Three-Dimensional Aerodynamic Shape Optimization Using Discrete Sensitivity Analysis

Gregory Wayne Burgreen
Old Dominion University

Follow this and additional works at: https://digitalcommons.odu.edu/mae_etds

Part of the Aerospace Engineering Commons, and the Mechanical Engineering Commons

Recommended Citation
https://digitalcommons.odu.edu/mae_etds/215

This Dissertation is brought to you for free and open access by the Mechanical & Aerospace Engineering at ODU Digital Commons. It has been accepted for inclusion in Mechanical & Aerospace Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.
THREE-DIMENSIONAL AERODYNAMIC SHAPE OPTIMIZATION
USING DISCRETE SENSITIVITY ANALYSIS

by

Gregory Wayne Burgreen
B.S., August 1986, The University of Alabama in Huntsville
M.S., December 1987, The University of Alabama in Huntsville

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

DOCTOR OF PHILOSOPHY
MECHANICAL ENGINEERING

Old Dominion University
May 1994

Approved by:

O. Baysal (Director)  A. C. Taylor III

S. N. Tiwari  P. A. Newman

Sobieski

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ABSTRACT

THREE-DIMENSIONAL AERODYNAMIC SHAPE OPTIMIZATION
USING DISCRETE SENSITIVITY ANALYSIS

Gregory W. Burgreen
Old Dominion University

Director: Dr. O. Baysal

An aerodynamic shape optimization procedure based on discrete sensitivity analysis is extended to treat three-dimensional geometries. The function of sensitivity analysis is to directly couple computational fluid dynamics (CFD) with numerical optimization techniques, which facilitates the construction of efficient direct-design methods. The development of a practical three-dimensional design procedures entails many challenges, such as: 1) the demand for significant efficiency improvements over current design methods; 2) a general and flexible three-dimensional surface representation; and 3) the efficient solution of very large systems of linear algebraic equations. It is demonstrated that each of these challenges is overcome by: 1) employing fully implicit (Newton) methods for the CFD analyses; 2) adopting a Bezier-Bernstein polynomial parameterization of two- and three-dimensional surfaces; and 3) using preconditioned conjugate gradient-like linear system solvers. Whereas each of these extensions independently yields an improvement in computational efficiency, the combined effect of implementing all the extensions simultaneously results in a significant factor of 50 decrease in computational time and a factor of eight reduction in memory over the most efficient design strategies in current use. The new aerodynamic shape optimization procedure is demonstrated in the design of both two- and three-dimensional inviscid aerodynamic problems including a two-dimensional
supersonic internal/external nozzle, two-dimensional transonic airfoils (resulting in supercritical shapes), three-dimensional transonic transport wings, and three-dimensional supersonic delta wings. Each design application results in realistic and useful optimized shapes.
ACKNOWLEDGMENTS

The author would like to thank O. Baysal for his consistent financial support during this student's tenure. I also would like to thank my former coworker Dr. M. E. Eleshaky for his many useful technical contributions and discussions during the early stages of my research. The funding for this research was obtained from NASA Langley Research Center under Grant NAG-1-1188. The technical monitor was David S. Miller.

I sincerely thank Dr. J. Sobieski, Prof. S. N. Tiwari, Dr. A. C. Taylor III, and Dr. P. A. Newman for serving on my Dissertation Committee. Their helpful suggestions, professionalism, and genuine interest were greatly appreciated and most refreshing.

I would like to express my gratitude to my closest friends in the Hampton Roads area. My colleague James C. Newman III has been to me an inexhaustible source of encouragement, jocularity, and intellectual stimulation. I wish the best to my other coworkers and friends, especially those who are stoically striving to soon finish. Heartfelt thanks are due to my many dear friends of Immanuel Presbyterian Church, particularly Bill and Debi Harrell, all of whom have immeasurably enriched me during my brief time with them.

Finally, I wish to express my appreciation to my parents and other loved ones at home who have surrounded me with their continual encouragement and untold prayers. I especially thank my father for teaching me during many a hot and dusty day that "there is always a better way" to do things. This invaluable lesson has been one of the guiding principles in my life.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>xii</td>
</tr>
<tr>
<td>Chapter</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Rationale</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Survey of Early Aerodynamic Design Methods</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Some Limitations of Early Aerodynamic Design Methods</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Survey of the Recent Advances in Aerodynamic Design</td>
<td>5</td>
</tr>
<tr>
<td>1.5 Objectives of the Present Work</td>
<td>8</td>
</tr>
<tr>
<td>2. AERODYNAMIC SHAPE OPTIMIZATION</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Inviscid Fluid Dynamic Equation</td>
<td>11</td>
</tr>
<tr>
<td>2.2 Numerical Optimization Technique</td>
<td>12</td>
</tr>
<tr>
<td>2.2.1 Gradient Based Approach</td>
<td>12</td>
</tr>
<tr>
<td>2.2.2 Sensitivity Coefficients</td>
<td>14</td>
</tr>
<tr>
<td>2.3 Discrete Aerodynamic Sensitivity Analysis</td>
<td>15</td>
</tr>
<tr>
<td>2.3.1 Direct Differentiation Formulation</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Adjoint Variable Formulation</td>
<td>16</td>
</tr>
<tr>
<td>2.3.3 Comments on Both Formulations</td>
<td>17</td>
</tr>
<tr>
<td>2.3.4 Approximate Flow Analysis</td>
<td>18</td>
</tr>
<tr>
<td>2.4 Surface Representation</td>
<td>19</td>
</tr>
<tr>
<td>3. IMPLICIT SOLUTION METHODOLOGIES</td>
<td>22</td>
</tr>
<tr>
<td>3.1 Discrete Fluid Dynamic Equation</td>
<td>22</td>
</tr>
<tr>
<td>3.1.1 Alternating Direction Implicit Methods</td>
<td>22</td>
</tr>
<tr>
<td>3.1.2 Fully Implicit Methods</td>
<td>23</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>The Aerodynamic Shape Optimization Procedure</td>
</tr>
<tr>
<td>5.2</td>
<td>Options for Solution Methodologies</td>
</tr>
<tr>
<td>6.1</td>
<td>Proposed Strategies for the Optimization Procedure</td>
</tr>
<tr>
<td>6.2</td>
<td>Computational Statistics for the Supersonic Nozzle Optimization</td>
</tr>
<tr>
<td>6.3</td>
<td>Computational Statistics for the Transonic Airfoil Optimization</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Approximate flow analysis of Mach 3 flow past a flat plate with a 10-deg surface deflection</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>Three-dimensional Bezier-Bernstein representation of a wing upper-surface</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Spatial adaptation of a typical surface-normal grid line</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>The “unit wing” geometry</td>
<td>44</td>
</tr>
<tr>
<td>4.4</td>
<td>Parameter distributions and their resulting half-planform shapes</td>
<td>45</td>
</tr>
<tr>
<td>5.1</td>
<td>Convergence histories of three-dimensional CFD analysis using various relaxation strategies</td>
<td>57</td>
</tr>
<tr>
<td>6.1</td>
<td>Formulation of the nozzle shape optimization problem</td>
<td>83</td>
</tr>
<tr>
<td>6.2</td>
<td>Convergence history of Newton’s method for the nozzle optimization</td>
<td>84</td>
</tr>
<tr>
<td>6.3</td>
<td>Objective function history for the nozzle optimization</td>
<td>84</td>
</tr>
<tr>
<td>6.4</td>
<td>A comparison of the optimized nozzle designs</td>
<td>86</td>
</tr>
<tr>
<td>6.5</td>
<td>Evolution of the optimized nozzle shape</td>
<td>86</td>
</tr>
<tr>
<td>6.6</td>
<td>History of strategy 2 design variable values for the nozzle optimization</td>
<td>87</td>
</tr>
<tr>
<td>6.7</td>
<td>Formulation of the airfoil optimization problem</td>
<td>88</td>
</tr>
<tr>
<td>6.8</td>
<td>Final optimized airfoil shape and Cp distribution for Mach 0.75</td>
<td>89</td>
</tr>
<tr>
<td>6.9</td>
<td>Evolution of the optimized Mach 0.75 airfoil design for strategy 6 and ( \Delta \alpha = 0.0001 )</td>
<td>90</td>
</tr>
<tr>
<td>6.10</td>
<td>A comparison of strategy 6 optimized Mach 0.75 airfoil designs for various ( \Delta \alpha ) step sizes</td>
<td>91</td>
</tr>
<tr>
<td>6.11</td>
<td>A comparison of typical CFD convergence histories for two different Newton method solution strategies</td>
<td>92</td>
</tr>
<tr>
<td>6.12</td>
<td>Histories of the objective function and aerodynamic constraints during the Mach 0.75 airfoil optimization process</td>
<td>93</td>
</tr>
</tbody>
</table>
6.13 A comparison of optimized airfoil designs for three different design Mach numbers. .................................................. 94
6.14 Evolution of the optimized airfoil design for Mach 0.80. .............................................. 95
6.15 An optimized flexible wing: \( M_\infty = 0.75, \alpha = 3.0\text{-deg.} \) ............................................. 96
6.16 Parameter distributions for the optimized flexible wing. ........................................... 97
6.17 Optimized design of a transport wing: \( M_\infty = 0.75, \alpha = 3.0\text{-deg.} \) ...................... 98
6.18 Surface pressure contours and \( C_p \) distributions for the transport wing design: \( M_\infty = 0.75, \alpha = 3.0\text{-deg.} \) ....................................................................................... 99
6.19 History of the aerodynamic coefficients for the Mach 0.75 transport wing optimization. .......................................................... 100
6.20 Evolution of the planform shape during the Mach 0.75 transport wing optimization. ........................................................................ 101
6.21 Performance curves for three transonic wing designs. .............................................. 102
6.22 Initial supersonic delta wing geometry. ................................................................. 103
6.23 Optimized asymmetric supersonic delta wing: \( M_\infty = 1.62, \alpha = 3\text{-deg.} \) .......... 105
6.24 Optimized cranked supersonic delta wing: \( M_\infty = 1.5, \alpha = 3\text{-deg.} \) ............... 107
6.25 Optimized cranked supersonic delta wing with mid-span chord constraint: \( M_\infty = 1.5, \alpha = 3\text{-deg.} \) ................................................................. 109

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
LIST OF SYMBOLS

A wing section area
A generic coefficient matrix
\(a_{rc}\) vector of arclength distributions
B Bernstein polynomial
b generic right-hand-side vector
C preconditioning matrix
\(C_D, G_D\) coefficient of drag
\(C_L, G_L\) coefficient of lift
Cp coefficient of pressure
\(D, \mathbf{d}\) vector of design variables
\(\hat{e}\) unit basis vector
\(\hat{\mathbf{F}}, \hat{\mathbf{G}}, \hat{\mathbf{H}}\) inviscid flux vectors in generalized coordinates
F objective function
f projected normalized distribution function, general function
G aerodynamic constraint
H geometric constraint
I identity matrix
ILU incomplete LU decomposition
\(k\) discrete computational index
k GMRES search directions
L arclength value
\(L_2\)-norm Euclidean norm of a vector
LHS left-hand-side vector
LU  lower/upper decomposition

$M$ coordinate transformation metrics, Mach number

$m$ GMRES restart cycles

$NDV$ number of design variables

$NCONQ$ number of aerodynamic constraints

$NCOND$ number of geometric constraints

$O(*)$ order of accuracy

$P$ vector of Bezier control points, pressure

$Q$ vector of conserved flow variables

$R$ CFD residual

$r$ residual update vector

$\text{RHS}$ right-hand-side vector

$S$ $NDV$-dimensional search direction

$S_2, S_3$ Bezier-Bernstein surfaces

$T$ coefficient of thrust

$t/c$ thickness-to-chord ratio

$u, v$ Bezier-Bernstein computational arclengths

$V$ wing volume

$X$ three-dimensional vector of grid points

$x, y, z$ physical space coordinate directions

$x, y, z$ vector of discrete grid points

$y$ GMRES update vector

$z$ generic vector of unknowns

$\vec{v}$ vector quantity
**Subscripts**

- $b$: grid boundary nodes
- $c, r$: conditions at cowl and ramp, respectively
- $n$: one-dimensional search iteration index
- $n, N, m, M$: order of Bernstein polynomial
- $0$: initial condition
- $TE$: trailing-edge
- $x, y, z$: physical space coordinates
- $\infty$: freestream conditions
- $*$: indicates best design

**Superscripts**

- $L, U$: lower and upper bounds, respectively
- $m$: iteration index
- $n$: CFD time level
- $T$: transpose
- $-1$: matrix inverse
- $*$: critical $C_p$
- $\circ$: degrees

**Greek Symbols**

- $\alpha$: scalar move parameter, angle of attack
- $\alpha, \beta, \omega, \eta$: coefficients computed by conjugate gradient algorithm
- $\beta$: trailing-edge deflection angle
- $\Delta$: delta-difference operator
- $\delta$: numerical difference operator, incremental correction operator
- $\xi, \eta, \zeta$: generalized curvilinear coordinate directions
\( \lambda \)  adjoint vector
\( v \)  Krylov subspace vector
\( \theta \)  trailing-edge included angle
\( \tau \)  computational time scale
\( \omega \)  scalar relaxation parameter
\( \nabla \)  gradient operator
\( \| \)  vector norm
\( \infty \)  infinity
Chapter 1

INTRODUCTION

1.1 Rationale

The ingenuity and tenacity of aerodynamic designers have served them well over the history of aeronautics. Near-optimal designs for many aerodynamic components have been produced with the investment of many hours of analysis and experimental testing. An excellent example of this is the development of the supercritical airfoil shape by Whitcomb [1] in the 1960's. His superior insight and understanding of flow physics, in conjunction with extensive wind tunnel testing, guided the systematic evolution of this radically new and efficient airfoil shape.

Over the years, this heuristic type of approach has been successfully employed in the design of many complex configurations including complete aircraft whose behavior is a resultant of complex interactions among many different physical phenomena and hardware components. However, with the advent of advanced computers, computational analysis has become an invaluable tool to guide design decisions, and more recently formal optimization methods are increasingly being used as tools for determining the values of design variables [2].

In the last decade, an emerging trend in the analytical design of complex engineering systems is the integration of all appropriate disciplines in the design process. This new discipline is referred to as Multidisciplinary Design Optimization (MDO) [3]. One of the principle components of MDO is sensitivity analysis, which quantifies the sensitivity of the system outputs to design changes. Intrinsic to the future success of MDO applications is the maturation of sensitivity analysis-based optimization procedures within the
individual engineering disciplines [4]. The present work concentrates on one such disciplinary design topic—aerodynamic shape optimization.

A promising new method for aerodynamic shape optimization has recently emerged. The unique feature of this design method is its use of discrete aerodynamic sensitivity analysis. The function of sensitivity analysis is to directly couple computational fluid dynamics (CFD) with numerical optimization techniques, which facilitates the development of efficient direct-design procedures. Such procedures have the capability to automatically determine optimal geometric surfaces that are not biased by intuition or experience in engineering [5].

1.2 Survey of Early Aerodynamic Design Methods

Aerodynamic direct-design methods based on numerical optimization techniques have been around for a number of years. These methods directly extremize some measure of merit (i.e., an objective function) for a given design problem. Direct-design methods should not be confused with inverse-design methods, e.g., [6–8], which determine the geometric shape that best matches a prescribed target distribution of some aerodynamic quantity (e.g., pressure, Mach number, etc.).

Aerodynamic designers focused their earliest computational efforts on the design of two-dimensional airfoils. One of the first applications of aerodynamic optimization was presented in 1965 by Schmit and Thorton [9] for the optimization of a supersonic double-wedge airfoil. Vanderplaats et al. later introduced methodologies to improve both the computational efficiency [10] and geometric modeling flexibility [11] of airfoil optimization. Several application papers using this basic methodology followed thereafter by different researchers, e.g., [12]. However, the state-of-the-art for the numerical design of airfoils was not significantly advanced until recently.

Like airfoil design, the direct-design of three-dimensional wings using CFD and numerical optimization techniques has only moderately evolved since first introduced in
1977. In that year, Hicks and Henne [13] extended the widely successful airfoil design techniques of Vanderplaats to perform 3D wing design. Designers have subsequently applied this same basic approach for wing design for many years [14–16].

Some early examples of direct-design of supersonic airfoils and wings using numerical optimization techniques are reported in Refs. [17–19].

The early aerodynamic design methods may be clearly identified by noting their common distinctives, which include: 1) the potential equation is solved to predict the flow physics; 2) a finite-difference approach is used to compute the sensitivity information for the gradient-based optimization codes; and 3) their surface modeling capabilities are not very general. Let us define an "early design methodology" as one having these distinctives.

1.3 Some Limitations of Early Aerodynamic Design Methods

One of the critical issues of aerodynamic design has always been the high computational costs involved. The bulk of these costs is manifested as the CPU time invested in computing numerous steady flow solutions. The majority of these flow solutions are apportioned toward the evaluation of the aerodynamic performance characteristics of the intermediate designs (e.g., $C_L$, $C_D$, etc.). The remaining flow solutions go toward the evaluation of the required optimization gradient information. Traditionally, this gradient information has been numerically determined by finite-difference approximations.

The early aerodynamic design methods exclusively relied on solving the potential flow equation to obtain flow solutions. This practice permitted design at reasonable costs due to the inexpensiveness of the flow analyses. However, potential flow methods are limited by an inherent inability to correctly predict strong shocks and do not allow for distributed vorticity fields [20]. In light of recent advances in computer technology and in numerical algorithms [21,22], the more accurate fluid dynamic models based on the
Euler and Navier-Stokes equations should be incorporated within aerodynamic design optimization.

Both the performance and generality of direct-design methods are seriously impacted by the use of a finite-difference approach to compute the sensitivity information. Slooff [23] correctly identified the finite-difference approach as the weakest point of a numerical optimization design strategy. In particular, the finite-difference approach imposes a severe limitation on the number of design variables in order to keep the computational effort within reasonable bounds. In addition, this approach has the potential drawback of unwittingly introducing numerical noise into the sensitivity gradients, which would lead to erroneous optimization search directions.

One important aspect of aerodynamic shape optimization is the representation of the surface to be optimized. In design applications one of the following approaches is typically used: 1) analytical definition of the shape [24,25]; 2) local perturbation of a baseline geometry by means of the superposition of a set of weighted basis shapes [13]; or 3) definition of a new composite shape via a linear combination of weighted basis shapes [11]. These basis shapes may consist of polynomial functions, orthogonal functions, "aeroshape" functions [26], or spline functions. The manner in which all of these approaches have been applied in most aerodynamic design problems to date have modeled only limited regions of the geometry and/or suffered from a lack of generality for representing geometries that are not common in the aerodynamic community. For instance, the basis shapes or aeroshape functions to be used for, say, designing a scramjet-afterbody nozzle are not readily obvious or available without additional research or expertise. The analytical approach is limited in that many complex geometric shapes do not easily yield to analytical descriptions; furthermore, if an analytical description is obtained, the resulting surfaces are constrained to a certain class of shapes.

In view of the limitations of the early aerodynamic design methods, significant advancements are needed before three-dimensional direct-design procedures can be
considered as practical design tools. Fortunately, much innovative research is currently being done in the area of computational aerodynamic analysis and design. It is imperative that the next generation of design procedures incorporate the most promising concepts from this ongoing research.

1.4 Survey of Recent Advances in Aerodynamic Design

The latest contributions in aerodynamic design may be systematically reviewed by examining how each of the limitations of the early aerodynamic design methods have been addressed. The incorporation of the more accurate fluid dynamic models into the "early design methodology" has recently been achieved by several researchers [20, 27-29]. Some aerodynamic design applications that include substantial improvements to the geometry representations of the "early design methodologies" may be found in Refs. [30-32]. Huddleston and Mastin [33] present a design procedure that is based on an Euler/Navier-Stokes solver and, in addition, allows for very general two- and three-dimensional surface modifications. All of these works, however, still rely on a finite-difference approach for obtaining gradient information.

Many problems associated with obtaining the gradient information within a direct-design method may be alleviated by using zero-th order optimization methods, which would eliminate the explicit need for gradients. These approaches require only function evaluations to construct the information necessary to effectively move the design toward an optimum. Such design methods have been successfully demonstrated in Refs. [34-36].

Currently many research efforts are being directed toward the analytical determination of the optimization gradient information. This rapidly maturing technology is referred to as aerodynamic sensitivity analysis. Sensitivity analysis adopts either a variational (i.e., continuous) approach or a discrete approach; these approaches differ in
that the order of discretizing the continuous problem and of applying calculus are interchanged [37].

Variational sensitivity analysis develops a set of "adjoint" equations by applying the fundamental principles of calculus of variations to the continuous governing equations of the design problem. The adjoint equations are then discretized and solved in order to eventually obtain the direction and magnitude of change (i.e., gradient information) of the objective function to be supplied to the optimization procedure. Aerodynamic design methods based on variational methods have been reported in Refs. [38–44]. Borggaard et al. [45] proposed a unique approach in which the continuous Euler equations are directly differentiated with respect to design variables and then solved in discrete form for the unknown flowfield derivatives. To date, no papers have been published on variational aerodynamic sensitivity analysis for three-dimensional flow problems.

Discrete sensitivity analysis differentiates the discrete form of the governing equations of the design problem according to the Implicit Function Theorem. The resulting discrete "sensitivity equation" is linear in the unknown derivatives and constitutes a linear algebraic system of equations. This area has received much attention lately; an excellent review paper on the subject was recently given by Taylor et al. [46]. The first effort in this area is due to Bristow and Hawk [47] and was developed for subsonic panel methods. Subsequently, Yates [48] derived a formulation for lifting surface theory, which was valid for all compressible flow speed regimes. Elbanna and Carlson [49] were the first to compute the sensitivity coefficients for the two-dimensional full potential equation. Drela [50] performed sensitivity analysis on the quasi-two-dimensional ("streamline-based") Euler equations. Baysal and Eleshaky [51] later demonstrated a method for treating the two-dimensional Euler equations written in a generalized coordinate system (i.e., on fixed grids). Taylor et al. [52] first computed sensitivity derivatives for the laminar thin-layer two-dimensional Navier-Stokes equations. Eleshaky and Baysal [53,54] developed a procedure for performing discrete
sensitivity analysis on multi-block grids. Korivi et al. [55] outlined an "incremental iterative" solution strategy for sensitivity analysis. Lorence and Hall [56] recently computed the unsteady sensitivity coefficients for the two-dimensional Euler equations. Finally, recent interesting developments include those of obtaining sensitivity derivatives via symbolic manipulation [57] and automatic differentiation [58,59] of FORTRAN.

The primary difficulty in extending discrete sensitivity analysis procedures to handle large two- and three-dimensional problems is the efficient solution of the resulting large linear system of equations. For these large cases, the memory requirements for direct linear solvers based on Gaussian-elimination decompositions are prohibitive. Some recent approaches specifically developed to attack these larger problems are reported in Refs. [57, 60, and 61].

Integration of the new discrete sensitivity analysis technology into a functional aerodynamic shape optimization procedure has only recently been accomplished by a few groups. These groups include for two-dimensional design: Drela [50,62], Baysal et al. [54,63–67], Taylor et al. [52,68], Grossman et al. [27], Ghattas et al. [69], and Young et al. [70]; and for three-dimensional design: Grossman et al. [71,72], and Baysal et al. [73,74]. The results from these applications have indicated that in general the design methods obtain a final optimum aerodynamic shape via an evolution of successively improved shapes, although sometimes at a rather high computational cost due to the large number of flow analyses involved.

For maximum geometric flexibility in representing design surfaces, each surface grid point may be treated as a geometric design variable. However, this approach is somewhat computationally expensive in practice. Generally, it is always desirable to reduce the number of design variables, yet retain the capability to accurately represent a wide range of complex surface shapes. Toward this end, parameterizations of the design surface using Legendre polynomials [35], Bezier-Bernstein polynomials [30,33,66,73–75] and
NURBS (Non-Uniform Rational B-splines) [76,77] have been used with much success recently.

1.5 Objectives of the Present Work

The development of a practical three-dimensional design procedure entails many challenges such as: 1) the demand for significant efficiency improvements over previous design methods; 2) a general and flexible three-dimensional surface representation; and 3) the efficient solution of very large systems of linear algebraic equations. In the present work, all of these challenges will be addressed, and novel methods that successfully overcome each challenge will be demonstrated.

The primary means proposed to reduce the high costs of aerodynamic design is two-fold. First, discrete sensitivity analysis is used to compute the optimization gradient information. Second, a truly practical use of Newton's method is introduced within the optimization procedure. Only a few examples of three-dimensional aerodynamic sensitivity analysis have been published. This work represents one of the first efforts to successfully integrate this new technology into a three-dimensional direct-design procedure. The main objective of the present work is to develop an efficient and functional three-dimensional Aerodynamic Shape Optimization Procedure (AeSOP). This effort has resulted in the development of the direct-design codes, AeSOP2D and AeSOP3D, and has been reported in Refs. [66, 67, and 73].
Chapter 2
AERODYNAMIC SHAPE OPTIMIZATION

The aim of aerodynamic design optimization is the minimization of an objective function $F$ subject to constraints $G$ and $H$. Both the objective function and constraints may be nonlinear functions of the design variables $D$ and the flowfield variables $Q$.

The constrained aerodynamic optimization problem may be mathematically formulated as

$$\text{minimize } F(D, Q(D))$$

subject to inequality constraints,

$$G_j(D, Q(D)) \leq 0 \quad j = 1, \ldots, NCONQ$$

$$H_k(D) \leq 0 \quad k = 1, \ldots, NCOND$$

and to side constraints,

$$D_i^L \leq D_i \leq D_i^U \quad i = 1, \ldots, NDV$$

where $NCONQ$ and $NCOND$ are the number of aerodynamic and geometric inequality constraints, respectively.

The unique feature of aerodynamic design optimization is that the fluid dynamic flowfield plays an integral role in the optimization problem. To illustrate this, the components of an aerodynamic optimization problem are now briefly described with particular emphasis placed on each one’s interrelationship with the aerodynamic flowfield.

The design variables $D$ is a vector of independent variables that dictate the design configuration. Optimum designs are sought within an $NDV$-dimensional design space,
where $NDV$ is the number of design variables. For aerodynamic shape optimization, the design variables are of geometric-type. That is, they describe the geometric shape of the aerodynamic configuration and, hence, influence the aerodynamic flowfield through the surface boundary conditions. Mathematically, this implicit dependence is denoted as $Q = Q(D)$, where $Q$ is the vector of conserved variables representing the flowfield solution.

For aerodynamic design optimization, the objective function $F(Q,D)$ is a mathematical function of both the design variables and the flowfield solution. Because of its dependence on the flowfield solution $Q$, which is governed by a set of highly nonlinear equations, the objective function is typically nonlinear. For maximization of the objective function, $-F(Q,D)$ is minimized.

During the design process, upper or lower limits on various quantities are imposed by means of constraints. The aerodynamic inequality constraints $G(Q,D)$ restrict quantities that are functionally dependent upon the flowfield variables (e.g., magnitudes of force coefficients or actual flow variable values). The geometric inequality constraints $H(D)$ place limits on quantities that depend only upon the geometric-type design variables (e.g., geometric thicknesses, angles, or curvatures). The side constraints limit the allowable values of the design variables within lower and upper bounds, $D^L$ and $D^U$.

Several common elements may be found in every computational aerodynamic shape optimization procedure. These include: 1) a CFD solver based on an appropriately chosen set of fluid dynamic equations; 2) a numerical optimization technique; and 3) a procedure for modifying the surface geometry. An additional element may be an efficient sensitivity analysis procedure to compute the optimization gradient information (if necessary) and thereby directly couple CFD and a numerical optimization technique. These elements, as used in the present work, are now described.
2.1 Inviscid Fluid Dynamic Equation

The governing equations for three-dimensional compressible inviscid flow may be written as

$$\frac{\partial Q}{\partial t} = -R(Q, M)$$  \hspace{1cm} (2.5)

where the steady-state residual vector is

$$R(Q, M) = \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta}$$  \hspace{1cm} (2.6)

$\hat{F}$, $\hat{G}$, and $\hat{H}$ are the inviscid flux vectors in generalized coordinates. The residual $R$ represents the net balance of mass, momentum, and energy across the domain. For steady flow, the residual $R$ is equal to zero [since the unsteady term of Eq. (2.5) vanishes].

Application of the Euler implicit formulation to the unsteady term and linearization of the inviscid flux vectors in time yield the discrete fully implicit formulation of Eq. (2.5)

$$\left[ \frac{I}{\Delta t} + \frac{\partial R}{\partial Q} \right]^{n} \Delta Q^{n} = -R\left(Q^{n}, M\right)$$  \hspace{1cm} (2.7)

where

$$\frac{\partial R}{\partial Q} = \delta^{t}_{\xi} \left( \frac{\partial \hat{F}}{\partial Q} \right)^{n} + \delta^{t}_{\eta} \left( \frac{\partial \hat{G}}{\partial Q} \right)^{n} + \delta^{t}_{\zeta} \left( \frac{\partial \hat{H}}{\partial Q} \right)^{n}$$  \hspace{1cm} (2.8a)

$$\Delta Q^{n} = Q^{n+1} - Q^{n}$$  \hspace{1cm} (2.8b)

The superscript $n$ denotes the time level, and $\delta$ represents a numerical difference operator. In this study, Eq. (2.7) is discretized in space using a cell-centered control volume formulation. The geometric information of the cell interfaces is determined from the coordinate transformation metrics $M$, which involve the transformation from physical space $\{x, y, z\}$ to generalized curvilinear computational space $\{\xi, \eta, \zeta\}$. The inviscid flux vectors and the Jacobian matrix $\partial R/\partial Q$ are evaluated using the flux-vector-splitting technique of Van Leer [78]. The cell interface $Q$ values are determined using a spatially second-order accurate upwind biased MUSCL interpolation with the optional inclusion of
Van Albada flux limiting [79,80]. The numerical boundary conditions are consistently linearized and implicitly treated in Eq. (2.7) [81]. Details of the derivation of the fully implicit fluid dynamic equation are given in Appendix A.

Throughout this work, analytical derivatives are always used for the Jacobian elements, i.e., the true Jacobian matrix is used. This is opposed to an approximate linearization or a finite-difference evaluation of the Jacobian matrix, both of which have been successfully used in CFD applications [82,83]. The unfactored Jacobian \( \frac{\partial R}{\partial Q} \) is a large sparse unsymmetric matrix having nine 4x4 block diagonals for two-dimensional applications or thirteen 5x5 block diagonals for three-dimensional applications.

### 2.2 Numerical Optimization Technique

#### 2.2.1 Gradient-Based Approach

The optimization algorithm employed in this work is the Method of Feasible Directions as applied by Vanderplaats and Moses [84]. This numerical search technique requires the first-order sensitivity gradient information \( \nabla F, \nabla G, \nabla H \) and, hence, is referred to as a gradient-based approach.

One of the most basic operations in numerical optimization techniques is the local minimization of the objective function via a systematic search that requires numerous function evaluations, i.e., numerous flowfield solutions. This search is termed a one-dimensional search. New design points are obtained by iteratively updating the vector of design variables via

\[
D_n^m = D_n^{m-1} + \alpha_n S^m
\]

Here \( m \) is the design iteration number, \( n \) is the one-dimensional search iteration number, the subscript \( * \) denotes the best design of the previous design iteration, \( S \) is an \( NDV \)-dimensional search direction, and \( \alpha \) is a scalar move parameter for determining the amount of change in \( D \).
An optimization design iteration consists of two major steps [5]. The first is the determination of the search direction $S^m$, and the second is the determination of the scalar move parameter $\alpha^*$ that will minimize $F$ as much as possible in direction $S$.

In the Method of Feasible Directions, the search direction is always in the so-called usable-feasible direction within the design space. To determine this direction, a subproblem must be solved which satisfies certain conditions [84]. Namely, the usable directions must satisfy

$$\nabla F(D, Q) \cdot S \leq 0$$  \hspace{1cm} (2.10)

and the feasible directions are governed by the satisfaction of

$$\nabla G_j(D, Q) \cdot S \leq 0 \hspace{1cm} j \in 1, \ldots, NCONQ_{active}$$  \hspace{1cm} (2.11)

and

$$\nabla H_k(D) \cdot S \leq 0 \hspace{1cm} k \in 1, \ldots, NCOND_{active}$$  \hspace{1cm} (2.12)

Hence, the search direction $S$ is determined such that: 1) the design will first be directed into a feasible design space; and 2) any feasible designs will be strictly directed toward lower objective functions. (A feasible design space is that region in which every design point satisfies all constraints.) If during the one-dimensional search the design encounters a constraint or the objective function increases, the current design iteration terminates. Then, assuming that the optimization convergence criteria has not been satisfied, a new search direction $S^{m+1}$ is computed, and the optimization proceeds in the newly computed direction.

In Eq. (2.9), the scalar move parameter $\alpha$ is incremented in the $n$-th one-dimensional search iteration by

$$\alpha_n = \alpha_{n-1} + \Delta \alpha_n$$  \hspace{1cm} (2.13)

where $\alpha_0 = 0$. It is common practice to adopt a sophisticated one-dimensional search strategy such as the golden section method or a polynomial approximation [5]. These methods take several large $\Delta \alpha$ steps to either systematically bracket the local minimum.
of the objective function or to mathematically estimate the local minimum using only sparse information. However, in the present work, an alternate strategy is employed—the \( \Delta \alpha \) step size is held constant throughout the optimization process. This simple strategy is chosen for two reasons. First, this approach imposes a uniform means of traversing the design space and hence provides the basis for consistently evaluating the many different optimization strategies investigated in this work. Second, for highly “nonlinear” aerodynamic design problems, the fluid dynamics is sometimes very sensitive to the surface shape (e.g., consider the location and strength of the normal shock of a transonic airfoil). Relatively small design deviations may considerably alter the flowfield, which in turn would directly affect the computational expense of the subsequent CFD analysis. The present strategy permits explicit control over the magnitude of each design deviation. Finally, it is recognized that for “linear” aerodynamic design problems the present strategy would most likely adversely impact the overall efficiency of the optimization process.

2.2.2 Sensitivity Coefficients

The gradients of the objective function, \( \nabla F \), and the constraints, \( \nabla G \) and \( \nabla H \), are referred to as the sensitivity coefficients. These gradients may be evaluated by finite-differences. However, this simple approach has serious drawbacks. First, it may produce, at times, highly inaccurate gradient approximations due to uncertainties in choosing the proper finite-difference step size. Its accuracy deteriorates with the step size in nonlinear problems, but making the step size too small may incur excessive truncation errors [3]. Second, finite-differencing is computationally expensive, requiring \( NDV + 1 \) steady flow solutions (i.e., \( NDV + 1 \) CFD analyses) for the evaluation of the sensitivity coefficients. Hence, the cost of finite-differencing grows linearly with the number of design variables and becomes prohibitive for large problems.
Alternatively, the sensitivity coefficients may be determined analytically by

\[ \nabla F = \frac{\partial F(D,Q)}{\partial D_i} \hat{e}_i = \left[ \left( \frac{\partial F}{\partial D_i} \right)_Q + \left( \frac{\partial F}{\partial D_i} \right)_D \right] \hat{e}_i \quad i \in 1,\ldots,NDV \quad (2.14) \]

\[ \nabla G_j = \frac{\partial G_j(D,Q)}{\partial D_i} \hat{e}_i = \left[ \left( \frac{\partial G_j}{\partial D_i} \right)_Q + \left( \frac{\partial G_j}{\partial D_i} \right)_D \right] \hat{e}_i \quad j \in 1,\ldots,NCONQ \quad (2.15) \]

\[ \nabla H_k = \frac{dH_k(D)}{dD_i} \hat{e}_i \quad k \in 1,\ldots,NCOND \quad (2.16) \]

### 2.3 Discrete Aerodynamic Sensitivity Analysis

The analytical evaluation of the sensitivity coefficients within the aerodynamic design frame has been the subject of much research recently. In particular, sensitivity analysis procedures have been developed to compute these coefficients. The object of aerodynamic sensitivity analysis is the efficient and accurate calculation of the sensitivity coefficients, which is imperative for practical aerodynamic design at reasonable costs.

Modern CFD methods provide the numerical foundations upon which the present discrete sensitivity analysis procedure is based. It is now shown how aerodynamic sensitivity analysis is derived from these CFD foundations.

For aerodynamic shape optimization, all of the design variables \( D \) are of geometric-type. Consequently, the computational grid and its coordinate transformation metrics will assume the functional forms \( X = X(D) \) and \( M = M\{X(D)\} \), where \( X \) represents the three-dimensional vector of grid points making up the computational mesh in the physical plane, that is, \( X = \{x, y, z\} \). Furthermore, because of the implicit dependence of the flowfield \( Q \) upon the design variables \( D \), the discrete residual \( R \) of Eq. (2.6) takes the following implicit functional form

\[ R(Q,M) = R\left[Q(D), M\{X(D)\}\right] \quad (2.17) \]

This implicit relationship, which has its genesis in CFD, forms the basis of aerodynamic sensitivity analysis.
In the discrete sensitivity analysis approach, one of two formulations can be used to compute the sensitivity coefficients.

2.3.1 Direct Differentiation Formulation

If the flow derivatives \( \frac{\partial Q}{\partial D} \) can be obtained, the sensitivity coefficients may be easily computed from Eqs. (2.13) to (2.15). One approach for computing the flow derivatives is based on the Implicit Function Theorem and is developed as follows. Given a steady-state flow solution, the residual \( R \) is equal to zero and may be analytically differentiated with respect to the design variables to give the sensitivity equation

\[
\frac{dR(Q,M)}{dD_i} = \left( \frac{\partial R}{\partial Q} \right) \frac{\partial Q}{\partial D_i} + \left( \frac{\partial R}{\partial M} \right) \frac{\partial M}{\partial X} \frac{dX}{dD_i} = 0 \quad i \in 1,\ldots,NDV
\]  

or

\[
\frac{\partial R}{\partial Q} \frac{\partial Q}{\partial D_i} = -\frac{\partial R}{\partial M} \frac{\partial M}{\partial X} \frac{dX}{dD_i} \quad i \in 1,\ldots,NDV
\]

where \( dX/dD \) are the grid sensitivity terms. The flow derivatives \( \partial Q/\partial D \) may be directly obtained from Eq. (2.19). This linear system must be solved for \( NDV \) right-hand-side (RHS) vectors, which corresponds to \( NDV \) design variables. Details of the derivation of Eq. (2.19) are given in Appendix B.

2.3.2 Adjoint Variable Formulation

Alternatively the sensitivity coefficients may be obtained using the adjoint variable formulation, which begins by substituting Eq. (2.19) into Eqs. (2.14) and (2.15). The resulting adjoint vectors may be conveniently defined

\[
\lambda_T^F = \left( \frac{\partial F}{\partial Q} \right)^T \left( \frac{\partial R}{\partial Q} \right)^{-1}
\]

\[
\lambda_{Tj}^F = \left( \frac{\partial G_j}{\partial Q} \right)^T \left( \frac{\partial R}{\partial Q} \right)^{-1} \quad j \in 1,\ldots,NCONQ
\]

\[
\lambda_{Tj}^G = \left( \frac{\partial G_j}{\partial Q} \right)^T \left( \frac{\partial R}{\partial Q} \right)^{-1} \quad j \in 1,\ldots,NCONQ
\]

The set of adjoint equations is thus obtained and is given by...
where $\frac{\partial F}{\partial Q}$ and $\frac{\partial G_j}{\partial Q}$ are column vectors defining the partial derivatives of the objective function and the aerodynamic inequality constraints with respect to the flowfield variables. The number of linear systems to be solved is $N_{\text{CONQ}} + 1$. The sensitivity coefficients may then be obtained by

$$
\nabla F_i = \frac{\partial F}{\partial D_i} \hat{e}_i = \left[ \left( \frac{\partial F}{\partial D_i} \right)_{Q} - \lambda_F^T \frac{\partial R}{\partial M} \frac{\partial M}{\partial X} dX \right] \hat{e}_i \quad i \in 1, \ldots, NDV
$$

(2.24)

$$
\nabla G_j = \frac{\partial G_j}{\partial D_i} \hat{e}_i = \left[ \left( \frac{\partial G_j}{\partial D_i} \right)_{Q} - \lambda_{G_j}^T \frac{\partial R}{\partial M} \frac{\partial M}{\partial X} dX \right] \hat{e}_i \quad j \in 1, \ldots, N_{\text{CONQ}}
$$

(2.25)

$$
\nabla H_k = \frac{dH_k(D)}{dD_i} \hat{e}_i \quad k \in 1, \ldots, N_{\text{COND}}
$$

(2.26)

2.3.3. Comments on Both Formulations

In both formulations, $\frac{\partial R}{\partial Q}$ is the Jacobian matrix of the residual vector $R$ and is identical to the true Jacobian matrix of the fully implicit formulation of the fluid dynamic equation. For residual vectors not amenable to analytical hand-differentiation (e.g., those incorporating turbulence models, complex flux formulations, etc.), an alternate method such as a finite-difference approach or an automatic differentiation technique may be adopted to obtain the needed Jacobian derivatives.

Note that the sensitivity equations for both formulations are linear in their mathematical nature. Hence, no modifications or approximations can be made to either the Jacobian matrix or the RHS vectors of these equations without compromising their true solutions.

Comparing both formulations, one finds that obtaining the sensitivity coefficients requires the solution of either $NDV$ or $N_{\text{CONQ}} + 1$ linear systems. Since the solution of
one system for either formulation requires approximately the same amount of computational work, the formulation of the optimization problem dictates which method should be used to produce the sensitivity coefficients most efficiently (i.e., whether $NDV > NCONQ + 1$ or vice versa).

2.3.4 Approximate Flow Analysis

A useful by-product of sensitivity analysis is the capability of computing an approximate flowfield solution without resorting to conventional CFD procedures [10,51,64]. This technique is termed approximate flow analysis and, in essence, predicts a flowfield via a single linear approximation of $Q$ about a baseline design point. Approximate flow analysis requires the solution of the direct differentiation formulation of the sensitivity equation.

The technique is described as follows. First, the flow derivatives $\partial Q/\partial D$ at a given design point $D_0 (= D_{0,i}, i = 1, \ldots, NDV)$ are obtained by solving the direct differentiation formulation of the sensitivity equation, Eq. (2.19). The flow solution for a neighboring design point $D_1$ [cf. Eq. (2.9)] may then be approximated by

$$Q(D_1) = Q(D_0) + \sum_{i=1}^{NDV} \frac{\partial Q}{\partial D_i} \bigg|_{D_0} (D_{1,i} - D_{0,i})$$

(2.27)

It has been shown in Ref. 63 that evaluations of nonlinear objective functions and aerodynamic constraints based on $Q(D_1)$ are more accurate than corresponding evaluations based on a direct linear approximations of these functions.

An immediate extension of approximate flow analysis is its generalization to a "multi-level approximation." For example, a second-level flowfield approximation for design point $D_2$ may be obtained by

$$Q(D_2) = Q(D_1) + \sum_{i=1}^{NDV} \frac{\partial Q}{\partial D_i} \bigg|_{D_1} (D_{2,i} - D_{1,i})$$

(2.28)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
It should be noted that $\partial Q/\partial D$ in Eq. (2.28) [or, more accurately, $\partial R/\partial Q$ and $\partial R/\partial X$ in Eq. (2.19)] is based on the approximate flowfield $Q(D)$. This procedure allows flowfield solutions to be progressively "built up" from previous flowfield approximations, all of which have the common genesis of a single initial CFD analysis solution. Thus, starting from an initial design, a flowfield solution for some later design may be obtained through a multi-level approximation that is based on incremental design perturbations. Otherwise, a grossly erroneous flowfield prediction may be produced if an equivalent single large perturbation is attempted.

To demonstrate the effectiveness and accuracy of multi-level approximations, the following case is briefly examined. Consider a flat plate initially in uniform inviscid flow at Mach 3. Based on this initial condition, the flowfield is predicted for a $\theta=10$-deg surface deflection using multi-level approximations and various incremental deflection sizes: namely, 1-, 2.5-, 5-, and 10-deg increments. A comparison of the predicted pressure coefficients distributions along the 10-deg deflected surface along with the CFD solution is shown in Fig. 2.1. As observed from this figure, the results improve progressively with decreasing incremental deflection sizes. Moreover, a flowfield discontinuity (shock) is predicted based on a flowfield that does not initially have that physical phenomenon (shock-free). Thus, a multi-level approximation is shown to accurately predict the flowfield of a largely deformed shape, provided that the final shape is attained through a sequence of sufficiently small incremental shape changes.

2.4 Surface Representation

A critical element in the success of any shape optimization method is the capability to generate a great variety of physically realistic shapes. Ideally, the shape perturbation method should incorporate as much geometric flexibility as possible with as few design variables as possible. This philosophy permits access to a large design space, yet at
minimal computational costs. Due to the importance and lengthiness of this particular topic, it will be fully treated in Chapter 4.
Fig. 2.1 Approximate flow analysis of Mach 3 flow past a flat plate with a 10-deg surface deflection.
Chapter 3
IMPLICIT SOLUTION METHODOLOGIES

The overall computational efficiency of the present three-dimensional design procedure critically depends on the choices of implicit solution methodology used to solve the discrete fluid dynamic equation and the sensitivity equation. This chapter provides a brief overview of the implicit solution methodologies employed in this study.

3.1 Discrete Fluid Dynamic Equation

A typical aerodynamic optimization may require hundreds of evaluations of the objective function, each of which requires an updated aerodynamic flowfield solution. These numerous steady flow solutions constitute the bulk of the total CPU time required for an optimization problem. Hence, efficient CFD solution methodologies are sought to minimize these costs.

Although many efficient CFD solution procedures are built around explicit schemes, the present work examines only the implicit methodologies. Specifically, the schemes considered are the alternating direction implicit (ADI) schemes and the fully implicit methodologies.

3.1.1 Alternating Direction Implicit Methods

The most popular approaches for solving the discrete CFD equation Eq. (2.7) are based upon approximate factorization formulations, in which the left-hand-side (LHS) operator is split into three one-dimensional operators [85]. The resulting equation becomes

\[
\left[ \frac{I}{\Delta t} + \delta_z \left( \frac{\partial \Phi}{\partial Q} \right) \right]^n \left[ \frac{I}{\Delta t} + \delta_n \left( \frac{\partial \Phi}{\partial Q} \right) \right]^n \left[ \frac{I}{\Delta t} + \delta_t \left( \frac{\partial \Phi}{\partial Q} \right) \right]^n \Delta Q^n = -R(Q^n, M) \tag{3.1}
\]

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In the ADI formulation, the presence of the unsteady terms enables the operator-splitting of the unfactored LHS and favorably conditions the diagonals of the resulting ADI-space factored operators.

This spatially-split three-factor approximation of the left-hand-side introduces $O(\Delta t^2)$ factorization error, which strongly affects convergence properties of the scheme. Nevertheless, the ADI solution method is found to be stable and convergent for small-to-moderate time-step sizes. The ADI scheme involves solving a sequence of easily invertible equations. Each equation requires the inversion of a tri- or penta-block diagonal coefficient matrix, either of which can be accomplished very efficiently and requires little computational storage. The primary disadvantage of ADI schemes is their relatively slow rates of convergence to a steady state.

### 3.1.2 Fully Implicit Methods

Much progress has been made in recent years in reducing the time required to obtain steady state solutions to the nonlinear fluid dynamic equations. The most promising means for quickly obtaining steady solutions are techniques based on implicit methodologies that allow high Courant numbers. Recent investigations of unfactored fully implicit algorithms have indicated that significant increases in the speed of convergence to steady state are possible [82]. Newton's method—the most implicit of all methods—is the only technique that provides second-order (quadratic) convergence.

#### 3.1.2.1 Newton's Method

By allowing the time step to approach infinity, the linear system of Eq. (2.7) becomes

$$\frac{\partial R^n}{\partial Q} \Delta Q^n = -R(Q^n, M)$$

(3.2)

Direct solution of this linear system obtains a "numerically exact" solution for $\Delta Q^n$. This scheme is referred to as Newton's method and is known for its high rates of convergence [86,87]. Due to the nonlinear nature of the flowfield equations and the functional...
dependence of $R$ on $Q$, more than one Newton iteration is necessary to drive the residual to zero. If the initial (or a subsequent) flowfield solution is close to the final solution, quadratic convergence to a steady state is obtained. However, if the initial solution is far from the final solution, the iterative process either diverges or requires many iterations to obtain quadratic convergence (and, consequently, may become somewhat expensive).

### 3.1.2.2 Modified Newton's Method

For non-time-accurate calculations, the time-step term of Eq. (2.7) may be regarded as a relaxation parameter. Judicious specification of the time-step size leads to a time relaxation strategy that is numerically robust as well as capable of attaining quadratic rates of convergence.

For example, divergent behavior may be avoided by supplying the unsteady term of Eq. (2.7) with a pseudo-time-step that is inversely proportional to the $L_2$-norm of the CFD residual [88]

$$\Delta t^n = \frac{\Delta t_0}{||R^n||}$$

(3.3)

As a steady state is approached, the residual decreases; the unsteady term vanishes; and Newton's method is recovered. This time relaxation strategy is referred to as modified Newton's method.

### 3.1.2.3 Preconditioned Iterative Newton's Methods

Recently, preconditioned conjugate gradient-like methods have been used in conjunction with CFD codes to produce highly efficient solution algorithms [89–91]. Wigton et al. [90] points out that one solution cycle of a preconditioned conjugate gradient-like method is an approximation to one iteration of Newton's method. The common practice is to solve the linear system (2.7) inexactly and proceed to the next CFD time level. This quasi-Newton method, as applied in this study, has an exact Jacobian matrix and uses a matrix inversion that only approximates the exact inversion. Advantages of such methodologies include low memory requirements and high
convergence rates. Herein, these solution procedures will be referred to as preconditioned iterative Newton's methods.

The particular conjugate gradient-like solver that is used in this work is the Generalized Minimal Residual (GMRES) algorithm [92], which is a conjugate gradient-like method that has been generalized to efficiently solve non-symmetric linear systems. The GMRES algorithm is briefly outlined as follows [89]. Let $z_0$ be an approximate solution of the system

$$Az - b = 0$$

where $A$ is an invertible matrix. The solution is advanced from $z_0$ to $z_k$ as

$$z_k = z_0 + y_k$$

The GMRES($k$) method finds the best possible solution for $y_k$ over the Krylov subspace $\langle v_1, A^2v_1, A^3v_1, ..., A^{k-1}v_1 \rangle$ by solving the minimization problem

$$\| r_k \| = \min_y \| v_1 + Ay \|$$

$$v_1 = A z_0 - b \quad r_k = A z_k - b$$

In practice, the GMRES procedure forms an orthogonal basis $v_1, v_2, ..., v_k$ (termed search directions) spanning the Krylov subspace by a modified Gram-Schmidt method. As $k$ increases, the storage increases linearly and the number of operations increase quadratically. To mitigate this, Saad and Schultz [92] describe GMRES($k, m$), where the $k$ search directions are discarded and recomputed every $m$ cycles.

Preconditioning of the linear system $Az = b$ is essential to achieve solution procedures that have high convergence rates. The linear system with left preconditioning has the form

$$C^{-1}Az = C^{-1}b$$
where $C$ is the preconditioning matrix. Since preconditioning plays such an important role in the convergence of this approach, a short general discussion follows.

The effect of preconditioning is to cluster the eigenvalues of the particular problem around unity. This leads to a more favorable condition number of $C^{-1}A$ as compared to that of $A$ and, hence, results in higher convergence rates. The choice of $C$ is crucial to the success and efficiency of an iterative scheme. It is desirable to choose a preconditioning matrix that is: 1) inexpensive to invert, 2) lends itself to efficient matrix-vector multiplications, and 3) leads to a stable and convergent numerical procedure.

A class of preconditioners frequently employed are those based on regular splittings of the $A$ matrix [93]. By performing a lower/upper (LU) decomposition of $A$, but neglecting the fill-in of certain arbitrary off-diagonal elements that are chosen in advance, very sparse preconditioning matrices which resemble $A$ may be obtained. These types of sparse approximations are called Incomplete LU (ILU) decompositions of $A$. ILU decompositions that retain the same sparsity pattern as matrix $A$ are referred to as ILU(0) decompositions. Decompositions that allow fill-in beyond the original non-zero pattern of $A$ require increasingly greater memory storage requirements and computational work and are referred to as ILU($n$) decompositions ($n \in 1, 2, 3, ...$). The advantage of ILU($n$) decompositions is that the inverse of $A$ is more accurately represented as the value of $n$ increases.

Another option for the preconditioning matrix is to choose $C$ to be of the product of $C_1C_2$, where $C_1$ and $C_2$ have a simple matrix structure. Preconditioning of the linear system may then be applied by the solution of a sequence of easily invertible equations. In fact, the spatially-split approximately factored operators of the ADI scheme [cf. Eq. (3.1)] fall into this category of preconditioner.

In this work, all applications of the preconditioned conjugate gradient-like method use ILU(0) preconditioning as implemented by Anderson and Saad [94]. This approach has been shown to give good vector processing performance for CFD applications [89].
3.2 Discrete Sensitivity Equation

3.2.1 Comparison Between the Fluid Dynamic Equation and the Sensitivity Equation

Note that both the fully implicit fluid dynamic equation, Eq. (2.7), and the sensitivity equations, Eqs. (2.19), (2.22), and (2.23), may be considered as linear systems of the form \( A \mathbf{z} = \mathbf{b} \). Also, the Jacobian matrix \( \frac{\partial R}{\partial Q} \) of the sensitivity equation is identical to that of the fully implicit formulation of the CFD equation. Hence, in many respects, solving the sensitivity equation is similar to solving one Newton iteration of the fluid dynamic equation. The most obvious difference is that the sensitivity equation must be solved for multiple right-hand-side vectors. Thus, it is natural to question whether some of the standard solution techniques practiced in CFD may be applied toward the implicit solution of the aerodynamic sensitivity equation.

However, it is important to first recognize that many CFD practices are not directly applicable to sensitivity analysis procedures for several reasons. Since the CFD equation (2.7) is nonlinear (i.e., its Jacobian matrix and residual are both dependent upon the latest \( Q \) vector), obtaining a final CFD solution requires the solution of a sequence of intermediate linear problems. This iterative type of approach allows the freedom to make many approximations to the CFD left-hand-side operator so long as the steady state residual is driven to zero [82,85]; moreover, at each intermediate time level, the CFD linear system can be solved inexactly without sacrificing favorable convergence rates. In contrast, the sensitivity equation is a mathematically linear equation. Both its coefficient matrix (i.e., the true Jacobian matrix \( \frac{\partial R}{\partial Q} \) evaluated at a steady flow condition) and its right-hand-side vectors are known and invariant. This linear system must be solved exactly for each RHS vector. No modifications or approximations can be made to either the Jacobian matrix or the RHS vectors of the sensitivity equation without compromising its true solution. For example, failure to utilize the true Jacobian of a CFD residual which

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
incorporates a turbulence model will result in incorrect solutions to the sensitivity equation [55].

Nevertheless, modern CFD solution techniques indirectly do provide an abundance of ideas for solving the sensitivity equation, which may be broadly classified as follows.

3.2.2 Direct Inversion Methods

Since the sensitivity equation is linear, its solution requires only one matrix inversion. The most straightforward procedure is to compute an exact LU decomposition of \( \partial R/\partial Q \) \[or\] \( (\partial R/\partial Q)^T\), depending on the formulation] and then directly solve for the unknowns. This procedure enjoys the distinct advantage of reusing the LU decomposition to efficiently compute the unknowns for multiple right-hand-side vectors. That is, after the initial work of computing the inverse of \( \partial R/\partial Q \), the solution for each RHS involves only inexpensive forward and backward substitutions. Therefore, the overall cost of this direct inversion procedure is relatively insensitive to the number of right-hand-side vectors.

3.2.3 First Degree Iterative Methods

To solve the linear system \( A z = b \), a first degree iterative (defect-correction) method can be written as

\[
\begin{align*}
\delta z^m &= b - A z^m \\
z^{m+1} &= z^m + \delta z^m
\end{align*}
\]

where \( (b - A z) \) is the defect vector and \( \delta z \) is the incremental correction at stage \( m \). To accelerate the convergence of this simple scheme, a preconditioning matrix \( C \) may be introduced to yield the following system

\[
C \delta z^m = b - A z^m
\]

Most iterative CFD methodologies may be loosely regarded as preconditioned first degree iterative schemes since the LHS coefficient matrix controls the convergence process and the RHS vector contains the "physics" of the problem and defines the
accuracy of the solution [88]. The main difficulty involved with first degree iterative schemes is the need to specify iteration parameters (e.g., pseudo-time-step sizes) that directly affect numerical convergence rates. Improper selection of these parameters may result in slow convergence or even divergence.

Hence, the choice of preconditioning matrix is crucial to the success and efficiency of this iterative scheme. However, this iterative strategy does have the following important characteristic: any preconditioning matrix $C$ that drives the RHS vector $(b - Az)$ to zero may be used to obtain the correct solution to the linear system. For example, Jacobi, Gauss-Seidel, or ADI-factored operators may be considered as candidate preconditioning matrices for this scheme.

Likewise, the linear sensitivity equation can be reformulated into a preconditioned first degree iterative system. For example, Eq. (2.19) may be written as

$$C \delta \left( \frac{\partial Q}{\partial D} \right)^m = - \frac{\partial R}{\partial M} \frac{dX}{dD} \frac{\partial R}{\partial Q} \left( \frac{\partial Q}{\partial D} \right)^n$$

(3.11)

$$\left( \frac{\partial Q}{\partial D} \right)^{m+1} = \left( \frac{\partial Q}{\partial D} \right)^m + \delta \left( \frac{\partial Q}{\partial D} \right)^m$$

(3.12)

Korivi et al. [55,61] have recently investigated this “incremental iterative” strategy for solving the aerodynamic sensitivity equation.

### 3.2.4 Second Degree Iterative Methods

It can be shown that the conjugate gradient algorithm for solving $Az = b$ may be written as a three-term recurrence relation [95]

$$\left( z^{m+2} - 2z^{m+1} + z^m \right) + \left( 1 - \frac{\alpha_{m+1}}{\alpha_m} \beta_m \right) \left( z^{m+1} - z^m \right) = \alpha_{m+1} r^{m+1}$$

(3.13a)

or

$$\delta^2 z^m + \omega \delta z^m = \eta r^{m+1}$$

(3.13b)

where $\alpha$ and $\beta$ are scalar coefficients computed by the conjugate gradient algorithm, and $r$ is an update of the initial defect vector $(b - Az^0)$. Clearly this is a second degree...
iterative scheme. If $\beta = 0$ for all $m$, Eq. (3.13) then reduces to a first degree iterative scheme. One inherent advantage of second degree iterative schemes is that no estimation of iteration parameters is required to obtain high convergence rates [95].

The preconditioned conjugate gradient-like algorithms of Section 3.1.2.2 can be used to solve the linear systems of aerodynamic sensitivity analysis. In this context, this method may be viewed as a preconditioned version of a second degree iterative scheme

$$C \delta^2 z^m + \omega C \delta z^m = \eta^m = \eta^{m+1}$$

(3.14)

This class of solution methodology as applied to the aerodynamic sensitivity equation has been examined only recently [53,54,57,67,73]. Its effectiveness in the present design procedure is thoroughly investigated in Chapters 5 and 6.
In aerodynamic shape optimization procedures, the shape of the body surface and its surrounding computational grid are dictated by the vector of geometric-type design variables $D$. This is mathematically stated as $X = X(D)$. The design variables should be selected such that the grid may be easily regenerated as the design variable vector changes. Thus, it is desirable to obtain an explicit analytical function for $X(D)$, especially since the grid sensitivity terms $dX/dD$ may then be determined analytically. Finite-difference approximations of the grid sensitivity terms are possible but are prone to inaccuracies.

In this chapter, all of the shape-related aspects of the design procedure are examined in detail including general representations of the design surface, analytical procedures for grid adaptation, and grid sensitivity derivatives.

**4.1 Surface Representation**

4.1.1 Grid Point-Based Approach

The most obvious choice for geometric-type design variables is the surface grid points themselves. This approach leads to a direct representation of the surface, which is advantageous not only in its flexibility and generality but also with regard to its correlation to grid generation. However, the number of design variables ($NDV$) resulting from this approach is large. An immediate negative consequence of this is the large memory requirement for storing the $NDV$ right-hand-side vectors of the direct differentiation formulation of the sensitivity equation, Eq. (2.19). Note that these same
vectors (and hence memory) are also required in the adjoint variable formulation for computing the sensitivity coefficients [cf. Eq. (2.24)].

4.1.2 Bezier-Bernstein Parameterizations

In order to reduce the number of design variables, the use of a Bezier-Bernstein polynomial parameterization of the design surface is an attractive alternative. This procedure, which has found wide success in the grid generation field [96], can accurately represent a complex surface shape with a relatively small number of geometric control points. Hence, a reduced set of design variables may be adopted for use in the design procedure.

A two-dimensional contour can be represented by a $N$-th degree Bezier-Bernstein curve defined by

$$S_2(u) = \sum_{n=0}^{N} B_{n,N}(u) \cdot P_n$$

(4.1)

where $S_2(u) = \{\bar{x}_b(u), \bar{y}_b(u)\}$ and $P_n = \{\bar{P}_x, \bar{P}_y\}_n$.

The Bezier control parameters consist of the normalized computational arclength $u$ along the curve and the vector of geometric coefficients $P$, which are called the Bezier control points.

The basis functions $B_{n,N}$ are $N$-th degree Bernstein polynomials, which are given by

$$B_{n,N}(u) = \frac{N!}{n!(N-n)!} \cdot u^n \cdot (1-u)^{N-n}$$

(4.2)

A three-dimensional surface can be represented in the Bezier-Bernstein framework via a tensor product scheme, which is basically a bidirectional curve scheme. The three-dimensional surface has the form

$$S_3(u,v) = \sum_{n=0}^{N} \sum_{m=0}^{M} B_{n,N}(u) \cdot B_{m,M}(v) \cdot P_{nm}$$

(4.3)

where $S_3(u,v) = \{\bar{x}_b(u,v), \bar{y}_b(u,v), \bar{z}_b(u,v)\}$ and $P_{nm} = \{\bar{P}_x, \bar{P}_y, \bar{P}_z\}_{nm}$. The Bezier control points $P_{nm}$ are arranged in a bidirectional network (see Fig. 4.1).
For many reasons, the vector of Bezier control points is a natural choice for the geometric-type design variables. First, the formulation of Eqs. (4.1) to (4.3) are mathematically simple and numerically efficient. Second, the control points have a very geometrical interpretation. That is, a two-dimensional Bezier curve passes identically through its first and last control points, and in addition, its endpoint slopes may be specified exactly. Third, the shape produced is very smooth and does not have spurious waves between the control points. Last, since each individual point along the Bezier surfaces is influenced to some degree by every control point, this formulation conveniently lends itself to an analytical computation of the grid sensitivity terms.

One potential difficulty in employing Bezier curves in the present design procedure is the initial specification of the Bezier control parameters. In other words, the proper specification of \( P, u, \) and \( v, \) which recovers the shape and nodal distribution of the initial discretized surface boundary needs to be determined. A procedure, similar to that of Ref. 30, has been developed to solve this inverse Bezier problem and was applied to the design configurations of this work.

### 4.2 Grid Adaptation

Once a new surface shape has been defined, it remains to construct the surrounding computational grid about the shape. The task then becomes to develop an explicit relationship between the interior grid points and the surface boundary points.

The approach adopted here is a relatively simple but effective one—the original surrounding grid is spatially adapted to account for the new surface shape. The spatial adaptation experienced by a typical grid line, which is described by \( i_{\text{max}} \) discrete nodes, is depicted in Fig. 4.2.

The adaptation procedure begins by defining normalized distribution functions that may be used to parameterize each surface-normal grid line. For example, a projected normalized distribution function in terms of the \( x \)-coordinate is given by
Each grid line is then adapted to account for the new surface boundary shape via the following relationship

$$\hat{f}_x(i) = \frac{x_i - x_b^{old}}{x_{imapx} - x_b^{old}} \quad i \in 1, \ldots, imax \quad (4.4)$$

The normalized distribution function is assumed to be locally invariant, and the outer boundary point \((x, y, z)_{imapx}\) is assumed to be spatially fixed. Relationships analogous to Eqs. (4.4) and (4.5) govern the adaptation of the normal grid line in terms of the \(y\)- and \(z\)-coordinates.

However, numerical problems arise in this approach if one of the coordinate values of both the outer boundary point and the surface boundary point are equal or nearly equal (cf. Fig. 4.2). If the denominator of Eq. (4.4) is identically zero, the normalized distribution function is undefined; if the denominator is very nearly zero, roundoff errors will introduce numerical noise into the adaptation procedure. This problem is circumvented by adopting an arclength-based approach for grid adaptation. The new adapted normal grid line may be described by

$$x_i^{new} = x_b^{new} + \hat{f}_x(i) \cdot (x_{imapx} - x_b^{new}) \quad (4.5)$$

where

$$\text{arc}(i) = \frac{\sum_{t=2}^{i} L_t}{\sum_{t=2}^{imax} L_t} \quad (4.7a)$$

$$L_t = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2 + (z_t - z_{t-1})^2} \quad (4.7b)$$

Here, the old values are assumed to be spatially fixed. Relationships similar to Eq. (4.6) may be written for the \(y\)- and \(z\)-coordinates. Note that if the surface boundary point is not relocated, then the grid line simply retains its original shape.

Thus, an explicit analytical relationships are obtained between the interior points and the boundary points for each surface-normal grid line. Repeated applications of Eq. (4.5)
or Eq. (4.6) lead to a very simple and efficient grid regeneration procedure that accounts for a new surface shape and is based on information from the original grid. For small deformations of the surface shape, the quality of the adapted grids is comparable to that of the original.

4.3 Grid Sensitivities

4.3.1 Grid Point-Based Approach

Assuming that the vector of geometric-type design variables \( D = \{D_x, D_y, D_z\} \) consists of the vector of surface boundary nodes \( X_b = \{x_b, y_b, z_b\} \), the grid sensitivity terms may be expressed by

\[
\frac{dX}{dD} = \frac{dX}{dX_b}
\]  

(4.8)

For the grid \( X = \{x, y, z\} \), a straightforward differentiation of the approach given by Eq. (4.4) with respect to \( x_b \), \( y_b \), \( z_b \) yields the following analytical grid sensitivity terms

\[
\frac{dx}{dD_x} = \frac{dx}{dx_b^{new}} = 1 - \frac{1}{r_x} \quad \frac{dx}{dD_y} = \frac{dx}{dy_b^{new}} = 0 \quad \frac{dx}{dD_z} = \frac{dx}{dz_b^{new}} = 0
\]  

(4.9)

Similarly, the approach of Eq. (4.6) yields the following grid sensitivity terms

\[
\frac{dx}{dD_x} = \frac{dx}{dx_b^{new}} = 1 - \frac{1}{ar_x} \quad \frac{dx}{dD_y} = \frac{dx}{dy_b^{new}} = 0 \quad \frac{dx}{dD_z} = \frac{dx}{dz_b^{new}} = 0
\]  

(4.10)
4.3.2. Bezier-Bernstein Parameterizations

For the Bezier-Bernstein surface representations, the boundary nodes $X_b \equiv S_2(u,v)$ are dependent upon the design variables $D$, which are taken to be the Bezier control points $P_{nm}$ and hence the grid sensitivity terms are determined by

$$\frac{dX}{dD} = \frac{\partial X}{\partial X_b} \frac{dX_b}{dD} = \frac{\partial X}{\partial X_b} \frac{dX_b}{dP_{nm}}$$  \hspace{1cm} (4.11)

Using Eq. (4.3), the term $dX_b/dP_{nm} \equiv dS_3/dP_{nm}$ may be more explicitly expressed as

$$\frac{dx_b}{dP_{x,nm}} = B_{n,N}(u) \cdot B_{m,M}(v) \quad \frac{dy_b}{dP_{y,nm}} = 0 \quad \frac{dz_b}{dP_{z,nm}} = 0$$

$$\frac{dx_b}{dP_{x,nm}} = 0 \quad \frac{dy_b}{dP_{y,nm}} = 0 \quad \frac{dz_b}{dP_{z,nm}} = 0$$

$$\frac{dx_b}{dP_{z,nm}} = B_{n,N}(u) \cdot B_{m,M}(v) \quad \frac{dy_b}{dP_{y,nm}} = 0 \quad \frac{dz_b}{dP_{z,nm}} = 0$$

Thus, by combining the grid adaptation technique [Eq. (4.6), for example] with the Bezier-Bernstein parameterization of the surface contour [Eqs. (4.11) and (4.12)], the grid sensitivity terms become

$$\frac{dx}{dD_x} = (1-\alpha c) \cdot B_N(u) \cdot B_M(v) \quad \frac{dx}{dD_y} = 0 \quad \frac{dx}{dD_z} = 0$$

$$\frac{dy}{dD_x} = (1-\alpha c) \cdot B_N(u) \cdot B_M(v) \quad \frac{dy}{dD_y} = 0 \quad \frac{dy}{dD_z} = 0$$

$$\frac{dz}{dD_x} = (1-\alpha c) \cdot B_N(u) \cdot B_M(v) \quad \frac{dz}{dD_y} = 0 \quad \frac{dz}{dD_z} = 0$$

(4.13)

These analytical expressions explicitly describe the sensitivity of the computational grid with respect to the Bezier control point design variables. Two-dimensional analogies to these expressions may be easily written. These flexible shape-related procedures have been used with much success in previous shape optimization applications [63–67,73].

4.4 Wing Geometry Model

A very flexible wing geometry model that is totally based on two- and three-dimensional Bezier-Bernstein parameterizations is described in this subsection.
4.4.1 Geometric Deformations

Consider in Fig. 4.3 the discrete computational mesh of an elementary wing surface. This geometrically simple wing is unswept, untwisted, and rectangular with both its chord and span equal to unity; this wing will be referred to as the "unit wing." Each airfoil section of this wing is a NACA-0012 cross-section that is strictly defined in an $x$-$z$ plane. Oriented at zero degrees angle-of-attack, all chord lines of the unit wing lie in the $z = 0$ plane. Let the set of discrete points that describe the unit wing be denoted by \( \{x_0, y_0, z_0\} \).

For design purposes, it is desired to manipulate or deform the unit wing into a new improved shape. In order to generate a great variety of shapes, the geometric description of a general wing should include the following features:

1) arbitrary wing section (airfoil) definitions,
2) arbitrary taper distribution,
3) arbitrary axial displacement of each airfoil section (i.e., sweep),
4) arbitrary span length,
5) arbitrary normal displacement of each airfoil section (i.e., spanwise bending),
6) arbitrary geometric twist schedule,
7) arbitrary global angle-of-attack, and
8) consistent and realistic treatment of wing tip region.

The combined geometric deformations of features 2 through 4 will yield the planform shape and aspect ratio of an untwisted wing.

The present wing design model has been specifically developed to incorporate all of the above geometric features in an efficient and functional manner. Each feature is implemented as a distinct and independent geometric operation. These operations are now described.

A. The first geometric operation is the unique centerpiece of the flexible wing model—the airfoil sections are partially defined by imposing the desired thickness and
chordwise camber distributions onto the unit wing. This is accomplished by locally displacing the surface points of each airfoil section in a direction normal to its chord line. In this work, one of two approaches is used to perform this operation:

A.1. The airfoil thicknesses may be varied in the spanwise direction to define a wing made up of a sequence of symmetric NACA-00xx cross-section definitions. The wing’s chordwise camber remains unchanged. Hence, only a vector of thickness scale factors as a function of span (thkscal) is required. The new wing is described by

\[ x_A = x_0, \quad y_A = y_0, \quad z_A = z_0 \times \text{thkscal}(k) \quad (4.14) \]

(Note that the discrete computational index \( k \) runs along the \( y \)-direction from the root station to the last span station before the tip region. The \( k \)-th scale parameter operates on the corresponding \( k \)-th airfoil section. For convenience, the discrete indices are omitted from the wing coordinates \( \{x, y, z\} \).)

A.2. For more general airfoil definitions, the upper and lower wing surfaces (excluding the tip region) may be represented via a three-dimensional Bezier-Bernstein parameterization of each respective surface (see Fig. 4.1). This approach permits very general distributions of both thickness and chordwise camber across the wing. Details regarding this parameterization were given in section 4.1.2, but suffice it to say here that the wing is described by

\[ x_A = x_0, \quad y_A = y_0, \quad z_A = f(u, v, P) \quad (4.15) \]

where \( u, v, \) and \( P \) are Bezier control parameters. The design variables for each surface are taken as the \( z \)-components of the 25 interior Bezier control points, i.e., all control points except those located on the wing’s leading- and trailing-edges.

B. Since each airfoil section of the unit wing has a chord of unity and also has its leading-edge point located on the \( y \)-axis, the taper distribution may be efficiently handled by the specification of a vector of chord scale factors as a function of span (chdscal).
This operation will simply shrink or enlarge the chord length of each spanwise airfoil section via
\[ x_B = x_A \cdot \text{chdscal}(k), \quad y_B = y_A, \quad z_B = z_A \] (4.16)

At this point, a wing having the desired airfoil shapes has been created.

C. The spanwise axial and normal displacements of the wing are handled by prescribing for each airfoil section the \( x \) and \( z \) locations of a specified reference point that lies on the chord line (\( f_{\text{chd}} \)). For example, the aerodynamic center of a NACA-0012 cross-section (i.e., the quarter-chord) may be selected as the reference chord point. This operation requires two translation distributions as a function of span (\( trn_x \) and \( trn_z \)) for specifying the \( x \) and \( z \) locations of \( f_{\text{chd}} \). In addition, the taper distribution is considered to be centered about the \( f_{\text{chd}} \) reference point and, hence, requires including a corresponding axial displacement.

\[ x_c = x_B + trn_x(k) - f_{\text{chd}} \cdot \text{chdscal}(k), \quad y_c = y_B, \quad z_c = z_B + trn_z(k) \] (4.17)

D. Since the unit wing's root station lies in the \( y = 0 \) plane, the half-span length may be simply handled through a single scalar multiplier (\( spn \)).

\[ x_D = x_c, \quad y_D = y_c \cdot spn, \quad z_D = z_c \] (4.18)

At this point, a wing with the desired airfoil definitions, planform shape, and spanwise bending distribution has been defined. This was achieved through the systematic application of scaling factors and spatial translations to the unit wing.

E. The wing's geometric twist is obtained by locally rotating each airfoil section according to a twist distribution that is defined as a function of span (\( twst \)). Each airfoil section may be rotated about a specified reference chord point (\( f_{twst} \)); for example, the quarter-chord location may be selected.

\[ x_B = + (x_D - twst) \cdot \cos[twst(k)] + (z_D - twst) \cdot \sin[twst(k)] + twst \]

\[ Y_B = Y_D \]
\[ z_h = -(x_D - x_{twst}) \sin[\text{twst}(k)] + (z_D - z_{twst}) \cos[\text{twst}(k)] + z_{twst} \]
\[ x_{twst} = (f\text{twst} - f\text{chd}) \times \text{chdscal}(k) + \text{trnx}(k) \]
\[ z_{twst} = \text{trnz}(k) \quad (4.19) \]

F. The angle-of-attack (aoa) is imposed by rotating the entire wing as a rigid body about the root section quarter-chord location. After the appropriate mathematical modifications have been made, this geometric transformation may also be described by Eq. (4.19).

G. Finally, the new wing tip region is generated by applying analogous operations A through F with extrapolated geometric quantities to the unit wing tip region.

At this point, the final desired wing shape has been generated. For ease and consistency of application, it is recommended that the twisting operations E and F be performed after the scaling and translational operations A through D.

Summarizing, a new wing shape has been derived from the “unit wing” shape by applying a sequence of geometrical deformations based on five spanwise parameter distributions (th\text{kscal}, chdscal, trnx, trnz, and twst) and four scalar parameters (spn, aoa, fchd, and ftwst). Since the design shape of the wing depends on these parameter distributions, the manner in which these distributions are represented will dictate the type and number of design variables to be used in the shape optimization procedure.

4.4.2 Spanwise Parameter Distributions

The spanwise parameter distributions should be either smoothly or piecewise continuous. The most general treatment of these distributions would be to assign a parameter value (i.e., a design variable) to each discrete spanwise station, but this approach has two obvious disadvantages. First, this approach would yield a large number of design variables, which would adversely impact the computational memory and work requirements of the design procedure. Second, if two neighboring design variable values...
are very discrepant (discontinuous), a poor aerodynamic design would likely be produced.

In many cases, a parameter distribution may be sufficiently described by a piecewise linear variation. For example, a linear taper schedule may be efficiently prescribed using only two design variables (see Fig. 4.4a).

\[
\text{chdscal}(k) = D_1*\left[1-y(k)\right] + D_2*y(k) \tag{4.20}
\]

A more general taper schedule may be produced by introducing more interior interpolation locations (see Fig. 4.4b). This approach is naturally suited to model wing planform breaks, etc., due its representation of a geometric feature in a piecewise continuous fashion.

Smoothly continuous parameter distributions are not guaranteed for the approach involving the prescription of one design variable per spanwise station. Also, smoothly continuous distributions are not produced when using a piecewise linear variation (except for the two design variable case). A novel approach, which promises enhanced geometric flexibility, has been adopted in this work to represent the spanwise parameter distributions; this approach proposes the use of a two-dimensional Bezier-Bernstein parameterization of the spanwise distributions (see Fig. 4.4c). This approach has several advantages including: 1) the possibility of modeling smoothly continuous variations; 2) a relatively small number of design variables can produce a wide range of realistic distributions; and 3) the design variables take on very geometrical interpretations.

4.4.3 Grid Sensitivities

It is desirable to obtain the grid sensitivity terms analytically since a finite-difference approach may introduce significant numerical errors. This means that the geometric deformations to the unit wing as well as the grid adaptation procedure must be analytically differentiable with respect to all design variables. Hence, all operations of
the flexible wing model and also the adaptation procedure were developed from the outset to be differentiable. A chain-rule type of evaluation may be used to compute the grid sensitivities of the wing surface points by differentiating each geometric operation independently. The arclength-based grid adaptation procedure is used in this wing model, which has been shown (in section 4.3.2) to be analytically differentiable with respect to all design variables as well.
Fig. 4.1 Three-dimensional Bezier-Bernstein representation of a wing upper-surface.

Fig. 4.2 Spatial adaptation of a typical surface-normal grid line.
Fig. 4.3 The "unit wing" geometry.
(a) Linear root-to-tip variation

(b) Three-point piecewise continuous

(c) Fourth degree Bezier-Bernstein

Fig. 4.4 Parameter distributions and their resulting half-planform shapes.
Chapter 5

A FUNCTIONAL THREE-DIMENSIONAL DESIGN PROCEDURE

This chapter discusses the practical issues related to the integration of the separate elements of aerodynamic shape optimization into a functional design procedure. Several numerical aspects of the present design procedure are critically examined.

5.1 The General Optimization Procedure

The general optimization procedure used in this work is outlined in Table 5.1.

<table>
<thead>
<tr>
<th>Table 5.1 The Aerodynamic Shape Optimization Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Formulate the optimization problem and select an initial vector of design variables.</td>
</tr>
<tr>
<td>2. Obtain the optimization gradient information.</td>
</tr>
<tr>
<td>a. Compute a high fidelity flow solution for the current design.</td>
</tr>
<tr>
<td>b. Perform the aerodynamic sensitivity analysis, i.e., compute the sensitivity coefficients.</td>
</tr>
<tr>
<td>3. Obtain the search direction vector $S$ and initialize $\alpha$.</td>
</tr>
<tr>
<td>4. Perform a one-dimensional search along $S$.</td>
</tr>
<tr>
<td>a. Increment $\alpha$ and update $D$ to obtain a new design point.</td>
</tr>
<tr>
<td>b. Compute a flow solution for the new design.</td>
</tr>
<tr>
<td>c. Evaluate the objective function and constraints.</td>
</tr>
<tr>
<td>d. If the design is improved, go to step 4a; else terminate search.</td>
</tr>
<tr>
<td>5. Check the optimization convergence criteria.</td>
</tr>
<tr>
<td>If the termination criteria is met, stop; else go to step 2 (i.e., begin a new design iteration).</td>
</tr>
</tbody>
</table>
For three-dimensional aerodynamic optimization procedures, the use of efficient implicit solution methodologies is of the utmost importance. Extensions to three-dimensions of even the most efficient two-dimensional design procedures will invariably result in substantial increases in both CPU time and memory requirements. A critical examination of current solution methodologies as applied to different aspects of an aerodynamic shape optimization procedure is examined in this chapter and also in Chapter 6. Some of the available options to perform the most computationally intensive operations within the design procedure are compiled in Table 5.2.

| Flow Solutions Prior to Sensitivity Analysis (cf. step 2a of Table 5.1) | ADI Methodology | Direct Inversion Newton’s Method | Preconditioned Iterative Newton’s Method |
| Flow Solutions Within the One-Dimensional Search (cf. step 4b of Table 5.1) | Approximate Flow Analysis | Direct Inversion Newton’s Method | Preconditioned Iterative Newton’s Method |
| Solution of the Aerodynamic Sensitivity Equation (cf. step 2b of Table 5.1) | Direct Inversion | Preconditioned First Degree Iterative | Preconditioned Second Degree Iterative |

5.2 Numerical Aspects of CFD Within the Optimization Procedure

This section outlines the underlying reasons for selecting the particular flow solution methodologies used within the optimization procedure.

5.2.1 CFD Prior to the Sensitivity Analysis

Proper derivation of the sensitivity equation is based on a steady state flowfield solution, that is, a CFD flow solution with $R = 0$. Consequently, it follows that the CFD solution computed prior to the sensitivity analysis (cf. step 2a of Table 5.1) should always be highly converged. This high fidelity CFD solution is required to compute accurate
sensitivity coefficients and, hence, ensures an accurate search direction $S$. Any explicit or implicit CFD method may be used to obtain such a solution; however, it is imperative that the sensitivity equation based on this solution be a consistent differentiation of the corresponding CFD residual including the boundary conditions [63,97].

Thus, selection of a CFD solution methodology that is both accurate and efficient is desirable for obtaining these high fidelity flow solutions. In general, the initial CFD flowfield for the optimization process is most efficiently obtained by using ADI methodologies. In the present work, the subsequent CFD flowfields computed prior to the sensitivity analysis are obtained by using either an ADI solution method or one of the fully implicit methods. The practical evaluation of each solution method within the design procedure will be addressed in Section 6.1.

The rationale for employing the fully implicit methods in the present design procedure is as follows. In standard CFD applications, the implementation of Newton’s method is sometimes considered overly burdensome due to the need for an exact Newton-linearization and the solution of an unfactored linear algebraic system of large bandwidth. Furthermore, this method requires a fairly close guess to the final solution before the method is numerically stable and can give quadratic convergence. However, the present design procedure is ideally suited to Newton’s method because: 1) neighboring designs (and hence their flow solutions) are only incrementally different from one another; and 2) the linear algebraic system of the sensitivity equation and its numerical solution closely resembles that of the fully implicit fluid dynamic equation. Thus, it is desired to retain a fully implicit CFD formulation within the three-dimensional design procedure. However, this in itself is a formidable numerical challenge; only a few recent examples of three-dimensional unfactored implicit CFD calculations have been found in the literature [98–101].
5.2.2 CFD Within the One-Dimensional Searches

Although highly accurate CFD solutions are necessary to compute the sensitivity coefficients, coarse CFD solutions may be sufficient during the numerous objective function evaluations of the one-dimensional searches (cf. step 4b of Table 5.1).

5.2.2.1 Approximate Flow Analysis

This relaxed requirement of CFD accuracy opens the possibility of using approximate flow analysis as an alternative to full CFD analysis (cf. Section 3.3.4). In fact, approximate flow analyses have been used with much success within the one-dimensional searches of the optimization design procedure [10,63,64].

However, there are several disadvantages of using the approximate analysis method within the one-dimensional search. First, this method produces an approximation of the actual flow solution at the neighboring design point. The quality of the predicted solution is dependent upon the magnitude of the design deviation from the baseline design point; increasing inaccuracies occur for increasingly large design deviations (cf. Fig 2.1). Due to this adverse dependence on the size of allowable design deviations, the optimization process is forced to proceed using relatively small changes in the geometric shape. In other words, when using approximate flow analysis, an accurate one-dimensional search will require many small $\Delta\alpha$ step size iterations [cf. Eq. (2.9) and (2.13)]. Larger design deviations would reduce the number of one-dimensional search iterations, but such deviations require the use of conventional CFD analyses in order to provide sufficiently accurate flow solutions. Second, since approximate flow analysis requires the use of the direct differentiation formulation of the sensitivity equation, high or prohibitive computational costs may result depending on the problem size and the number of design variables.

To overcome these difficulties, the feasibility of using conventional CFD procedures within the one-dimensional search has been examined. The use of iterative ADI schemes
are prohibited due to their slow and expensive rates of convergence. However, a survey of the more efficient CFD approaches reveals two viable candidates.

5.2.2.2 Direct Inversion Newton's Method

One alternative is the use of a modified Newton's method that is based on a direct inversion (LU decomposition) linear system solver. In practice, however, this approach encounters several problems.

For moderately large $\Delta \alpha$ step sizes, 4 to 10 modified Newton iterations are typically required before quadratic convergence is realized for each succeeding design. This leads to a rather high cost for each CFD analysis within the one-dimensional search. To minimize these costs, an optimal choice must be made for the $\Delta \alpha$ step size in the one-dimensional search. For small $\Delta \alpha$ step sizes (small design changes), Newton's method quickly attains quadratic convergence, but the overall number of the one-dimensional search iterations remains high. (Recall that a CFD analysis is required for each one-dimensional search iteration.) For large $\Delta \alpha$ step sizes (larger design changes), more Newton iterations are necessary to obtain a sufficiently converged solution, however fewer one-dimensional search iterations are required.

It turns out that the most critical factor when choosing an optimal $\Delta \alpha$ step size is not the computational cost, but rather, the desired resolution of the final design. It is found that use of too large of a $\Delta \alpha$ step size: 1) can produce non-optimal search directions that may ultimately lead to an inferior final design; and 2) can inhibit "fine-tuning" of the design when near its final local optimum. This finding will be demonstrated in practice in Section 6.1.2.

Another problem of this approach centers around a convergence anomaly associated with Newton methods. In particular, large saw-tooth oscillations are sometimes observed to occur in this method's convergence history (e.g., see Refs. [89 and 102]). If a preset CFD convergence criterion is prescribed, it is typically satisfied during one of the
oscillations exhibiting a small residual error. The potential danger of this circumstance is
the unintentional use of inaccurate flow solutions that could mislead the optimizer. For
example, if the strength and/or location of a shock on a transonic airfoil is not correctly
predicted due to a poorly converged flow solution, incorrect aerodynamic force
coefficients would result for that particular design, and the quality of that design would
be erroneously represented to be either better or worse than it actually is.

5.2.2.3 Preconditioned Iterative Newton’s Method

Another option for CFD analysis within the one-dimensional searches is the use of
Newton’s method that is based on preconditioned conjugate gradient-like solvers. The
prevailing conclusion regarding the general use of Newton’s method has been that it is
very effective within its domain of attraction, but is impractically slow otherwise.
However, Venkatakrishnan [89] has shown that fully implicit methods coupled with
preconditioned conjugate gradient-like solvers are competitive with the best iterative
methods for two-dimensional problems. Furthermore, these methods exhibit high rates of
convergence with minimal saw-tooth convergence oscillations.

5.3 Numerical Aspects of Sensitivity Analysis Within the Optimization
Procedure

A major disadvantage exists when using either the first or second degree iterative
solution strategy to solve the sensitivity equation. Both of these iterative approaches
require a complete solution of the sensitivity equation linear system for each right-hand-
side vector. Moreover, the computational time required for each linear system solution is
non-trivial. This is unlike the direct inversion method in which efficient forward and
backward substitutions may be performed to obtain solutions for each RHS vector. Thus,
for many RHS vectors, the iterative approaches may become quite CPU time intensive.
In fact, depending on the efficiency of the CFD flow solver, the solution of the direct
differentiation formulation of the sensitivity equation (NDV linear system solutions)
using an iterative solution strategy may be more expensive than a finite-difference approach \((NDV + 1)\) CFD analyses) for computing the flow derivatives \(\partial Q/\partial D\).

Due to the lack of diagonal dominance associated with the higher-order differencing of the CFD steady state residual, the linear algebraic systems of the aerodynamic sensitivity equation and also the time-asymptotic fully implicit CFD equation are ill-conditioned. This does not pose a problem for solution methods based on direct inversion since an LU decomposition will exactly invert the coefficient matrix. However, for the iterative solution methods, the possibility always exists that the method will fail to converge (i.e., stall) for these poorly conditioned linear systems.

In some of the present two-dimensional design cases (transonic flows about airfoils), the ordering of the equations of the sensitivity analysis linear system proves critical as to whether the preconditioned GMRES algorithm converges or not. It is felt that this is due to flux-vector-splitting and the locally supersonic character of the flow (i.e., at supersonic points, zero elements occur in the Jacobian matrix due to upwind-differencing). Ordering the equations in the cross-stream direction places these zero matrix elements in the outermost diagonals of the sensitivity coefficient matrix; this leads to convergent behavior in the ILU(0) preconditioned GMRES. A streamwise ordering places the zero elements within the innermost matrix diagonals and leads to stalled convergence. (Interestingly, the GMRES convergence of the CFD linear system for this particular design problem did not display any dependence on the equation ordering. However, Orkwis [103] has reported the failure of an ILU(0)/conjugate gradient-like combination in a CFD context that was attributed to zeroes within the bands of the Jacobian matrix.)

For sensitivity analysis, failure of the iterative methods to converge is especially detrimental. First, poorly converged solutions to the sensitivity equation will yield inaccurate sensitivity coefficients and, consequently, lead to erroneous search directions. Second, because of its linear mathematical nature, little can be done to improve the spectral radius of the Jacobian matrix. Hence, for these ill-conditioned linear systems, the
choice of preconditioning matrix is of vital importance in order to simply obtain converged solutions. This numerical issue will be further examined in Section 5.4.2.

5.4 Numerical Aspects of the Implicit Solution Methodologies

The major challenge for the present three-dimensional design optimization procedure is resolving the demanding computational issues associated with the numerical solution of its large linear algebraic systems. The implicit solution methodologies considered in Chapter 4 will be further examined here with focus on their suitability for the three-dimensional CFD equation, Eq. (2.7), and sensitivity equations, Eqs. (2.19), (2.22), and (2.23).

5.4.1 Direct Inversion Methods

In this work, an LU decomposition solver is used for all applications of the direct inversion methods. Highly vectorized solvers based on LU decomposition can perform the “numerically exact” inversion of the Jacobian matrix quite efficiently for small two-dimensional problems [104]. However, direct linear solvers based on Gaussian elimination-type decompositions suffer from large fill-in and, consequently, will result in prohibitive memory requirements and unreasonable CPU costs for practical three-dimensional problems. Out-of-core direct solvers [105] may significantly mitigate the in-core memory requirements for three-dimensional problems. However, this type of direct solver still requires large amounts of auxiliary disk storage, and if solid-state disks (SSD) are not utilized, its unreasonable CPU costs may be further exacerbated by increased I/O costs.

Hence, for practical three-dimensional problems, one must resort to solution techniques that have reduced memory requirements. Toward this end, the use of domain decomposition techniques is a viable option and has been investigated recently [60,74]. The use of the low-memory preconditioned conjugate gradient-like iterative solver is another valid option [57,73,98,100,101] and is evaluated in the following subsection.
5.4.2 Preconditioned Iterative Methods

In the present three-dimensional design applications, the standard ILU(0)/GMRES combination described in Section 3.1.2.3 failed to converge (i.e., stalled) for both the fully implicit fluid dynamic equation and the aerodynamic sensitivity equation. Reordering the equations to locate any zeroes in the outermost matrix diagonals did not improve the convergence characteristics. The convergence problem was finally resolved by appropriately modifying the preconditioning matrix and the RHS vectors as described below.

First, it is helpful to recall that the LHS operator of Eq. (3.13) [and also Eq. (3.10)] controls the convergence process and that the RHS vector contains the “physics” of the problem and defines the accuracy of the solution. Consequently, the convergence characteristics of the preconditioned iterative methods may be improved by choosing $C$ to be based on a diagonally-augmented version of $A$, that is, $C = \text{ILU}(0)$ of $A_{\text{LHS}}$, where

$$A_{\text{LHS}} = \frac{I}{\omega_{\text{LHS}}} + \frac{\partial R}{\partial Q} \quad (5.1)$$

and $\omega$ is a scalar relaxation parameter (e.g., a pseudo-time-step size for CFD applications). The accuracy of the solution is maintained by using the correct (or a consistent) coefficient matrix $A$ in the RHS vector, that is, $b - A_{\text{RHS}} z$, where

$$A_{\text{RHS}} = \frac{I}{\omega_{\text{RHS}}} + \frac{\partial R}{\partial Q} \quad (5.2)$$

Options for the relaxation parameters $\omega_{\text{LHS}}$ and $\omega_{\text{RHS}}$ include

$$\omega_{\text{INF}} = \infty \quad (5.3a)$$

$$\omega_{\text{RES}} = \frac{\omega_0}{\|R\|} \quad (5.3b)$$

$$\omega_{\text{SER}} = \min(\omega_{\text{RES}}, \omega_{\text{max}}) \quad (5.3c)$$

where $\omega_0$ is an appropriately chosen constant, $\|R\|$ is the $L_2$-norm of the CFD residual, and $\omega_{\text{max}}$ is the maximum allowable relaxation parameter. In this work, $\omega_0 = 0.05$ and
\( \omega_{\text{max}} = 1400. \) Equation (5.3c) is frequently referred to as the Switched Evolution-Relaxation (SER) strategy.

A systematic study was performed to investigate the convergence characteristics of the preconditioned conjugate gradient-like method when applied to the fully implicit fluid dynamic equation. The CFD analysis involved computing a Mach 0.76 steady state flowfield about a 17x17x43 transonic transport wing geometry using a converged Mach 0.75 flowfield as an initial condition. Two GMRES restart cycles were performed at each time step, and 20 GMRES search directions were employed during the iterative solution of the linear system, i.e., GMRES(20, 2). The fully implicit CFD solver required 15.5 Mwords of memory and approximately 14 Cray Y-MP seconds per Newton iteration.

In Fig. 5.1, the convergence histories for various relaxation strategies are shown in terms of the CFD residual. General observations include: 1) the use of preconditioners having no diagonal-augmentation \((\omega_{\text{LHS}} = \omega_{\text{INF}})\) leads to numerical divergence; 2) preconditioners based on \(\omega_{\text{LHS}} = \omega_{\text{RES}}\) become ill-conditioned as \(\|R\| \to 0\) and lead to stalled rates of convergence; and 3) preconditioners that retain diagonal dominance \((\omega_{\text{LHS}} = \omega_{\text{SER}})\) provide stable and convergent results. The choice of relaxation in the \(A_{\text{RHS}}\) matrix tends to affect the solution speed as well—too much relaxation \((\omega_{\text{RHS}} = \omega_{\text{SER}})\) leads to very slow linear rates of convergence.

A similar investigation was performed for the three-dimensional sensitivity analysis. The following relaxation factors were used \(\omega_{\text{LHS}} = \text{constant}\) and \(\omega_{\text{RHS}} = \omega_{\text{INF}}\). Numerical experimentation indicated that the best convergence rates were obtained for \(\omega_{\text{LHS}} = 1000.\) Values much greater or lesser than this were found to result in stalled GMRES convergence. It is imperative that the true unmodified Jacobian \(\partial R/\partial Q\) appear in the RHS vector in order to obtain correct solutions to the linear sensitivity equation. A solution convergence tolerance of \(1.0 \times 10^{-5}\) is usually easily met in less than 30 GMRES restart cycles using 20 GMRES search directions, i.e., GMRES(20, 30).
In summary, a necessary and key element in obtaining solutions to the present three-dimensional unfactored linear algebraic systems is that the preconditioning matrix $C$ be based on a diagonally-dominant coefficient matrix.
Fig. 5.1 Convergence histories of three-dimensional CFD analysis using various relaxation strategies.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Chapter 6
RESULTS AND DISCUSSION

The purposes of this chapter are two-fold. First, the computational aspects of the design procedure are critically examined in order to increase the efficiency of aerodynamic shape optimization as much as possible. This is a necessary step before practical three-dimensional design can be seriously contemplated. This examination is carried out strictly on two-dimensional design problems, which require only moderate amounts of computational memory and execution time. Second, upon establishment of the most efficient optimization strategy, aerodynamic design applications are then performed for practical three-dimensional problems. Hence, Section 6.1 will primarily focus on the computational issues of two-dimensional design, and Section 6.2 will consider three-dimensional wing design applications.

6.1 Two-Dimensional Results

As inferred from Table 5.2, a number of possible combinations of methods are available to define a unique strategy for the optimization procedure. The proposed optimization strategies used in this work are outlined in Table 6.1.

6.1.1 Supersonic Internal-External Nozzle

The first design problem considered is a supersonic internal-external nozzle configuration whose ramp section is redesigned to maximize the axial component of the thrust vector. Optimization strategies 1, 2, and 3 of Table 6.1 are applied to this problem. This configuration has been previously examined and shape-optimized (using strategy 1)
by Baysal et al. [64]. Options for surface representation and $CFD_{VF}$ are critically assessed in this section.

The salient features of the problem are illustrated in Fig. 6.1. The initial design shape of the ramp section is a flat surface declined at a 10-deg initial expansion angle. The physical domain is discretized using 53x41 grid points (with 48 points defining the ramp surface). The optimization problem is formulated to include three aerodynamic inequality constraints ($NCONQ = 3$) and no geometric inequality constraints ($NCOND = 0$). The purpose of the aerodynamic constraints is to limit the static pressure values at the ramp and cowl tips such that no reverse flow occurs there. The surface of the ramp section is represented by a vector of design variables consisting of either 47 local relative slopes ($NDV = 47$) or the $y$-coordinates of six of the seven Bezier control points that define the ramp shape ($NDV = 6$). The movements of all grid points as well as the Bezier control points are restricted to the $y$-direction. The $a priori$ specification of the degree of the Bezier curve and also the fixed axial locations of the Bezier control points does not permit the generation of every possible shape; nevertheless, ample geometric flexibility does exist to deform the ramp to physically realistic shapes. Note that the design problem is formulated such that the adjoint variable formulation of the sensitivity equation is preferred since $NCONQ + 1 < NDV$.

The computational statistics associated with the shape optimization for the different optimization strategies is summarized in Table 6.2. All computations were performed on a Cray Y-MP supercomputer.

Two procedural points are in order here. First, the termination criterion for the CFD analyses of this design problem was the execution of 650 cycles for the ADI solver or three Newton iterations for the direct inversion Newton's method. Both practices were necessary to consistently yield CFD residual $L_2$-norms smaller than $1.0 - 06$, which ensures a sufficiently converged flowfield solution. Second, the $\Delta \alpha$ step sizes [cf. Eq. (2.13)] were chosen such that objective functions identical to four significant digits were
obtained. This rather stringent requirement is necessary to obtain an almost identical final ramp shape for each optimization strategy. (This is because the present objective function is a weak nonlinear function of the ramp's surface shape.) Final optimization results prove to be quite sensitive to the chosen $\Delta \alpha$ step size since $\Delta \alpha$ directly effects the accuracy of the approximate flow analyses (cf. Fig. 2.1). In particular, for each strategy, doubling of the $\Delta \alpha$ step size: 1) causes the objective function to deviate in the fourth decimal point; 2) produces moderately different final shapes; and 3) causes a significant degradation of performance for Newton's method (due to the receipt of inferior initial guesses).

Two immediate and substantial savings are realized for strategy 2, in which the number of design variables were reduced from 47 to 6. First, a 4.3 MWord reduction of memory was achieved due to the smaller sizes of the arrays associated with the grid sensitivity terms. Second and more significantly, the CPU cost of each approximate flow analysis within the one-dimensional search was decreased by 60 percent. This is directly due to the reduction of the number of right-hand-sides when solving the direct differentiation formulation of the sensitivity equation [Eq. (2.19)]. After including the additional miscellaneous computational overhead to the actual cost (0.071 sec/RHS) of a forward and backward substitution, the effective CPU cost for each RHS was approximately 0.6 sec/RHS, which is a non-trivial cost when $NDV$ is large.

Shown in Fig. 6.2 is the typical convergence behavior of Newton's method during the strategy 3 optimization process. The observed quadratic rate of convergence indicates that good initial solutions are being provided to Newton's method; hence, it is concluded that approximate flow analysis is adequately predicting intermediate flow solutions within the one-dimensional searches. The slow and expensive rate of convergence of the ADI method is evidenced by the substantial time savings of strategy 3, which employs the Newton method.
The time histories of the objective function during the optimization process are shown in Fig. 6.3. The symbols denote the objective function value at the beginning of each design iteration. Note the very regular and asymptotic paths of the objective functions toward their final values. This indicates that the optimizer is receiving accurate sensitivity gradient information for each optimization strategy.

Figure 6.4 provides a qualitative comparison of the final ramp shapes and their corresponding Cp distributions. Note the final locations of the Bezier control points that define the ramp surface. The slight discrepancies between strategy 1 and strategies 2 and 3 may be due to an insufficient number of Bezier control points representing the initial expansion region of the ramp. The most significant observation is that practically identical surface shapes are obtained that are independent of the method used to represent the surface.

An interesting feature of a shape optimization process is the evolution of the surface from its initial shape to its final optimized shape. In Fig. 6.5, the manner in which the ramp shape approaches its optimum appears to be markedly different between strategy 1 and strategy 2. However, a closer examination reveals that this is not true; in fact, many similarities exist in the evolution of the shape. First, note that the predominant geometric feature influencing the magnitude of axial thrust is the initial expansion angle at the throat exit. For the surface representation using local relative slopes (strategy 1), it is observed in Fig. 6.5a that the ramp first systematically approaches the optimal expansion angle before beginning to display any concavity in its shape. This indicates that the physically most influential design variables (i.e., the relative slopes nearest the expansion) are the first to be driven to their optimal values during the optimization. In the final design iterations, the shape is then “fine-tuned” by including influences from the rest of the design variables. This same proposition applies to the Bezier formulation of strategies 2 and 3 (Fig. 6.5b). Note that the physically most influential Bezier control points are the ones nearest the ramp’s initial expansion. Again, the correct expansion
angle is attained during the first few design iterations, and the remaining design iterations allow the ramp tip to adjust to its final shape via the physically least influential Bezier control points (i.e., those which define the aft tip section).

For strategy 2, this systematic deformation from initial to final shape is better observed in Fig. 6.6. In the first design iteration, the physically most influential design variables \((D_1, D_2, D_3)\) are generously moved toward their final values, whereas, the least influential design variables \((D_4, D_5, D_6)\) change to a lesser degree. The fine-tuning process is initiated during the second design iteration and continues until the least influential design variables "damp out" to their final values. Finally, note that the side constraints (0 and 1) are not encountered during the optimization process.

In summary, results from this first design problem suggest two *modi operandi* for improving the efficiency of aerodynamic shape optimization.

1. It is recommended that a Newton method be used in lieu of an ADI method to calculate the highly converged flow solutions needed prior to the sensitivity analysis (cf. step 2a of Table 5.1). This practice was observed to reduce the CPU time by 50 percent.

2. The use of a Bezier-Bernstein representation of the surface is recommended due to observed reductions in both CPU time and computational memory. Both savings are directly attributable to a decrease in the number of design variables.

For this design problem, a factor of eight decrease in the computational time for the optimization process was achieved by implementing both of these recommendations.

6.1.2 Transonic Airfoil

The main impetus behind this design problem is to *further* improve the efficiency of aerodynamic shape optimization. This is accomplished by critically examining the implicit solution methodologies used within the design procedure.
However, the choice of problem—transonic airfoil design—is not without purpose. This design problem is one which involves highly nonlinear physics, namely, inviscid transonic flow with shocks. There exists a very strong interaction of the flowfield with the surface boundary, i.e., the location of the shock wave is extremely sensitive to the airfoil shape. Due to its nonlinearity, this particular design problem should confirm the robustness as well as expose any deficiencies of the present design procedure.

For this problem, strategies 3 to 6 of Table 6.1 are applied toward the shape optimization of the upper and lower surfaces of an initially symmetric (NACA-0012) airfoil at zero degrees angle-of-attack. The airfoil is optimized for three different Mach numbers, which are 0.60, 0.75, and 0.80. The computational domain about the airfoil consists of a 121x33 C-type grid. Figure 6.7 shows the pertinent information of the optimization problem as well as the initial NACA-0012 profile that is parameterized using 16 Bezier control points. In order to maintain a fixed angle-of-attack, the Bezier control points defining the leading and trailing edge points are spatially held fixed. The y-coordinates of the remaining 14 interior Bezier control points are taken to be the geometric-type design variables ($NDV=14$).

The constraints of the present design problem (see Fig. 6.7) are formulated based on the general design guidelines for supercritical airfoils as outlined in Ref. [106]. The lift constraint corresponds to a representative lift coefficient of transonic transports operating at cruise conditions. The wave drag constraint is arbitrarily, but reasonably chosen. The $C_p$ constraint ensures that the upper-surface pressure at 83 percent chord remains subcritical; this serves two purposes: 1) to locate the upper-surface shock at approximately three-quarter chord and 2) to produce near-sonic flow conditions immediately behind the shock. The geometric constraint on the trailing-edge included-angle prevents the formation of a very sharp and thin trailing-edge. Finally, it is observed that the present design formulation has three aerodynamic constraints ($NCONQ=3$) and
14 design variables ($NDV=14$). This suggests that the adjoint variable formulation of the sensitivity equation should produce the sensitivity coefficients most efficiently.

The final optimized shape and its corresponding $C_p$ distribution for the design Mach number 0.75 are displayed in Fig. 6.8. The optimized design shape is that of a supercritical airfoil, which is characterized by reduced curvature of the middle region of the upper-surface, substantial aft camber, an extended upper-surface pressure plateau, and a sonic pressure plateau behind the shock wave [1]. Remarkably, a geometric feature of the latest supercritical phase 3 airfoil designs that improves their low-speed performance characteristics, appears in the optimized design, namely, an undercutting of the forward lower-surface which results in an effectively smaller leading edge radius [106]. It may be clearly inferred from Fig. 6.8 that the geometrically most influential design variables (i.e., Bezier control points) are those which define the aft section of the airfoil. Note how the design variables work in conjunction with one another to form a very realistic airfoil shape. The smoothness of the airfoil surface profiles is evidenced by the fact that no non-physical discontinuous flow features are present in the corresponding $C_p$ distributions.

The evolution of the shape optimization is illustrated in Fig. 6.9, which plots the intermediate design shapes and pressure distributions corresponding to the beginning of selected design iterations. Observe that within the first few design iterations the design takes on supercritical airfoil shapes and clearly develops transonic flow structures. It is observed that the upper-surface shock moves aft and peaks in strength (at design iteration 9) while simultaneously the aft lower-surface attains its optimal shape. In the final design iterations, the design is fine-tuned, i.e., the nose radius decreases, the aft upper-surface curvature decreases, the upper-surface pressure plateau forms, and the shock decreases strength and locates as far aft as the $C_p$ constraint will permit.

The optimization strategies 3 to 6 were applied to the Mach 0.75 airfoil design problem of Fig. 6.7. Each proposed strategy was run (on a Cray Y-MP) with a prescribed
set of constant $\Delta\alpha$ step sizes. Table 6.3 presents a detailed summary of the computational statistics for these cases. Five major points can be drawn:

1. Strategies 5 and 6 require significantly less CPU time than strategies 3 and 4.
2. The memory requirements of strategy 6 are reduced by a factor of 4 as compared to the other strategies.
3. The final optimization results are dependent upon the selected $\Delta\alpha$ step size.
4. Strategy 4 is the most CPU intensive of the four strategies and becomes prohibitively expensive for small $\Delta\alpha$ step sizes.
5. For large $\Delta\alpha$ step sizes, strategy 3 fails to converge to an optimal design.

The underlying explanation of point 1 is as follows. The primary reason for the factor of five to ten reduction of CPU times for strategies 5 and 6 over the other two strategies is solely due to the remarkably low cost of the CFD analyses (which are based on the preconditioned iterative Newton's method). The impact of these inexpensive CFD analyses is significant since 75 to 97 percent of the total CPU time is expended in the calculation of steady state flow solutions. It is incidentally noted that the one-dimensional searches account for 60 to 90 percent of the total CFD analysis costs.

The underlying explanation of point 2 is as follows. Strategy 6 is totally built around the low memory preconditioned iterative linear system solvers. The key factor responsible for the large reduction of runtime memory is the successful use of the second degree iterative method for solution of the sensitivity equation. However, this memory savings is not without computational penalty—each sensitivity analysis evaluation requires approximately 30 percent more CPU time as compared to the direct inversion method. This additional effort is due to the requirement of a complete solution cycle for each RHS vector. For the present adjoint variable formulation, the number of RHS vectors is $NCONQ+1=4$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The underlying explanation of point 3 is as follows. From Table 6.3, it is apparent that many aspects of the optimization procedure are very dependent on the selected \( \Delta \alpha \) step size. As the \( \Delta \alpha \) step size is increased, the design space is traversed by larger increments as dictated by Eqs. (2.9) and (2.13). This directly leads to fewer iterations within the one-dimensional searches and fewer optimization design iterations (since the present procedure begins a new design iteration after a maximum of 50 one-dimensional search iterations). As the design deviations become larger, progressively worse initial guesses are provided to the Newton method, and consequently the CFD analyses become increasingly more expensive. The only statistics of Table 6.3 that remains practically independent of the \( \Delta \alpha \) step size are the CPU cost per sensitivity analysis and the runtime memory requirements.

Figure 6.10 displays the final airfoil shapes and their \( C_p \) distributions for strategy 6 and a variety of \( \Delta \alpha \) step sizes. Note that slightly different final shapes elicit significantly different aerodynamic responses. Final results which are practically independent of \( \Delta \alpha \) step size are observed for \( \Delta \alpha = 0.00005 \) (computed using strategy 6 in 3.25 Cray Y-MP hours). Only the airfoils designed using the smaller \( \Delta \alpha \) step sizes exhibit the smaller leading-edge radius characteristic of the supercritical phase 3 airfoils. These results suggest that any future applications utilizing a Bezier representation of an airfoil shape may benefit by placing more control points in the vicinity of the leading and trailing edges.

The underlying explanation of point 4 is as follows. All calculations were performed with CFD \( L_2 \)-norm convergence tolerances of \( TOL_{RF} = TOL_{1D} = 1.0 \times 10^{-9} \). Figure 6.11 provides plots of typical CFD convergence histories of a direct inversion versus a preconditioned iterative modified Newton method. Observe that quadratic convergence is eventually attained by both methods, although at significantly different total costs. Each iteration of the direct inversion Newton method is 18 times more expensive than an iteration of the preconditioned iterative method. Attempts of freezing
the Jacobians of the direct inversion Newton method in order to increase its computational efficiency did not yield any noteworthy savings. In fact, freezing the Jacobians too soon led to numerical divergence due to the nonlinearity of the problem. Within the one-dimensional searches, the approximate flow analysis, which requires only one Jacobian matrix inversion, computationally outperforms the direct inversion Newton method. This is because the latter method typically requires at least four Newton iterations (i.e., matrix inversions) to satisfy the $TOL_{1D}$ criterion; and hence for small $\Delta \alpha$ step sizes, becomes prohibitively expensive due to the large number of flow solutions required.

The underlying explanation of point 5 is as follows. Although approximate flow analysis works well for small $\Delta \alpha$ step sizes, the quality of its computed flow solutions deteriorates as $\Delta \alpha$ is increased. This is due to its inherent approximate (linear) nature. Since this technique involves only a single prediction, the prescribed convergence criterion $TOL_{1D}$ does not apply. Nevertheless, an $L_2$-norm of the CFD residual may serve as a useful indicator of the quality of the predicted flow solution. For strategy 3, the average $L_2$-norms of the steady state residual within the one-dimensional searches were $1.5E-07$, $7.1E-07$, and $12.9E-07$ for the cases with $\Delta \alpha = 0.0001$, $0.0005$, and $0.0010$, respectively. The implications of these coarse flow solutions are explained with the aid of Fig. 6.12.

The histories of the objective function and aerodynamic constraints during the optimization process are shown in Fig. 6.12. The constraint limits are shown in order to demarcate the feasible and infeasible regions of the design space. Many critical design decisions are made by the optimizer based on the aerodynamic performance information (i.e., flow solutions) that is computed within the one-dimensional searches. Figure 6.12 compares the aerodynamic information computed at the end of the one-dimensional search of each design iteration (and based on the appropriate $CFD_{yp}$ option) against the
same information at the same design point obtained from the highly converged CFD flow solution at the beginning of the sequent design iteration (cf. steps 4b and 2a of Table 5.1).

A comparison of Figs. 6.12a and 6.12b indicates that approximate flow analysis does an adequate job of predicting the flow solutions within the one-dimensional searches for $\Delta \alpha = 0.0001$. For the larger $\Delta \alpha$ step sizes, the coarse-grained behavior of the optimization process is evidenced in Fig. 6.12c, and furthermore strategy 3 fails to converge to an optimal design (see Fig. 6.12d). This is because the approximate flow analysis predicts close but critically inferior aerodynamic information. Specifically, approximate analysis indicates that the later designs satisfy all constraint relations when, in fact, the subsequent full CFD analysis indicates that the designs sometimes lie in the infeasible region of the design space. Hence, an inconsistency arises within the optimization process and leads to a limit cycle behavior, and it can be concluded that in this case the optimization is being misled by the erroneous approximate analysis flow solutions.

Additional parametric studies have indicated that consistent optimization results for this design problem are obtained only for CFD $L_2$-norm convergence criterion $1. \times 10^{-9}$. Increasing either one or both of the tolerances (i.e., $TOL_{\alpha F}$ and $TOL_{\alpha D}$) to $1. \times 10^{-8}$ lead to slightly different final results and/or limit cycles. The reason for this can be logically inferred from Fig. 6.11; namely, if a particular flow solution is deemed to be converged during one the transient saw-tooth oscillations, the quality of that solution may be questionable.

Finally, the present method is applied toward the shape optimization of the initially symmetric airfoil for two additional design Mach numbers, namely, 0.60 and 0.80. The same constraints as outlined in Fig. 6.7 are employed for each case. Both the Mach 0.60 case and the Mach 0.80 case were performed using strategy 6 with $\Delta \alpha = 0.0010$ and required 0.36 and 0.60 Cray Y-MP hours, respectively. The resulting final shapes and $C_p$ distributions (along with the Mach 0.75 case for reference) are shown in Fig. 6.13.
The Mach 0.60 airfoil optimization results in a supercritical design shape, although in a completely subcritical flow regime. The final design has much more aft camber than the higher Mach number designs.

The Mach 0.80 case results in a thinner airfoil (10.1 percent thick) with a very flat upper-surface contour. This low curvature upper-surface reduces the local velocities ahead of shock and hence minimizes the wave losses, which are approximately proportional to the local Mach number [1]. In fact, satisfaction of the drag constraint dominates the entire optimization process; hence, the lower \( C_L \) value. An interesting feature of this case is that the initial designs lower-surface shock is absent in the final design. The weakening and eventual disappearance of the lower shock is observed in Fig. 6.14, which shows the evolution the design during its shape optimization.

Summarizing, the present aerodynamic shape optimization procedure was successfully applied to a highly nonlinear problem—the design of an inviscid transonic airfoil. Beginning from a symmetric NACA-0012 shape, supercritical airfoil shapes were automatically obtained while optimizing for maximum lift. Observations drawn from this design problem include:

1. Optimization strategies that are totally based on preconditioned conjugate gradient-like solution methodologies yield significant reductions in CPU time and memory over those that employ direct inversion methods.

2. For highly nonlinear design problems, the coarse flow solutions predicted by approximate flow analyses may lead to a design limit cycle (i.e., a failure to converge to an optimal design) for large design deviations.

3. Final optimization results, both aerodynamically and computationally, may be very dependent upon the size of the design deviations (\( \Delta \alpha \) step size) within the one-dimensional searches.
6.2 Three-Dimensional Results

The practical three-dimensional design of both transonic and supersonic wings is considered in this section. Unlike many of the wing design efforts of other researchers, the optimized wings predicted in this work have final shapes that differ considerably from their initial shapes. In this section, the present design procedure is shown to obtain realistic wing designs, even when starting from very elementary initial geometries. In addition, the suitability of the design procedure for preliminary design applications is demonstrated in which non-intuitive shapes may possibly be generated.

Based on the results of the two-dimensional design cases, all of the present wing design cases employ optimization strategy 6 of Table 6.1, which exclusively utilizes the low-memory preconditioned conjugate gradient-like solution methodologies. In addition, the wing surface is represented using the wing geometry model of Section 4.4, which integrally incorporates two- and three-dimensional Bezier-Bernstein parameterizations.

6.2.1 Transonic Transport Wings

The initial transonic wing geometry is taken to be the unit wing oriented at an angle-of-attack of three degrees (\(\alpha = 3\)) and with a half-span length of 2 root chords (\(s_p = 2\)). For each design case, the wing shape is optimized for inviscid transonic flow conditions. The computational domain about the wing is a 17x17x43 C-O grid with parabolic singular lines located at the leading- and trailing-edges of the wing tip. The boundary conditions at the parabolic singular lines and the coincident wake planes are implicitly treated.

The transonic wing optimization problem is formulated as

\[
\text{maximize } \frac{C_L}{C_D} \tag{6.1}
\]

Subject to

Aerodynamic Constraints:

\[
C_L \geq G_L \quad C_D \leq G_D \tag{6.2}
\]
Geometric Constraints at 0.00, 0.53, and 0.98 semispan stations:

\[
\begin{align*}
5^\circ & \leq \theta_{0.90chord} \leq 20^\circ \\
5^\circ & \leq \theta_{0.98chord} \leq 20^\circ \\
\beta_{TE} & \leq 10^\circ
\end{align*}
\]  \quad (6.3)

where \( \theta \) is the included angle formed between the trailing-edge point and the upper- and lower-surface coordinates at the specified chord location. The angle \( \beta \) is the mean angle of deflection of the trailing-edge relative to the wing's angle-of-attack. No constraints are imposed on the wing volume or airfoil section areas. Different combinations of constraints and design variables are used to obtain different final wing shapes. The choice of aerodynamic constraint values, \( G_L \) and \( G_D \), is critical in driving the wing design toward reasonable shapes. The number of design variables will be dictated by both the choice of included wing deformation operations and the method of representation of the spanwise distributions. For all cases, the number of design variables is much greater than the number of aerodynamic constraints, therefore the adjoint variable formulation of the sensitivity equation is solved to most efficiently obtain the sensitivity coefficients.

6.2.1.1 Optimized Flexible Wing

The first wing design case is formulated to optimize a wing in Mach 0.75 flow with the primary intent of including the almost-full geometric flexibility of the wing geometry model of Section 4.4. In particular, the spanwise distributions \( chdscal \), \( thksca l \), \( trn z \), \( trnx \), and \( twst \) are represented using fourth degree Bezier-Bernstein parameterizations [cf. Eq. (4.1) and Fig. 4.4c]. For each distribution, the value of the Bezier control point located at the root section is held fixed, and the remaining four outboard control points are treated as design variables. In addition, the half-span length parameter \( spn \) is taken as a design variable. The wing's 3-deg angle-of-attack is held fixed throughout the optimization. Thus, the total number of design variables used to describe this wing is 21 (i.e., \( NDV = 21 \)). Finally, the values of \( G_L = 0.9 \) and \( G_D = \infty \) are used in the aerodynamic constraints, Eq. (6.2).
The wing optimization generates a quite unexpected shape, which bears no slight resemblance to a sea-bird's wing (Fig. 6.15). Although the structural integrity of such a shape is questionable, the design does possess some merit as a preliminary design concept. An upper-surface shock exists across the entire wing span, and the lower-surface is shock-free. The final design attains a $C_L/C_D = 6.877$ and a $C_L = 0.926$. The complete optimization required 4.58 Cray Y-MP hours and 19.6 Mwords of memory.

To further visualize the geometric subtleties of this design, the final spanwise distributions along with their corresponding Bezier control points are shown in Fig. 6.16. All design variables were given a large range of side-constraint bounds, and no side-constraints were active or violated during the optimization. The feasibility and efficiency of using Bezier representations for the spanwise distributions is clearly demonstrated. In fact, this final design suggests that the degree of geometric flexibility of the wing needs to be reduced in order to produce more realistic results. Historically, this is not a type of correction commonly called for in three-dimensional wing design procedures.

6.2.1.2 Realistic Transport Wing

For this reduced flexibility wing design case, the complete optimization is carried out in three distinct stages. Each stage yields an optimized design for the given problem formulation. The optimization problem for stage 1 is identical to that of the previous case except that the distributions $t_r n z$, $t r n x$, and $t w s t$ are here represented using a linear root-to-tip variation [cf. Eq. (4.20) and Fig. 4.4a]. The linear distribution of $t_r n z$ is equivalent to the specification of wing dihedral, and $t r n x$ now effectively dictates the sweep angle of the wing's quarter-chord line. Upper side-constraints are placed on the span length and tip twist angle; namely, $s p n$ must be less than 2.5 root chords, which is typical of transport wings, and $t w s t$ at the tip must be less than $+0.1$-deg, which prevents severe wash-in of the wing tip. Stage 2 of the optimization is simply a continuation of stage 1, but with $G_L = 0.35$. The number of design variables for both the first and second
stages is 12 (i.e., \( NDV = 12 \)). Stage 3 incorporates a more general airfoil definition by replacing the \( \text{tkecal} \) distribution with 3D Bezier-Bernstein parameterizations of both the upper and lower wing surfaces [cf. Eq. (4.3) and Fig. 4.1]. The values of \( G_L = 0.9 \) and \( G_L = 0.04 \) are used in the aerodynamic constraints, Eq. (6.2). The number of design variables for the third stage is 58 (i.e., \( NDV = 58 \)).

The final optimized wing shape of the Mach 0.75 design is shown in Fig. 6.17. The geometrical features of the wing include an aspect ratio of 9.71, a taper ratio (tip-chord/root-chord) of 0.31, and a quarter-chord sweep angle of 9.6-deg. The optimized wing dihedral is +2.05-deg. The linear twist distribution is superimposed onto the wing's +3-deg angle-of-attack and results in angles of incidence of +3.000-deg at the root and +3.095-deg at the tip. Figure 6.17b indicates that supercritical airfoil sections exist along the half-span length, which is 2.5 root chords long. The wing exhibits the following airfoil section thicknesses (t/c): 11.7% at 0.0 semispan, 8.2% at 0.28 semispan, 4.1% at 0.63 semispan, and 4.2% at 0.95 semispan. Figure 6.17c shows that the wing tip was treated in a consistent and realistic manner. The only active geometrical-related constraints of the final design (none were violated) include the tip twist upper side-constraint, the span upper side-constraint, and the minimum \( \theta_{0.98\text{chord}} \) geometrical constraint at the wing tip station. Other than these influences, the wing shape was not biased in any geometrical way to attain this realistic and useful final optimized design.

The aerodynamic flowfield generated by this wing is no less impressive. The surface pressure contours (\( \Delta \text{Cp} = 0.071 \)) and selected \( \text{Cp} \) distributions are shown in Fig. 6.18. An upper surface shock lies at approximately 65 percent chord along the majority of the span and then weakens and disappears at the far outboard stations. The lower surface elicits a well behaved, shock-free flow pattern. The three-dimensional character of the flowfield is clearly observed. The optimized Mach 0.75 wing at 3-deg angle-of-attack attains a \( C_L/C_D = 17.778 \) and a \( C_L = 0.794 \).
The history of the aerodynamic coefficients during the optimization process is shown in Fig. 6.19. The corresponding evolution of the wing planform shape for stages 1 and 2 (the planform shape only minutely changed during stage 3) is shown in Fig. 6.20. The choice of maximizing $C_L/C_D$ combined with a violated $C_L$ constraint proved to best provide an optimization search direction that led to non-trivial wing shapes. This combination kept $C_D$ low without the explicit need for a drag constraint. Other objective function/constraint combinations generally resulted in poor designs due to the gradient-based optimizer being prematurely "stranded" at a local maximum or terminated by conflicting constraints. By relaxing the $C_L$ constraint in stage 2, the design method was briefly free to significantly increase $C_L/C_D$ in an unconstrained optimization. The primary geometric changes observed during stage 2 were an overall thinning of the wing thickness, which reduced drag substantially, and an increase in the taper ratio (see Fig. 6.20b). Stage 3 allowed for the formation of arbitrary airfoil section shapes due to the use of the 3D Bezier parameterizations of the wing surfaces. Significant increases in $C_L$ are observed as the supercritical airfoil shapes were formed. Attempts to include the 3D Bezier surface parameterization from the beginning of the optimization resulted in poor designs having "near-unit wing" planforms but with supercritical airfoil shapes. This is because the sensitivity coefficients associated with the airfoil section design variables overwhelmed the comparatively lesser influences associated with the other wing deformations. Finally, note that the final wing design equally violates both of the stage 3 aerodynamic constraints; this typically occurs if conflicting violated constraints "compete" with one another.

The computational aspects of this design case deserve detailed consideration. The complete optimization required 35 design iterations, each of which calls for a sensitivity analysis. A total of 322 highly converged three-dimensional CFD analyses were performed during the optimization; this includes 35 $CFD_{\varphi}$ and 287 $CFD_{\varphi D}$ analyses. The CFD flow solutions were converged to residual $L^2$-norms of $TOL_{\varphi} = 1.0 \times 10^{-9}$ and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
$TOL_{1D} = 1.0 \times 10^{-8}$. Each $CFD_{VF}$ required 118.6 Cray Y-MP seconds; each $CFD_{1D}$ required 75.4 seconds; and each sensitivity analysis required 283.2 seconds. The complete optimization required a total of 10.26 Cray Y-MP hours. Thus, the total CPU time may be accounted for by the following percent usage: $CFD_{1D} = 59\%$ of the total CPU time, sensitivity analyses = 27\%, $CFD_{VF} = 11\%$, and the remaining 3\% was expended on the optimization algorithm and I/O operations. The required memory was 18.3 Mwords for stages 1 and 2 ($NDV = 12$) and 29.8 Mwords for stage 3 ($NDV = 58$).

It is found that the computational efficiency of the present design method critically depends on the use of the low-memory preconditioned conjugate gradient-like based solution methods to provide inexpensive solutions to both the 3D sensitivity equation and the fully implicit CFD equation.

However, the most noteworthy aspect of the present design procedure is its demonstration of the essential role that discrete sensitivity analysis plays in the development of an efficient and practical three-dimensional design procedure that involves a large number of design variables. If a finite-difference approach had been adopted for the calculation of the sensitivity coefficients, the total CPU time required for the sensitivity analyses alone is estimated to be 28 Cray Y-MP hours!

6.2.1.3 Multi-Point Transport Wing

One strength of design approaches that are based on sensitivity analysis and numerical optimization is the potential to perform multi-point design using a suitable multi-point objective function [62]. In this work, however, a simpler approach is adopted to develop an improved wing design, namely, shape-averaging [107]. In particular, beginning with the stage 2 final wing shape, a stage 3 optimization is repeated for Mach 0.80 flow. The resulting Mach 0.80 final design shape is averaged with the Mach 0.75 final design shape to give a multi-point wing design. The Mach 0.80 final design differs from the Mach 0.75 final design primarily in having decreased chordwise camber for
each airfoil section. Figure 6.21 shows the performance curves for the three wings. In Fig. 6.21b, it is noted that the shape-averaged (multi-point) wing outperforms its parent designs for a wide range of \( C_L \). From Fig. 6.21c, the shape-averaged wing retains relatively high \( C_L \) at the lower Mach numbers, exhibits a delayed drag divergence, and closely follows the Mach 0.80 design curve at the higher Mach numbers.

6.2.2 Supersonic Delta Wings

Two supersonic design cases are briefly examined in this section: 1) the design of a Mach 1.62 asymmetric delta wing, and 2) the design of a Mach 1.5 cranked delta wing.

For all supersonic wing design cases, the initial wing geometry is taken to be a clipped delta wing with NACA-0004 cross-sections, a 65-deg leading-edge sweep angle, and oriented at three degrees angle-of-attack (see Fig. 6.22). The computational domain is a 17x17x43 C-O grid with parabolic singular lines located at the leading- and trailing-edges of the wing tip. In fact, this initial delta wing is derived from the “unit wing” of Fig. 4.3 with appropriate spanwise linear distributions of chord (chdsca1) and thickness (thkscal) scales.

6.2.2.1 Asymmetric Delta Wing

For this design case, the wing surface model incorporates the asymmetric shearing transformation of Wood and Bauer [108]. In particular, a symmetric wing shape is first defined using a three-dimensional Bezier-Bernstein parameterization, and then asymmetry is introduced through a scalar “shearing” parameter that imposes constant asymmetry over the whole wing. During the optimization, the planform shape does not change, and no geometric twist is permitted. The design variables consist of one scalar asymmetry parameter and the z-components of 15 Bezier control points (i.e., \( NDV = 16 \)).

The optimization problem formulation consists of minimizing \( C_D \) subject to no aerodynamic constraints and 15 geometric constraints. The geometric constraints include

\[
V_{\text{wing}} \geq v_{\text{initial}} \quad A_{\text{root}} \leq 0.75 a_{\text{initial}} \quad A_{\text{tip}} \geq A_{\text{tip}} \quad (6.4)
\]
and at the 0.00, 0.53, and 0.98 semispan stations:

\[ 2^\circ \leq \theta_{0.90\text{chord}} \leq 20^\circ \]  \[ 2^\circ \leq \theta_{0.98\text{chord}} \leq 20^\circ \]  \hspace{1cm} (6.5)

where \( V \) denotes wing volume, \( A \) denotes wing section area, and \( \theta \) is the trailing-edge included-angle. The purpose of the volume constraint and the wing tip area constraint is to keep the wing from becoming too thin. The root station area constraint forces the redistribution of wing volume to the outboard stations.

Figure 6.23 shows the optimized wing design for Mach 1.62 flow conditions. This optimization required 4.7 Cray Y-MP hours and 19.6 Mwords of memory. From Fig. 6.23a, the wing displays a “near-biconvex” airfoil shape at the root station [i.e., the maximum thickness (\( \frac{t}{c} = 3.6\% \)) is located at 0.59 chord]. The asymmetric shearing transformation is clearly evidenced at the outboard stations. During the optimization the inviscid drag coefficient (objective function) was reduced from 0.0115 to 0.0107 (6.9%); however, \( C_L / C_D \) also decreased from 9.38 to 7.99 (14.7%). From Fig. 6.23b, the pressure contours indicate a reduced leading-edge expansion followed by a stronger inboard recompression in the spanwise direction as compared to that of the initial wing (Fig. 6.22d). Both of these effects would tend to decrease lift of the final wing design. In Fig. 6.23c, the present optimized geometry is compared against the empirically determined “natural flow” wing design of Ref. 108. Note that both geometries display some similar features: namely, constant leading-edge radii along the entire wing span and large areas of rearward-facing slopes on the lower surface. The primary geometric differences between the two wings may be attributed to: 1) the different approaches of modeling the wing’s trailing-edge; 2) the smaller amount of wing volume redistribution to the outboard stations in the present design; and 3) the further aft location of the root station’s maximum thickness in the present design.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
6.2.2.2 Cranked Delta Wing

The design variables of this optimization problem allow for the formation of planform breaks in the spanwise linear distributions of chord, thickness, sweep, and geometric twist. NACA-00xx wing sections are maintained throughout the optimization, and the root station's NACA-0004 wing section remains unchanged.

The optimization problem formulation consists of maximizing $C_L/C_D$ subject to one aerodynamic constraint $C_L \geq 0.14$ and 14 geometric constraints. The geometric constraints include:

$$V_{\text{wing}} \geq 0.9 V_{\text{wing}}^{\text{initial}}$$

$$A_{\text{midspan}} \geq 0.6 A_{\text{midspan}}^{\text{initial}}$$

(6.6)

and the same trailing-edge included angle constraints of Eq. (6.5).

The optimized cranked delta wing design is shown in Fig. 6.24. The geometric features of the wing include a $67^\circ$ to $36^\circ$ leading-edge sweep, $+0.1$-deg twist at the midspan and tip stations, and thickness-to-chord ratios of 2.9% at the midspan and 1.1% at the wing tip. Not all constraints were satisfied in the final design; in particular,

$$C_L = 0.122, \quad V_{\text{wing}} = 0.87 V_{\text{wing}}^{\text{initial}}, \quad \theta_{0.98\text{chord}} = 1.74^\circ.$$  

The final design has a $C_L/C_D = 10.22$ and $A_{\text{midspan}} = 0.61 A_{\text{midspan}}^{\text{initial}}$. The optimization required 2.1 Cray Y-MP hours and 17.5 Mwords of memory ($NDV = 9$).

A second cranked delta wing optimization problem was formulated in order to produce a wing design having a smaller midspan chord length. In particular, $C_L/C_D$ was maximized with no aerodynamic constraints and with the geometric constraints:

$$\text{chord}_{\text{midspan}} \geq 0.3 \text{chord}_{\text{root}}^{\text{initial}}$$

$$A_{\text{midspan}} \geq 0.3 A_{\text{midspan}}^{\text{initial}}$$

(6.7)

and the same trailing-edge included angle constraints of Eq. (6.5).

For this case, a cranked wing having a $73^\circ$ to $44.5^\circ$ leading-edge sweep is produced (Fig. 6.25). Additional geometric features include at the midspan station: $\nu/c = 3.9\%$, $+0.15$-deg twist; and at the tip station: $\nu/c = 6.8\%$, $-0.12$-deg twist. The final wing volume is 71 percent of the initial delta wing geometry. Figure 6.25c indicates that a low
pressure region exists over the entire upper-surface of the cranked section. This final design has a $C_L / C_D = 8.96$; hence, constraining the midspan chord reduces the Mach 1.5 cruise performance as compared to the first cranked wing design case. However, this latter wing shape (Fig. 6.25) should improve the low-speed performance over that of the first design shape. For this case, the complete optimization required 1.7 Cray Y-MP hours and 18.0 Mwords of memory (NDV = 11).
Table 6.1 Proposed Strategies for the Optimization Procedure

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Surface Representation</th>
<th>CFD$_V F$</th>
<th>CFD$_D B$</th>
<th>Sensitivity Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Point-Based</td>
<td>ADI $^c$</td>
<td>AA</td>
<td>DI</td>
</tr>
<tr>
<td>2</td>
<td>Point-Based</td>
<td>DI</td>
<td>AA</td>
<td>DI</td>
</tr>
<tr>
<td>3</td>
<td>Bezier</td>
<td>DI</td>
<td>AA</td>
<td>DI</td>
</tr>
<tr>
<td>4</td>
<td>Bezier</td>
<td>DI</td>
<td>DI</td>
<td>DI</td>
</tr>
<tr>
<td>5</td>
<td>Bezier</td>
<td>PCG</td>
<td>PCG</td>
<td>DI</td>
</tr>
<tr>
<td>6</td>
<td>Bezier</td>
<td>PCG</td>
<td>PCG</td>
<td>PCG</td>
</tr>
</tbody>
</table>

$^a$ CFD methodology used prior to the sensitivity analysis (cf. step 2a of Table 5.1).

$^b$ CFD methodology used within the one-dimensional search (cf. step 4c of Table 5.1).

$^c$ Nomenclature: ADI = Alternating Direction Implicit
AA = Approximate Analysis
DI = Direct Inversion
PCG = Preconditioned Conjugate Gradient-like
<table>
<thead>
<tr>
<th>Strategy</th>
<th>$\Delta \alpha$</th>
<th>Total CPU [hour]</th>
<th>Runtime Memory [MWord]</th>
<th>Number of Opt. Design Iterations</th>
<th>CPU per CFD$_{VF}$ [sec]</th>
<th>CPU per Sensitivity Analysis [sec]</th>
<th>Number of 1-D Search Iterations</th>
<th>CPU per CFD$_{1D}$ [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0005</td>
<td>8.05</td>
<td>11.7</td>
<td>14</td>
<td>313.0</td>
<td>15.5</td>
<td>599</td>
<td>40.6</td>
</tr>
<tr>
<td>2</td>
<td>0.0050</td>
<td>2.12</td>
<td>7.4</td>
<td>15</td>
<td>313.0</td>
<td>15.5</td>
<td>172</td>
<td>15.8</td>
</tr>
<tr>
<td>3</td>
<td>0.0050</td>
<td>0.97</td>
<td>7.4</td>
<td>14</td>
<td>40.6</td>
<td>15.5</td>
<td>166</td>
<td>15.8</td>
</tr>
</tbody>
</table>
Table 6.3 Computational Statistics for the Transonic Airfoil Optimization

<table>
<thead>
<tr>
<th>Strategy</th>
<th>$\alpha$ Step Size $\Delta \alpha$</th>
<th>Final Design Lift Coefficient</th>
<th>Total CPU Runtime [hour]</th>
<th>Total Memory [MWord]</th>
<th>Number of Opt. Design Iterations</th>
<th>CPU per CFD$_{VF}$ Analysis [sec]</th>
<th>CPU per Sensitivity Analysis [sec]</th>
<th>Number of 1-D Search Iterations</th>
<th>CPU per CFD$_{ID}$ [sec]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.0001</td>
<td>0.8437</td>
<td>14.40</td>
<td>18.5</td>
<td>24</td>
<td>230.3</td>
<td>55.6</td>
<td>723</td>
<td>62.2</td>
</tr>
<tr>
<td>4</td>
<td>see Note a</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0.8381</td>
<td>1.86</td>
<td>18.5</td>
<td>17</td>
<td>5.0</td>
<td>55.6</td>
<td>640</td>
<td>8.8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.8364</td>
<td>1.78</td>
<td>4.7</td>
<td>16</td>
<td>6.1</td>
<td>70.6</td>
<td>626</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.8261</td>
<td>3.28</td>
<td>18.5</td>
<td>9</td>
<td>434.0</td>
<td>55.6</td>
<td>121</td>
<td>62.2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0005</td>
<td>0.8197</td>
<td>6.47</td>
<td>8</td>
<td>164.4</td>
<td>55.6</td>
<td>130</td>
<td>164.4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8219</td>
<td>0.53</td>
<td>18.5</td>
<td>6</td>
<td>16.3</td>
<td>55.6</td>
<td>107</td>
<td>13.7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.8218</td>
<td>0.69</td>
<td>4.7</td>
<td>6</td>
<td>17.3</td>
<td>72.8</td>
<td>107</td>
<td>17.9</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>see Note b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0010</td>
<td>0.8257</td>
<td>4.66</td>
<td>6</td>
<td>242.3</td>
<td>55.6</td>
<td>61</td>
<td>236.8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8255</td>
<td>0.44</td>
<td>18.5</td>
<td>6</td>
<td>24.0</td>
<td>55.6</td>
<td>59</td>
<td>18.2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.8251</td>
<td>0.57</td>
<td>4.7</td>
<td>6</td>
<td>33.6</td>
<td>74.1</td>
<td>59</td>
<td>23.3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>see Note b</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0020</td>
<td>0.7828</td>
<td>3.65</td>
<td>5</td>
<td>362.7</td>
<td>55.6</td>
<td>30</td>
<td>349.8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.7826</td>
<td>0.26</td>
<td>18.5</td>
<td>4</td>
<td>27.8</td>
<td>55.6</td>
<td>28</td>
<td>21.0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.7824</td>
<td>0.32</td>
<td>4.7</td>
<td>4</td>
<td>28.4</td>
<td>69.4</td>
<td>28</td>
<td>26.7</td>
<td></td>
</tr>
</tbody>
</table>

*a* Case deemed too computationally expensive to run. Estimated CPU time = 30 Cray Y-MP hours.

*b* Strategy 3 does not converge to an optimum design, but enters a limit cycle.
Maximize $T_X$
Subject to:
$P_r > 0.55 P_\infty$
$1.3 P_\infty < P_c < 2.3 P_\infty$

Fig. 6.1 Formulation of the nozzle shape optimization problem.
Fig. 6.2 Convergence history of Newton’s method for the nozzle optimization.

Fig. 6.3 Objective function history for the nozzle optimization.
Fig. 6.4 A comparison of the optimized nozzle designs.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
(a) Strategy 1 (design variables \(\equiv 47\) local relative slopes)

(b) Strategy 2 (design variables \(\equiv 6\) Bezier control points)

Fig. 6.5 Evolution of the optimized nozzle shape.
Fig. 6.6 History of strategy 2 design variable values for the nozzle optimization (side constraints at 0 and 1).
Maximize $C_L$

Subject to

Aerodynamic Constraints:

$C_L \geq 0.75 \quad C_D \leq 0.01 \quad C_p \text{ @ 0.83 Chord } \geq C_p^*$

Geometric Constraints:

$10^\circ \leq \theta_{TE} \leq 25^\circ$

Fig. 6.7 Formulation of the airfoil optimization problem.
Fig. 6.8 Final optimized airfoil shape and Cp distribution for Mach 0.75.
Fig. 6.9 Evolution of the optimized Mach 0.75 airfoil design for strategy 6 and $\Delta\alpha = 0.0001$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Fig. 6.10 A comparison of strategy 6 optimized Mach 0.75 airfoil designs for various $\Delta\alpha$ step sizes.
Fig. 6.11 A comparison of typical CFD convergence histories for two different Newton method solution strategies.
Fig. 6.12 Histories of the objective function and aerodynamic constraints during the Mach 0.75 airfoil optimization process.
Fig. 6.13 A comparison of optimized airfoil designs for three different design Mach numbers.
Fig. 6.14 Evolution of the optimized airfoil design for Mach 0.80.
(a) Half-planform view

(b) Perspective view

(c) Upper-surface pressure contours

Fig. 6.15 An optimized flexible wing: $M_\infty = 0.75, \alpha = 3.0\text{-deg.}$
Fig. 6.16 Parameter distributions for the optimized flexible wing.
Fig. 6.17 Optimized design of a transport wing: $M_\infty = 0.75$, $\alpha = 3.0$-deg.
Fig. 6.18 Surface pressure contours and Cp distributions for the transport wing design: 
$M_{\infty} = 0.75, \alpha = 3.0$-deg.
Fig. 6.19 History of the aerodynamic coefficients for the Mach 0.75 transport wing optimization.
Fig. 6.20 Evolution of the planform shape during the Mach 0.75 transport wing optimization.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Fig. 6.21 Performance curves for three transonic wing designs.
Fig. 6.22 Initial supersonic delta wing geometry.
Fig. 6.22 Concluded.

(e) Surface elevation cuts at $\alpha = 0$-deg

(d) Surface pressure contours: $M_\infty = 1.62$, $\alpha = 3$-deg
Fig. 6.23 Optimized asymmetric supersonic delta wing: $M_\infty = 1.62$, $\alpha = 3$-deg.
The "natural flow" wing of Wood and Bauer [108]

(c) Surface elevation cuts

Fig. 6.23 Concluded.
Fig. 6.24 Optimized cranked supersonic delta wing: $M_\infty = 1.5, \alpha = 3$-deg.
Fig. 6.24 Concluded.

(d) Surface elevation cuts at $\alpha = 0$-deg

(c) Surface pressure contours: $M_{\infty} = 1.5$, $\alpha = 3$-deg

Upper-surface

Lower-surface
Fig. 6.25 Optimized cranked supersonic delta wing with mid-span chord constraint: $M_\infty = 1.5$, $\alpha = 3$-deg.
Fig. 6.25 Concluded.

(c) Surface pressure contours: $M_\infty = 1.5$, $\alpha = 3$-deg

(d) Surface elevation cuts at $\alpha = 0$-deg
Chapter 7
SUMMARY AND CONCLUSIONS

The latest developments toward constructing an efficient and functional three-dimensional aerodynamic shape optimization procedure have been reported. The present work is shown to offer significant advancements over the "early design methodologies," including: 1) all CFD solutions are obtained by solving the Euler equations using a fully implicit algorithm; 2) the optimization gradient information is computed using discrete aerodynamic sensitivity analysis; and 3) the design surface geometry is modeled by using both two- and three-dimensional Bezier-Bernstein parameterizations. The high computational efficiency of the present design procedure is due to the use of sensitivity analysis, which permits the efficient treatment of a large number of design variables, and also due to the exclusive use of low-memory preconditioned conjugate gradient-like methodologies to solve the fully implicit fluid dynamic equation and the three-dimensional sensitivity equation. Proper preconditioning is found to be a vital element in achieving stable and convergent implicit solution algorithms for the present three-dimensional unfactored linear algebraic systems.

This work presents for the first time many practical numerical issues related to the integration of the separate elements of aerodynamic shape optimization into a functional three-dimensional design procedure. Such issues pertain to the numerical aspects of CFD, sensitivity analysis, and implicit solution methodologies within the overall design procedure. Toward this end, the major findings of the present work include:

(1) Newton methods are viable, effective, and preferable alternatives to ADI schemes for the numerous CFD analyses;
(2) optimization strategies that are totally based on preconditioned conjugate gradient-like solution methodologies yield significant reductions in CPU time and memory over those that employ direct inversions methods;

(3) a necessary and key element in obtaining solutions to the present three-dimensional linear algebraic systems is that the preconditioning matrix be based on a diagonally-dominant coefficient matrix;

(4) for highly nonlinear problems, the coarse flowfield solutions predicted by approximate flow analyses may lead to a failure to converge to an optimal design; and

(5) final optimization results, both aerodynamically and computationally, may be very dependent upon the size of the design deviations within the one-dimensional searches.

Elaborating, one of the critical findings and contributions of the present work is that a fully implicit CFD formulation (i.e., Newton's method) turns out to be a very practical and useful component of three-dimensional design procedures. This is due to a unique circumstance that arises within the present design process, namely, neighboring designs along with their corresponding flow solutions are incrementally "close" to one another. Hence, in this circumstance, a Newton's method may be used to update the flowfield solution for each new design, and furthermore it is found that the Newton's method almost always lies within its domain of attraction (i.e., yields quadratic convergence to a steady state). This efficient strategy is convincingly demonstrated in one of the present design applications that required 322 highly converged three-dimensional CFD analyses (convergence $L_2$-norms $\leq 1.0 \times 10^{-8}$). Here each CFD analysis required an average of only 80 Cray Y-MP seconds. Finally, this study performs the first known fully implicit three-dimensional CFD analysis that is based on an exact Newton-linearization (i.e., a second-order LHS operator).
The present work extends the current state-of-the-art for surface representation in
direct-design procedures. Novel two- and three-dimensional Bezier-Bernstein
parameterizations of the design surface have been developed and successfully
demonstrated. In particular, a flexible wing geometry model has been developed in
which very general wing shapes may be generated by applying a sequence of geometrical
deformations that are based on five spanwise parameter distributions and four scalar
parameters. When used within the present design procedure, this model was
demonstrated to produce non-intuitive preliminary design concepts and to yield
remarkably realistic wing designs as well. The effectiveness of the present two- and
three-dimensional surface representation techniques is proven in that, unlike many of the
design efforts of previous researchers, the optimized shapes predicted in this work have
final shapes that differ considerably from their initial shapes.

The present shape optimization procedure is applied toward the design of both two-
and three-dimensional inviscid flow problems including ones that involve highly
nonlinear physics—inviscid transonic flow with shocks. For two-dimensional design,
beginning from a symmetric NACA-0012 shape, supercritical airfoil shapes similar to
those of Whitcomb [1] were automatically obtained while optimizing for maximum lift.
The three-dimensional wing design applications of the present work include: 1) a realistic
transonic (Mach 0.75) transport wing whose final geometry evolved from a very
elementary initial shape; 2) a transonic transport wing based on a multi-point design
technique; 3) a supersonic (Mach 1.62) asymmetric delta wing; and 4) a Mach 1.5
cranked delta wing. Thus, the present design procedure is shown to be applicable over a
wide range of compressible flow regimes. All of the design applications of this work are
examples involving substantial shape changes; however, the present design method can
equally handle the localized shape change strategies that have been practiced by previous
numerical designers.
In conclusion, many critical issues still need to be addressed by future research efforts in the area of aerodynamic design. The consistent inclusion of turbulent flow effects into aerodynamic sensitivity analysis procedures is of the utmost importance. Until such effects are properly accounted for, truly realistic aerodynamic design is precluded. In addition, future sensitivity analysis procedures should be extended to handle the latest advances in CFD technologies including the more complex space-integration formulations (e.g., Roe's flux-difference-splitting [109] or Total Variational Diminishing (TVD) [110] schemes); convergence acceleration techniques such as mesh-sequencing or multi-gridding [111]; and methods that handle complex geometries such as grid-overlapping (Chimera) schemes [112]. Successful implementation of these types of issues will further increase the efficiency, generality, and applicability of aerodynamic sensitivity analysis, and hence increasingly move these new design tools into the mainstream of aerodynamic design optimization.
REFERENCES


A.1 Fully Implicit Formulation

The inviscid fluid dynamic equations are a first order hyperbolic system and can be written as

$$\frac{\partial Q}{\partial t} + R(Q) = 0 \quad (A.1)$$

where the steady state residual for one spatial dimension is

$$R(Q) = \frac{\partial F(Q)}{\partial x} \quad (A.2)$$

After transforming from physical \((x,t)\) space to generalized computational \((\xi, \tau)\) space, applying Euler implicit time-integration, and spatially discretizing Eq. (A.1) in a finite-volume sense, the resulting system of difference equations can be written as

$$\frac{\Delta Q_i^n}{\Delta t} + R(Q^{n+1}, M) = 0 \quad (A.3)$$

where \(n\) is the time level, \(i\) is the computational grid index, \(M\) represents the coordinate transformation metrics, and the update vector is

$$\Delta Q_i^n = Q_i^{n+1} - Q_i^n \quad (A.4)$$

The discrete steady state residual becomes (taking \(\Delta \xi = 1\))

$$R(Q^{n+1}, M) = \frac{\delta_i \hat{F}(Q^{n+1}, M)}{\Delta \xi} = \hat{F}_{i+1/2}(Q^{n+1}, M) - \hat{F}_{i-1/2}(Q^{n+1}, M) \quad (A.5)$$
In this work, the flux-vectors \( \hat{F} \) are evaluated using the flux-vector-splitting scheme of Van Leer [78]. One property of this upwind scheme is that a flux may be written as

\[
\hat{F} = \hat{F}^+ + \hat{F}^-
\]

where \( \hat{F}^\pm \) contains directionally dependent physical information. Consequently, Eq. (A.3) becomes

\[
\frac{\Delta Q_i^n}{\Delta t} + \left[ \hat{F}_{i+\frac{1}{2}}^+(Q^{n+1}, M) + \hat{F}_{i+\frac{1}{2}}^-(Q^{n+1}, M) \right] - \left[ \hat{F}_{i-\frac{1}{2}}^+(Q^{n+1}, M) + \hat{F}_{i-\frac{1}{2}}^-(Q^{n+1}, M) \right] = 0 \quad (A.7)
\]

Spatially higher-order accurate schemes may be constructed via the MUSCL formulation of Van Leer [109] in which flow variable values at the cell interface are interpolated from neighboring cell-centered values. A flux vector based on this type of cell interface interpolation is given, in general, by

\[
\hat{F}_{i \pm \frac{1}{2}}^\pm(Q, M) = \hat{F}_{i \pm \frac{1}{2}}^\pm(Q_{i \pm \frac{1}{2}}, Q_{i \pm \frac{1}{2}}, M_{i \pm \frac{1}{2}}) \quad (A.8)
\]

This flux may be evaluated at other interfaces by shifting \( i \) appropriately, e.g.,

\[
\hat{F}_{i+\frac{1}{2}}^+(Q, M) = \hat{F}_{i+\frac{1}{2}}^-(Q_{i+\frac{1}{2}}, M_{i+\frac{1}{2}}) = \hat{F}_{i+\frac{1}{2}}^-(Q_i, Q_{i+1}, M_{i+\frac{1}{2}}) \quad (A.9)
\]

Linearizing \( \hat{F}(Q^{n+1}) \) with respect to time gives, in general,

\[
\hat{F}_{i \pm \frac{1}{2}}^\pm(Q^{n+1}, M) = \hat{F}_{i \pm \frac{1}{2}}^\pm(Q_{i-1}^{n+1}, Q_i^{n+1}, Q_{i+1}^{n+1}, M_{i \pm \frac{1}{2}}^{n+1})
\]

\[
= \hat{F}_{i \pm \frac{1}{2}}^\pm(Q^n, M) + DR_{i \pm \frac{1}{2}, i-1}^\pm \cdot \Delta Q_{i-1}^n + DR_{i \pm \frac{1}{2}, i}^\pm \cdot \Delta Q_i^n
\]

\[
+ DR_{i \pm \frac{1}{2}, i+1}^\pm \cdot \Delta Q_{i+1}^n + O(\Delta t^2) \quad (A.10)
\]

where \( DR_{i \pm \frac{1}{2}, i}^\pm \) is defined, in general, by

\[
DR_{i \pm \frac{1}{2}, i}^\pm = \frac{\partial \hat{F}_{i \pm \frac{1}{2}}^\pm(Q_{i \pm \frac{1}{2}}, M_{i \pm \frac{1}{2}})}{\partial Q_i} \cdot \frac{\partial Q_{i \pm \frac{1}{2}}}{\partial Q_i} \quad (A.11)
\]

Applying the time linearization of Eq. (A.10) to Eq. (A.7) gives symbolically

\[
\left[ \frac{I}{\Delta t} + \frac{\partial R}{\partial Q} \right] \Delta Q^n = -R(Q^n, M) \quad (A.12)
\]
where the LHS of Eq. (A.12) is given by

\[
\left[ \frac{\partial R}{\partial Q} \right] \Delta Q^n = \left[ -DR^-_{i-1/2,j-1} \right] \Delta Q^n_{i-1} + \left[ +DR^+_{i+1/2,j-1} - DR^-_{i+1/2,j-1} - DR^-_{i-1/2,j-1} + DR^+_{i-1/2,j-1} \right] \Delta Q^n_{i-1} + \left[ +DR^+_{i+1/2,j+1} + DR^-_{i+1/2,j+1} - DR^-_{i-1/2,j+1} - DR^+_{i-1/2,j+1} \right] \Delta Q^n_{i+1} + \left[ +DR^-_{i-1/2,j+2} \right] \Delta Q^n_{i+2}
\]  

(A.13)

and the RHS of (A.12) is

\[
R(Q^n,M) = \hat{R}_{i+1/2}(Q^n,M) - \hat{R}_{i-1/2}(Q^n,M)
\]

(A.14)

The three-dimensional extension of Eq. (A.1) is

\[
\frac{\partial Q}{\partial t} + \frac{\partial F(Q)}{\partial x} + \frac{\partial G(Q)}{\partial y} + \frac{\partial H(Q)}{\partial z} = 0
\]

(A.15)

or discretely

\[
\frac{\Delta Q_{i,j,k}}{\Delta \tau} + \frac{\delta_i \hat{F}_{i+1/2,j,k}^{n+1,M}}{\Delta \xi} + \frac{\delta_j \hat{G}_{i,j+1/2,k}^{n+1,M}}{\Delta \eta} + \frac{\delta_k \hat{H}_{i,j,k+1/2}^{n+1,M}}{\Delta \zeta} = 0
\]

(A.16)

The discrete fully implicit linear system of Eq. (A.16) is still given symbolically by Eq. (A.12) where additional appropriate terms are included in Eqs. (A.13) and (A.14).

It now remains to define expressions for the inviscid flux vectors and also the terms of \( DR^\pm \). The flux vector terms \( \hat{F}^\pm \) and \( \partial \hat{F}^\pm / \partial Q \) are developed in Section A.2, and the cell interface terms \( Q_{i+1/2}^- \) and \( \partial Q_{i+1/2}^+ / \partial Q_i \) are developed in Section A.3.

A.2 Van Leer Flux-Vector-Splitting

The idea behind flux-vector-splitting is to construct a stable upwind differencing scheme based on the hyperbolic nature of the inviscid time-dependent fluid dynamic equations. In particular, physical information based on the sign of the eigenvalues are introduced, whereby the flux terms are split and discretized directionally according to the sign of the associated propagation speeds [114]. The flux splitting developed by Van
Leer [78] has the property of being continuously differentiable through sonic and stagnation points.

Flux-vector-splitting schemes define the flux vector as

$$\hat{F} = \hat{F}^+ (Q^-) + \hat{F}^- (Q^+)$$  \hspace{1cm} (A.17)

For supersonic flow, the full inviscid flux vector is used in an upwind fashion, i.e.,

For $\hat{M}_\xi \geq 1$

$$\hat{F}^+ = \hat{F}_{full} \quad \hat{F}^- = 0$$  \hspace{1cm} (A.18a)

For $\hat{M}_\xi \leq -1$

$$\hat{F}^+ = 0 \quad \hat{F}^- = \hat{F}_{full}$$  \hspace{1cm} (A.18b)

For subsonic flow, the Van Leer split-flux-vector is used, i.e.,

For $-1 < \hat{M}_\xi < 1$

$$\hat{F}^+ = \hat{F}_{VL} \quad \hat{F}^- = \hat{F}_{VL}$$  \hspace{1cm} (A.19)

The flux vectors of Eqs. (A.18) and (A.19) are defined as

$$\hat{F}_{full} (Q, M) = \left[ \frac{\nabla Q}{J} \right] \begin{pmatrix} \rho \hat{U} \\ \rho \hat{u} + \hat{e}_j, p \\ (\rho e_o + p) \hat{U} \end{pmatrix}$$  \hspace{1cm} (A.20)

$$\hat{F}_{VL} (Q, M) = \left[ \frac{\nabla Q}{J} \right] \begin{pmatrix} f_{\text{mass}}^x \\ f_{\text{mass}}^z \hat{g}_{\text{mom}}^z \\ f_{\text{mass}}^z \hat{g}_{\text{energy}}^z \end{pmatrix}$$  \hspace{1cm} (A.21)

For the MUSCL formulation, $Q$ and $M$ are evaluated at the appropriate cell interface. The basic nomenclature and definitions for Eqs. (A.20) and (A.21) include

$$Q = \left[ \rho, \rho u_j, \rho e_o \right]^T = \text{vector of conserved variables}$$  \hspace{1cm} (A.22a)

$$u_j = [u, v, w]^T = \text{Cartesian components of the velocity vector}$$  \hspace{1cm} (A.22b)

$$\hat{e}_j = \left[ \frac{\xi}{\left| \nabla e_j \right|}, \frac{\zeta}{\left| \nabla e_j \right|}, \frac{\eta}{\left| \nabla e_j \right|} \right]^T = \text{cell interface direction cosines}$$  \hspace{1cm} (A.22c)
\[ \frac{\left| \nabla \xi \right|}{J} = \text{cell interface area} \quad (\xi_j, \left| \nabla \xi \right| \text{ constitute the "metric" terms, } M) \quad (A.22d) \]

\[ \vec{U} = \xi_j u_j = \xi_x u + \xi_y v + \xi_z w = \text{directed contravariant velocity vector} \quad (A.22c) \]

\[ \dot{M}_\xi = \frac{\xi_j u_j}{a} = \text{cell interface normal Mach number} \quad (A.22f) \]

\[ a = \left( \frac{\gamma p}{\rho} \right)^{1/2} = \text{speed of sound} \quad (A.22g) \]

\[ p = (\gamma - 1) \left[ \rho e_o - \rho \frac{u_j u_j}{2} \right] = \text{static pressure} \quad (A.22h) \]

\[ f_{\text{mass}}^{\pm} = \pm \frac{1}{4} \rho a \left( \dot{M}_\xi \pm 1 \right)^2 \quad (A.22i) \]

\[ g_{j,\text{mom}}^{\pm} = \frac{\xi_j}{\gamma} a \left( \dot{M}_\xi \pm 2 \right) + u_j \quad (A.22j) \]

\[ g_{\text{energy}}^{\pm} = g_{\xi 1}^{\pm} + g_{\xi 2}^{\pm} \quad (A.22k) \]

\[ g_{\xi 1}^{\pm} = (\rho a)^2 \left[ (\gamma - 1) \dot{M}_\xi^2 \pm 2(\gamma - 1)\dot{M}_\xi + 2 \right] \quad (A.22l) \]

\[ g_{\xi 2}^{\pm} = \rho^2 (\gamma - 1) \quad (A.22m) \]

\[ g_{\xi 3}^{\pm} = \frac{u_j u_j}{2} = \frac{u^2 + v^2 + w^2}{2} \quad (A.22n) \]

In this work, the exact (or true) flux Jacobians are used. Here, the following nomenclature is used: \( Q_m \) will denote the components of the vector of conserved variables, i.e.,

\[ Q_m = [\rho, \rho u_j, \rho e_o]^T = [\rho, \rho u, \rho v, \rho w, \rho e_o]^T; \]
and \( \frac{\partial \hat{F}}{\partial Q_m} \) will denote the appropriate row vector of the Jacobian matrix, i.e.,

\[
\frac{\partial \hat{F}}{\partial Q_m} = \begin{bmatrix}
\frac{\partial \hat{F}}{\partial Q_1}, & \frac{\partial \hat{F}}{\partial Q_2}, & \frac{\partial \hat{F}}{\partial Q_3}, & \frac{\partial \hat{F}}{\partial Q_4}, & \frac{\partial \hat{F}}{\partial Q_5}
\end{bmatrix}.
\]

The Jacobian of the full inviscid flux vector, \( \frac{\partial \hat{F}_{\text{full}}}{\partial Q_m} \), is commonly given in many CFD texts (e.g., Ref. 115) and will not be repeated here.

The Van Leer split-flux Jacobian can be written as

\[
\frac{\partial \hat{F}_{\text{VL}}}{\partial Q_m} = \frac{[\nabla \eta]}{J} \begin{bmatrix}
\frac{\partial f_{\text{mass}} \pm \eta}{\partial Q_m} \\
\frac{\partial f_{\text{energy}}}{\partial Q_m}
\end{bmatrix} = \frac{[\nabla \eta]}{J} \begin{bmatrix}
\frac{\partial f_{\text{mass}} \pm \eta}{\partial Q_m} \\
\frac{\partial f_{\text{energy}}}{\partial Q_m}
\end{bmatrix} \tag{A.23}
\]

where

\[
\frac{\partial f_{\text{mass}}}{\partial Q_m} = \pm \frac{1}{4} \left( 1 - \dot{M}_\xi \right) \frac{\partial \rho a}{\partial Q_m} \pm \frac{1}{2} \left( \dot{M}_\xi \pm 1 \right) A_m \tag{A.24a}
\]

\[
\frac{\partial g_{\eta,\text{mom}}}{\partial Q_m} = - \frac{\rho}{\rho} \frac{\partial \rho}{\partial Q_m} \pm \frac{2 \bar{\varepsilon}_j}{\gamma} \frac{\partial \rho a}{\partial Q_m} - \frac{\xi_j}{\gamma \rho} A_m + \frac{1}{\rho} \frac{\partial \rho u_j}{\partial Q_m} \tag{A.24b}
\]

\[
\frac{\partial g_{\text{energy}}}{\partial Q_m} = \frac{1}{\varepsilon_2} \left[ \frac{\partial \beta_{\text{gas}}}{\partial Q_m} - \frac{\varepsilon_2}{\varepsilon_2} \frac{\partial \beta_{\text{gas}}}{\partial Q_m} + \frac{\partial \beta_{\text{gas}}}{\partial Q_m} \right] + \frac{\partial \beta_{\text{gas}}}{\partial Q_m} \tag{A.24c}
\]

\[
\frac{\partial \beta_{\text{gas}}}{\partial Q_m} = 2 \rho a \left\{ \pm (\gamma - 1) \dot{M}_\xi + 2 \right\} \frac{\partial \rho a}{\partial Q_m} + 2 \rho a (\gamma - 1) \left\{ - \dot{M}_\xi \pm 1 \right\} A_m \tag{A.24d}
\]

\[
\frac{\partial \beta_{\text{gas}}}{\partial Q_m} = 2 \rho (\gamma^2 - 1) \frac{\partial \rho}{\partial Q_m} \tag{A.24e}
\]

\[
\frac{\partial \beta_{\text{gas}}}{\partial Q_m} = - \frac{u_j u_j}{\rho} \frac{\partial \rho}{\partial Q_m} + \frac{1}{2 \rho^2} B_m \tag{A.24f}
\]

\[
\frac{\partial \rho}{\partial Q_m} = \frac{[1, 0, 0, 0, 0]}{\partial Q_m} \tag{A.24g}
\]
\[
\frac{\partial \rho u_j}{\partial Q_m} = 1 \quad \text{only if} \quad \rho u_j = Q_m \tag{A.24h}
\]

\[
\frac{\partial \rho a}{\partial Q_m} = \frac{\gamma (\gamma - 1)}{2 \rho a} \left\{ C_m - \frac{1}{2} B_m \right\} \tag{A.24i}
\]

\[
A_m = \frac{\partial}{\partial Q_m} \left( Q_2 \tilde{\xi}_x + Q_3 \tilde{\xi}_y + Q_4 \tilde{\xi}_z \right) = [0, \tilde{\xi}_x, \tilde{\xi}_y, \tilde{\xi}_z, 0] \tag{A.24j}
\]

\[
B_m = \frac{\partial}{\partial Q_m} \left( Q_2^2 + Q_3^2 + Q_4^2 \right) = [0, 2Q_2, 2Q_3, 2Q_4, 0] \tag{A.24k}
\]

\[
C_m = [\rho e_\rho, 0, 0, 0, \rho] \tag{A.24l}
\]

For the other spatial directions, simply replace \( \tilde{\xi} \) with \( \eta \) or \( \zeta \) for the flux vectors \( \tilde{G} \) or \( \tilde{H} \), respectively.

### A.3 Cell Interface Interpolation Formulas

Second-order spatial accuracy can be achieved by introducing more upwind points into the schemes [114]. The MUSCL approach as developed by Van Leer [113] compute the flow variables at the cell interface by interpolation between the neighboring cell-centered values. For such interpolations, formal orders of spatial accuracy may be determined from a Taylor series expansion of a flow variable around its cell-centered location [114].

In general, the flow variable value at a cell interface may be represented as

\[
Q_{i \pm 1/2, j, k} = Q_i \pm \Omega_i \tag{A.25}
\]

where \( \Omega_i \) is a correction term that is computed from local gradients of the flow variable. This formula may be evaluated at other interfaces by shifting \( i \) appropriately, e.g.,

\[
Q_{i+1/2} = Q_{i+1} - \Omega_{i+1} \tag{A.25}
\]

For higher-order accurate schemes, the \( \phi - \kappa \) interpolation polynomial may be used [116], where \( \Omega_i \) takes the form
\[ \Omega_i^+ = \frac{\phi}{4} \left( (1 + \kappa) \nabla_i + (1 \pm \kappa) \Delta_i \right) \]  
(A.26)

\[ \nabla_i = Q_i - Q_{i-1} \quad \Delta_i = Q_{i+1} - Q_i \]  
(A.27)

For first-order upwind differencing: \( \phi = 0 \); for second-order fully upwind: \( \phi = 1, \kappa = -1 \); for third-order fully upwind: \( \phi = 1, \kappa = 1/3 \); for central differencing: \( \phi = 1, \kappa = 0 \).

The straightforward replacement of the first-order upwind space differences by appropriate higher-order accurate formulas leads to numerical deficiencies, in particular, the generation of oscillations around discontinuities [114]. A nonlinear "limiter" may be introduced to control the gradients of the computed solution and thus to prevent the appearance of these over- or undershoots [114]. In this work the differentiable limiter of Van Albada [80] is used. This interface interpolation formula may be also represented by Eq. (A.25) with \( \Omega_i \) defined as

\[ \Omega_i^+ = \frac{s_i}{4} (\nabla_i + \Delta_i)^+ \, \frac{s_i}{4} (s_i, \kappa)(\nabla_i - \Delta_i) \]  
(A.28)

\[ s_i = \frac{2 \nabla_i \Delta_i + \varepsilon}{\nabla_i^2 + \Delta_i^2 + \varepsilon} \]  
(A.29)

Thus, for both \( \phi - \kappa \) and Van Albada interpolation formulas, the cell interface \( Q \) value assumes the following implicit functional dependence

\[ Q_{i \pm 1/2} = f(Q_{i-1}, Q_i, Q_{i+1}) \]  
(A.30)

A variation of this function gives

\[ \delta Q_{i \pm 1/2} = \frac{\partial Q_{i \pm 1/2}}{\partial Q_{i-1}} \delta Q_{i-1} + \frac{\partial Q_{i \pm 1/2}}{\partial Q_i} \delta Q_i + \frac{\partial Q_{i \pm 1/2}}{\partial Q_{i+1}} \delta Q_{i+1} \]  
(A.31)

Likewise, a variation of Eq. (A.25) gives

\[ \delta Q_{i \pm 1/2} = \delta Q_i \pm \delta \Omega_i^+ \]  
(A.32)

It can be shown that both the \( \phi - \kappa \) and Van Albada interpolation formulas take the following form
\[ \delta Q_{i+1/2} = (\mp \alpha_{1,i}^\pm) \cdot \delta Q_{i-1} + (1 \pm \alpha_{1,i}^\pm \mp \alpha_{2,i}^\pm) \cdot \delta Q_i + (\pm \alpha_{2,i}^\pm) \cdot \delta Q_{i+1} \]  \hspace{1cm} (A.33)

where, for the \( \phi - \kappa \) interpolation,
\[ \alpha_{1,i}^\pm = \frac{1}{2}(1 \mp \kappa) \quad \alpha_{2,i}^\pm = \frac{1}{2}(1 \mp \kappa) \]  \hspace{1cm} (A.34)

and for the Van Albada limiter,
\[ \alpha_{1,i}^\pm = \beta_i^\pm \omega_{1,i} + \alpha_i^\pm \quad \alpha_{2,i}^\pm = \beta_i^\pm \omega_{2,i} + \alpha_i^\pm \]  \hspace{1cm} (A.35)

\[ \beta_i^\pm = \frac{1}{4} (\nabla_i \pm \Delta_i) + \frac{\kappa}{2} (\nabla_i - \Delta_i) \]  \hspace{1cm} (A.36a)
\[ \omega_{1,i} = 2(\Delta_i - s_i \nabla_i) / (\nabla_i^2 + \Delta_i^2 + \varepsilon) \]  \hspace{1cm} (A.36a)
\[ \omega_{2,i} = 2(\nabla_i - s_i \Delta_i) / (\nabla_i^2 + \Delta_i^2 + \varepsilon) \]  \hspace{1cm} (A.36a)
\[ \alpha_i^\pm = \frac{\kappa}{4} (1 \mp s_i \kappa) \]  \hspace{1cm} (A.36a)

Finally, comparing Eqs. (A.31) and (A.33), one finds
\[ \frac{\partial Q_{i \pm 1/2}^\pm}{\partial Q_{i-1}} = \mp \alpha_{1,i}^\pm \]  \hspace{1cm} \frac{\partial Q_{i \pm 1/2}^\pm}{\partial Q_i} = 1 \pm \alpha_{1,i}^\pm \mp \alpha_{2,i}^\pm \]  \hspace{1cm} \frac{\partial Q_{i \pm 1/2}^\pm}{\partial Q_{i+1}} = \pm \alpha_{2,i}^\pm \]  \hspace{1cm} (A.37)

These are the terms that are required in Eq. (A.11). To evaluate these expressions at other interfaces, simply shift \( i \) appropriately, e.g.,
\[ \frac{\partial Q_{i+1/2}^+}{\partial Q_i} = + \alpha_{1,i+1}^+ \quad ; \quad \frac{\partial Q_{i-1/2}^-}{\partial Q_i} = 1 + \alpha_{1,i-1}^- - \alpha_{2,i-1}^- \quad ; \quad \text{etc.} \]
APPENDIX B
DERIVATION OF THE DISCRETE SENSITIVITY EQUATION

From Section 3.3, it was noted that proper derivation of the aerodynamic sensitivity equation is based on

\[ R(Q, M) = 0 \]  \hspace{1cm} (B.1)

Also it was established that \( Q = Q(D) \) and \( M = M(D) \) where \( D \) is a vector of geometric-type design variables. Differentiating Eq. (B.1) with respect to the design variables \( D \) gives

\[ \frac{\partial R(Q, M)}{\partial D} = \left( \frac{\partial R(Q, M)}{\partial D} \right)_{M} + \frac{\partial R(Q, M)}{\partial D} \bigg|_{Q} = 0 \]  \hspace{1cm} (B.2)

or

\[ \frac{\partial R(Q, M)}{\partial D} \bigg|_{M} = -\frac{\partial R(Q, M)}{\partial D} \bigg|_{Q} \]  \hspace{1cm} (B.3)

where from Eqs. (A.5) and (A.6),

\[ R(Q, M) = \hat{F}_{i+1/2}(Q, M) - \hat{F}_{i-1/2}(Q, M) \]

\[ = \left[ \hat{F}_{i+1/2}(Q, M) + \hat{F}_{i-1/2}(Q, M) \right] - \left[ \hat{F}_{i+1/2}(Q, M) + \hat{F}_{i-1/2}(Q, M) \right] \]  \hspace{1cm} (B.4)

B.1 Left-Hand-Side of Sensitivity Equation

The left-hand-side of Eq. (B.3) may be simply given as

\[ \frac{\partial R(Q, M)}{\partial D} \bigg|_{M} = \left( \frac{\partial \hat{F}_{i+1/2}(Q, M)}{\partial D} \right)_{M} + \left( \frac{\partial \hat{F}_{i-1/2}(Q, M)}{\partial D} \right)_{M} \]  \hspace{1cm} (B.5)

Following the development in section A.1 [esp., Eqs. (A.8)-(A.10)], in general,
where \( DR^+ \) is defined by Eq. (A.11).

Applying the partial differentiation of Eq. (B.6) to Eq. (B.4) gives symbolically

\[
\text{LHS} = \frac{\partial R}{\partial Q} \frac{\partial Q}{\partial D}
\]  

(B.7)

which is identical to Eq. (A.13), except that the unknowns \( \Delta Q_{ixp} \) of Eq. (A.13) are replaced with the sensitivity unknowns \( \frac{\partial Q_{ixp}}{\partial D} \).

**B.2 Right-Hand-Side of Sensitivity Equation**

The right-hand-side of Eq. (B.3) may be simply given as

\[
\frac{\partial R(Q, M)}{\partial D} \bigg|_Q = \frac{\partial \hat{F}_{i+1/2}(Q, M)}{\partial D} \bigg|_Q + \frac{\partial \hat{F}_{i-1/2}(Q, M)}{\partial D} \bigg|_Q
\]  

(B.8)

Examination of the flux vectors of Eqs. (A.20) and (A.21) indicates that either flux vector can be written in the form

\[
\hat{F}(Q, M) = \frac{|\nabla \xi|}{J} \hat{F}^*
\]  

(B.9)

where \( \hat{F}^* \) is easily inferred from each respective equation.

Differentiation of Eq. (B.9) with respect to the geometric-type design variables \( D \) gives

\[
\frac{\partial \hat{F}(Q, M)}{\partial D} \bigg|_Q = \frac{\partial}{\partial D} \left( \frac{|\nabla \xi|}{J} \right) \cdot \hat{F}^* + \frac{|\nabla \xi|}{J} \frac{\partial \hat{F}^*}{\partial D} \bigg|_Q
\]  

(B.10)

For the full inviscid flux vector of Eq. (A.20),

\[
\frac{\partial \hat{F}^*}{\partial D} \bigg|_Q = \begin{bmatrix}
\rho \frac{\partial U}{\partial D} \\
\rho u_j \frac{\partial U}{\partial D} + p \frac{\partial E_j}{\partial D} \\
(p e_o + p) \frac{\partial U}{\partial D}
\end{bmatrix}
\]  

(B.11)
For the Van Leer split-flux formulation of Eq. (A.21),

\[ \frac{\partial \hat{F}^*}{\partial D} \bigg|_Q = \begin{bmatrix} \frac{\partial f^\pm_{\text{mass}}}{\partial D} \\ \frac{\partial f^\pm_{\text{mass}}}{\partial D} (f^\pm_{\text{energy}} \cdot \hat{g}^{j,\text{mom}}_{j,m}) \\ \frac{\partial f^\pm_{\text{mass}}}{\partial D} (f^\pm_{\text{energy}}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f^\pm_{\text{mass}}}{\partial D} \\ \frac{\partial f^\pm_{\text{mass}}}{\partial D} (f^\pm_{\text{energy}} + f^\pm_{\text{mass}} \frac{\partial g^j_{j,m}}{\partial D}) \end{bmatrix} \] (B.12)

where

\[ \frac{\partial \hat{U}}{\partial D} = u_j \frac{\partial \hat{e}_j}{\partial D} = u \frac{\partial \hat{e}_x}{\partial D} + v \frac{\partial \hat{e}_y}{\partial D} + w \frac{\partial \hat{e}_z}{\partial D} \] (B.13a)

\[ \frac{\partial f^\pm_{\text{mass}}}{\partial D} = \frac{\partial}{\partial D} \left( \hat{M}_x \pm 1 \right) \frac{\partial \hat{U}}{\partial D} \] (B.13b)

\[ \frac{\partial g^j_{j,m}}{\partial D} = \frac{a}{\gamma} \left( -\hat{M}_x \pm 1 \right) \frac{\partial \hat{e}_j}{\partial D} - \hat{e}_j \frac{\partial \hat{U}}{\partial D} \] (B.13c)

\[ \frac{\partial g^\pm_{\text{energy}}}{\partial D} = \frac{1}{\epsilon^2_{\text{a}}} \left[ 2(\gamma - 1) \rho^2 a \left( \hat{M}_x \pm 1 \right) \frac{\partial \hat{U}}{\partial D} \right] \] (B.13d)

Notice that the metric sensitivity terms $\frac{\partial \hat{e}_j}{\partial D}$ and $\frac{\partial}{\partial D} \left( \frac{\nabla k}{J} \right)$ are required. These terms are developed below.

**B.3 Sensitivity of the Transformation Metrics**

Physically, we desire to know the directed area of an arbitrary cell interface, that is,

\[ \frac{\nabla k}{J} = \frac{k_x}{J} \hat{i} + \frac{k_y}{J} \hat{j} + \frac{k_z}{J} \hat{k} \] (B.14)

where $k$ represents one of the coordinate directions $\xi$, $\eta$, or $\zeta$. The directed area is more useful for our purposes if it is decomposed into a cell interface area, $|\nabla k|/J$, and its direction cosines, $\hat{k}_i = \left[ \hat{k}_x, \hat{k}_y, \hat{k}_z \right]^T = \left[ \frac{k_x}{|\nabla k|}, \frac{k_y}{|\nabla k|}, \frac{k_z}{|\nabla k|} \right]^T$.

Consider in sketch (a) the parallelogram in three-space defined by two diagonal vectors, $\bar{a}$ and $\bar{b}$. This parallelogram may conveniently approximate any cell interface.
The diagonal vectors are defined as
\[ \vec{a} = a_1 \hat{i} + a_2 \hat{j} + a_3 \hat{k} \quad \text{and} \quad \vec{b} = b_1 \hat{i} + b_2 \hat{j} + b_3 \hat{k} \]  
where
\[ a_1 = x_3 - x_1 \quad a_2 = y_3 - y_1 \quad a_3 = z_3 - z_1 \]
\[ b_1 = x_4 - x_2 \quad b_2 = y_4 - y_2 \quad b_3 = z_4 - z_2 \]

Now, \( \vec{a} \times \vec{b} \) is twice the area of the parallelogram and is directed normal to the interface 1234.

\[ \vec{a} \times \vec{b} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \]
\[ = (a_1 b_2 - a_2 b_1) \hat{i} + (a_3 b_1 - a_1 b_3) \hat{j} + (a_2 b_3 - a_3 b_2) \hat{k} \]
\[ = P_1 \hat{i} + P_2 \hat{j} + P_3 \hat{k} = \vec{P} \]

Thus, the interface cell area may be computed by
\[ \frac{|\nabla k|}{J} = \frac{1}{2} |\vec{a} \times \vec{b}| = \frac{1}{2} |\vec{P}| = \frac{1}{2} \sqrt{P_1^2 + P_2^2 + P_3^2} \]

and the direction cosines may be given by
\[ \hat{k}_i = \frac{P_i}{|\vec{P}|} \quad i \in 1,2,3 \]
Differentiating the above "metric" terms with respect to the geometric-type design variables $D$ can be shown to give

\[
\frac{\partial}{\partial D} \left( \frac{\nabla k}{J} \right) = \frac{1}{2} \kappa_i \frac{\partial P_i}{\partial D}
\]

\[
= \frac{1}{2} \left( \kappa_x \frac{\partial P_1}{\partial D} + \kappa_y \frac{\partial P_2}{\partial D} + \kappa_z \frac{\partial P_3}{\partial D} \right)
\]  

(B.19)

and

\[
\frac{\partial k_i}{\partial D} = \frac{1}{2} \frac{J}{\nabla k} \left[ \frac{\partial P_i}{\partial D} - \kappa_i \left\{ 2 \frac{\partial}{\partial D} \left( \frac{\nabla k}{J} \right) \right\} \right]
\]  

(B.20)

where, for example,

\[
\frac{\partial P_1}{\partial D} = \left[ \frac{\partial a_1 \partial b_2 + a_1 \partial b_2}{\partial D} \right] - \left[ \frac{\partial a_2 \partial b_1 + a_2 \partial b_1}{\partial D} \right]
\]  

(B.21)

Similar expressions for $\partial P_2/\partial D$ and $\partial P_3/\partial D$ can be easily written. The terms of Eq. (B.21) may be further expanded as

\[
\frac{\partial a_1}{\partial D} = \frac{\partial x_3}{\partial D} - \frac{\partial x_1}{\partial D}
\]  

(B.22a)

\[
\frac{\partial b_2}{\partial D} = \frac{\partial y_4}{\partial D} - \frac{\partial y_2}{\partial D}
\]  

(B.22b)

Finally, observe that $\left\{ \frac{\partial x}{\partial D}, \frac{\partial y}{\partial D}, \frac{\partial z}{\partial D} \right\}$ are the grid sensitivity terms and are explained in detail in Chapter 5.
VITA

Gregory W. Burgreen was born in Athens, Alabama on February 19, 1963. He graduated from East Limestone High School in 1981. After being awarded a one year Outstanding High School Student Scholarship, he chose to attend The University of Alabama in Huntsville. During his undergraduate studies, he was employed by General Research Corporation in Huntsville, Alabama through the cooperative education program. Graduating with honors in August of 1986, he received a Bachelor of Science degree in Mechanical Engineering. He thereafter began graduate studies in Mechanical Engineering at the same academic institution, which culminated in the receipt of a Master of Science degree in December of 1987. From 1988 to 1990, Mr. Burgreen was employed by Teledyne Brown Engineering in Huntsville, Alabama where he gained significant experience in applied computational fluid dynamics. Since the the fall of 1990, he has pursued doctoral studies in Mechanical Engineering at Old Dominion University in Norfolk, Virginia with all of his research being performed at the NASA Langley Research Center in Hampton, Virginia.

Mr. Burgreen holds membership in the academic honor societies Phi Kappa Phi, Tau Beta Pi, and Pi Tau Sigma. He is the author and co-author of ten technical publications including those of Refs. [64, 66, 67, and 73].