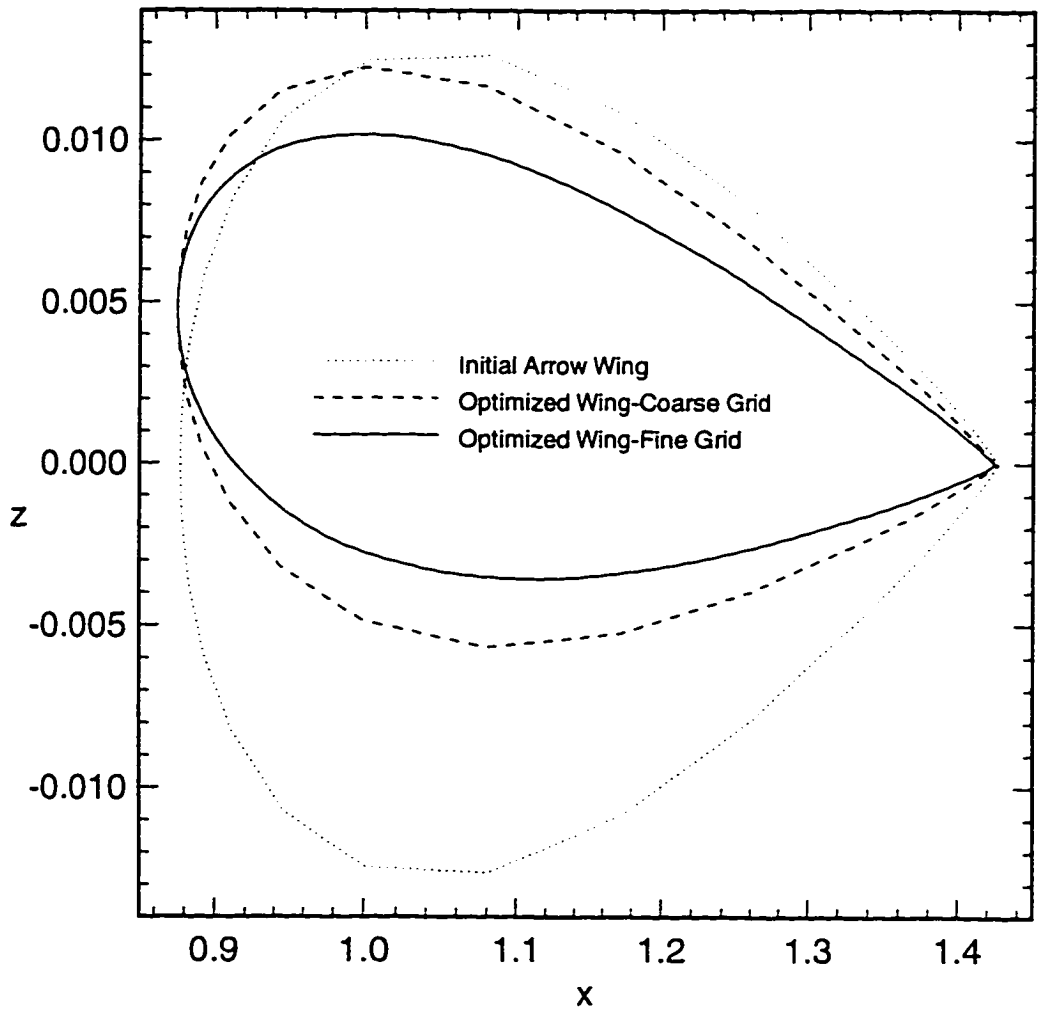
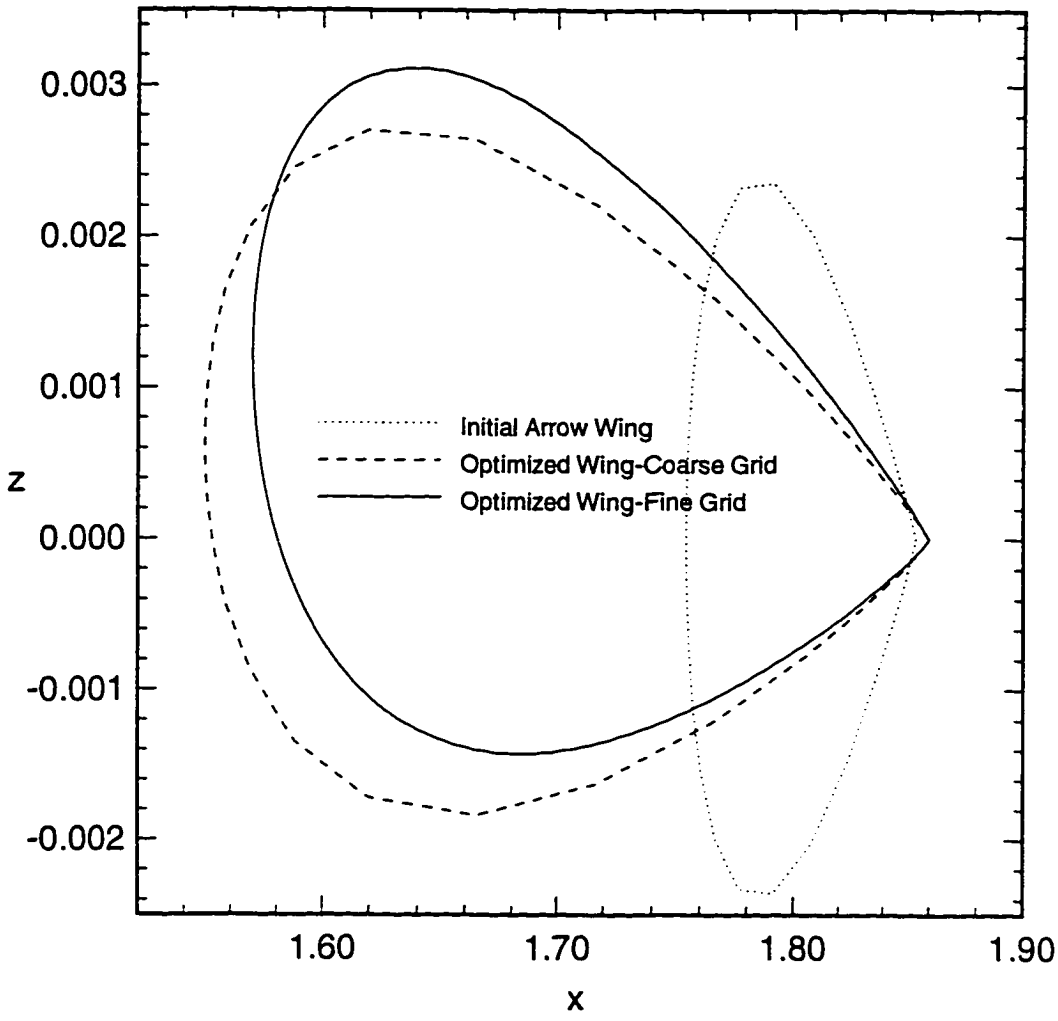


Figure 5.3 Normalized pressure contours by ADI method on fine grid (127×43×25). M=2.4, aoa=3-deg.



Mid Section

Figure 5.4 Evolution of wing sections from initial to coarse- and fine-grid optimized shapes.



Tip Section

Figure 5.4 Concluded.

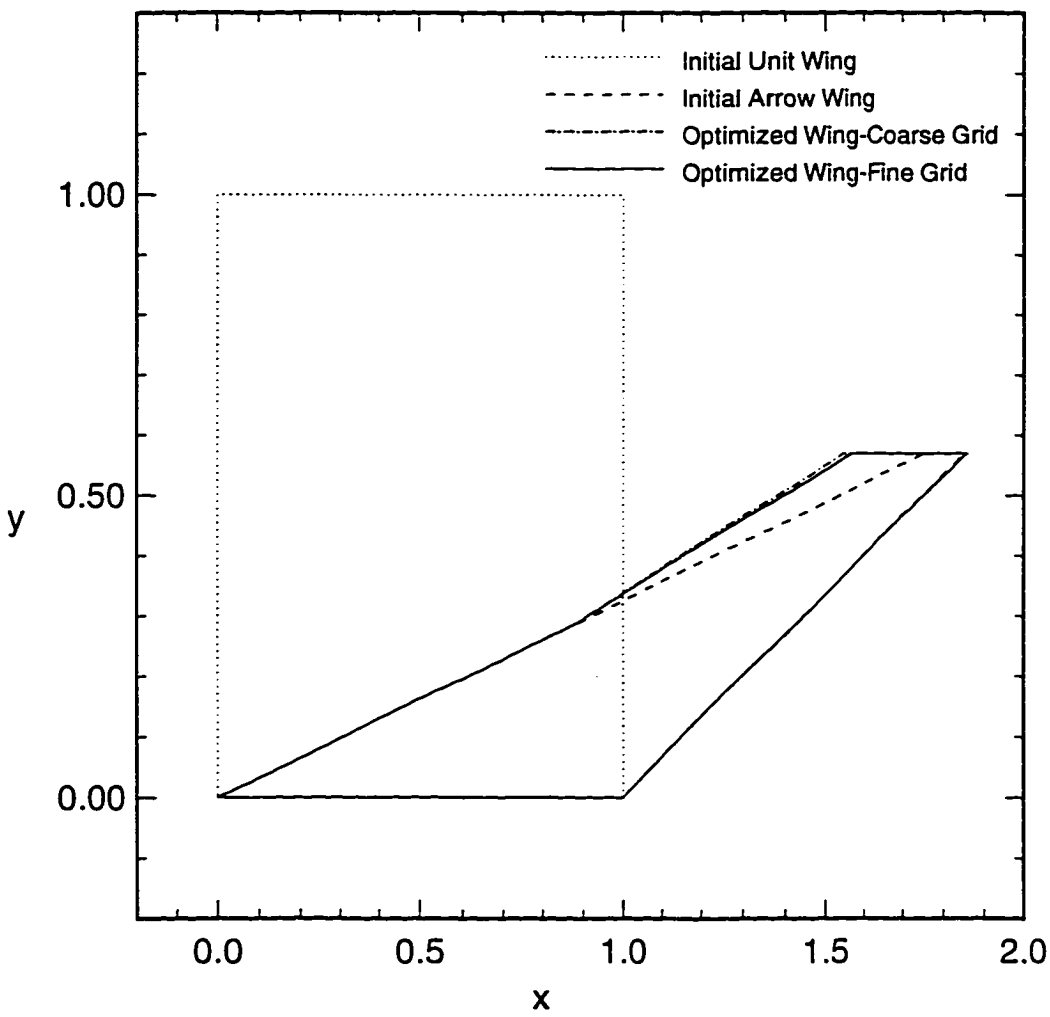


Figure 5.5 Evolution of wing planform shape.

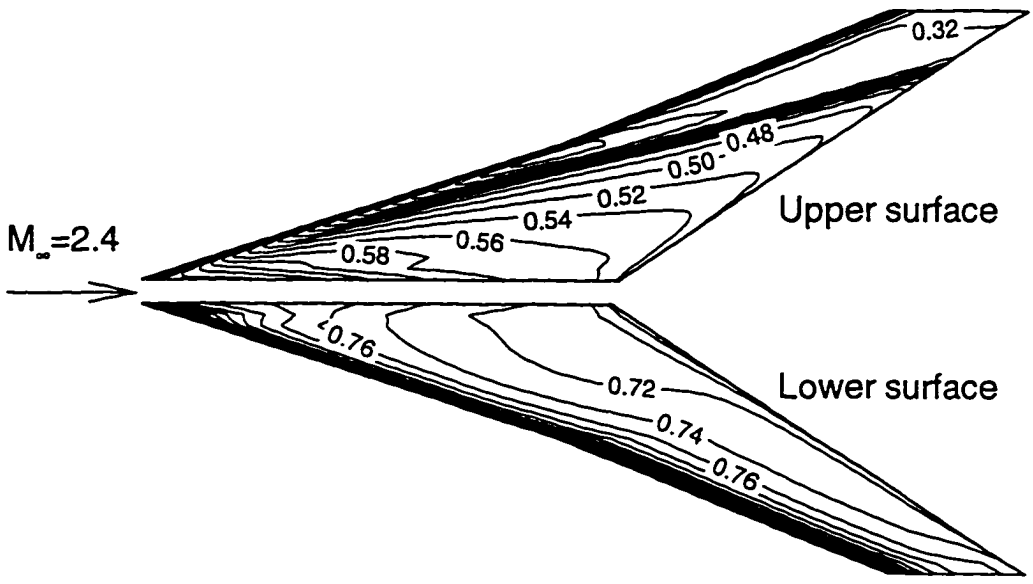


Figure 5.6 Normalized pressure contours on optimized arrow wing obtained by ADI method on fine grid (127×43×25).

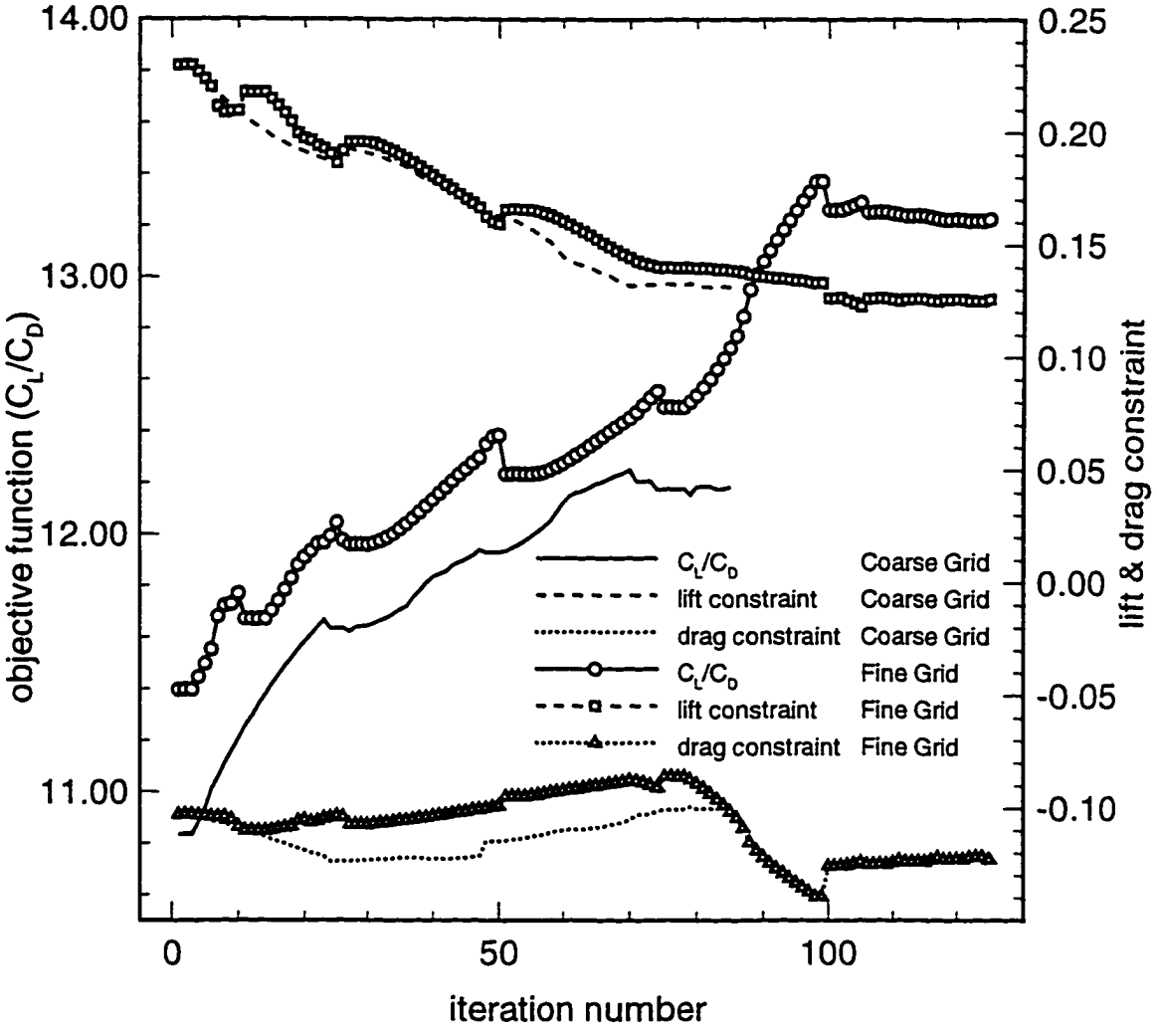


Figure 5.7 Histories of objective function and aerodynamic constraints.

Chapter 6

RESULTS ON VISCOUS OPTIMIZATION OF WING

In the previous chapter, the feasibility of the present optimization methodology for large-scale problems has been demonstrated by optimizing a wing using an inviscid flow assumption. Another important aspect of any practical optimization methodology is its capability to capture the flow physics in its full details. Most of the 3D aerodynamic optimization procedures reported thus far are based on the Euler equations. Rationale for considering only inviscid flow physics is that viscosity effects do not substantially influence overall flow over the wings at normal cruise lift conditions. However, viscosity plays predominant role in the leading edge separation and vortex formation, which substantially alter the aerodynamics of highly swept wings, as mentioned in the literature review of swept wing aerodynamics (Subsection 1.1.3). Therefore, the present shape optimization methodology has been extended for the viscous flows by using the Thin-Layer Navier-Stokes (TLNS) equations. However, application of the same for optimization problems is an expensive proposition. Hence, along with the development of a viscous shape optimization methodology, it is imperative to study strategies to employ this capability, such as ways in which inviscid and viscous shaping may be hybridized.

In order to evaluate the viability of this extended gradient-based optimization method, and explore the ways to employ it wittingly, several problems related to wing optimization have been formulated and solved. Some studies are also aimed at formulating certain algorithmic strategies with regard to the specification of initial constraints conditions, for the optimization problems. Results of these studies are presented next.

6.1 Viscous Optimization and Its Impact on Optimized Shapes

In order to study the effect of viscous flow physics on the optimized shapes and aerodynamics, all the optimization studies in the present section are conducted first by performing viscous optimization, followed by inviscid optimization on the same grid. Geometry of the initial arrow wing (Table 5.1) is again retained for the present computations, however to reflect the HSCT-type wing configuration at the cruise conditions, the present wing is considered to be at a 4.5-degree angle-of-attack to an oncoming freestream flow of Mach 2.4. To retain the validity of the laminar flow assumption for the current viscous optimization, the freestream Reynolds number is assumed to be the relatively low value of 2.0×10^5. To resolve and capture the boundary layer and leading edge flow patterns adequately, these studies also involve the use of two levels of grids, designated as coarse grid $43\times15\times9$ and fine grid $85\times29\times17$,[1] as in the studies of Chapter 5. However, fine grid density for all the studies in the present chapter is rearranged as compared to its counterpart in the previous studies for computational efficiency (Figure 6.1). For the fine grid case, grid distribution in the direction normal to the planform is made to resolve the boundary layer with about ten points.

[1] Dimensions in streamwise, normal and spanwise directions, respectively.

6.1.1 Effect and Accuracy of Viscosity on Flow Analysis and Sensitivity Analysis

The flowfield on the initial wing is computed by fine grid analyses performed using the presently developed viscous code and CFL3D (version 3.0) [125]. Results of these analyses show excellent agreement as manifested in the comparison of the chordwise pressure coefficient distributions at three spanwise sections the of initial wing (Figure 6.2). For this case, the present code predicts the lift coefficient to be 0.0189 and lift-to-drag ratio to be 6.068, whereas the corresponding values predicted by CFL3D are 0.0187 and 6.055, respectively.

These comparisons serve to demonstrate the present code's capability in accurate representation of the flow physics and its feasibility for the optimization studies. Subsequently, this code is used to compute the flowfield on the initial wing corresponding to the Euler- and TLNS-based flow analyses on coarse and fine grids. Surface pressure coefficient distributions corresponding to these computations are then contrasted for the three sections of this wing (Figure 6.3).

It is evident that fine-grid computations facilitate crisper definition of the suction peaks and slightly increased levels of the lower surface pressures. In addition, upper surface pressure variations are significantly different for the outboard portion of the wing. It is also observed that fine grid-computations, in contrast to coarse-grid computations, reveal a marked difference in the pressure distributions between the Euler and TLNS cases (Figure 6.3). It is concluded that fine-grid computations facilitate adequate resolution of the boundary layer, whereas coarse-grid computations fail to resolve the boundary layer sufficiently. Convergence patterns, predicted aerodynamic coefficients, and computational resource requirements of flow analyses for various cases are summarized in Table 6.1.

Subsequently, sensitivities are computed corresponding to the Euler-and TLNS-based coarse- and fine-grid flow analyses (Table 6.2). Direct differentiation

method (Eq. (3.18)) is used with the convergence criterion of 2.5 orders of magnitude reduction in the residual. Before the selection of this convergence criterion, effects of a stricter convergence criterion on the gradients have been studied, and importance of the accurate and efficient computational procedure of the same on the overall performance of the optimization methodology, has been considered. Hence, the choice made is deemed to be appropriate.

It is observed that viscous fine-grid sensitivities require a greater number of iterations to converge and higher CPU time. Also, computed sensitivities are quite different from the corresponding sensitivities of viscous coarse-grid, Euler coarse-grid and Euler fine-grid cases. This difference also highlights the fact that viscous fine-grid computations resolve the boundary layer adequately as opposed to viscous coarse-grid computations.

6.1.2 Shape Optimization

Prior to performing detailed optimization studies, certain precursor optimization studies are conducted. These studies are aimed at proper selection of design space and threshold values of the aerodynamic and geometric constraints. This selection has a governing influence on the optimizer's latitude for the design improvement and consequently on the optimized shape. The present problem formulation has thirteen design variables (`spn`, and at root, mid and tip sections of wing: `thkscal`, `chdscal`, `tranx` and `twst`). Each section of this wing is initially a NACA-0004 airfoil defined in the $x - z$ plane. The objective function is again selected to be the maximization of C_L/C_D, subject to 14 geometric and two aerodynamic inequality constraints:

$$C_L \geq C_{L_{\min}} \quad , \quad \text{and} \quad C_D \leq C_{D_{\max}} \tag{6.1}$$

$$V_{\text{wing}} \geq 0.8 V_{\text{wing}}^{\text{initial}} \quad , \qquad A_{\text{midspan}} \geq 0.6 A_{\text{midspan}}^{\text{initial}} \tag{6.2}$$

at the root, mid, and tip sections:

$$2^\circ \leq \theta_{0.90\text{chord}} \leq 20^\circ \quad , \qquad 2^\circ \leq \theta_{0.98\text{chord}} \leq 20^\circ \tag{6.3}$$

where $C_{L_{\min}}$ and $C_{D_{\max}}$ are the threshold values of the lift and drag coefficients respectively. V denotes the wing volume, A denotes wing section area and θ is the trailing-edge included angle. Utility of imposing such volume and angle constraints has been discussed in Chapter 5. The current problem formulation also imposes side (equality) constraints on the design variables; for example, half-span is tightly bounded to prevent the optimizer from performing ingenuous design featuring increased span in pursuit of higher lift (see Table 6.2, sensitivity with respect to span).

It is felt that a reliable and comprehensive wing design methodology should be able to address issues such as effect of grid refinement and viscosity on the optimized shape. Thus, the primary objective of this study is to quantify these effects and elevate the wing design methodology from the realm of conjectures on these issues.

The difficulty attendant with these studies is that for all the cases, initial flowfields themselves are different in the character, and consequently predict different values of lift and drag coefficients for the identical initial shape (Figure 6.3 and Table 6.1). These deviations generate, for the same values of $C_{L_{\min}}$ and $C_{D_{\max}}$, different initial aerodynamic constraints and lead the optimizer to compute different search directions and optimized shapes. In order to neutralize this initial influence on the optimizer and facilitate judicious evaluation of the effect of grid refinement and viscous computations enroute the optimized shape, $C_{L_{\min}}$ and $C_{D_{\max}}$ are altered in corresponding cases to obtain identical initial constraints. Various optimization cases and their rationale are summarized in Table 6.3. As in Chapter 5, the present optimization studies also use full CFD analyses for flowfield computations prior to the search direction evaluations as well as for the one-dimensional searches. During the one-dimensional searches, the wing geometry is perturbed by $\Delta\alpha = 0.01$ (Eqs. 4.6 and 4.7) and successive flowfield analyses are performed with convergence tolerance $\vartheta(4)$.

The first phase of the study consists of cases 6.1 to 6.4 (Table 6.3). The primary objectives of this phase are to study the effects of initial constraints on the optimized shapes and to compare the optimized shapes generated by the Euler- and TLNS-based computations for the identical values of initial aerodynamic and geometric constraints. Due to the consideration of computational efficiency, only coarse-grid computations are performed. In order to make the comparisons broad-based, two different initial aerodynamic constraint conditions are selected. For the first condition, lift constraint is violated and drag constraint is inactive (lift-driven design, cases 6.1 and 6.2), whereas for the second condition, lift as well as drag constraints are violated, with the drag violation being more severe than the lift violation (drag- and lift-driven design, cases 6.3 and 6.4).

Presented in Table 6.4 is a summary of aerodynamic features, geometric changes, and statistics for optimization cases 6.1–6.4. In Figure 6.4, optimized wing shapes for all cases are contrasted against initial shape. For this purpose, planform shape, root and mid sections (with zero angle of attack) are selected. Evolution of the objective function and aerodynamic coefficients are given in Figure 6.5. It can be observed that optimization results are very distinct for two different initial constraint conditions for both Euler and TLNS optimization cases (case 6.1 vs 6.3, and case 6.2 vs 6.4, respectively, in Table 6.4, Figures 6.4 and 6.5). As explained before, these two constraint conditions represent lift-driven, and drag- and lift-driven designs, respectively. This difference in the initial conditions leads the optimizer to derive aerodynamic performance improvement along different search directions in the design space and hence, produces very distinct designs as illustrated further below.

For the cases with initially violated lift constraint and inactive drag constraint (cases 6.1 and 6.2), search directions are evaluated with the emphasis on lift improvement, even incurring a drag penalty. This is reflected in optimized shapes having reduced leading-edge sweep of inboard wing and positive geometric twists of

the sections. The first search direction also results into substantial drag increase with the concomitant lift improvement. This increase in drag violates the drag constraint and, in the later stages of optimization, is sought to be satisfied via reduction of wave drag due to volume. Upon contrasting the pressure coefficient distributions of initial and optimized shapes (not shown here), it is observed that optimized shapes have reduced pressures on the upper surface with no change in pressure distribution pattern along the chord. Additionally, optimization also produces increased pressures on the lower surface of the wing. Optimization results in a lift coefficient increase of about 8% and drag coefficient increase of about 3%, with reference to their respective initial values (cases 6.1 and 6.2, and Table 6.4). Euler- and TLNS-based optimizations produce the shapes which are nearly the same (cases 6.1 and 6.2, and Figure 6.4).

For the cases that initially violated lift and drag constraints, with the drag violation being more severe (cases 6.3 and 6.4), the optimizer generates the search directions with the focus on reduction of drag coefficient. This focus is instrumental in producing optimized shapes having increased leading-edge sweep, reduced thickness-to-chord ratios and substantially trimmed wings. These changes lead to the reduction in wave drag and pressure drag.

Another remarkable feature of the optimized design is that outboard portions of the optimized wings have drooped leading edges. TLNS-based optimized wing has an outboard portion with more pronounced leading edge-drooping as compared to that of Euler-based optimized wing (Figure 6.4). This trend can be linked to the mechanism for the realization of leading-edge thrust potential by avoiding separation which has been reported in the literature [114, 118, 93]. The main idea is to reduce the suction peaks in the vicinity of subsonic leading edges and entailing adverse pressure gradients due to pressure recovery downstream of the leading edge, by aligning the wing at high angle of attack with the freestream at the leading edge.

Case 6.3 and 6.4 optimization results in the drag coefficient reduction of 12 - 14 % and lift coefficient increase of about 3 % with reference to their respective initial values (Table 6.4). It can be seen that evolution trends (Figure 6.5) of respective objective functions and aerodynamic constraints are very similar, although large difference in the values of objective functions prevail, which is a natural outcome of exclusion of friction drag from the Euler computations, and insignificant variation of friction drag in the TLNS-based optimization, where total drag reduction is brought about by reducing the inviscid drag. It is again noted that Euler- and TLNS-based optimized shapes and their aerodynamics are quite different for the identical values of $C_{L_{\min}}$ and $C_{D_{\max}}$ (cases 6.1 vs 6.4, Table 6.4).

Subsequently, the second phase of the study is initiated which involves fine-grid-based optimization studies using Euler and TLNS (cases 6.5 and 6.6, respectively, in Table 6.3) computations. For the fine-grid-based computations, only one initial aerodynamic constraints condition, violated lift and drag constraints, is chosen (comparable to cases 6.3 and 6.4).

For the demonstration of optimization results, normalized pressure contours for the initial shape are contrasted against that for fine-grid Euler-based (case 6.5) optimized shape (Figure 6.6). Summary of aerodynamics, geometric changes, and statistics for the optimization cases in the second phase of the study, is presented in Table 6.5.

Based upon various optimization studies conducted, effects of grid refinement, inclusion of the viscous physics in the optimization, and adequate resolution of boundary layers can be ascertained. For this purpose optimization results from Cases 6.3–6.6 (Table 6.3) can be contrasted, as they have uniform initial aerodynamic constraints conditions. This is facilitated by contrasting the initial and various optimized shapes (Figures 6.7 and 6.8), and evolution of objective function and aerodynamic coefficients (Figure 6.9).

The optimized shape from the fine-grid TLNS-based computations is substantially different as compared to those from the fine-grid Euler- and coarse-grid TLNS-based computations. The leading-edge drooping, as observed in the outboard portion of the optimized wing from coarse-grid TLNS-based computations, is absent from that of fine-grid TLNS-based computations (case 6.4 vs case 6.6, Figures 6.7 and 6.8). Although, the optimized shapes are not quite identical in various cases, the pattern of aerodynamic performance improvement towards the respective optimized shapes is almost identical (Figure 6.9). However, as seen in Table 6.4 and Table 6.4, fine-grid TLNS-based optimization require much higher CPU time as compared to the same of case 6.4 and case 6.5 (18.52 hr vs. 1.10 hr and 6.78 hr).

Finally, various optimized wing designs generated with different computational strategies (cases 6.3–6.6), are analyzed via the fine-grid TLNS-based flowfield computations. This high fidelity, post-optimization flowfield analysis provides a common measure of merit for various optimization cases and serves to illustrate the trade-off between computational efficiency and performance gain. A summary of these results is presented in Table 6.6. Note that C_L/C_D values now compare differently for the cases 3.3–3.6 (Tables 6.4, 6.5 vs 6.6). It is observed from the Table 6.6 that the optimized wing from coarse-grid TLNS-based computations (case 6.4) yields aerodynamic performance comparable to that of optimized wing from fine-grid TLNS-based computations (case 6.6), while retaining a much higher computational efficiency. However, the fine-grid Euler-based optimized wing design (case 6.5) yields the poorest aerodynamic performance.

6.2 Summary of Results

In this chapter, extension of the ADI-based optimization procedure has been demonstrated for viscous flow conditions, by optimizing an arrow wing in supersonic freestream. These results demonstrate the capability of the present optimization

procedure to learn important aerodynamic lessons, enroute the design evolution. Fluid dynamic analysis using ADI and CFL3D compare well for the viscous flow condition and therefore, highlight the ADI-method's accuracy. The ADI-based flow and sensitivity analyses also reveal the strong influence of the viscous effects, when they are adequately resolved by an increased grid size. Optimization studies demonstrate the strong influence of the initial constraints conditions on the optimization path and the overall aerodynamic performance improvement. Results show that the drag- and lift-driven optimization renders better performance, as compared to merely a lift-driven optimization. The present results also highlight the necessity for a uniform and a reliable method of evaluating various designs, generated via several optimization strategies. From the standpoint of aerodynamic performance improvement, viscous fine-grid design is the most effective and Euler fine-grid design the least effective, whereas viscous coarse-grid design ranks between these extremes (quite close to viscous fine-grid design). However, the CPU-time requirement of viscous fine-grid design is the highest and that of viscous coarse-grid design the lowest, whereas Euler fine-grid design ranks between these extremes. From these results, it is concluded that to produce an aerodynamically efficient design, it is imperative to include viscous physics in the optimization procedure with proper resolution. However, if CPU-time constraints do not permit this option, it is advantageous to incorporate inadequately resolved viscous flow physics in lieu of properly resolved inviscid flow physics.

Based upon the present computations, it is recommended that to better utilize computational resources, a number of viscous coarse grid cases using ADI method, should initially be explored to improve optimization problem definition, design space and initial shape. Optimized shapes should be analyzed using high fidelity (viscous fine grid resolution) flow analysis to evaluate their true performance potential. Subsequently, a fine-grid shape optimization should be conducted, using ADI method, to accurately obtain the final optimized shape.

As reported in the previous chapter, the present optimization studies should be performed with a stricter convergence tolerance of $\vartheta(5)$ and design perturbations based on $\Delta\alpha = 0.02$, during one-dimensional searches. These modifications can minimize numerical noise in the design procedure and reduce the number of flow analyses and CPU time. Also, grid refinement effects on the optimized shapes should be studied with three levels of grid. The present optimization problem formulation is such that viscous flow physics is swamped by the inviscid flow physics. In order to activate the influence of viscous effects during the optimization, certain related aerodynamic constraints (for example, skin friction drag constraint) should be included in the optimization problem formulation. Also, certain geometric quantities linked directly to the inviscid flow effects (for example, thickness-to-drag ratio and wave drag) should not be designated as the design a variable. The present optimization procedure, which is based upon the laminar flow assumption, should be extended for the turbulent flows with consistent sensitivities. To ensure the feasibility of the present optimization procedure for the realistic problems involving multiple design variables and constraints, its CPU time efficiency should be improved by augmenting the ADI method with multigrid technique.

Table 6.1 Efficiency and accuracy comparisons of various cases for flowfield analysis*

	Euler		TLNS	
	Coarse Grid (43×15×9)	Fine Grid (85×29×17)	Coarse Grid (43×15×9)	Fine Grid (85×29×17)
Iterations to convergence	237	401	236	686
CPU time (sec)	278.7	3202.1	281.5	5646.7
Memory (MW)	2.25	18.2	2.25	18.2
C_L	0.12762	0.13298	0.12515	0.11488
C_D	0.01248	0.01200	0.01629	0.01893
(C_L/C_D)	10.226	11.079	7.681	6.068

*Computed flowfield for $M_\infty = 2.4$ on initial arrow wing.

Table 6.2 Efficiency and accuracy comparison of various cases for sensitivity analysis*

	Euler		TLNS	
	Coarse Grid (43×15×9)	Fine Grid (85×29×17)	Coarse Grid (43×15×9)	Fine Grid (85×29×17)
Iterations to convergence	50	115	49	182
CPU time (sec)	174.3	2893.9	173.6	4580.0
Memory (MW)	3.58	20.99	3.60	21.01
$\partial C_D/\partial(\texttt{chdscal}_{\texttt{mid}})$**	-0.1061	-0.0708	-0.1193	-0.3310
$\partial C_D/\partial(\texttt{thkscal}_{\texttt{root}})$	0.1059	0.1106	0.1125	0.0550
$\partial C_D/\partial(\texttt{tranx}_{\texttt{tip}})$	-0.0754	-0.0489	-0.0888	-0.3176
$\partial C_D/\partial(\texttt{twst}_{\texttt{mid}})$	0.1662	0.1729	0.1627	0.1163
$\partial C_D/\partial(\texttt{spn})$	1.0717	1.0486	1.0714	1.3114
$\partial(C_L/C_D)/\partial(\texttt{chdscal}_{\texttt{mid}})$	2.8407	2.1976	1.9114	3.3196
$\partial(C_L/C_D)/\partial(\texttt{thkscal}_{\texttt{root}})$	-1.5735	-1.7550	-1.0308	-0.6720
$\partial(C_L/C_D)/\partial(\texttt{tranx}_{\texttt{tip}})$	1.3308	1.5557	0.8090	0.3608
$\partial(C_L/C_D)/\partial(\texttt{twst}_{\texttt{mid}})$	-0.8053	-0.9446	-0.2808	0.0044
$\partial(C_L/C_D)/\partial(\texttt{spn})$	-5.8609	-7.0277	-2.1348	-1.1447

*Computed flowfield for $M_\infty = 2.4$ on initial arrow wing; direct sensitivity method with $\vartheta(2.5)$ convergence.

**Parameters as defined in Figure 4.2.

Table 6.3 Summary and rationale of optimization cases

Case no.	Equations	Grid	$C_{L\min}$	$C_{D\max}$	Initial Constraints[¶]	
					Lift	Drag
6.1	Euler	Coarse[#]	0.15000	0.01285	0.14923 V[†]	-0.02879 I
6.2	TLNS	Coarse	0.14710	0.01678	0.14925 V	-0.02901 I
6.3	Euler	Coarse	0.15297	0.00984	0.16575 V	0.26830 V
6.4	TLNS	Coarse	0.15000	0.01285	0.16570 V	0.26800 V
6.5	Euler	Fine[*]	0.15940	0.00946	0.16575 V	0.26832 V
6.6	TLNS	Fine[**]	0.13698	0.01479	0.16569 V	0.26794 V

[#]Coarse: 43×15×9 points; [*]Fine: 85×29×17 points; [**]Fine: 85×29×9 points.
[¶]Lift = $1 - C_L/C_{L\min}$, Drag = $C_D/C_{D\max} - 1$; [†]V: violated, I: inactive.

Rationale	Effect of Constraints	Effect of Viscosity	Effect of Grid
Cases	Euler 6.1 Vs 6.3 TLNS 6.2 Vs 6.4	Coarse 6.1 Vs 6.2 || 6.3 Vs 6.4 Fine 6.5 Vs 6.6	Euler 6.3 Vs 6.5 TLNS 6.4 Vs 6.6

Table 6.4 Optimization results for cases 6.1–6.4*

	Case 6.1	Case 6.2	Case 6.3	Case 6.4
C_L	0.13790	0.13541	0.13156	0.12839
C_D	0.01290	0.01680	0.01075	0.01438
C_L/C_D	10.69	8.06	12.24	8.93
Lift constraint	0.0807	0.0794	0.1399	0.1441
Drag constraint	0.0037	0.0012	0.0923	0.1188
Aspect ratio	0.862	0.877	0.775	0.725
$\%A_{\text{mid}}^{\text{initial}}$	101	99	80	86
$\%V_{\text{wing}}^{\text{initial}}$	95	92	81	80
Λ_{LE} (deg)	69.4-72.4	68.9-71.9	72.6-74.3	70.8-73.9
$(t/c)_{\text{root,mid,tip}}$ (%)	3.7, 3.1, 1.5	3.8, 3.0, 1.4	2.29, 1.97, 1.41	2.1, 1.62, 1.33
$(\texttt{twst})_{\text{root,mid,tip}}$ (deg)	0.10, 0.08, 0.01	0.11, 0.09, 0.01	0.13, -0.03, -0.07	0.23, -0.11, -0.1
Lift change (%)	8.05	8.19	3.09	2.59
Drag change (%)	3.35	3.13	-13.86	-11.76
1-D searches	36	37	57	73
Gradient evaluations	4	4	11	10
Memory (MW)	3.58	3.60	3.58	3.60
CPU time (hr)	0.63	0.58	1.15	1.10

*Coarse grid: $43\times15\times9$ points.

Table 6.5 Optimization results for cases 6.5–6.6*

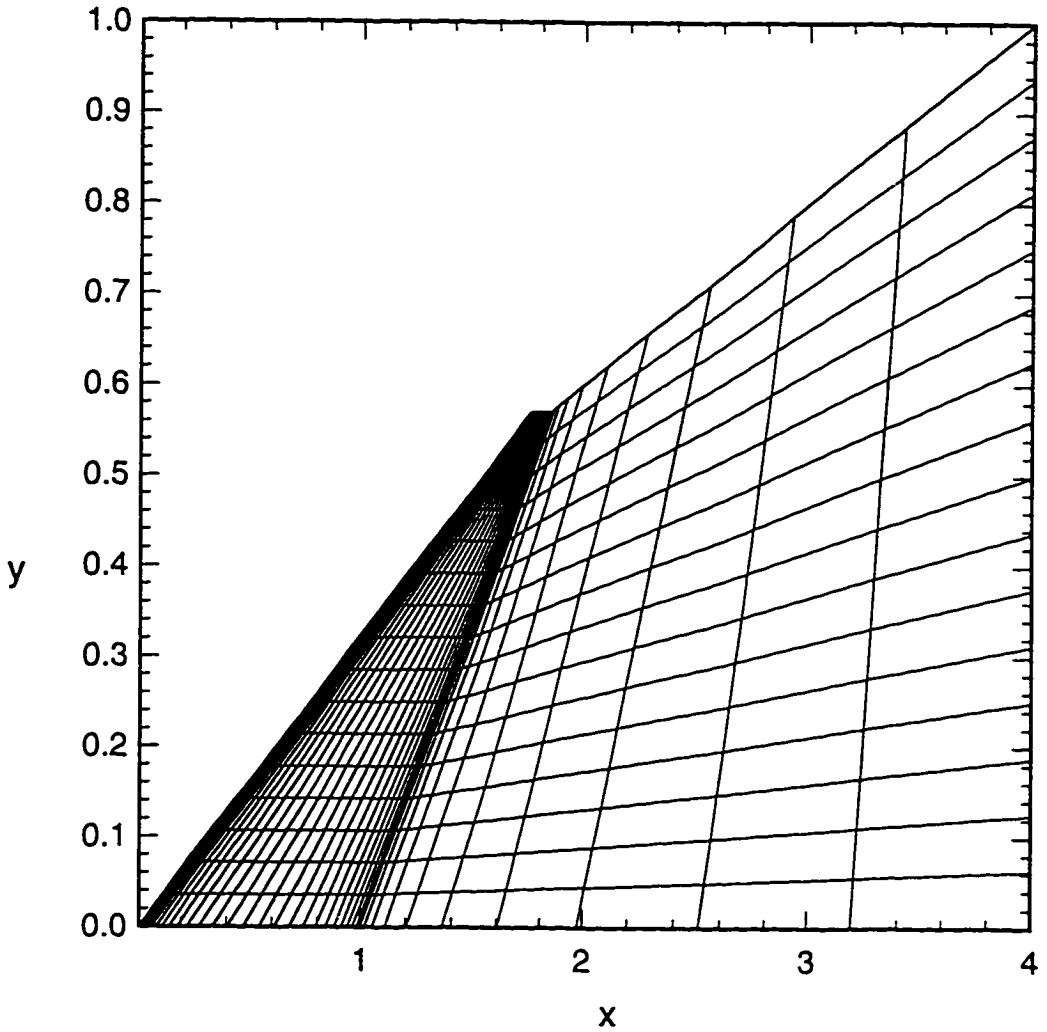
	Case 6.5	Case 6.6
C_L	0.13425	0.12453
C_D	0.01073	0.01673
C_L/C_D	12.52	7.44
Lift constraint	0.1578	0.0909
Drag constraint	0.1333	0.1317
Aspect ratio	0.786	0.607
$\%A_{\text{mid}}^{\text{initial}}$	107	119.6
$\%V_{\text{wing}}^{\text{initial}}$	106	107
Λ_{LE} (deg)	72.5-74.6	70.4-73.5
$(t/c)_{\text{root,mid,tip}}$ (%)	3.0, 2.7, 1.5	2.2, 1.4, 1.54
$(\texttt{twst})_{\text{root,mid,tip}}$ (deg)	0.05, 0.01, -0.01	0.25, 0.14, -0.01
Lift change (%)	0.96	8.97
Drag change (%)	-10.58	-11.62
1-D searches	31	82
Gradient evaluations	3	8
Memory (MW)	20.99	12.37
CPU time (hr)	6.78	18.52

*Fine grid: $85 \times 29 \times 9$ points.

Table 6.6 Aerodynamic performance and efficiency comparisons of optimized shapes from various cases*

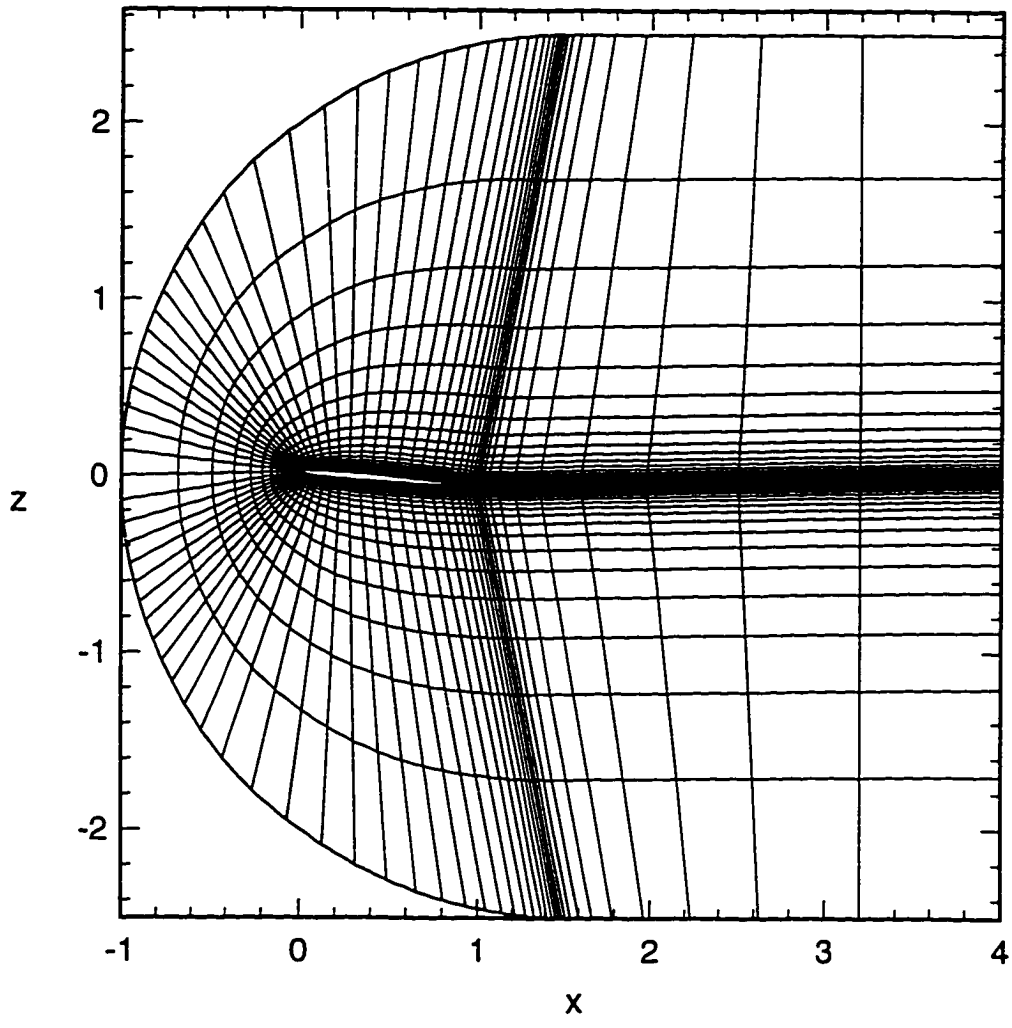
	Case 6.3′	Case 6.4′	Case 6.5′	Case 6.6′
C_L	0.11915	0.12323	0.11558	0.12453
$C_{D\text{inviscid}}$	0.00966	0.00991	0.00974	0.01037
$C_{D\text{viscous}}$	0.00701	0.00680	0.00708	0.00636
$C_{D\text{total}}$	0.01667	0.01671	0.01682	0.01673
C_L/C_D	7.15	7.38	6.87	7.44
CPU** time (hr)	1.97	1.92	7.63	18.52

*TLNS flowfield analysis for $M_\infty = 2.4$, 85×29×9 C-H grid.
**Includes respective optimization and current flowfield analysis times.



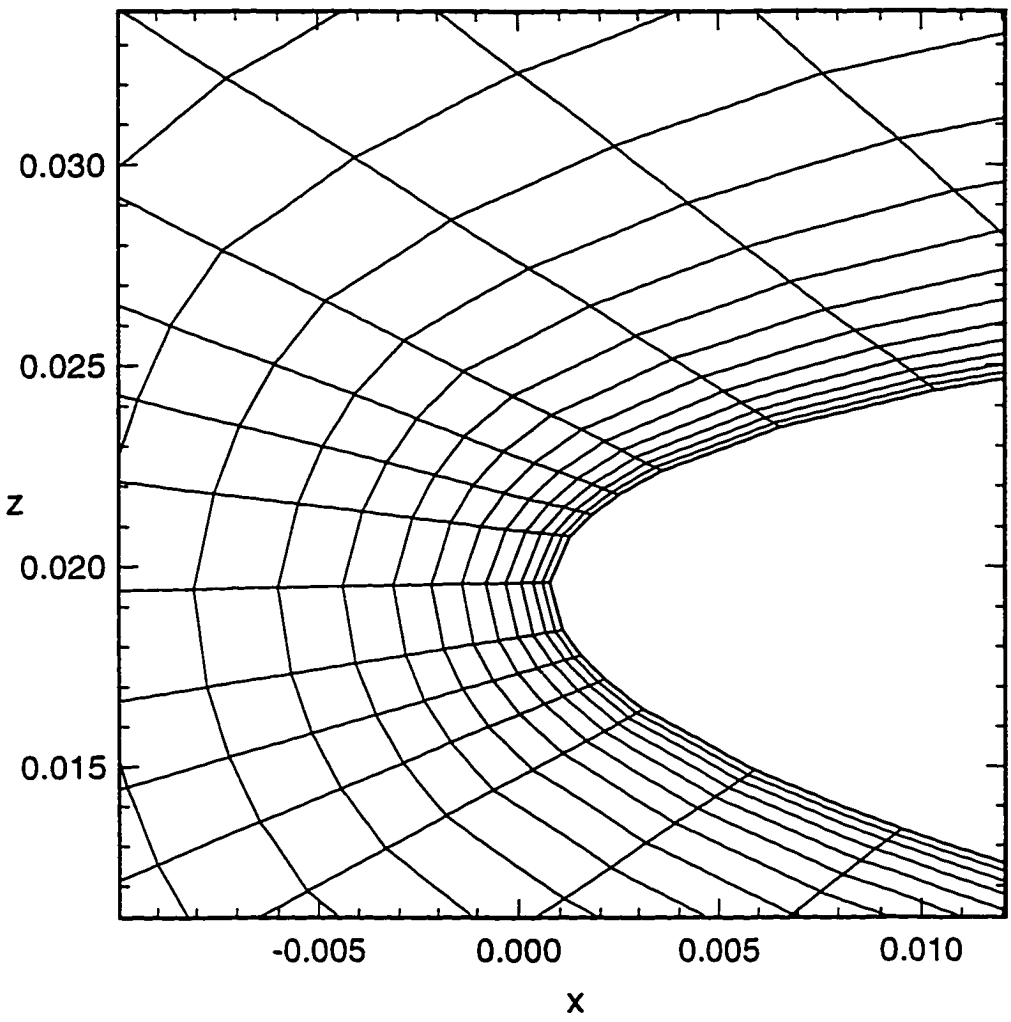
Full View of Planform Grid

Figure 6.1 Computational grid (C-H 85×29×17) for arrow wing in supersonic flow-field. M=2.4, aoa=4.5-deg.



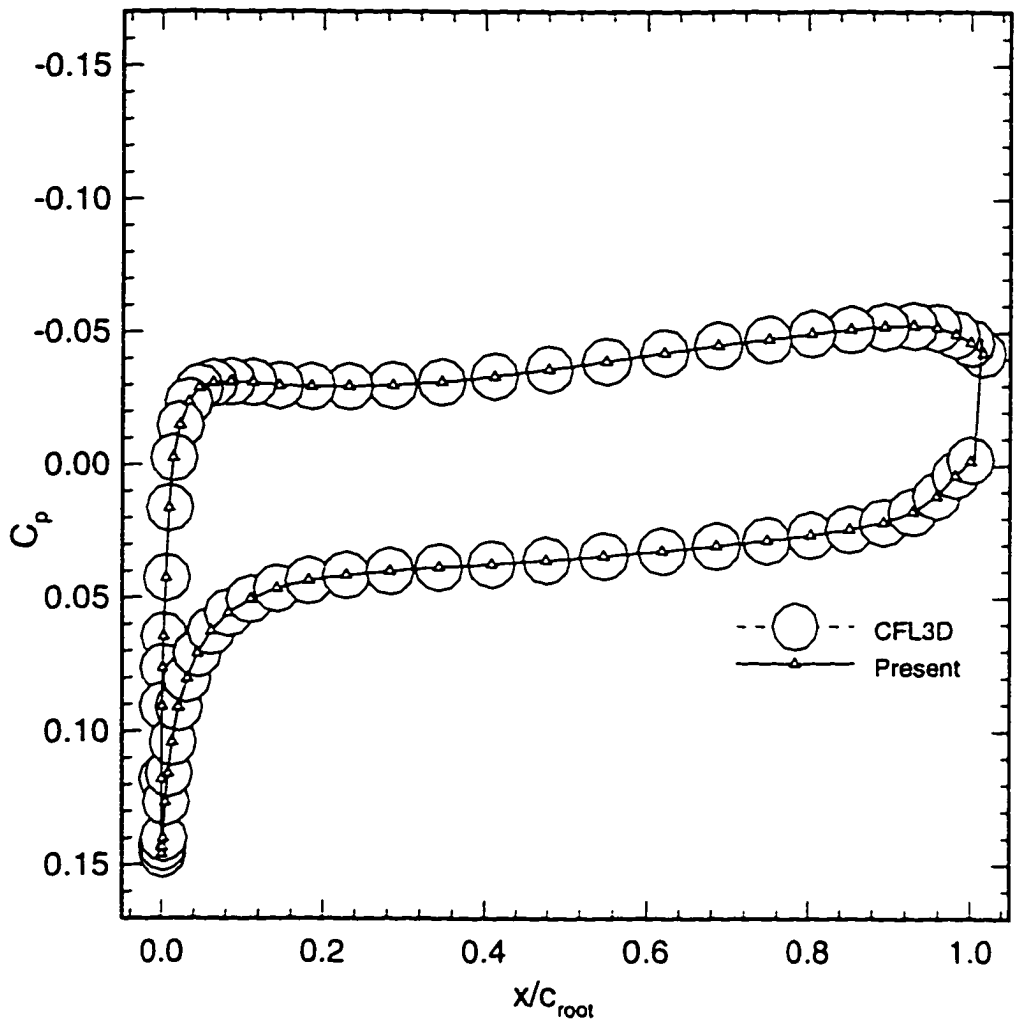
Full View of Root Section Grid

Figure 6.1 Continued.



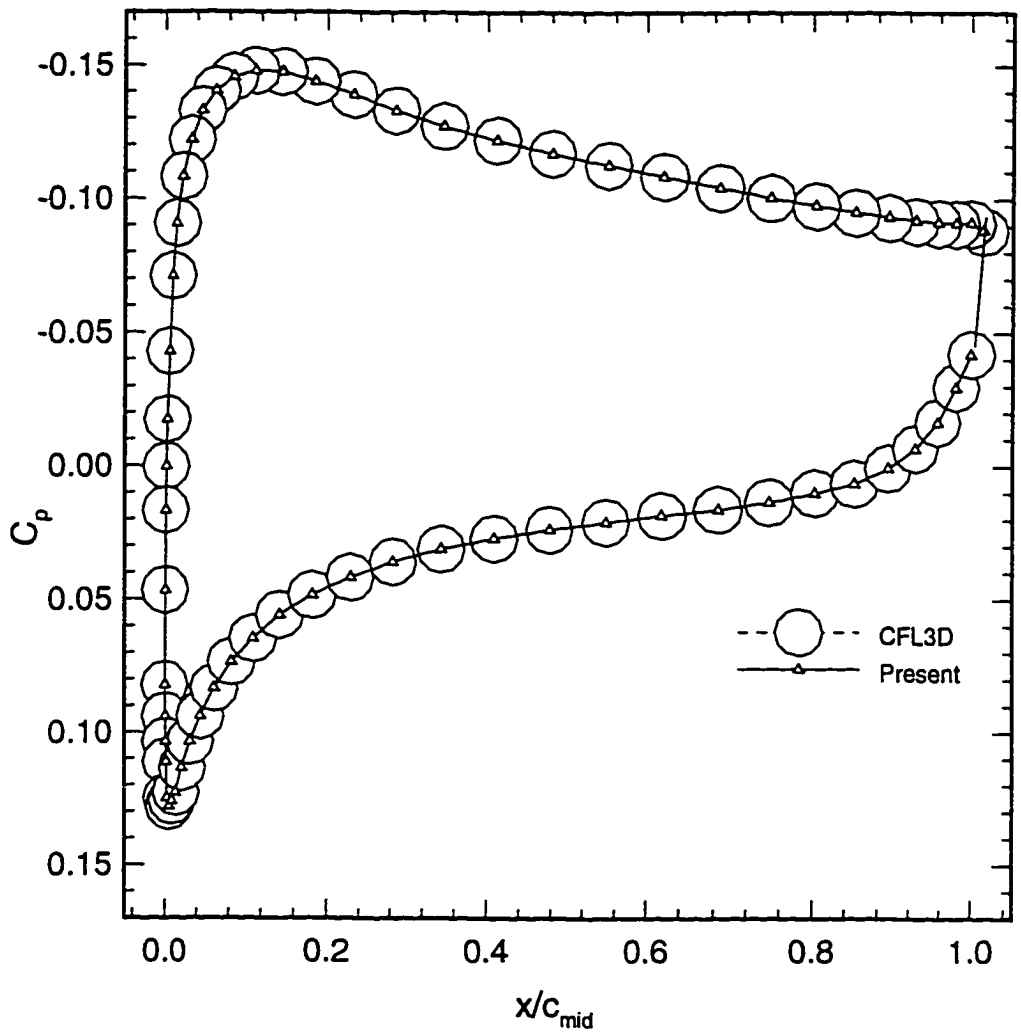
Close-up of Root Section Leading Edge Grid

Figure 6.1 Concluded.



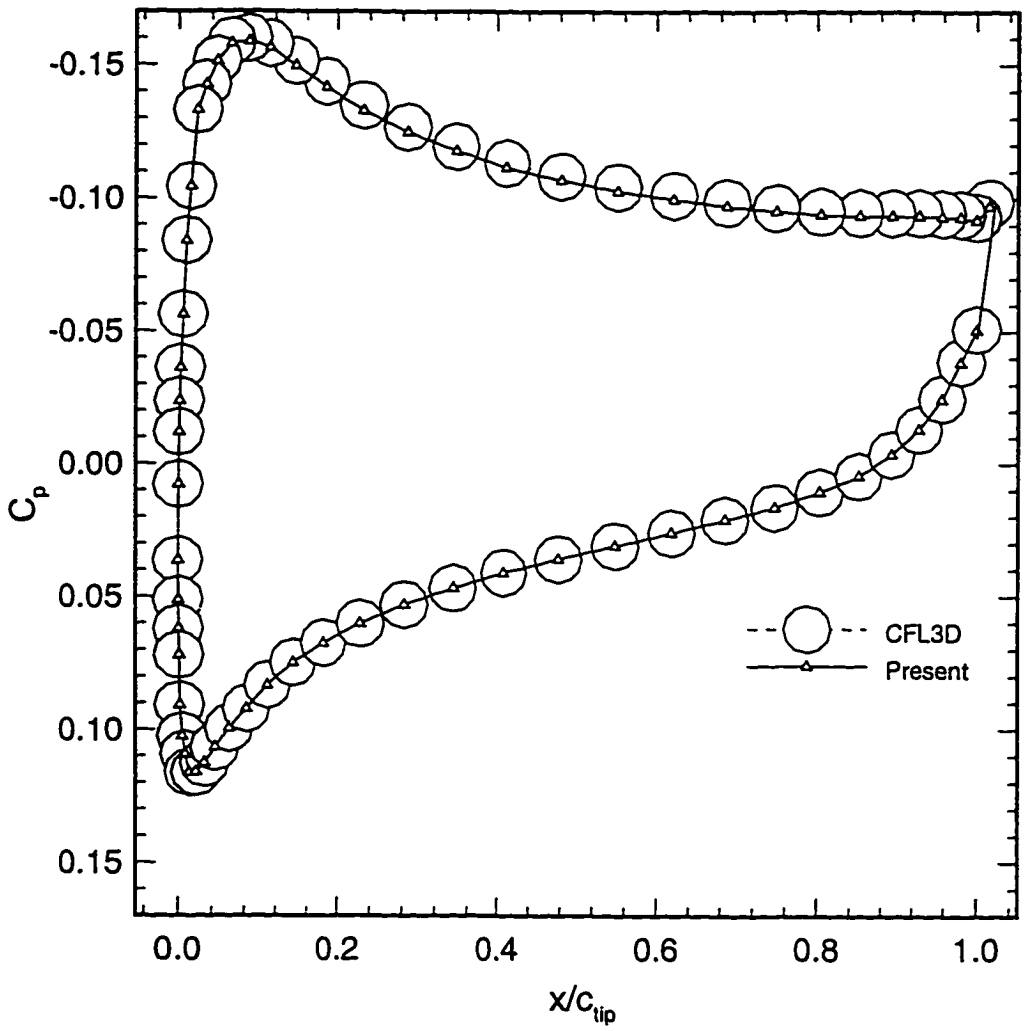
Root Section

Figure 6.2 Comparison of chordwise pressure coefficients on fine grid ($85 \times 29 \times 17$). M=2.4, aoa=4.5-deg.



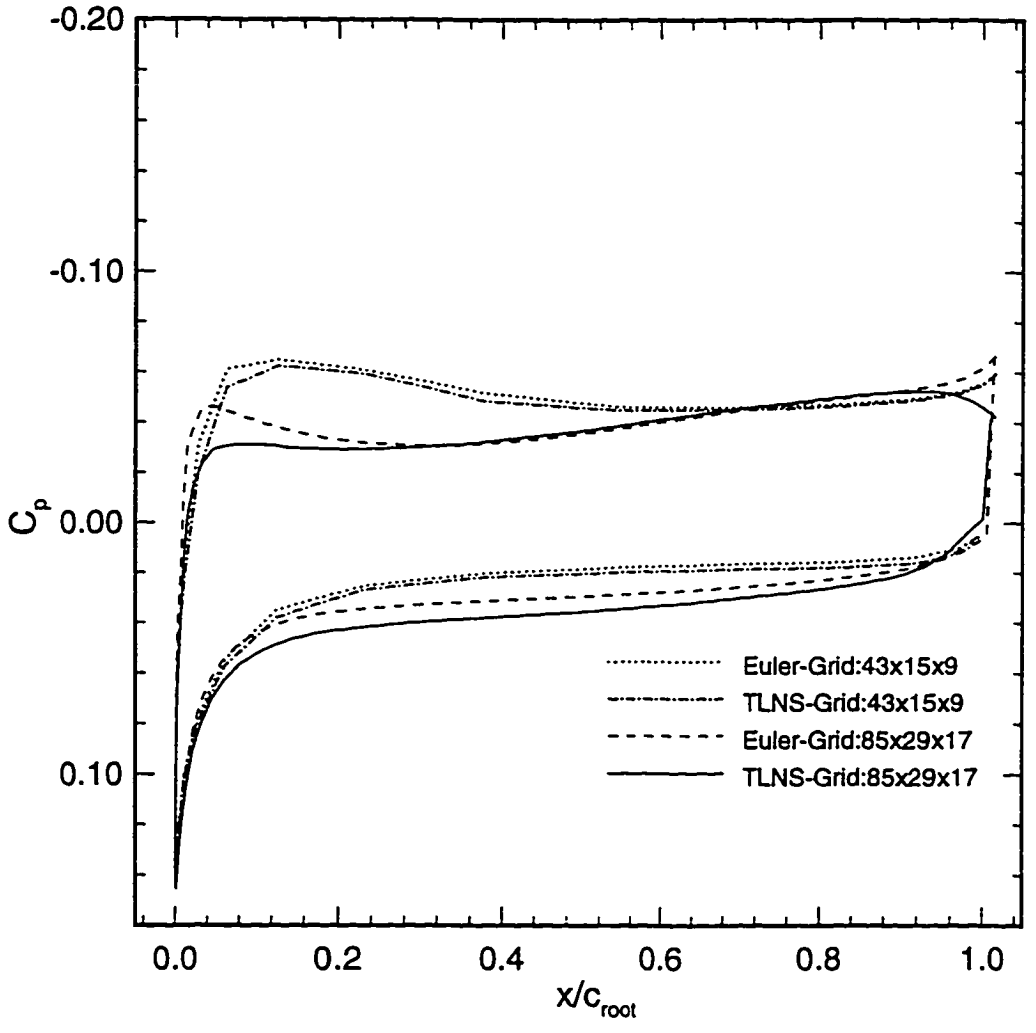
Mid Section

Figure 6.2 Continued.



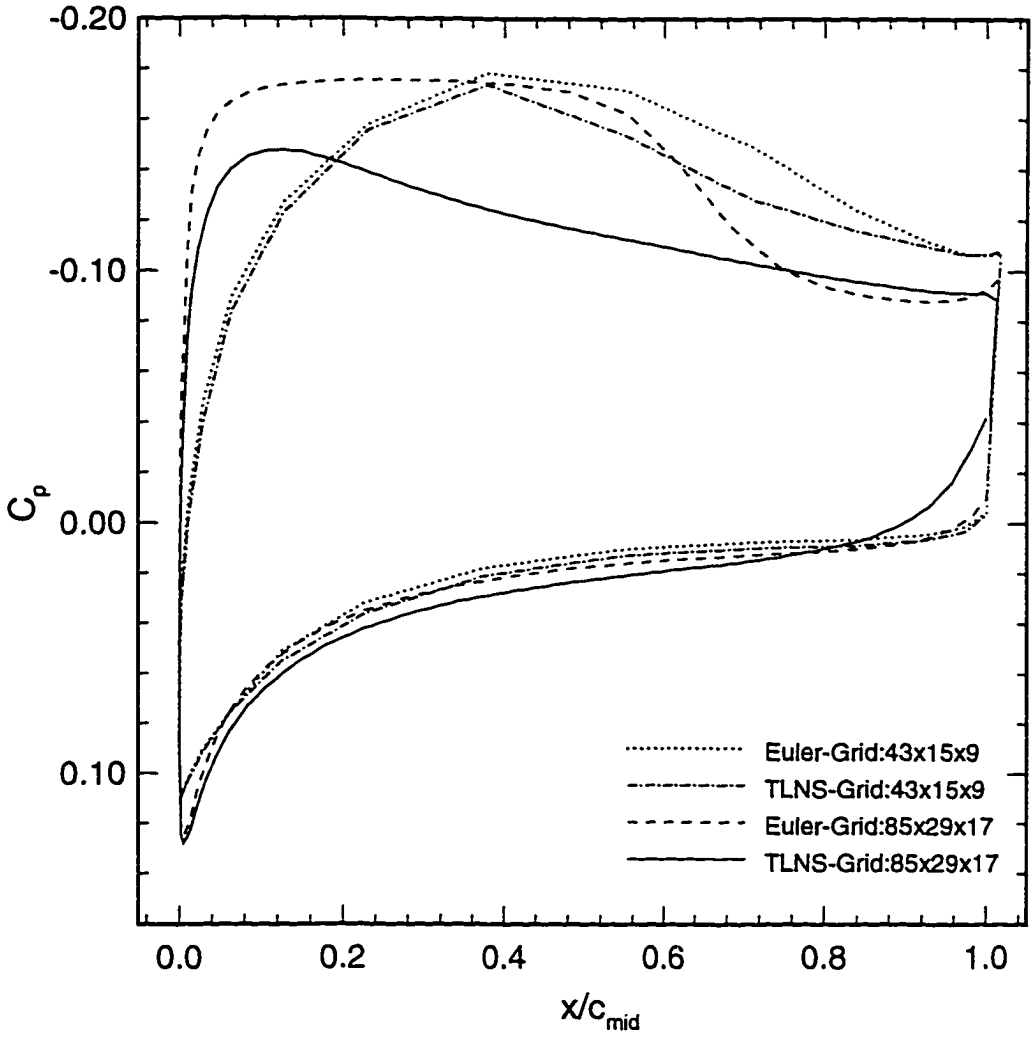
Tip Section

Figure 6.2 Concluded.



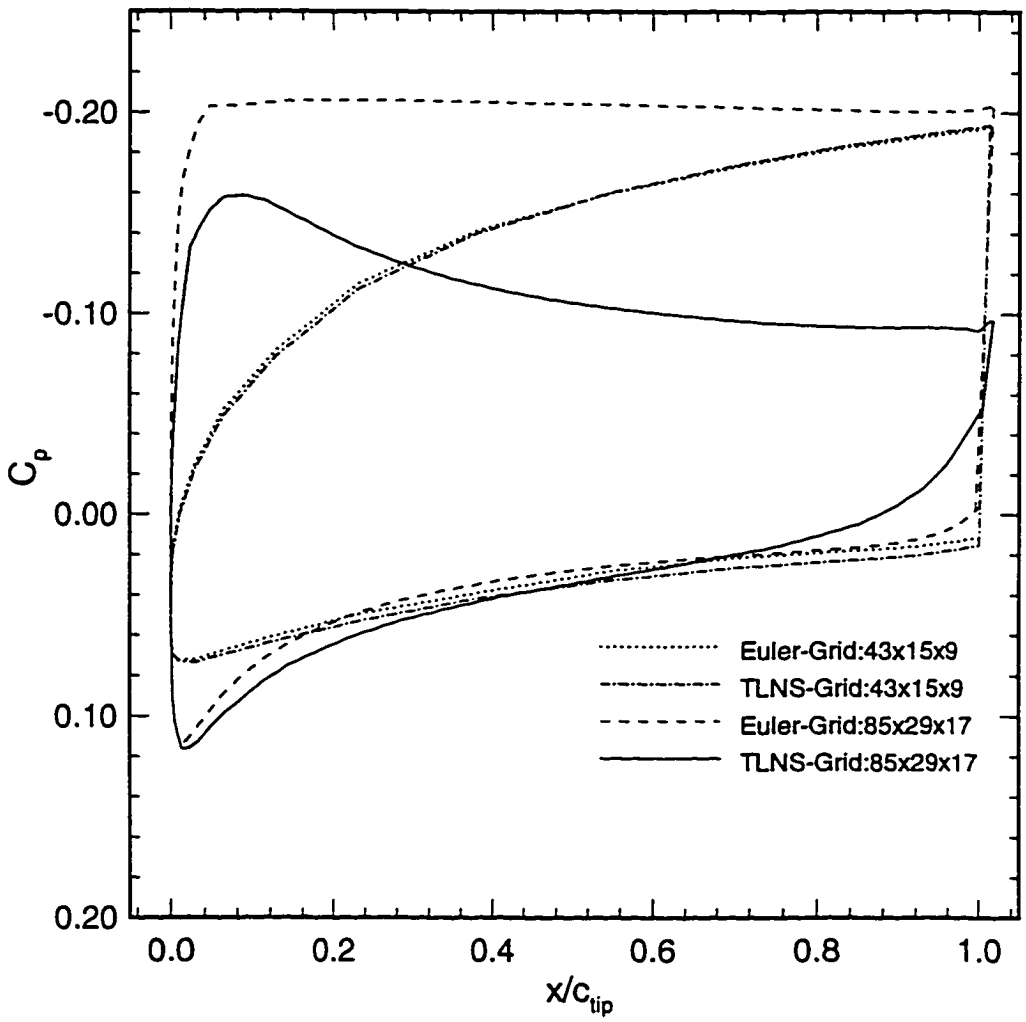
Root Section

Figure 6.3 Effects of grid refinement and viscosity on the chordwise pressure coefficient distributions. M=2.4, aoa=3-deg.



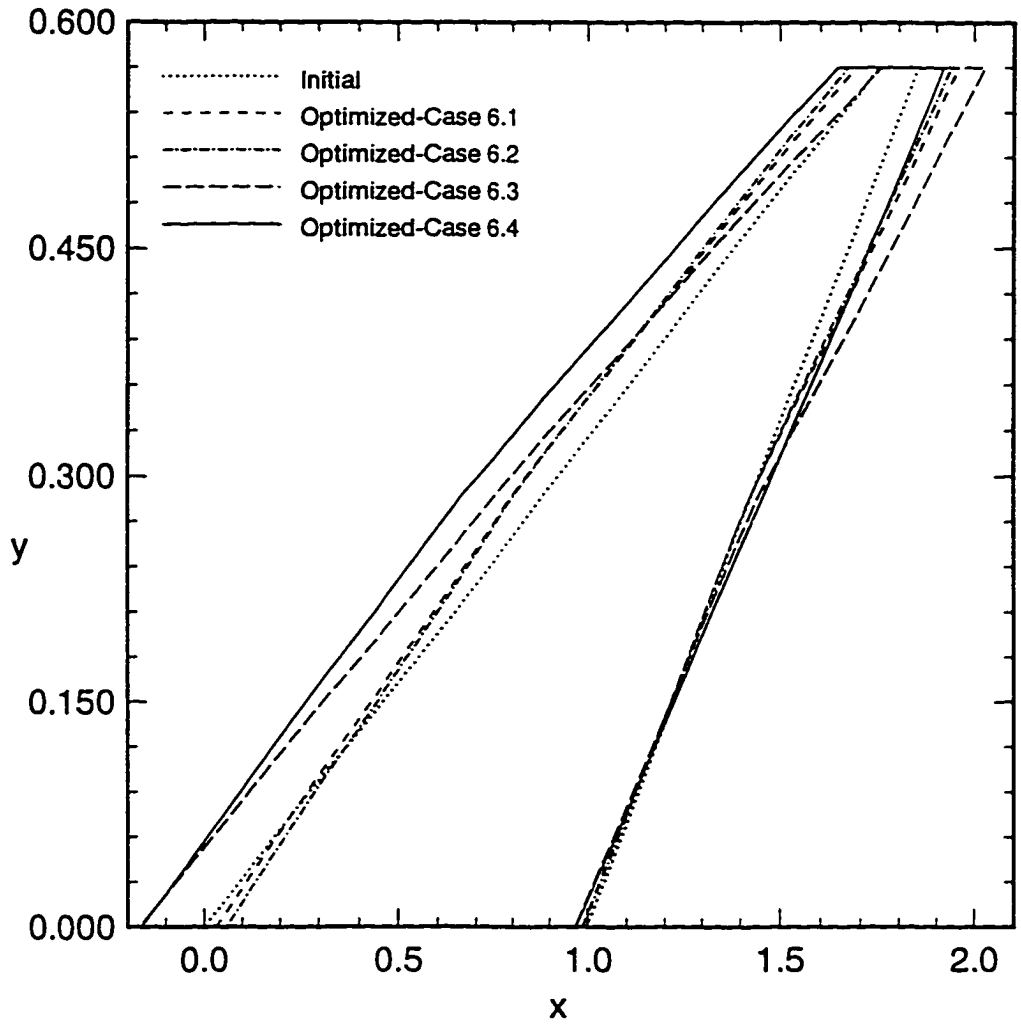
Mid Section

Figure 6.3 Continued.



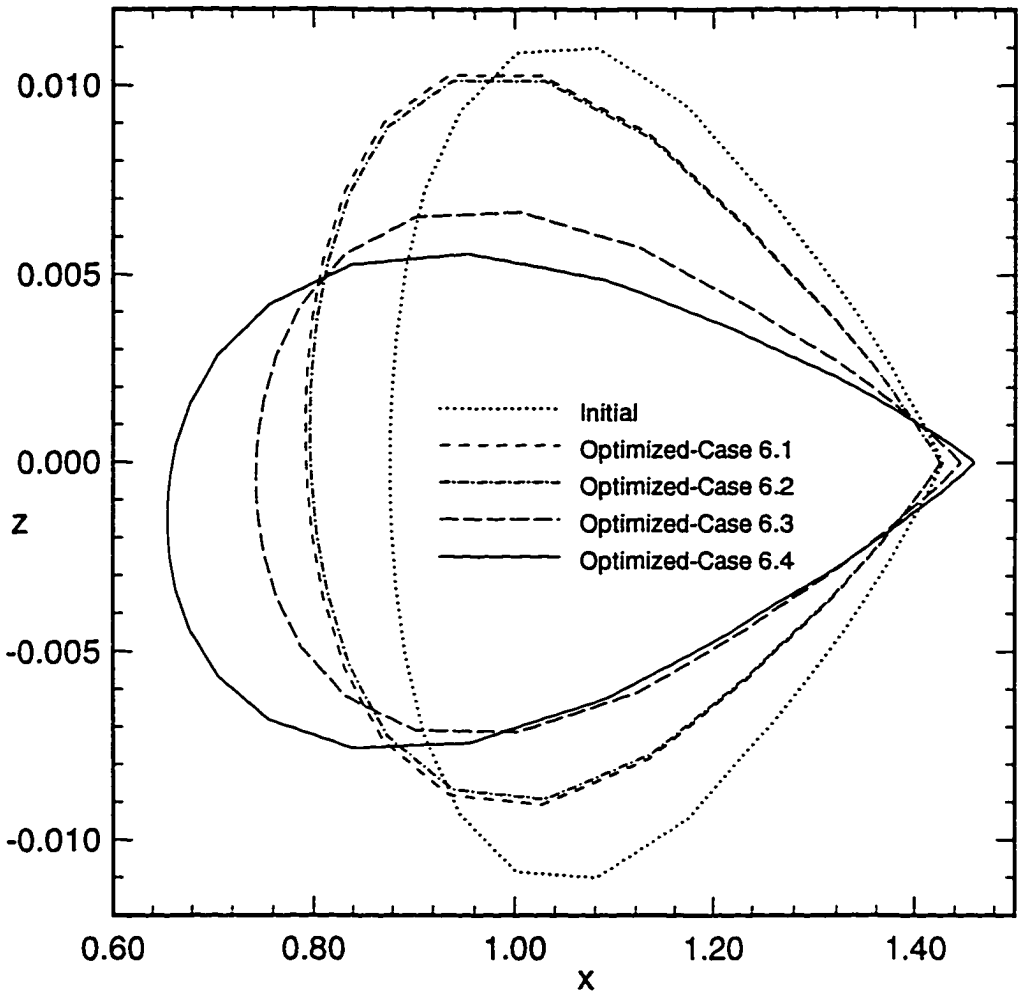
Tip Section

Figure 6.3 Concluded.



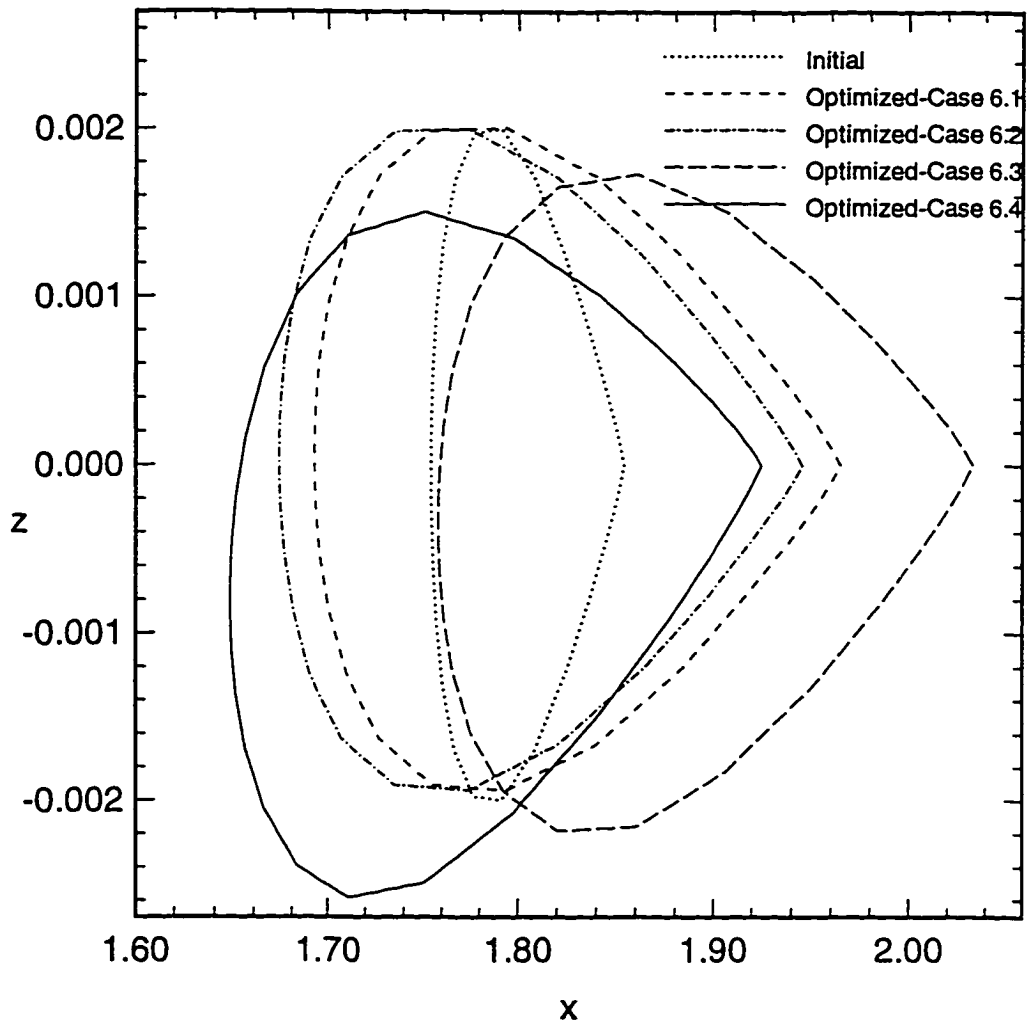
Planform

Figure 6.4 Comparison of optimized shapes from cases 6.1 - 6.4.



Mid Section

Figure 6.4 Continued.



Tip Section

Figure 6.4 Concluded.

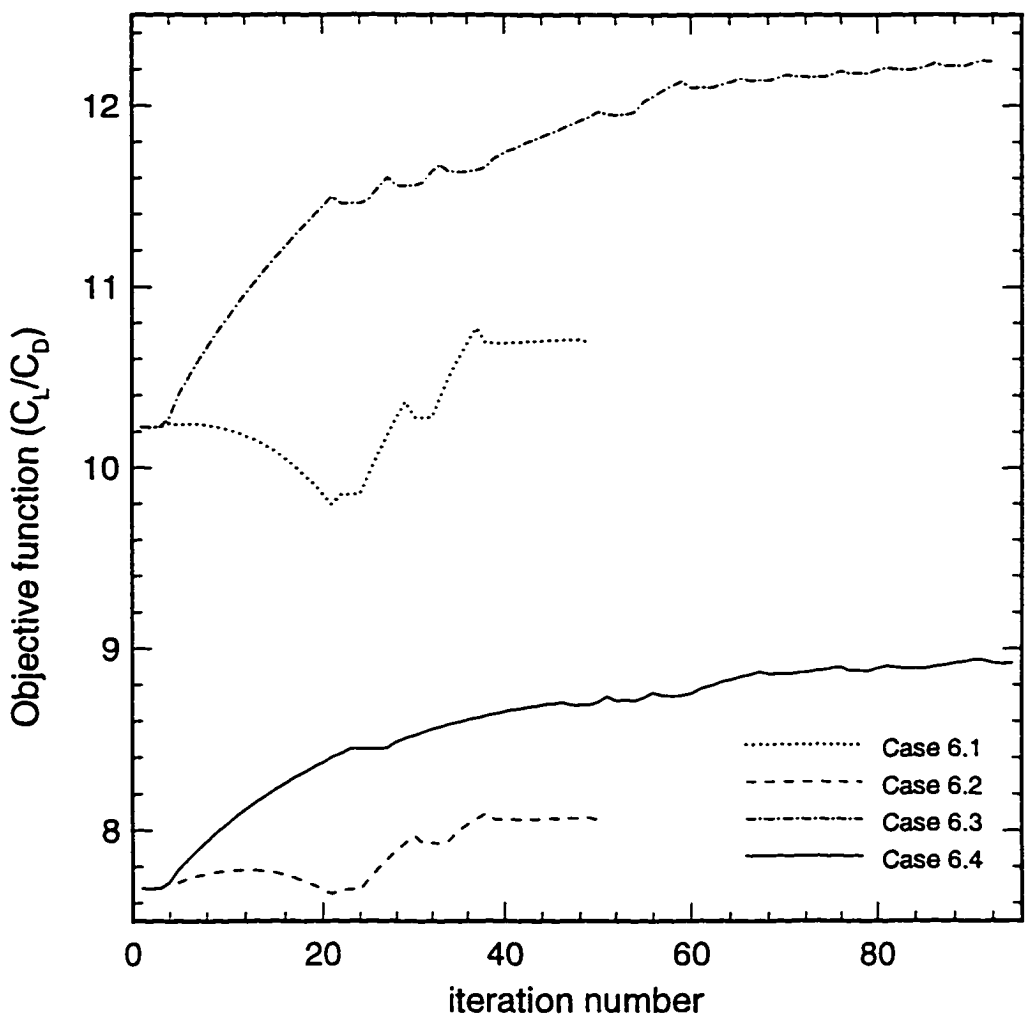


Figure 6.5 Evolution of objective function and aerodynamic coefficients from optimization cases 6.1 – 6.4.

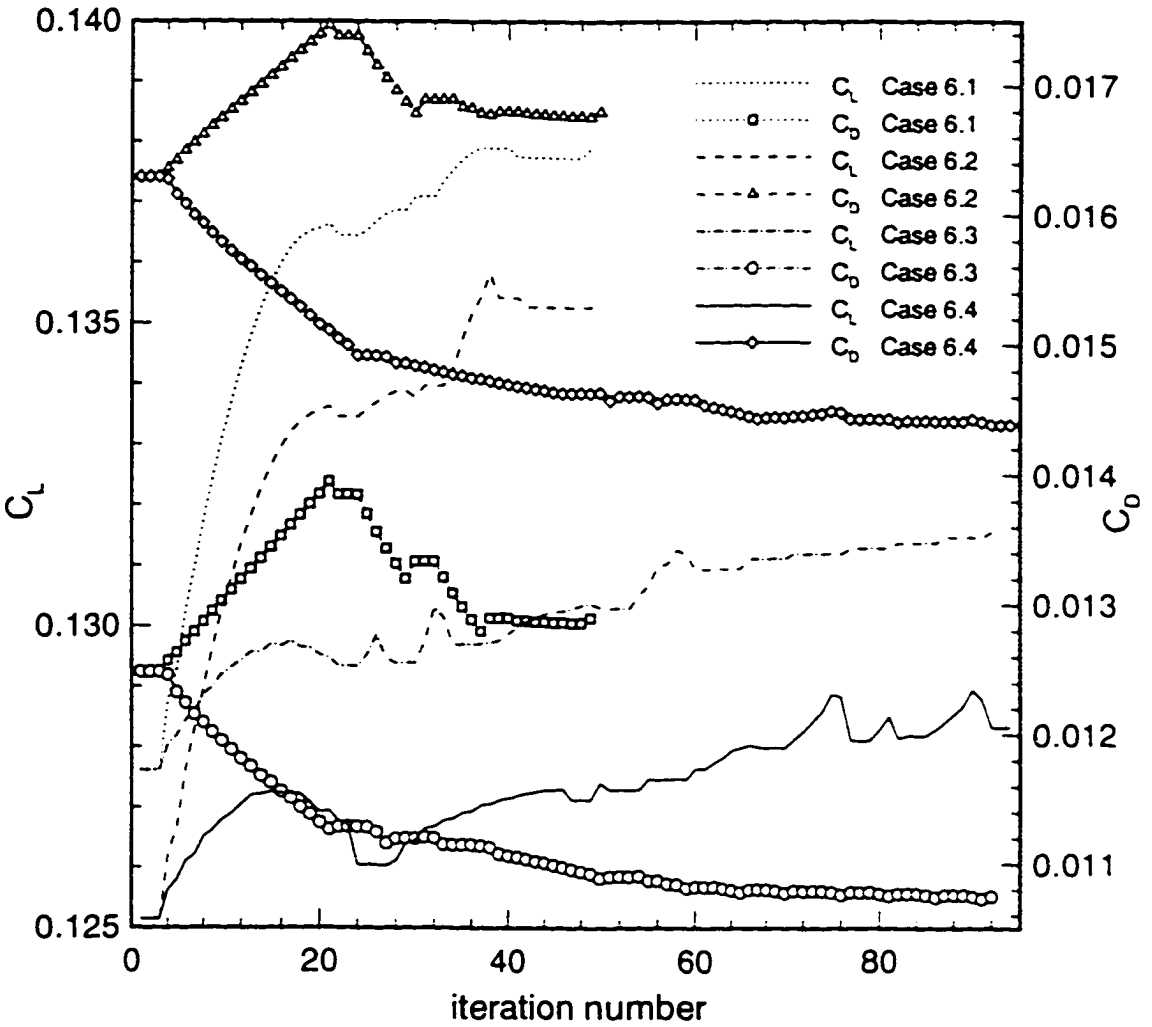
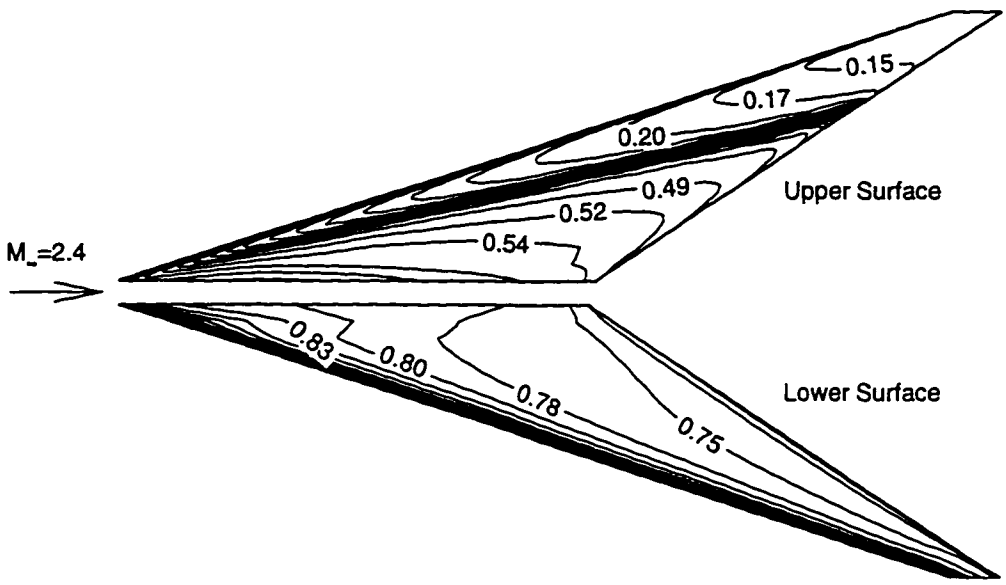
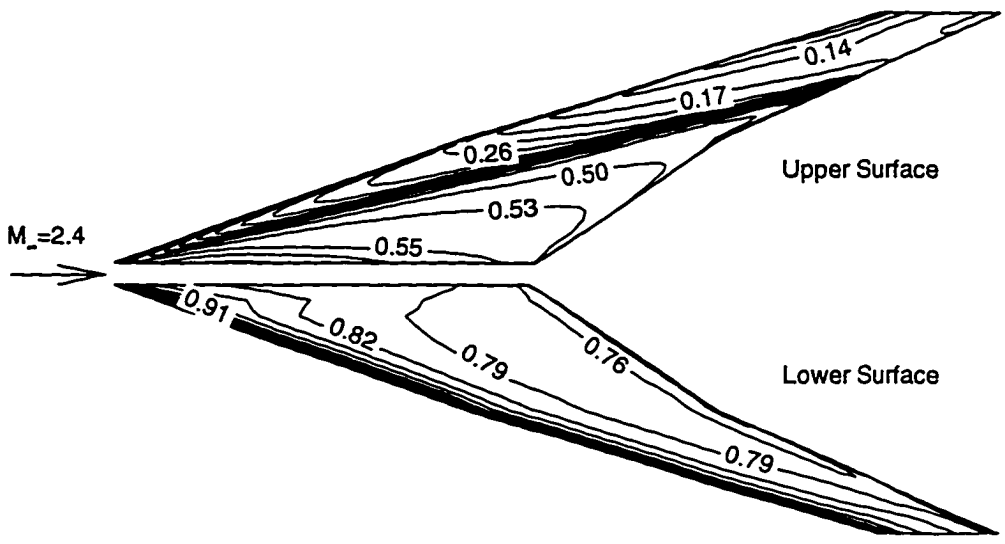


Figure 6.5 Concluded.



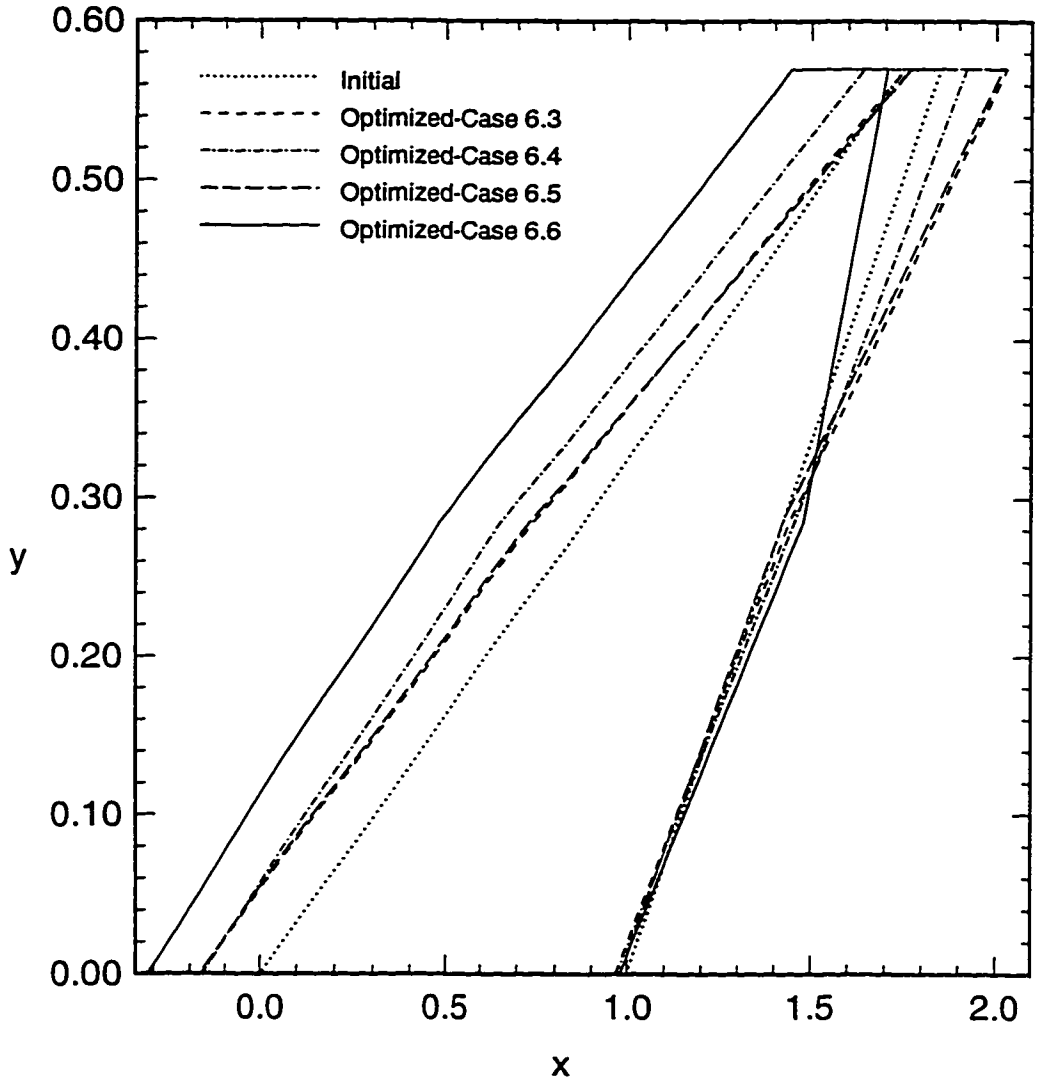
Initial Wing

Figure 6.6 Normalized pressure contours for initial and optimized wing based on fine grid (85×29×17) Euler analysis. M=2.4, aoa=3-deg.



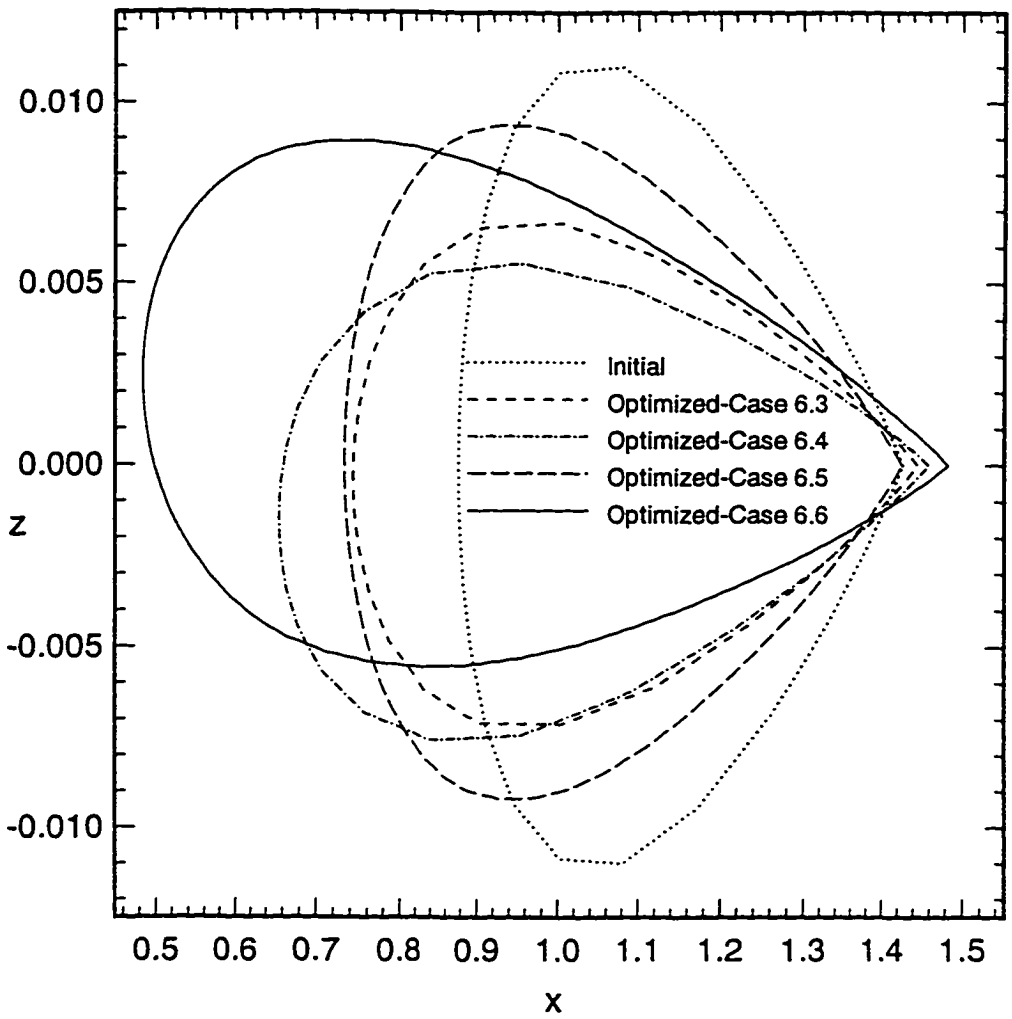
Optimized Wing

Figure 6.6 Concluded.



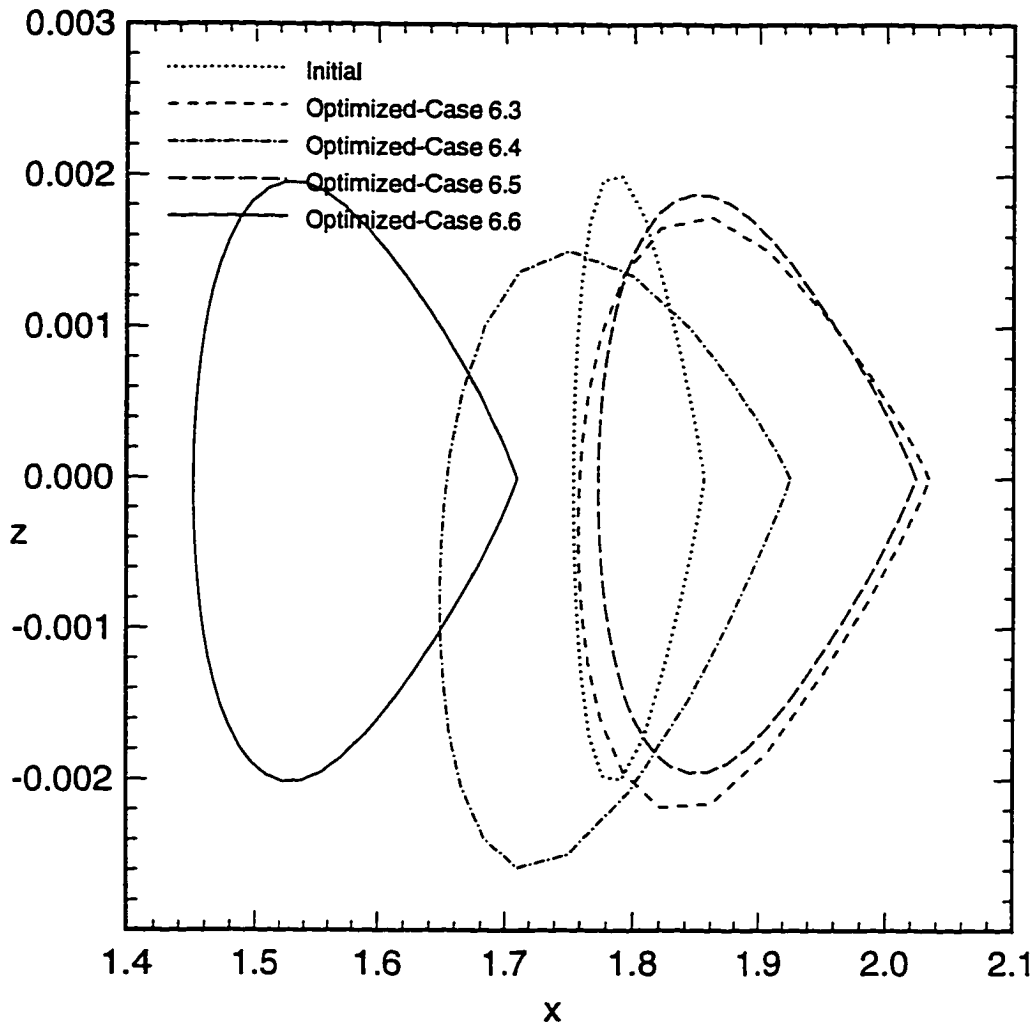
Planform

Figure 6.7 Comparison of optimized shapes from cases 6.3–6.6



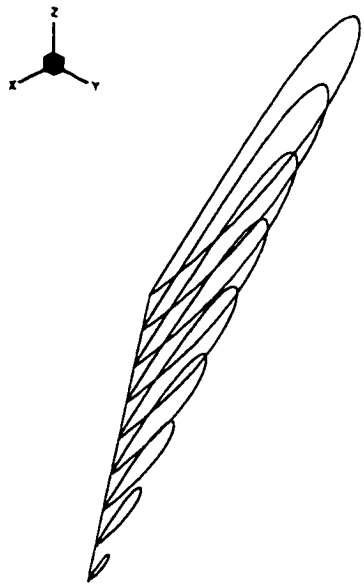
Mid Section

Figure 6.7 Continued.

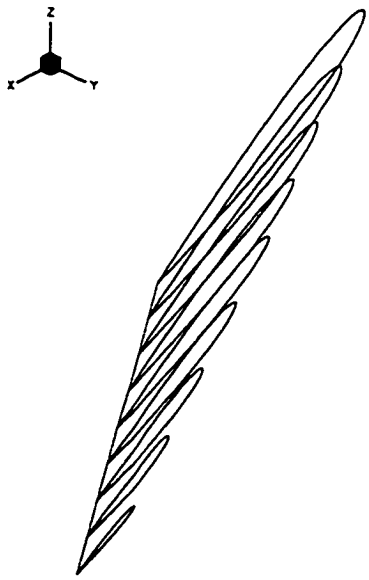


Tip Section

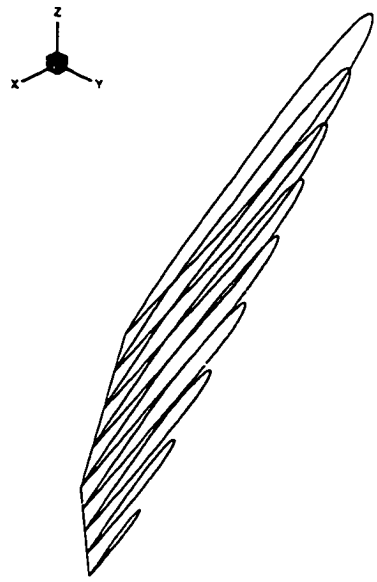
Figure 6.7 Concluded.



Initial arrow wing



Optimized arrow wing (Case 6.4)



Optimized arrow wing (Case 6.6)

Figure 6.8 Perspective view of initial and optimized shapes from cases 6.4 and 6.6.

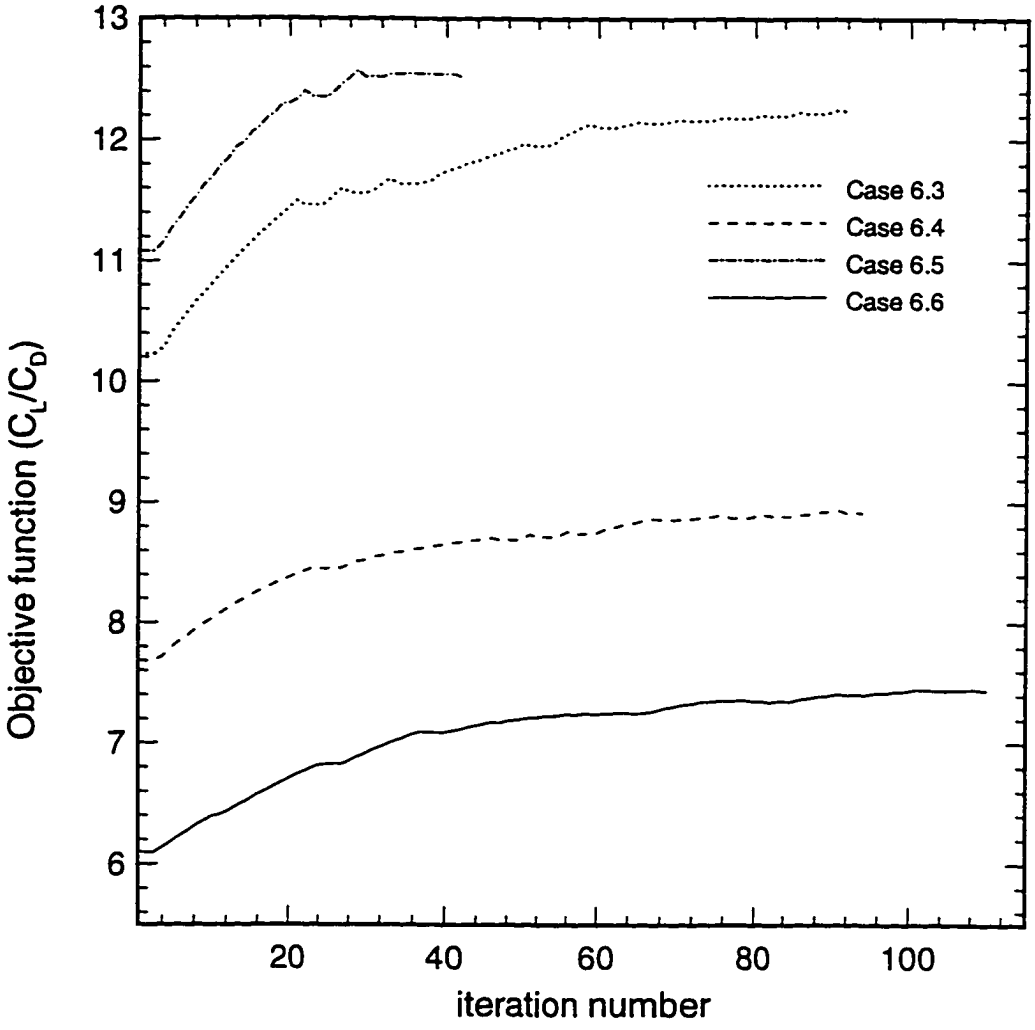


Figure 6.9 Evolution of objective function and aerodynamic coefficients from cases 6.3–6.6.

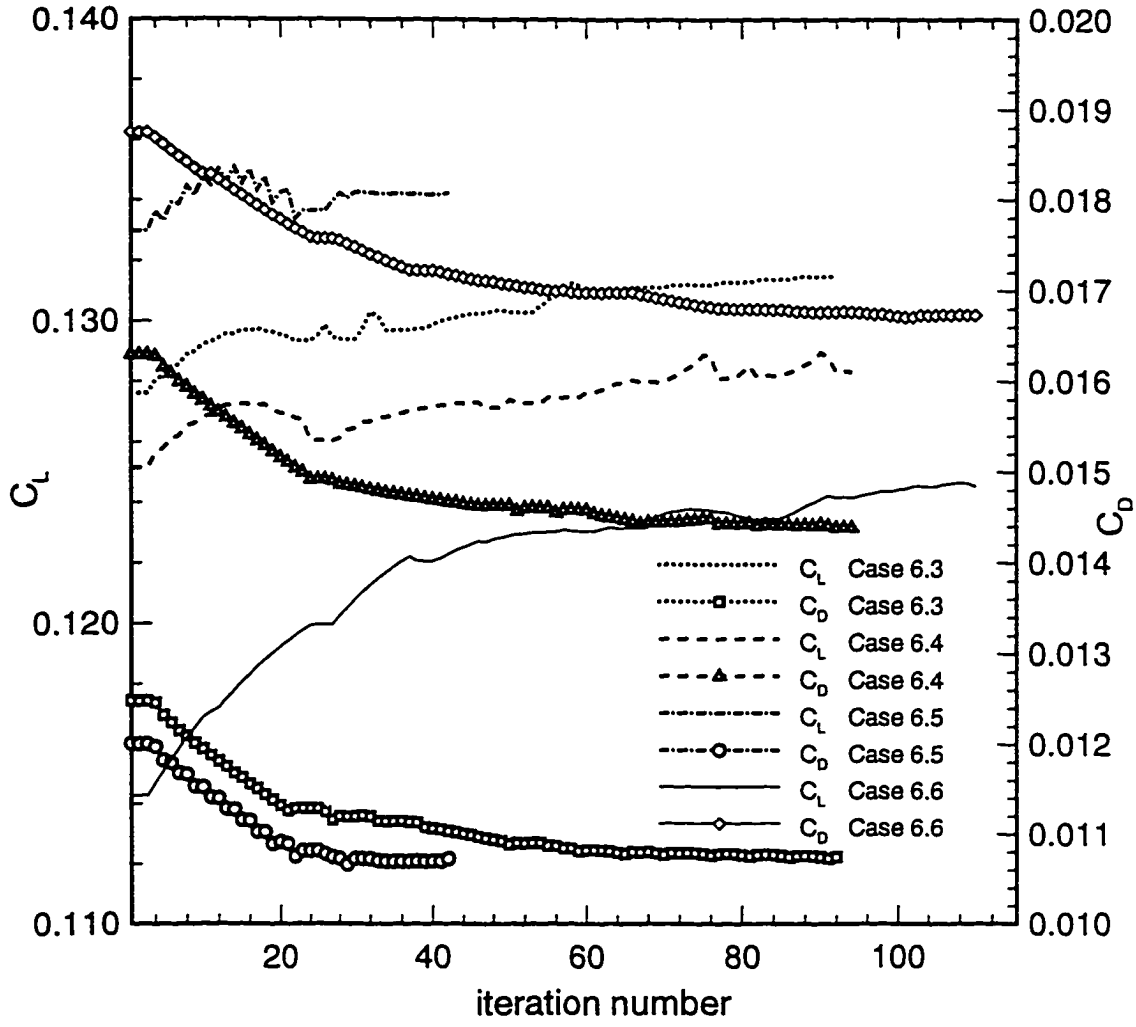


Figure 6.9 Concluded.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

A gradient-based shape optimization methodology intended for reasonably practical three-dimensional aerodynamic applications has been developed. This methodology uses CFD-based flow analysis in conjunction with the discrete sensitivity analysis. To accommodate the inherent large size of such problems, the fluid dynamic and the aerodynamic sensitivity equations have been solved using the ADI algorithm for memory efficiency. To broaden the applicability of the new methodology for the design conditions in which viscous flow physics plays a significant role, it has been subsequently extended for TLNS equations. Present work represents one of the early attempts reporting on the three-dimensional, fully integrated viscous methodology, with fine-grid resolution capability and applicability for subsonic through supersonic flow regime.

The results obtained from a flow analysis for inviscid flow over an arrow wing at Mach 2.4 compares well with results obtained from flow analyses using an unfactored approach (PCG), and an extensively used CFD code (CFL3D). The sensitivities of the present method also compare well with those obtained using the PCG approach and finite difference approach. As expected, the ADI method reduced the storage memory but increased the computing time as compared to the PCG method.

It should be mentioned that although initial flowfield analysis for the PCG method appeared to be more time intensive as compared to ADI method (Table 5.2), in a typical optimization cycle, two successive designs are only incrementally different and this feature can be exploited by PCG algorithms to render CPU-time efficient designs. These results highlight the need to synthesize the two methods in an optimization methodology for a practical application.

The results obtained from a viscous flow analysis for the previously generated initial arrow wing, compares well with those obtained using an extensively validated and used CFD code (CFL3D). The effects of viscosity and grid density have been assessed by computing the flowfield and corresponding sensitivities, using inviscid as well as viscous flow equations for both coarse and fine grids. The results indicate that adequately resolved viscous effects substantially alter the wing aerodynamics and sensitivities.

The new procedure has been demonstrated in the design of an arrow wing in the inviscid flow initially. Effects of grid refinement and convergence tolerance on the analyses as well as the shape optimization results have been explored. The results indicate that shape optimization problems requiring a large number of grid points can be resolved with a gradient-based approach. Additionally, these results indicate superior aerodynamic performance of the fine-grid optimization, and thereby establish the necessity of performing such a design. However, this option incurs a significant increase in CPU time and memory.

Next, the extended procedure based upon TLNS computations has also been implemented for optimizing the arrow wing in viscous flowfield. Several optimizaion cases are formulated to investigate the role of initial constraints, viscosity and grid size on the shapes and aerodynamics of the optimized wings.

Results show that the drag- and lift-driven optimization render better performance, as compared to merely a lift-driven design. To evaluate the influence of

properly resolved viscous effects on the optimized wing shape, design studies have been conducted based on the Euler- and TLNS-based computations for both coarse and fine grids. Coarse-grid TLNS-based optimization produces negative twist of the wing sections. This characteristic, known to alleviate vulnerability of the supersonic wings to the subsonic leading edge separation, highlights the capability of the optimization procedure to learn aerodynamic lessons enroute the evolution of shapes. TLNS-based fine-grid design produces the shape, which is quite different from its coarse-grid counterpart and Euler-based coarse- and fine-grid based shapes. This difference was expected based on the sensitivity computations.

The present results also demonstrate the necessity for a uniform and a reliable method of evaluating various designs, generated via several optimization strategies. For this purpose, post-optimization viscous high grid flowfield analysis has been performed on various optimized designs. The results of these analyses suggest that from the standpoint of aerodynamic performance improvement, viscous fine-grid design is the most effective and Euler fine-grid design the least effective, whereas viscous coarse-grid design ranks between them (quite close to viscous fine-grid design). However, CPU time requirement of viscous fine-grid design is the highest and that of viscous coarse-grid design the lowest, whereas Euler fine-grid design ranks between them. From these results, it is concluded that for an aerodynamically efficient design, it is imperative to include viscous physics in the optimization procedure with proper resolution. However, if CPU time constraints do not permit this option, it is advantageous to incorporate inadequately resolved viscous flow physics in lieu of properly resolved inviscid flow physics.

Based upon the present computations, it is recommended that to better utilize computational resources, a number of viscous coarse-grid cases using PCG (preferably) or ADI method should initially be explored to improve optimization problem definition, design space and initial shape. Optimized shapes should be analyzed

using high fidelity (viscous fine-grid resolution) flow analysis to evaluate their true performance potential. Subsequently, a fine-grid shape optimization should be conducted, using ADI metohd, to obtain the final optimized shape accurately.

Present optimization studies should be investigated further using a stricter convergence tolerance of $\vartheta(5)$ and design perturbations based on $\Delta\alpha = 0.02$ during one-dimensional searches. These modifications can minimize numerical noise in the design procedure and reduce the number of flow analyses and CPU time. Also, grid refinement effects on the optimized shapes should be comprehensively studied, by performing design computations with three levels of grid. Optimization problem formulation should facilitate explicit influence of the viscous effects on the design by including, for example, skin friction drag as a constraint. Also, design computations should be performed for multi-point and multi-objective problem formulations to highlight the amenability of the presently developed gradient-based numerical optimization proccedure for such problems.

Applicability of the present procedure should be extended for more challenging and practical problems, such as wing with trim, wing-fuselage, or wing-nacelle optimization involving several hundred design variables and constraints, to account for the natural aerodynamic interference. To meet these goals, several important extensions of the present methodology have to be made so that its CPU time and memory requirements become reasonable.

In this context, the current design procedure can be made more efficient by incorporating multigrid technique to the present ADI-based solution methodology. This extension can make design computations more efficient by increasing the convergence rate of each of the multiple CFD analyses required for the optimization. The present methodology, which is based on the laminar flow assumption, should be extended for the turbulent flows with the consistent computations for the sensitivities, to make it more representative of practical flowfields. The optimization

strategy followed in this work employs the constant step size of the design variable perturbations during the one-dimensional searches. Instead, use of other commonly applied strategies in structural optimization, such as golden-section method should be explored. This modification can shorten the length of the one-dimensional search loop and reduce the number of flow analyses and CPU time. Memory constraints of complex and inherently large optimization problems indicated previously can be addressed by either parallelizing the present code in distributed memory environment, which also would reduce the turnaround time. Another alternative is to implement the concept of SADD (sensitivity analysis on decomposed domains) in the present procedure, which would also ease the difficulty in generating the structured grid for the complex geometries. In order to represent complex shapes accurately and perform their design in a more automated manner, surface parameterization techniques such as NURBS or PDE-based methods should be incorporated in the present procedure. Finally, aerodynamic optimization methodology should be extended for the aeroelastic applications to reflect the coupling which actual problems exhibit.

REFERENCES

[1] Talay, T. A., "Introduction to the Aerodynamics of Flight," NASA SP 367, 1975.

[2] Bushnell, D. M., "Supersonic Aircraft Drag Reduction," AIAA Paper 90-1596, June 1990.

[3] Masuda, K., and Yoshida, K., "Improving the Lift to Drag Characteristics of SST," AIAA Paper 91-3105, September 1991.

[4] Silver, B., and Ashley, H., "Optimization Techniques in Aircraft Configuration Design," Chapter 12, *Structural Design Application of Mathematical Programming Technique*, AGARD, 1970, pp. 174-194.

[5] Malone, J. B., Narramore, J. C., and Sankar, L. N., "Airfoil Design Method Using the Navier-Stokes Equations," *Journal of Aircraft*, Vol. 28, No. 3, March 1991, pp. 216-224.

[6] Baysal, O., and Eleshaky, M. E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," *AIAA Journal*, Vol. 30, No. 3, March 1992, pp. 718-725.

[7] Cosentino, G. B., and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," *Journal of Aircraft*, Vol. 23, No. 3, March 1986, pp. 192-199.

[8] Sobieski, J. S., "Multidisciplinary Optimization for Engineering Systems: Achievements and Potential," NASA TM 101566, March 1989.

[9] Bos, D., "Multidisciplinary Design Optimization of a Supersonic Transport Aircraft using a Hybrid Genetic/Gradient-Based Algorithm," AIAA Paper 96-4055-CP.

[10] Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations" *Journal of Fluids Engineering*, Vol. 113, No. 4, December 1991, pp. 681-688.

[11] Baysal, O., Eleshaky, M. E., and Burgreen, G. W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," *Journal of Aircraft*, Vol. 30, No. 6, November-December 1993, pp. 953-961.

[12] Sobieski, J. S., "The Case for Aerodynamic Sensitivity Analysis," NASA CP-2457, January 1987, pp. 77-96.

[13] Iott, J., Haftka, R. T., and Adelman, H. M., "Selecting Step Sizes in Sensitivity Analysis by Finite Differences," NASA TM 86382, August 1985.

[14] Eyi, S., and Lee, K. D., "Efficiency Improvement in Sensitivity Evaluation in Aerodynamic Shape Optimization," AIAA Paper 96-2506-CP.

[15] Haftka, R. T., and Gurdal, Z., *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, 1992.

[16] Ibrahim, A. H., and Baysal, O., "Design Optimization Using Variational Methods and CFD," AIAA Paper 94-0093, January 1994.

[17] Burkardt, J., and Gunzburger, M., "Sensitivity Discrepancy For Geometric Parameters," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 9-15, International Mechanical Engineering Conference and Exposition, November 1995.

[18] Jameson, A., and Reuther, J., "Control Theory Based Airfoil Design Using the Euler Equations," AIAA Paper 94-4272, September 1994.

[19] Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*, (Translated by S. K. Mitter), Springer-Verlag, New York, 1971.

[20] Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, Part 1, 1974, pp. 97-110.

[21] Angrand, F., "Optimum Design for Potential Flows," *International Journal for Numerical Methods in Fluids*, Vol. 3, 1983, pp. 265-282.

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

[23] Cabuk, H., and Modi, V., "Optimum Plane Diffusers in Laminar Flow," *Journal of Fluid Mechanics*, Vol. 237, 1992, pp. 373-393.

[24] Huan, J., and Modi, V., "Design of Minimum Drag Bodies in Incompressible Laminar Flow," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 37-44, International Mechanical Engineering Conference and Exposition, November 1995.

[25] Jameson, A., "Aerodynamic Design via Control Theory," NASA CR 181749, November 1988.

[26] Reuther, J. and Jameson, A., "Control Theory Based Design for Potential Flow and a Finite Volume Discretization," AIAA Paper 91-499, January 1994.

[27] Reuther, J., and Jameson, A., "Aerodynamic Shape Optimization of Wing and Wing-Body Configuratons Using Control Theory," AIAA Paper 95-0123, January 1995.

[28] Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients Based on the Transonic Small Perturbation Formulation," *Journal of Aircraft*, Vol. 27, No. 6, June 1990, pp. 507-515.

[29] Elbanna, H. M., and Carlson, L. A. "Determination of Aerodynamic Sensitivity Coefficients Based on the Three-Dimensional Full Potential Equation," AIAA Paper 92-2670-CP, June 1992.

[30] Eleshaky, M. E., and Baysal, O, "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluids Engineering*, Vol. 115, No. 1, March 1993, pp. 75-84.

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

[32] Taylor, A. C. III, Hou, G. W., and Korivi, V. M., "A Methodology for Calculating Aerodynamic Sensitivity Derivatives," *AIAA Journal*, Vol. 30, No. 10, October 1992, pp. 2411-2419.

[33] Hou, G. W., Taylor, A. C. III, and Korivi, V. M., "Discrete Shape Sensitivity Equations for Aerodynamic Problems," *International Journal For Numerical Methods in Engineering*, Vol. 37, 1994, pp. 2251-2266.

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

[35] Korivi, V. M., Taylor, A. C. III, Newman, P. A., Hou, G. W., and Jones, H. E., "An Approximately Factored Incremental Strategy For Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," AIAA Paper 92-4746-CP, September 1992.

[36] Burgreen, G. W., and Baysal, O., "Aerodynamic Shape Optimization Using Preconditioned Conjugate Gradient Methods," *AIAA Journal*, Vol. 32, No. 11, November 1994, pp. 2145-2152.

[37] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856-869.

[38] Korivi, V. M., Taylor, A. C. III, Newman, P. A., Hou, G. W., and Jones, H. E., "An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives," *Journal of Computational Physics*, Vol. 113, No. 2, August 1994, pp. 336-346.

[39] Korivi, V. M., Taylor, A. C. III, Hou, G. W., Newman, P. A., Jones, H. E., "Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code Using Incremental Iterative Strategy," *AIAA Journal*, Vol. 32, No. 6, June 1994, pp. 1319-1321.

[40] Eleshaky, M. E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for Three-Dimensional Aerodynamic Sensitivity Analysis," *AIAA Journal*, Vol. 32, No. 12, December 1994, pp. 2489-2491.

[41] Eleshaky, M. E., and Baysal, O., "Discrete Aerodynamic Sensitivity Analysis On Decomposed Computational Domains," *Journal of Computers and Fluids*, Vol. 23, No. 4, 1994, pp. 595-611.

[42] MACSYMA Reference Manual, Version 13, Computer Aided Mathematics Group, Symbolics, Inc., 1988.

[43] Bischof, C. H., Corliss, C. A., Griewank, A. and Hovland, P., "ADIFOR: Generating Derivative Codes from Fortran Programs," *Journal of Scientific Programming*, Vol. 1, No. 1, 1992, pp. 1-29.

[44] Green, L., Newman, P., and Haigler, K., "Sensitivity Derivatives for Advanced CFD Algorithm and Viscous Modeling Parameters via Automatic Differentiation," AIAA Paper 93-3321, 1993.

[45] Sherman, L. L., Taylor, A. C. III, Green, L. L., Newman, P. A., Hou, G. W. and Korivi, V. M., "First- and Second-Order Aerodynamic Sensitivity Derivatives via Automatic Differentiation with Incremental Iterative Methods," AIAA Paper 94-4262, 1994.

[46] Baysal, O., and Cordero, Y., "A Perspective on Automatic Differentiation for Gradient-Based Aerodynamic Shape Optimization," *Proceedings of 3rd ECCOMAS CFD Conference*, Paris, France, September 1996.

[47] Taylor, A. C. III, Newman, P. A., Hou, G. W., and Jones, H. E.. "Recent Advances in Steady Compressible Aerodynamic Sensitivity Analysis," Volumes in Mathematics and its Application (IMA), Vol. 68, 1994, pp. 341-356; Volume title: *Flow Control*, Ed: M. D. Gunzburger, Springer-Verlag.

[48] Frank, P. D., and Shubin, G. R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamic Design Problem," *Journal of Computational Physics*, Vol. 98, No. 1, January 1992, pp. 74-89.

[49] Lorence, C. B., and Hall, K. C., "Unsteady Aerodynamics Loads in Cascades," AIAA Paper 94-0064, January 1994.

[50] Barthelemy, J. M., and Bergen, F. D., "Shape Sensitivity Analysis of Wing Static Aeroelastic Characteristics," *Journal of Aircraft*, Vol. 26, No. 8, August 1989, pp. 712-722.

[51] Arslan, A. E., and Carlson, L. A., "Integrated Determination of Sensitivity Derivatives for an Aeroelastic Transonic Wing," AIAA Paper 94-4400, September 1994.

[52] Sorensen, T. M. and Drela, M., "Aeroelastic Sensitivity Calculations Using a Newton-Based Full Potential Equation Solver," AIAA Paper 95-1834-CP, 1995.

[53] Lighthill, M. J., "A Method of Two-Dimensional Aerodynamic Design," Aeronautical Research Council, R and M 2112, June 1945.

[54] Bauer, F., Garabedian, P., and Korn, D., and Jameson, A., *A Theory of Supercritical Wing Section, with Computer Program and Examples*, Springer-Verlag, NY, 1972.

[55] Sobieczky, H., Fung, K. Y., Seebass, R, A., and Yu, N. J., "New Method for Designing Shock-Free Transonic Configurations," *AIAA Journal*, Vol. 17, No. 7, 1979, pp. 722-729.

[56] Tranen, J. L., "A Rapid Computer Aided Transonic Airfoil Design Method," AIAA Paper 74-501, 1974.

[57] Volpe, G., and Melnik, R. E., "Role of Constraints in Inverse Design for Transonic Airfoils," *AIAA Journal*, Vol. 22, No. 12, December 1984, pp. 1770-1778.

[58] Henne, P. A., An Inverse Transonic Wing Design Method," AIAA Paper 80-0330, January 1980.

[59] Garabedian, P., and McFadden, G., "Design of Supercritical Swept Wings," *AIAA Journal*, Vol. 20, No. 3, 1982, pp. 289-291.

[60] Birckelbaw, L., "Inverse Airfoil Design Using the Navier-Stokes Equations," AIAA Paper 89-2202-CP, July-August 1989.

[61] Giles, M. B., and Drela, M., "Two Dimensional Transonic Aerodynamic Design Method," *AIAA Journal*, Vol. 25, No. 9, September 1987, pp. 1199-1206.

[62] Greff, E., Forbrich, D., and Schwarten, H., "Application of Direct Inverse Analogy Method And Viscous Design Optimization Techniques," *Proceedings of Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-III)* (Washington D. C.), (Ed. G. S. Dulikravich), October 1991, pp. 307-324.

[63] Campbell, R. L., and Smith, L. A., "A Hybrid Algorithm for Transonic Airfoil and Wing Design," AIAA Paper 87-2552-CP, 1987.

[64] Yu, N. J., and Campbell, R. L., "Transonic Airfoil and Wing Design Using Navier-Stokes Codes," AIAA Paper 92-2651-CP, June 1992.

[65] Mineck, R. E., and Campbell, R. L., "Demonstration of Multipoint Design Procedures," AIAA Paper 93-3114, July 1993.

[66] Campbell, R. L., "Efficient Constrained Design Using Navier-Stokes Codes," AIAA Paper 95-1808-CP, 1995, pp. 372-380.

[67] Naik, D. A., Krist, S. E, Campbell, R. L., Vatsa, V. N., Buning, P. G., Gea, L. M., "Inverse Design of Nacelles Using Multi-Block Navier Stokes Codes," AIAA Paper 95-1820-CP, 1995.

[68] Green, B. E., Whitesides, J. L., Campbell, R. L., and Mineck, R. E., "A Method for the Constrained Design of Natural Laminar Flow Airfoils," AIAA Paper 96-2502-CP, 1996.

[69] Lewis, J. C. and Agarwal, R. K., "Airfoil Design Via Control Theory Using Full Potential and Euler Equations," AIAA Paper 96-2483-CP, 1996.

[70] Dulikravich, G. S., "Aerodynamic Shape Design and Optimization," AIAA Paper 91-0476, January 1991.

[71] Gregg, R. D., and Misegades, K. P., "Transonic Wing Optimization Using Evolution Theory," AIAA Paper 87-0520, January 1987.

[72] Obayashi, S., and Tsukahara, T., "Comparison of Optimization Algorithms for Aerodynamic Shape Design," AIAA Paper 96-2394-CP, 1996.

[73] Hicks, R. M., Murman, E. L. and Vanderplaats, G. N., "An Assessment of Airfoil Design by Numerical Optimization," NASA TM X-3092, July 1974.

[74] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, July 1978, pp. 407-412.

[75] Ghielmi, L., Marazzi, R., and Baron, A., "A Tool for Automatic Design of Airfoils in Different Operating Conditions," AGARD-CP-463, Leon, Norway, May 22-23, 1989, pp. 18.1-18.12.

[76] Pittman, J. L., "Supersonic Airfoil Optimization," *Journal of Aircraft*, Vol. 24, No. 12, December 1987, pp. 873-879.

[77] Lores, M. E. and Smith, P. R., "Supercritical Wing Design Using Numerical Optimization and Comparisons with Experiment," AIAA Paper 79-0065, January 1979.

[78] Haney, P. P., and Johnson, R. R., "Computational Optimization and Wind Tunnel Testing," AIAA Paper 79-0080, January 1979.

[79] Dutt, H. N. V., and Srikanth, A. K., "Design of Supersonic Airfoils by Numerical Optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 19, 1979, pp. 417-427.

[80] Hager, J. O., Eyi, S., and Lee, K. D., "A Multipoint Optimization For Transonic Airfoil Design," AIAA Paper 92-4681-CP, 1992.

[81] Lee, K. D., and Liu, P. H., "A Design Optimization Method Using the Euler Equations," *Proceedings of the Sixth International Conference on Numerical Methods in Laminar and Turbulent Flow*, (Swansea), (Eds. C. Taylor et al.), Vol. 6, Part 1, July 1989, pp. 773-782.

[82] Chang, I-C., Torres, F. J., and van Dam, C. P., "Wing Design Code Using Three-Dimensional Euler Equations And Optimization," AIAA Paper 91-3190, September 1991.

[83] Reuther, J., Cliff, S. E., Hicks, R. M., and van Dam, C. P., "Practical Design Optimization of Wing/Body Configurations Using the Euler Equations," AIAA Paper 92-2633-CP, 1992.

[84] Cliff, S., Reuther, J., Baker, T., Mann, M. and Hicks, R., "Supersonic Transport Wing Design by Numerical Optimization with Superimposed Nacelle Pressures," First NASA/Industry High Speed Research Propulsion/Airframe Integration Workshop, Cleveland, October 1993.

[85] Venkataraman, P., "Optimum Airfoil Design in Viscous Flows," AIAA Paper 95-1876-CP, 1995.

[86] Reneaux, J., and Thibert, J-J., "The Use of Numerical Optimization for Airfoil Design," AIAA Paper 85-5026, October 1985.

[87] Chen, M-S., and Chow, C-Y., "Numerical Design of Transonic Airfoils," *Proceedings of the Sixth International Conference on Numerical Methods in Laminar and Turbulent Flow*, (Swansea), (Eds. C. Taylor et al.), Vol. 6, Part 1, July 1989, pp. 905-915.

[88] Hager, J. O., Eyi, S., and Lee, K. D., "Design Efficiency Evaluation for Transonic Airfoil Optimization: A Case for Navier-Stokes Design," AIAA Paper 93-3112, July 1993.

[89] Greenman, R. M., Cheung, S., and Tu, E. L., "Optimization: A Tool to Study the Physical Features of a Transonic Wing," AIAA Paper 96-2486-CP, 1996.

[90] Eleshaky, M. E., and Baysal, O., "Aerodynamic Shape Optimization Via Sensitivity Analysis on Decomposed Computational Domains," AIAA Paper 92-4698-CP, September 1992.

[91] Eleshaky, M. E., and Baysal, O., "Design of 3-D Nacelle Near Flat-Plate Wing Using Multiblock Sensitivity Analysis (ADOS)," AIAA Paper 94-0160, January 1994.

[92] Lacasse, J. M., and Baysal, O., Design Optimization of Single- and Two-Element Airfoils on Multiblock Grids," AIAA Paper 94-4273, September 1994.

[93] Item, C. C., and Baysal, O., "Wing-Section Optimization for Supersonic Viscous Flow," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 37-44, International Mechanical Engineering Conference and Exposition, November 1995.

[94] Burgreen, G. W., Baysal, O., and Eleshaky, M. E., "Improving the Efficiency of Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 32, No. 1, January 1994, pp. 69-76.

[95] Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," *AIAA Journal*, Vol. 34, No. 9, September 1996, pp. 1761-1770.

[96] Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Supersonic Delta Wings," AIAA Paper 94-4271-CP, September 1994.

[97] Korivi, V. M., Newman, P. A., and Taylor, A. C. III, "Aerodynamic Optimization Studies Using a 3-D Supersonic Euler Code With Efficient Calculation of Sensitivity Derivatives," AIAA Paper 94-4270-CP, September 1994.

[98] Pandya, M. J., and Baysal, O., "Gradient-Based Aerodynamic Shape Optimization Using Alternating Direction Implicit Method," To be Published in *Journal of Aircraft*, Vol. 34, No. 3, May-June 1997.

[99] Pandya, M. J., and Baysal, O., "3D Viscous ADI Algorithm and Strategies for Shape Optimization," AIAA Paper 97-1853, June-July 1997.

[100] Elliott, J. and Peraire, "Practical 3D Aerodynamic Design and Optimization Using Unstructured Meshes," AIAA Paper 96-4170-CP, 1996.

[101] Newman, J. C. III, Taylor, A. C. III, Three-Dimensional Aerodynamic Shape Sensitivity Analysis and Design Optimization Using the Euler Equations on Unstructured Grids," AIAA Paper 96-2464-CP, 1996.

[102] Appel, J. R., Godfrey, A. G., Gunzburger, M. D., and Cliff, E. M., "Optimization-Based Design in High-Speed Flows," *CFD for Design and Optimization* (Ed. O. Baysal), ASME FED-Vol. 232, pp. 61-68, International Mechanical Engineering Conference and Exposition, November 1995.

[103] Jameson, A., Martinelli, L. and Pierce, N., "Optimum Aerodynamic Design Using the Navier-Stokes Equation," AIAA Paper 97-0101, January 1997.

[104] Reuther, J., Saunders, D., Jameson, A., and Alonso, J., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers," AIAA Paper 97-0103, January 1997.

[105] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 97-0643, January 1997.

[106] Rizk, M. H., "Application of the Single-Cycle Optimization Approach to Aerodynamic Design," *AIAA Journal*, Vol. 22, No. 6, June 1985, pp. 509-515.

[107] Ta'asan, S., Kuruvila, G., and Salas, M. D., "Aerodynamic Design and Optimization in One Shot," AIAA Paper 92-0025, January 1992.

[108] Iollo, A., Kuruvila, G., and Ta'asan, S., "Pseudo-Time Method for Optimal Shape Design Using the Euler Equations," ICASE Report 95-59, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, August 1995.

[109] Hou, G. W., Taylor, A. C. III, Mani, S. V., and Newman, P. A., "Formulation for Simultaneous Aerodynamic Analysis and Design Optimization," *International Journal of Numerical Methods in Fluids*, Vol. 37, 1994, pp. 2251-2266.

[110] Küchemann, D., *The Aerodynamic Design of Aircraft*, Pergamon Press, Oxford, 1978.

[111] Brown, C. E., and McLean, F. E., "The Problem of Obtaining High Lift-Drag Ratios at Supersonic Speeds," *Journal of Aerospace Sciences*, Vol. 26, No. 5, May 1959, pp. 298-302.

[112] Busemann, A., "Aerodynamischer Auftrieb bei Überschallgeschwindigkeit" Fifth Volta Conference on High Velocities in Aviation, Rome, September-October 1935.

[113] Rogers, E. W. E., and Hall, I. M., "An Introduction to the Flow about Plane Swept-back Wings at Transonic Speeds," *Journal of the Royal Aeronautical Society*, Vol. 64, No. 596, August 1960, pp. 449-464.

[114] Kulfan, R. M., and Sigalla, A., "Real Flow Limitations in Supersonic Airplane Design," AIAA Paper 78-147, January 1978.

[115] Ghorai, S. C., "Leading-Edge Vortices and Shock-Detachment Flow over Delta Wings," *Journal of Aircraft*, Vol. 6, No. 3, May-June 1969, pp. 228-232.

[116] Miller, D. S., and Wood, R. M., "Lee-Side Flow Over Delta Wings at Supersonic Speeds," NASA TP 2430, 1985.

[117] Carlson, H. W., McElory, M. O., Lessard, W. B., and McCullers, L. A., "Improved Method for Prediction of Attainable Wing Leading-Edge Thrust," NASA TP 3557, April 1996.

[118] Carlson, H. W., and Mann, M. J., "Survey and Analysis of Research on Supersonic Drag-Due-to-Lift Minimization with Recommendations for Wing Design," NASA TP 3202 September 1992.

[119] Pittman, J. L., Miller, D. S., and Mason, W. H., "Supersonic, Nonlinear, Attached-Flow Wing Design for High Lift with Experimental Validation," NASA TP 2336, August 1984.

[120] Wood, R. M., "Supersonic Aerodynamics of Delta Wings," NASA TP 2771, March 1988.

[121] Wood, R. M., and Bauer, S. X. S., "The Natural Flow Wing-Design Concept," NASA TP 3193, May 1992.

[122] Kulfan, R. M., "Application of Favorable Aerodynamic Interference to Supersonic Airplane Design," SAE Paper 90198, October 1990.

[123] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, New York, 1984.

[124] Newsome, R., and Thomas, J. L., "Computation of Leading-Edge Vortex Flows," NASA CP 2416, October 1985.

[125] Thomas, J. L., Taylor, S. L., and Anderson, W. K., "Navier-Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings," AIAA Paper 87-0207, January 1987.

[126] Walters, R. W., and Thomas, J. L., "Advances in Upwind Relaxation Methods," *State of the Art Surveys of Computational Mechanics*, Ed. A.K. Noor, pp. 145-183, ASME Publication, 1989, New York.

[127] Riggins, D. W., Walters, R. W., and Pelletier, D., "The Use of Direct Solvers for Compressible Flow Computations," AIAA Paper 88-0229, January 1988.

[128] Thomas, J. L., Van Leer, B. , and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA Paper 85-1680, July 1985.

[129] Kelley, C. T., and Keyes, D. E., "Convergence Analysis of Pseudo-Transient Continuation," ICASE Report 96-46, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, July 1996.

[130] Venkatakrishnan, V., "Preconditioned Conjugate Gradient Methods for the Compressible Navier-Stokes Equations," AIAA Paper 90-0586, January 1990.

[131] Beam, R. W., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393-402.

[132] Gropp, W. D., Keyes, D. E., and Mounts, J. S., "Implicit Domain Decomposition Algorithms for Steady, Compressible Aerodynamics," Proceedings of the Sixth International Conference on Domain Decomposition Methods (Eds. A. Quarteroni et al.), American Mathematical Society, Providence 1994.

[133] Van Leer, B., "Flux Vector Splitting for the Euler Equations," ICASE Report 82-30, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA, September 1982.

[134] Anderson, W. K., Thomas, J. L., and Van Leer, B., "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal*, Vol. 24, No. 9, September 1986, pp. 1453-1460.

[135] Van Albada, G. D., Van Leer, B., and Walters, R. W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Astronomy and Astrophysics*, Vol. 108, April 1982, pp. 76-84.

[136] Steinbrenner, J. P., Chawner, J. R., and Fouts, C. L., "The GRIDGEN 3D Multiple Block Grid Generation System," WRDC-TR-90-3022, Vol. I and II, Wright-Patterson AFB, Ohio, July 1990.

[137] Jones, W. T., and Samareh, J. A., "A Grid Generation System for Multidisciplinary Design Optimization," *Proceedings of Workshop on Surface Modeling, Grid Generation and Related Issues in CFD Solutions*, NASA-CP 3291, pp. 11-21.

[138] Eleshaky, M. E., "A Computational Aerodynamic Design Optimization Method Using Sensitivity Analysis," Ph. D. Dissertation, Old Dominion University, Norfolk, Virginia, May 1992.

[139] Burgreen, G. W., Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis," Ph. D. Dissertation, Old Dominion University, Norfolk, Virginia, May 1994.

[140] Korivi, V. M., "Aerodynamic Design Optimization with Consistently Discrete Sensitivity Derivatives Via the Incremental Iterative Method," Ph. D. Dissertation, Old Dominion University, Norfolk, Virginia, May 1995.

[141] Pandya, M. J., and Baysal, O., "Gradient-Based Aerodynamic Shape Optimization Using ADI Method for Large-Scale Problems," AIAA Paper 96-0091, January 1996.

[142] Vanderplaats, G. N., and Hicks, R. N., "Numerical Airfoil Optimization Using a Reduced Number of Design Coordinates," NASA TM X-73151, 1976.

[143] Samareh, J. A., "Use of CAD Geometry in MDO," AIAA Paper 96-3991-CP, 1996.

[144] Aidala, P. V., Davis Jr., W. H., and Mason, W. H., "Smart Aerodynamic Optimization," AIAA Paper 83-1863, July 1983.

[145] Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W., *Numerical Grid Generation Foundations and Applications* North-Holland, NY, 1985.

[146] Hartwich, P. M., and Agrawal, S., "Orthonormal Functions for Airfoil and Wing Parameterization," AIAA Paper 96-2419-CP, 1996, pp. 359-369.

[147] Huddleston, D. H., and Mastin, C. W., "Optimization of Aerodynamic Designs Using Computational Fluid Dynamics," AGARD-CP-463, May 1989.

[148] Venkataraman, P., "Optimum Airfoil Design in Viscous Flows," AIAA Paper 95-1876-CP, 1995.

[149] Thomas, A. M., Smith, R. E., and Tiwari, S. N., "Rational B-Spline and PDE Surfacces with Unstructured Grid For Aerospace Vehicle Design," AIAA Paper 94-0419, January 1994.

[150] Mastin, C. W., Smith, R. E., Sadrehaghighi, I., and Wiese, M. R., "Geometric Model for a Parametric Study of the Blended-Wing-Body Airplane," AIAA Paper 96-2416-CP, 1996.

[151] Bloor, M. I. G., and Wilson, M. J., "Efficient Parameterization of Generic Aircraft Geometry," *Journal of Aircraft*, Vol. 32, No. 6, November-December 1995, pp. 1269-1275.

[152] Smith R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. M., "Rapid Airplane Parametric Input Design," AIAA Paper 95-1687, 1995.