Spring 1992

A Computational Aerodynamic Design Optimization Method Using Sensitivity Analysis

Mohamed E. Eleshaky
Old Dominion University

A COMPUTATIONAL AERODYNAMIC DESIGN OPTIMIZATION METHOD USING SENSITIVITY ANALYSIS

by

Mohamed E. Eleshaky
B.S., June 1982, Alexandria University, Alexandria, Egypt
M. Sc., June, 1987, Alexandria University, Alexandria, Egypt

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 1992

Approved by:

Oktay Baysal (Director)

Osama A. Kandil

Albert S. Roberts

David S. Miller

ABSTRACT

A COMPUTATIONAL AERODYNAMIC DESIGN OPTIMIZATION METHOD USING SENSITIVITY ANALYSIS

Mohamed E. Eleshaky
Old Dominion University
Director: Dr. Oktay Baysal

A new and efficient procedure for aerodynamic shape optimization is presented. The salient lineaments of this procedure are: (1) use of sensitivity analysis approach to determine analytically the aerodynamic sensitivity coefficients (gradients of the objective function and constraints); (2) the flowfield solution is obtained either by a computational fluid dynamics (CFD) analysis or, alternatively, by a flowfield extrapolation method (flow prediction method) which is based on a truncated Taylor's series; (3) the aerodynamic shape definition is not restricted to any class of surfaces and the optimizer automatically shapes the aerodynamic configuration to any arbitrary geometry; and (4) the procedure does not require any expertise other than that needed for formulating the optimization problem. This procedure is successfully demonstrated on two optimization problems.

In the first optimization problem, the ramp shape of a scramjet nozzle-afterbody configuration is optimized to yield a maximum thrust force coefficient. To gain a detailed understanding of the complex flowfield features of the nozzle-afterbody configurations prior to its design optimization, a CFD capability for the mixing of two-dimensional, viscous, multispecies flows has been developed. It is shown that heavier exhaust mixture (simulated by a Freon-Argon mixture) undergoes gasdynamic expansion at a smaller rate than does lighter "air" exhaust flow. The mixing of these flows is modeled in two different ways in order to compare their relative fidelities in modeling the flow physics and computational efficiencies. The computationally less expensive model achieves comparable quantitative results, but it produces some nonphysical oscillations.

In the second optimization problem, two transonic airfoils are shape optimized. The shape of the first airfoil is optimized to achieve a minimum drag while maintaining the lift above a specified value. The shape of the second airfoil is optimized to achieve a maximum lift while constraining the drag under a specified limit.

In the sensitivity analysis approach, two methods, namely, the direct sensitivity method and the adjoint variable method, are formulated and discussed in regards to their relative accuracy and computational requirements. These methods are also compared with the traditional finite-difference approach to assess their efficiencies. In this study, two flowfield governing equations are used, namely, the Euler equations and the thin-layer Navier-Stokes equations. Their discretized equations are solved using an implicit, upwind-biased, finite-volume scheme. The flux-vector splitting of Van Leer is used in the discretization of the pressure and convective terms.

Two direct matrix solvers and an iterative solver, which are deemed most applicable to the large linear systems of algebraic equations arising in the sensitivity approach, are investigated with regards to their accuracy, efficiency, and computer memory requirements. These solvers are shown to be feasible only for small two-dimensional problems, and due to the prohibitively high memory requirements, they become impractical for large two-dimensional problems and inapplicable for any of the three-dimensional problems. To alleviate this limitation, a new scheme based on domain-decomposition principles has been developed and is called the Sensitivity Analysis Domain-Decomposition (*SADD*) scheme.

TABLE OF CONTENTS

Part II: AERODYNAMIC DESIGN OPTIMIZATION

Chapter

APPENDICES

ACKNOWLEDGMENTS

I wish to express my sincere appreciation to my advisor Dr. Oktay Baysal for his guidance, ideas, and support throughout this study. I would like to thank my committee members, Dr. Osama A. Kandil, Dr. Albert S. Roberts, and David S. Miller for their valuable suggestions and editorial comments on this dissertation. In addition, I would like to extend my thanks to Greg Burgreen, Wendy Hoffman, and Walter C. Engelund, my research group, who have contributed to this research effort.

I wish to thank my wife Maissa for her patient support, encouragement and understanding during the course of this study. I would also like to thank my parents and family for teaching me to expect the best from myself.

This work has been supported by NASA Langley Research Center under Grant No. NAG-1-811 and Grant No. NAG-1-1188. The technical monitor was David S. Miller.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

A	surface area
	Jacobian matrix, $\partial\hat{F}/\partial Q$
A_s, B_s, C_s, D_s, E_s	coefficients of the specific heat function for species, s
$\bar{A}, \bar{B}, \bar{C}$	some elements of the coefficient matrix, $\partial R/\partial Q$
a	speed of sound
C_D	Drag coefficient
C_L	lift coefficient
Cp	specific heat at constant pressure
	pressure coefficient
C_{max}	the upper limit to the absolute value of relative slopes
CFD	computational fluid dynamics
CFL	Courant number
CPU	central processing unit
C_1, C_2, C_3	normalization constants for inequality constrains
c	chord length
D	diffusion coefficient, cm^2 sec^{-1}
$\bar{D}, \bar{E}, \bar{F}$	some elements of the coefficient matrix, $\partial R/\partial Q$
e	total energy per unit volume
F	objective function
$\vec{F}$	flux vector
F, G	inviscid flux vectors in Cartesian coordinates
$\hat{F}, \hat{G}$	inviscid flux vectors in generalized coordinates

$\hat{F}_v$, $\hat{G}_v$	viscous flux vectors in generalized coordinates
$\tilde{F}_{axial}$	axial component of the thrust force
$\tilde{F}_{inflow}$	normalizing force defined by Eq. (6.8)
f	mass fraction
$\bar{G}$, $\bar{H}$, $\bar{I}$	some elements of the coefficient matrix, ∂R/∂Q
G_j	the *j*th inequality constraints
G_v	viscous flux vector in Cartesian coordinates
H	combustor exit height
h	enthalpy
h^o	enthalpy of formation
I	number of cells in ξ-direction
$\vec{i}$, $\vec{j}$	unit vector in *x,y* directions, respectively
J	Jacobian of the coordinate transformation
	number of cells in η-direction
K	resultant force
k	coefficient of thermal conductivity
L	reference length
	difference operator
	lower factor of the coefficient matrix
Le	Lewis number
M	Mach number
	geometric terms, e.g., projected area
	matrix defined by Eq. (8.23)
N	number of species
NBL	number of subdomains
$NCON$	number of inequality constraints
$NCON_f$	number of flow-type inequality constraints

NDV	number of design variables
n	unit normal
	number of rows or columns of the coefficient matrix, $\partial R/\partial Q$
P	static pressure
Pr	Prandtl number
Q	vector of conserved variables in Cartesian coordinates
$\hat{Q}$	vector of conserved variables in generalized coordinates
q	primitive variables
$\dot{q}$	heat flux
R	universal gas constants
	steady-state residual
	Riemann invariants
Re	Reynolds number
r	number of right hand sides
S	Sutherland coefficient
	vector of search direction
	limiter control function
Sc	Schmidt number
s	arc length
T	temperature
T_o	Sutherland constant
t	time
	airfoil thickness-to-chord ratio
V	velocity vector
U	upper factor of the coefficient matrix
U,V	contravariant velocity components
u,v	velocity components in Cartesian coordinates

$\tilde{u}$, $\tilde{v}$	diffusion velocity components in Cartesian coordinates
$\vec{V}$	velocity vector
V	cell volume
V_C	control volume
w	molecular weight
$\overline{X}_D$	vector of design variables
X	species mole fraction
X_D	design variable
X_{Di}^{lower} , X_{Di}^{upper}	lower and upper limits of the *i*th design variable, respectively
x,y	Cartesian coordinates
Z	matrix defined by Eq. (8.24)

Greek Symbols

α	angle of attack
	nozzle ramp angle
α^*	move parameter in the one-dimensional search cycle
β	nozzle cowl angle
γ	ratio of specific heats
Δ	change
δ, ∇, Δ	difference operators
ε	very small number to prevent division by zero
ε_x, ε_y	artificial viscosity
θ	deflection angle
	the angle between the tangent to the aerodynamic surface and the x-direction
κ	parameter to determine the spatial accuracy of the upwind-biased formula

λ	second coefficient of viscosity
	vector of adjoint variables
μ	molecular viscosity coefficient
μ_o	Sutherland coefficient
μ_t	coefficient of turbulent viscosity
ξ,η	generalized curvilinear coordinates
ρ	density
Σ	summation
σ	effective collision diameter, Ångstrom
	safety factor
τ	viscous stress terms
	torsion spring factor in the adaptive gridding
Ω	collision integral (dimensionless)
ω	vorticity
	weighting factor in the adaptive gridding
$\lvert u \rvert , \lvert v \rvert$	magnitude of u and v velocities
∇	gradient
∂	partial derivative

Overlines

~	diffusion

Subscripts

b	backward
e	effective
f	forward
i, j	spatial indices
mix	mixture

min	minimum
max	maximum
r, s	index for species
t	turbulent
th	combustor exit
u, l	upper and lower
x, y	Cartesian coordinates
w	wall value
ξ, η	generalized curvilinear coordinates
∞	freestream value

Superscripts

b	boundary-interface cells of a subdomain
i	internal cells of a subdomain
n	time level
$\overline{n+1}$	intermediate time level
r	number of optimization iterations
T	transpose
$\circ$	reference value
	baseline value
$*$	extrapolated values
$+, -$	positive and negative fluxes
	conditions to the left and the right of cell interface

Chapter 1

OVERVIEW AND RATIONALE

In conventional aerodynamic optimization procedures, a dominant contributor to the computational cost is the calculation of the gradients information (aerodynamic sensitivity coefficients). Traditionally, the sensitivity coefficients are determined by a finite-difference approximation which requires repeating the aerodynamic analysis with incremented values of design variables. The cost of this approach is usually prohibitive particularly when the flowfield governing equations are expensive to solve. Moreover, this approach sometimes may provide highly inaccurate gradient information. A preferable approach for obtaining these coefficients can be achieved by using a sensitivity analysis (an analytical determination of the sensitivity coefficients). This approach not only eliminates the costly and repetitive flow analyses, but it also provides exact coefficients.

Another contributor to the computational cost in the above procedures is the repetitive flowfield analyses (CFD analyses) required during the evolution of the optimum shape. Usually, these analyses are performed every time the shape changes, even when the shapes are very close to each other. Therefore, it is desirable to develop a method that can extrapolate the flowfield solution when the changes in the shape are small and then predict a new solution at a relatively low computational cost.

In order to investigate the above concepts and thereby develop an efficient optimization procedure, demonstrative optimization problems have to be defined. One of the classical optimization problems is maximizing the thrust of an exhaust nozzle. This may be achieved by shaping the contour of the nozzle. A similar problem of today's current interest is the optimization of the shape of the nozzle-afterbody of a hypersonic vehicle powered by a scramjet. This problem allows examination of the procedure in high speed flow regimes since it involves an expansion of a supersonic flow which mixes with a hypersonic flow. Furthermore, the nozzle-afterbody configuration involves not only an internal flow, as do the conventional nozzles, but also an interaction between internal and external flows. Consequently, this design example facilitates the testing of the procedure concurrently in different types of flows.

To provide a better understanding of the flowfield features of the nozzle-afterbody configuration and in turn help its aerodynamic design at high speeds, a CFD capability for mixing flows has to be developed. For this purpose, the present study is divided into two parts. Part I focuses primarily on the issue of mixing flows through an internal-external nozzle of a scramjet engine. Whereas, Part II focuses on the details of the different components of the new aerodynamic optimization procedure.

PART I: NOZZLE-AFTERBODY FLOWFIELD ANALYSIS

Chapter 2

INTRODUCTION TO NOZZLE-AFTERBODY FLOW ANALYSIS

In recent years, the ability to predict the performance characteristics of aircraft nozzle-afterbody configurations at hypersonic speeds quickly and accurately has become increasingly important. One of the principle reasons for this is to provide a better understanding of the flowfield features of these configurations, and hence help their aerodynamic design at very high speeds. Another important reason is that, above Mach number of approximately 8, ground test facilities do not duplicate the relevant flight simulation parameters such as Mach number, Reynolds numbers, gas composition, and enthalpy level. Therefore, other alternatives should be found for the testing of such aircraft configurations.

At hypersonic speeds, the propulsion system efficiency must be high, the engine must add minimum drag and weight to the vehicle, and the engine must process as much air as possible. These objectives are best met by considering the entire underside of the vehicle as part of the propulsion system. This concept, referred to as the airframe-integrated modular scramjet, is illustrated in Fig. 2.1. The concept utilizes the aircraft forebody for part of the inlet compression and the aftbody as part of the nozzle expansion. This leads to reducing the size as well as the weight of the nozzle while obtaining the required thrust at the high Mach numbers. However, the impingement of the highly underexpanded exhaust gases on the afterbody and the control surfaces of the aircraft could significantly affect the overall vehicle stability. Therefore, obtaining the optimum thrust while limiting unfavorable forces and moments on the vehicle is a crucial issue in the design of the nozzle and afterbody region.

The aerodynamic analysis and testing of such nozzle-afterbody configurations is an important facet of the propulsion-airframe integration effort. As mentioned before, ground-based testing of actual scramjet-afterbody geometries is not only difficult, but also expensive due to the high temperatures and pressures incurred in the combustion process. One of the alternatives for this is the use of a simulant gas to substitute for the actual combustion products, provided that the dynamic and thermodynamic similitude are enforced. Perhaps a more economical alternative would be to do the preliminary design analysis using Computational Fluid Dynamics (CFD) codes, due, in part, to the extremely rapid growth in the speed and storage capability of the digital computers. Therefore, an overwhelming degree of reliance has been placed on CFD, especially in the achievement of hypersonic flight with a NASP-like vehicle (National Aero-Space Plane). This reliance covers the range of design activities, from the design of aircraft configuration to the design of the integrated engine system. In fact, the belief that CFD can be used to predict all of the relevant flow physics, from aircraft take off to orbital speeds and return, has been one of the reasons for the revival of hypersonic research. However, there is currently very little experimental data for very high Mach number flows. Hence, some means of calibrating and validating CFD codes must be achieved before they can be used with complete confidence in the design process.

2.1 Literature Survey

In the mid-1970's, an experimental technique for the cold gas simulation of a scramjet exhaust [1-3] was developed as part of the X-24C program. The objectives of this study were to establish the standards of similitude for a hydrogen-air scramjet exhaust interacting with a hypersonic vehicle afterbody, and to experimentally validate the procedures that would be usable in conventional wind tunnels. It was determined in this study, that in addition to the usual nondimensional similitude parameter requirements for inviscid flows (i.e. Mach numbers, pressure ratios, temperature ratios, etc.) the ratio of specific heats (γ) of the combustion products must also be matched by the simulant gases. This concept is discussed in further detail in Ref. [4]. It was also determined in Refs. [1–3] that the nonequilibrium dissociation of gases at high exhaust temperature is deemed secondary in influencing the aerodynamics. However, the aerodynamic forces were very sensitive to flow perturbations caused by the nozzle geometry.

An extension of this work was carried out at NASA Langley Research Center by Cubbage and Monta [5] and Monta [6]. In these studies, a wind tunnel model of a single-module scramjet nozzle-afterbody configuration was constructed for testing (Fig. 2.2). The simulant gas mixture was fed into a high pressure plenum chamber via a mounting strut. The gas in this plenum chamber was expanded through a converging-diverging supersonic nozzle to approximately Mach 1.7 at the combustor exit plane, where it was further expanded over the nozzle-afterbody section of the model. This supersonic exhaust flow also encountered a hypersonic (Mach 6) freestream air flow, through which mixing occured in free shear layer containing additional expansions and shock waves. A removable tapered flow fence was used to simulate a quasi two-dimensional flow. When this fence was removed, the nozzle flow also mixed with the hypersonic freestream in the lateral direction through a spanwise expansion, causing the flow to become fully three-dimensional. Experimental data was obtained for a scaled scramjet nozzle-afterbody flowfield using both air and a Freon/Argon mixture as the simulant gas. Static pressures were measured on the afterbody surface, for both two-dimensional and three-dimensional flows, with various nozzle-afterbody geometries. Also, by using a flow rake specifically designed for this purpose, the off-surface flow was surveyed to obtain the pitot pressures. The data obtained from these experiments were used to compare with the present computational results.

Extensive numerical studies were conducted on rocket and nozzle based flows without solid afterbody surface interaction, that is, the flow from a nozzle exit plane which flows directly into the freestream. Most of these studies focused mainly on the thrust vectors generated by the nozzle itself, and not on the actual afterbody flow field plume structures. However, there were several of these studies which did attempt to analyze the downstream flow. Deiwert [7] used the thin-shear-layer formulation of the Navier-Stokes equations in the study of supersonic axisymmetric flow over boattails containing a centered propulsive jet. Solutions were presented for jet flows expanding supersonically into a low pressure supersonic freestream flow, with an in-depth analysis of the afterbody flow structure. Comparison with experimental data showed good agreement in many of the key flow features, such as, exhaust plume shape and structure, and the location of the external compression shocks. However, the quantitative comparison of nozzle flow exit angle was poor, and thought to be due to the improper modeling of turbulent transport phenomena in the region of separated flow at the base of the nozzle.

Hoffman et al. [8] computed the afterbody flowfield of an axisymmetric cold gas rocket nozzle flowfield using a complete Navier-Stokes formulation algorithm. They presented the results of a grid refinement study which showed that the grid resolution is very important in regions of shear layers and recirculation. It was also shown that the results obtained on a fine grid, where the grid point spacing was very dense in regions of expected large gradients, compared much more favorably with the experimental results.

In recent years, with the renewed interest in hypersonics and propulsion airframe integration, there have been numerous computational studies related to nozzle-afterbody flows. Most of the literature concentrated on one particular aspect of the flow or its approximation such as, turbulence modeling, adaptive gridding schemes, shear layer analysis, and real gas effects. Several of the studies which are directly related to the present work are given below.

Barber and Cox [9] presented an overview of computational works being conducted towards the design of hypersonic airbreathing aircraft, with a section specifically devoted to the integration of the propulsion system. Povinelli [10] provided a summary of the current computational works which were directly related to the propulsion systems, including the nozzle-afterbody section. Harloff et al. [11] conducted two-dimensional Navier-Stokes analysis of scramjet nozzle flowfields at design and off-design conditions. In this study, both the nozzle exhaust flow and the external flow were assumed to be air. Nozzle afterbody flowfields were computed over a range of nozzle exit Mach numbers for over-expanded and under-expanded flows. Nozzle efficiencies were computed for all of the cases, however, the numerical results were not compared with experimental data.

In an effort to provide a better definition of the nozzle flow field features, use of adaptive gridding was studied by Hsu [12]. Several cases of air flows were calculated using a two-dimensional Navier-Stokes code with and without the use of adaptive grids. It was found that the use of adaptive grid yielded a sharper shock and thinner boundary layer, as well as diminishing the large region of shock-induced-boundary-layer-separation.

Baysal and Hoffman [13, 14] conducted three-dimensional calculations of non-axisymmetric air flow, in which the nozzle jet was allowed to expand and shear in all three directions. This was accomplished by adding an external side wall and side ramp to the internal nozzle. Although the general trend of the off-surface experimental data was followed by the computational results, some discrepancies were observed in both shear layer and shock regions. The computational results, however, agreed very well with the experimental surface pressure data.

Lai and Nelson [15] used the three-dimensional Navier-Stokes code, PARC, to compute nonaxisymmetric nozzle flows. The numerical scheme of this code employed three-point central differences uniformly throughout the flowfield to approximate spatial derivatives. Second- and fourth-order Jameson type artificial dissipations were also included. Laminar flow computations were carried out for the experimental nozzle geometry of Re and Leavitt [16], which exhausted into still air. This study indicated that the shear layer is highly three-dimensional in its structure with significant variations from the top wall to the bottom and the side walls.

In an effort to illustrate the CFD capability in simulating the mixing of multispecies flows in a nozzle-afterbody, Baysal et al. [17] performed two-dimensional analyses and compared with the experimental results of Cubbage et al. [5, 18] and Pittman [19]. The flow was analyzed using two different codes. The first code used was an implicit, upwind, finite-volume, Navier-Stokes solver which assumed a constant ratio of specific heats (γ). The second code was an explicit MacCormack-scheme based Navier-Stokes solver. This multispecies code included species continuity equations to account for variable γ due to the mixing of the simulant gas (Freon/Argon mixture) with the external freestream air. The results of further two-dimensional calculations, in which two different mixing models were compared, was also reported by Baysal et al. [20]. This work was extended by Baysal et al. [21, 22] to three-dimensional flows and adaptive gridding. The two-dimensional computational meshes were refined using flow adaptive grids to enhance the computational solution in the regions of high gradients, and to reduce the overall computational error. Three-dimensional flow solutions were computed for the nozzle-afterbody test section and compared with the three-dimensional experimental surface pressure data [5]. It was found that the Argon/Freon simulant gas expanded at a slower rate than air, which led to a more pronounced reverse flow region at the cowl tip.

Ray et al. [23] presented two- and three-dimensional results based on Euler (inviscid) calculations of single and multiple module scramjet afterbody flows using a Freon/Argon simulant gas mixture. Their results showed good agreement with the experimental data [3]; however, their calculations were based on the nozzle flow expansion into quiescent air not the actual hypersonic flow of air. Also, since they did not include the viscous effects, shear layers and separated flow regions were not captured.

Edwards [24] studied the exhaust plume/afterbody interactions using the thin-layer Navier-Stokes equations with a coupled species continuity equation. This allowed for the solution of a binary gas mixture flow. Computations were performed for the external flow of air (γ =1.4) mixing with a simulant gas (γ =1.26). Additional computations were done with the simulant gas assumed to be air (γ =1.4). It was concluded from the computational results and a kinetic theory of gases rationalization, that the drop in static pressure in supersonic expansion will be smaller for the γ =1.26 gas than for the γ =1.4 gas, and hence produced greater afterbody forces.

Ruffin et al. [25] have used two- and three-dimensional upwind Navier-Stokes solvers to perform preliminary analyses for a planned nozzle-afterbody experiment. Laminar nonreacting computations of a nozzle exhaust flowfield were performed on a grid of (48x60x3) for the two-dimensional flows and on two patched grids for the three-dimensional flows. Two-dimensional calculations were conducted with parametric studies over various Mach numbers, pressure ratios, and afterbody ramp angles in order to help determine the experimental model loads and optimum afterbody ramp angle and length. Three-dimensional calculations were performed to predict the shape of the jet plume and the flow spillage from the windward side of the model into the expanding flow region. Three-dimensional results were also used to determine optimal locations for experimental probes and flow measurements devices, and to aid in the design of side flow fences which were used to minimize the flow spillage.

Tatum et al. [26] conducted two- and three-dimensional analyses of scramjet external nozzle flowfields and compared surface and off-surface flowfield data of Cubbage and Monta [5] and of Monta [6]. Air and an Argon/Freon mixture were used as simulant gases. The off-surface flowfield was predicted best when calculations were for a turbulent flow. However, the laminar solution predicted more accurate comparisons of forces and moments.

Huebner et al. [27] performed two-dimensional CFD analyses on afterbody configurations to study a number of issues pertinent to hypersonic, airbreathing vehicles. A comparison of computational results with the schlieren photographs of a powered, hypersonic, airbreathing cruise missile configuration showed that accurate prediction of afterbody flow features necessitated the solution of Navier-Stokes equations. It also showed that, at the least, thin-layer Navier-Stokes solutions were required for any conditions at which cowl tip boundary layer separation was likely to occur.

2.2 OBJECTIVES

This part of the dissertation is directed towards gaining a detailed understanding of the complex flowfield features of the nozzle-afterbody configurations. It also aims at developing a CFD capability for the mixing of multispecies, supersonic and hypersonic flows. This, in turn, aids the aerodynamic design optimization of the afterbody of hypersonic vehicles. Also, this computational capability augments the experimental studies of cold-gas simulation. These objectives are accomplished by:

(1) performing adaptive computations of viscous, multispecies mixing of a supersonic jet and a hypersonic freestream of air;

(2) addressing two important issues, which arise from the efforts of choosing the appropriate multispecies mixing model. Firstly, since the flow is turbulent and has high-speed, some simplifications are possible in modeling the diffusive multispecies mass transport. The second issue is related to the numerical modeling error that

manifests itself in the mass deficit. The common practice of assuming that the sum of species mass ratios is unity, may hide some of this error;

(3) investigating the differences between using air as opposed to a simulant gas (Freon/Argon mixture) in modeling the exhaust flows.

The organization of this part of the dissertation is as follows. Chapter 3 conveys the basic formulation for cold gas nozzle-afterbody flow analysis, i.e., the governing equations for flows with variable specific heats. Also, the baseline solution algorithm as well as different diffusion velocity models are discussed in Chapter 3. Some representative results for two types of flows, namely, Air-Air flows and multispecies flows are presented in Chapter 4.

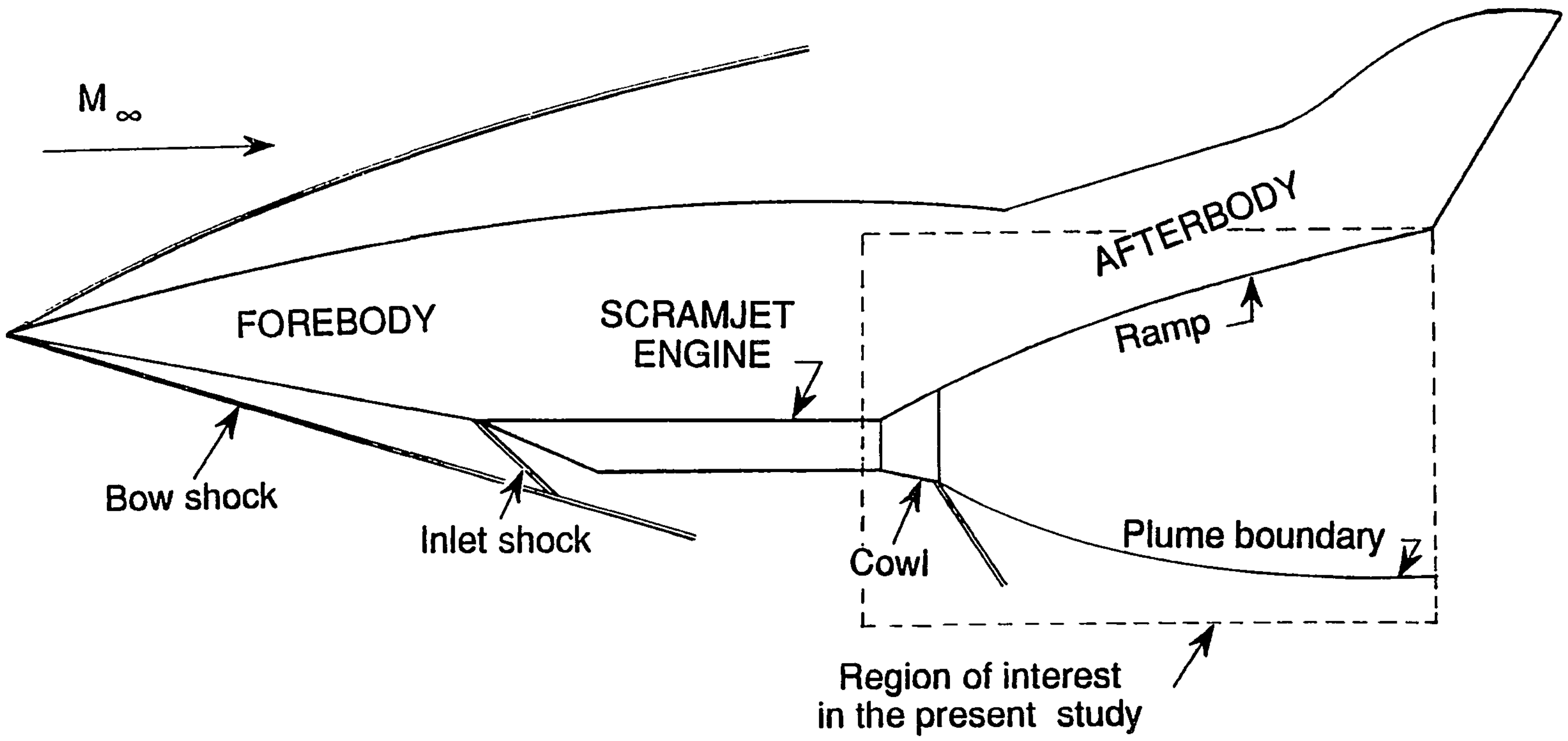


Fig. 2.1 A schematic of the flowfield around a generic hypersonic vehicle.

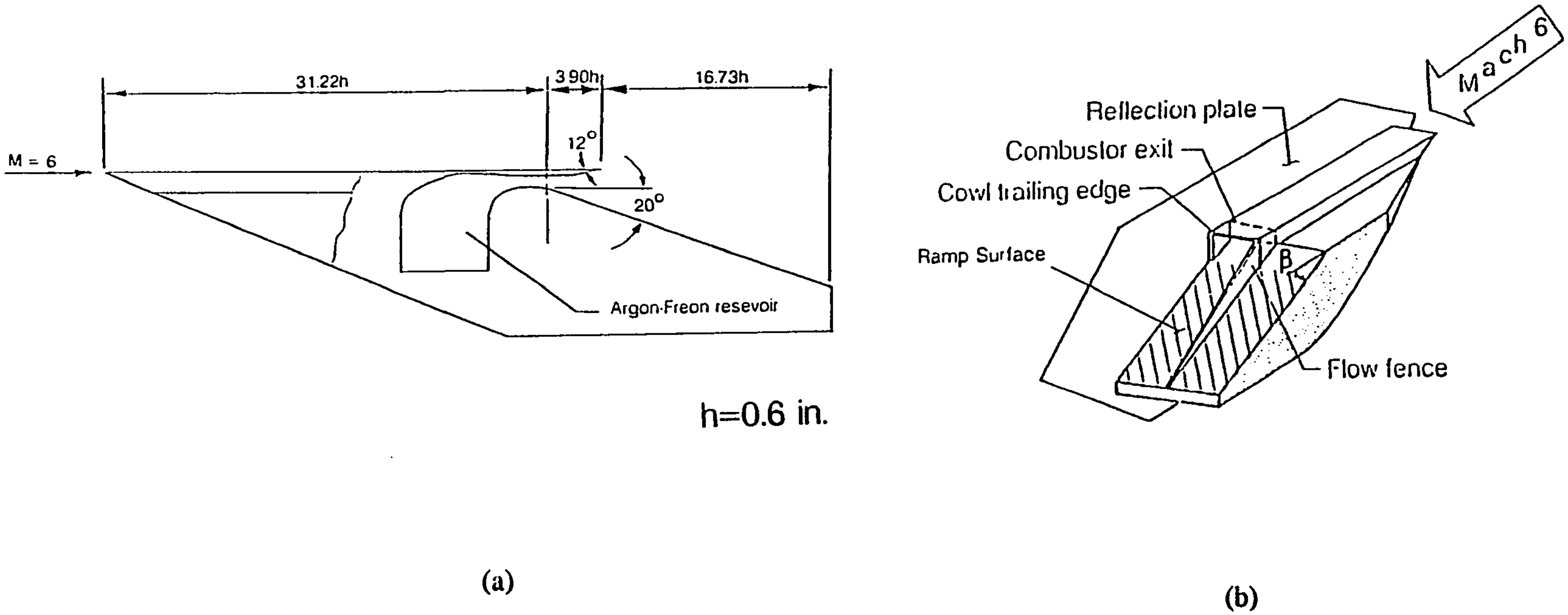


Fig. 2.2 A wind tunnel model of a single-module scramjet nozzle and afterbody: (a) sectional view (b) isometric view.

Chapter 3

FORMULATION FOR MULTISPECIES FLOW ANALYSIS

3.1 Governing Equations

The nozzle-afterbody flowfield is governed by the Navier-Stokes equations coupled with a set of continuity equations describing each of the species present in the flowfield. Since the simulant exhaust gas mixture is cold, the flow is non-reacting and thence the chemical production terms disappear from the species continuity equations as well as the Navier-Stokes equations. The conservation form of the time-dependent, compressible, two-dimensional, Reynolds-averaged Navier-Stokes equations, and species continuity equations written in generalized coordinates [17] is given below:

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial\left(\hat{F} - \hat{F}_v\right)}{\partial \xi} + \frac{\partial\left(\hat{G} - \hat{G}_v\right)}{\partial \eta} = 0 \tag{3.1}$$

where

$$\hat{Q} = \frac{Q}{J} = \frac{1}{J}\left\{\begin{matrix} \rho \\ \rho u \\ \rho v \\ \rho e \\ \rho f_s \end{matrix}\right\} \qquad \hat{F} = \frac{1}{J}\left\{\begin{matrix} \rho U \\ \rho U u + \xi_x P \\ \rho U v + \xi_y P \\ (e + P)\, U \\ \rho U f_s \end{matrix}\right\} \qquad \hat{G} = \frac{1}{J}\left\{\begin{matrix} \rho V \\ \rho V u + \eta_x P \\ \rho V v + \eta_y P \\ (e + P)\, V \\ \rho V f_s \end{matrix}\right\}$$

$$\hat{F}_v = \frac{1}{J}\left\{\begin{matrix} o \\ \xi_x \tau_{xx} + \xi_y \tau_{xy} \\ \xi_x \tau_{xy} + \xi_y \tau_{yy} \\ \xi_x b_x + \xi_y b_y \\ \rho\left(\xi_x \tilde{u}_s + \xi_y \tilde{v}_s\right) f_s \end{matrix}\right\} \qquad \hat{G}_v = \frac{1}{J}\left\{\begin{matrix} o \\ \eta_x \tau_{xx} + \eta_y \tau_{xy} \\ \eta_x \tau_{xy} + \eta_y \tau_{yy} \\ \eta_x b_x + \eta_y b_y \\ \rho\left(\eta_x \tilde{u}_s + \eta_y \tilde{v}_s\right) f_s \end{matrix}\right\}$$

The coordinate transformation Jacobian J is given by:

$$J = \frac{1}{x_\xi y_\eta - y_\xi x_\eta} \tag{3.2}$$

The contravariant velocities, U and V, are defined as:

$$U = \xi_x u + \xi_y v \tag{3.3a}$$

$$V = \eta_x u + \eta_y v \tag{3.3b}$$

The shear stress, τ, and the heat flux, $\dot{q}$, terms are written as follows:

$$\tau_{xx} = (\lambda + \mu) \left(u_\xi \xi_x + u_\eta \eta_x\right) + \lambda \left(v_\xi \xi_y + v_\eta \eta_y\right) \tag{3.4a}$$

$$\tau_{yy} = (\lambda + \mu) \left(v_\xi \xi_y + v_\eta \eta_y\right) + \lambda \left(u_\xi \xi_x + u_\eta \eta_x\right) \tag{3.4b}$$

$$\tau_{xy} = \mu \left(u_\xi \xi_y + u_\eta \eta_y + v_\xi \xi_x + v_\eta \eta_x\right) \tag{3.4c}$$

$$\dot{q}_x = -k\left(\xi_x \frac{\partial T}{\partial \xi} + \eta_x \frac{\partial T}{\partial \eta}\right) + \rho \sum_{s=1}^{N} h_s f_s \tilde{u}_s \tag{3.5a}$$

$$\dot{q}_y = -k\left(\xi_y \frac{\partial T}{\partial \xi} + \eta_y \frac{\partial T}{\partial \eta}\right) + \rho \sum_{s=1}^{N} h_s f_s \tilde{v}_s \tag{3.5b}$$

The terms b_x and b_y in both F_v and G_v are given by:

$$b_x = u\,\tau_{xx} + v\,\tau_{xy} - \dot{q}_x \tag{3.6a}$$

$$b_y = u\,\tau_{xy} + v\,\tau_{yy} - \dot{q}_y \tag{3.6b}$$

In the above system, the mass fraction of each species (s) is defined as $f_s = \rho_s / \rho$. The enthalpy of each species (s) is denoted by h_s. The pressure and temperature of the mixture are denoted by P and T, respectively. Stokes hypothesis for bulk viscosity, $\lambda+(2/3)\mu = 0$, is invoked. The Cartesian components of the species diffusion velocity $\tilde{\vec{V}}_s$ are $\tilde{u}_s$ and $\tilde{v}_s$, which will be defined in Section 3.4.

For the mixing of N species, the subscript (s) varies from 1 to (N–1). Hence, there are (N–1) species continuity equations to be solved for f_s in addition to the global continuity equation, two momentum equations, and the energy equation. By virtue of the conservation of mass, the mass fraction of the Nth species, f_N, can then be found from the following identity [28],

$$\sum_{s=1}^{N} f_s = 1.0 \tag{3.7}$$

Therefore, (N+3) coupled partial differential equations, Eq. (3.1), need to be solved for the vector of conserved quantities Q. However, in an attempt to compute the global mass conservation error, the computations are repeated by solving N species continuity equations, that is, a total of (N+4) coupled equations.

3.1.1 Thermodynamic Model

In this study, all the gases are assumed to be thermally perfect but calorically real gases. Hence, the enthalpy (h) of each species (s), the total internal energy, and the pressure can be expressed as:

$$h_s = h_s^{\circ} + \int_{298.25}^{T} Cp_s \, dT \tag{3.8}$$

$$e = \sum_{s=1}^{N} h_s f_s - \frac{P}{\rho} + \frac{1}{2}\left(u^2 + v^2\right) \tag{3.9}$$

$$P = \rho R_{mix} T \tag{3.10}$$

The enthalpy of formation and mixture gas constant are denoted by h^o and R_{mix}, respectively. Note that the enthalpy of the mixture is the summation of the partial enthalpies of the species. Equation (3.10) is the ideal gas equation written for a mixture with N species.

Other thermodynamic quantities, such as, the specific heat at constant pressure for each species, Cp_s, and the ratio of specific heats, γ, are calculated as follows. The specific heat for each species is first defined by a fourth-order polynomial in temperature [28],

$$\frac{Cp_s}{R_s} = A_s + B_s T + C_s T^2 + D_s T^3 + E_s T^4 \tag{3.11}$$

where R_s is the species gas constant. The coefficients in Eq. (3.11) are given in Table A.1 for each species involved in this study. Knowing the specific heat of each species, the enthalpy of each species can then be found from Eq. (3.8) with the use of Table A.2. The total internal energy is then computed from Eq. (3.9). The ratio of specific heats is obtained from the following relationship:

$$\gamma = \frac{R_{mix}}{\left(Cp_{mix} - R_{mix}\right)} + 1 \tag{3.12}$$

where Cp_{mix} is the mixture specific heat at constant pressure and is given by:

$$Cp_{mix} = \sum_{s=1}^{N} Cp_s f_s \tag{3.13}$$

The expression for the mixture gas constant, used in both Eq. (3.10) and Eq. (3.12), is given by:

$$R_{mix} = \sum_{s=1}^{N} f_s R_s \tag{3.14}$$

where,

$$R_s = \frac{R}{w_s} \tag{3.15}$$

R is the universal gas constant given in Appendix A. The molecular weight, w, for different species are given in Table A.2.

3.1.2 Transport Properties

The determination of the mixture transport properties, namely, the coefficients of mixture viscosity, mixture thermal conductivity, and diffusion, is discussed in this section.

3.1.2.1 Coefficient of viscosity for gas mixture:

For a mixture of gases of markedly different species, mixture viscosity varies strongly with species mass fraction. The semi-empirical formula of Wilke [29] given below is deemed quite adequate in determining the mixutre viscosity:

$$\mu_{mix} = \sum_{s=1}^{N} \frac{X_s\,\mu_s}{\sum_{r=1}^{N} X_r\,\Phi_{sr}} \tag{3.16}$$

in which

$$\Phi_{sr} = \frac{\left[1 + (\mu_s/\mu_r)^{1/2}(w_r/w_s)^{1/4}\right]^2}{\left[8 + 8w_s/w_r\right]^{1/2}} \tag{3.17}$$

Here, X_s and X_r are the mole fractions of species s and r; μ_s and μ_r are the molecular viscosities of species s and r at the system temperature and pressure; and w_s and w_r are the corresponding molecular weights. Note that Φ_{sr} is dimensionless and, r = s when Φ_{sr} = 1.

In terms of the mass fraction, f_s, the mole fraction of species (s) [30] is given by:

$$X_s = \frac{f_s/w_s}{\sum_{r=1}^{N} f_r/w_r} \tag{3.18}$$

The viscosity of an individual species is computed from the Sutherland law [30],

$$\frac{\mu_s}{\mu_o} = \left(\frac{T}{T_o}\right)^{3/2} \frac{T_o + S}{T + S} \tag{3.19}$$

where μ_s and T_o are reference values and S is the Sutherland constant. These values for the species involved in this study are given in Table A.3.

For turbulent flow computations, the turbulent Reynolds stresses in the time smoothed Navier-Stokes equations are assumed to be proportional to the laminar stress tensor with a coefficient of proportionality defined as the eddy viscosity, μ_t. Therefore, for turbulent calculations, the effective mixture viscosity coefficient, $\mu_{e,mix}$, defined below is used,

$$\mu_{e,mix} = \mu_{mix} + \mu_t \tag{3.20}$$

A detailed description of the eddy viscosity coefficient, μ_t, is given in Refs. [4,34,35].

3.1.2.2 Coefficient of thermal conductivity for gas mixture:

Coefficient of thermal conductivity for the gas mixture is estimated by a method analogous to that previously given for viscosity [31],

$$k_{mix} = \sum_{s=1}^{N} \frac{X_s\, k_s}{\sum_{r=1}^{N} X_r\, \Phi_{sr}} \tag{3.21}$$

Here, k_s is the thermal conductivity of species (s) and is computed from the Sutherland law (Eq. (3.19)),

$$\frac{k_s}{k_o} = \left(\frac{T}{T_o'}\right)^{3/2} \frac{T_o' + S'}{T + S'} \tag{3.22}$$

The reference values k_o and T'_o, and the Sutherland constants, S', are also tabulated for the species involved in this study in Table A.4.

The dimensionless quantity Φ_{sr} is identical to the one which appeared in the viscosity equation (Eq. (3.17)).

For turbulent flow computations, the effective thermal conductivity coefficient of the mixture is given by

$$k_{e,mix} = k_{mix} + k_{t,mix} \tag{3.23}$$

where $k_{t,mix}$ is the turbulent thermal conductivity of the mixture. It is obtained by specifying turbulent Prandtl number and using the following relation

$$k_{t,mix} = \frac{Cp_{mix}\,\mu_t}{Pr_t} \tag{3.24}$$

In this study, the turbulent Prandtl number is given the value of 0.9.

3.1.2.3 Binary-Diffusion Coefficient:

The Chapman-Enskog kinetic theory formula for the viscosity and the thermal conductivity were given earlier by Eq. (3.16) and Eq. (3.21); the corresponding formula [30] for the binary-diffusion coefficient D_{sr} between species (s) and (r) is:

$$D_{sr} = \frac{0.0018583\sqrt{T^3\left(\frac{1}{w_s}+\frac{1}{w_r}\right)}}{p\sigma_{sr}^2\,\Omega_{D,sr}} \tag{3.25}$$

in which $\Omega_{D,sr}$ is a dimensionless function of temperature and of the intermolecular potential field for one molecule of (r) and one molecule of (s). This dimensionless quantity is called the diffusion collision integral and is approximated by:

$$\Omega_{D,sr} = T^{*\,-0.145} + \left(T^* + 0.5\right)^{-2.0} \tag{3.26}$$

where $T^* = T/T_{\varepsilon,sr}$. The effective temperature, $T_{\varepsilon,sr}$, and the effective collision diameter, σ_{rs}, are averages computed from the separate molecular parameters of each species:

$$\sigma_{sr} = \frac{1}{2}(\sigma_s + \sigma_r) \tag{3.27}$$

$$T_{\varepsilon,sr} = (T_{\varepsilon,s}\, T_{\varepsilon,r})^{1/2} \tag{3.28}$$

These molecular parameters are listed in Table A.5 for the species involved in this study.

For some engineering calculations [32], the mass diffusivity of all species can be assumed identical. Hence, the binary-diffusion coefficients, D_{sr}, can be assumed the same for all species and be approximated by:

$$D_{sr} = \frac{\mu_{mix}}{\rho Sc} \tag{3.29}$$

where the Schmidt number, Sc, is given the value of 0.22.

For turbulent flow computations, the effective binary diffusion coefficient is given by

$$D_{e,sr} = D_{sr} + D_{t,sr} \tag{3.30}$$

where $D_{t,sr}$ is the turbulent binary diffusion, and is obtained by specifying turbulent Schmidt number along with the use of the following relation

$$D_{t,sr} = \frac{\mu_t}{\rho Sc_t} \tag{3.31}$$

Since for most turbulent mixing problems the turbulent Lewis number, which is the ratio of the turbulent Prandtl and Schmidt numbers, is approximately unity, the turbulent Schmidt number is equal to 0.9.

3.2 Determination of Diffusion Velocities

The diffusion velocity $\tilde{\vec{V}}_s$ is the velocity of each species introduced upon the convection velocity due to all diffusion processes. The diffusion velocity $\tilde{\vec{V}}_s$ is calculated using two different methods; the simple binary interaction model and the complete multicomponent diffusive interaction model.

In the binary model, mass diffusivities of all the species are assumed identical, and only concentration gradient effects are included. Hence, the diffusion velocity is computed using the Fick's law [31] written as:

$$\tilde{\vec{V}}_s = - D_{e,sr} \nabla f_s \tag{3.32}$$

where the diffusion coefficients, $D_{e,sr}$, are computed using Eqs. (3.29) and (3.30).

In the complete multicomponent model, it is assumed that there is no thermal diffusion and that the same body force per unit mass is acting upon each species. Hence, the species diffusion velocity is determined using the reduced form of the multicomponent diffusion equation [33] derived from the kinetic theory and written as,

$$\nabla X_s = \sum_{r=1}^{N} \left[\left(\frac{X_s X_r}{D_{e,sr}} \right) \left(\tilde{\vec{V}}_r - \tilde{\vec{V}}_s \right) \right] + \left(f_s - X_s \right) \frac{\nabla p}{p} \tag{3.33}$$

Here the diffusion coefficients, $D_{e,sr}$, for each species is computed using Eqs. (3.25) and (3.30).

Equation (3.33) requires solving (N) simultaneous algebraic equtions for each component of the velocity. It should be noted that for N species, however, the system of N equations defined by Eq. (3.33) is not linearly independent. Therefore, one of the equations must be replaced by the following constraint

$$\sum_{s=1}^{N} \rho f_s \tilde{\vec{V}}_s = 0 \tag{3.34}$$

The resulting system of algebraic equations is solved using a lower-upper (LU) decomposition method.

When solving (*N*) species continuity equations, the complete multicomponent model cannot be applied, because Eqs. (3.17) and (3.34) can no longer be satisfied in an exact manner due to the computational error.

The local mass error, *LME*, distribution due to the modeling of multispecies mixing is computed from the formula below, which is evaluated at every grid point

$$LME = \rho - \sum_{s=1}^{N} \rho_s \tag{3.35}$$

The global mass conservation error is also computed by numerically integrating the mass along the computational domain boundaries.

3.3 Solution Algorithm

When the spatially unsplit, predictor-corrector technique of MacCormack [36] is applied to the governing equations (Eq. (3.1)), the following algorithm results:

Predictor:

$$\hat{Q}_{i,j}^{\overline{n+1}} = \hat{Q}_{i,j}^{n} - \Delta t \left[\Delta_\xi\left(\hat{F} - \hat{F}_v\right)_{i,j}^{n} + \Delta_\eta\left(\hat{G} - \hat{G}_v\right)_{i,j}^{n}\right] \tag{3.36}$$

Corrector:

$$\hat{Q}_{i,j}^{n+1} = \frac{1}{2}\left\{\hat{Q}_{i,j}^{n} + \hat{Q}_{i,j}^{\overline{n+1}} - \Delta t \left[\nabla_\xi\left(\hat{F} - \hat{F}_v\right)_{i,j}^{\overline{n+1}} + \nabla_\eta\left(\hat{G} - \hat{G}_v\right)_{i,j}^{\overline{n+1}}\right]\right\} \tag{3.37}$$

where Δ_ξ and Δ_η are forward spatial differences, and ∇_ξ and ∇_η are backward spatial differences. The coordinate transformation is chosen such that $\Delta\xi = \Delta\eta = 1.0$.

This explicit finite difference scheme is second-order accurate in both time and space. In the present form of this scheme, forward differences are used for all spatial derivatives in the predictor step while backward differences are used in the corrector step. This type of differencing is alternated at every other time step in order to eliminate any bias due to the one-sided differencing and to ensure symmetric computation. An example for this symmetric sequence is given below for two temporal steps [32]:

$$\hat{Q}_{i,j}^{n+1} = L_b L_f \hat{Q}_{i,j}^{n} \tag{3.38}$$

$$\hat{Q}_{i,j}^{n+2} = L_f L_b \hat{Q}_{i,j}^{n+1} \tag{3.39}$$

where L_b and L_f are unsplit backward and forward difference operator, respectively.

In order to maintain seccnd-order spatial accuracy, the derivatives appearing in the viscous terms, Eq. (3.4), are differenced with one-sided spatial differencing in the direction opposite to those of the fluxes. The scheme is conditionally stable and the time step is restricted by the CFL condition. In the present study, the following formula of Tannehill et al. [37] is used to determine the time step

$$\Delta t \leq \frac{\sigma\,(\Delta t)_{CFL}}{1 + 2/Re_\Delta} \tag{3.40}$$

where σ is the safety factor (≈ 0.9), $(\Delta t)_{CFL}$ is the inviscid condition given by

$$(\Delta t)_{CFL} \leq \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + a\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}} \right)^{-1} \tag{3.41}$$

Re_Δ is the minimum mesh Reynolds number given by

$$Re_{\Delta} = min\left(Re_{\Delta x}, Re_{\Delta y}\right) \tag{3.42}$$

where

$$Re_{\Delta x} = \frac{\rho |u| \Delta x}{\mu_{e,mix}} \tag{3.43}$$

$$Re_{\Delta y} = \frac{\rho |v| \Delta y}{\mu_{e,mix}} \tag{3.44}$$

where a is the local speed of sound. Before each step, Δt is computed for each grid point using Eq. (3.40). The smallest Δt value is then used to advance the solution over the entire mesh. After each predictor or corrector step, the primitive variables (ρ,u,v,e,P,T,f_s) are found by "decoding" the Q vector.

Flows containing strong shocks often cause numerical oscillation which can lead to program failure. The oscillation is caused by numerical truncation error and can be reduced by mesh refinement. This can be impractical, particularly for calculations for which the oscillation is far removed from the region of interest. In the present study, therefore, fourth-order damping terms known as "artificial viscosity" are added to the differenced governing equations to suppress the numerical oscillations. The fourth-order smoothing scheme used here, which is an alternative form of the fourth-order type of smoothing scheme devised by MacCormak and Baldwin [38], is given by

$$\varepsilon_x \left(\Delta x\right)^4 \frac{\partial}{\partial x}\left[\left(|u| + a\right)\left(\frac{1}{4P}\left|\frac{\partial^2 P}{\partial x^2}\right| + \frac{1}{4T}\left|\frac{\partial^2 T}{\partial x^2}\right|\right)\frac{\partial Q}{\partial x}\right] \tag{3.45}$$

$$\varepsilon_y \left(\Delta y\right)^4 \frac{\partial}{\partial y}\left[\left(|v| + a\right)\left(\frac{1}{4P}\left|\frac{\partial^2 P}{\partial y^2}\right| + \frac{1}{4T}\left|\frac{\partial^2 T}{\partial y^2}\right|\right)\frac{\partial Q}{\partial y}\right] \tag{3.46}$$

where $0 \le \varepsilon_x, \varepsilon_y \le 0.5$ for stability. These smoothing terms have very small magnitudes except in regions of pressure and temperature oscillations where truncation error is already adversely affecting the calculations.

The computer program for this formulation is block structured and vectorized. Its average processing rate on the CRAY-2 computer of the National Aerodynamic Simulation system of NASA is 60μs per iteration per grid point.

3.4 Flow-Adaptive Grids

The computational domain includes a region above the cowl where the flow is hypersonic. The rest of the computational domain is bounded by the lower surface of the cowl and the ramp, where the supersonic flow through the internal nozzle expands (Fig. 3.1). This computational domain is selected to be $(18.5\ H \times 14\ H)$, where H is the combustor exit height. The cowl and the ramp angles are 12° and 20°, respectively. A fixed, boundary fitted grid is generated with appropriate clustering in the regions where high-flow gradients are expected (Fig. 3.2a).

Table 3.1 Computational grid sizes

Block	Idim	Jdim	Total
1	42	41	1722
2	42	41	1722
3	65	42	2730
4	65	41	2665

The global grid, which consists of 8839 cells, is divided into four blocks (Table 3.1). The grid lines are contiguous across the block interfaces, where the solutions from each side of the interface are matched. In the normal direction, the cowl separates Blocks 1 and 2, and a horizontal line extending from the cowl tip to the downstream separates Blocks 3 and 4. In the streamwise direction, the normal line at the cowl tip separates Blocks 1 and 3, and Blocks 2 and 4. This multiblock approach of domain decomposition alleviates the numerical errors that might occur if the boundaries and the interior of the cowl were included in the computational grids [16]. This approach also reduces the computer core memory requirement, since the largest Block (2730 cells) is about 30% of the global grid. The computer core memory needed for this study is about 16 Megabytes.

The governing equations are initially solved on this fixed grid until the global error is reduced by about 2 orders of magnitude. Then the grid is adapted to the current local flowfield solution using the two-dimensional spring-analogy approach of [39]. The flowfield is then recomputed on the adapted grid (Fig. 3.2b). This grid adaptation procedure enhances the solution by reducing the global error by another 2 orders of magnitude.

The adaptation is done as a sequence of one-dimensional operations. For example, the operation starts in the ξ-direction by redistributing the grid points according to a specified weighting function starting from the η=0 line to the $\eta = \eta_{max}$ line. Then the process is repeated in the η-direction on the ξ lines. The weighting function, in the present study, is derived from the gradient of the composite function, [0.5 ρ+ 0.3 u+0.2 γ],a specified minimum step size, and a specified maximum step size.

The η-direction adaptations are performed separately for the region above the cowl (Blocks 1 and 3) and the region below the cowl (Blocks 2 and 4). The ξ-direction adaptations are also performed separately, first for Block 1, then for Block 2, and finally for Blocks 3 and 4 together. At the end, all these separate parts are blended together by applying the adaptations only to the block interfaces. This practice ensures maintaining the original shape of the cowl and the block interfaces. Further details of this flow-adaptive grid scheme, including the necessary equations, are given in Appendix B.

3.5 Boundary and Initial Conditions

Improper treatment of boundary and initial conditions can lead to serious errors and perhaps instability. Therefore, special considerations are required in the determination of these conditions.

The boundary conditions are specified explicitly. At the inflow boundary, the flow is supersonic at the combustor exit and hypersonic at the upstream of the external flow on the upper cowl surface. Hence, the velocities , the pressure , the temperature, and the species mass fractions are specified and kept fixed. However, for viscous flow calculations, appropriate boundary-layer profiles for these properties are required at this boundary. Therefore, the mass fractions of the species are first specified at the combustor exit and at the upstream of the external flow. Then, two-dimensional boundary-layer profiles are generated using a boundary-layer program [40] and imposed at the combustor exit and at the upstream of the external flow on the upper surface of the cowl. These profiles ensure the matching with the experimentally specified Mach number, ratio of specific heats, total pressures, and total temperatures.

At the solid boundary, the walls are considered to be noncatalytic, impermeable, and adiabatic. The wall pressure is calculated from the boundary-layer assumption that its normal derivative vanishes at the wall. The density is then calculated from the state equation. Consequently, the conditions are given by

$$u = 0\,,\, v = 0\,, \frac{\partial T}{\partial n} = 0,\ \frac{\partial p}{\partial n} = 0, \frac{\partial f_s}{\partial n} = 0 \qquad (3.47)$$

At the downstream boundary, the outflow is supersonic and the values of the velocities, static temperature and pressure, and species mass fractions are determined by first-order extrapolation from those immediately adjacent upstream values.The density is then calculated by employing the state equation.

The outer boundary always lies in the freestream of the external flow. Furthermore, since the normal to the outer boundary is normal to the freestream velocity, the outflow at this boundary is subsonic. Hence, the characteristic theory necessitates that one of the flow characteristics point from the outside toward the inside of the computational domain. Therefore, one of the boundary conditions must be specified analytically. In this study, the pressure at the outer boundary is set equal to the free-stream value and the remaining flow variables are computed from the interior flow solution by a zeroth-order extrapolation

$$\frac{\partial u}{\partial \eta} = 0 \, , \frac{\partial v}{\partial \eta} = 0 \, , \frac{\partial T}{\partial \eta} = 0 \, , \frac{\partial f_s}{\partial \eta} = 0, P = P_\infty \tag{3.48}$$

Due to the decomposition of the computational domain into four blocks, additional boundaries are introduced at the block interfaces. The solutions are matched at two sets of grid points on each side in the normal direction to an interface. This provides second-order spatial accuracy consistent with the accuracy of the solution algorithm.

The governing equations, Eq. (3.1), also require a set of initial conditions. The flowfield is initialized by setting the values of velocities, static temperature and pressure, and species mass fractions at each block to the values of the corresponding upstream boundary-layer profile.

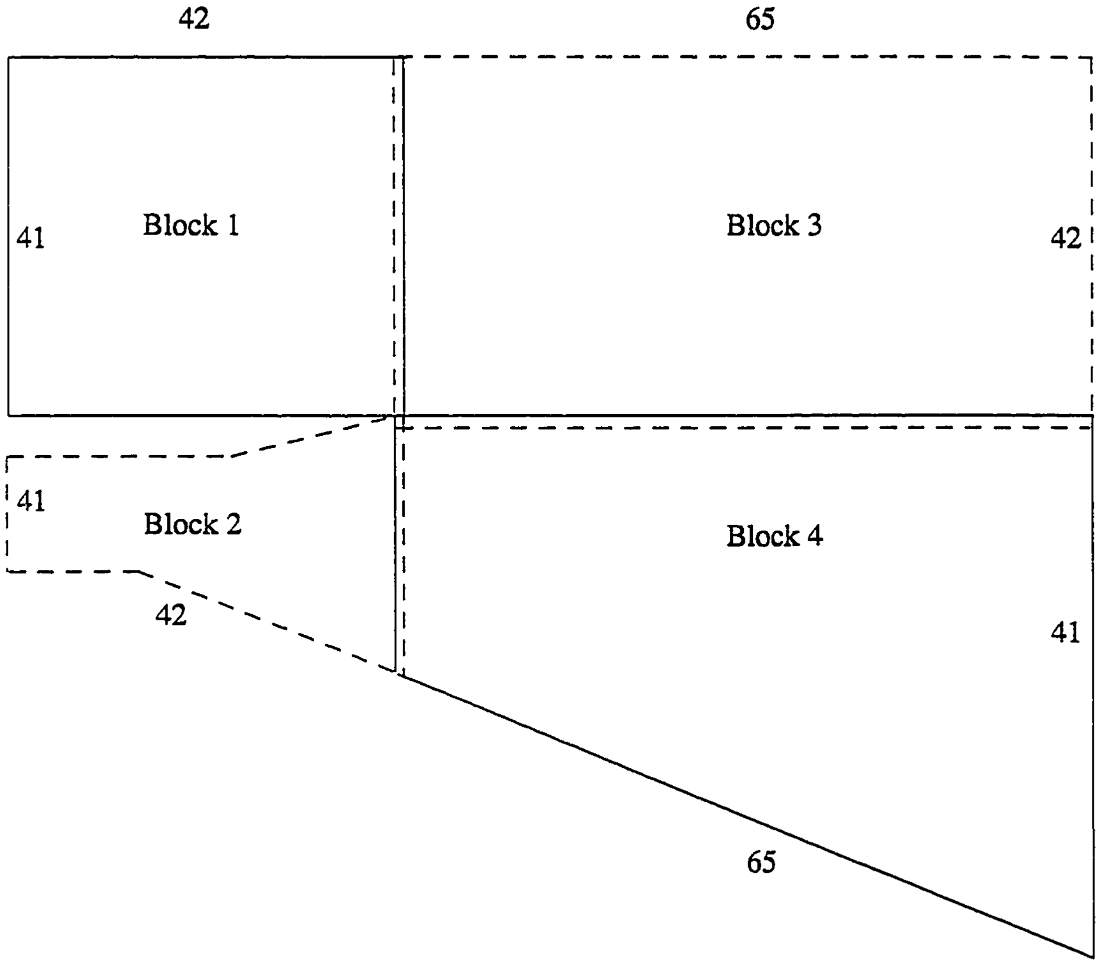


Fig. 3.1 Dimensions of the blocks forming the computational grid.

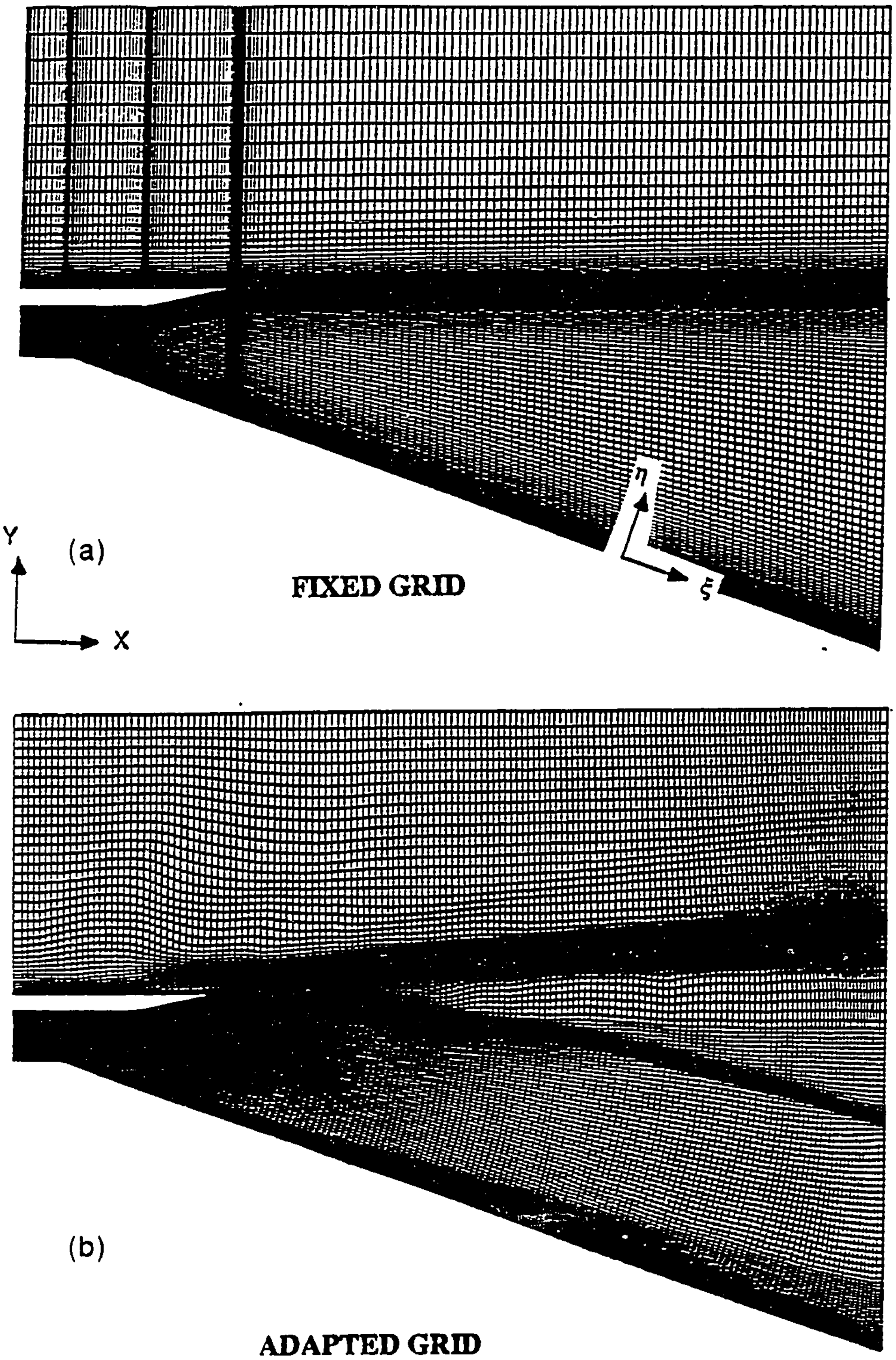


Fig. 3.2 The grids used for the computations: (a) fixed grid for initial computation (b) a flow-adapted grid for improved solutions.

Chapter 4

RESULTS FOR NOZZLE-AFTERBODY FLOW ANALYSIS

A study of four cases is included in this chapter in an attempt to answer the objectives and issues raised in Chapter 2. The upstream conditions of the nozzle exhaust flow and the external flow are given in Table 4.1, where the combustor exit height, *H*, is 0.01524m. All of the flows are considered to be fully turbulent. Only Case 1 assumes the exhaust gases to be air in order to show the differences between expanding air as opposed to a cold simulant gas. In this study, the cold simulant gas for the exhaust gases is a Freon-12-Argon mixture (Cases 2-4).

Table 4.1 Flow conditions at upstream of computational domain

Case	Flow	Fluid (by volume)	Mach No.	Reynolds No. based on *(H)*	Total temp. (°K)	Total pressure (kPa)
1	Combustor exit	Air: 79% N_2 21% O_2	1.7	192,000	475	166.0
2-4	Combustor exit	50% Freon-12 50% Argon	1.7	7500	467	172.4
1-4	External flow	Air: 79% N_2 21% O_2	6.0	346,000	478	2517.0

The computational models for (Cases 2-4) are presented in Table 4.2. Case 1 assumes a homogeneous composition of the fluid everywhere and at all times. Therefore, the species continuity equations and the terms representing the multispecies mixing are not used for Case 1. However, air is assumed to be only thermally perfect and calorically real. Computed in Cases 2-4 are the mixing of four species, namely, Nitrogen, Oxygen, Freon-12, and Argon. The complete multicomponent diffusion model is used in Case 3, but the binary diffusion model is used in Cases 2 and 4. Only Case 4 does not assume that the sum of mass ratios of all the species is unity (Eq. (3.7)).

Table 4.2 Computational mixing models

Case	Multispecies mass transport	Specific heats ratio is function of	Diffusive mass transport model	No. of solved partial differential equations
1	No	Temperature	—	4
2	Yes	Temperature and composition	Eqs.(3.29), (3.30), (3.32)	4 + 3
3	Yes	Temperature and composition	Eqs.(3.25), (3.30), (3.33), (3.34)	4 + 3
4	Yes	Temperature and composition	Eqs.(3.29), (3.30), (3.32)	4 + 4

Before discussing the physical aspects of the individual cases, it is appropriate to show the accuracy of the computational models. Generally speaking, it is rather difficult to quantify the computational uncertainty for problems where no exact solutions exist. An indicator of the computational accuracy is the comparison of the computed results with available experimental data. Presented in Fig 4.1 are the computed and experimentally measured pressure distributions on the ramp surface. All pressures are normalized with the pertinent pressure values at the upstream corner, P_3. The rate of expansion of airflow (Case 1) is much higher than that of Freon-Argon mixture (Cases 2-4) at the corner. The difference between the expansion rates gradually decreases, but the pressure ratios of

Freon-argon mixture are consistently higher than those of air. The computed pressure values from Cases 2-4 are very close to each other. In comparison with the experimental data, they are initially slightly lower, then slightly higher. The computed values of surface pressures for the air expansion (Case 1) are also slightly lower than the experimental values, but they match almost identically down the ramp. The major causes for the discrepancies are related to the following: firstly, an estimated boundary-layer profile is computationally prescribed for the upstream flow conditions at the combustor exit, where the boundary-layer profile is not measured experimentally.

In an attempt to assess the effect of employing the commonly used assumption, that the sum of computationally obtained species mass ratios is unity (Eq. (3.7)), on the solution accuracy (e.g., Ref. [28]), a flow analysis case, where this assumption is not used, is performed (Case4). The local mass error, Eq. (3.35), is computed first and the results are plotted in Fig. 4.2. This error, of course, is identically zero for the other cases. As expected, this mass error is occuring within the shear layer, where most of the mixing takes place. It grows in the downstream direction to a maximum of 1%. This indicates that Eq. (3.7) is not just an assumption, but is a necessary closure equation for the conservation of mass in mixing flows.

Next, in order to quantify the global mass conservation error resulting from not using Eq. (3.7), the mass deficits of Cases 2 and 4 are computed (Fig. 4.3). The mass deficit is defined herein as the difference between the numerically integrated outflux and influx of mass through the boundaries of the computational domain. The solutions of Cases 2 and 4 show convergence in about 4000 iterations (pseudo-time steps). The mass deficit is less than 1% for Case 2, where Eq. (3.7) is being used in addition to three other species continuity equations to solve for four species mass fractions. When Eq. (3.7) is not used, and consequently four species continuity equations are solved for four species (Case 4), the mass deficit increases to about 8%. This reconfirms the above finding that Eq. (3.7) is not just an assumption, but is a necessary closure equation for the conservation of mass in mixing flows.

In order to gain an insight into the nature of the nozzle-afterbody flowfield, Cases (1-3) are thoroughly analyzed. The solution for Case 1 is obtained first on a fixed grid, then on two consecutively flow-adapted grids. The velocity deficit of the shear layer (Fig. 4.4) is captured more accurately due to the adaptation, which results in clustering the grid along the shear-layer region. The structure of the flowfield is shown through the density contours (Fig. 4.5a). The internal flow experiences two centered expansion fans prior to clearing the cowl exit plane and the expansion continues downstream along the ramp. This is illustrated further by the pressure contours (Fig. 4.5b). The internal nozzle flow underexpands at the cowl exit plane, where the pressure ratio (based on the combustor exit value) varies from 0.054 at the cowl tip to 0.20 at the ramp. The minimum pressure ratio of the flowfield (based on the combustor exit value) is 0.03. The pressure difference across the cowl varies from 9.988 kPa at the corner to 0.908 kPa at the tip. The shear layer originates at this point with an expansion, and deflects up 7° with an included angle of 8°.

The density contours for the flow of the Freon-argon mixture are shown in Fig.4.6. Although the general structure of this flow resembles that of air (Case 1), there are significant differences. The rate of expansion is slower, and the spreading rate of the shear layer is higher. The deflection of the shear layer is 10° and the included angle is 15°. This is mainly attributable to the molecular weight of the simulant gas being three times larger than air. The pressure ratio (based on the combustor exit value) at the cowl tip plane varies from 0.106 at the cowl tip to 0.50 at the ramp. The pressure difference across the cowl varies from 16.4 kPa to 2 kPa at the tip. The flow expands further downstream of the external nozzle to achieve a minimum pressure ratio of 0.055 at the ramp tip. This is shown for

Case 3 in Fig. 4.7. This indicates that the Freon-argon mixture expands with a slower expansion rate than that of air, which consequently results in higher forces acting on the afterbody.

The contours of the specific heat ratio (γ) are plotted in Fig. 4.8. It varies throughout the flowfield as a function of temperature and local gas composition. In Case 1, however, the variation of γ is only due to temperature. Since the temperature variation is relatively small in the cold-gas simulations, the variation of γ from the upstream value of 1.4 is negligible (± 0.02). For Cases 2-4, the value of γ is 1.196 at the combustor exit, and 1.4 at the upstream boundary-layer edge above the cowl. The values of γ change in the streamwise direction and the normal direction, with the large gradients being in the shear-layer and boundary-layer regions.

The mass fraction contours for Freon-12 and Nitrogen of Case 3 are shown in Fig.4.9. The fluid composition at the edges of the shear layer is slightly different from its upstream mixtures. There is a very large gradient of mixture composition through the shear layer. Some Freon-12 and Argon mixture is entrained upstream with the reversed flow on the upper surface of the cowl. When Figs. 4.8 and 4.9 are inspected together, it is observed that the major cause for the variation of γ in the shear layer is the change in the composition of the multispecies fluid. In other regions, such as near the walls, the variation of γ is primarily due to temperature gradients.

Effects of the diffusive mass transport models can be observed by comparing Fig.4.6a with Fig. 4.6b and comparing Fig. 4.8a with Fig. 4.8b. The complete multicomponent diffusion model, i.e., Case 3, produces smoother shear layer and boundary layers. The binary model (Cases 2 and 4) produces more oscillations, with quantitative results varying by about 2%. These oscillations exist despite the apparent converged solution of Case 2 as indicated by an examination of Fig. 4.2. Therefore, the extra computational cost of the model used in Case 3 may be justified only if better accuracy and oscillation-free solutions are desired. It should be pointed out that the flows under consideration are turbulent, high-speed flows. The differences in the results from these two models are expected to be more pronounced for laminar flows.

The main findings of the nozzle-afterbody flow analysis can be summarized as follows:

(1) A reversed flow region is detected at the cowl tip due to the large pressure ratio between the nozzle exit plane and the freestream.

(2) The deflection of the shear layer depends on the utilized simulant gas; the heavier simulant gas expands at a slower rate than air.

(3) Using flow adaptive grids improves the resolution and the quality of the solution by decreasing the computational errors.

(4) Modeling the diffusive mass transport by the multicomponent diffusion equation, Eq. (3.33), is computationally more expensive. However, it produces better results than that obtained by using the Fick's law (Eq. (3.32).

The above findings help the aerodynamic design formulation of the nozzle-afterbody configuration given in Chapter 7. In addition, they augment the experimental studies of cold gas simulation. Some conclusions and recommendations on the flow analysis of the nozzle-afterbody configurations are given in Chapter 11.

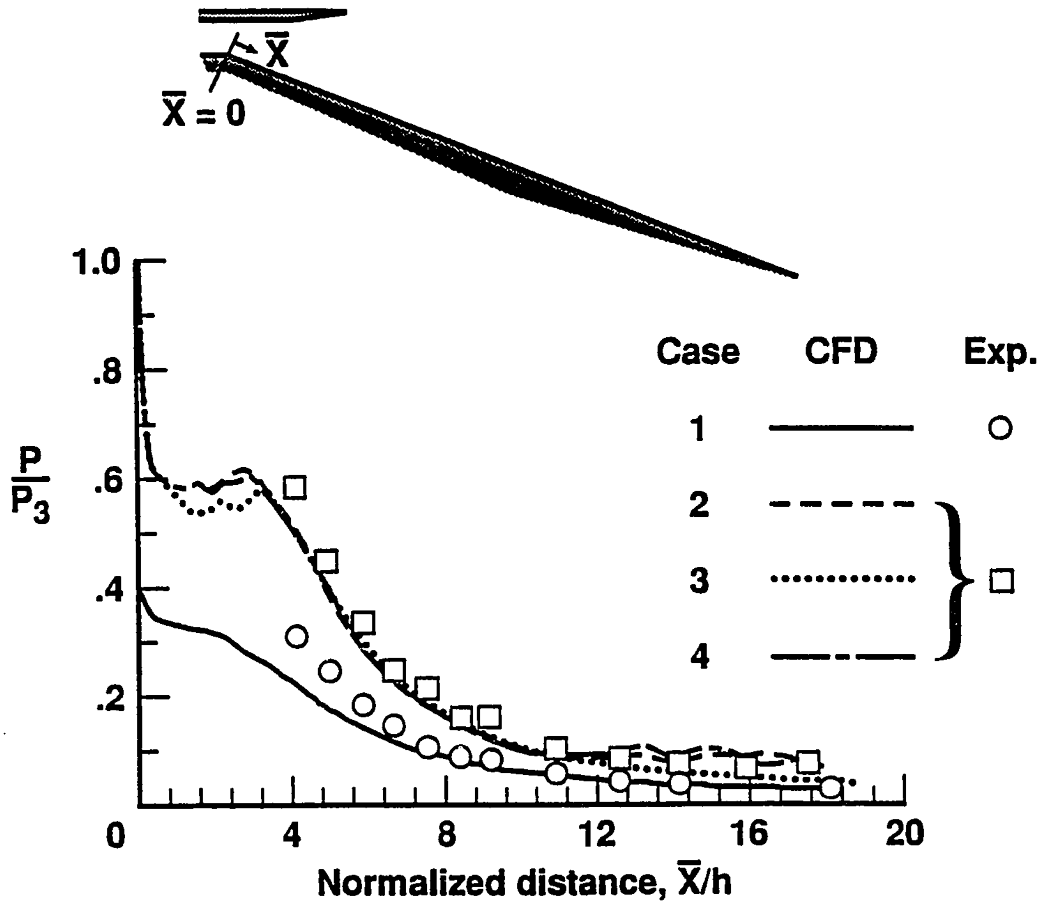


Fig. 4.1 Pressure distribution on the ramp surface.
(The experimental data are from Ref. [5]).

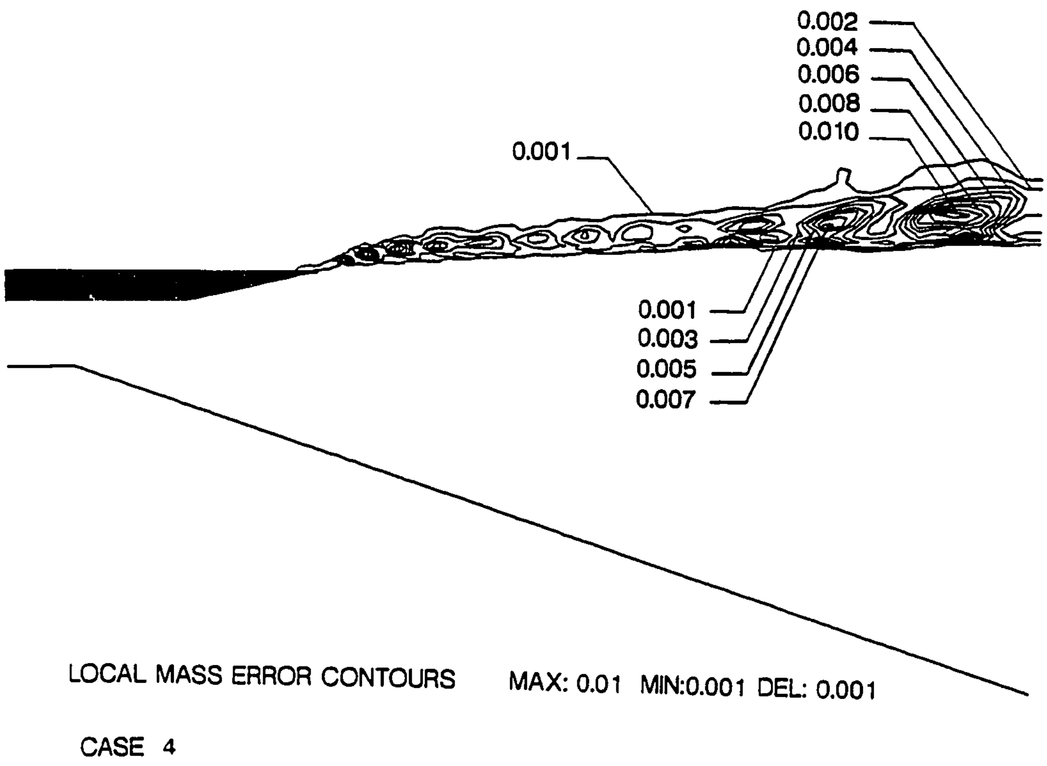


Fig. 4.2 Local mass error due to mixing of species (Case 4).

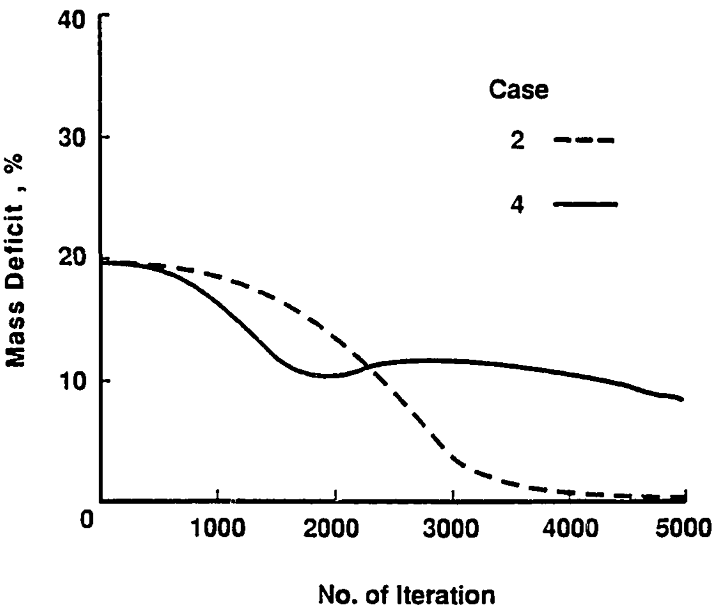


Fig. 4.3 Global mass conservation error for Cases 2 and 4.

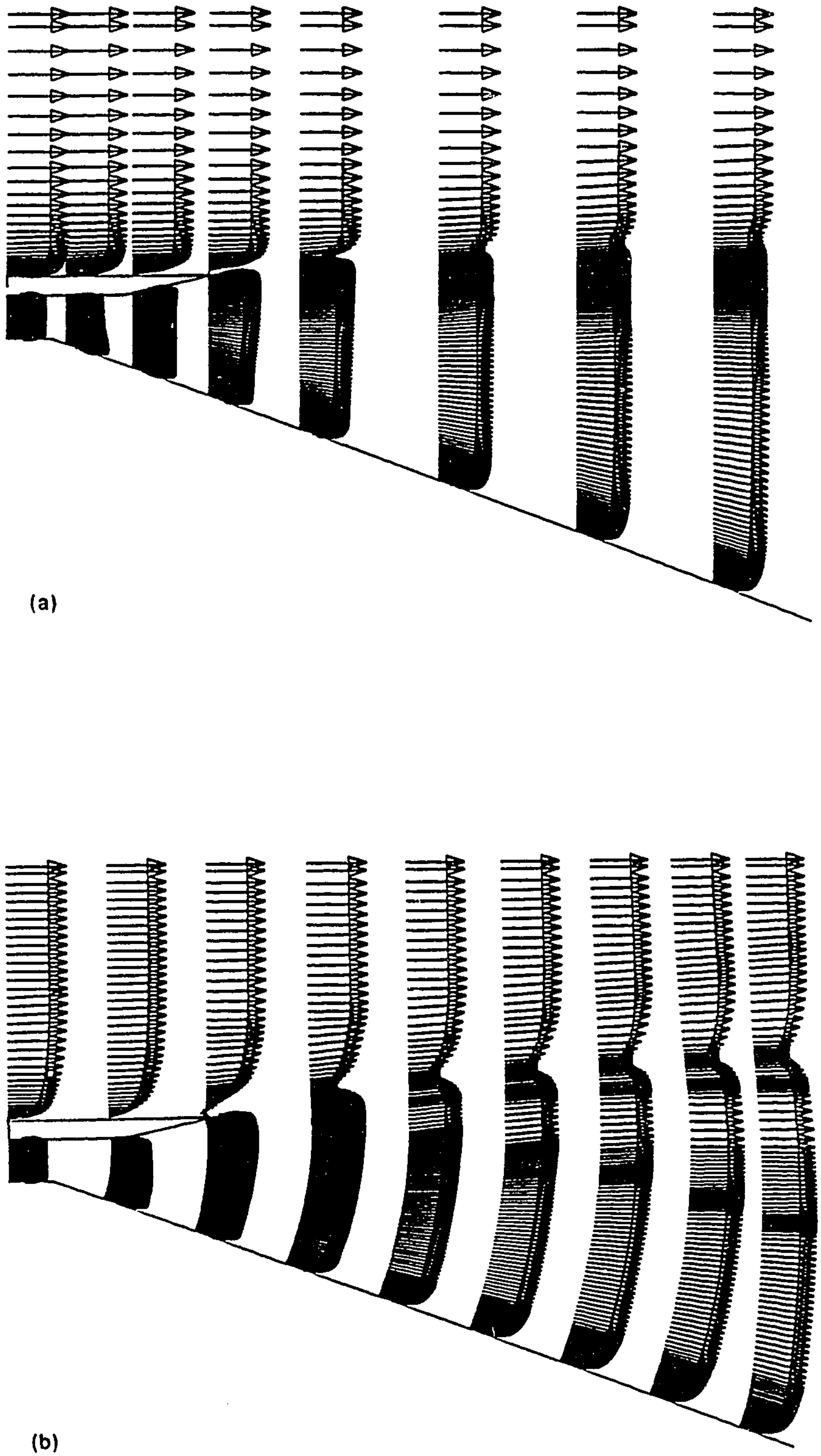


Fig. 4.4 Velocity vectors of Case 1: (a) on fixed grid (b) on flow–adapted grid.

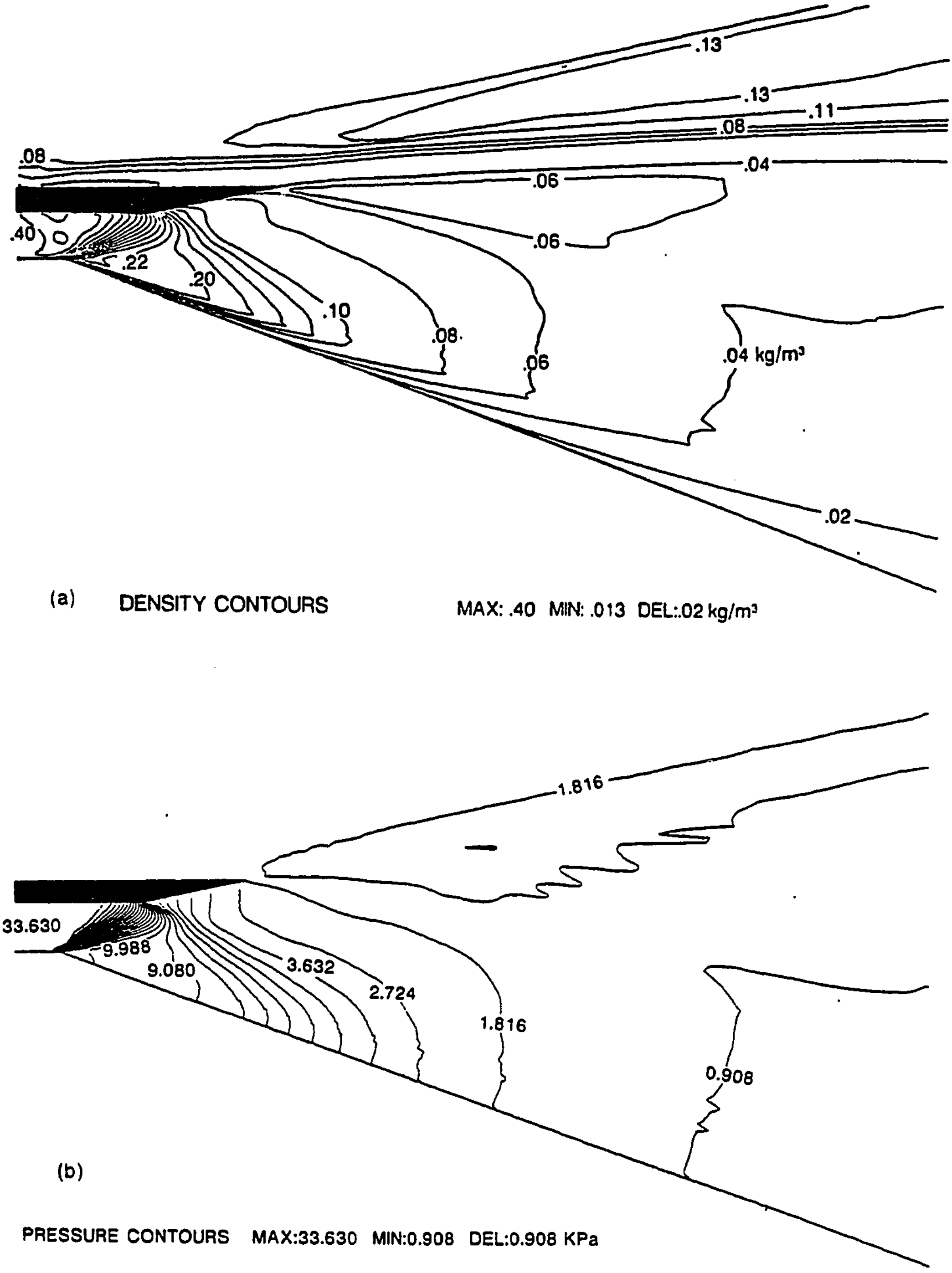
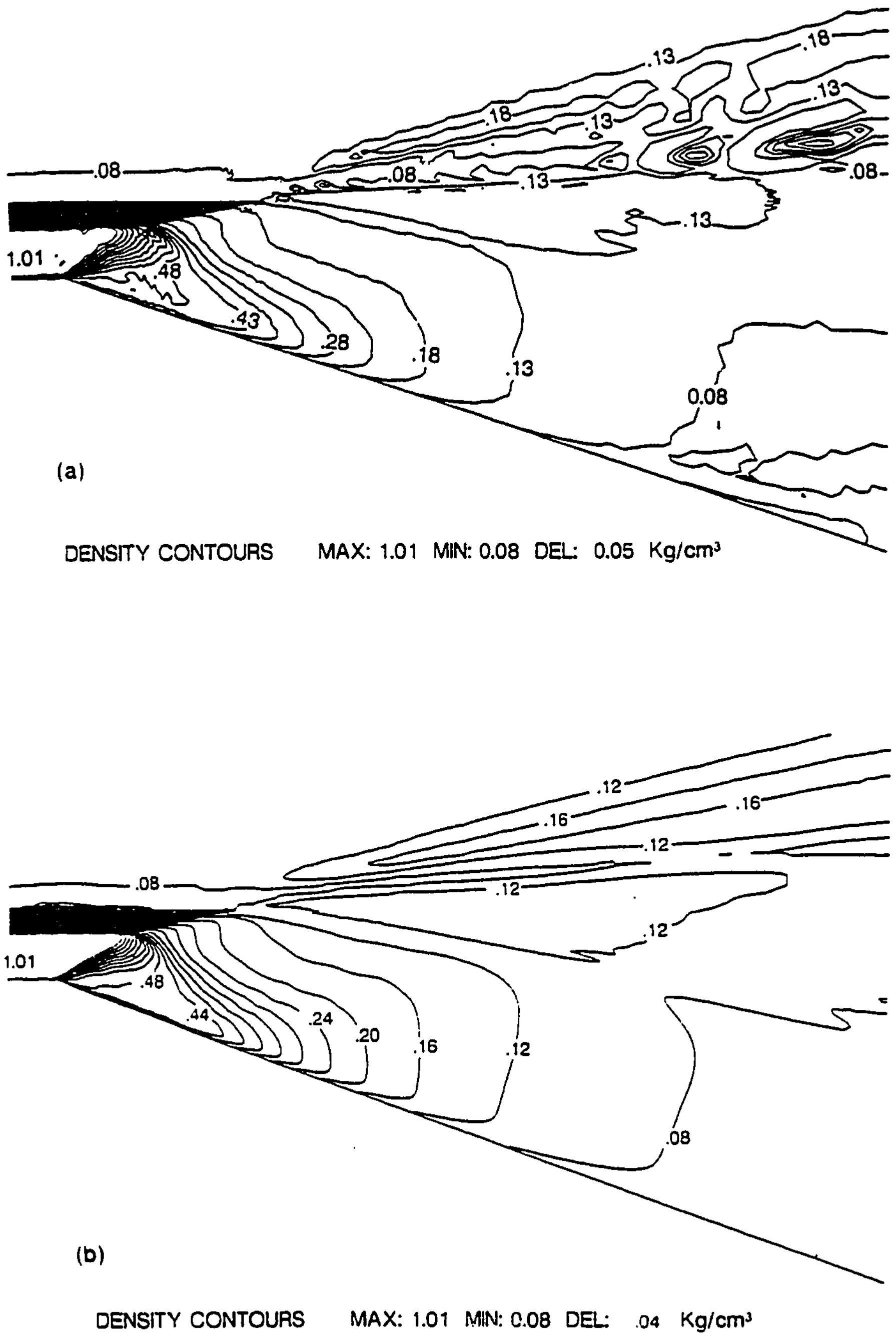


Fig. 4.5 (a) Density contours of Case 1 (b) Pressure contours of Case 1 ($\gamma \approx 1.4$).



DENSITY CONTOURS MAX: 1.01 MIN: 0.08 DEL: 0.05 Kg/cm^3

DENSITY CONTOURS MAX: 1.01 MIN: 0.08 DEL: .04 Kg/cm^3

Fig. 4.6 (a) Density contours of Case 2 (b) Density contours of Case 3 (variable -γ).

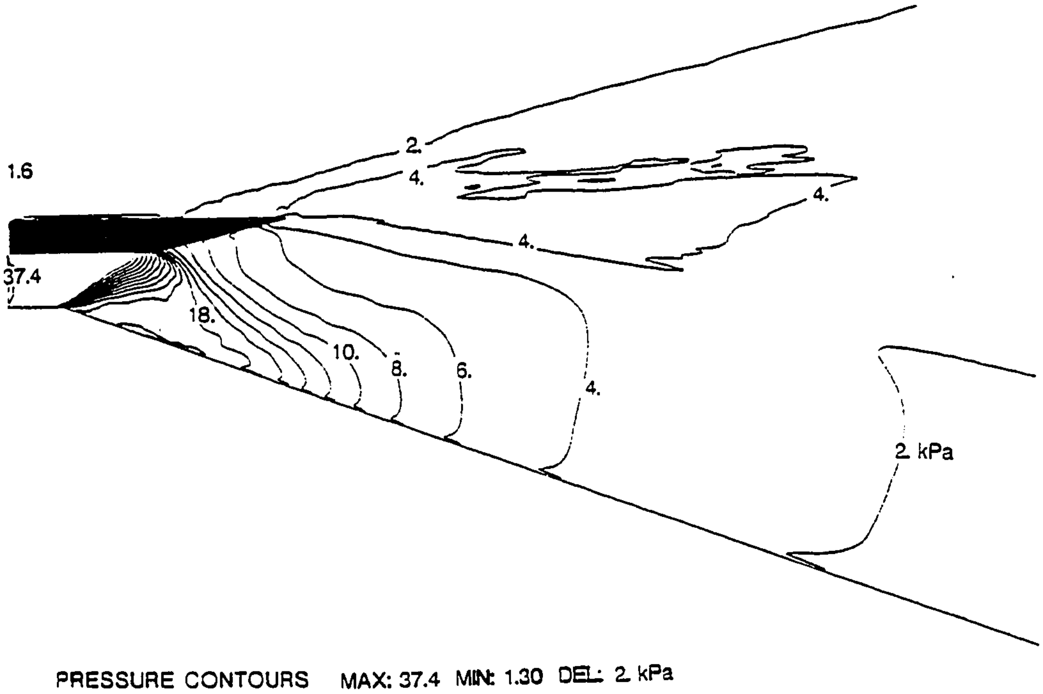


Fig. 4.7 Pressure contours of Case 3.

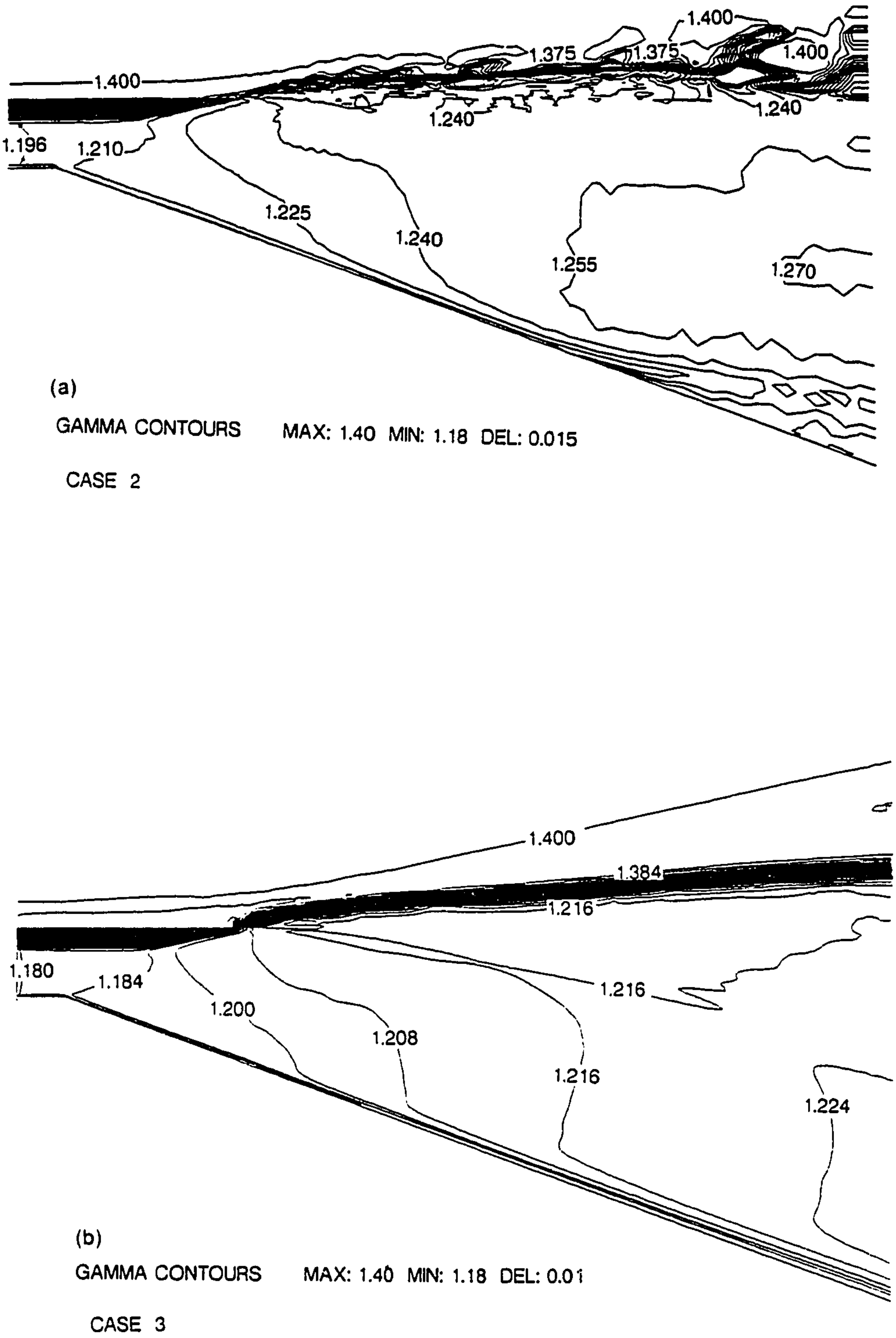


Fig. 4.8 Contours of the ratio of specific heats: (a) Case 2 (b) Case 3.

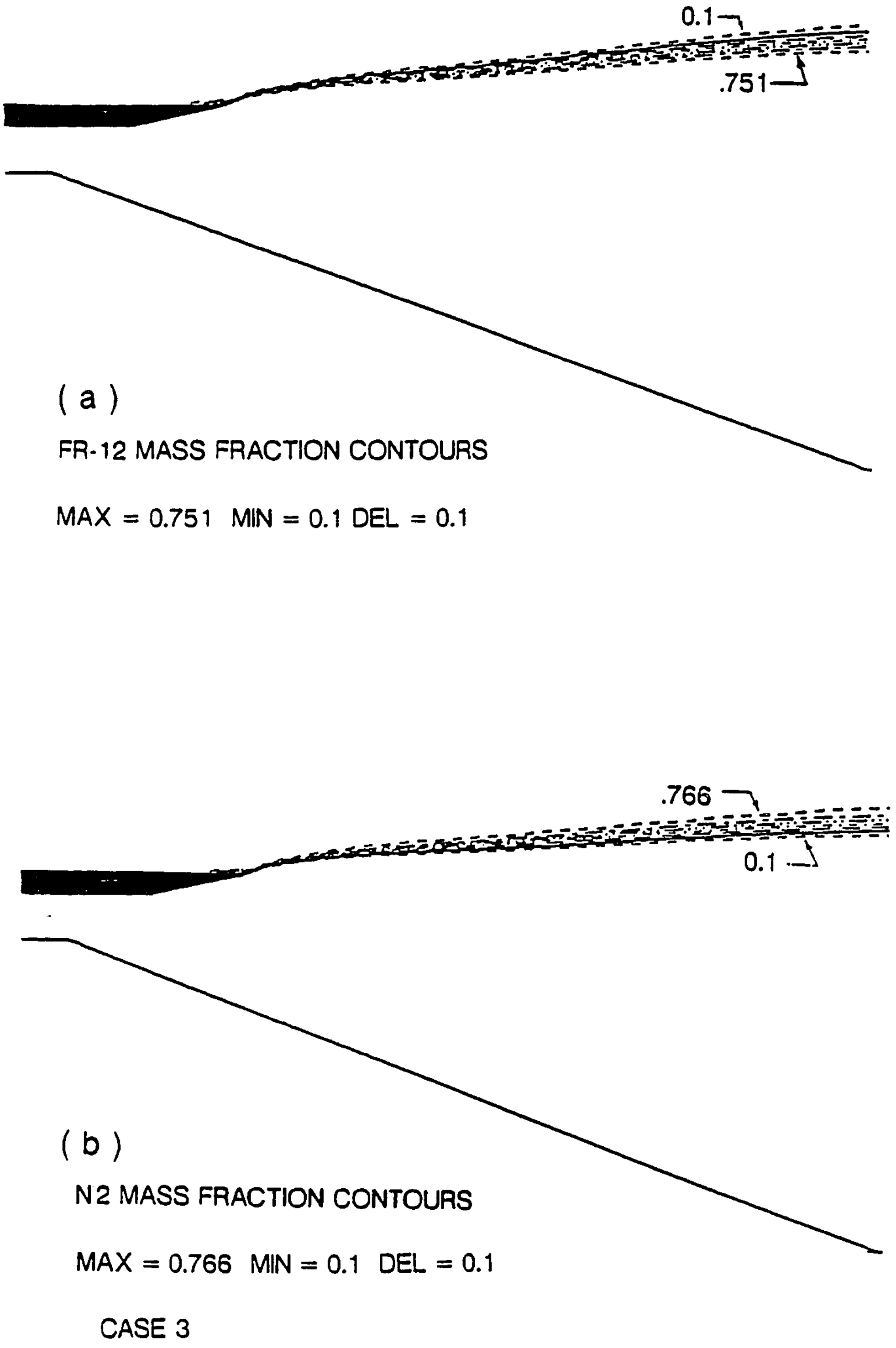


Fig. 4.9 Mass fraction (fs) contours of Case 3: (a) Freon-12 (b) Nitrogen.

PART II: AERODYNAMIC DESIGN OPTIMIZATION

Chapter 5

INTRODUCTION TO DESIGN OPTIMIZATION

5.1 Rationale

The field of aerodynamic shape optimization is still a relatively new field undergoing rapid changes in methods and focuses. This field of study involves the ability to determine the boundary shape of aerodynamic configurations. Usually, these configurations have to achieve certain aerodynamic characteristics subject to given constraints and have to satisfy the governing equations of their flowfield. Generally, the aerodynamic shape optimization methods are classified into two basic groups, where several methods have been developed in each one of them. The first group is the inverse design methods whereby the aerodynamic shape is generated to match some specified aerodynamic characteristics. The second group is the numerical optimization methods, which couple a flowfield analysis algorithm with an optimization algorithm. With these methods, the engineer is free to choose any function to be maximized or minimized without imposing preconceived conditions about the optimum. The present aerodynamic shape optimization method falls into the latter group of methods.

Due to the rapid advances in computer hardware and architecture, computational fluid dynamics (CFD) has evolved to a level of maturity to simulate complex flows with reasonable fidelity to fluid physics. As a result, aerodynamic shape optimization now more than ever includes the results of CFD as an important ingredient of its procedure. However, most design engineers employ complex general-purpose CFD software packages for flowfield analysis, and have only scant knowledge of the details of the analysis algorithms used in these software packages. Therefore, a major challenge faced by researchers in aerodynamic optimization is to develop methods that are suitable for use in conjunction with such software packages. Another challenge is the high computational cost associated with the flow analysis of many complex real-life problems. In many cases, the designer cannot afford to analyze the flowfield more than a handful of times. A last challenge associated with the aerodynamic shape optimization is the computational mesh deformation due to the changes in the aerodynamic configuration. Simple remeshing rules that translate grid points at the boundaries usually lead to highly skewed computational meshes, which are not suitable for CFD flow analyses. This environment motivates a focus on aerodynamic shape optimization procedures that call for minimal interference with the CFD analysis package, require only a small number of CFD analyses, and use efficient remeshing techniques.

Typically, the aerodynamic shape optimization procedure consists of the following major components (Fig. 5.1): (1) formulation of the objective function and the pertinent physical as well as geometrical constraints; (2) definition of the aerodynamic shapes, either by explicitly prescribing these shapes from a matrix of basis shapes, or preferably, by determining them automatically through the optimization algorithm; (3) CFD analysis; (4) gradients of the objective function and the constraints with respect to the design variables (sensitivity coefficients); and (5) optimization algorithm.

The first component of the aerodynamic shape optimization procedure is case dependent and requires a level of expertise in the field of application in question. The second component also requires a great deal of experience and skill, if the matrix of candidate shapes for optimization is to be prescribed effectively. However, this latter requirement can almost be eliminated, if a method is developed to automatically shape the aerodynamic configuration to any arbitrary geometry during the optimization process.

The third and fourth components are the dominant contributors to the computational cost as well as to the accuracy of the results. Therefore, it is desirable to develop an approximate analysis method that can predict the flowfield solution for any modification of the shape at a computational cost which is equal to a small fraction of the complete CFD analysis cost. Furthermore, generating the first-order gradients of the objective and the constraint functions by a finite-difference approximation requires repeating the aerodynamic analysis with incremented values of the design variables. This simple approach has the disadvantage of being potentially computer intensive, particularly when a complex set of governing equations is being used for the flow analysis. A preferable approach is to obtain the sensitivity coefficients directly and analytically from an appropriate set of equations to eliminate the costly and repetitive flowfield analyses.

The resources required for the solution of the set of equations resulting from the analytical calculations of the sensitivity coefficients usually increase with the dimensionality of the problems at a rate which is more than linear. That is, if the number of design variables or the dimensionality of the aerodynamic configuration increases, the cost of the solution will typically be more than double. Large problems may also require excessive computer memory allocations. For these reasons, it is desirable to seek ways for breaking a large problem into a series of smaller problems. One of the popular methods for achieving such a break-up is the domain-decomposition technique. The process of decomposition usually consists of dividing the computational domain around the aerodynamic configuration into a number of smaller subdomains and then solving each subdomain separately. This technique not only requires low computer core memory, but also eases the generation of the computational grids for complex-geometry problems.

The fifth component is also a field of intense research, where many unanswered issues still exist. A perennial question, for example, is determining the criteria for the global optimum. Such issues are beyond the objectives of the present study, but rather the approach here is to utilize the existing optimization algorithms in a modular manner. Nevertheless, some of these algorithms need to be modified in order to minimize the interference between the optimization package and the CFD analysis package.

5.2 Literature Survey

In the following section, the design optimization methods are broadly classified according to their origin and concepts. Examples of the pioneering works in each class are also given in this section. Sensitivity analysis is categorized based upon its concepts and it is reviewed in Section (5.2.2). Applications of sensitivity analysis to decomposed computational domains have been nonexistent in the field of aerodynamic optimization to the present date and the closest applications have been in the field of structural mechanics. Hence, only the earlier germane studies in that field are reviewed in Section (5.2.3).

5.2.1 Design Optimization Methods

In attaining the optimum geometry of a configuration, two basic groups of aerodynamic optimization methods can be found in the literature, where several methods were developed in each group.

The first group of design optimization methods is called the *inverse design method*, in which one has to specify target aerodynamic characteristics that need to be met at the optimum design point (e.g., the surface pressure distribution). Inverse methods were widely used in the design of wings and airfoils because of their computational efficiencies. For example, Bauer et al. [41] developed a hodograph method to design wings. This method solved the full potential equations in the hodograph plane, and hence could not guarantee a proper treatment of discontinuities, such as shocks, occuring sometimes in the flowfield. To overcome this difficulty, Giles and Drela [42] developed another method, based on the Euler equations, which treated the two-dimensional flows around an airfoil as a set of streamtubes coupled through the position of, and pressure at, the stream interfaces.

Recently, some of the inverse design methods included the viscous effects through a coupled integral boundary layer analysis. Examples of this approach include the methods developed by Drela and Giles [43,44], and Lee and Eyi [45]. Another approach to include the viscous effect is the one applied by Malone et al. [46], whereby two-dimensional Navier-Stokes equations were used in the design of an airfoil. All the above methods were applied to configurations at the transonic speeds. For supersonic and hypersonic flow regimes, Lee and Mason [47] recently developed a three-dimensional inverse design method based on the Euler equations.

Instead of achieving the prescribed aerodynamic characteristics, some design methods use a constrained optimization process to improve the design by minimizing or maximizing some aerodynamic quantities, such as drag, lift, or pitching moment. These methods constitute the second group and are commonly called the *numerical optimization methods*. Unlike the inverse design methods, numerical optimization methods were not widely used in practical airfoil design primarily due to the large amounts of computational resources needed. Nevertheless, the methods will continue to be developed since they have many advantages, such as the automated design capability, ability to handle multi-point design, varieties of constraints, and their amenability for inclusion into multidisciplinary design of complete vehicles.

Examples of the numerical optimization methods, when applied to airfoil design at the transonic flow regime, were presented by Hicks et al. [48], Vanderplaats [49,50], Chow [51], and Lee and Eyi [52]. At supersonic flow regime, Pittman [53] applied one of these method to airfoil design. Generally, these methods require many flowfield analyses to develop converged designs. Some improvements of the efficiency of these methods were obtained by using Rizk's method [54]. This method was based on a numerical optimization methodology but it allowed for a stable iterative algorithm, where an optimizer was used on each updated configuration even before the flowfield solution converged for a new geometry. As a consequence, a typical airfoil design involved only an equivalent of 5 to 10 fully converged flowfield solutions. However, like the other methods in this group, Rizk's method still used finite-difference approximation to calculate the gradients of the objective function and constraints. Sorensen [55] developed a design tool for airfoil designs which was based on an integral boundary layer scheme [56]. Although in this new capability the gradient information was calculated analytically, it still required many evaluations of the objective function and constraints before reaching the optimum design. In addition, this design tool, which is known as XFOIL, is a specialized package for airfoil design and cannot be applied to other aerodynamic applications.

Some conclusions can be drawn from this brief literature survey. First of all, each of these groups has its advantages and its shortcomings, hence they have to be considered as complementary and not competitive. Secondly, each offers a different way of finding efficient aerodynamic shapes without resorting to an expensive cut-and-try wind tunnel testing. However, the general theme in all of them is that they require a significant level of experience and skill to define both the target aerodynamic characteristics and the geometrical aerodynamic shapes. For completion of the above survey, the readers should be referred to a survey paper by Dulikravich [57], where other aerodynamic design concepts were presented and classified according to their origin.

5.2.2 Sensitivity Analysis

One of the key elements of an optimization analysis is the sensitivity analysis. The sensitivity analysis helps to assess the influence of variations of the design variables on the analytical model, and guide the selection of the important design variables. In fact, the sensitivity analysis constitutes a major part of the total optimization calculation. Hence, it is important to carry out the design sensitivity analysis as efficiently as possible, especially for complex aerodynamic optimization problems.

The sensitivity analysis methodology has been available for well over two decades in structural mechanics [58,59]. However, nonstructural sensitivity analysis, such as aerodynamic sensitivity analysis, was nonexistent until recently. The major contributors for such a phase lag are the nonlinear nature of the fluid dynamic equations, existence of discontinuities, such as shocks, and the difficulties associated with the aerodynamic surface and volume grid generation. Also, unlike most of the structural problems, the coefficient matrix of the sensitivity equation for aerodynamics is not symmetric. Only recently and due to the advent of the supercomputers with large memories, it was possible to solve an aerodynamic sensitivity equation, since the size of a CFD grid is generally much larger than a structural grid.

Following the structural mechanics classification [60], aerodynamic sensitivity analyses may also be broken into two categories, which differ basically in how the sequence, in which the discretization of the governing equations and their differentiations with respect to design variables take place. In the first category, called the *discrete sensitivity analysis,* the continuous flowfield governing equations are discretized before they are differentiated with respect to design variables. Whereas, in the second category, called the *variational sensitivity analysis,* the continuous flowfield governing equations are differentiated with respect to design variables first and then they are discretized, if it is necessary.

In 1986 Sobieski [61] made a plea to the CFD community for extending their present capability to include the sensitivity analysis. This was followed in 1987 by Yates [62] who proposed a new variational sensitivity analysis method which was based on the linearized lifting-surface theory and calculated the rates of change of lifting pressures with respect to general changes in aircraft geometry, including planform variations. This method was applicable to steady and unsteady planar and nonplanar lifting surfaces in subsonic, sonic, and supersonic flow. In 1988, Jameson [63] proposed another variational sensitivity analysis method which was based on Euler equations and used the control theory concepts. The method regarded the aerodynamic design problem as a control problem in which the control function was the shape of the boundary, and consequently, it was the mapping function. The variational sensitivity equations were obtained by first rewritting the Euler equations in terms of the mapping function and then differentiating the resulting equations with respect to design variables. Recently, this method wa applied to the airfoil design at transonic speeds by Lewis et al. [64].

Much of the literature concerning the discrete aerodynamic sensitivity analysis methods appeared in the past three years. For example, in 1989 Elbanna and Carlson [65] applied one of the discrete aerodynamic sensitivity analysis methods to determine the sensitivity coefficients for the nonlinear, transonic, small perturbation equation in two dimensions. This was followed by Baysal and Eleshaky [66] in 1990 who applied three different methods of the discrete aerodynamic sensitivity analysis to determine the aerodynamic sensitivity coefficients based on the Euler equations. In addition, they developed a new method to predict the flowfield solution and successfully compared it to the flowfield solution obtained via an analysis code.

In an effort to assess the differences between the results of a variational sensitivity analysis method and a discrete sensitivity analysis method, Frank and Shubin [67] applied the two methods to one-dimensional duct flow problem using Euler equations in 1990. Their results showed that the two methods are equivalent to each other; however, the discrete sensitivity analysis methodology can be retrofitted to most existing analysis codes to turn them into design codes. In 1991, Baysal and Eleshaky [68] incorporated their sensitivity analysis methodology in an aerodynamic design optimization procedure and developed a new efficient method for aerodynamic design optimization. They also investigated the sensitivity of the optimum solution to the variation in problem parameters. In 1991, another research effort by Taylor et al. [69,70] emerged in the area of discrete aerodynamic sensitivity analysis. They focused on the development of an approximate analysis method similar to the one developed by Baysal and Eleshaky [66]. However, the derivatives of the geometrical terms in their sensitivity equations were treated inefficiently using finite-difference approximations instead of the direct differentiation used in Ref. [66]. Smith and Sadrehaghighi [71] presented an analytical approach to determine the derivatives of the coordinates of the grid points with respect to design variables in late 1991.

5.2.3 Sensitivity analysis on Decomposed Computational Domain

The process of shape optimization begins with an initial design, generates a sequence of improved designs in an iterative algorithm and finally obtains the optimum shape. A reduction in the size of the model being iterated upon in this process can clearly increase the ability to treat presently very large aerodynamic problems. One way to achieve this is the use of domain decomposition.

The idea of domain decomposition has a long history in CFD. This technique was used to ease the generation of grids for complex-geometry problems. A complex geometry configuration may have multiple, joint or disjoint components, all of which may be geometrically nonsimilar [72-74]. A Hybrid Domain Decomposition (HDD) approach reported in Refs.[72-74] was developed in order to obtain the flowfield solutions for these problems. HDD takes advantage of all popular domain decomposition techniques used in CFD, that is, multiblock grids, zonal grid, and overlapped grids. However, in the present study and at the early stages of aerodynamic sensitivity analysis, only the multiblock approach is considered.

Aerodynamic sensitivity analysis applied to subdomains (multiblocks) is very similar to the sensitivity analysis applied to the substructures recently reported in Ref. [75] for structural mechanics. However, aerodynamic sensitivity analysis on decomposed computational domains has never been investigated to this date. The present study represents the first successful attempt in this field and the developed scheme will be referred to as *Sensitivity Analysis Domain-Decomposition (SADD)* scheme.

In the field of structural mechanics, the domain decomposition is usually called substructuring. The original motivation of domain decomposition was to break up a large problem into small parts which could be solved separately, and then the final solution could be obtained from these subproblems [76]. The substructuring process can be continued by substructuring the subproblems; this idea was developed by George [77] and was called *nested dissection.* It was particularly useful for certain problems arising from partial differential equations.

Sobieski [78] developed a procedure for calculating the sensitivity of a multidisciplinary system with respect to design variables and called it the global sensitivity equation (GES). Each subsystem may represent a different disciplinary analysis of the same system. This approach required the derivatives of the individual disciplinary responses with respect to the input of all other disciplines. Sobieski pointed out that the cost of these calculations can be very large when the front of the interaction between disciplines is very large. This can be the case for the aerodynamic sensitivity analysis when the number of grid points in the interfaces is large.

The conclusion that can be drawn from Sections [5.2.2] and [5.2.3] is, that there are very few investigations in the area of aerodynamic sensitivity analysis, and they all are at their very early stages. Hence, there appears to be an urgent need for further and more comprehensive investigations, in particular, the inclusion of viscous effects, shape design, and domain decomposition. The present study should provide a clearer path to the efforts in two subjects, namely, aerodynamic sensitivity analysis and shape design optimization.

5.3 Present work

The second part of the dissertation covers the development of a new, efficient, and accurate numerical optimization procedure, which automatically shapes the aerodynamic configuration to any arbitrary geometry, and which does not require any experience other than that needed for formulating the optimization problem. The utilized flowfield governing equations in this study are the Euler equations and the thin-layer Navier-Stokes equations.

The following steps are taken to achieve the above objectives:

(1) Investigating several methods for calculating the aerodynamic sensitivity coefficients, which are based on the discretized forms of the flowfield governing equations. Besides obtaining accurate values of these coefficients, a major concern is the computational feasibility of these methods. By far, the dominant consumer of the computational time is solving a set of linear algebraic equations resulting from the sensitivity equation. Therefore, different methods for solving this set of equations are incorporated and assessed from their computational efficienies point of view.

(2) Developing a domain-decomposition technique to divide the large aerodynamic optimization problems into smaller ones and to ease the generation of the computational grid for complex-geometry problems.

(3) Developing an accurate flow prediction method that is capable of predicting all physical phenomena, which may occur in the flowfield, without performing a complete CFD analysis.

(4) Demonstrating the capability of the new procedure by optimizing the shape of the scramjet-nozzle configuration of Fig. (2.2), and by optimizing the shape of a transonic airfoil.

The organization of this part proceeds as follows. Chapter 6 gives a description of the flow analysis algorithm used in the present optimization procedure. Chapter 7 provides some basic definitions of constrained optimization problems deemed necessary to the present development. Also, it includes the formulation of two example optimization problems. Details of flowfield prediction method, the methods for calculating the aerodynamic sensitivity coefficients, and the Sensitivity Analysis Domain Decomposition (SADD) scheme are given in Chapter 8. Given in Chapter 9 are the modifications of an optimization algorithm and the description of the overall shape optimization procedure. Results of the demonstrative examples are discussed in Chapter 10. Finally, the conclusions and recommendations are given in Chapter 11.

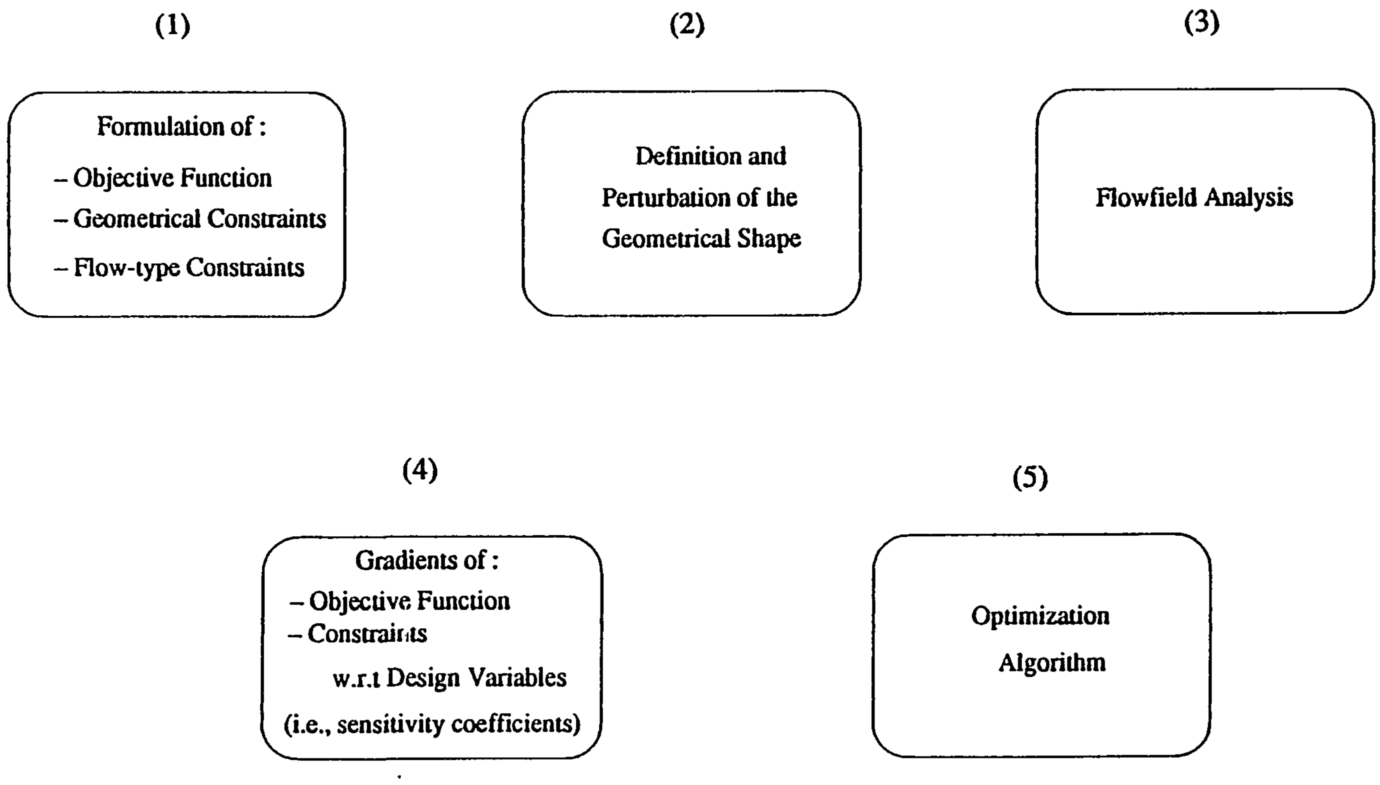


Fig. 5.1 Components of a typical aerodynamic design optimization problem.

Chapter 6

FLOW ANALYSIS FORMULATION

One of the key elements in the design optimization procedure is the flowfield analysis. A flowfield solution can be obtained either via CFD algorithm or via the flowfield prediction method described in Chapter 8. However, the development of the flowfield prediction method as well as the discrete sensitivity analysis methodology is dependent on the CFD algorithm used to analyze the flowfield. Moreover, as it will become obvious, the design optimization process has relatively high computational cost. Hence, the design procedure will be demonstrated only in two dimensions and using two sets of flow equations, namely, Euler equations and thin-layer Navier-Stokes equations. The flow is assumed laminar and the fluid is air, which is assumed to be a single species gas. The extension to two-dimensional, multispecies flows is straightforward, but computations are certainly more costly.

6.1 Governing Equations

Based on the reasons stated above, the governing equations given in Chapter 3 are reduced for the flow of air by neglecting the species continuity equations and the terms representing the species mixing. In addition, the air is assumed to be a thermally and calorically perfect gas. Under these assumptions as well as the thin-layer approximation the governing equations [Eqs. (3.1)–(3.6)] will take the following form:

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} - \frac{\partial \hat{G}_v}{\partial \eta} = 0 \tag{6.1}$$

where

$$\widehat{Q}=\frac{Q}{J}=\frac{1}{J}\left\{\begin{matrix}\rho\\ \rho u\\ \rho v\\ e\end{matrix}\right\} \qquad \widehat{F}=\frac{1}{J}\left\{\begin{matrix}\rho U\\ \rho U u+\xi_x P\\ \rho U v+\xi_y P\\ (e+P)\,U\end{matrix}\right\} \qquad \widehat{G}=\frac{1}{J}\left\{\begin{matrix}\rho V\\ \rho V u+\eta_x P\\ \rho V v+\eta_y P\\ (e+P)\,V\end{matrix}\right\}$$

$$\widehat{G}_v=\frac{1}{J}\left\{\begin{matrix}0\\ \eta_x\,\tau_{xx}+\eta_y\,\tau_{xy}\\ \eta_x\,\tau_{xy}+\eta_y\,\tau_{yy}\\ \eta_x\,b_x+\eta_y\,b_y\end{matrix}\right\} \tag{6.2}$$

Under the thin-layer assumption, all the derivatives with respect to ξ in both the shear stress, τ, and the heat flux, $\dot{q}$, terms are set to zero. Consequently, Eqs. (3.4)-(3.6) for the case of non-mixing flows become

$$\tau_{xx}=M_\infty Re_L^{-1}\left\{(\lambda+\mu)\;u_\eta\eta_x+\lambda\;v_\eta\eta_y\right\} \tag{6.3a}$$

$$\tau_{yy}=M_\infty Re_L^{-1}\left\{(\lambda+\mu)\;v_\eta\eta_y+\lambda\;u_\eta\eta_x\right\} \tag{6.3b}$$

$$\tau_{xy}=M_\infty Re_L^{-1}\,\mu\;(u_\eta\eta_y+v_\eta\eta_x) \tag{6.3c}$$

$$\dot{q}_x=\left(\frac{-M_\infty\,\mu}{Re_L\,Pr\,(\gamma-1)}\right)\eta_x\,(a^2)_\eta \tag{6.4a}$$

$$\dot{q}_y=\left(\frac{-M_\infty\,\mu}{Re_L\,Pr\,(\gamma-1)}\right)\eta_y\,(a^2)_\eta \tag{6.4b}$$

Hence, the terms b_x and b_y in both $\widehat{G}_v$ are given by:

$$b_x=u\;\tau_{xx}+v\;\tau_{xy}-\dot{q}_x \tag{6.5a}$$

$$b_y=u\;\tau_{xy}+v\;\tau_{yy}-\dot{q}_y \tag{6.5b}$$

The pressure is related to the conserved variables through the ideal gas law

$$P=(\gamma-1)\left[e-\frac{\rho}{2}(u^2+v^2)\right] \tag{6.6}$$

The contravariant velocities (U and V) and the transformation Jacobian, J, are defined by Eqs. (3.2)–(3.3). Stokes hypothesis for bulk viscosity, $\lambda + (2/3\mu) = 0$, is invoked, and the molecular viscosity is calculated from the Sutherlands law, Eq. (3.19), using the constants given in Appendix A.

In Eq. (6.1), the conserved variables of vector Q, Eq. (6.2), are normalized with the freestream values of density and the local speed of sound for convenience,

$$\rho = \frac{\tilde{\rho}}{\tilde{\rho}_\infty} \ ; u = \frac{\tilde{u}}{\tilde{a}_\infty} ; v = \frac{\tilde{v}}{\tilde{a}_\infty} \ ; e = \frac{\tilde{e}}{\tilde{a}_\infty^2 \tilde{\rho}_\infty} \tag{6.7}$$

Also used in the non-dimensionalization are the reference length L and the freestream molecular viscosity, μ_∞.

6.2 Finite-Volume Formulation

Unlike in Part I, Eq. (6.1) is solved using an implicit, finite-volume, upwind, spatially factored algorithm [72-73]. The finite-volume differencing in this algorithm is formulated by integrating the flowfield governing equation, Eq. (6.1), over a a stationary control volume (area in two-dimensional flows),

$$\frac{\partial}{\partial t} \iiint_{V_c} Q \, dV_c + \iint_{A} \vec{F} \cdot \vec{n} \, dA = 0 \tag{6.8}$$

where $\vec{n}$ is the unit normal vector pointing outward from the surface, A, bounding the control volume, V_c, and $\vec{F}$ is a flux vector given for the assumed two-dimensional flow by:

$$\vec{F} = (F) \ \vec{i} + (G - G_v) \vec{j} \tag{6.9}$$

where (F, G, G_v) are the inviscid and viscous fluxes in Cartesian coordinates and $\left(\vec{i}, \vec{j} \right)$ are the unit vectors in Cartesian coordinates. Equation (6.8) is applied to each cell of the computational domain (ξ, η) and the following semi-discrete, finite-volume representation is obtained:

$$\frac{1}{J}\left(\frac{\partial Q}{\partial t}\right)_{i,j} + \frac{\left(\hat{F}\right)_{i+1/2,j} - \left(\hat{F}\right)_{i-1/2,j}}{\Delta\xi} + \frac{\left(\hat{G} - \hat{G}_V\right)_{i,j+1/2} - \left(\hat{G} - \hat{G}_V\right)_{i,j-1/2}}{\Delta\eta} = 0 \qquad (6.10)$$

where $\Delta\xi = \Delta\eta = 1$. It should be noted that $Q_{i,j}$ represents the conserved variables at the cell center (cell averaged) and the fluxes are evaluated at the cell surfaces.

6.3 Flow Analysis Algorithm

The governing equations can be treated as time-dependant equations, which are hyperbolic. Hence, the steady-state solution is obtained by marching in pseudo-time until the convergence is achieved.

Applying the implicit Euler time integration to Eq. (6.1) and then linearizing the fluxes yields the following form:

$$\left[\frac{I}{J\,\Delta t} + \delta_\xi \frac{\partial\left(\hat{F}\right)}{\partial Q} + \delta_\eta \frac{\partial\left(\hat{G} - \hat{G}_v\right)}{\partial Q}\right]^n \Delta Q = -R\left(Q^n\right) \qquad (6.11)$$

where $\left[\partial\left(\hat{F}\right)/\partial Q\right]$, $\left[\partial\left(\hat{G} - \hat{G}_v\right)/\partial Q\right]$ denotes the flux Jacobian matrices and

$$\Delta Q = Q^{n+1} - Q^n \qquad (6.12)$$

The steady-state residual, $R(Q^n)$, is comprised of the spatial derivative terms evaluated at time level n and are given by

$$R\left(Q^n\right) = \left[\delta_\xi\left(\hat{F}\right) + \delta_\eta\left(\hat{G} - \hat{G}_v\right)\right]^n \qquad (6.13)$$

Equation (6.11) represents a large banded system of linear equations in which ΔQ is the unknown. The bandwidth of its coefficient matrix depends on the particular choice of spatial discretization and on the grid sizes. It is noteworthy that the coefficient matrix of Eq.(6.11), in the limit as Δt approaches infinity, is very similar to the one which is used in the sensitivity analysis approach. This point is further explained in Chapter 8.

To obtain the steady-state solution of Eq. (6.11), the residual given by Eq. (6.13) is forced to zero. One way to achieve this is to set Δt to infinity and then solve the resulting equation using Newton's method. Usually, this approach exhibits a quadratic convergence. However, if the initial guess of the solution is not close to the final solution, oscillations may occur in the solution and, as a result, cause a program failure. To circumvent this problem, Eq. (6.11) can be iteratively solved by a direct method [79]. In this procedure as Δt approaches infinity, Eq. (6.11) becomes the equation of Newton's method. This procedure is commonly known as the modified Newton's method. In problems where the initial solution is very far from the steady-state solution, this procedure requires a large number of iterations to realize quadratic convergence. Hence, this procedure becomes computationally expensive when it is employed to obtain the initial steady-state solutions of the optimization problems.

Another way to obtain the steady-state solution is to factor the the left side of Eq. (6.11) into a sequence of one-dimensional operators [80]. Then, the resulting equation is solved in a series of sweeps throughout the computational mesh. This approach is called the approximate factorization method; and when it is applied to Eq. (6.11), it yields the form given below.

$$\left[\frac{I}{J\,\Delta t}+\delta_\xi\frac{\partial\left(\hat{F}\right)}{\partial Q}\right]^n\left[\frac{I}{J\,\Delta t}\right]^{-1}\left[\frac{I}{J\,\Delta t}+\delta_\eta\frac{\partial\left(\hat{G}-\hat{G}_v\right)}{\partial Q}\right]^n\Delta Q=-R\left(Q^n\right) \qquad (6.14)$$

or

$$\left[\frac{I}{J\,\Delta t} + \delta_\xi \frac{\partial(\hat{F})}{\partial Q}\right]^n \Delta Q^{\overline{n+1}} = -R(Q^n)$$

$$\left[\frac{I}{J\,\Delta t} + \delta_\eta \frac{\partial(\hat{G} - \hat{G}_v)}{\partial Q}\right]^n \Delta Q = \left(\frac{I}{J\,\Delta t}\right)\Delta Q^{\overline{n+1}} \tag{6.15}$$

$$Q^{n+1} = Q^n + \Delta Q$$

In this form, the solution is easily obtained by solving the two one-dimensional problems (Eq. (6.15)). In each problem, the inversion of only a block tri- or penta-diagonal matrix is required. Since this is not expensive, this approach is much cheaper than the modified Newton's method. Hence, this procedure is used in this study to obtain the initial steady-state solutions of the optimization problems.

For a steady-state solution, each cell in the computational domain is advanced at its own time step according to the following equation:

$$\Delta t = \text{CFL}\left\{|U| + |V| + a\left[|grad(\xi)| + |grad(\eta)|\right]\right\}^{-1} \tag{6.16}$$

where CFL is the Courant number which may vary from two to five.

6.4 The Inviscid Discrete Form of the Residual

6.4.1 Flux-Vector Splitting

In the present study, the pressure and convective terms are upwind differenced by the flux-vector splitting method of Van Leer [81]. The inviscid fluxes are spatially differenced by backward or forward differencing, depending on the characteristics of the flow at each grid cell. Upwind methods are naturally dissipative and require no artificial dissipation terms to reduce high gradients in the flowfield.

In flux-splitting methods [82], the spatial derivatives of the inviscid fluxes are written conservatively as flux balances across the cell. For example in the ξ-direction:

$$\left(\frac{\partial \widehat{F}}{\partial \xi}\right)_i = \widehat{F}_{i+1/2} - \widehat{F}_{i-1/2} \tag{6.17}$$

where the interface flux is constructed as

$$\widehat{F}_{\mathrm{i+1/2}} = \widehat{F}\,(Q^-, Q^+)_{\mathrm{i+1/2}} \tag{6.18}$$

The conserved state variables on cell interfaces, $Q^{\pm}_{i\pm 1/2}$, are determined from an upwind-biased interpolation of the primitive variables,

$$\begin{aligned} Q^-_{\mathrm{i+1/2}} &= Q_{\mathrm{i}} + \left\{\frac{\phi}{4}\left[(1-\kappa)\,\nabla + (1+\kappa)\,\Delta\right]\right\}_{\mathrm{i}} \\ Q^+_{\mathrm{i+1/2}} &= Q_{\mathrm{i+1}} - \left\{\frac{\phi}{4}\left[(1+\kappa)\,\nabla + (1-\kappa)\,\Delta\right]\right\}_{\mathrm{i+1}} \end{aligned} \tag{6.19}$$

where

$$\begin{aligned} \Delta_{\mathrm{i}} &= Q_{\mathrm{i+1}} - Q_{\mathrm{i}} \\ \nabla_{\mathrm{i}} &= Q_{\mathrm{i}} - Q_{\mathrm{i-1}} \end{aligned} \tag{6.20}$$

The type of differencing is determined by assigning different values to κ.

The Van Leer method splits the inviscid flux terms, $\widehat{F}$ and $\widehat{G}$ into two vectors as a forward flux and a backward flux, providing that the Jacobian matrices $\partial \widehat{F}^+/\partial Q$ have non-negative eigenvalues and $\partial \widehat{F}^-/\partial Q$ have non-positive eigenvalues.

For a supersonic flow ($|M_\xi| > 1$),

$$\begin{aligned} &F^+ = F \qquad F^- = 0 \qquad \text{for} \quad M_\xi \geq 1 \\ &F^+ = 0 \qquad F^- = F \qquad \text{for} \quad M_\xi < 1 \end{aligned} \tag{6.21}$$

where M_ξ is the contravariant Mach number in the ξ-direction.

$$M_\xi = \frac{\bar{u}}{a}$$

$$\bar{u} = \frac{U}{|grad\,(\xi)|} \tag{6.22}$$

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

$$\hat{F}^{\pm} = \frac{|grad\,(\xi)|}{J} \begin{pmatrix} f^{\pm}_{\text{mass}} \\ f^{\pm}_{\text{mass}} \left\{ \left[\hat{\xi}_x \left(-\bar{u} \pm 2a \right) / \gamma \right] + u \right\} \\ f^{\pm}_{\text{mass}} \left\{ \left[\hat{\xi}_y \left(-\bar{u} \pm 2a \right) / \gamma \right] + v \right\} \\ f^{\pm}_{\text{energy}} \end{pmatrix} \tag{6.23}$$

where

$$f^{\pm}_{\text{mass}} = \pm \rho a \left(M_\xi \pm 1 \right)^2 / 4 \tag{6.24}$$

$$f^{\pm}_{\text{energy}} = \pm f^{\pm}_{\text{mass}} \left[\frac{(1-\gamma)\bar{u}^2 \pm 2\,(\gamma-1)\,\bar{u}\,a + 2a^2}{\left(\gamma^2 - 1\right)} + \frac{\left(u^2 + v^2\right)}{2} \right] \tag{6.25}$$

and $\hat{\xi}_x$,$\hat{\xi}_y$ are the directional cosines.

The differences of split flux vectors are implemented as a balances across a cell. The inviscid flux balance in the ξ-direction, Eq. (6.17), for an interior cell *(i, j)* is given, for example, by:

$$\delta^-_\xi \hat{F}^+ + \delta^+_\xi \hat{F}^- = \left[\hat{F}^+ \left(Q^-_{i+1/2,j} \right) + \hat{F}^- \left(Q^+_{i+1/2,j} \right) \right] - \left[\hat{F}^+ \left(Q^-_{i-1/2,j} \right) + \hat{F}^- \left(Q^+_{i-1/2,j} \right) \right] \tag{6.26}$$

A similar expression can be obtained for the inviscid flux balance in η-direction. Consequently, the discrete form of the inviscid, steady-state residual contribution in, Eq.(6.13), is written as,

$$\textit{The inviscid part of Eq.}\,(6.13) = \left[\widehat{F}^{+}\left(Q^{-}_{i+1/2,j}\right) + \widehat{F}^{-}\left(Q^{+}_{i+1/2,j}\right)\right] - \left[\widehat{F}^{+}\left(Q^{-}_{i-1/2,j}\right) + \widehat{F}^{-}\left(Q^{+}_{i-1/2,j}\right)\right]$$
$$+ \left[\widehat{G}^{+}\left(Q^{-}_{i,j+1/2}\right) + \widehat{G}^{-}\left(Q^{+}_{i,j+1/2}\right)\right] - \left[\widehat{G}^{+}\left(Q^{-}_{i,j-1/2}\right) + \widehat{G}^{-}\left(Q^{+}_{i,j-1/2}\right)\right] \tag{6.27}$$

6.4.2 Flux Limiter

Numerical oscillations such as overshoots and undershoots are expected to appear in the presence of large flow gradients when an upwind-biased scheme is used. A flux limiter can be used to reduce the upwind-biased scheme to a fully one-sided upwind scheme in these regions. This in turn ensures a monotonic interpolation and eliminates the overshoots and the undershoots.

The Van Albada limiter [83] is used in the present study and is implemented by rewriting the Eq. (6.19) as

$$\begin{aligned} Q^{-}_{\mathrm{i}+1/2} &= Q_{\mathrm{i}} + \phi\left\{\frac{S}{4}\left[(1-\kappa S)\,\nabla + (1+\kappa S)\,\Delta\right]\right\}_{\mathrm{i}} \\ Q^{+}_{\mathrm{i}+1/2} &= Q_{\mathrm{i}+1} - \phi\left\{\frac{S}{4}\left[(1+\kappa S)\,\nabla + (1-\kappa S)\,\Delta\right]\right\}_{\mathrm{i}+1} \end{aligned} \tag{6.28}$$

where

$$S = \frac{2\nabla\Delta + \varepsilon}{\Delta^2 + \nabla^2 + \varepsilon} \tag{6.29}$$

and ε is a small number to prevent the division by zero in the regions of zero gradient. The quantity S is the limiter control function. Its value reaches unity at the regions of small gradients, thus using no limiter for these regions. However, its value goes to zero for very large gradients regions, hence reducing to a first-order interpolation. This limiter acts in a continuously differentiable fashion, and it has the advantage of overcoming the convergence problem usually encountered with the minimum-modulus(min-mod) type of limiter [82].

6.5 The Viscous Discrete Form of the Residual

In the finite-volume formulation, the spatial derivative of the viscous flux $\widehat{G}_v$ is treated as a flux balance across the cell interfaces. That is,

$$\left(\frac{\partial \widehat{G}_v}{\partial \eta}\right)_{i,j} = \left(\widehat{G}_v\right)_{i,j+1/2} - \left(\widehat{G}_v\right)_{i,j-1/2} \tag{6.30}$$

where $\left(\widehat{G}_v\right)_{i,j+1/2}$ is given by

$$\left(\widehat{G}_v\right)_{i,j+1/2} = \frac{M_\infty}{Re_L}\left(\frac{\mu}{J}\right)_{i,j+1/2}\begin{pmatrix}\left(G_v^1\right)_{i,j+1/2}\\ \left(G_v^2\right)_{i,j+1/2}\\ \left(G_v^3\right)_{i,j+1/2}\\ \left(\widehat{G}_v^4\right)_{i,j+1/2}\end{pmatrix} \tag{6.31}$$

The derivative terms in $\left(\widehat{G}_v\right)_{i,j+1/2}$ are one-side differenced to maintain a second-order approximation. Hence, the terms in $\left(\widehat{G}_v\right)_{i,j+1/2}$ take the following form:

$$\left(G_v^1\right)_{i,j+1/2} = 0 \tag{6.32a}$$

$$\left(G_v^2\right)_{i,j+1/2} = \left|\nabla\eta\right|^2_{i,j+1/2} \cdot \left(u_{i,j+1} - u_{i,j}\right) + \frac{1}{3}\,\eta_{x\ i,j+1/2}\left[\nabla\eta_{i,j+1/2} \cdot \left(\vec{V}_{i,j+1} - \vec{V}_{i,j}\right)\right] \tag{6.32b}$$

$$\left(G_v^3\right)_{i,j+1/2} = \left|\nabla\eta\right|^2_{i,j+1/2} \cdot \left(v_{i,j+1} - v_{i,j}\right) + \frac{1}{3}\,\eta_{y\ i,j+1/2}\left[\nabla\eta_{i,j+1/2} \cdot \left(\vec{V}_{i,j+1} - \vec{V}_{i,j}\right)\right] \tag{6.32c}$$

$$\begin{aligned}\left(G_v^4\right)_{i,j+1/2} = {} & \left|\nabla\eta\right|^2_{i,j+1/2} \cdot \left\{\frac{1}{2}\left(\left|\nabla\vec{V}\right|_{i,j+1} - \left|\nabla\vec{V}\right|_{i,j}\right) + \frac{1}{(\gamma-1)Pr}\left(a^2_{i,j+1} - a^2_{i,j}\right)\right\} \\ & + \frac{1}{3}\,V_{i,j+1/2}\left[\nabla\eta_{i,j+1/2} \cdot \left(\vec{V}_{i,j+1} - \vec{V}_{i,j}\right)\right]\end{aligned} \tag{6.32d}$$

where u and v are the Cartesian components of the velocity $\vec{V}$ and V denotes the contravariant component given by Eq. (3.3b). As seen from Eq. (6.32), $\left(\widehat{G}_v\right)_{i,j+1/2}$ can be viewed as a function of the cell-centered values of the conserved flow variables at points (i,j) and $(i,j+1)$. Similar argument can be established for $\left(\widehat{G}_v\right)_{i,j-1/2}$. Hence, by combining Eqs. (6.27) and (6.30), the upwind discrete form of the steady-state residual, Eq. (6.13), at an interior cell *(i,j)* becomes,

$$\begin{aligned} R_{i,j}\ (Q^n) = & \left[\widehat{F}^+\left(Q^-_{i+1/2,j}\right)+\widehat{F}^-\left(Q^+_{i+1/2,j}\right)\right]-\left[\widehat{F}^+\left(Q^-_{i-1/2,j}\right)+\widehat{F}^-\left(Q^+_{i-1/2,j}\right)\right] \\ & +\left[\widehat{G}^+\left(Q^-_{i,j+1/2}\right)+\widehat{G}^-\left(Q^+_{i,j+1/2}\right)\right]-\left[\widehat{G}^+\left(Q^-_{i,j-1/2}\right)+\widehat{G}^-\left(Q^+_{i,j-1/2}\right)\right] \\ & -\left[\widehat{G}_v\ (Q_{i,j+1},\, Q_{i,j})-\widehat{G}_v\,(Q_{i,j},\, Q_{i,j-1})\right] \end{aligned} \qquad (6.33)$$

It is clear from Eq. (6.33) that this discrete form is a function of the cell-face values of the conserved variables $Q^{\pm}_{i\pm1/2,j\pm1/2}$ and the cell-centered values of the conserved variables at the points (i,j) and $(i,j\pm1)$, i.e., $Q_{i,j}$, and $Q_{i,j\pm1}$. This form is forced to zero at all computational cells in order to obtain the steady-state solution.

6.6 Initial and Boundary Conditions

The initial conditions are chosen to be the freestream conditions and the boundary conditions are specified explicitly. For the viscous flow calculations, the walls are considered to be impermeable and adiabatic. The pressure at the solid surfaces is evaluated using a zeroth-order extrapolation from the interior point value. The density is then calculated by employing the state equation. Consequently,

$$u = 0\,,\ v = 0\,,\ \frac{\partial T}{\partial n} = 0,\ \frac{\partial P}{\partial n} = 0 \qquad (6.34)$$

For the inviscid flow calculations, the flow tangency condition is imposed by setting the the contravariant component of the total velocity normal to the wall to zero. In addition, the fluid density and pressure at the surface are extrapolated from the interior. This means that at the inviscid wall, the following conditions are applied:

$$V = 0 , \frac{\partial \rho}{\partial n} = 0, \frac{\partial P}{\partial n} = 0 \tag{6.35}$$

For both inviscid and viscous flow calculations, the locally one-dimensional characteristic boundary conditions are employed for the farfield boundaries. For each cell, the velocity normal to the boundary and the speed of sound are calculated from the two locally one-dimensional Riemann invariants given by

$$R^{\pm} = U \pm \frac{2}{\gamma - 1} a \tag{6.36}$$

The invariants are constant along the characteristics defined by

$$\left(\frac{d\xi}{dt}\right)^{\pm} = U \pm a \tag{6.37}$$

The local normal velocity at the boundary is calculated by summing the Riemann invariants and the speed of sound by subtracting the two. The appropriate boundary conditions are specified after the magnitude of the contravariant Mach number (*U*/*a*) and the direction of the contravariant velocity at each cell are checked.

$$|M_{\xi}| > 1 \; ; U < 0 \qquad u = u_{\infty} , v = v_{\infty} , T = T_{\infty} , P = P_{\infty} \tag{6.38}$$

$$|M_{\xi}| > 1 \; ; U > 0 \qquad \frac{\partial u}{\partial \xi} = 0 , \frac{\partial v}{\partial \xi} = 0 , \frac{\partial T}{\partial \xi} = 0 , \frac{\partial P}{\partial \xi} = 0 \tag{6.39}$$

where ξ is the streamwise coordinate.

$$|M_{\xi}| < 1 \; ; U < 0 \qquad u = u_{\infty} , v = v_{\infty} , T = T_{\infty} , \frac{\partial P}{\partial \xi} = 0 \tag{6.40}$$

$$|M_\xi| < 1 \,;\, U > 0 \qquad \frac{\partial u}{\partial \xi} = 0 \,,\, \frac{\partial v}{\partial \xi} = 0 \,,\, \frac{\partial T}{\partial \xi} = 0 \,,\, \rho = \rho_\infty \qquad (6.41)$$

For supersonic inflow, all flow characteristics point from the outside toward the inside of the computational domain. Hence, the inflow boundaries are set equal to the free stream conditions (Eq. (6.38)).

In the case of supersonic outflow, the flow signals propagate from the inside toward the outside. Therefore, the flow variables at the outflow boundaries are computed from the interior flow solution by a zeroth-order extrapolation (Eq. (6.39)).

Finally, due to the decomposition of the computational domain into smaller subdomains, additional boundaries are introduced at the block interfaces. The solutions are matched at two sets of grid points on each side in the normal direction to an interface. This provides second-order spatial accuracy consistent with the accuracy of the solution algorithm.

Chapter 7

BASIC OPTIMIZATION DEFINITIONS

Aerodynamic optimization may be achieved by utilizing one of the constrained optimization methods. Usually, the formulation of aerodynamic optimization problems is crucial for the success of the optimization process and is case dependent. However, there are general guidelines for this formulation that will lead to an efficient optimization process. Section 7.1 focuses on the discussion of this issue along with some definitions deemed necessary for the present method. This is followed by the standard formulation of the nonlinear constrained optimization problem. Finally, the standard formulation is applied to two example problems to be used later for the demonstration of the present method.

7.1 Elements of Constrained Optimization Problems

7.1.1 Design Variables

A typical constrained minimization or maximization problem entails a group of quantities called *design variables*. These variables usually vary during the optimization process in a space within which the search for the optimum takes place (i.e., design space). In aerodynamic applications, the design variables are generally of a geometric-type; for example, the coordinates of the surface describing the aerodynamic body are sometimes considered as design variables. Mathematically, the design variables are denoted by a vector $\overline{X}_D$ given by

$$\overline{X}_D = \{X_{D1}, X_{D\,2}, \ldots\ldots\ldots\ldots, X_{D\;NDV}\} \tag{7.1}$$

where NDV is the number of design variables.

For aerodynamic shape optimization, different forms of design variables can be used; however, the choice of the proper form depends on some criteria. These criteria are: (1) design variables must be selected to converge the optimization process quickly; (2) they should be able to generate a wide variety of geometries; and (3) they should be consistent with the flow analysis model. To clarify these criteria, the common ways of describing the aerodynamic shapes are considered. The shapes of aerodynamic configurations are commonly described by two classes of functions [84]. The first class is the analytical functions such as the Legendre polynomial, Wagner functions, and Hicks-Henne functions. For this class of functions, the design variables usually are the coefficients of these functions.

The second class of functions is those shape functions that have aerodynamic origin, such as, the aerofunctions [84] and the airfoil library, which contains a combination of basic airfoils. The design variables in this case commonly are the weighting coefficients of the added shape functions. Unfortunately, using such functions in describing the aerodynamic shapes do not neccessarly generate arbitrary geometries. Moreover, some experience and skill are required to select the best one that can be used to describe the shape. Hence, it may be attractive to use the grid point coordinates of the aerodynamic shape as design variables. However, to ensure the smoothness of the shape during the optimization process, this usually requires many additional geometrical constraints, and consequently increases the computational cost. Otherwise, the optimization process may lead to nonsmooth shapes, which are not adequate for the required flowfield analyses. Therefore, the consistency between the design variables, which control the configuration shape, and the analysis model has to exist.

In the present study, to reduce the amount of expertise required for choosing the shape functions and to alleviate the problem of smoothness of the shape, the aerodynamic shape is defined by the surface grid points. Equivalently, the relative slopes between these grid points may be considered as the design variables (Fig. 7.1). Hence, the optimizer will shape automatically the aerodynamic configuration to any arbitrary geometry during the optimization process. Although this approach of shape description results in a large number of design variables, it has the advantages of being flexible, general and easy to implement. Moreover, this approach eliminates the closure problem which is usually encountered in airfoil design problems.

7.1.2 Objective Function

A typical constrained optimization problem also entails a function which the optimizer will drive to the lowest or highest possible value, subject to stated constraints. This function is the *objective function.* For aerodynamic applications, the objective function depends not only on the design variables, but also on the flowfield solution which is an implicit function of the design variables. This is mathematically translated into $F\left(Q(\overline{X}_D), \overline{X}_D\right)$.

Unlike the inverse design optimization methods, the present direct numerical optimization method does not prescribe a priori some target aerodynamic characteristics, which need to be met at the optimum design, e.g., the surface pressure distribution which yields high lift coefficient. Hence, the present optimization method has the advantage of not requiring the significant level of skills and experience needed for the choice of the target aerodynamic characteristics, and consequently the objective function. However, if a function is selected as an objective function but does not have a minimum or maximum under the specified constraints, the optimization process will never converge. Hence, a judicious choice of the objective function is crucial for the success of the optimization process. For example, the objective function in optimizing an airfoil shape can simply be the drag coefficient, the lift coefficient, or a combination of these coefficients.

7.1.3 Constraints

In a typical constrained optimization problem, the constraints come in two fundamental forms; namely, side constraints and inequality constraints. Side constraints are the less restrictive constraints. Their sole purpose is to put lower and upper bounds on the design variables in order to have, for example, a smooth shape. Usually, the side constraints are enforced directly onto the design variables. Whereas, the inequality constraints, in general, cannot be enforced directly onto the design variables, but are functions of either the design variables or both design variables and the flowfield solution. Inequality constraints serve to limit how the design can be changed, or to complete the objective function. For instance, the thicknesses at several prescribed locations of an airfoil can be considered as inequality constraints. Since they do not depend on the flowfield solution, they are referred to here as *goemetric-type inequality constraints*. Another example of the inequality constraints is the range of the lift coefficient for which it is required to have a reduction in the drag coefficient. This type of inequality constraint, which usually depends implicitly on the flow solution, is referred to here as *flow-type inequality constraint*. It is important to notice that the side constraints are degenerate geometric-type inequality constraints, since each side constraint can be written as two geometric-type inequality constraints.

Since the side constraints are enforced directly onto the design variables, generally they cut a block out of the design space. This block is usually divided by the inequality constraints into two domains, namely, a *feasible domain* and an *infeasible domain*. The feasible domain contains all possible design points that satisfy all the inequality constraints. Whereas, the infeasible domain contains all the points that violate at least one of the constraints. Hence, the inequality constraints are analogous to fences separating areas in the design space which allow permissible designs from those areas where the constraints are violated. A constraint is referred to as satisfied if its equation is true. An inequality constraint is considered *active* if its value is very near or equal to zero and affects the optimization process. Moreover, for an optimum to be found, all constraints must be active or at least satisfied. A general, two-design-variable, constrained optimization space is shown in Fig. 7.2.

7.2 Standard Formulation for Optimization Problems

In aerodynamic applications, the above constrained optimization problem is usually expressed in mathematical terms as finding the vector $\overline{X}_D$, such that

$$\text{Minimize} \quad F\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) \tag{7.2}$$

subject to the following constraints:

(1) flow-type inequality constraints

$$G_j\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) \leq 0\,, j=1, NCON_f \tag{7.3}$$

(2) geometric-type inequality constraints

$$G_j\left(\overline{X}_D\right) \leq 0\,,\ j= NCON_f + 1, NCON \tag{7.4}$$

(3) side constraints

$$X_{D_i}^{lower} \leq X_{D_i} \leq X_{D_i}^{upper},\ i=1, NDV \tag{7.5}$$

The fact that the optimization problem is assumed to be a minimization rather than a maximization is not restrictive since instead of maximizing a function it is always possible to minimize its negative. Similarly, if there is an inequality constraint of an opposite type, i.e., greater-than-zero type, it can be transformed to a less-than-zero by multiplying it by −1.

Usually, some numerical difficulties are encountered when solving the above problem if there is a large discrepancy between the magnitudes of the different design variables and constraints. Hence, it is necessary for the success of the optimization process that all design variables and constraints are scaled. A common practice is to normalize them with a proper coefficient.

7.3 Formulation of Example Optimization Problems

The scramjet nozzle-afterbody configuration, described earlier in Chapter 2, is chosen as the first example (Fig. 2.2). This configuration allows examining the method in high speed flow regimes, since the flow of this configuration involves the expansion of a supersonic flow and mixes with a hypersonic external flow. The reasons for this particular example problem can be itemized as follows: (1) The computational flow simulations for this configuration are produced in Chapter 2-4 and in Refs. [16-19]. Therefore, the flowfield formulation needed in the optimization problem is readily available; (2) The formulation of the objective function was set up for a planar nozzle by Nickerson et al. [85]; (3) The flow-type constraints for this optimization problem are relatively well established as a result of extensive high speed research on the National Aerospace Plane [86]; (4) The cut-and-try approach was performed by Tatum and Flanagan [87]; and (5) The configuration involves not only an external flow, but also an internal flow.

As a second example, it is chosen to consider the optimization of a transonic airfoil shape (Fig. 7.3) [88]. This example helps examining the behavior of the present optimization procedure in the transonic flow regime, where more nonlinear physical phenomena exist. The computational flow simulations for this configuration are well established in the literature, e.g., Rumsey et al. [89]. Therefore, the flowfield formulation needed in the optimization problem is readily available. Moreover, this optimization problem has been studied by other researchers [90,91] using different ways of defining the airfoil shape.

7.3.1 Scramjet Nozzle-Afterbody Design

In the first demonstrative optimization problem, two cases are considered (Figs. 7.4 and 7.5). In both cases, the objective function in both cases is the axial thrust force coefficient, F, defined by

$$F\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = \frac{\widetilde{F}_{axial}}{\widetilde{F}_{inflow}} \tag{7.6}$$

where $\widetilde{F}_{axial}$ is the axial component of the thrust force due to nozzle wall shape. It is obtained by integrating numerically the pressure, P, over the ramp and cowl surfaces.

$$\widetilde{F}_{axial} = \int_k^l P_{ramp}\, dy + \int_m^n P_{cowl}\, dy \tag{7.7}$$

In Eq. (7.6), $\widetilde{F}_{inflow}$ is a normalizing force for $\widetilde{F}_{axial}$ and is associated with the inflow. Generally, this force is given by

$$\widetilde{F}_{inflow} = \int_k^c P_{th}\left(1+\gamma M_{th}^2\right) dy \tag{7.8}$$

In the case of an inflow parallel to the cowl with a constant Mach number, M_{th}, this force, i.e., $\widetilde{F}_{inflow}$, is centered at the mid-point of the line segment $\overline{kc}$, and takes the form

$$\widetilde{F}_{inflow} = P_{th}\left(1+\gamma M_{th}^2\right) H \tag{7.9}$$

This axial thrust force coefficient is subject to three flow-type inequality constraints (i.e., $NCON_f = 3$). The first constraint requires that the static pressure at the ramp tip, P_l, be forced to reach a percentage, C_l, of the freestream static pressure, P_∞, such that a maximum expansion over the ramp is reached without any reverse flow, i.e.

$$G_1\left(Q(\overline{X}_D), \overline{X}_D\right) = 1 - \frac{P_l}{C_1 P_\infty} \leq 0 \tag{7.10}$$

The other two constraints require that the static pressure at the cowl tip, P_n, should be within specified limits, (C_2 and C_3), of the freestream static pressure, P_∞, such that expansion waves emanating from the ramp initial expansion do not induce any reverse flow on either the internal or external cowl surfaces, i.e.

$$G_2\left(Q(\overline{X}_D), \overline{X}_D\right) = 1 - \frac{P_n}{C_2 P_\infty} \leq 0 \tag{7.11}$$

$$G_3\left(Q(\overline{X}_D), \overline{X}_D\right) = \frac{P_n}{C_3 P_\infty} - 1 \leq 0 \tag{7.12}$$

In addition to the flow-type inequality constraints stated above, there are some geometrical requirements on the configuration (Fig. 7.4). For example, in order to maintain the total length of the aircraft as a constant, the axial length of the ramp is fixed. Also, the thickness of the cowl is kept constant as well as the combustor exit height and the cowl length. These geometrical requirements are not included in the optimization problem.

In the first design case (Fig. 7.4), it is desired to determine the angles of the nozzle ramp, α, and of the cowl, β, that yield a maximum axial thrust force coefficient, F, (Eq. (7.6)) subject to constraints G_j's, given by Eqs. (7.10)—(7.12). Hence, there are only two design variables, $\overline{X}_D$, in these stages. The fact that the number of design variables is small eases the development of the new optimization procedure in its early stages. Side constraints (i.e., upper and lower limits) are imposed on the ramp and cowl angles.

In the second design case (Fig. 7.5), it is desired to optimize the nozzle ramp shape which yields a maximum axial thrust force coefficient, F, subject to constraints, G_j's given by Eqs. (7.10)—(7.12). The surface is described by one of its coordinates (x or y) and the relative slope at each surface grid point. This coordinate is kept fixed during the optimization process while the relative slope is modified to change the geometrical shape. Hence, the design variables, $\overline{X_D}$, are the relative slopes at the surface grid points. Side constraints (i.e., upper and lower limits) are imposed on the relative slopes of the surface grid points in order to maintain an acceptably smooth aerodynamic surface shape. Generally, a judicious choice of the upper and lower bounds for the design variables accelerates the convergence of the optimum solution. Although this way of describing the shape is inefficient due to the large number of design variables, it has the advantage of being flexible, general, and easy to implement. It should be emphasized that the purpose of this study is not to design a nozzle of common interest; rather, it demonstrates a new capability for aerodynamic shape optimization.

7.3.2. Transonic Airfoil Shape Design

In the second design optimization problem, the optimization of the NACA 0012 airfoil shape is considered. The same approach used in the previous example for defining the aerodynamic shape is applied in this design problem. That is, the contour of the airfoil to be optimized is defined by one of its coordinates (x or y) and the relative slope at each surface grid point (Fig. 7.3). This coordinate is kept fixed during the optimization process while the relative slope is modified to change the geometrical shape. Hence, the design variables, $\overline{X_D}$, are the relative slopes at the surface grid points.

Two different design cases are considered in the second design optimization problem. The first case is to optimize the upper surface of a NACA 0012 airfoil, which provides a reduced drag coefficient (C_D) while leaving the lift coefficient (C_L) greater than a specified value, $C_{L_{min}}$. The second case is to optimize the lower surface of a NACA 0012 airfoil, which provides an increased C_L while keeping C_D less than a specified value, C_{Dmax}.

Hence, in the first case, the objective function, F, is given by:

$$F\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = C_D = C_{Dinviscid} + C_{Dviscous} \tag{7.13}$$

and the lift constraint is written as:

$$G_1\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = 1 - \frac{C_L}{C_{Lmin}} \leq 0 \tag{7.14}$$

where C_{Lmin} is a lower limit of C_L. The lift coefficient,C_L, is given by:

$$C_L = C_{Linviscid} + C_{Lviscous} \tag{7.15}$$

In the second case, the objective function, F, is C_L and it is given by:

$$F\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = C_L \tag{7.16}$$

and the drag constraint is written as:

$$G_1\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = \frac{C_D}{C_{Dmax}} - 1 \leq 0 \tag{7.17}$$

where C_{Dmax} is an upper limit of C_D.

In addition to the flow-type inequality constraints stated above for the two design cases, there are some geometric-type inequality constraints which are imposed on these design cases. These constraints involve the thickness-to-chord ratio, which must be positive and not to exceed a specified value, and the thickness at the trailing edge (T.E.),which must be zero for a closed airfoil;

$$0 \leq (y_u - y_l) / c \leq t_{max} \tag{7.18}$$

To enforce the zero T.E. thickness, the initial airfoil is started with zero T.E. thickness and the T.E. grid point is simply excluded from the design variables. Instead, a geometrical constraint is imposed on the relative slope at the surface grid point *($\Delta\theta$)* adjacent to the T.E. point, i.e.,

$$\left|\left(\Delta\theta\right)_{T.E.}\right| \leq C_{max} \tag{7.19}$$

Side constraints (i.e., upper and lower limits) are imposed on the relative slopes of the surface grid points in order to maintain an acceptably smooth aerodynamic surface shape. However, it should be emphasized once again that the purpose of these demonstrative cases is not finding an optimized airfoil of common interest, rather, they serve as examples to illustrate the new of design optimization capability in the transonic flow regime.

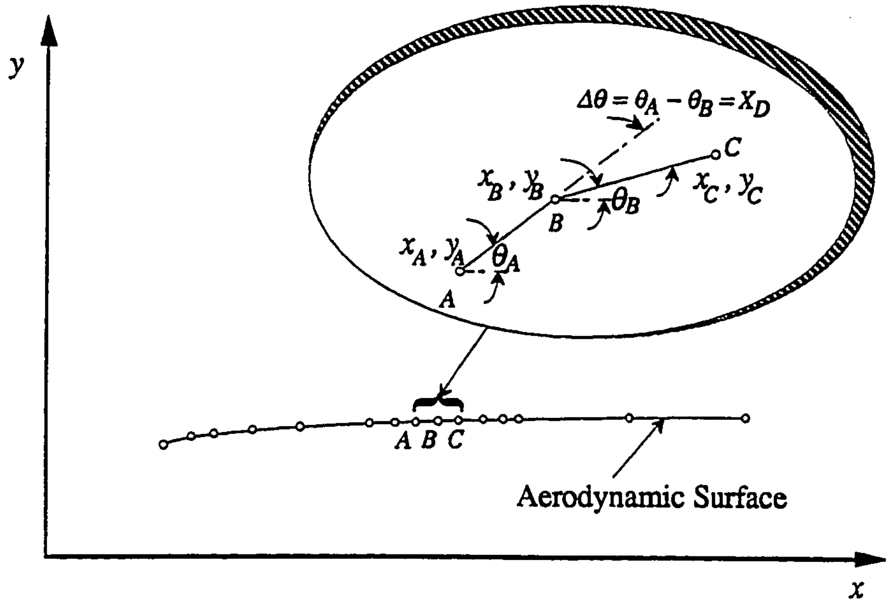


Fig. 7.1 Aerodynamic shape definition using grid point coordiates and the relative slopes as design variables.

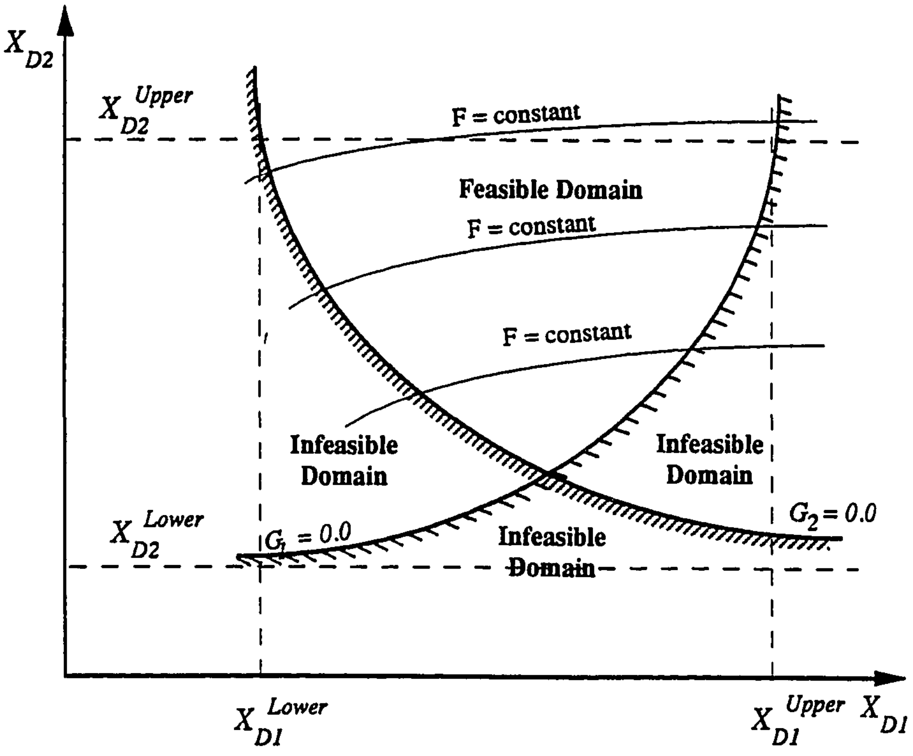


Fig. 7.2 A general constrained optimization space with two design variables.

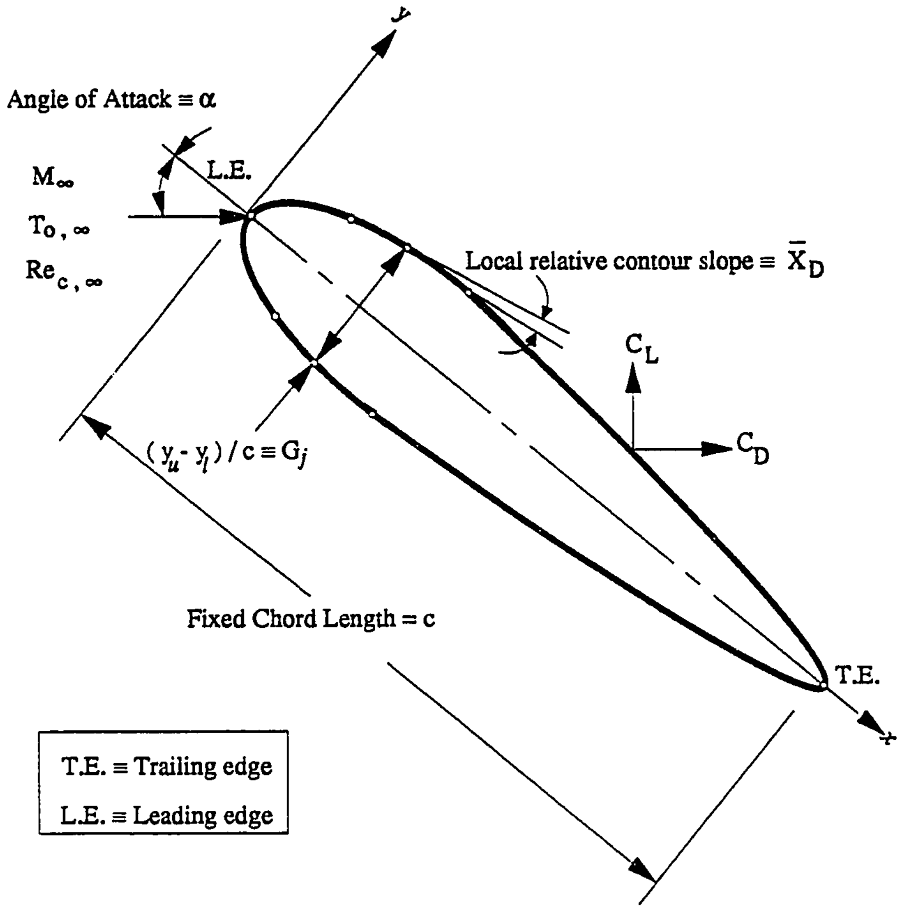


Fig. 7.3 Definition of the airfoil optimization problem.

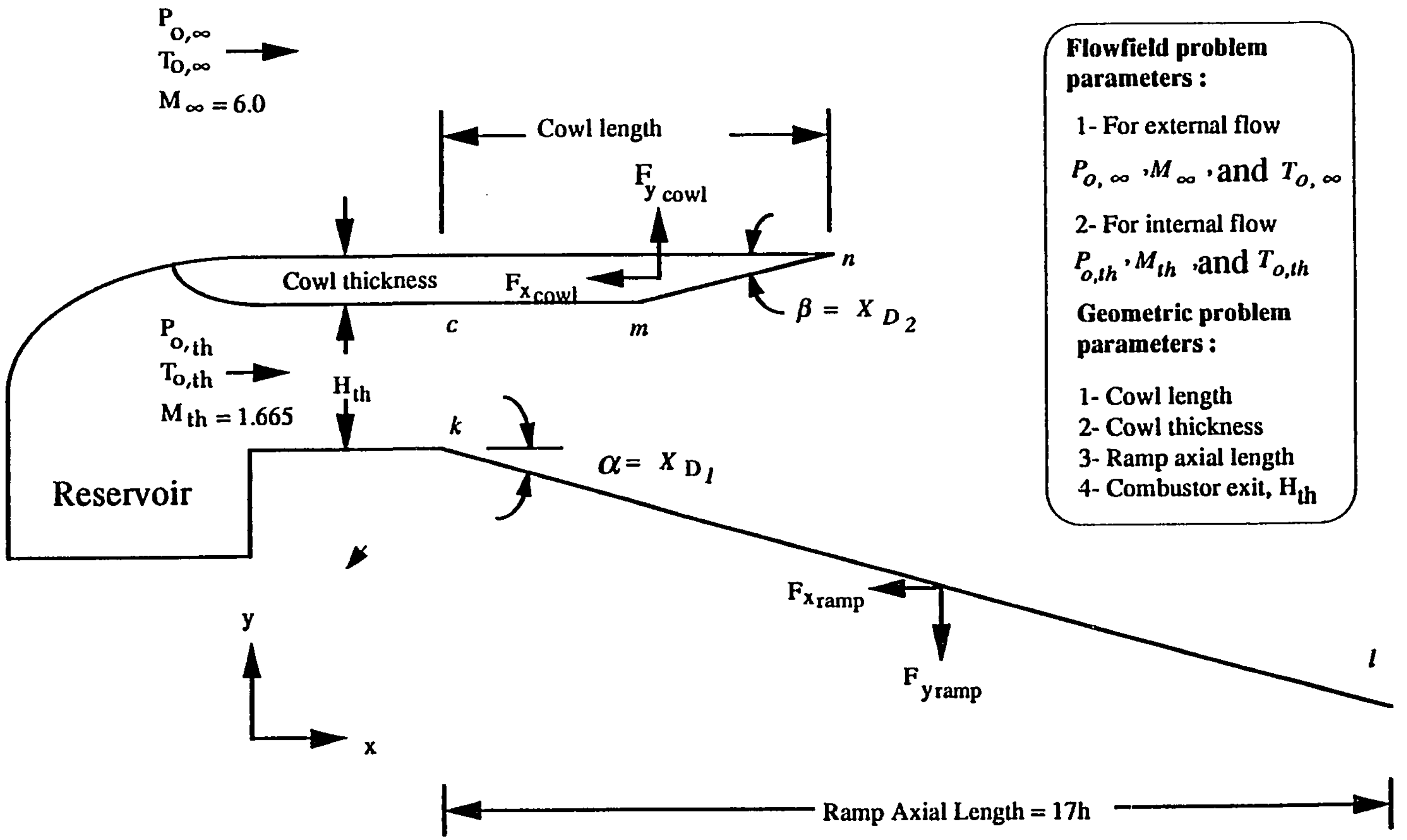


Fig. 7.4 A geometric interpretation of the planar nozzle-afterbody optimization problem.

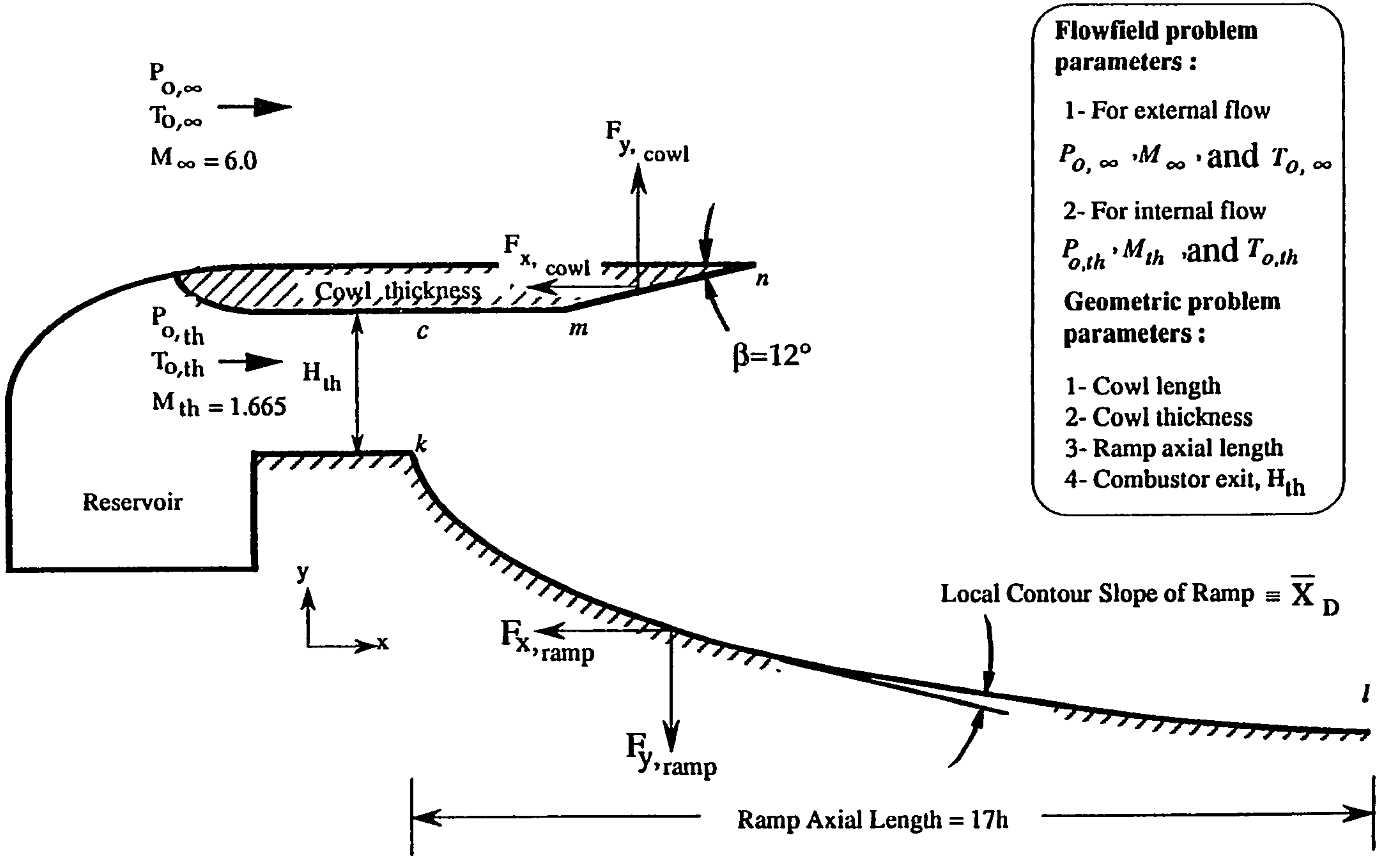


Fig. 7.5 A geometric interpretation of the nozzle-afterbody shape optimization problem.

Chapter 8

SENSITIVITY ANALYSIS METHODOLOGY

Usually, the analysis of a complex flow begins by the spatial discretization of the continuous equations into a finite-difference or a finite-volume model. The next step requires solving the resultant equations for the conserved flowfield variables, Q. In aerodynamic applications, the sensitivity calculation is then equal to obtaining the derivatives of the solution to the flowfield equations with respect to the design variables. This is due to the functional dependence of the objective function and the constraints on the flowfield solution. The sensitivity calculation is often the major computational cost of the optimization process. Hence, it is desirable in any optimization procedure to have efficient algorithms for the sensitivity calculations. In addition, these algorithms must be highly accurate for the success of the optimization process. This is the main subject of the present chapter.

8.1 Aerodynamic Sensitivity Coefficients

The derivatives of the objective function, F, and constraints, G_j, with respect to the design variables, $\overline{X_D}$, are commonly known as the sensitivity coefficients. In the field of structural mechanics, there are two popular approaches for calculating these coefficients [60]. These are the finite-difference approach and the analytical (or sensitivity analysis) approach. In this section, these methods are adapted in order to compute the aerodynamic sensitivity coefficients needed for aerodynamic shape optimization problems.

8.1.1 Finite-Difference Approach

The simplest technique for calculating the derivative of F and G_j is the finite-difference approximation (Fig. 8.1a). For example, the derivative of F with respect to a certain design variable, X_D, may be calculated using a first-order forward-difference approximation given by,

$$\frac{\partial F}{\partial X_D} \approx \frac{F\left[Q\left(X_D + \Delta X_D\right),\ X_D + \Delta X_D\right] - F\left[Q\left(X_D\right), X_D\right]}{\Delta X_D} \tag{8.1}$$

Another commonly used finite-difference approximation is the second-order central-difference

$$\frac{\partial F}{\partial X_D} \approx \frac{F\left[Q\left(X_D + \Delta X_D\right),\ X_D + \Delta X_D\right] - F\left[Q\left(X_D - \Delta X_D\right),\ X_D - \Delta X_D\right]}{2\Delta X_D} \tag{8.2}$$

As seen from the above equations, additional flowfield analyses are required to evaluate the finite-difference approximations. The total number of these analyses depends on the type of approximation and on how many design variables exist in the design space. For instance, for *NDV* design variables, it requires *NDV* additional flowfield analyses for the forward-difference approximation, Eq. (8.1), and *2*NDV* additional flowfield analyses for the central-difference approximation (Eq. (8.2)). Therefore, this approach is very expensive when the numbers of both design variables and flowfield governing equations are large. Even if these numbers are small, the accuracy of the approximation is greatly dependent on the step size, ΔX_D. Usually, the selection of the step size and the type of approximation is an estimate of the degree of the required accuracy [92]. Hence, this approach has the disadvantages of being neither cheap nor accurate. However, it is widely used becuase of its generality and its ease of implementation.

8.1.2 Analytical (or Sensitivity Analysis) Approach

Because of the functional dependence of the objective function and the constraints on design variables and flowfield solution, which is an implicit function of the design variables, the sensitivity coefficients can be obtained only via the chain rule of differentiation as follows:

$$\nabla F \equiv \frac{\partial F}{\partial \overline{X}_D} = \left(\frac{\partial F}{\partial \overline{X}_D}\right)_Q + \left(\frac{\partial F}{\partial Q}\right)^T \cdot \frac{\partial Q}{\partial \overline{X}_D} \tag{8.3}$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial \overline{X}_D} = \left(\frac{\partial G_j}{\partial \overline{X}_D}\right)_Q + \left(\frac{\partial G_j}{\partial Q}\right)^T \cdot \frac{\partial Q}{\partial \overline{X}_D}, \qquad j = 1, NCON \tag{8.4}$$

These gradients include two parts, namely, an explicit part $(\partial F/\partial \overline{X}_D)_Q$ plus an implicit part through the dependence on Q. The explicit part is often easy to obtain compared to the implicit part. Therefore, only the details for the computation of the implicit part are discussed in this subsection.

The discrete residual form of the governing equations, Eq. (6.33), can be rewritten in terms of the design variables and conserved flow variables as follows:

$$R\left(Q\left(\overline{X}_D\right), \overline{X}_D\right) = 0 \tag{8.5}$$

Consequently, the analytical approach begins with the differentiation of Eq. (8.5) with respect to the design variables to yield the sensitivity equation [61],

$$\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{\partial Q}{\partial \overline{X}_D}\right\} = -\left[\frac{\partial R}{\partial \overline{X}_D}\right] \equiv \boldsymbol{R_v} \tag{8.6}$$

Premultiplying Eq. (8.6) once by $[(\partial F/\partial Q)^T \cdot (\partial R/\partial Q)^{-1}]$ and another time by $[(\partial G_j/\partial Q)^T \cdot (\partial R/\partial Q)^{-1}]$ yields

$$\left[\frac{\partial F}{\partial Q}\right]^T \cdot \left\{\frac{\partial Q}{\partial \overline{X}_D}\right\} = \left[\frac{\partial F}{\partial Q}\right]^T \cdot \left[\frac{\partial R}{\partial Q}\right]^{-1} . \, \boldsymbol{R}_v \tag{8.7}$$

$$\left[\frac{\partial G_j}{\partial Q}\right]^T \cdot \left\{\frac{\partial Q}{\partial \overline{X}_D}\right\} = \left[\frac{\partial G_j}{\partial Q}\right]^T \cdot \left[\frac{\partial R}{\partial Q}\right]^{-1} . \, \boldsymbol{R}_v, \qquad j = 1, NCON_f \tag{8.8}$$

Numerically, the calculation of Eqs. (8.7) and (8.8) is performed in two ways. These are the direct sensitivity method and the adjoint variable method.

8.1.2.1 Direct Sensitivity Method

This method consists of solving Eq. (8.6) for $(\partial Q/\partial \overline{X}_D)$ and then taking the scalar product with $(\partial F/\partial Q)$ and $(\partial G_j/\partial Q)$. Here, Equation (8.6) needs to be solved once for each design variable, X_D, so the direct sensitivity method becomes costly when the number of design variables is large. However, one of the advantages of this method, as will be seen later in Section 8.3, is its use in predicting the flowfield solution without actually solving the governing equations of the field flow (Eq. (8.5)).

8.1.2.2 Adjoint Variable Method

The adjoint variable method defines vectors of adjoint variables (λ_1, λ_{2j}), which are the solutions of the systems,

$$\left[\frac{\partial R}{\partial Q}\right]^T . \, \boldsymbol{\lambda}_1 = \left[\frac{\partial F}{\partial Q}\right] \tag{8.9}$$

and

$$\left[\frac{\partial R}{\partial Q}\right]^T . \, \boldsymbol{\lambda}_{2j} = \left[\frac{\partial G_j}{\partial Q}\right], \qquad j = 1, NCON_f \tag{8.10}$$

The adjoint variable method requires the solution of Eq. (8.9) and also requires the solution of Eq. (8.10) for each flow constraint G_j. However, it should be noticed that the adjoint systems of equations (8.9) and (8.10) are independent of any differentiation with respect to $\overline{X}_D$. Hence, the vectors λ_1 and λ_{2j} in Eqs. (8.9) and (8.10) remain the same for all the elements of vector $\overline{X}_D$.

The difference between the computational effort associated with the direct sensitivity method and with the adjoint variable method depends on the relative number of adjoint vectors and design variables. For example, the adjoint variable method is more efficient than the direct method, when the number of adjoint vectors $(NCON_f+1)$ is smaller than the number of design variables. On the other hand, the direct sensitivity method is more efficient when the number of design variables is smaller than the number of adjoint vectors. Flow charts of these two methods are given in Fig. 8.1b.

Both the direct and adjoint methods involve the solution of a system of equations as a major part of their computational effort. However, most of the elements of the coefficient matrix, $[\partial R/\partial Q]$, are usually available from the flow solver, hence few computations are needed, as will be seen later. In addition, when solving for either the adjoint vectors or for $\{\partial Q/\partial \overline{X}_D\}$, the coefficient matrix needs be factorized once and for all, unlike in CFD analysis. As a result, the solution of the adjoint vectors or $\{\partial Q/\partial \overline{X}_D\}$ is much cheaper than the flowfield solution. This provides the major computational advantage of these two analytical methods over the finite-difference approach. This advantage becomes more pronounced for a large number of design variables.

8.2 Sensitivity Equation

As mentioned before, the sensitivity calculation is equivalent to the mathematical problem of obtaining the derivatives of the solution of the discretized flowfield governing equations with respect to the design variables. Therefore, the discretized residual form of the flowfield governing equations, Eq. (6.33), should now be rewritten in a form which explicitly includes the design variables or , at least, the terms that are directly related to the design variables. Unfortunately, the first possibility does not exist in aerodynamic shape optimization, since the design variables are usually the parameters that control the shape of the configuration. Hence, the second possibility is the only choice to express Eq. (6.33). In the present study, the second possibility is adopted as follows. An implicit relationship is defined between the controlling parameters and some geometric terms, such as the projected area and the cell volume, which are involved in the transformation to the generalized coordinates. Then, Eq. (6.33) is rewritten in terms of these chosen geometric terms ($M_{i\pm 1/2, j\pm 1/2}$) to yield the following functional form for the interior-cell *(i,j)*:

$$\begin{aligned}R_{i,j} = {} & \widehat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right) - \widehat{F}^{+}\left(Q^{-}_{i-1/2,j}, M_{i-1/2,j}\right) \\ & + \widehat{F}^{-}\left(Q^{+}_{i+1/2,j}, M_{i+1/2,j}\right) - \widehat{F}^{-}\left(Q^{+}_{i-1/2,j}, M_{i-1/2,j}\right) \\ & + \widehat{G}^{+}\left(Q^{-}_{i,j+1/2}, M_{i,j+1/2}\right) - \widehat{G}^{+}\left(Q^{-}_{i,j-1/2}, M_{i,j-1/2}\right) \\ & + \widehat{G}^{-}\left(Q^{+}_{i,j+1/2}, M_{i,j+1/2}\right) - \widehat{G}^{-}\left(Q^{-}_{i,j-1/2}, M_{i,j-1/2}\right) \\ & - \left[\widehat{G}_{v}\left(Q_{i,j+1}, Q_{i,j}, M_{i,j+1/2}\right)\right]_{i,j+1/2} \\ & + \left[\widehat{G}_{v}\left(Q_{i,j}, Q_{i,j-1}, M_{i,j-1/2}\right)\right]_{i,j-1/2}\end{aligned} \qquad (8.11)$$

The sensitivity equation (Eq. (8.6)) is always simultaneous, linear, and algebraic, regardless of the mathematical nature (nonlinear, transcendental, etc.) of the flowfield governing equations. In Eq. (8.6), the coefficient matrix is a Jacobian matrix of the partial derivatives with respect to the conserved flowfield variables, and the right hand side vector contains the partial derivatives with respect to a particular design variable. These partial derivatives are evaluated using the $\overline{X_D}$ and Q values which satisfy Eq. (8.5). In other words, a solution Q of the flowfield governing equations is a prerequisite to forming and solving the sensitivity equation.

To form the sensitivity equation (Eq. (8.6)), the above discrete form, Eq. (8.11), needs to be differentiated with respect to the conserved flowfield variables and the geometric terms. This methodology is called the ***discrete sensitivity analysis*** and it is described in Secs. 8.2.1 and 8.2.2.

8.2.1 Left Hand Side (LHS)

To construct the LHS of the sensitivity equation, $R_{i,j}$ has to be differentiated with respect to the cell-centered values of the conserved flow variables for all cells in the flowfield. Fortunately, as seen from Eq. (8.11), $R_{i,j}$ is only functionally dependent on the cell-centered values of the conserved flow variables, $Q_{i,j}$ *and* $Q_{i,j\pm1}$, and the cell-face values of the conserved variables $Q^{\pm}_{i\pm1/2,j\pm1/2}$. In addition, by using Eqs. (6.20), (6.28), and (6.29), $Q^{\pm}_{i\pm1/2,j\pm1/2}$ can be viewed as functions of the conserved flowfield variables of the neighboring cells evaluated at their cell centers, i.e., *(*$Q_{i,j}$, $Q_{i\pm1,j\pm1}$, *and* $Q_{i\pm2,j\pm2}$*)*. Therefore, $R_{i,j}$ needs to be differentiated only with respect to $Q_{i,j}$, $Q_{i\pm1,j\pm1}$, *and* $Q_{i\pm2,j\pm2}$ in order to obtain the following linear vector form of the LHS at the interior-cell *(i,j)* :

$$\begin{aligned}[LHS\ of\ Eq.\ (8.6)]_{i,j} &= \overline{D}\frac{\partial Q_{i-2,j}}{\partial \overline{X}_D} + \overline{A}\frac{\partial Q_{i-1,j}}{\partial \overline{X}_D} + \overline{B}\frac{\partial Q_{i,j}}{\partial \overline{X}_D} \\ &+ \overline{C}\frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} + \overline{E}\frac{\partial Q_{i+2,j}}{\partial \overline{X}_D} + \overline{H}\frac{\partial Q_{i,j-2}}{\partial \overline{X}_D} \\ &+ \overline{F}\frac{\partial Q_{i,j-1}}{\partial \overline{X}_D} + \overline{G}\frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} + \overline{I}\frac{\partial Q_{i,j+2}}{\partial \overline{X}_D} \qquad (8.12)\end{aligned}$$

The coefficients $\overline{A}$ through $\overline{I}$ in the above equation are 4x4 blocks for 2-D equations and are functions of the Jacobian matrices of the split fluxes $\widehat{F}^{\pm}$ and $\widehat{G}^{\pm}$, and the Jacobians of the viscous fluxes. Further details of Eq. (8.12) are described in Appendix C.

Since all Jacobian matrices are available for the interior cells from the flowfield solver, few computations are needed to assemble the coefficient matrix $[\partial R/\partial Q]$ at the interior points. However, it is necessary to revise Eq. (8.12) at the boundary points in order to include the boundary conditions. For example, if the conserved flowfield variables of the boundary condition at $i = 1$ boundary is related only to conserved flowfield variables of the first adjacent interior-cell, Eq. (8.12) becomes

$$\begin{aligned}[LHS\ of\ Eq.\ (8.6)]_{i,j} = &\left[\overline{D}\frac{\partial Q_{i-2,j}}{\partial Q_{i,j}} + \overline{A}\frac{\partial Q_{i-1,j}}{\partial Q_{i,j}} + \overline{B}\right]\frac{\partial Q_{i,j}}{\partial \overline{X}_D} \\ &+ \overline{C}\frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} + \overline{E}\frac{\partial Q_{i+2,j}}{\partial \overline{X}_D} + \overline{H}\frac{\partial Q_{i,j-2}}{\partial \overline{X}_D} \\ &+ \overline{F}\frac{\partial Q_{i,j-1}}{\partial \overline{X}_D} + \overline{G}\frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} + \overline{I}\frac{\partial Q_{i,j+2}}{\partial \overline{X}_D}\end{aligned} \qquad (8.13)$$

where $\overline{A}$, $\overline{B}$, and $\overline{D}$ are modified according to the relations between $Q_{i,j}$ and both $Q_{i-2,j}$ and $Q_{i-1,j}$. Similar expressions can also be obtained for the other boundaries. Another example for a revised expression for Eq. (8.12) as applied to Eq. (6.35) is given in Appendix C.

The coefficient matrix $[\partial R/\partial Q]$ as well as its transpose are sparse square matrices with a block-banded structure, wherein all the nonzero elements are confined within a band formed by diagonals parallel to the main diagonal. The number of nonzero diagonals in the matrix depends on the type of both discretization of the governing equations and the flow regime. For example, for two-dimensional problems, the first-order upwind discretization, which requires a five-point stencil, results in five nonzero block diagonals in which the elements are (4x4) blocks. Whereas, the second- or third-order upwind discretizations, which require a nine-point stencil, yield nine nonzero block diagonals. Moreover, when dealing with a supersonic flow in the streamwise direction and using an upwind scheme to discretize of the governing equations, either the positive or the negative fluxes in the streamwise direction are zeros. Hence, the number of nonzero diagonals reduces either to four in the case of first-order upwind discretization or to seven in the case of second- or third-order upwind discretizations. More details about the structure of the coefficient matrix are given in Section (8.4.1).

An important advantage of using an upwind scheme over a central difference scheme is that, in the latter, zero elements may appear in the diagonal. This causes the failure of the solution to the system of linear equations if no pivoting strategy is employed in the solution algorithm.

8.2.2 Right Hand Side (RHS)

The *(i,j)* row of the RHS of the sensitivity equation $[\partial R_{i,j}/\partial \overline{X}_D]$ is constructed by differentiating first the $R_{i,j}$ with respect to the geometrical terms $(M_{i\pm 1/2,j\pm 1/2})$ and then multiplying the results by the derivatives of these geometric terms with respect to the design variables, $\overline{X}_D$, as follows:

$$\begin{aligned}\frac{\partial R_{i,j}}{\partial \overline{X}_D} = & \left[\frac{\partial \widehat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial M_{i+1/2,j}} + \frac{\partial \widehat{F}^{-}\left(Q^{+}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial M_{i+1/2,j}}\right] \cdot \frac{\partial M_{i+1/2,j}}{\partial \overline{X}_D} \\ & - \left[\frac{\partial \widehat{F}^{+}\left(Q^{-}_{i-1/2,j}, M_{i-1/2,j}\right)}{\partial M_{i-1/2,j}} + \frac{\partial \widehat{F}^{-}\left(Q^{+}_{i-1/2,j}, M_{i-1/2,j}\right)}{\partial M_{i-1/2,j}}\right] \cdot \frac{\partial M_{i-1/2,j}}{\partial \overline{X}_D} \\ & + \left[\frac{\partial \widehat{G}^{+}\left(Q^{-}_{i,j+1/2}, M_{i,j+1/2}\right)}{\partial M_{i,j+1/2}} + \frac{\partial \widehat{G}^{-}\left(Q^{+}_{i,j+1/2}, M_{i,j+1/2}\right)}{\partial M_{i,j+1/2}}\right] \cdot \frac{\partial M_{i,j+1/2}}{\partial \overline{X}_D} \\ & - \left[\frac{\partial \widehat{G}^{+}\left(Q^{-}_{i,j-1/2}, M_{i,j-1/2}\right)}{\partial M_{i,j-1/2}} + \frac{\partial \widehat{G}^{-}\left(Q^{+}_{i,j-1/2}, M_{i,j-1/2}\right)}{\partial M_{i,j-1/2}}\right] \cdot \frac{\partial M_{i,j-1/2}}{\partial \overline{X}_D} \\ & - \frac{\partial \widehat{G}_v\left(Q_{i,j+1}, Q_{i,j}, M_{i,j+1/2}\right)}{\partial M_{i,j+1/2}} \cdot \frac{\partial M_{i,j+1/2}}{\partial \overline{X}_D} \\ & + \frac{\partial \widehat{G}_v\left(Q_{i,j}, Q_{i,j-1}, M_{i,j-1/2}\right)}{\partial M_{i,j-1/2}} \cdot \frac{\partial M_{i,j-1/2}}{\partial \overline{X}_D}\end{aligned} \tag{8.14}$$

In Eq. (8.14), the derivatives of both split-fluxes and viscous fluxes with respect to the geometric terms are straightforward to obtain, and detailed examples are given in details in Appendix D.

Determining the derivatives of the geometric terms $(M_{i\pm 1/2,j\pm 1/2})$ with respect to the design variables, $\overline{X}_D$, requires an analytical expression for $M = M(\overline{X}_D)$. In other words, it depends on whether the Cartesian coordinates $(\overline{x}, \overline{y})$, and consequently the equations for the grid generation routine, are directly related to the design variables; usually, they are not, except at the boundaries of the aerodynamic body where the shape is changed by the design variables. Therefore, the geometrical terms $(M_{i\pm 1/2,j\pm 1/2})$ need to be first expressed in terms of the coordinates $(\bar{x}, \bar{y})$ which can easily be expressed in terms of the coordinates at the boundaries $(\bar{x}_b, \bar{y}_b)$. Then, the coordinates $(\bar{x}_b, \bar{y}_b)$ of the boundaries are expressed in terms of the design variables. This procedure of expressing $(M_{i\pm 1/2,j\pm 1/2})$ enjoins that their derivatives are expressed as,

$$\frac{\partial(M_{i\pm1/2,j\pm1/2})}{\partial\bar{X}_D} = \frac{\partial(M_{i\pm1/2,j\pm1/2})}{\partial(\bar{x})} \cdot \frac{\partial(\bar{x})}{\partial\bar{x}_b} \cdot \frac{\partial(\bar{x}_b)}{\partial\bar{X}_D} + \frac{\partial(M_{i\pm1/2,j\pm1/2})}{\partial(\bar{y})} \cdot \frac{\partial(\bar{y})}{\partial\bar{y}_b} \cdot \frac{\partial(\bar{y}_b)}{\partial\bar{X}_D} \qquad (8.15)$$

Advantages of this procedure are its generality, its ease of implementation, and its high accuracy. These are usually the necessary requirements for a good sensitivity analysis routine.

Another method for determining these derivatives is using of a finite-difference approximation for $\partial(M_{i\pm 1/2,j\pm 1/2})/\partial\bar{X}_D$ with a small step size, $\Delta\bar{X}_D$. In spite of the simplicity of this method and its generality, it produces highly inaccurate results due to the truncation and condition errors inherent into the approximation [92]. Hence, the finite-difference approximations are not recommended to be used on a regular basis other than in providing a check for the terms of Eq. (8.15).

8.3 Flowfield Prediction Method (Approximate Analysis)

In any optimization procedure the objective function and constraints are usually evaluated several times before reaching their optimum values. In addition, for aerodynamic applications each of these evaluations requires a flowfield solution. Therefore, it would be very expensive if a CFD analysis is performed each time. This necessitates finding a technique that takes advantage of the computations performed at one design point to reduce the computational cost of the analysis at another design point. Fortunately, by-products of the direct sensitivity method (Sec. 8.1.2.1) for calculating the aerodynamic sensitivity coefficients are the derivatives of the flowfield solution with respect to the design variables. Hence, using these derivatives at a certain design point along with the flowfield solution at the same point can provide the solution at another design point. This concept is referred herein as the *flowfield prediction method*, and it is inspired by the reanalysis used in the structural design optimization [93]. Although this technique is approximate in nature, it works well in the structural design optimization problems when the latter design point is close to the former.

In the flowfield prediction method, the new flowfield solution vector $\{Q^*\}$ corresponding to a modified set of design variables $\overline{X_D}^*$ is approximated by a truncated Taylor's series,

$$Q^*\left(\overline{X_D^*}\right) = Q^o\left(\overline{X_D^o}\right) + \sum_{i=1}^{NDV} \left(X_{Di}^* - X_{Di}^o\right) \left\{\frac{\partial Q}{\partial X_{Di}}\right\}_{\overline{X_D} = \overline{X_D^o}} \tag{8.16}$$

where $\{\partial Q/\partial X_{D\,i}\}$ is the first-order sensitivity derivative of the flowfield solution vector, and $\{Q^o\}$ is the flowfield solution vector corresponding to the set of design variables, $\overline{X_D^o}$.

Substitution of Eq. (8.6) into Eq. (8.16) results in,

$$\left[\frac{\partial R\left(Q^o(\overline{X_D^o}),\overline{X_D^o}\right)}{\partial Q}\right] \Delta Q = -\left[\frac{\partial R\left(Q^o(\overline{X_D^o}),\overline{X_D^o}\right)}{\partial \overline{X_D}}\right]\left(\overline{X_D^*} - \overline{X_D^o}\right) \tag{8.17}$$

where $\Delta Q = Q^*\left(\overline{X_D}^*\right) - Q^o\left(\overline{X_D^o}\right)$. Equation (8.17) gives the changes in Q due to changes in $\overline{X_D}$. In other words, the flowfield solution, $\{Q^*\}$, associated with the configuration, $\overline{X_D}^*$, is obtained via Eq. (8.17), when the flowfield solution, $\{Q^o\}$, associated with the configuration, $\overline{X_D^o}$, is given.

The flowfield solution, $\{Q^o\}$, in this method can either be obtained via a CFD analysis or be a previously predicted flowfield solution. The latter procedure is called *prediction based on prediction*, whereas the former is called *prediction based on analysis*. The second procedure allows flowfield solutions to be progressively "built up" from previous predictions, all of which have the common genesis of a single initial CFD analysis solution. Thus, a flowfield solution for a complex final shape may be obtained through incrementally additive design variables modifications. Otherwise, a grossly erroneous prediction is produced if an equivalent large design variables modification is attempted. It should be noted, however, that the first-order sensitivity derivative of the flowfield solution has to be updated at every prediction using the updated flowfield solution.

Due to the truncation error of the first-order Taylor's series, the flow prediction is certainly less accurate than the CFD analysis. Consequently, it is natural to question the quality and accuracy of the successive flowfield predictions when different step sizes, $\Delta\overline{X}_D$'s, are used. This is crucial for the success of the optimization procedure since it helps in putting the proper limits on $\Delta\overline{X}_D$ during the optimization process. Thus, a section in the chapter describing the results (Sec. 10.2) is devoted to explore this issue. Nevertheless, solving flowfield prediction of Eq. (8.16) costs only a small fraction of solving Eq. (8.11), since [∂R/∂Q] and [$\partial R/\overline{X}_D$] are already assembled (i.e., Eqs. (8.12) and (8.14)) in solving the sensitivity equation (Eq. (8.6)). Therefore, for relatively small values of $\Delta\overline{X}_D$, significant time savings are realized at the expense of some accuracy.

8.4 Solution Methods for the Linear Equations of the Analytical Approach

As mentioned in Section 8.2, both direct sensitivity and adjoint variable methods involve the solution of a system of linear equations (e.g., Eq. (8.6) and Eqs. (8.9)–(8.10)) as a major part of their computational effort. Hence, it is necessary for the numerical methods which solve these systems to be efficient and feasible in order not only to reduce the cost of obtaining the sensitivity coefficients, but also to increase the efficiency of the overall optimization procedure. Methods for solving systems of linear algebraic equations are readily divided into two classes, namely, the direct methods and the iterative methods. The choice of the proper methods is often hard since it is usually affected by one of the following factors: storage requirements, applicability, accuracy obtained, and CPU-time required. Some of these factors which are deemed important to the present study are investigated in the next subsections.

8.4.1 Comments on Direct Matrix Inversion Methods

For two-dimensional problems, in which the computational domain has I cells in the ξ-direction and J cells in the η-direction, the coefficient matrix $[\partial R/\partial Q]$ (or its transpose) has the dimension $(n \times n)$, where n is $(4IJ)$. The storage requirement is about the same for most of the general direct methods, that is, a minimum of about $n^2 + rn$, where r is the number of right-hand sides. Hence, for problems where $(16{*}(IJ)^2)$ is large, the computer memory allocation will be prohibitively high, sometimes to the extent that it requires much larger computer core memory than those available in supercomputers. Therefore, the storage requirement is usually considered the most important factor affecting the choice in direct methods.

Consequently, two types of storage techniques [94] are devised in the present study, that take advantage of the matrix structure to reduce the storage. The first technique is the sparse matrix storage technique in which only the nonzero elements are stored. The second technique is the diagonal storage of the banded matrices in which all the elements within the bandwidth including the zeros are stored. In spite of the large reduction in storage usually achieved by these techniques, the computer memory requirement is still high for very large two-dimensional problems as well as three-dimensional problems.

Hence, further measures are sought to reduce the storage requirements, and they will be discussed next.

One effective measure is using of different strategies in ordering the unknowns, such that it takes advantage of the flow regime, the type of computational grid, and the number of unknowns in each flow direction. For example, when dealing with a supersonic flow in the streamwise direction and using an upwind scheme to discretize the governing equations, either the positive or the negative fluxes in the streamwise direction are zeros. Therefore, if the unknowns are ordered such that the outer sweep is in tthe normal direction (η-direction), the coefficient matrix will have super- and sub-diagonal bandwidths $(4I+3)$ or $(8I+3)$ depending on the type of discretization (Fig. 8.2). Consequently, when using the banded storage technique, it is required to store $(8I+7)*(4IJ)$ for first-order discretization and $(16I+7)*(4IJ)$ elements for second- or thrid-order discretizations. Whereas, if the unknowns are ordered such that the outer sweep is in the streamwise direction (ξ-direction), the coefficient matrix will have a super-diagonal bandwidth of $(4J+3)$ and sub-diagonal bandwidth of 7, or a super-diagonal bandwidth $(8J+3)$ and sub-diagonal bandwidth of 11 depending on the type of discretization (Fig. 8.3). In this type of ordering, the number of elements needed to be stored is only $(4J+11)*(4IJ)$ for the first-order discretization and it is only $(8J+15)*(4IJ)$ for the second-order discretization. Therefore, if I and J are equal, the second type, which takes advantage of the flow regime, will result in about 60% memory saving over the first type.

To see the effect of unknown ordering on the memory requirements when different types of computational grids are used, a representative C-grid around an airfoil (Fig. 8.4a) is considered. The cell nodes are ordered in one of various ways, which is called Order A herein [79,88]. For third-order upwind discretization of the governing equations, this ordering leads to a fully banded matrix (Fig. 8.4b) with a half-bandwidth of $(16*J+3)$. Shown in Fig. 8.5a is a C-grid for an airfoil with the more commonly used ordering, which is called Order *B* herein. This ordering does not lead to a banded structure because of the coupling between both sides of the wake cut. Therefore, the coefficient matrix $[\partial R/\partial Q]$ has a banded matrix with a half-bandwidth of $(8*J+3)$ plus extraneous enteris outside the band (Fig. 8.5b). This results in requiring a sparse matrix solver [95], or full matrix storage, which requires a prohibitively large memory. The third type of ordering is called Order *C* herein and it is shown in Fig. 8.6a. Although, for the third-order upwind discretization Order *C* yields a fully banded matrix (Fig. 8.6b) like Order *A* but with a half-bandwidth of $(8*I+3)$, it has the disadvantage of requiring relatively more memory since *I* is usually greater than *J*. Therefore, Order A is considered the most efficient one for this type of computational grid. In fact, Order A for the *C*-type grid is equivalent to using an *H*-grid. Hence, different strategies in ordering the unknowns, which take advantage of the type of computational grid, and the number of unknowns in each direction, can affect the memory requirements significantly.

The accuracy of the direct methods depends on the specific system of equations in question and the matter is too complex to be said in a general statement. However, to highlight this issue, it is important to know the basic idea of direct methods. Direct methods usually solve the system of linear equations in a finite number of operations which can be specified in advance, so that the number of operations performed is independent of the accuracy desired. An exact solution would be obtained if there were no round-off errors. Hence, the accuracy of the approximate solution obtained by a computer is a function of the condition and size of the matrix, the accuracy of the arithmetic as performed by the computer, and the algorithm used for solution [96].

In the present study, two direct matrix solvers have been used to solve the systems given by Eq. (8.6) and Eqs. (8.9)-(8.10), and are compared with each other to assess their efficiency, accuracy, and storage requirements. These are the sparse matrix solver [95] and the banded matrix solver [97]. Both of these solvers use the standard Gauss elimination method after the off-diagonal zeros have been eliminated.

8.4.2 Comments on Iterative Methods

Generally, iterative methods are preferred for solving a single large sparse system of linear equations for which convergence is known to be rapid. Iterative methods, however, are less useful for solving systems with many right-hand sides. Usually, iterative methods begin with an approximate solution to the linear system and obtain an improved solution with each step of the process. An iterative process would require an infinite number of steps to obtain an exact solution. The accuracy of the obtained solution depends on the number of iterations performed, the condition of the matrix as well as its size, the accuracy of the arithmetic performed by the computer and the particular algorithm used.

One of the important advantages of the iterative methods over the direct methods is the relatively low computer memory requirements. However, it is important to distinguish between two types of iterative methods. The first type is the relaxation schemes which require diagonally dominant coefficient matrices. Since this restriction is satisfied only by the first-order upwind finite-volume representations of the flowfield governing equations [98], the relaxation schemes are restricted to this class of discretization of the flowfield governing equations. The second type of methods is the iterative schemes which are based on the conjugate gradient methods, such as, the generalized minimum residual (*GMRES*) methods [99]. This latter type does not require diagonally dominant coefficient matrices, but it does require the eigenvalues of the coefficient matrices to be clustered around unity. Thus, preconditioning of the coefficient matrices is essential for the success of these methods. In addition, the consequent choice of an effective and stable preconditioner is extremely important to the success of these methods, as it is further discussed in Chapter 10.

8.5 Sensitivity Analysis Domain Decomposition (SADD) scheme

For large two-dimensional and three-dimensional problems, both direct matrix inversion and iterative methods become impractical to be used because of the prohibitively high computer memory requirements. A viable and principally different alternative is to develop a domain-decomposition technique that breaks up the single-domain into small subdomains and then to solve the sensitivity equation of each subdomain separately. Hence, the domain-decomposition technique reduces the high memory requirements of the single-domain. Another advantage of this technique over a single-domain technique is the ease of grid generation for complex-geometry problems. A complex aerodynamic configuration may have multiple, joint or disjoint components, all of which may be geometrically nonsimilar [72-74]. The Hybrid Domain Decomposition (HDD) approach reported in Refs. [72-74] has been developed in order to obtain the flowfield solutions for these problems. The HDD scheme takes advantage of all popular domain decomposition techniques used in CFD, that is, multiblock grids, zonal grid, and overlapped grids. However, in the present approach the sensitivity analysis will only be applied to problems where the decomposed computational domain is of multiblock type. This technique, referred to here as *Sensitivity Analysis Domain Decomposition (SADD) scheme*, is inspired by the boundary-element substructuring methods developed in strucutral mechanics [75] and the global sensitivity equation (GSE) procedure developed by Sobieski [78] to calculate the sensitivity of a multidisciplinary system with respect to design variables.

The Sensitivity Analysis Domain Decomposition (*SADD*) scheme works by dividing the computational domain around the configuration into a number of subdomains. Then, each subdomain is divided again into internal and boundary-interface cells (Fig. 8.7). The idea behind the break-up within the subdomain is to find a set of unknowns which decouples the computations of the individual subdomains. As a result, the sensitivity equation for each subdomain can be written as:

$$\begin{bmatrix} \left[\dfrac{\partial R^{i}}{\partial Q^{i}}\right]^{k} & \left[\dfrac{\partial R^{i}}{\partial Q^{b}}\right]^{k} \\ \left[\dfrac{\partial R^{b}}{\partial Q^{i}}\right]^{k} & \left[\dfrac{\partial R^{b}}{\partial Q^{b}}\right] \end{bmatrix} \bullet \begin{pmatrix} \left\{\dfrac{\partial Q^{i}}{\partial \bar{X}_D}\right\}^{k} \\ \left\{\dfrac{\partial Q^{b}}{\partial \bar{X}_D}\right\} \end{pmatrix} = \begin{bmatrix} \left[\dfrac{\partial R^{i}}{\partial \bar{X}_D}\right]^{k} \\ \left[\dfrac{\partial R^{b}}{\partial \bar{X}_D}\right] \end{bmatrix}, \qquad k = 1, NBL \tag{8.18}$$

where i refers to the internal cells of the kth subdomain and b refers to the boundary-interface cells for all subdomains, which are used in the decoupling process. The total number of subdomains is *NBL*. The coefficient matrix for the interior of the kth subdomain is $[\partial R^i/\partial Q^i]^k$. The coefficient matrices resulting from the interaction between the kth subdomain and all boundary-interface cells and vice versa are $[\partial R^i/\partial Q^b]^k$ and $[\partial R^b/\partial Q^i]^k$. The coefficient matrix for all the boundary-interface cells is $[\partial R^b/\partial Q^b]$. An example for the assembling of these matrices is illustrated in Fig. 8.8. The terms $\{\partial Q^i/\partial \bar{X}_D\}^k$ and $\{\partial Q^b/\partial \bar{X}_D\}$ are the sensitivity derivatives of the flowfield solution for the internal cells of the kth subdomain and all boundary-interface cells, respectively. The terms $[\partial R^i/\partial \bar{X}_D]^k$ and $[\partial R^b/\partial \bar{X}_D]$ are the sensitivity derivatives of the discretized residual form of the governing equations for the internal cells of the kth subdomain and all boundary-interface cells, respectively.

The sensitivity of the flowfield solution for the internal cells $\{\partial Q^i/\partial \bar{X}_D\}^k$ is expressed in terms of the sensitivities of all flowfield solutions at boundary-interface cells $\{\partial Q^b/\partial \bar{X}_D\}$. This can be shown by using Eq. (8.18) as follows:

$$\left\{\frac{\partial Q^{i}}{\partial \bar{X}_D}\right\}^{k} = -\left[\left[\frac{\partial R^{i}}{\partial Q^{i}}\right]^{k}\right]^{-1}\left[\frac{\partial R^{i}}{\partial Q^{b}}\right]^{k}\left\{\frac{\partial Q^{b}}{\partial \bar{X}_D}\right\} + \left[\left[\frac{\partial R^{i}}{\partial Q^{i}}\right]^{k}\right]^{-1}\left[\frac{\partial R^{i}}{\partial \bar{X}_D}\right]^{k}, \tag{8.19}$$

Substituting Eq.(8.19) into Eq.(8.18) to eliminate $\{\partial Q^i/\partial \bar{X}_D\}^k$ and repeating this procedure for other subdomains yields

$$\left[\frac{\partial R}{\partial Q}\right]_e \left\{\frac{\partial Q^b}{\partial \overline{X}_D}\right\} = \left[\frac{\partial R}{\partial \overline{X}_D}\right]_e \tag{8.20}$$

where $[\,\partial R \,/\, \partial Q\,]_e$ is the effective coefficient matrix for the boundary-interface cells given by,

$$\left[\frac{\partial R}{\partial Q}\right]_e = \left[\frac{\partial R^b}{\partial Q^b}\right] - \sum_{k=1}^{NBL} \left[\frac{\partial R^b}{\partial Q^i}\right]^k \left[\left[\frac{\partial R^i}{\partial Q^i}\right]^k\right]^{-1} \left[\frac{\partial R^i}{\partial Q^b}\right]^k \tag{8.21}$$

and $[\partial R \,/\, \partial \overline{X}_D]_e$ is the effective *RHS* for the boundary-interface cells given by,

$$\left[\frac{\partial R}{\partial \overline{X}_D}\right]_e = \left[\frac{\partial R^b}{\partial \overline{X}_D}\right] - \sum_{k=1}^{NBL} \left[\frac{\partial R^b}{\partial Q^i}\right]^k \left[\left[\frac{\partial R^i}{\partial Q^i}\right]^k\right]^{-1} \left[\frac{\partial R^i}{\partial \overline{X}_D}\right]^k \tag{8.22}$$

The matrix inversions in Eqs. (8.21) and (8.22) are not done explicitly. Instead, the following products given by Eqs. (8.23) and (8.24) are obtained by solving the systems of Eq. (8.25) and Eq. (8.26), respectively:

$$\left[\left[\frac{\partial R^i}{\partial Q^i}\right]^k\right]^{-1} \left[\frac{\partial R^i}{\partial Q^b}\right]^k = [\,M\,]_e^k, \tag{8.23}$$

and

$$\left[\left[\frac{\partial R^i}{\partial Q^i}\right]^k\right]^{-1} \left[\frac{\partial R^i}{\partial \overline{X}_D}\right]^k = [\,Z\,]_e^k \tag{8.24}$$

$$\left[\frac{\partial R^i}{\partial Q^i}\right]^k [\,M\,]_e^k = \left[\frac{\partial R^i}{\partial Q^b}\right]^k \tag{8.25}$$

and

$$\left[\frac{\partial R^i}{\partial Q^i}\right]^k [\,Z\,]_e^k = \left[\frac{\partial R^i}{\partial \overline{X}_D}\right]^k \tag{8.26}$$

It should be noticed that Eq. (8.25) and Eq. (8.26) have the same coefficient matrix and are similar to Eq. (8.6) for a single domain. This means that the previous decoupling process has resulted in breaking up the single domain into small sub-domains which can be solved separately. Hence, the *SADD* scheme can now be summarized in the following steps:

(1) Form the *LU*-decomposition of the coefficient matrix $[\partial R^i/\partial Q^i]^k$ of the subdomain k and solve the systems Eqs. (8.25) and (8.26). If obtaining the *LU*-decomposition is expensive, then it is better to save the *LU* factors into a computer diskfile.

(2) Form the products $[\partial R^b/\partial Q^i]^k \cdot [M]_e^k$ and $[\partial R^b/\partial Q^i]^k \cdot [Z]_e^k$ for the subdomain k.

(3) Substitute into Eq. (8.21) and Eq. (8.22).

(4) Repeat the steps 1 through 3 for all the subdomains to form $[\partial R/\partial Q]_e$ and $[\partial R/\partial \overline{X}_D]_e$.

(5) Solve Eq. (8.20) to obtain $\{\partial Q^b/\partial \overline{X}_D\}$.

(6) Substitute $\{\partial Q^b/\partial \overline{X}_D\}$ into the following modified form of Eq. (8.19)

$$[\,LU\,]^k \left\{\frac{\partial Q^i}{\partial \overline{X}_D}\right\}^k = -\left[\frac{\partial R^i}{\partial Q^b}\right]^k \left\{\frac{\partial Q^b}{\partial \overline{X}_D}\right\} + \left[\frac{\partial R^i}{\partial \overline{X}_D}\right]^k \qquad (8.27)$$

then use the saved *LU* factors of the subdomain k to obtain $\{\partial Q^i/\partial \overline{X}_D\}^k$.

(7) Repeat Step 6 for all the subdomains, i.e., *NBL* times, to obtain the sensitivity of the flowfield solutions of their internal cells.

It is clear from the SADD scheme that it is highly efficient for parallel computers since the subdomains are independent of each other. However, if a domain is broken-up into too many small subdomains in order to reduce the memory requirements, many boundary-interface cells will be created. Consequently, the number of right hand sides of Eq. (8.25) will dramatically increase. This means that reducing the memory requirements results in high computational costs. Therefore, it is recommended that the number of subdomains be kept as small as possible in order not to offset the computational advantage of the scheme.

When the *SADD* scheme is applied to the systems of the adjoint vectors given by Eqs. (8.9) and (8.10), it yields

$$\begin{bmatrix} \left[\left[\dfrac{\partial R^i}{\partial Q^i}\right]^T\right]^k & \left[\left[\dfrac{\partial R^b}{\partial Q^i}\right]^T\right]^k \\ \left[\left[\dfrac{\partial R^i}{\partial Q^b}\right]^T\right]^k & \left[\dfrac{\partial R^b}{\partial Q^b}\right]^T \end{bmatrix} \bullet \begin{Bmatrix} \left\{\lambda_1^i\right\}^k \\ \left\{\lambda_1^b\right\} \end{Bmatrix} = \begin{bmatrix} \left[\dfrac{\partial F}{\partial Q^i}\right]^k \\ \left[\dfrac{\partial F}{\partial Q^b}\right] \end{bmatrix}, \qquad k = 1, NBL \qquad (8.28)$$

and

$$\begin{bmatrix} \left[\left[\dfrac{\partial R^i}{\partial Q^i}\right]^T\right]^k & \left[\left[\dfrac{\partial R^b}{\partial Q^i}\right]^T\right]^k \\ \left[\left[\dfrac{\partial R^i}{\partial Q^b}\right]^T\right]^k & \left[\dfrac{\partial R^b}{\partial Q^b}\right]^T \end{bmatrix} \bullet \begin{Bmatrix} \left\{\lambda_{2j}^i\right\}^k \\ \left\{\lambda_{2j}^b\right\} \end{Bmatrix} = \begin{bmatrix} \left[\dfrac{\partial G_j}{\partial Q^i}\right]^k \\ \left[\dfrac{\partial G_j}{\partial Q^b}\right] \end{bmatrix}, \qquad \begin{matrix} k = 1, NBL \\ j = 1, NCON_f \end{matrix} \qquad (8.29)$$

where $\left(\left\{\lambda_1^i\right\}^k, \left\{\lambda_{2j}^i\right\}^k\right)$ and $\left(\left\{\lambda_1^b\right\}, \left\{\lambda_{2j}^b\right\}\right)$ are the adjoint vectors for the internal cells of the *k*th subdomain and all the boundary-interface cells, respectively.

8.5.1 The Analytical (or Sensitivity Analysis) Approach Using the *SADD* Scheme:

Equations (8.3) and (8.4) of the aerodynamic sensitivity coefficients given in Section [8.2] are adapted to the multi-domains using Eqs. (8.18), (8.28), and (8.29). For instance, these coefficients take the following forms in case of the direct sensitivity method:

$$\nabla F \equiv \frac{\partial F}{\partial X_D} = \left(\frac{\partial F}{\partial \overline{X}_D}\right)_Q + \left(\frac{\partial F}{\partial Q^b}\right)^T \bullet \left\{\frac{\partial Q^b}{\partial \overline{X}_D}\right\} + \sum_{k=1}^{NBL} \left[\left(\frac{\partial F}{\partial Q^i}\right)^T \bullet \left\{\frac{\partial Q^i}{\partial \overline{X}_D}\right\}\right]^k \qquad (8.30)$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial \overline{X}_D} = \left(\frac{\partial G_j}{\partial \overline{X}_D}\right)_Q + \left(\frac{\partial G_j}{\partial Q^b}\right)^T \bullet \left\{\frac{\partial Q^b}{\partial \overline{X}_D}\right\} + \sum_{k=1}^{NBL} \left[\left(\frac{\partial G_j}{\partial Q^i}\right)^T \bullet \left\{\frac{\partial Q^i}{\partial \overline{X}_D}\right\}\right]^k ,$$

$$j = 1, NCON_f \quad (8.31)$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial \overline{X}_D}, \qquad j = NCON_f + 1, NCON \quad (8.32)$$

The adjoint variable method, the aerodynamic sensitivity coefficients take the following forms after using Eqs. (8.28) and (8.29):

$$\nabla F \equiv \frac{\partial F}{\partial \overline{X}_D} = \left(\frac{\partial F}{\partial \overline{X}_D}\right)_Q + \left(\lambda_1^b\right)^T \bullet \left\{\frac{\partial R^b}{\partial \overline{X}_D}\right\} + \sum_{k=1}^{NBL} \left[\left(\lambda_1^i\right)^T \bullet \left\{\frac{\partial R^i}{\partial \overline{X}_D}\right\}\right]^k \quad (8.33)$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial \overline{X}_D} = \left(\frac{\partial G_j}{\partial \overline{X}_D}\right)_Q + \left(\lambda_{2j}^b\right)^T \bullet \left\{\frac{\partial R^b}{\partial \overline{X}_D}\right\} + \sum_{k=1}^{NBL} \left[\left(\lambda_{2j}^i\right)^T \bullet \left\{\frac{\partial R^i}{\partial \overline{X}_D}\right\}\right]^k ,$$

$$j = 1, NCON_f \quad (8.34)$$

$$\nabla G_j \equiv \frac{\partial G_j}{\partial \overline{X}_D}, \qquad j = NCON_f + 1, NCON \quad (8.35)$$

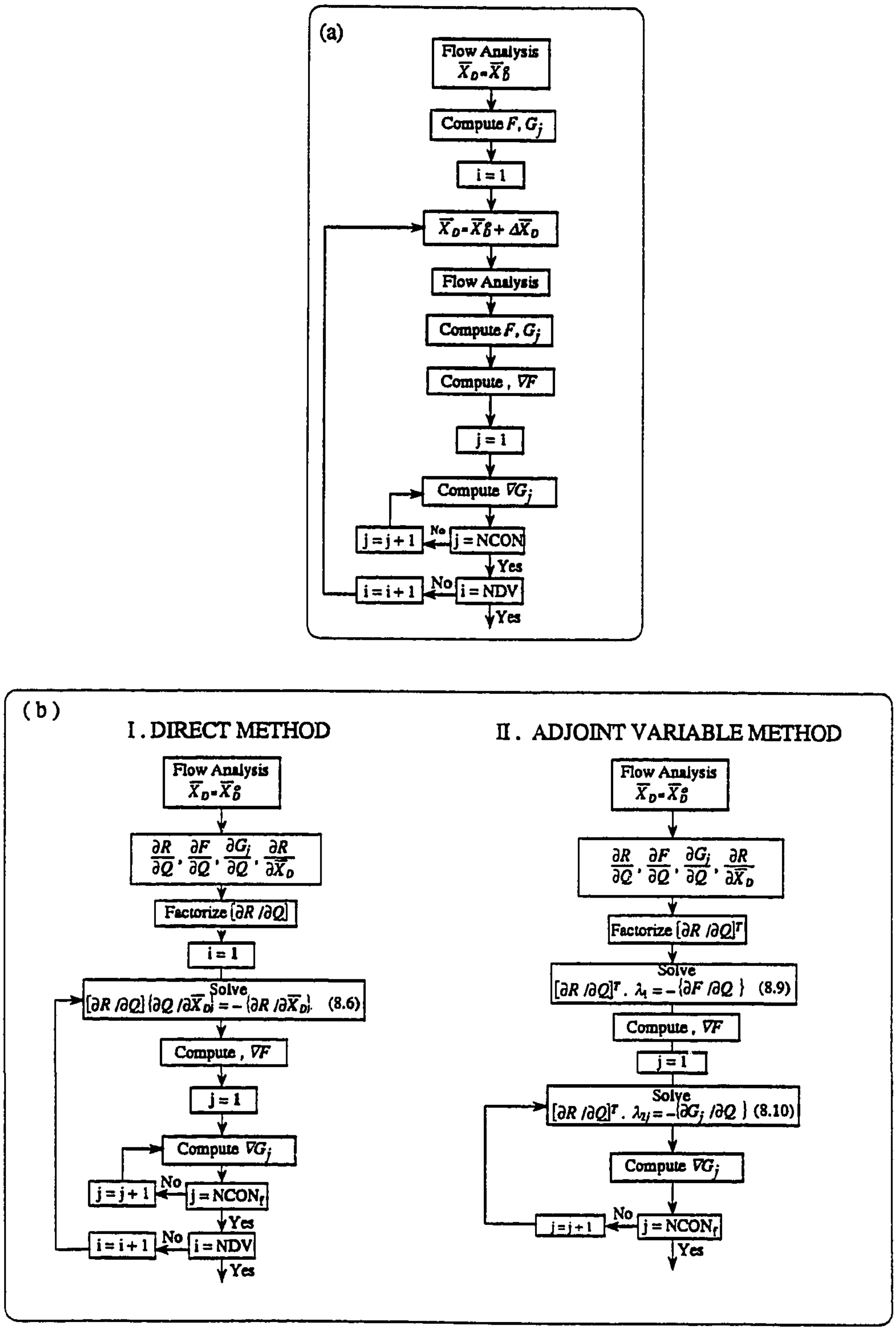

(a)
Flow Analysis $\bar{X}_D = \bar{X}_D^o$
Compute F, G_j
i = 1
$\bar{X}_D = \bar{X}_D^o + \Delta\bar{X}_D$
Flow Analysis
Compute F, G_j
Compute, ∇F
j = 1
Compute ∇G_j
j = j + 1
No
j = NCON
Yes
i = i + 1
No
i = NDV
Yes
(b)
I. DIRECT METHOD
Flow Analysis $\bar{X}_D = \bar{X}_D^o$
$\frac{\partial R}{\partial Q}, \frac{\partial F}{\partial Q}, \frac{\partial G_j}{\partial Q}, \frac{\partial R}{\partial \bar{X}_D}$
Factorize $[\partial R/\partial Q]$
i = 1
Solve
$[\partial R/\partial Q]\{\partial Q/\partial \bar{X}_{Di}\} = -\{\partial R/\partial \bar{X}_{Di}\}$ (8.6)
Compute, ∇F
j = 1
Compute ∇G_j
j = j + 1
No
j = NCON$_f$
Yes
i = i + 1
No
i = NDV
Yes
II. ADJOINT VARIABLE METHOD
Flow Analysis $\bar{X}_D = \bar{X}_D^o$
$\frac{\partial R}{\partial Q}, \frac{\partial F}{\partial Q}, \frac{\partial G_j}{\partial Q}, \frac{\partial R}{\partial \bar{X}_D}$
Factorize $[\partial R/\partial Q]^T$
Solve
$[\partial R/\partial Q]^T . \lambda_1 = -\{\partial F/\partial Q\}$ (8.9)
Compute, ∇F
j = 1
Solve
$[\partial R/\partial Q]^T . \lambda_{2j} = -\{\partial G_j/\partial Q\}$ (8.10)
Compute ∇G_j
j = j + 1
No
j = NCON$_f$
Yes

Fig. 8.1 (a) Details of the finite-difference approach
(b) Details of the sensitivity analysis approach.

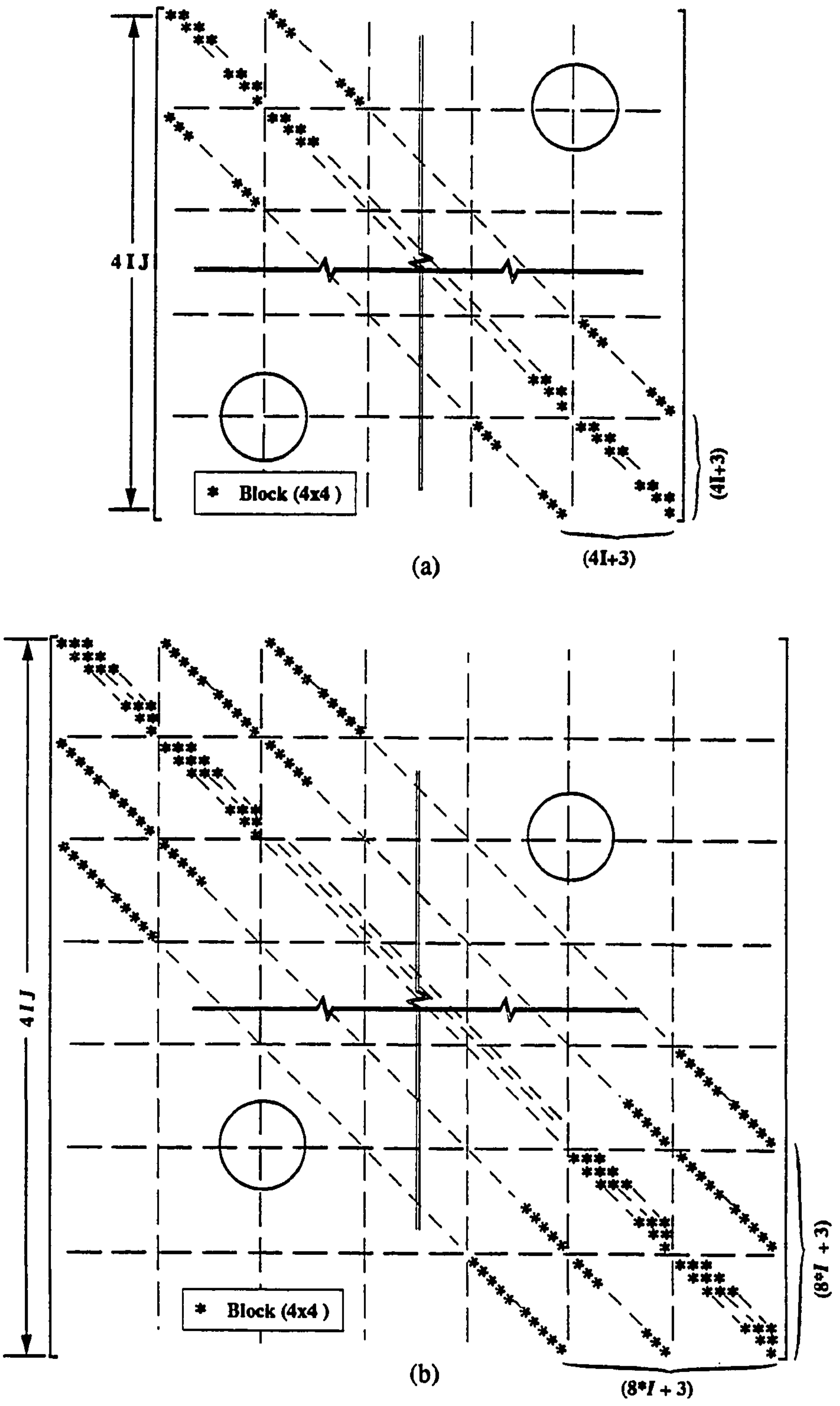


Fig. 8.2 The structure of the coefficient matrix $[\partial R/\partial Q]$ with the unknowns ordered in the normal direction: (a) first-order discretization (b) third-order discretization.

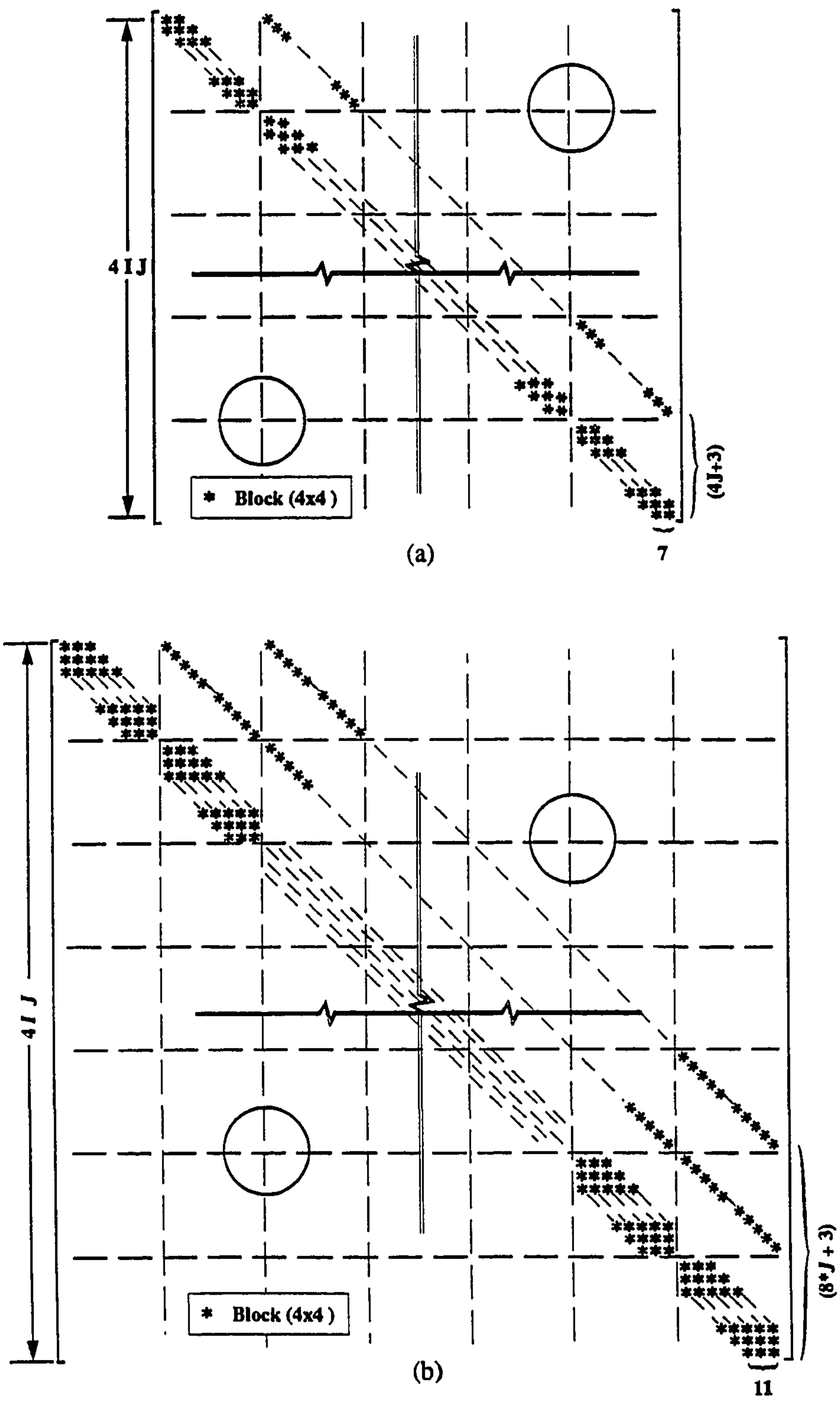


Fig. 8.3 The structure of the coefficient matrix $[\partial R/\partial Q]$ with the unknowns ordered in the streamwise direction: (a) first-order discretization (b) third-order discretization.

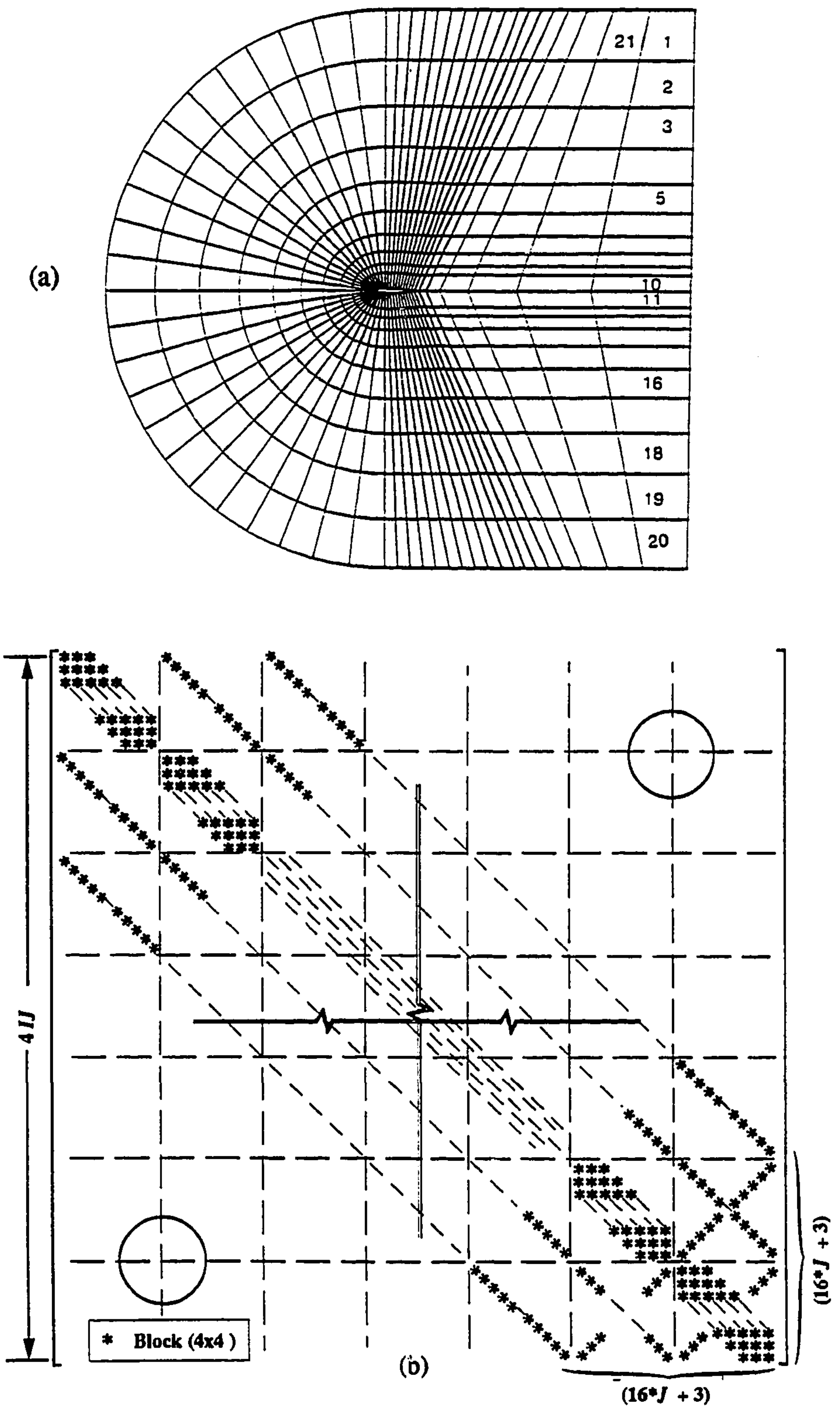


Fig. 8.4 (a) A coarse C-grid representing Type *A* unknown ordering
(b) The structure of the coefficient matrix [$\partial R/\partial Q$] corresponding to Type A unknown ordering.

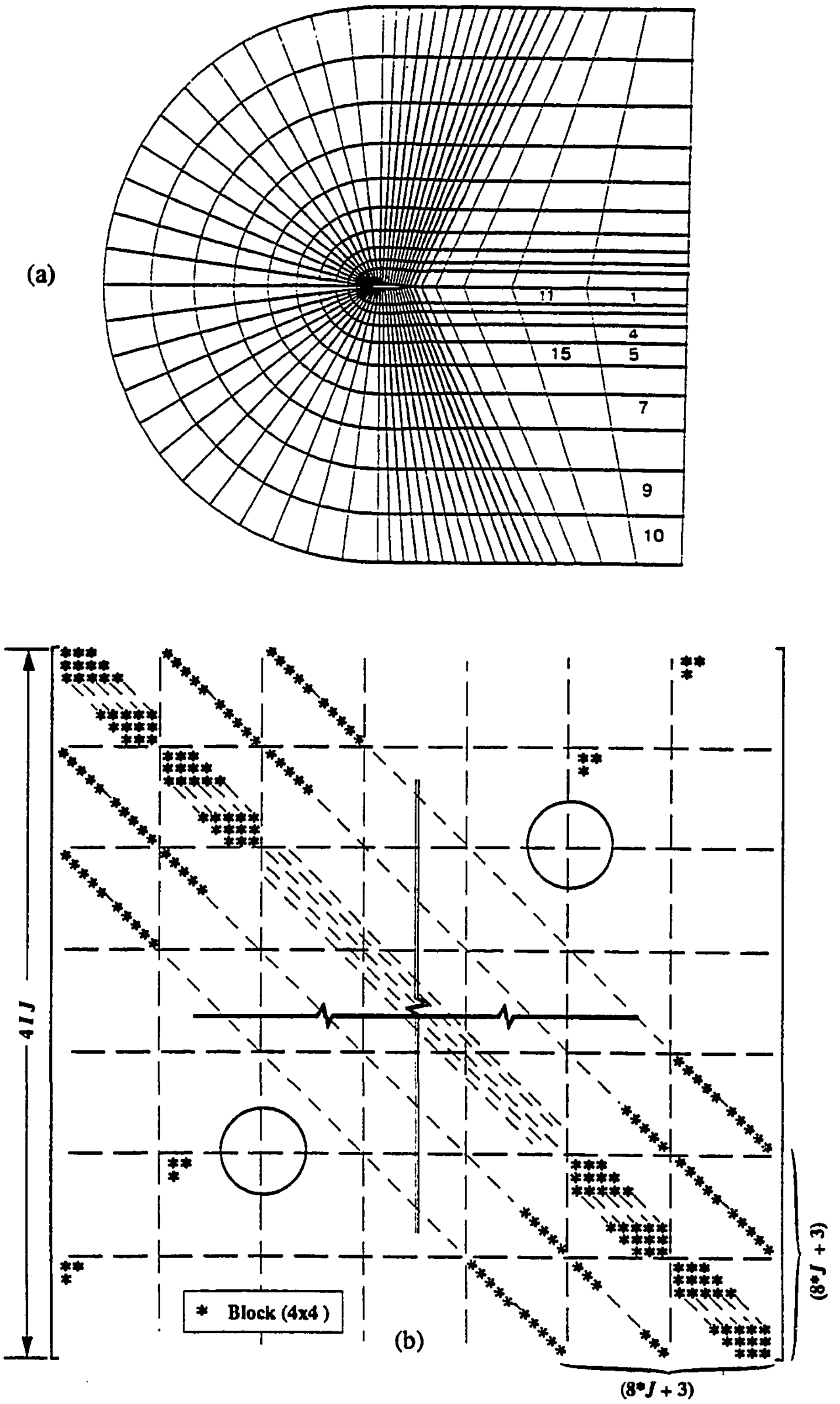


Fig. 8.5 (a) A coarse C-grid representing Type B unknown ordering
(b) The structure of the coefficient matrix $[\partial R/\partial Q]$ corresponding to Type B unknown ordering.

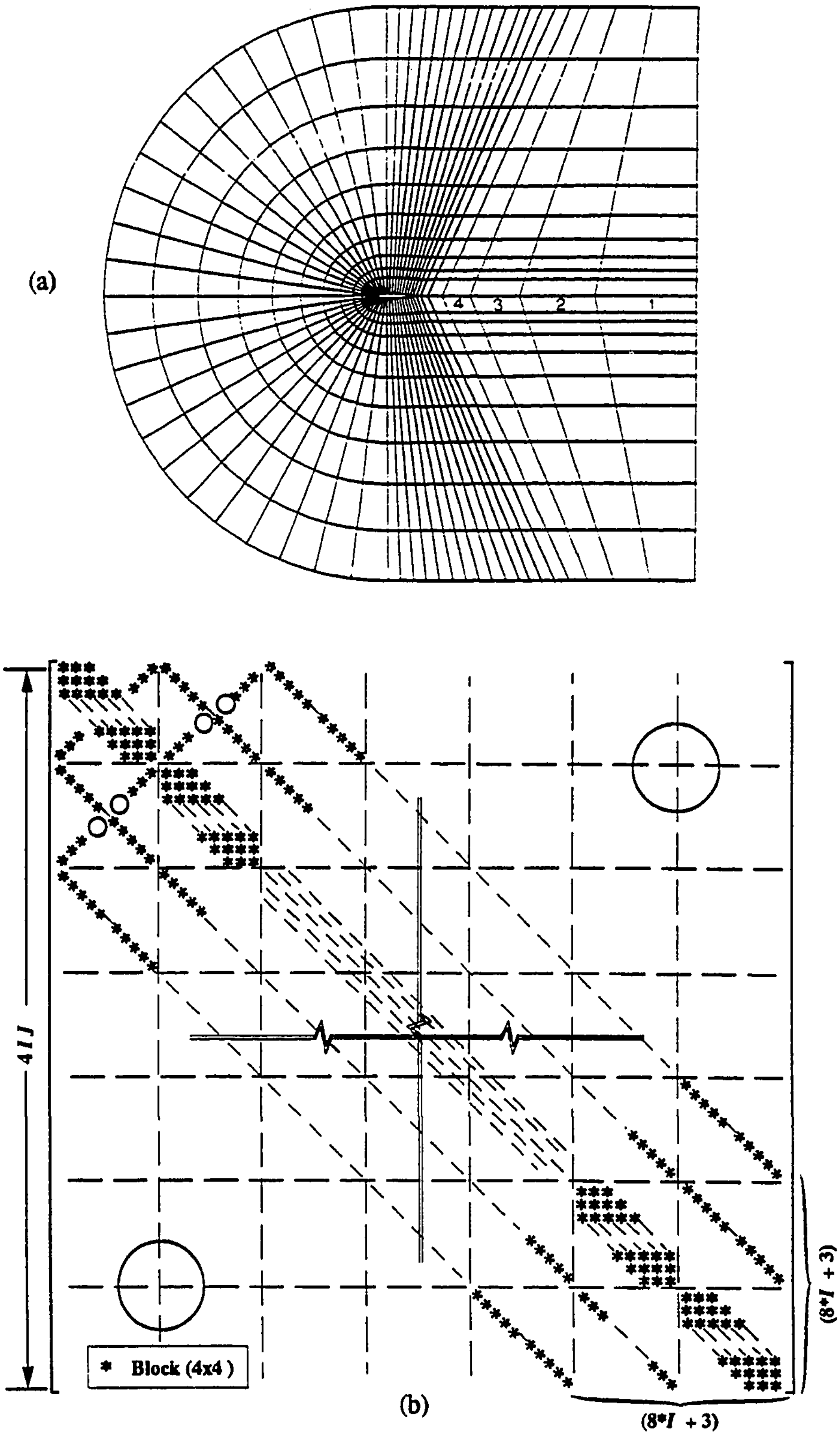


Fig. 8.6 (a) A coarse C-grid representing Type C unknown ordering
(b) The structure of the coefficient matrix $[\partial R/\partial Q]$ corresponding to Type C unknown ordering.

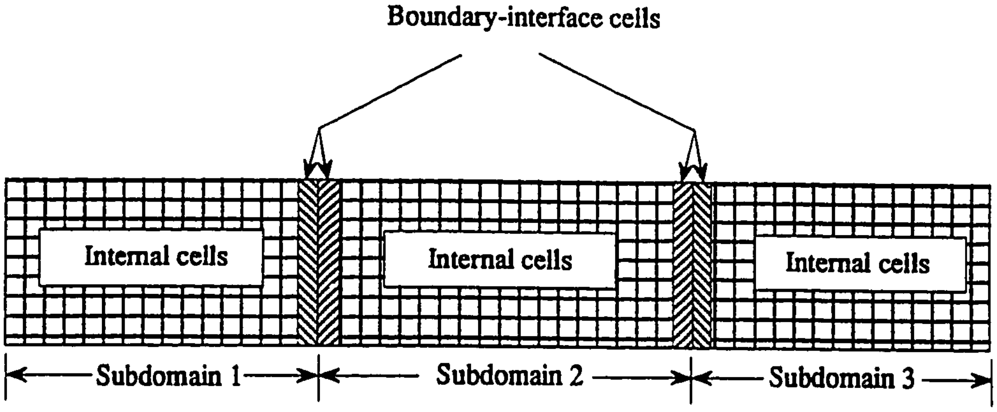
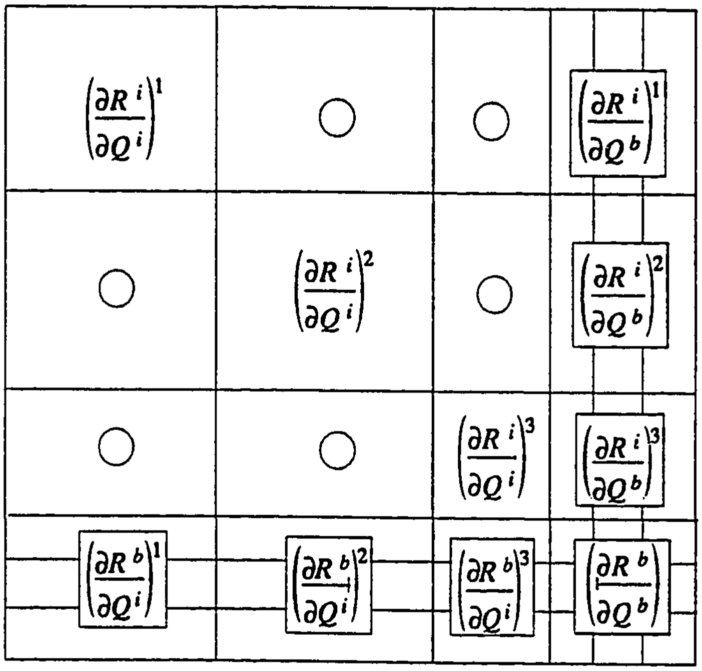


Fig. 8.7 Schematic showing a decomposed computational domain.



$$\begin{bmatrix} \left(\frac{\partial R^i}{\partial Q^i}\right)^1 & O & O & \left(\frac{\partial R^i}{\partial Q^b}\right)^1 \\ O & \left(\frac{\partial R^i}{\partial Q^i}\right)^2 & O & \left(\frac{\partial R^i}{\partial Q^b}\right)^2 \\ O & O & \left(\frac{\partial R^i}{\partial Q^i}\right)^3 & \left(\frac{\partial R^i}{\partial Q^b}\right)^3 \\ \left(\frac{\partial R^b}{\partial Q^i}\right)^1 & \left(\frac{\partial R^b}{\partial Q^i}\right)^2 & \left(\frac{\partial R^b}{\partial Q^i}\right)^3 & \left(\frac{\partial R^b}{\partial Q^b}\right) \end{bmatrix}$$

Fig. 8.8 Schematic showing the global coefficient matrix of the *SADD* scheme.

Chapter 9

AERODYNAMIC SHAPE OPTIMIZATION PROCEDURE

The overall process of aerodynamic design optimization consists of the following major components: CFD analysis, the sensitivity analysis, the approximate flow analysis, and the optimization method (Fig. 9.1).

9.1 CFD Analysis

The flowfield simulations are performed using the general purpose, finite-volume Euler/Navier-Stokes CFD code "VUMXZ3". The formulation given in Chapter 6 is implemented in this code. This code has been applied to analyze a variety of complex internal and external flows [72]. It produces consistent and repeatable flow simulations in the sense that small perturbations to design variables are accurately reflected in the flowfield solution. The CFD analysis is performed in any one of the following situations: (1) at the end of each optimization iteration to check the design; (2) when the number of approximate analyses reaches its specified limit as will be seen later; and (3) when new gradients of constraints and objective function are needed, i.e., when the search direction changes.

9.2 Sensitivity Analysis

A key part of the present design procedure is the sensitivity analysis described in Chapter 8, where the derivatives of the constraints, the objective function and the conserved flow variables, with respect to the design variables are computed. The derivatives quantify the effects of each design variable on the design and thereby identify the most important design changes to make en route to the optimum design. Hence, their accuracies are important for the success of the optimization process. This component constitutes the first difference between the present procedure and the conventional procedures.

9.3 Approximate Flow Analysis

In the present study, the *flow prediction method*, which is also given in Chapter 8, is used only through the one-dimensional search of the optimization process to obtain an approximate analysis. This component is crucial for the efficiency of the optimization process and constitutes the second difference between this procedure and the conventional procedures, as will be seen in the next section.

9.4 Optimization Algorithm

In choosing a design procedure for solving general aerodynamic shape optimization problems, several features are desirable. Firstly, the number of times the flowfield analysis must be performed should be kept as low as possible since the analysis is computationally expensive. This is particularly true when the number of flowfield governing equations is large. Secondly, the amount of gradient information required in the design process should be reduced or, if possible, eliminated. A wide variety of optimization algorithms, which satisfy the second feature, are available in general optimization software packages. Of interest here is the so-called *method of feasible directions* modified by Vanderplaats and Moses [100] and implemented in the general purpose optimization routine "ADS" [101]. This method has the features that it progresses rapidly to an optimum design and that it requires the gradient information for the objective and constraints active at only a given point in the optimization process.

When using the optimization routine (ADS) as a "black box" in a design procedure, it will require a handful of evaluations of the objective function and constraints, before the optimal design is reached. Hence, it is desirable to modify this optimization routine (ADS) such that the new procedure will have the capability of providing the objective function and constraints without resorting to the expensive CFD analysis. This capability is the so-called *flow prediction method* and is described in details in Chapter 8. In this section, limitations of this method when incorporated with the method of feasible directions, are discussed. Also, a brief review of the method of feasible directions is given in this section.

9.4.1 New Procedure for The Method of Feasible Directions

When using the method of feasible directions to solve the nonlinear constrained optimization problem defined by Eqs. (7.2) to (7.5), i.e., finding the vector $\overline{X}_D$, the optimization process proceeds iteratively by the common update formula,

$$\overline{X}_D^{r+1} = \overline{X}_D^{r} + \alpha^* S^r \tag{9.1}$$

In Eq. (9.1), $\overline{X}_D^r$ are the current aerodynamic shape design variables, $\overline{X}_D^{r+1}$ are the new aerodynamic shape design variables, $\alpha^* > 0$ is the move parameter, and S^r is a vector of search direction. Thus, $\alpha^* S^r$ is the step size for $\overline{X}_D$. The number of optimization iterations is r, where r=0 for the initial aerodynamic shape. It is obvious from Eq. (9.1) that for an iterative optimization approach there are only two parameters, α^* and S^r, that need to be calculated for each optimization iteration.

In general, one optimization iteration of the optimization process consists of the following eight stages:

(1) start with an initial vector of design variables, $\overline{X}_D^0$, i.e., initial geometry.

(2) perform a CFD analysis to get the flowfield solution, Q.

(3) calculate all information related to the current design such as the objective function, constraints and their gradient. Normalize the gradients.

(4) check the convergence criterion; continue, if not satisfied, terminate otherwise.

(5) determine a search direction, S^r, which is in the "usable-feasible" sector of the design space using the method of feasible directions [100].

(6) perform a one-dimensional search along the direction of S^r to evaluate the move parameter, α^*, which reduces the objective function as much as possible subject to the constraints.

(7) update X_D^{r+1} using Eq.(9.1) and generate a new computational grid accordingly.

(8) Increment r and return to Stage 1.

This procedure is implemented in the general purpose routine "*ADS*" and it is given in Fig. 9.2 for a "black box" use. Stages 1, 2, 7, and 8 are self explanatory. In Stage 3 the gradient information are either provided through an external routine or computed by a finite-difference approximation through "*ADS*". The convergence criteria used for the optimization algorithm, i.e., Stage 4 are: (1) the maximum number of iterations; (2) prescribing some values for the absolute or the relative change in the objective function. Stage 5 of finding "the usable-feasible" search direction by the method of feasible directions is explained in Ref. [100]. Stage 6 represents a major drawback when using the above procedure as a "black box" and constitutes one of the differences between the new and the conventional optimization procedures. Hence, this stage will be explained in more detail in the following subsection to show this difference and to elucidate this drawback.

9.4.2 A Modified One-dimensional Search Cycle

Once a search direction has been found, the move parameter, α^*, should be chosen such that it minimizes the objective function in this direction. This problem is called the *one-dimensional search* since the only variable is α. This is commonly done by one of the following methods: a polynomial interpolation, a golden section method, or a combination of them.

For each of these methods the objective function F is calculated for several values of α in order to find α^*. Hence, the optimization process will be expensive if actual flowfield analysis is performed in each cycle of the one-dimensional search. In other words, if the optimization routine is used as a "black box", the cost of the optimization process will be prohibitive. Fortunately, the changes in $\overline{X_D}$ during the one-dimensional search cycle are sometimes small. Hence, the present optimization procedure takes advantage of this situation and replaces the CFD analysis during the one-dimensional search cycle, where the changes of $\overline{X_D}$ are small, by the flow prediction method of Chapter 8.

Although this approach may seem attractive, it has two limitations. The first limitation is due to the fact that sometimes large values of $\overline{X_D}$ may cause a failure of the approximate flowfield analysis. Usually, this situation occurs when the first estimate for α^* is large. As a remedy for this problem, the present procedure limits the first estimate of α^* to a small value such that the changes in $\overline{X_D}$ are kept within 2 % of their normalized values. The second limitation is due to the fact that after using the approximate analysis successively for different values of α, the approximate flowfield solutions start to deteriorate giving crude estimates for the values of the objective function. This usually occurs if the number of cycles required for the one-dimensional search exceeds a certain limit. As a remedy for this problem, the present procedure restricts the number of one-dimensional search cycles, within which the successive approximate analysis is performed, to 15. Hence, if the value of the move parameter, α^*, is not reached within 15 cycles, one CFD analysis is performed to update the flowfield solution and to provide a new baseline solution for the approximate flowfield analysis.

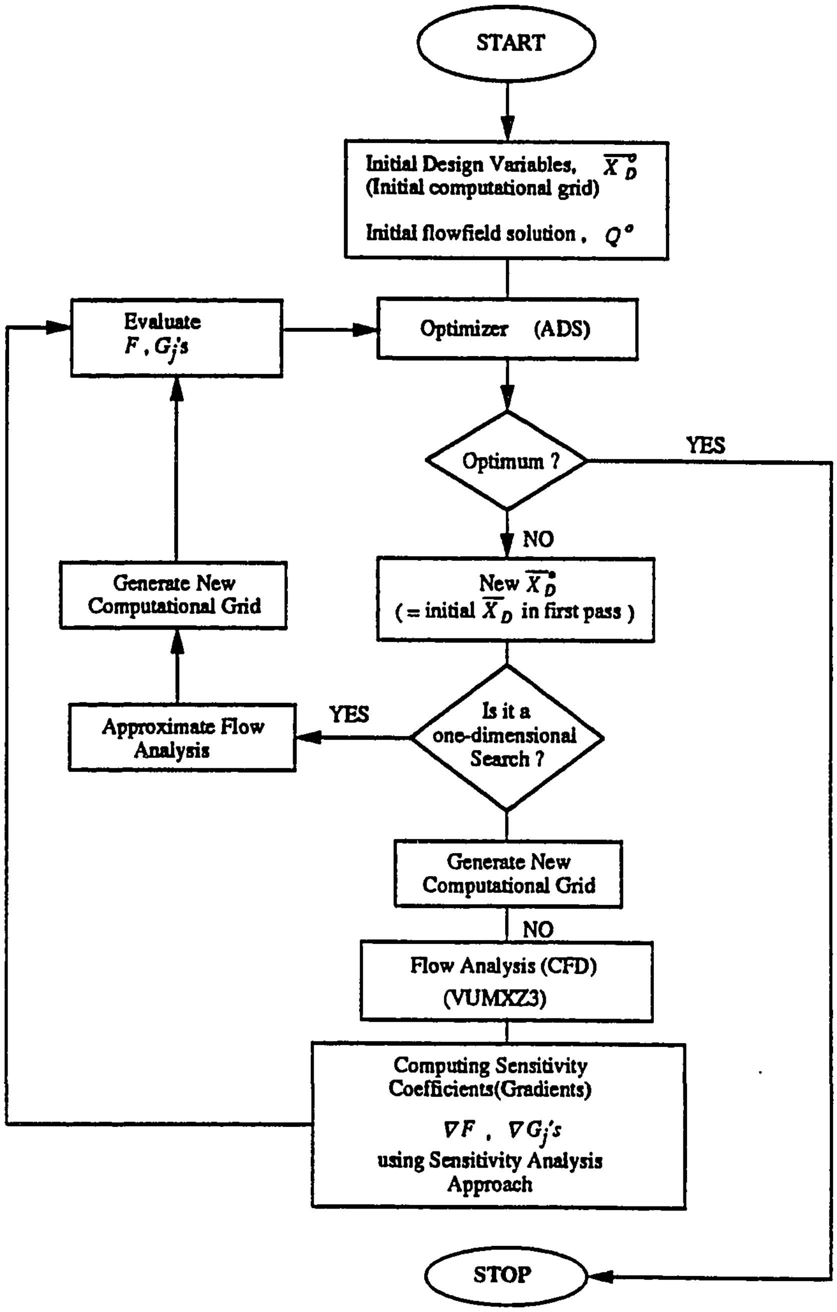

START
Initial Design Variables, $\overline{X}_D^0$
(Initial computational grid)
Initial flowfield solution, Q^0
Evaluate
F, G_j's
Optimizer (ADS)
Optimum ?
YES
NO
Generate New Computational Grid
New $\overline{X}_D^*$
(= initial $\overline{X}_D$ in first pass)
Approximate Flow Analysis
YES
Is it a one-dimensional Search ?
Generate New Computational Grid
NO
Flow Analysis (CFD)
(VUMXZ3)
Computing Sensitivity Coefficients(Gradients)
∇F , ∇G_j's
using Sensitivity Analysis Approach
STOP

Fig. 9.1 Flow chart of the new aerodynamic design optimization procedure.

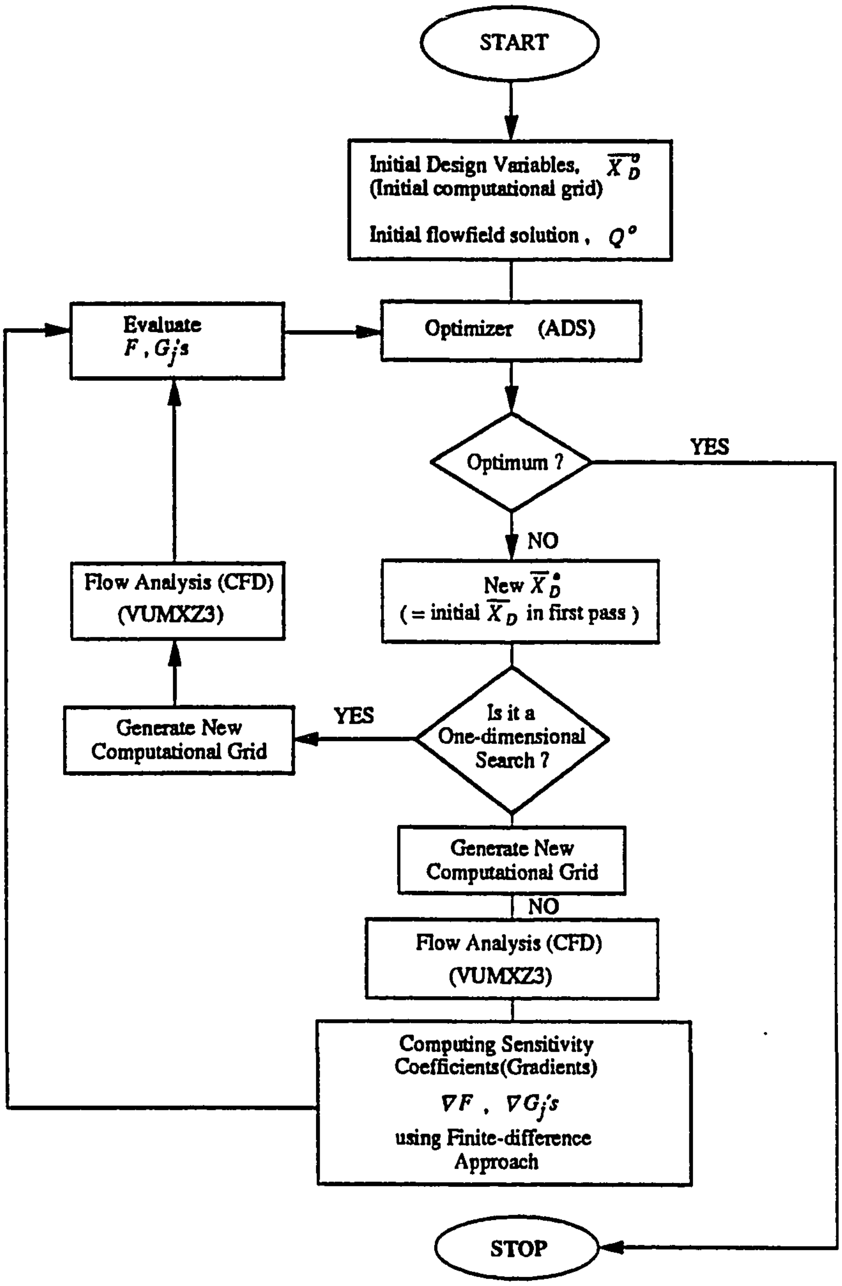


Fig. 9.2 Flow chart of the "black box" use of "ADS".

Chapter 10

RESULTS AND DISCUSSION

As mentioned in Chapter 5, the development of the new aerodynamic optimization procedure is achieved in four steps. The results of each step are presented and discussed in a separate section herein. Presented in Section 10.1 is an assessment of the numerical methods (including the *SADD* scheme) that are used to solve the system of linear equations of the analytical approach. Section 10.2 covers a series of computations used to assess the accuracy and feasibility of the flowfield prediction method in different flow regimes. A comparison between the different methods that are used to calculate the aerodynamic sensitivity coefficients is presented in Section 10.3. Finally, some demonstrations showing the design capability of the new optimization procedure are given in Section 10.4.

The upstream flow conditions for the demonstrative cases involving the nozzle-afterbody configuration (Fig. 2.2) are given in Table 10.1. These conditions are kept constant during an optimization process. The height of the combustor exit (H) is 0.01524 m and all the solid walls are assumed to be adiabatic.

Table 10.1 Upstream flow conditions for the nozzle-afterbody configuration.

Flow	Mach No.	Reynolds No. based on (H)	Total temp. (°K)	Total pressure (kPa)
Combustor exit (*th*)	1.665	340,000	338.0	189.6
External Flow (free stream)	6.000	330,000	491.3	2489.0

The upstream flow conditions for the demonstrative cases, in which the NACA 0012 airfoil is used as a baseline configuration, are 0.8 Mach number, 0.5 million Reynolds number based on the chord length, and zero angle of attack.

10.1 An Assessment of the Solution Methods for The System of Equations of The Analytical Approach

10.1.1 Results on Single Grids

In the present study, only the direct and iterative solvers used for the system of linear algebraic equations (Eq. (8.6)), which arise in the analytical approach, are considered. As mentioned in Section 8.4, only two direct solvers are deemed most suitable for the present study. These are the sparse matrix solver and the banded matrix solver. On the other hand, only one iterative solver is deemed most applicable to the system of linear equations of the present study. This solver is the generalized minimum residual (*GMRES*) solver. In this section, efficiencies, accuracies and storage requirements of these solvers are assessed. For this purpose, the configuration shown in Fig. 7.4 is considered, where the design variables X_{D1} and X_{D2} are given the values of 20° and 12°, respectively. The computations are performed using a second-order discretization of the flowfield governing equations and using a grid (Fig. 10.1) with (52x40) cells. Consequently, the system of linear equations has a square coefficient matrix with 8320 rows and, at most, 36 nonzero elements per row.

A comparison of the sparse matrix and the banded matrix solvers, as applied to the above system of equations, reveals that they both require 43 Megabytes of memory. However, unlike for the banded matrix solver, the appropriate memory allocation for the sparse matrix solver can only be determined after many trials. This may be explained as follows. When using the sparse matrix solver, the coefficient matrix of Eq. (8.6) is factorized in two stages, first symbolically and then numerically. Symbolic factorization is a factorization of only the nonzero structure of the coefficient matrix whereas numeric factorization is the standard *"LU"* factorization. Despite the fact that only the values and locations of the nonzero elements are stored when using the sparse matrix solver, additional memory allocations are also required for the fill-in elements (elements which result from zero entry of the coefficient matrix becoming a nonzero in the L or U as a consequence of the elimination process). Since the required number of fill-in is not known a priori, different numbers of fill-in and, consequently, its attendant memory allocations must be attempted in order to determine the most efficient one.

Comparing the results of the two direct solvers as used in the present study indicates that their numerical accuracies are the same. Whereas, a comparison of the computational times required by the two direct solvers reveals that the sparse matrix solver is significantly less efficient. For instance, the sparse matrix solver requires about 212 CPU seconds on the *CRAY-2* computer of the Numerical Aerodynamic Simulation (*NAS*) of *NASA* to solve the above system of equations. The banded matrix solver requires only 23 CPU seconds for the same system. Such a large difference in computational time is due to the fact that most of the computational time of the sparse matrix solver is spent in performing the symbolic factorization (199 CPU seconds for the present case). Usually, performing the symbolic factorization is considered a computational advantage if different systems of equations with the same nonzero structure of the coefficient matrices are solved. However, this is not the case in the present study since the same nonzero structure of the coefficient matrix is not always guaranteed. It should be noted that the time required by the sparse matrix solver for the numerical factorization and the forward and backward substitutions is less than that of the banded matrix solver (13 CPU seconds compared to 23 CPU seconds).

The *GMRES* solver implemented in the software package *SITRSOL* [102] is employed to solve the system of equations given by Eq. (8.6). Since preconditioning of the coefficient matrix of this system is essential for the success of the *GMRES* solver, three types of preconditioners implemented also in this software package are tried. These are the diagonal (Jacobi) preconditioner, the Incomplete *LU* factorization *(ILU(*κ*))*, where κ is the level of fill-in, and the truncated Neuman polynomial expansion. These preconditioners can be employed either as a left or as a right preconditioner. The computer core memory of this solver is found to be 24 Megabytes when no preconditioner is employed. However, depending upon the utilized preconditioner, additional memory allocations may be required.

As mentioned in Section 8.4.2, the choice of an effective and stable preconditioner is extremely important to the success of the *GMRES* solver. Since the best of the above three preconditioners is not known a priori, each one of them needs to be tried once as a left preconditioner and once as a right preconditioner. It is found that only the *ILU(1)* preconditioner succeeds to solve the present system when it is used as a left preconditioner. In this case, the total computer memory required for this solver is increased to 40 Megabytes, which is almost the same as that required for the direct solvers.

Usually a convergence criterion has to be met in the iterative solvers in order to declare the system of equations as solved, e.g., when the residual reaches a tolerance of 10^{-14}. However, this does not mean that the exact solution is reached by the solver. Hence, the solution of the banded matrix solver is considered in the present study as a baseline solution to check the accuracy of the final solution of the *GMRES* solver. In other words, to check whether an exact solution is reached or not, the following L_2–norm is defined:

$$L_2\text{–norm} = \sqrt{\sum_{m=1}^{m=NEQ}\left(\left(\frac{\partial Q^m}{\partial X_{D1}}\right)_G - \left(\frac{\partial Q^m}{\partial X_{D1}}\right)_B\right)^2} \qquad (10.1)$$

where subscripts G and B denote the *GMRES* solver and the banded matrix solver, respectively. *NEQ* denotes the number of unknowns. The L_2–norms given in Table 10.2

are obtained when using the *ILU* (1) left preconditioner. As can be seen from this table, the L_2–norm is of the same order as the tolerance criterion up to a tolerance of 10^{-8} where it starts to level off. Therefore, the system of equations is declared solved in the present study when a tolerance of 10^{-9} is reached.

Table 10.2 An assessment of the *GMRES* solver when using different tolerance criteria.

Tolerance criterion	Number of iterations	Execution time for only the iterative procedure[a,b] (CPU seconds)	L_2-norm
10^{-04}	14	2.1700[c]	$2.3457*10^{-04}$
10^{-05}	16	2.2240	$2.8291*10^{-05}$
10^{-06}	18	2.7864	$2.6592*10^{-06}$
10^{-07}	20	3.1440	$2.7893*10^{-07}$
10^{-08}	22	3.2260	$2.1717*10^{-08}$
10^{-09}	24	3.7710	$1.1207*10^{-08}$
10^{-10}	26	3.9760	$1.1010*10^{-08}$
10^{-11}	28	4.1860	$1.0980*10^{-08}$
10^{-12}	29	4.9732	$1.0976*10^{-08}$
10^{-15}	36	5.8220	$1.0976*10^{-08}$
10^{-20}	47	6.4030	$1.0976*10^{-08}$

[a] The time required for each right hand side.
[b] The time required for the *ILU* is about seven CPU seconds and is constant in all computations.
[c] The number of unknowns is 8,320.

The computational time of the *GMRES* solver as employed in the present study is divided into two parts– the time for Incomplete *LU* factorization and the time for the iterative procedure. The time for *ILU* does not change if the same coefficient matrix is used with different right hand sides, which is the case herein. On the other hand, the time for the iterative procedure changes according to the desired accuracy of the solution as indicated in Table 10.2. For instance, the iterative procedure requires about four CPU seconds for each right hand side when a tolerance of 10^{-11} is specified and it requires 2.78 CPU seconds when a tolerance of 10^{-6} is specified. This means that as the tolerance criterion increases, the time for the iterative procedure increases. However, unlike the L_2–norm, the time required for the iterative procedure consistently increases as the tolerance criterion decreases.

To summarize the above findings, the execution times and the memory requirements of the above solvers are compared in Table 10.3. All execution times givn in this table are normalized by that required for the *LU* factorization in the banded matrix solver which is 22 CPU seconds. Also, all memory requirements are normalized by that required for the banded matrix solver which, as mentioned above, is about 43 Megabytes. When employing the *GMRĖS* solver, a tolerance criterion of 10^{-10} is used.

Table 10.3 A Comparison of the different solvers applicable to the system of linear equations Eq.(8.6).

Solver	Execution time		Memory
	Factorization	Each *RHS*	requirement
Banded matrix solver	1.00	0.022	1.00
Sparse matrix solver	9.59	0.022	1.00
GMRES solver (using left *ILU(1)* preconditioner)	0.318	0.180	0.93

The number of unknowns is 8,320.

It can easily be concluded from Table 10.3 that the banded matrix solver is the most feasible one when the number of right hand sides is large. Hence, it will be used to solve the systems of equations in the *SADD* scheme (Eqs. (8.20), (8.25), and (8.26)).

10.1.2 Results on Domain Decomposed Grids

To assess the computational advantages of the *SADD* scheme, a computer program called *MAisSA* (Multiblock Aerodynamic Sensitivity Analysis) is developed. Three cases (Cases 5.a, 5.b, and 5.c) are considered. These cases differ basically in the number of subdomains into which the computational domain is divided, and consequently, in the number of boundary-interface cells. The configuration of Fig. 7.4 is considered for all three cases with a global computational domain of size (52x40) cells (Fig. 10.1). The X_{D1} and X_{D2} are given the values of 20° and 12°, respectively. The distinguishing features of these cases are given for reference in Table 10.4.

Table 10.4. The distinguishing features of the cases used to demonstrate the flow prediction.

Case	No. of subdomains	No of interfaces	Size of each subdomain (cells)
5.a	1	0	52x40 (Single domain)
5.b	2	1	20x40 32x40
5.c	4	3	20x20 20x20 32x20 32x20

The global computational grid consists of 52x40 cells.

In Case 5.a, the computational domain consists of one block. Hence, there are no boundary-interface cells. This is essentially similar to the previous calculations on a single grid. In Case 5.b, the computational domain is divided into two subdomains as shown in Fig. 10.2a. As can be seen from this figure, the number of boundary-interface cells in this case are 80, and as a result, the number of decoupling unknowns is 320. In Case 5.c, the computational domain is divided into four subdomains (Fig. 10.3a). As a result, the number of boundary-interface cells becomes 142, which means that the number of decoupling unknowns in Case 5.c is 568. Structures of the global coefficient matrices of Case 5.b and Case 5.c are given in Figs. 10.2b and 10.3b, respectively.

The effect of varying the number of decoupling unknowns on the execution time and the memory requirements is investigated and typical results are presented in Table 10.5. All execution times of this table are normalized by the time required for Case 5.a which is 23 CPU seconds. All memory requirements are normalized by the memory allocations required for Case 5.a which is 56 Megabytes.

Table 10.5. Effect of number of decoupling unknowns on the execution time and memory requirements.

Case	No. of decoupling unknowns	Execution time	Memory requirement[a]	L_2-norm
5.a	0	1.0	1.000	0
5.b	320	2.1	0.817	$0.12*10^{-16}$
5.c	568	2.8	0.588	$1.60*10^{-16}$

[a] The global computational grid consists of 52x40 cells.

As can be seen from the L_2-norms in Table 10.5, the numerical accuracies of all the cases are the same. However, their execution times and memory requirements are significantly different. In general, as the number of decoupling unknowns increases due to the increase in the number of subdomains, the memory requirements decrease and the execution times increase.

The memory requirement is reduced by more than 40% in Case 5.c (four subdomains) and by 19% in Case 5.b (two subdomains). The reason for this can easily be seen by the inspecing Table 10.4, which shows that the size of the largest subdomain in Case 5.c is about 30% of the global grid and is about 60% of the global grid in Case 5.b. Hence, it can be concluded that the savings in memory requirements are strongly related to the size of the largest subdomain. Therefore, for large 2-D problems and for all 3-D problems, where the size of the largest subdomain can be kept small, the savings in memory requirements are expected to be more pronounced. Thus, using the *SADD* scheme in large 2-D as well as 3-D problems should make it possible to solve these problems easily on the supercomputers available today.

In spite of the increase in the execution times of Case 5.b and Case 5.c, the rate of the increase decreases as the number of decoupling unknowns increases. For instance, in Case 5.b (320 decoupling unknowns), the execution time is increased by 100% of the time required for Case 1.a (zero subdomain or zero decoupling unknowns). Whereas, in Case 5.c (568 decoupling unknowns), the execution time is increased only by 40% of the time required for Case 5.b. It should be noted that the size of subdomains used in the present study is relatively small. It is believed that the decrease in the above rates becomes more pronounced for subdomains with larger sizes. However, the drawback of increasing the execution time is generally not as significant to offset the well-known advantages of the domain-decomposition techniques; these techniques subdivide a large problem into small manageable ones and ease the grid generation for a complex-geometry problem [72-74].

10.2 Demonstration of Flowfield Prediction Method

Three series of computations (Cases 6-8) are included in this section to investigate the behavior of the flowfield prediction method in different flow regimes, and to attempt to answer the issues raised in Section 8.3. In Cases 6 and 7, the simplified model of the nozzle-afterbody section (Fig. 7.4) is considered and the flow is assumed to be inviscid. These cases demonstrate the behavior of the method in the supersonic and hypersonic flow regimes. In Case 8, a NACA 0012 airfoil is considered (Fig. 7.3) and viscous effects are included by using the thin-layer Navier-Stokes equations (Eq. 6.1). This case serves to demonstrate the behavior of the flowfield prediction method in the transonic flow regime, which is highly nonlinear.

The main reasons behind Case 6 are to compare the results of the flowfield prediction method with the available experimental data and to assess the behavior of the method in the absence of shocks. In Cases 6.a and 6.b, the flowfield solutions for two configurations, where one configuration is obtained by a small modification of the other, are obtained by forcing the discrete form of the flowfield governing equations, (Eq. (6.27), to zero at all computational cells using the flow analysis code. The baseline configuration has $X_{D1} = 18°$ and $X_{D2} = 12°$ (Case 6.a), while the modified configuration has $X_{D1} = 20°$ and $X_{D2} = 12°$ (Case 6.b). That is, the changes in X_{D1} and X_{D2} are 11.1% and 0%, respectively. Equation (8.17) is solved for the changes in Q (i.e., ΔQ) due to the above changes in $\overline{X}_D$. The Jacobian matrix and the right hand side of this equation are evaluated using the flowfield solution (Fig. 10.4a) associated with the configuration ($X_{D1} = 18°$ and $X_{D2} = 12°$). Then, the predicted flowfield solution associated with the configuration ($X_{D1} = 20°$ and $X_{D2} = 12°$) is obtained by using Eq. (8.16). This is referred to as Case 6.c.

The normalized pressure contours obtained via the flow prediction technique (Case 6.c) are compared favorably with their corresponding contours (Case 6.b) obtained by the flow analysis code (Fig.10.4b). It is also seen from this figure that the axial thrust force coefficient, F, obtained using the flow prediction results differs only by 0.08% from that obtained using the flow analysis solution. The surface pressure coefficient (Cp) distributions on the ramp obtained from the predicted flowfield solution are compared in Fig. 10.5 with their corresponding values obtained via the flowfield analysis code and the experimental data of Cubbage and Monta [5]. Examination of this figure indicates that the numerical results agree with each other as well as with the experimental data. The slight discrepancy with the data is attributed to viscous effects, which are not accounted for in the governing equation of the flow (Eq. (6.27)). The results shown in Figs. 10.4 and 10.5

demonstrate the accuracy of predicting the flowfield solution when no shocks exist in the flowfield and also when the design variable, X_{D1}, is changed by 11.1%.

Next, the second series of computations (Case 7) is included to investigate the quality and accuracy of the flowfield solution in the presence of shocks as well as when different step sizes of design variables are used. In addition, these cases help assessing the accuracy of the flowfield solution when the technique of *prediction based on prediction* (discussed in Section 8.3) is used. The distinguishing features of Case 7 are presented in Table 10.6.

Table 10.6. The distinguishing features of the cases used to demonstrate the *SADD* scheme.

Case	Deflection Angle, θ	Flowfield Solution Method
7.a	0°	CFD Analysis
7.b	2.5°, 5°, 10°	CFD Analysis
7.c	2.5°, 5°, 10°	Prediction based on CFD analysis of $\theta = 0°$
7.d	5°	Prediction based on CFD analysis of $\theta = 2.5°$
7.e	5°	Prediction based on Prediction of $\theta = 2.5°$

The flowfield governing equations (Eq. (6.27)) are solved to obtain the flowfield solution first for a flat ramp surface at $\alpha = 10°$ (Case 7.a). Then, the ramp surface is modified in such a way that a shock can be generated– a compression corner is formed at 38% length from the ramp corner by turning the surface at an angle, θ, from the ramp surface (Fig. 10.6). Therefore, the angle θ serves as a design variable in Case 7. CFD analyses are performed for three values of θ : 2.5°, 5°, and 10° (Case 7.b). Finally, the flowfields for the above θ values are predicted using the analysis for the flat ramp surface, that is, $\theta = 0°$ (Case 7.c). In other words, the $\theta = 0°$ configuration (Case 7.a) is denoted by $\overline{X}_D^o$ in Eqs. (8.16) and (8.17), and any one of the $\theta \neq 0°$ configurations (Case 7.c) is denoted by $\overline{X}_D^*$. The flow of Case 7.a over the ramp is free of shocks. However, flows containing compression shocks due to ramp wall deflection ($\theta \neq 0°$) are predicted based on the shock-free flow.

The pressure coefficient distributions along the ramp for the Cases 7.a, 7.b, and 7.c are shown in Fig. (10.7). The analysis (Case 7.b) and the predicted (Case 7.c) results are indistinguishable up to the compression corner. The compression corner shocks are also predicted very well. As expected, discrepancies begin to appear for the larger θ values, i.e., for larger design variable modifications. It should be noticed that a discontinuous physical phenomenon (shock) is predicted based on a flow which does not have that phenomenon (shock-free).

A more quantitative measure of the discrepancies can be provided by plotting the percentage of the deviation of the predicted *Cp* to the *Cp* value obtained from analysis (Fig.10.8). The maximum deviation is only 2% for θ = 2.5°, but it increases to 22% for θ= 10°. These deviations are attributed to the fact that the flowfield solution depends nonlinearly on the design variables. It should be noticed that the predicted flowfield solution in this study is approximated by a Taylor's series, Eq. (8.16), in which the gauge function is the powers of the small changes in design variables, $\left(\overline{X}_D^* - \overline{X}_D^o\right)$. Therefore, the above observed deviations may decrease if other gauge functions, e.g., logarithmic, are used in the Taylor's series. However, the deviations shown here are typical of trends usually observed when a nonlinear problems are locally linearized. Therefore, it can be concluded that the prediction method, due to truncation error, exhibits increasing inaccuracies as the step size, $\Delta\overline{X}_D$, increases and eventually produces unacceptable solutions when the modification becomes too large.

The success of the prediction method for the off-surface flow is assessed by examining the density contours of the internal nozzle jet flow. Shown in Fig. 10.9 are the density contours of the shock-free flow (Case 7.a). When θ is equal to 2.5°, a compression corner shock is created (Fig. 10.10). Since this shock is embedded in the broad ramp expansion, it effectively delays the ongoing ramp expansion spatially downstream. As seen from Fig. 10.10, the contours of the predicted flow (Case 7.c, $\theta = 2.5°$) match those of the analysis (Case 7.b, $\theta = 2.5°$) up to the shock, and start deviating only slightly downstream of the compression corner. Furthermore, the angle and the strength of the shock are very well predicted. This clearly indicates that the prediction method inherently contains and correctly models the characteristics wave propagation of supersonic flow. In other words, this demonstrates that it is possible to accurately represent the flow physics by the prediction model for small design variable modifications.

Another issue, which appeals to the curiosity is the success of the prediction method when an existing modification in a configuration is enlarged; for example, predicting the flow for $\theta = 5°$ when the flow of $\theta = 2.5°$ is given. Shown in Fig. 10.11 are three sets of density contours in comparison: the flow analysis of $\theta = 5°$ (Case 7.b), the flow predictions of $\theta = 5°$ based on the flow analysis of $\theta = 2.5°$ (Case 7.d), and the flow predictions of $\theta = 5°$ based on the predictions of $\theta = 2.5°$ (Case 7.e). Two points are noteworthy here. Firstly, Case 7.d aims at predicting the flow due to an enlarged modification (θ from 2.5° to 5°), but not predicting a new physical phenomenon; that is, the prediction method is given a shock with which it begins. The comparison of this case with the analysis is as satisfactory as the one observed in Fig. 10.10, despite the fact that the shock in Fig. 10.11 is stronger. Secondly, this figure illustrates the feasibility and quality of the flowfield prediction when it is based on another flowfield prediction. For example, the prediction of $\theta = 5°$ flowfield (Case 7.e) is based on the Case 7.c prediction for $\theta = 2.5°$, which is based on the flowfield analysis of Case 7.a. Since the truncation error of Eq. (8.16) occurs twice and progressively during this process, although associated with smaller modifications, the agreement of Case 7.e is slightly less successful than that of Case 7.d. Therefore, it may be concluded that higher level predictions may be further used for crude, but efficient estimates of flowfield solutions of highly modified shapes.

Finally, the accuracy of the flowfield prediction method in case of viscous transonic flows is assessed. A series of computations (Case 8) of Fig. 10.12 is conducted, where all but the first design variable of the upper airfoil surface are changed by 5%. The first design variable is changed by 14% to achieve the perturbed shape shown in this figure. The distributions of pressure coefficients (Cp) shown in Fig. 10.13 are obtained from three different flowfield solutions. The first and the second flowfield solutions are obtained by solving the viscous flow governing equations (Eq. (6.33)) for the baseline configuration (Case 8.a), and the modified airfoil (Case 8.b), respectively. Whereas, the third flowfield solution is obtained using the flowfield prediction method (Eqs. (8.16) and (8.17)). These flowfield solutions are shown via the density contours in Figs. 10.14a and 10.14b. From Figs. 10.13 and 10.14, it can be seen that the analysis and the predicted results on the perturbed surface of the airfoil are indistinguishable after and before the region where the shock and the boundary layer interact. However, the slight discrepancies seen are believed to be due to the shock-boundary layer interactions which may increase the degree of nonlinearity of the relationship between the flowfield solution and the design variables (i.e., $Q = Q\,(\overline{X}_D)$). Hence, it is possibly the relatively large change (14%) in the first design variable, X_{D1}, that causes these discrepancies. This reconfirms the previous finding which shows that the accuracy of the prediction is very dependant on the magnitude of $\Delta\overline{X}_D$. Therefore, the maximum of $\Delta\overline{X}_D$ for each design variable per flow prediction step is conservatively restricted during the optimization process to 2% of its nominal value. This is important since an accurate prediction of the flow properties, particularly on the surface, is of great importance for a successful shape optimization. This point is further explained in Sec. 10.4.

10.3 A Comparison of Aerodynamic Sensitivity Coefficient Methods

The configuration shown in Fig. 7.4 is considered to examine both the accuracy and the efficiency of the sensitivity methods described in Section 8.1. Only the derivatives of the objective function, F, with respect to design variables are considered. As mentioned in Chapter 8, obtaining these derivatives via the finite–difference approximation (Eq. (8.1)) requires the perturbation of each design variable, X_D, by a step size ΔX_D. Then, the flowfield is computed for each new X_D using the flow analysis code. The sensitivity coefficients obtained via all three methods are normalized by the sensitivity coefficients obtained by the direct method, where Eq. (8.6) is evaluated using the flowfield solution associated with $X_{D1} = 20°$ and $X_{D2} = 12°$.

Since the best values for $\Delta\overline{X}_D$'s of the finite-difference approximation are not known a priori, different values of $\Delta\overline{X}_D$ are attempted. This is shown in Figs. 10.15 and 10.16 for X_{D1} and X_{D2}, respectively. Of course, the results of both direct and adjoint methods should be identical in this case since both of them are based on the same flowfield solution. As seen from these figures, the selection of the suitable step size of the finite-difference approximation is an estimate for the degree of desired accuracy. For example, if an accuracy of ± 5% is required, the changes in design variables should not exceed ±4%. This reconfirms the findings of a previous investigation [92] that the finite-difference method results are highly dependent on the step size.

The computational times for obtaining the sensitivity coefficients of the objective function, F, by all three methods are given in Table 10.7 for the case of two design variables. The system of linear equations arising from the analytical approach is solved using the banded matrix solver. All execution times in this table are normalized by the time required for the direct sensitivity method which is 23 CPU seconds on *CRAY-2* of NAS.

Table 10.7 Execution times to compute the sensitivity coefficients of the objective function, ∇F.

Approach[a]	Time
Finite difference approximation (3 flowfield solutions)	2.872
Direct Sensitivity Method[b] (1 flowfield solution plus sensitivity coefficients via banded solver)	1.000
Adjoint variable Method[c] (1 flowfield solution plus sensitivity coefficients via banded solver)	0.980

[a]Grid size is (53x41) in all computations.
[b]Number of design variables = 2.
[c]Number of adjoint vectors = 1.

It is concluded from Table 10.7 that the adjoint variable method requires slightly less time than the direct method. This is because there is only one adjoint vector (i.e., λ_1 for F) to be solved for when using the adjoint variable method. Whereas, in the direct method, there are two derivatives of the conserved flowfield variables, $\left(\partial Q/\partial X_{D1}, \partial Q/\partial X_{D2}\right)$ for which the equations need to be solved. In other words, the difference in the total execution time of the two methods is due to the extra time required for solving for an additional right hand side. The time required by the finite-difference approximation is significantly higher than that required by the direct method and the adjoint variable method. This is due to the repetitive flowfield analyses.

10.4 Demonstration of The Aerodynamic Shape Optimization Procedure

To assess the computational advantages of the proposed shape optimization procedure, a computer code called *ADOS* (Aerodynamic Design Optimization using

Sensitivity analysis) is developed. The example problems described earlier in Chapter 7 are solved using this computer code. To further demonstrate the shape optimization capability of *ADOS*, three design cases (Cases 9-11) are considered. In Case 9, the configuration shown in Fig. 7.4 is optimized using three different optimization procedures in order to investigate the efficiency of the new optimization procedure. In Case 10, the configuration shown in Fig. 7.5 is optimized starting from three different initial shapes in order to assess the accuracy of the proposed design procedure. Cases 9 and 10 exercise the new design capability in high speed flow regimes. In the third case (Case 10), the shape of the airfoil (Fig. 7.3) is optimized for different objective functions in order to examine the behavior of the optimization procedure in the transonic flow regime.

Table 10.8 Results of different aerodynamic design optimization procedures.

Procedure	X_{D1}, deg	X_{D2}, deg	F	G_1	G_2	G_3
			Initial values			
All	22.91	17.18	0.1201	–0.4223	1.2531	0.4082
			Optimum values			
1	18.34	16.42	0.1319	–0.0156	–0.3812	–0.1234
2	18.85	16.62	0.1322	–0.0162	–0.4030	–0.0685
3	18.95	16.53	0.1325	–0.0124	–0.3586	–0.1008

Presented in Table 10.8 are the results of the three optimization procedures used in Case 9. In Procedure 1, the sensitivity coefficients $[\nabla F, \nabla G_j]$ are computed by the traditional finite-difference approach (Eq. (8.1)). In Procedures 2 and 3, however, they are computed using the analytical approach, i.e.,the sensitivity analysis. In Procedures 1 and 2, a CFD analysis is performed every time the optimizer changes the design variables. In Procedure 3, an approximate flow analysis (Eqs. (8.16) and (8.17)) is performed if the optimizer is in the one-dimensional search cycle, and a CFD analysis is performed when the sensitivity coefficients are needed. All of these procedures are started from the same initial design.

Since the optimization requires that all constraints be satisfied (Eqs.(7.10)–(7.12)), the values of G_1, G_2, G_3 are all less than zero. The value of G_1 is the closest to zero, which indicates that it is the active constraint for this optimization problem. For the objective and constraints functions as defined by Eqs. (7.6) and (7.10)–(7.12), the optimum ramp and cowl angles are determined to be X_{D1} = 18.95° and X_{D2} = 16.62°, respectively. The values of the constants in Eqs. (7.10)–(7.12) used here are $C_1 = 0.6$, $C_2 = 1.0$, $C_3 = 1.5$. The values of the optimum axial thrust coefficient obtained by these procedures differ by ±0.03%. Therefore, all three procedures have comparable accuracies and they can be considered acceptable for engineering purposes.

In order to assess the relative efficiencies of Procedures 1–3, their computational execution times are given in Table 10.9. Also shown in this table are the number of CFD analyses and the number of approximate flow analyses performed in each procedure. All the execution times are normalized by that of Procedure 3, which is approximately 1.39 CPU hours on the *CRAY-2* computer at *NAS*. After a comparison of the execution times, it can easily be concluded that Procedure 3 is the most efficient one. This is due to the fact that a CFD flow analysis is the most expensive component of the present aerodynamic design optimization method and Procedure 3 uses the fewest number of CFD analyses.

Table 10.9 Normalized computational times required by different aerodynamic optimization procedures.

Procedure[a,b]	Execution time	No. of CFD analyses	No. of approx. flow analyses
1	4.480	32	–
2	2.848	20	–
3	1.000	6	14

[a] Grid size is (53x41) in all procedures.
[b] Number of design variables = 2.

Procedures 1–3 require six optimization iterations to get a converged optimum solution. The history of the optimization process for Procedure 3 is shown in Fig. 10.17. Shown in Figs. 10.18 and 10.19 are the pressure distribution histories on both the ramp and cowl surfaces for different optimization iterations. As expected, the flow expands more rapidly with the increased ramp angle, and that, in turn, increases the axial thrust force coefficient. However, it should be noted that the ramp angle is bounded by the constraint G_1 , which imposes the no-reverse-flow condition at the ramp tip.

Next, the ramp shape of the configuration shown in Fig. 7.5 is optimized. Three different and arbitrary shapes (Fig. 10.20) are chosen as the initial design shapes for the ramp; namely, a flat ramp surface at $\alpha = 10°$ (Case 10.a), a concave surface with its axis (the straight line connecting the corner point and the ramp endpoint) at $\alpha = 25.7°$ (Case 10.b), and a convex surface with its axis at $\alpha = 29.5°$ (Case 10.c). The distinguishing features of Case 10 are given in Table 10.10. The reason for starting the optimization from three different initial shapes is to determine how close the resulting optimized ramp shapes are to each other. Ideally, they should be identical irrespective of their initial shape, so that, the designer using this method in the production mode can start with any shape that is convenient. That is, the final optimized shape should ideally be independent of the initial shape.

Table 10.10. Distinguishing features of the ramp shape optimization cases.

Case[a,b]	Initial ramp shape	The slope of initial ramp expansion
10.a	Flat	10°
10.b	Concave	35.0°
10.c	Convex	23.5°

[a] Grid size is (53x41) in all cases.
[b] Number of design variables = 47.

Presented in Fig. 10.21 is the evolution of the ramp shapes during their optimization processes. For each case, the ramp corner point and the axial length are the only fixed parameters. As seen from Figs. 10.21a and 10.21b, shapes subsequent to the initial shape, denoted by the increasing iteration numbers, systematically approach the final optimum shape. For example, the flat shape surface of Case 10.a (Fig. 10.21a) changes to a concave shape at the first iteration and, as the optimization process continues, the concavity of the shape increases. It should be noticed that after the second iteration, the slope of the initial ramp expansion does not change with the same rate as that of the ramp tip. This means that no significant increases in the expansion rate at the ramp corner occurs after the second iteration despite the recompression of the flow by the succeeding shapes. The opposite of this occurs in Case 10.b (Fig.10.21b), where the shape concavity decreases during the last three iterations. A comparison of the optimum shapes of Case 10 along with their initial shapes is shown in Fig. 10.20. The optimum shapes are almost identical for 70% of the surface and show a small difference towards the tip. When the axis angles, α, of the optimum shapes are compared, it can be seen that the difference between Case 10.b and 10.c is indistinguishable (less than 0.3%) and that of Case 10.a differs from them by only 3%.

The effect of the optimization of the shape on the off-surface flowfield is just as pronounced as it is on the surface properties. The Mach number contours of both initial and optimum configurations of Case 10.a are presented in Fig. 10.22. The expansion patterns are significantly different. The rate of expansion is much higher inside the nozzle for the optimum shape (Fig. 10.22b), which results in a higher Mach number at the nozzle exit plane. The consequence of this is evidenced in the shear layer, which is thinner and has a smaller angle with the horizontal for the optimum shape. Also, the expansion along the external part of the nozzle ramp is no longer predominantly in the streamwise direction, but a significant portion is in the normal direction. This indicates that cancellation of the cowl corner centered-expansion waves occurs at the optimized ramp surface, which is a characteristic feature of bell-type nozzles.

The coefficient of thrust for each shape is an integrated value of the local flow quantity, *Cp*. To study this local flow quantity and its distribution over the optimum shape, the pressure coefficients of all three shapes are plotted in Fig. 10.23. Again, the initial distributions are vastly different as chosen deliberately. The optimum distributions are not identical but very close to each other. This should be expected since the objective function (Eq. (7.5)), which guides the shape optimization, is an integrated property *(F)*. It appears that each *Cp* distribution compensates for the slight differences exhibited in each case's initial rate of centered flow expansion over the ramp. For example, Case 10.b has the largest expansion at the corner, then it recompresses more downstream to match the expansion of the other cases. From Figs. (10.21) and (10.23), it is observed that any small difference in the ramp initial angle affects the centered expansion significantly. Hence, obtaining truly identical optimal shapes is highly unlikely. Nevertheless, the results of all cases are very close to each other.

Plotted in Fig. 10.24 are the histories of the objective functions, F, during the optimization iterations. The initial F value of Case 10.b is the highest, and all three shapes converge to an optimum F value within 14 optimization iterations. Case 10.a and 10.c have identical optimum F values to the fourth significant digit, and that of Case 10.b differs from them by less than 0.5%.

The computational time for each one of the above cases (Case 10) is about seven hours on the *CRAY-2* computer of *NAS*. For example, Case 10.a requires 180 evaluations of the objective function over the course of 14 optimization iterations. At the end of each iteration, there is a CFD analysis accompanied with the objective function evaluation for the newly improved shape. This means that only 14 CFD analyses are performed, whereas, 166 flowfield prediction calculations are performed. Comparing the computational time of an analysis (600 seconds) and a flowfield prediction (40 seconds), it can easily be realized that the aerodynamic optimization procedure is more efficient by employing the present prediction method.

Finally, the aerodynamic shape optimization procedure is applied for two transonic airfoil design problems, described earlier in Section 7.3, in order to evaluate its efficiency and performance at the transonic speeds (Case 11). In this case viscous computations are performed using a C-grid with 120x32 cells. The grid is regenerated for the various intermediate airfoil shapes that are evoluted during the optimization. Both design problems (Case 11.a and Case 11.b) start by the NACA 0012 airfoil as an initial baseline airfoil. The Mach contours for the flow past this baseline airfoil are shown in Fig. 10.25. In both cases, only one surface of the airfoil is shape optimized and the immediate vicinity of the leading edge is excluded. Therefore, only 29 points on one surface rather than all 89 points along the entire airfoil become the design variables.

The first design problem (Case 11.a) aims at reshaping the upper surface of the baseline airfoil in order to minimize the drag coefficient while keeping the lift coefficient greater than an arbitrarily assigned C_{Lmin} value of 0.2. The final shape, as shown in Fig.(10.16a), is appreciably different than the baseline. The pressure distribution computed on the redesigned airfoil is also given in Fig. 10.26a. The suction peak level on the upper surface is increased and the pressure gradient downstream of the shock is slightly increased. The new design results in an increased included angle at the trailing angle. This causes an increase in the pressure gradient, which in turn increases the viscous drag. On the other hand, the increased suction peak results in an increased wave drag, since the upper surface shock is located further downstream than that of the baseline airfoil (Fig. 10.27). Although, the lower surface shock changes in just the opposite manner, this change is much less than that of the upper surface shock. Therefore, the total drag is increased, as seen in the optimization history (Fig. 10.28). Certainly, the C_{Lmin} would not have been obtained otherwise. However, the optimum value of C_D (0.01713) is the minimum value for this specified C_{Lmin}.

The optimum airfoil shape of Case 11.a is obtained after 13 optimization iterations, where 178 evaluations of the objective functions are needed. Only at the end of each iteration, there is a CFD analysis accompanied with the objective function evaluation for the new improved shape. This means that 13 CFD analyses are performed, whereas the flow prediction is performed 165 times. Comparing the computational time of a CFD analyses (900 CPU seconds) and a viscous flow prediction (70 CPU seconds), it can easily be realized that the aerodynamic optimization procedure becomes much more efficient by the use of the viscous flow prediction method. The total computational time for this design problem is about eight hours on the *CRAY-YMP* computer of *NAS*.

The second design problem (Case 11.b) is the reshaping of the lower surface of the baseline airfoil to maximize the lift coefficient while keeping the drag coefficient less than C_{Dmax} value of 0.02. The pressure distribution computed on the redesigned airfoil is shown in Fig. 10.29. Similar to Case 11.a, the total drag at the optimum design is increased, but the main reason in this case is the increased wave drag as a consequence of moving the location of the lower surface shock further upstream. In addition, the included trailing edge angle of the final design is decreased, which in turn decreases the pressure gradient downstream of the shock. Consequently, the viscous drag is decreased. Shown in Fig. 10.30 are the Mach contours for the optimized airfoil. By comparing Figs. 10.27 and 10.30, it is observed that the upper surface shock of Case (11.a) is located further downstream than that of Case 11.b.

The optimization history of Case 11.b is plotted in Fig. 10.31. The optimum C_L value is 0.1751. The airfoil shape of Case 11.b was obtained after nine optimization iterations, which is less than that required in Case 11.a. However, 190 objective function evaluations are needed in this case. Only nine CFD analyses are performed to compute the objective function, whereas, the rest of the objective function evaluations are obtained by the flow prediction method. Case 11.b requires about seven hours on the *CRAY-YMP* computer of NAS.

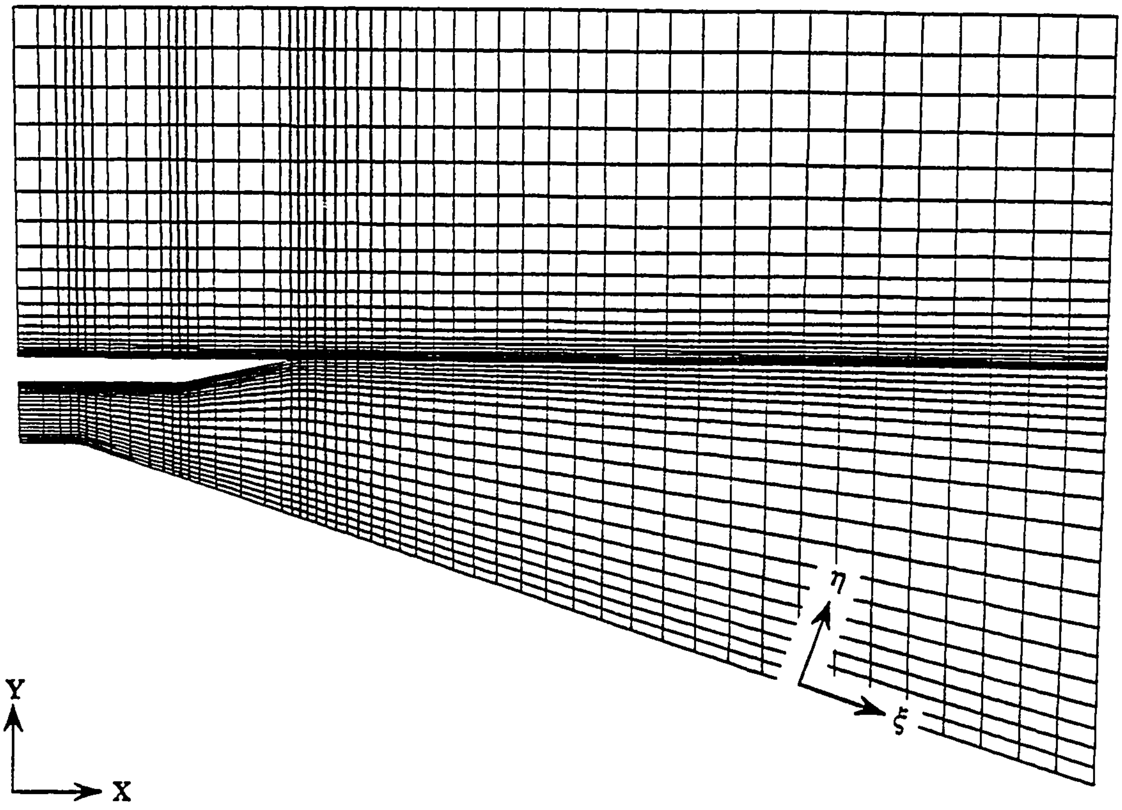


Fig. 10.1 Typical computational grid for Case 5.a.

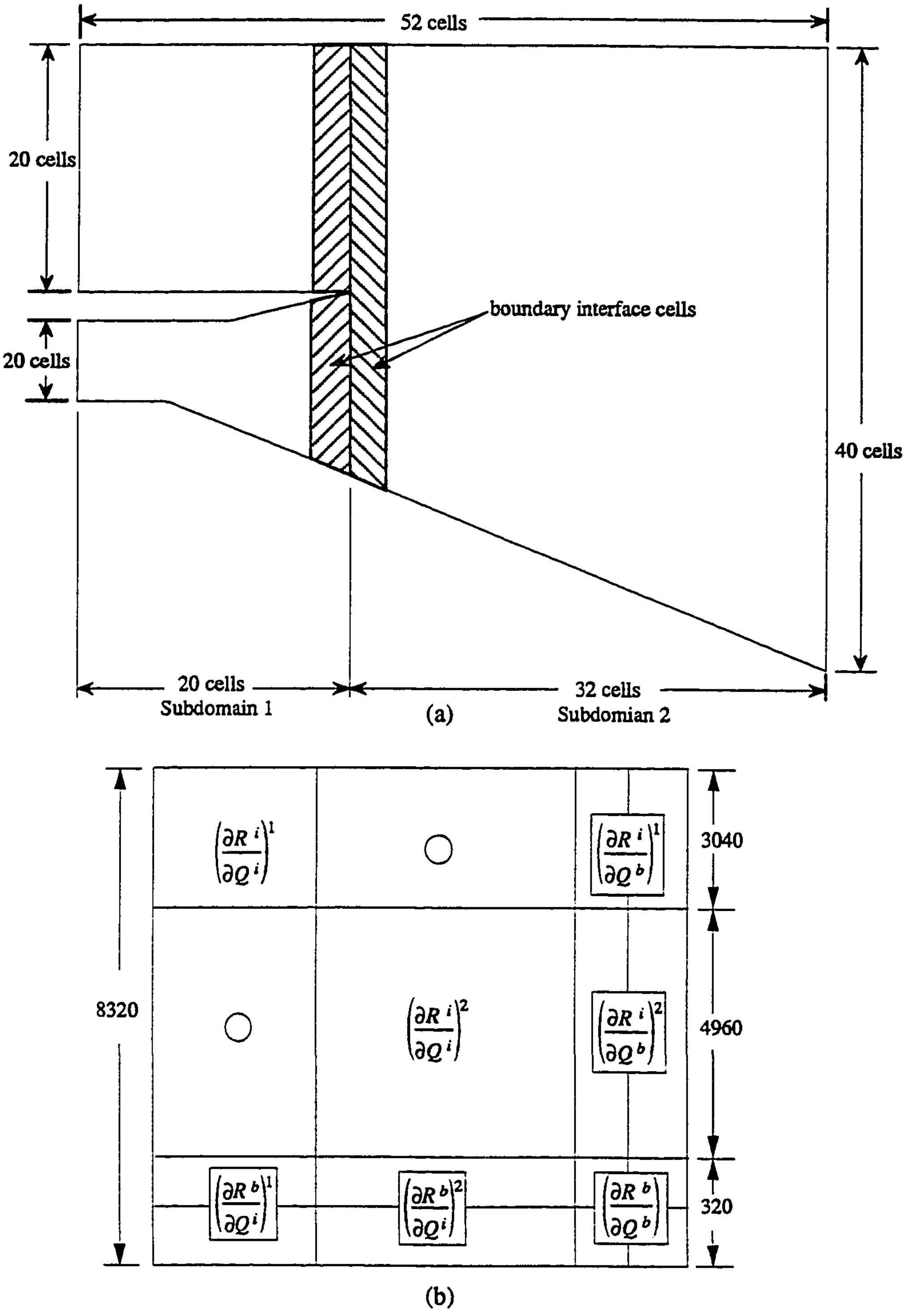

52 cells
20 cells
20 cells
boundary interface cells
40 cells
20 cells
Subdomain 1
(a)
32 cells
Subdomian 2
8320
$\left(\frac{\partial R^i}{\partial Q^i}\right)^1$
○
$\left(\frac{\partial R^i}{\partial Q^b}\right)^1$
3040
○
$\left(\frac{\partial R^i}{\partial Q^i}\right)^2$
$\left(\frac{\partial R^i}{\partial Q^b}\right)^2$
4960
$\left(\frac{\partial R^b}{\partial Q^i}\right)^1$
$\left(\frac{\partial R^b}{\partial Q^i}\right)^2$
$\left(\frac{\partial R^b}{\partial Q^b}\right)$
320
(b)

Fig. 10.2 (a) Schematic showing a decomposed computational domain of Case 5.b (b) Schematic showing the sizes of the components in the global coefficient matrix of Case 5.b.

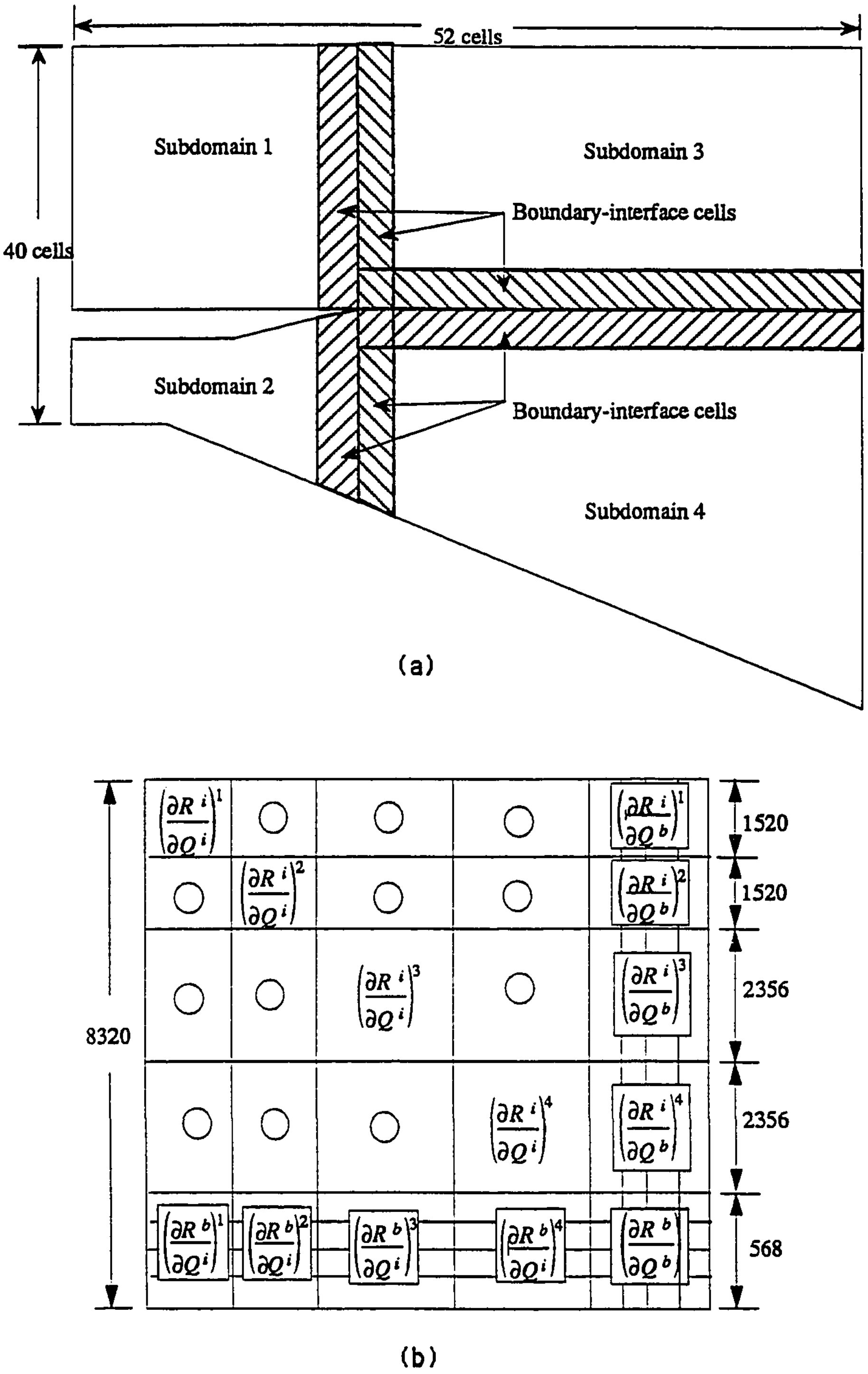
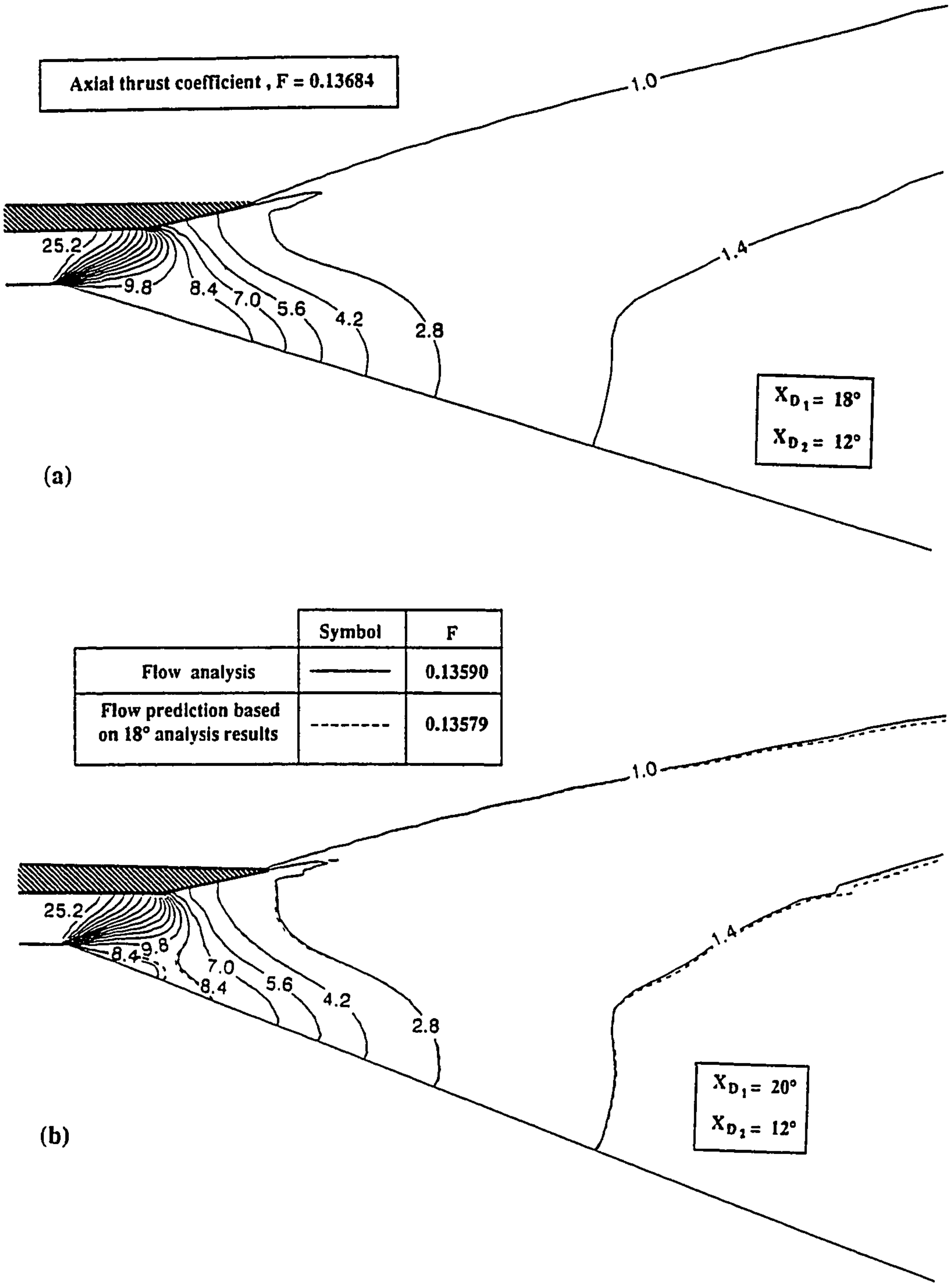

52 cells
40 cells
Subdomain 1
Subdomain 3
Boundary-interface cells
Subdomain 2
Boundary-interface cells
Subdomain 4
(a)
$\left(\frac{\partial R^i}{\partial Q^i}\right)^1$
$\left(\frac{\partial R^i}{\partial Q^i}\right)^2$
$\left(\frac{\partial R^i}{\partial Q^i}\right)^3$
$\left(\frac{\partial R^i}{\partial Q^i}\right)^4$
$\left(\frac{\partial R^i}{\partial Q^b}\right)^1$
$\left(\frac{\partial R^i}{\partial Q^b}\right)^2$
$\left(\frac{\partial R^i}{\partial Q^b}\right)^3$
$\left(\frac{\partial R^i}{\partial Q^b}\right)^4$
$\left(\frac{\partial R^b}{\partial Q^i}\right)^1$
$\left(\frac{\partial R^b}{\partial Q^i}\right)^2$
$\left(\frac{\partial R^b}{\partial Q^i}\right)^3$
$\left(\frac{\partial R^b}{\partial Q^i}\right)^4$
$\left(\frac{\partial R^b}{\partial Q^b}\right)$
8320
1520
1520
2356
2356
568
(b)

Fig. 10.3 (a) Schematic showing a decomposed computational domain of Case 5.c
(b) Schematic showing the sizes of the components in the global coefficient matrix of Case 5.c.



	Symbol	F
Flow analysis	———	0.13590
Flow prediction based on 18° analysis results	--------	0.13579

Fig. 10.4 Normalized pressure contours, P/P_∞, of the nozzle flow: (a) Flow analysis for X_{DI} = 18° (b) Comparison of the flow analysis and the flow prediction for X_{DI} = 20°.

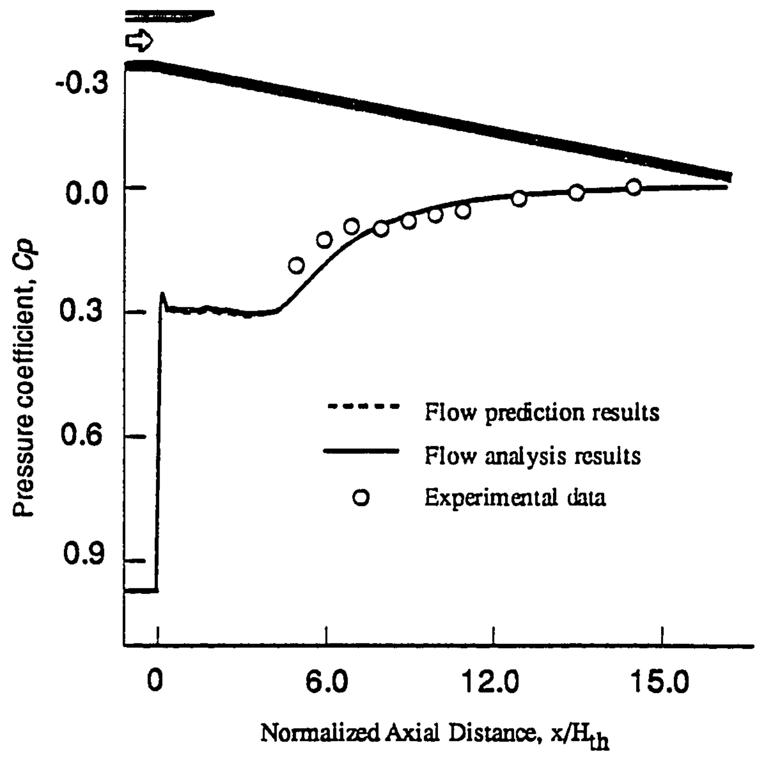


Fig. 10.5 Pressure coefficient distributions on the ramp surface obtained from different numerical approaches in comparison with the experimental data of Ref. [5].

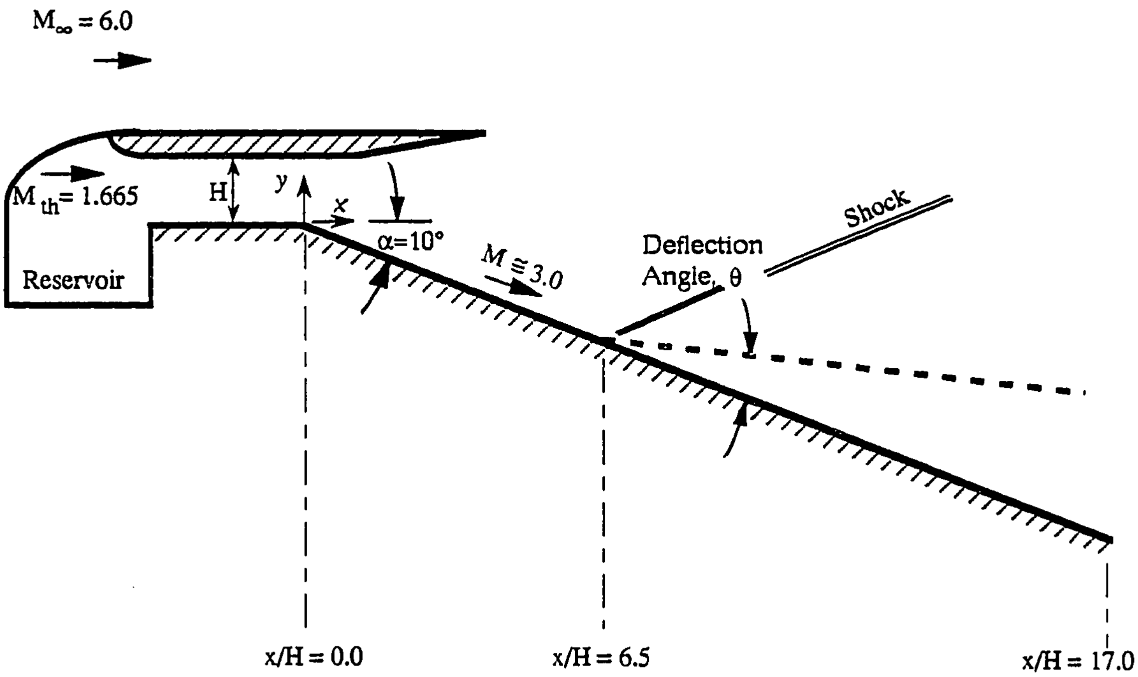


Fig. 10.6 Description of the flowfield prediction problem (Case 7).

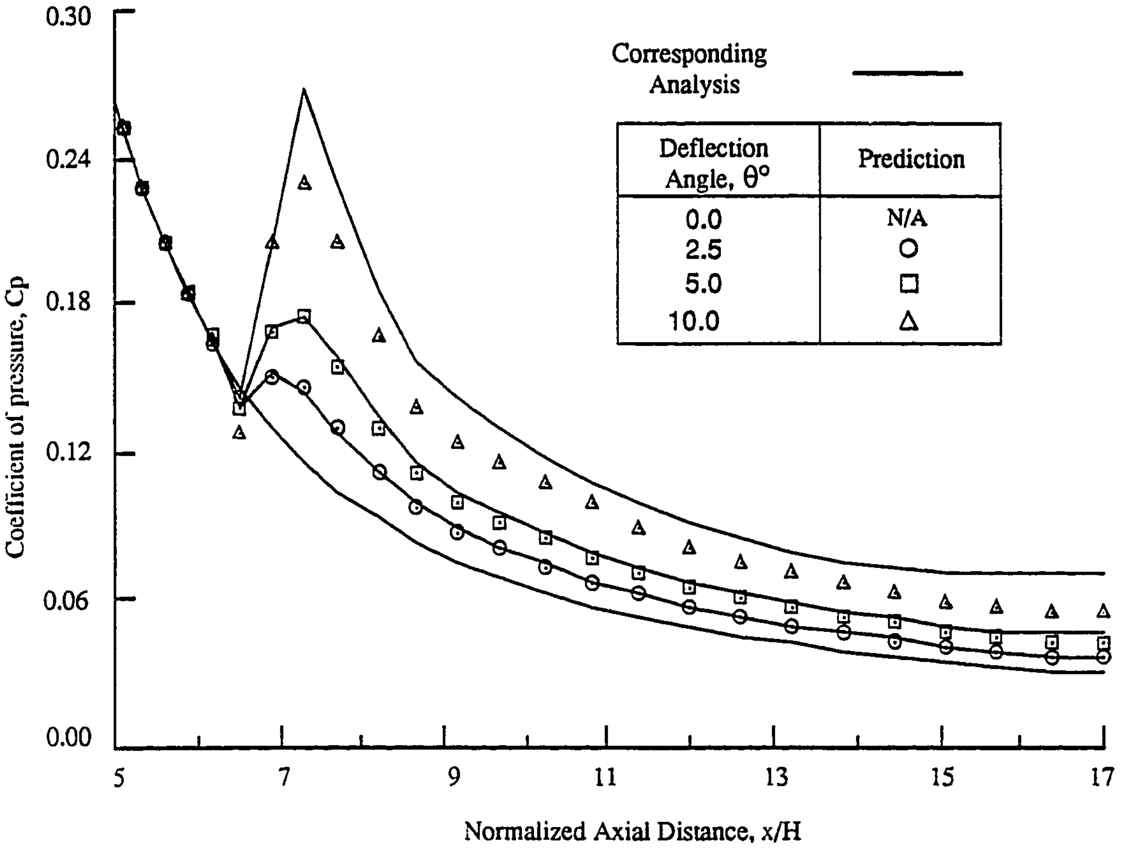


Fig. 10.7 Distribution of *Cp* along ramp for various deflection angles.

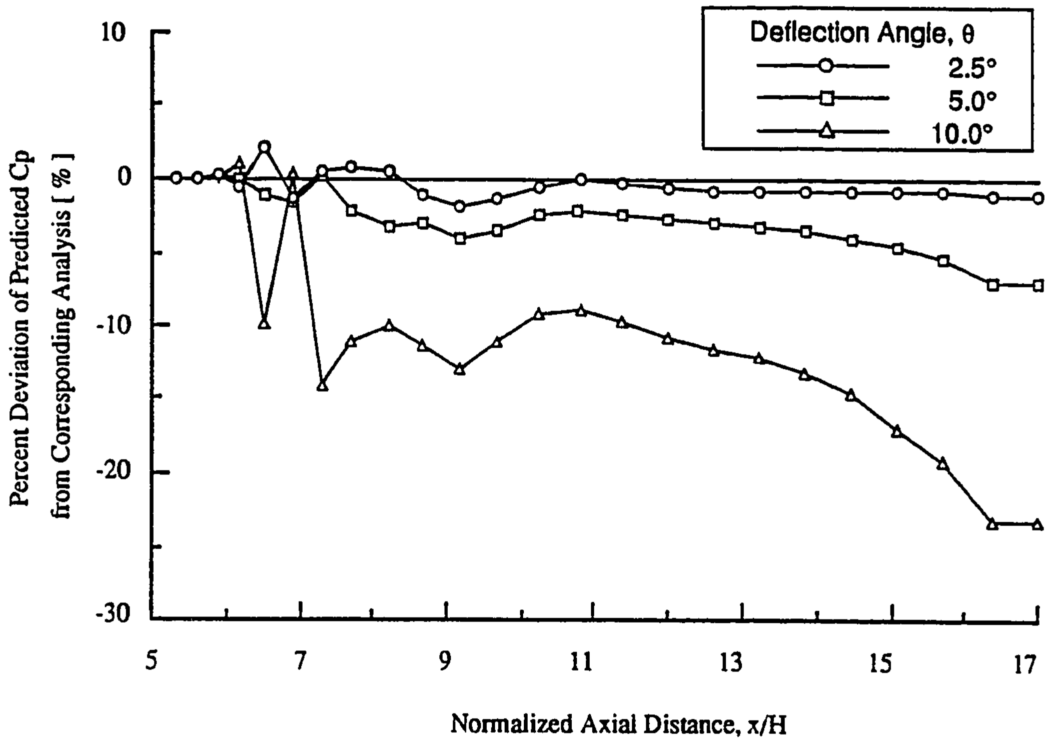


Fig. 10.8 Percent deviation of *Cp* along ramp for various ramp deflections.

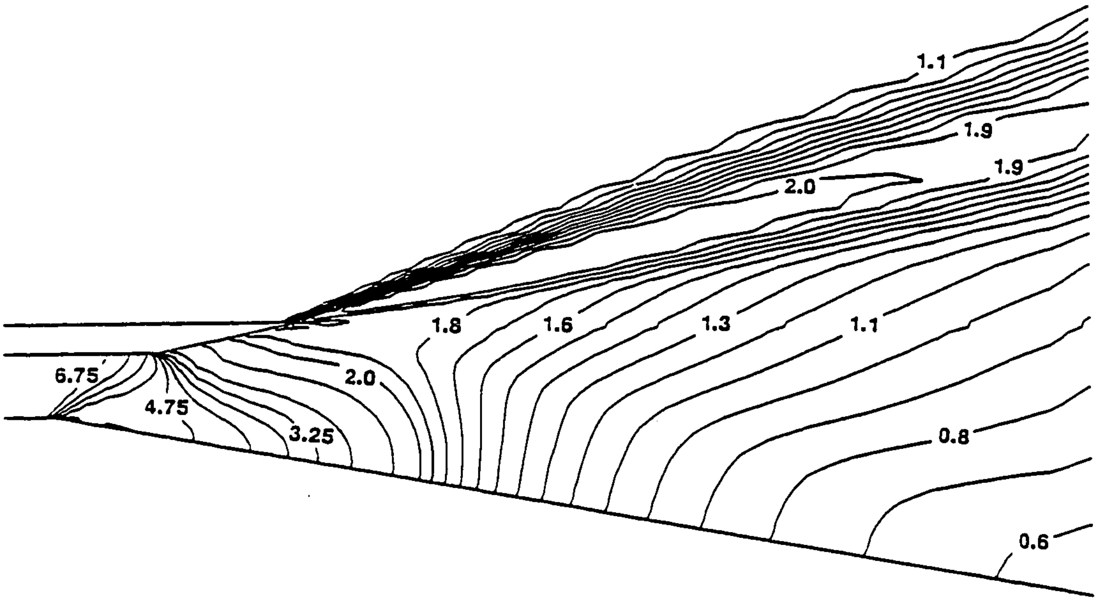


Fig. 10.9 Density contours of a 10° flat ramp case (Case 7.a).

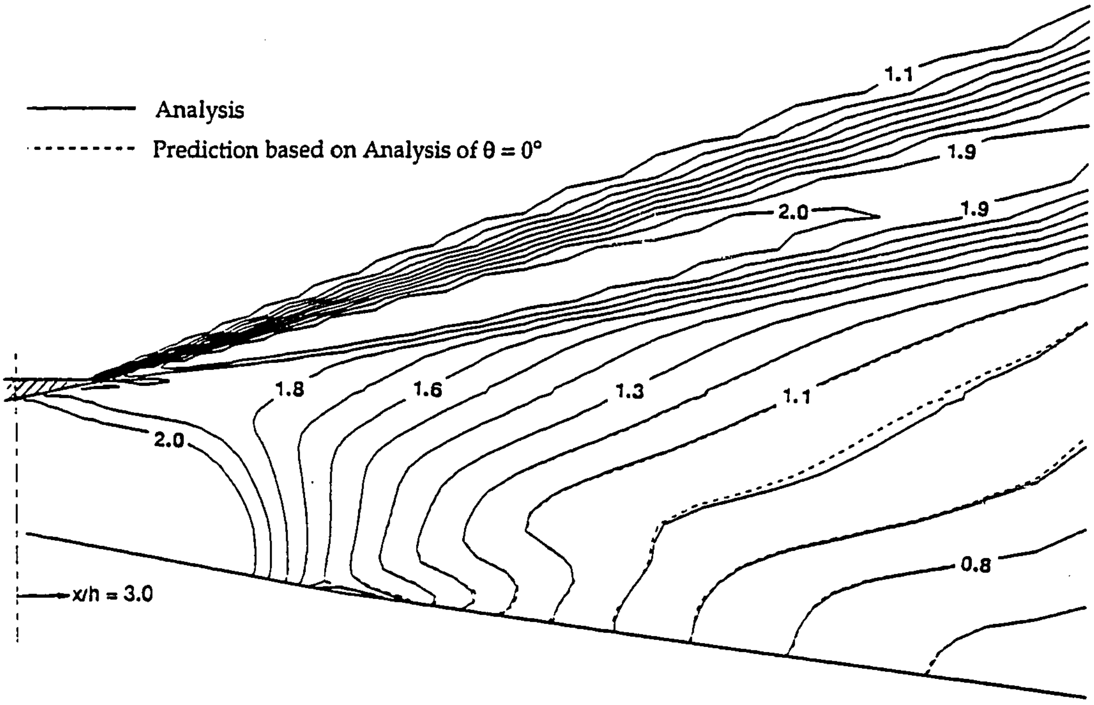


Fig. 10.10 Comparison of density contours for $\theta = 2.5°$ deflection.

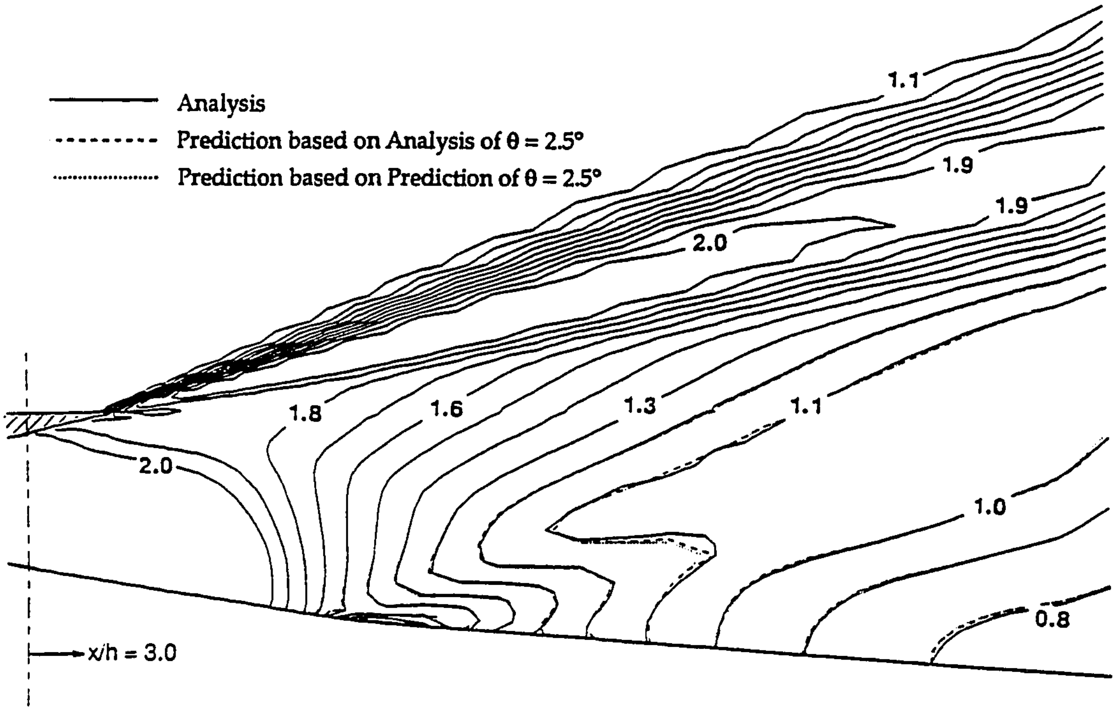


Fig. 10.11 Comparison of density contours for $\theta = 5.0°$ deflection.

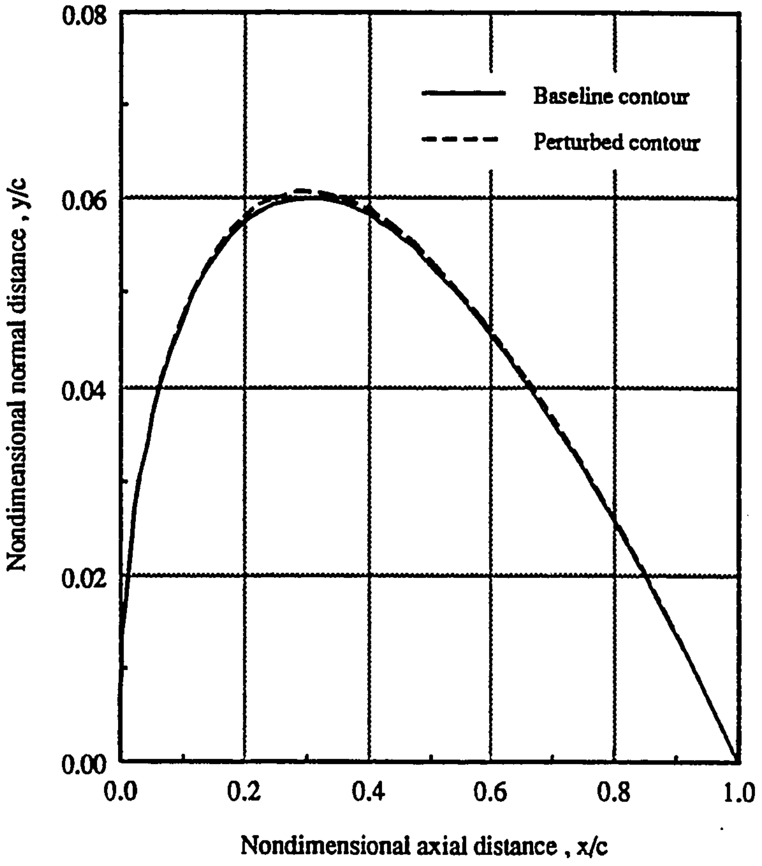


Fig. 10.12 Modifications of the upper surface of the airfoil.

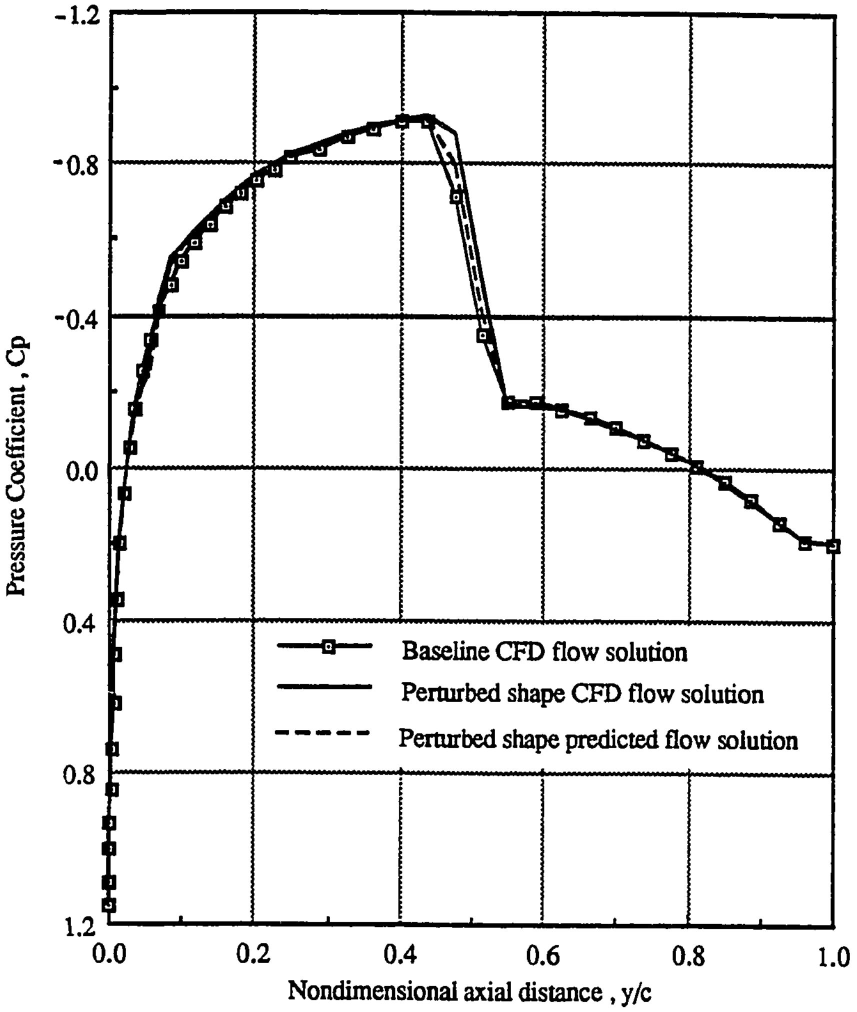


Fig. 10.13 A comparison of the pressure coefficient distributions along the upper airfoil surfaces .

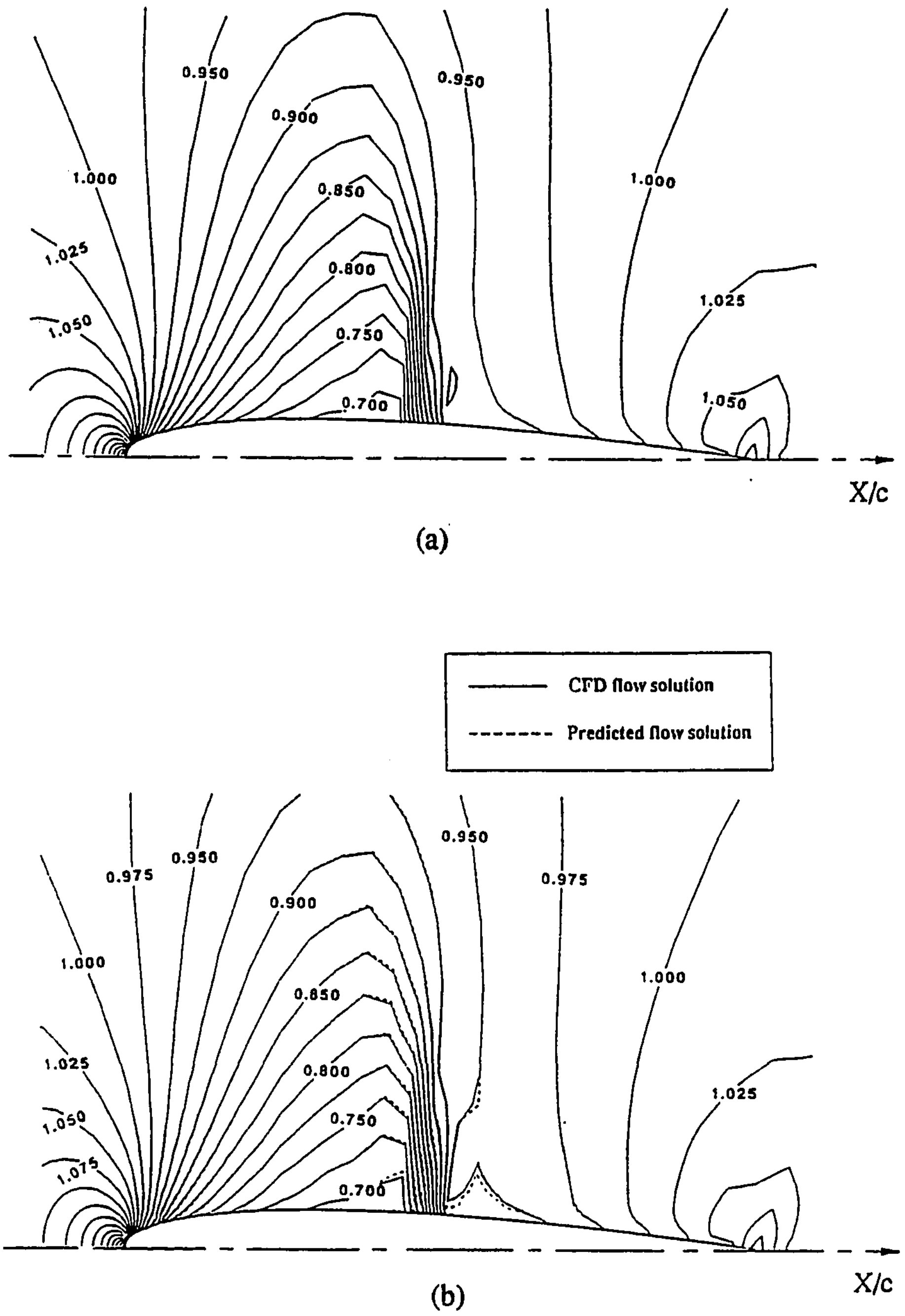


Fig. 10.14 (a) Density contours for NACA 0012 airfoil (Baseline)
(b) Comparison of computed and predicted density contours for perturbed upper airfoil surface.

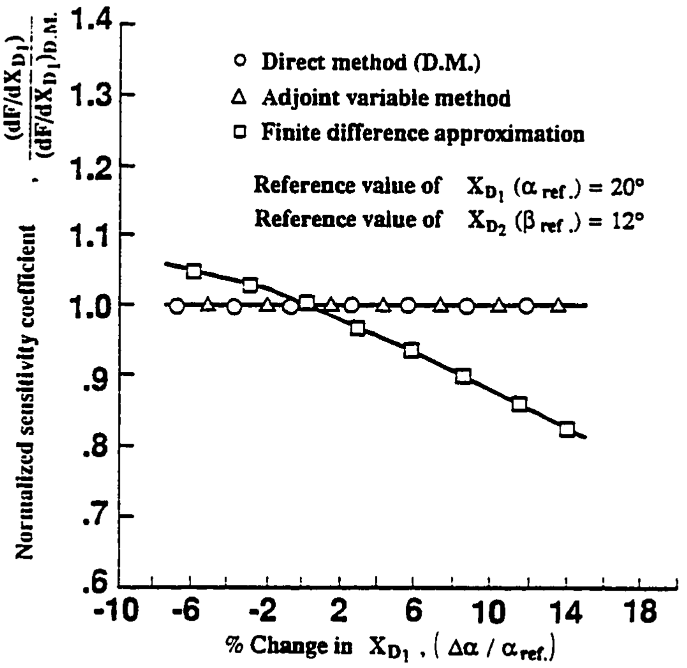


Fig. 10.15 Variation of the normalized sensitivity coefficient with the percent change of the design variable, X_{D1}, for three methods.

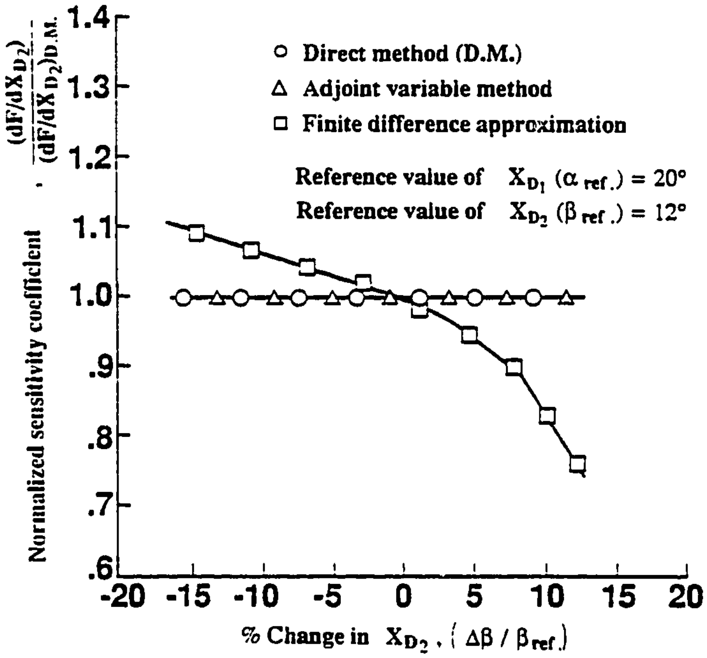


Fig. 10.16 Variation of the normalized sensitivity coefficient with the percent change of the design variable, X_{D2}, for three methods.

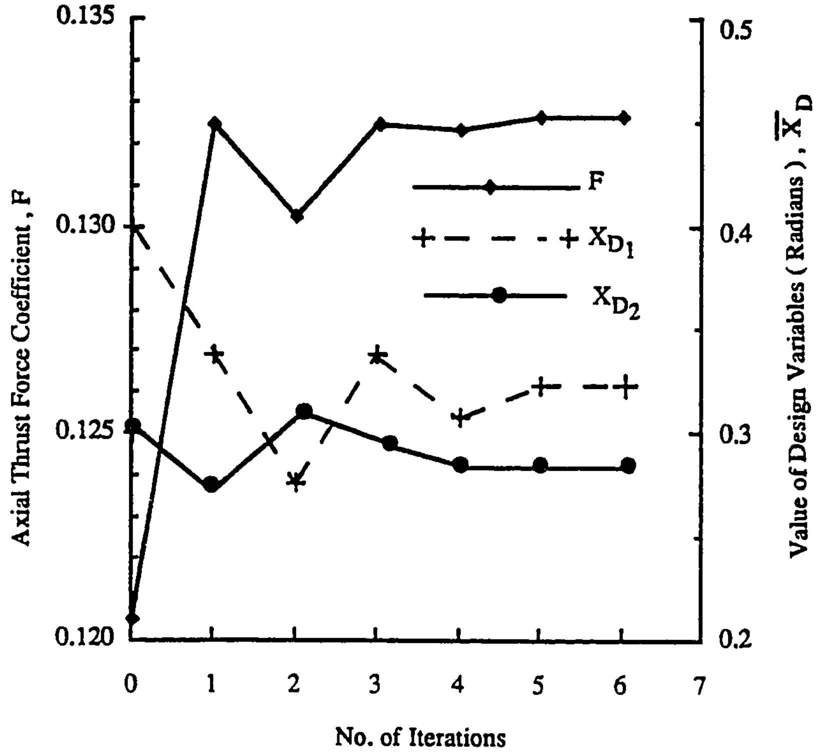


Fig. 10.17 History of the optimization process for Procedure 3.

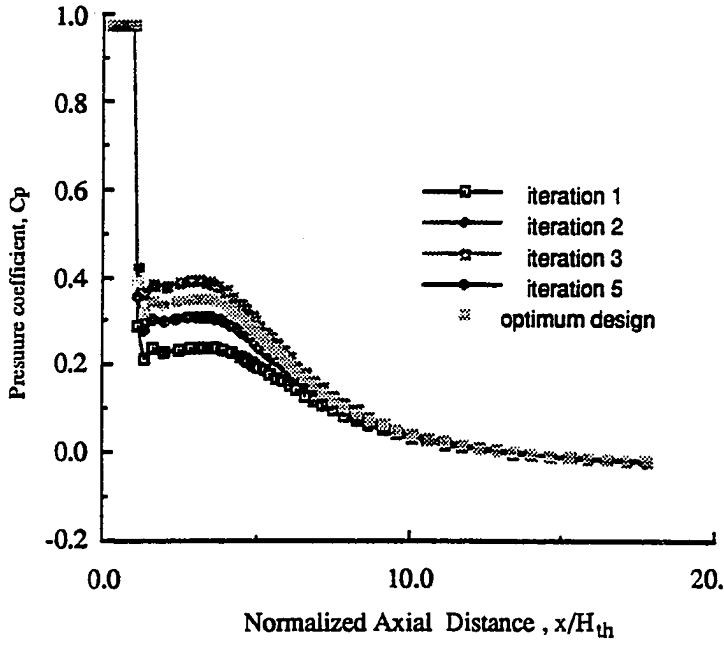


Fig. 10.18 Variation of the ramp surface pressure coefficients with the optimization iterations.

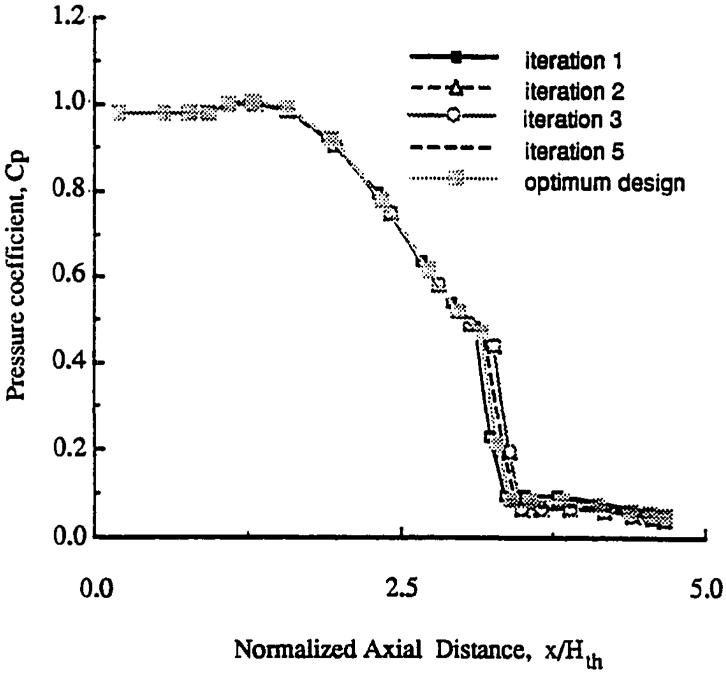


Fig. 10.19 Variation of the cowl surface pressure coefficients with the optimization iterations.

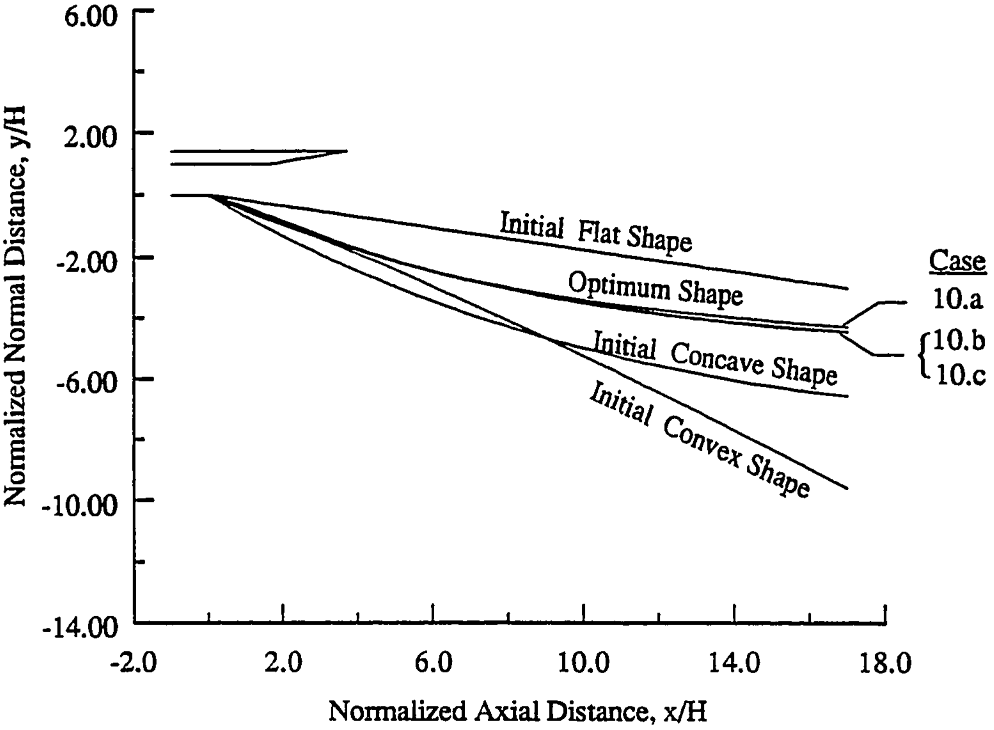


Fig. 10.20 Comparison of the final optimized ramp shapes (Cases 10.a, 10.b,10.c).

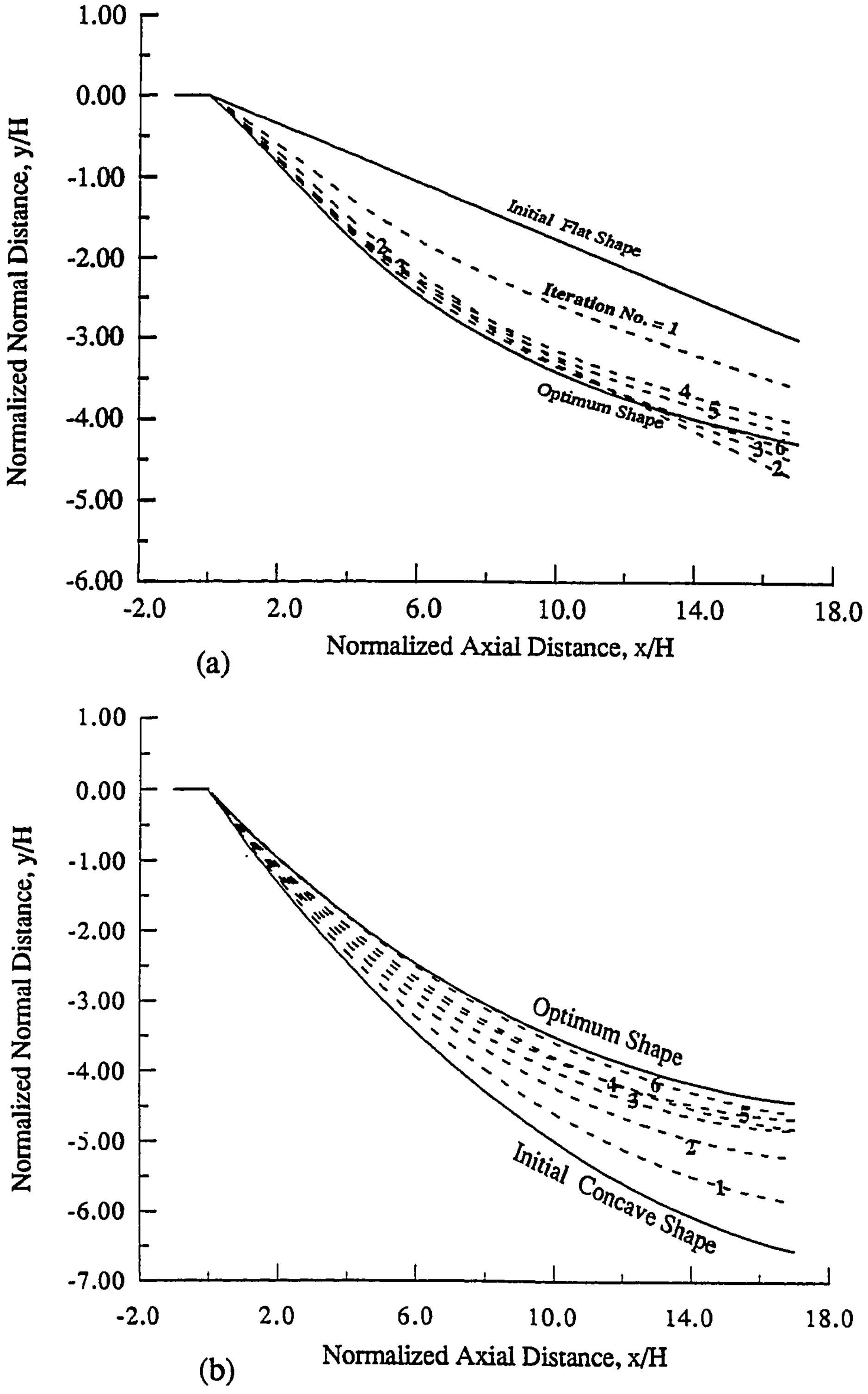


Fig. 10.21 (a) Evolution of the ramp shape for Case 10.a
(b) Evolution of the ramp shape for Case 10.b.

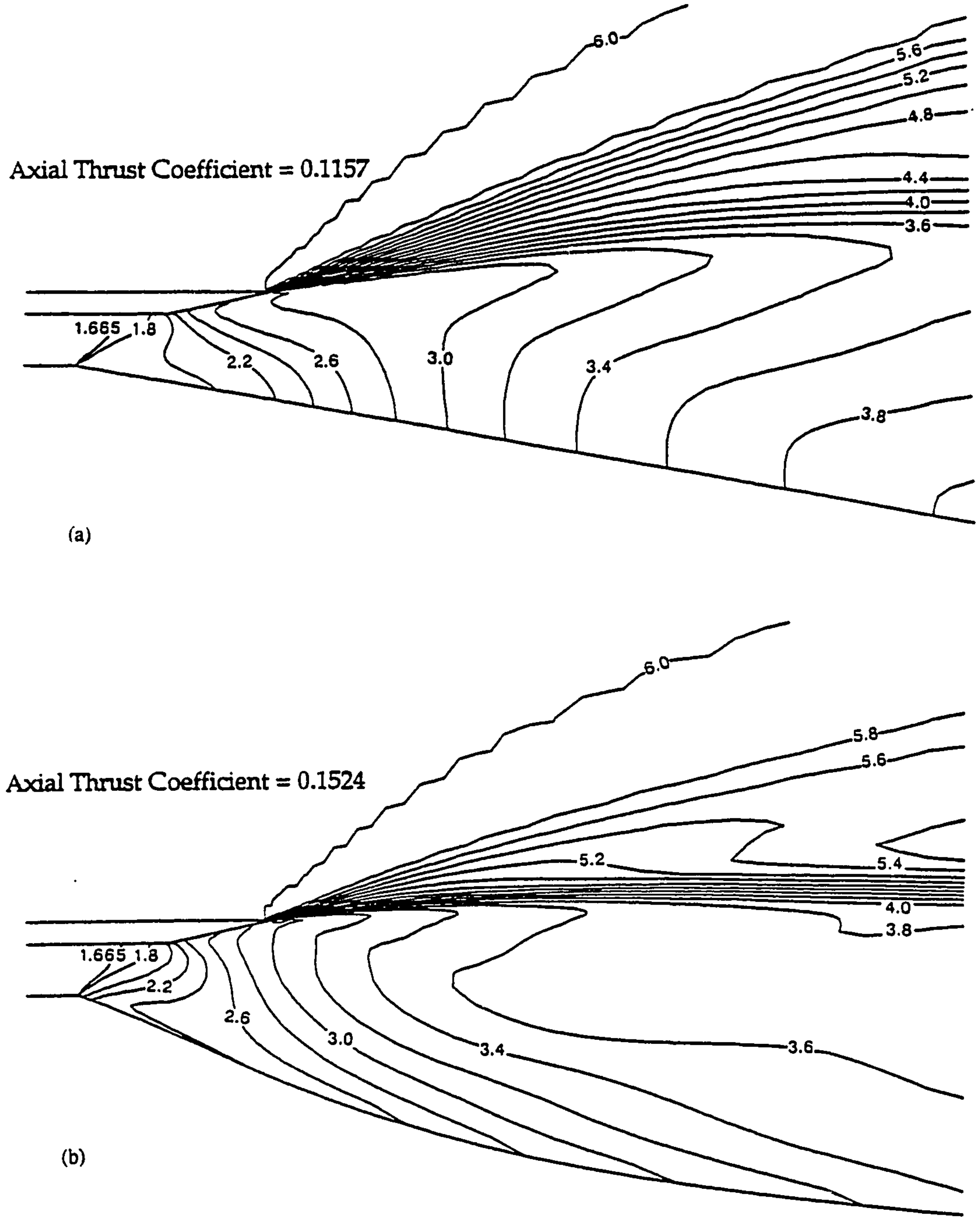


Fig. 10.22 (a) Mach contours of the initial ramp shape (Case 10.a)
(b) Mach contours of the optimized ramp shape (Case 10.a).

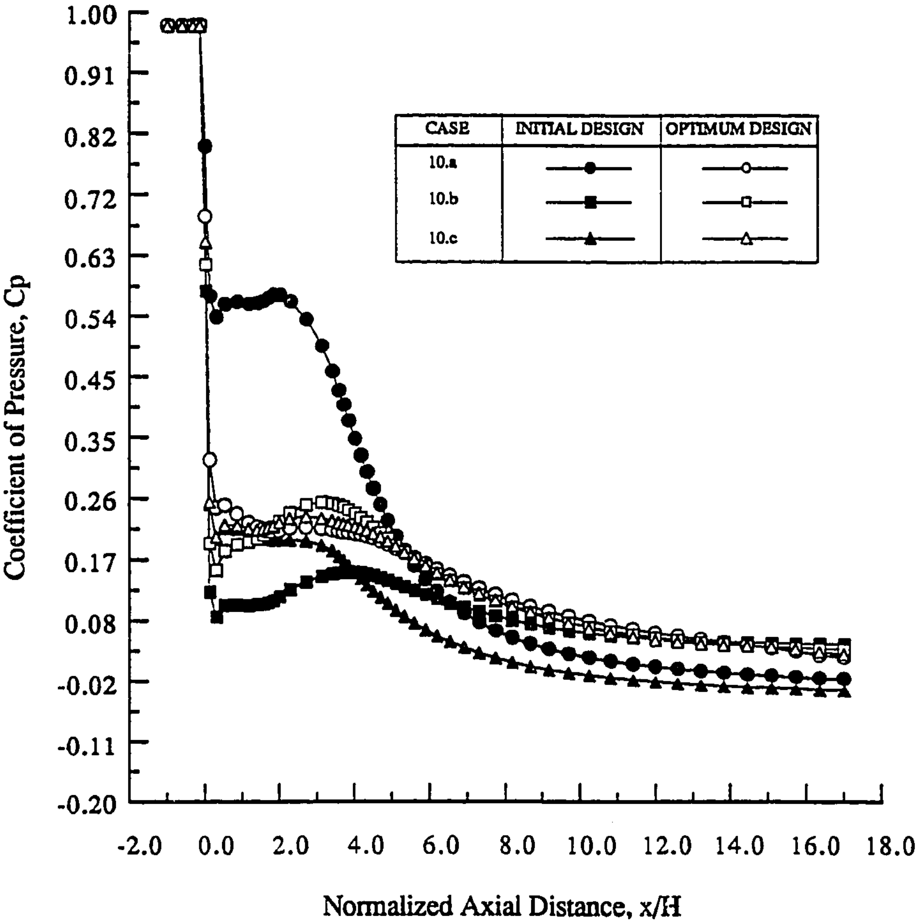
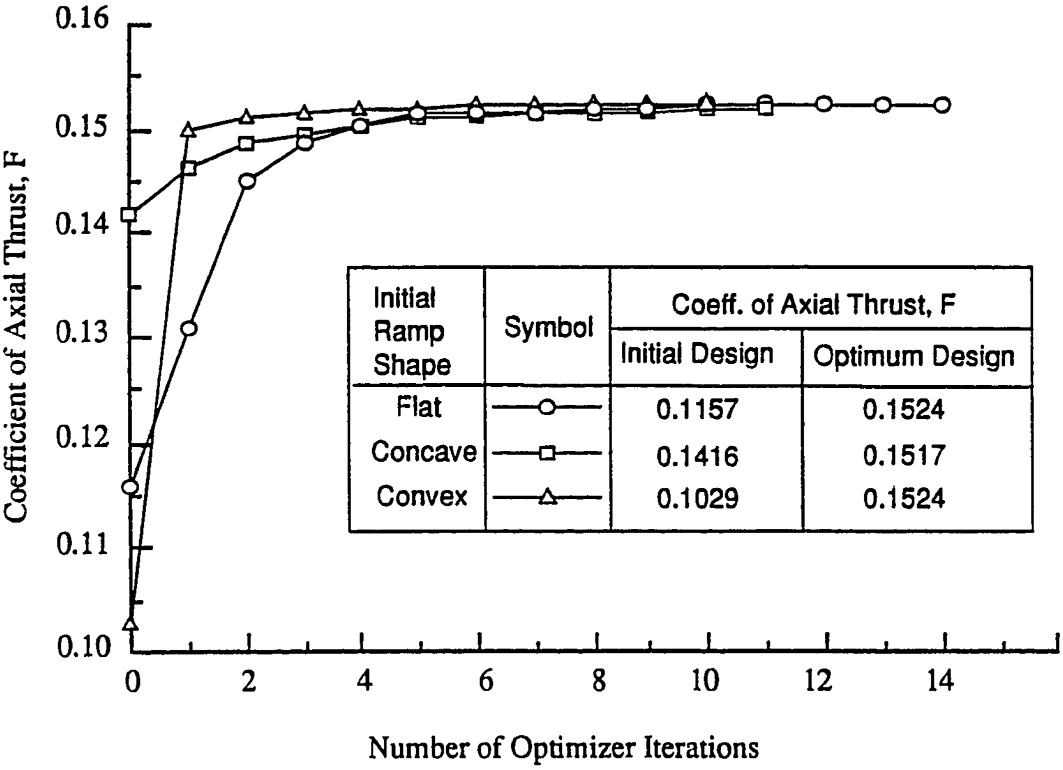


Fig. 10.23 Comparison of *Cp* distributions along ramp for initial and optimum shapes.



Initial Ramp Shape	Symbol	Coeff. of Axial Thrust, F	
		Initial Design	Optimum Design
Flat	—○—	0.1157	0.1524
Concave	—□—	0.1416	0.1517
Convex	—△—	0.1029	0.1524

Fig.10.24 Optimization history of the objective function for Cases 10.a, 10.b, 10.c.

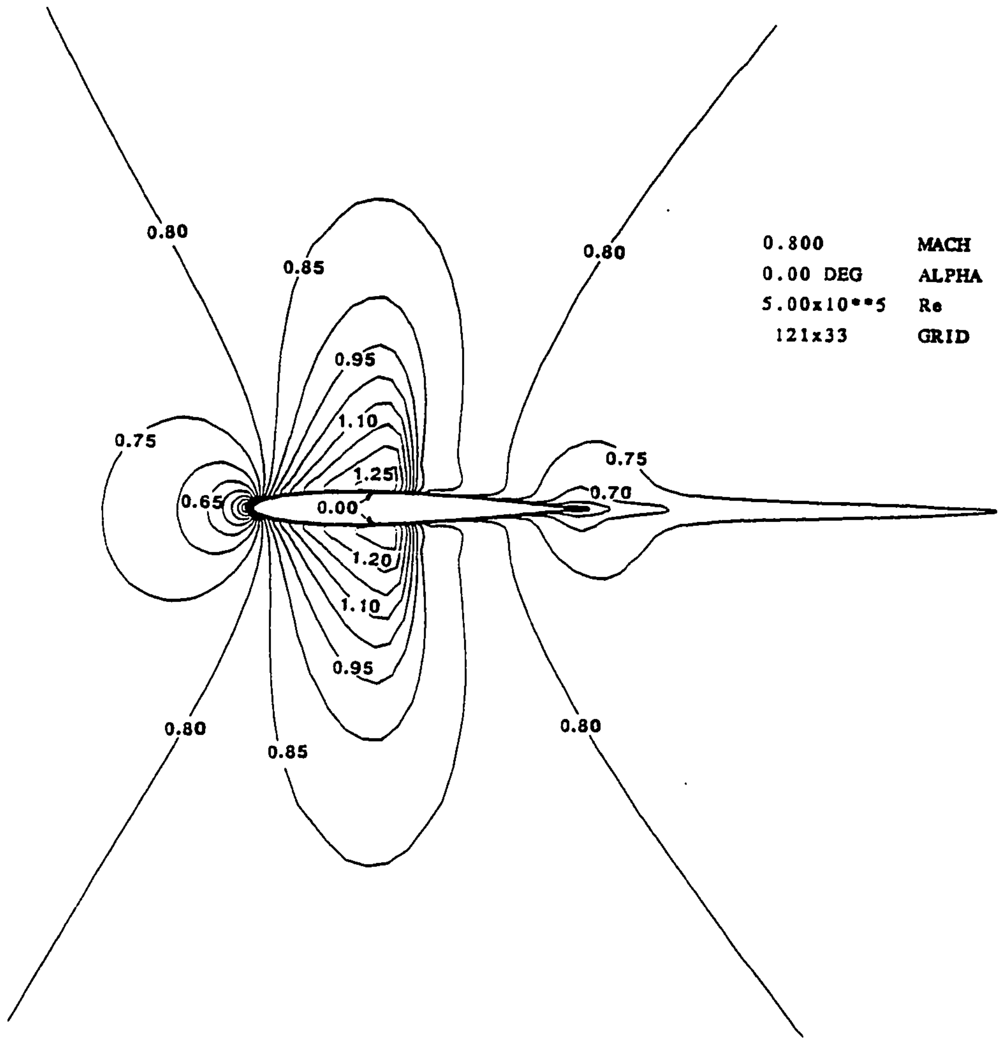
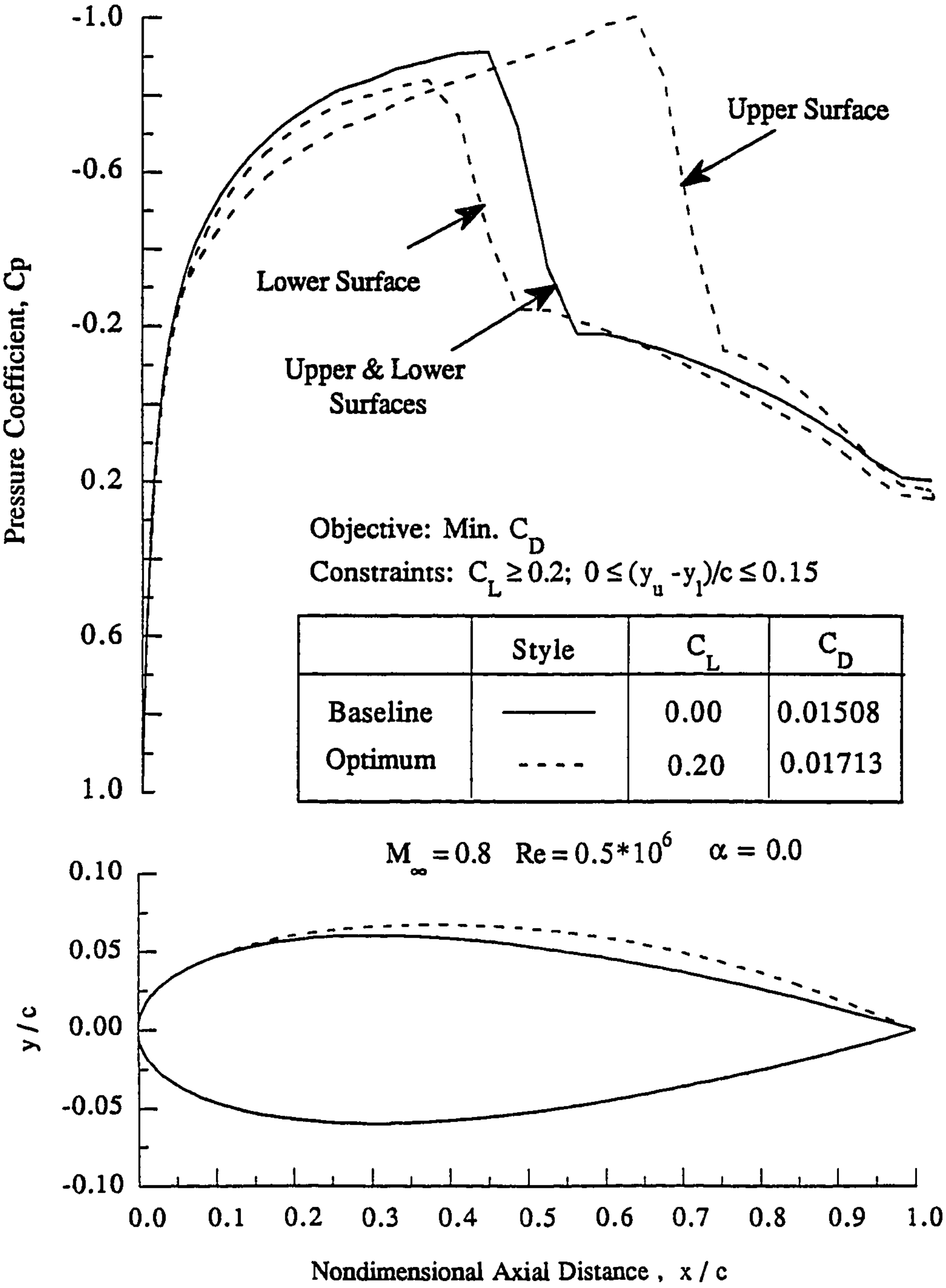


Fig. 10.25 Mach contours for NACA 0012 airfoil (baseline airfoil).



Objective: Min. C_D

Constraints: $C_L \geq 0.2$; $0 \leq (y_u - y_l)/c \leq 0.15$

	Style	C_L	C_D
Baseline	———	0.00	0.01508
Optimum	- - - -	0.20	0.01713

$M_\infty = 0.8$ $Re = 0.5*10^6$ $\alpha = 0.0$

Fig. 10.26 Airfoil shape and distribution of surface pressure coefficients (Case 11.a).

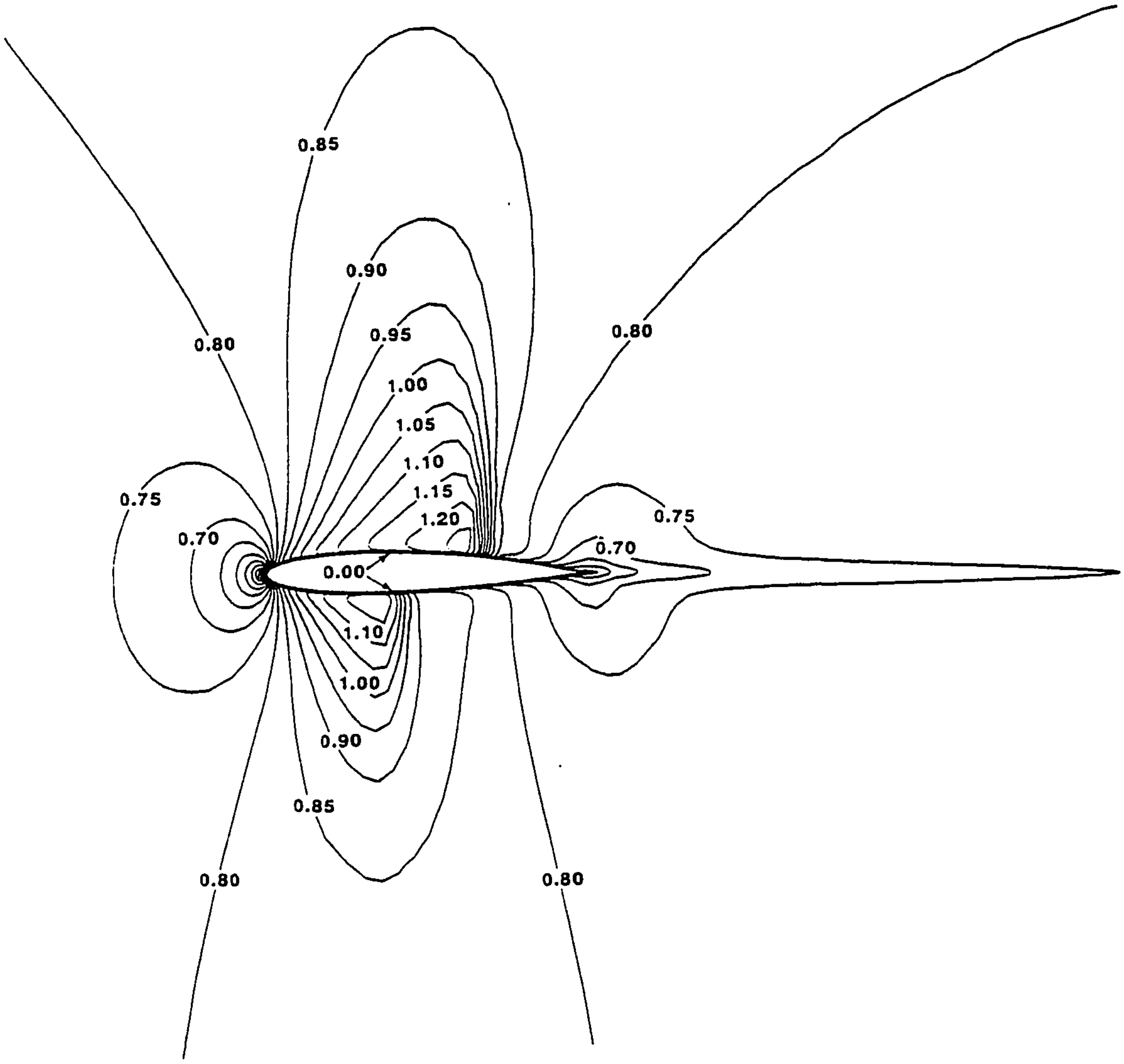


Fig. 10.27 Mach contours of the optimized airfoil (Case 11.a).

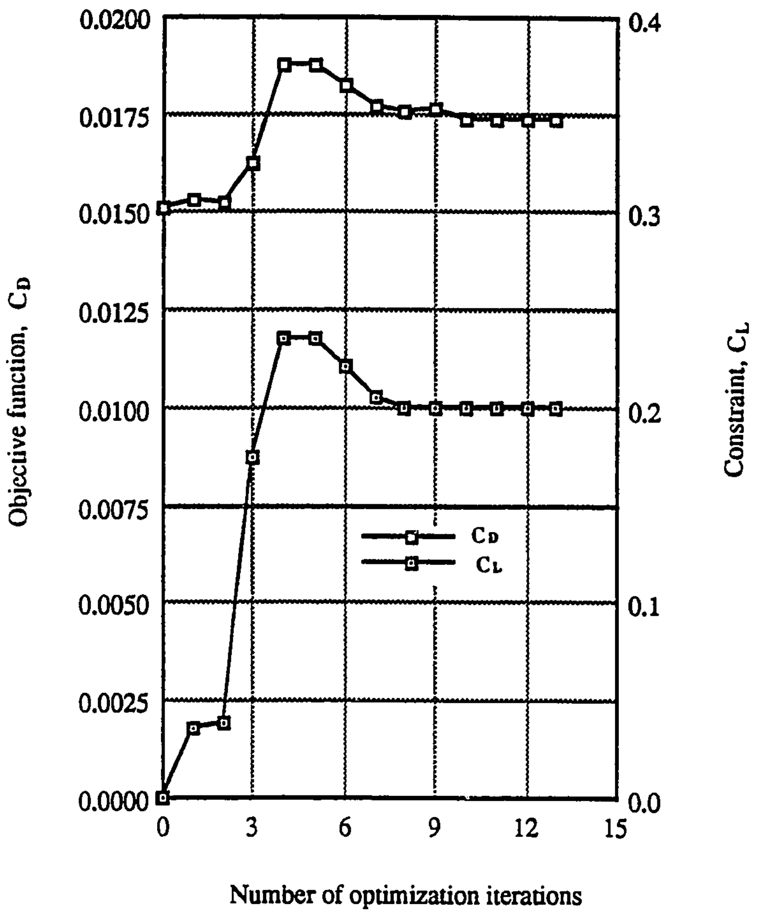


Fig. 10.28 Optimization history of Case 11.a (drag minimization).

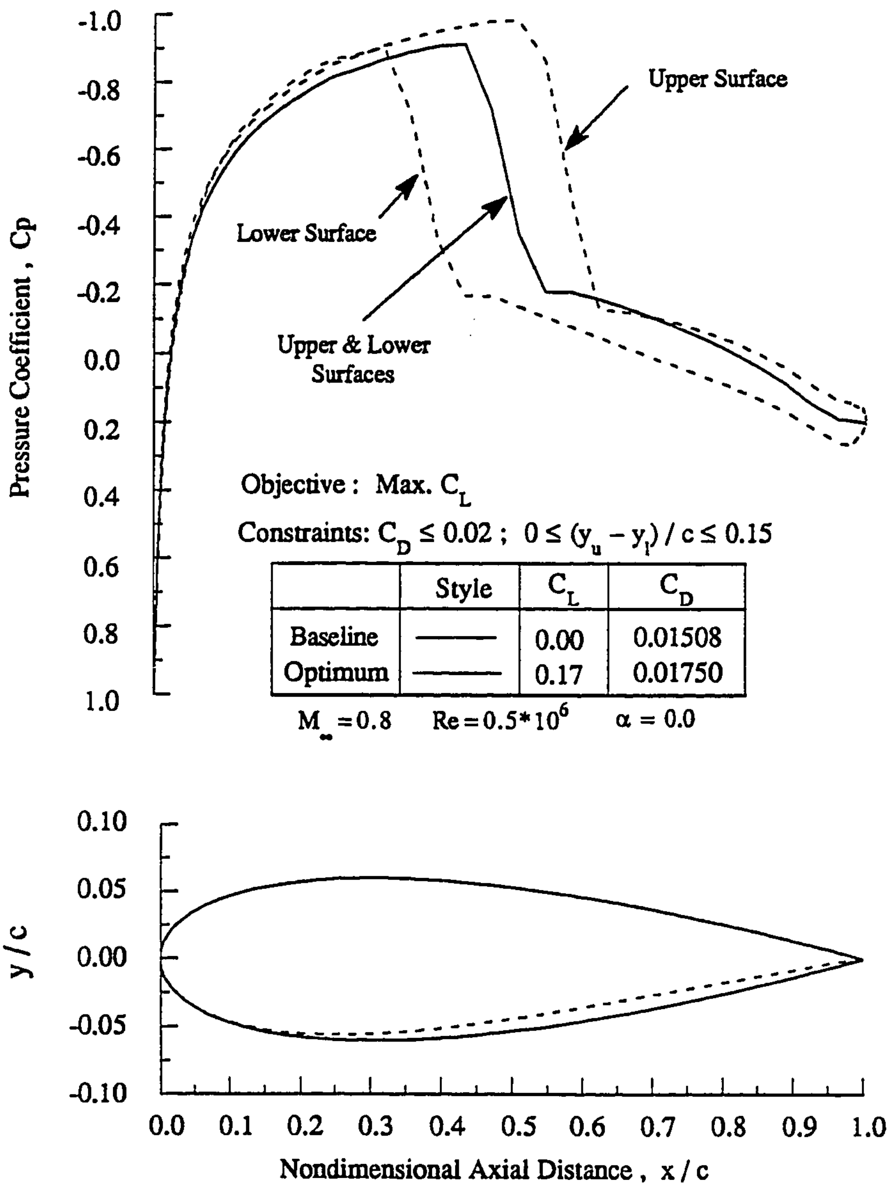


Fig. 10.29 Airfoil shape and distribution of surface pressure coefficients (Case 11.b).

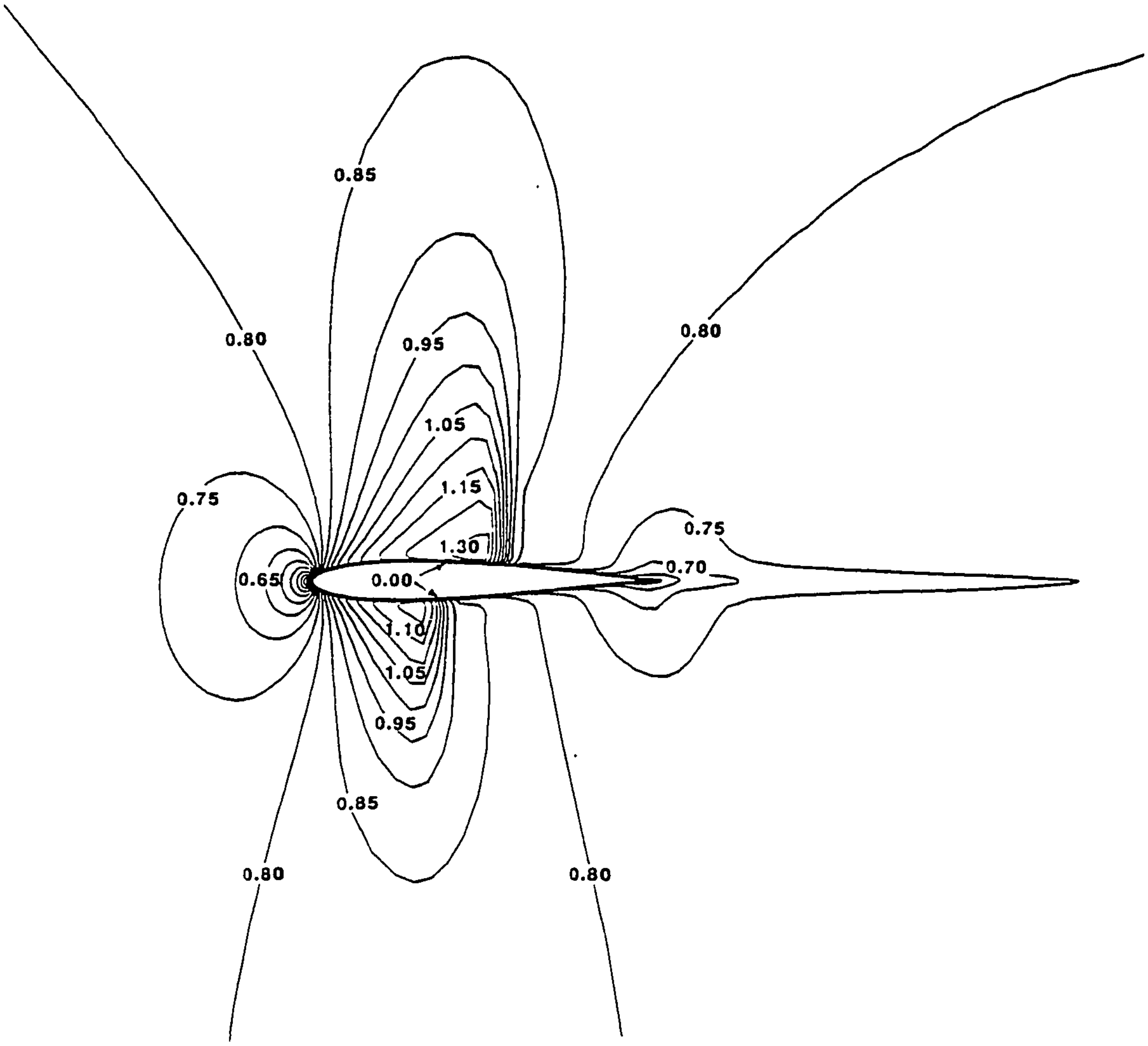


Fig. 10.30 Mach contours of the optimized airfoil (Case 11.b).

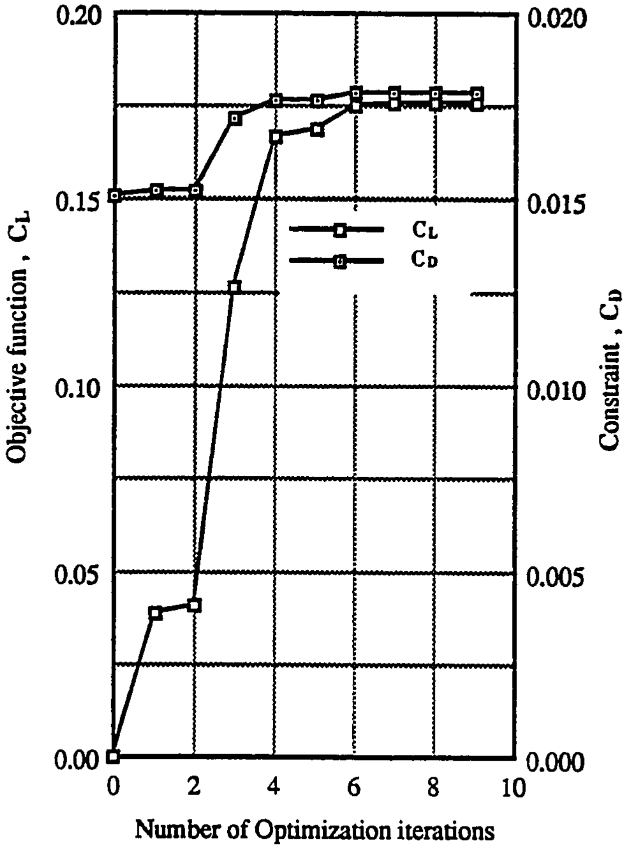


Fig. 10.31 Optimization history of Case 11.b (lift maximization).

Chapter 11

CONCLUSIONS AND RECOMMENDATIONS

To gain a detailed understanding of the complex flowfield features of the nozzle-afterbody configurations, a CFD capability for two-dimensional, viscous, mixing flows was developed and explained in the first part of the dissertation. The mixing of a supersonic jet and a hypersonic flow, during the expansion through an internal–external nozzle with and without multispecies, was analyzed. The surface pressure distributions compared favorably with the available experimental data. Although the computations on fixed grids produced reasonable solutions, adaptation of the grids to the flows improved the resolution by decreasing the computational errors. Comparisons between the case, where the exhaust was simulated by Freon–Argon mixture, and the case, where the exhaust was simulated by air, showed that the heavier simulant gas expands at a slower rate than air, which consequently results in higher forces acting on the afterbody.

Since the flows were turbulent and had high speeds, the diffusive mass transport could be modeled in at least two different ways. Using the computationally more expensive model, which was derived from the complete kinetic theory, could be justified for such mixing only if more accurate and quantitatively better results are required. Also, the results indicated that using an exact equation for the sum of mass ratios being unity to eliminate one of the species continuity equations, was necessary for the mass conservation in mixing flows.

One of the important flowfield features of the planar nozzle-afterbody configuration was the presence of a reversed flow region at the cowl tip due to the large pressure ratio between the nozzle exit plane and the freestream. Another feature of this flowfield was that the mixing occured through a shear layer which deflected upwards at an angle varying from 7° to 10° depending upon the gas that was utilized to model the exhaust flow.

The above CFD capability can be used to advance the conventional wind tunnel studies of scramjet afterbody flows using cold simulant exhaust gases, which in turn can help in the actual design of the lower aft internal–external nozzle of a hypersonic vehicle (e.g., NASP). However, it is suggested for future work on this subject that three–dimensional calculations be performed in order to include the effects of the spanwise expansion inherent in these configurations [14, 22].

Based upon the above results of the nozzle-afterbody configuration, a simplified design optimization problem was defined to serve as one of the demonstrative examples for the new design optimization procedure developed and explained in the second part of the dissertation. This optimization procedure consisted of the following major components: (1) CFD flow analyses; (2) evaluations of the required gradients by means of sensitivity analyses; (3) approximate flow analyses; and (4) a modified optimization algorithm. The present study focused on the efficiencies and the accuracies of component procedures two through four.

A mathematical formulation was developed for the analytical determination of the aerodynamic sensitivity coefficients based on discretized forms of flowfield governing equations. The utilized flowfield governing equations in this part of the study were the 2–D Euler equations and the 2–D thin layer Navier–Stokes equations for a single species. In the analytical approach, two methods, namely, the direct sensitivity method and the adjoint variable method, were assessed with regards to their relative accuracies and computational requirements. Results showed that the two methods yield identical results. However, the direct method is more economical than the adjoint method when the number of design variables is less than the number of adjoint vectors (the number of flow-type constraints plus one). Otherwise, the adjoint variable method is more economical. A comparison between the analytical approach and the traditional finite–difference approach showed that the analytical approach is significantly more computationally economical than the finite–difference approach.

An assessment of the direct and iterative methods, which were deemed most applicable to the large systems of linear equations arising in the present analytical approach, was presented. Based on a single-domain grid, computations were performed using three solvers, namely, a sparse matrix solver, a banded matrix solver, and an iterative solver (GMRES). Comparisons between these solvers indicated that the banded matrix solver is generally the most economical one when the number of right hand sides is large (e.g., a large number of design variables or a large number of adjoint vectors). The results indicated also that, due to large memory requirements, these solvers are not feasible for large two-dimensional problems and, also, inapplicable for any of the three-dimensional problems.

Since large systems could not be solved using the above methods, a principally different approach was proposed. The computational domain was divided into small subdomains and then each subdomain was solved separately. This approach is refered to herein as the *"SADD"* scheme. This scheme was successfully demonstrated by solving a system of linear equations based on a relatively small global computational grid (52x40 cells). The results showed that 40% savings in computer memory requirements can be achieved by using this scheme. However, these savings are expected to be more pronounced for larger two-dimensional problems and for all three-dimensional problems, where the size of the largest subdomain can be kept small. An important feature of this scheme is that it can be easily used to solve systems of linear equations arising in complex–geometry problems (e.g., multi-element airfoils). Currently, the *SADD* scheme is being implemented in a general purpose computer program called *"MAisSA"* [103]. The current version of this program is only applicable to two-dimensional aerodynamic optimization problems and its extension to three-dimensions is a topic of future investigation on this subject.

To replace most of the repetitive *CFD* analyses required in conventional aerodynamic optimization procedures, a flowfield extrapolation method, called *flow prediction method* herein, was developed. The behavior of this method in different flow regimes was investigated by predicting the flowfield of a perturbed scramjet-afterbody configuration and a perturbed transonic *NACA–0012* airfoil. The predicted flowfield and the predicted surface flow properties compared very well with those obtained by performing complete *CFD* analyses for the same perturbed shapes. However, discrepancies between these results started to emerge and grow when large perturbation steps were used. This was attributed to two reasons, namely, the type of gauge function of the Taylor's series used in this study and the truncation errors. Based on a shock–free flowfield solution of the unperturbed shape, the results showed that this method has the capability of predicting shocks and other large gradients generated when the shape is perturbed. However, as the step size of the changes increased, it exhibited increasing inaccuracies due to the above two reasons and eventually produced unacceptable solutions when the changes became too large. Hence, it was concluded that the flow prediction method can accurately represent the flow physics for small changes in design variables.

Additionally, the capability of using the flow prediction method to predict flowfields of successively perturbed shapes, without requiring intermediate *CFD* analysis, was successfully demonstrated. However, the results showed that discrepancies began to appear when too many flow predictions were successively performed. Hence, the method was used successively only for crude but efficient estimates of the flowfield solutions during the one-dimensional search cycle of the optimization process, where estimated values of the objective function and constraints were usually sufficient.

The computational time required by the flow prediction method was found to be an order of magnitude less than that required for a complete CFD analysis for the same configuration. This computational gain is expected to be more pronounced as the number of grid points, which represent the computational domain, increases.

The overall optimization procedure was implemented in a new general purpose computer program *"ADOS"*, and was successfully demonstrated by optimizing the above scramjet nozzle-afterbody configuration to yield a maximum axial thrust force coefficient. It was also demonstrated by optimizing the shapes of two transonic airfoils for minimum drag and maximum lift, respectively.

To demonstrate the feasibility and the accuracy of the present optimization procedure, three different design optimization procedures, including the present procedure, were presented. Comparisons indicated that these procedures yield almost identical optimization results. However, the new procedure, which employed the flow prediction method and used the analytical evaluation of the sensitivity coefficients, was shown to be the most economical one.

Finally, to reduce the amount of expertise for choosing the functions defining the aerodynamic shapes and to alleviate the problems of shape smoothness, the aerodynamic shape in the above optimization problems was defined by a combination of the surface grid points and the relative slopes between these points. The optimization results of nozzle-afterbody configurations showed that by starting from three different arbitrary designs, namely, a flat surface, a concave surface, and a convex surface, the same optimum shape could be obtained. This indicated the generality of the present method and its capability of obtaining optimum shapes irrespective of the initial design shapes. Despite the large number of design variables associated with this approach to defining the shape, it has the advantage of being flexible and easy to implement. A concurrent investigation [104] has shown that the use of Bernstein-Bezier polynomial parameterization of the design surface dramatically reduces the number of design variables, and in turn, improves the overall efficiency of the present procedure.

Although the iterative approximately factored (AF) algorithm used in this study accounted for a large portion of the overall CPU-time of the optimization procedure, it was essential for providing initial CFD solutions to optimization problems. Therefore, as an improvement for the overall efficiency of the procedure, it is suggested for future studies to replace the current CFD solver by another one that uses Newton's method [104].

In order to provide a design capability to help in optimizing the shape of an actual scramjet nozzle-afterbody of a hypersonic vehicle (e.g., NASP), future investigations should include the implementation of suitable turbulence modeling and the application of the present sensitivity analysis methodology to multispecies flowfield governing equations. This methodology, with its promising results and versatility, should subsequently be applied to other CFD algorithms. One such algorithm is Roe's flux-difference splitting. Finally, the extension of the present procedure to three-dimensional problems is necessary for the actual design of a complete hypersonic vehicle as well as other complex–geometry configurations (e.g., a wing-pylon-nacelle configuration).

REFERENCES

1. Oman, R. A., Foreman, K. M., Leng, J., and Hopkins, H. B., "Simulation of Hypersonic Scramjet Exhaust," NASA CR-2494, 1975.

2. Hopkins, H. B., Konopka, W., and Leng, J., "Validation of Scramjet Exhaust Simulation Technique," NASA CR-2688, March 1976.

3. Hopkins, H. B., Konopka, W., and Leng, J., "Validation of Scramjet Exhaust Simulation Technique at Mach 6," NASA CR-3003, March 1979.

4. Engelund, W. C., "Navier–Stokes Analysis of Cold Gas Scramjet Nozzle-Afterbody Flowfields," M. S. Thesis, Old Dominion University, Norfolk , VA, December 1989.

5. Cubbage, J. M. and Monta, W. J., "Parametric Experimental Investigation of a Scramjet Nozzle at Mach 6 With Freon and Argon or Air Used for Exhaust Simulation," NASA TP-3048, February 1991.

6. Monta, W. J., "Pitot Survey of Exhaust Field of a 2-D Scramjet Nozzle with Air or Freon and Argon Used for Exhaust Simulation at Mach 6," NASA-TM-4361, June 1991.

7. Deiwert, G. S., "A Computational Investigation of Low-Speed Nozzle/Afterbody Performance Over Boattails Containing a Centered Propulsive Jet," AIAA Paper 83-0462, January 1983.

8. Hoffman, J. J., Birch, S. F., Hopcroft, R. G., and Holcomb, J. E., "Navier-Stokes Calculations of Rocket Base Flows," AIAA Paper 87-0466, January 1987.

9. Barber, T. J. and Cox, G. B., "Hypersonic Vehicle Propulsion: A CFD Application Case Study," AIAA Paper 88-0475, January 1988.

10. Povinelli, L. A., "Advanced Computational Techniques for Hypersonic Propulsion," NASA TM-102005, September 1989.

11. Harloff, G. J., Lai, H. T., and Nelson, E. S., "Two-Dimensional Viscous Flow Computations of Hypersonic Scramjet Nozzle Flowfields at Design and Off-Design Conditions," AIAA Paper 88-3280, July 1988.

12. Hsu, A. T., "The Effect of Adaptive Grid on Hypersonic Nozzle Flow Calculations," AIAA Paper 89-0006, January 1989.

13. Baysal, O. and Hoffman, W. B., "Computation of Supersonic-Hypersonic Flow Through a Single-Module Scramjet Nozzle," *Proceedings of IMACS 1st International Conference on Computational Physics*, University of Colorado, Boulder, Co., June 1990, pp. 52-59.

14. Baysal, O. and Hoffman, W. B., "Simulation of 3-D Shear Flow Around a Nozzle-Afterbody at High Speeds," *Advances in Numerical Simulation of Turbulent Flows*, [Ed: I. Celik, et al.], FED-Vol. 117, Joint Meeting of ASME/JSME, June 1991, pp. 75-82. Also, *ASME Journal of Fluids Engineering*, Vol. 114, No. 2, June 1992.

15. Lai, H. and Nelson, E., "Comparison of 3–D Computation and Experiment for Non-Axisymmetric Nozzles," AIAA Paper 89-0007, January 1989.

16. Re, R. J. and Leavitt, D. L., " Static Internal Performance of Single-Expansion-Ramp Nozzles with Various Combinations of Internal Geometric Parameters," NASA TM-86270, December 1984.

17. Baysal, O., Engelund, W. C., and Tatum, K. E., "Navier-Stokes Calculations of Scramjet-Afterbody Flowfields," *Advances and Applications in Computational Fluid Dynamics*. [Ed: O. Baysal], FED-Vol. 66, ASME-Winter Anuual Meeting, November 1988, pp. 49-60.

18. Cubbage, J. M., Talcott, N. A. Jr., and Hunt, J. L., "Scramjet Exhaust Technique for Hypersonic Aircraft Design and Aerodynamic Tests," AIAA Paper 77-82, January 1977.

19. Pittman, J. L., "A Mach 6 External Nozzle Experiment with Argon-Freon Exhaust Simulation," SAE Paper No. 892315, Aerospace Technology Conference and Exposition, September 1989.

20. Baysal, O., Eleshaky, M. E., and Engelund, W. C., "Computations of Multispecies Mixing Between a Scramjet Nozzle Flow and Hypersonic Freestream," *AIAA Journal of Propulsion and Power*, Vol. 8, No. 2, March-April 1992, pp. 500–506.

21. Baysal, O., Engelund, W. C., Eleshaky, M. E., and Pittman, J. L., "Adaptive Computations of Multispecies Mixing Between Scramjet Nozzle Flows and Hypersonic Freestream," AIAA Paper 89-0009, January 1989.

22. Baysal, O., Eleshaky, M. E., and Engelund, W. C., "2-D and 3-D Mixing Flow Analyses of a Scamjet-Afterbody Configuration," Paper No. 14, *Proceedings of International Conference on Hypersonic Aerodynamics*, The Royal Aeronautical Society, September 1989.

23. Ray, R., Niggemeir, C., and Erdos, J., "CFD Analysis of 3-D Effects in Scramjet Exhaust Flow Fields," AIAA Paper 88-3262, July 1988.

24. Edwards, T., "The Effect of Exhaust Plume/Afterbody Interaction on Installed Scramjet Performance," AIAA Paper 89-0032, January 1989.

25. Ruffin, S. M., Venkatapathy, E., Keener, E. R., and Nagaraj, N., "Computational Design Aspects of a NASP Nozzle/Afterbody Experiment," AIAA Paper 89-0446, January 1989.

26. Tatum, K., Monta, W., Witte, D., and Walters, R. W., "Analysis of Generic Scramjet External Nozzle Flowfields Employing Simulant Gases," AIAA Paper 90-5242, October 1990.

27. Huebner, L. D. and Tatum, K. E., "Computational and Experimental Aftbody Flow Fields for Hypersonic, Airbreathing Configurations with Scramjet Exhaust Flow Simulation," AIAA Paper 91-1709, June 1991.

28. Drummond, J. P., Rogers, R. C., and Hussaini, M. Y., "A Detailed Numerical Model of a Supersonic Reacting Mixing Layer," AIAA Paper 86-1424, June 1986.

29. Wilke, C. R., "A Viscosity Equation for Gas Mixtures," *Journal of Chemical Physics*, Vol. 18, No. 4, April 1950, pp. 518-519.

30. White, F. M., *Viscous Fluid Flow*, McGraw-Hill Inc., New York, 1974, pp. 28-36.

31. Bird, R. B., Stewart, W. E., and Lightfoot, E. N., *Transport Phenomena*, John Wiley & Sons Inc., New York, 1960.

32. Carpenter, M. H., "The Effects of Finite Rate Chemical Processes on High Enthalpy Nozzle Performance: A Comparison Between "Spark" and "Seagull"," AIAA Paper 88-3157, July 1988.

33. Williams, F. A., *Combustion Theory*, Benjamin-Cummings Publishing, Menlo Park, CA, 1985, pp. 628-649.

34. Baldwin, B. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Flows," AIAA Paper 78-257, January 1978.

35. Cebeci, T., "Calculation of Compressible Turbulent Boundary Layer with Heat and Mass Transfer," AIAA Paper 70-741, June 1970.

36. MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, April 1969.

37. Tannehill, J. C., Holst, T. L., and Rakich, J. V., "Numerical Computation of Two-Dimensional Viscous Blunt Body Flows with an Impinging Shock," AIAA Paper 75-154, January 1975.

38. MacCormack, R. W. and Baldwin, B. S., " A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interactions," AIAA Paper 75-1, July 1975.

39. Nakahashi, k. and Deiwert, G. S., "Adaptive Grid Method With Application to Airfoil Flow," *AIAA Journal*, Vol. 25, No. 4, April 1987, pp. 513-520.

40. Julius, E. H. and Blanchard, D. K., "Computer Program for Solving Laminar, Transitional, or Turbulent Compressible Boundary Layer Equations for Two-Dimensional and Axisymmetric Flow," NASA TM-83207, 1982.

41. Bauer, F., Garabedian, P., and Korn, D., "Supercritical Wing Sections III," *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, New York., 1977.

42. Giles, M. B. and Drela, M., "Two-Dimensional Transonic Aerodynamic Design Methods," *AIAA Journal*, Vol. 25, No. 9, 1987, pp.1199-1206.

43. Drela, M. and Giles, M. B., "ISES: A Two-Dimensional Viscous Aerodynamic Design and Analysis Code," AIAA 87-0424, January 1987.

44. Drela, M. and Giles, M. B., "Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils," *AIAA Journal*, Vol. 25, No. 10, October 1987, pp. 1347-1355.

45. Lee, K. D. and Eyi, S., "Aerodynamic Design Via Optimization," ICAS 90-6.9.1, *17th Congress of International Council of Aeronautical Sciences*, September 1990, pp. 1808-1818.

46. Malone, J. B., Narramore, J. C., and Sankar, L. N., "An Efficient Airfoil Design Method Using the Navier-Stokes Equations," *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, Paper No. 5, 1989.

47. Lee, J. and Mason, W. H., "A Three-Dimensional Inverse Method for Supersonic and Hypersonic Body Design," AIAA Paper 91-3325-CP, September 1991.

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

49. Vanderplaats, G. N., Hicks, R. M., and Murman, E. M., "Application of Numerical Optimization Techniques to Airfoils Design," *NASA Conference on Aerodynamic Analysis Requiring Advanced Computers*, Langley Research Center, NASA SP-347, Part II, 1975, pp. 749-768.

50. Vanderplaats, G. N., "Approximation Concepts for Numerical Airfoil Optimization," NASA TN-1370, 1979.

51. Chen, M. S. and Chow, C. Y., "Numerical Optimization Design of Transonic Airfoils," *Numerical Methods in Laminar and Turbulent Flow*, Vol. 6, Part 1, Pineridge Press, 1989, pp. 905-915.

52. Lee, K. D. and Eyi, S., "Transonic Airfoil Design by Constrained Optimization," AIAA Paper 91-3287-CP, September 1991.

53. Pittman, J. L., "Supersonic Airfoil Optimization," *AIAA Journal*, Vol. 24, No. 12, 1987, pp. 873-879.

54. Rizk, M. H., "Optimization by Updating Design Parameters as CFD Iterative Flow Solutions Evolve," *Symposium on Multidisciplinary Applications of Computational Fluid Dynamics* [Ed: O. Baysal], FED-Vol. 129, pp. 51-62, ASME-Winter Annual Meeting, December 1991.

55. Sorensen, T., "Airfoil Optimization with Efficient Gradient Calculations," *Inverse Design Concepts and Optimization in Engineering Sciences*, [Ed: G. Dulikravich], ICIDES-III, October 1991, pp. 433-441.

56. Drela, M., "XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils," *Low Reynolds Number Aerodynamics* [Ed: T. Mueller], *Lecture Notes in Engineering*, No. 54, Springer-Verlag, June 1989.

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

58. Adelman, H. M. and Haftka, R. T. [Editors], *Sensitivity Analysis in Engineering*, NASA CP-2457, NASA Langely Research Center, Hampton, VA, 1986.

59. Adelman, H. M. and Haftka, R. T., "Sensitivity Analysis of Discrete Structural Systems," *AIAA Journal*, Vol. 24, No. 5, 1986, pp. 823-832.

60. Haftka, R. T. and Gurdal, Z., *Elements of Structural Optimization,* 3rd edition, Kluwer Academic Publishers, Boston, 1992.

61. Sobieski, J. S., "The Case for Aerodynamic Sensitivity Analysis," *Sensitivity Analysis in Engineering*, NASA CP-2457, NASA Langely Research Center, Hampton, VA, 1986, pp. 77-96.

62. Yates, E. C., Jr., "Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady, Nonplanar Lifting Surface Theory," NASA TM-100502, NASA Langely Research Center, Hampton, VA, 1987.

63. Jameson, A., "Aerodynamic Design Via Control Theory," NASA CR-181749, 1988. Also, *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, Paper No. 22, 1989.

64. Lewis, J. C., Peters, G. R., and Agarwal, R. K., "Airfoil Design Via Control Theory Using Euler Equations," *Symposium on Multidisciplinary Applications of Computational Fluid Dynamics* [Ed: O. Baysal], FED-Vol. 129, ASME-Winter Annual Meeting, December 1991, pp. 39-49.

65. Elbanna, H. and Carlson, L., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," *Journal of Aircraft*, Vol. 27, No. 6, June 1990, pp. 507-518. Also, AIAA Paper 89-0532, January 1989.

66. Baysal, O. and Eleshaky, M. E. "Aerodynamic Sensitivity Analysis Methods For the Compressible Euler Equations," *Recent Advances and Applications in CFD*, [Ed: O. Baysal], FED-Vol. 103, ASME-Winter Annual Meeting, November 1990, pp. 191-202. Also, [ASME] *Journal of Fluids Engineering*, Vol. 113, No. 4, December 1991, pp 681-688.

67. Frank, P. D. and Shubin, G. R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamic Design Problem," Technical Report ECA-TR-136-R1, Boeing Computer Services, October 1990.

68. Baysal, O. and Eleshaky, M. E., "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics," AIAA 91-0471, January 1991. Also, *AIAA Journal*, Vol. 30, No. 3, March 1992, pp. 718–725.

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

70. Taylor, A. C. III, Hou, G. W., and Korivi, V. M., "A Methodology for Determining Aerodynamic Sensitivity Derivatives with Respect to Variation of Geometric Shape," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference*, AIAA Paper 91-1101, April 1991, pp. 477-489.

71. Smith, R. E. and Sadrehaghighi, I., "Grid Sensitivity in Airplane Design," *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, Vol. 1, September 1991, pp. 1071-1076.

72. Baysal, O., Fouladi, K., and Lessard,V. R., "Multigrid and Upwind Viscous Flow Solver on 3-D Overlapped/Embedded Grids," *AIAA Journal*, Vol. 29, No. 6, June 1991, pp. 903-910.

73. Fouladi, K. and Baysal, O., "Viscous Simulation Method for Unsteady Flows Past a Configuration with Nonsimilar Multicomponents," *Recent Advances in Computational Fluid Dynamic* [Ed: O. Baysal], FED-Vol. 103, ASME-Winter Annual Meeting, November 1990, pp. 181-190. Also to appear in *Journal of Fluids Engineering*, Vol. 114, No. 3, September 1992.

74. Fouladi, K., Baysal, O., and Newman, J. C. III, "Hybrid Domain Decomposition for Configurations with Multiple Nonsimilar Components," *Proceedings of 5th SIAM Conference on Domain Decomposition Methods for Partial Differential Equations*, May 1991.

75. Snigal, S. and Kane, J. H. "Design Sensitivity Analysis of Boundary-Element Substructures," *AIAA Journal*, Vol. 28, No. 7, July 1990, pp. 1277-1284.

76. Noor, A. K., Kamel, H. A., and Fulton, R. E., "Substructuring Techniques–Status and Projections," *Computers & Structures*, Vol. 8, 1978, pp. 621-632.

77. George, J. A., "Nested Dissection of A Regular Finite Element Mesh," *SIAM Journal of Numerical Analysis*, Vol. 10, No. 2, 1973, pp. 345-363.

78. Sobieski, J. S.,"Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1 , January 1990, pp. 153-160.

79. Venkatakrishnan, V., "Newton Solution of Inviscid and Viscous Problems," *AIAA Journal*, Vol. 27, No. 7, July 1989, pp. 885-891.

80. Beam, R. and Warming, R. F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form," *Journal of Computational Physics*, Vol. 22, September 1976, pp. 87-110.

81. Van Leer, B., "Flux Vector Splitting for the Euler Equations," ICASE Report 82-30, September 1982.

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

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

84. Thibert, J. J., "One Point and Multipoint Design Optimization for Airplane and Helicopter Application," *Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD-R-780, Paper No. 10, November 1990.

85. Nickerson, G. R., Dunn, S. S., and Migdal D., "Optimized Supersonic Exhaust Nozzles for Hypersonic Propulsion," AIAA Paper 88-3161, July 1988.

86. Doty, J. H., Thompson H. D., and Hoffman, J. D., "Optimum Thrust Two-Dimensional NASP Nozzle Study," NASP CR-1069, NASA Langley Research Center, Hampton VA, 1989.

87. Tatum, K. E. and Flanagan, M., Private communications on data to be published, NASA Langley Research Center, 1990.

88. Eleshaky, M. E. and Baysal, O. "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Multidisciplinary Applications of CFD*, [Ed: O. Baysal], FED-Vol. 129, ASME-Winter Annual Meeting, December 1991, pp. 26-36.

89. Rumsey, C. L., Taylor, S. L., Thomas, J. L., and Anderson, W. K., "Application of an Upwind Navier-Stokes Code to Two-Dimensional Transonic Airfoil Flow," AIAA Paper 87-0413, January 1987.

90. Hicks, R. M., Vanderplaats, G. N., Murman, E. M., and King, R. R. "Airfoil Section Drag Reduction at Transonic Speeds by Numerical Optimization," NASA TM X-73,097, February 1976.

91. Vanderplaats, G. N. and Hicks, R. M., "Numerical Airfoil Optimization Using A Reduced Number of Design Coordinates," NASA TM X-73,151, July 1976.

92. Haftka, R. T. and Malkus, D. S., "Calculation of Senstivity Derivatives in Thermal Problems by Finite Differences," *International Journal for Numerical Methods in Engineering*, Vol. 17, 1981, pp. 1811-1821.

93. Noor, A. K. and Whitworth, S. L., "Sensitivity Analysis for Large-Scale Problems," *Sensitivity Analysis in Engineering*, NASA CP-2457, NASA Langley Research Center, Hampton, VA, 1986, pp. 357-374.

94. Pissanetzky, S., *Sparse Matrix Technology*, Academic Press, Inc., New York, 1986, pp. 13,14.

95. Sherman, A. H., "Yale Sparse Matrix User's Guide," Lawrence Livermore Laboratory Report UCID–30114, August 1975.

96. Westlake, J. R., *Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, John Wiley & Sons, Inc., New York, 1968.

97. Martin, R. S. and Wilkinson, J. H., "Solution of Symmetric and Unsymmetric Band Equations and the Calculation of Eigenvectors of Band Matrices," *Numerische Mathematik*, Bd. 9, 1967, pp. 279–301.

98. Thomas, J. L. and Walters, R. W., "Upwind Relaxation Algorithms for The Navier-Stokes Equations," *AIAA Journal*, Vol. 25, No.4, April 1987, pp. 527-534.

99. Venkatakrishnan, V., "Preconditioned Conjugate Gradient Method for The Compressible Navier–Stokes Equations," AIAA Paper 90-0586, January 1990.

100. Vanderplaats, G. N. and Moses, F. "Structural Optimization by Methods of Feasible Directions," *Computers and Structures*, Vol. 3, 1973, pp. 739–755.

101. Vanderplaats, G. N., "ADS-A FORTRAN Program for Automated Design Synthesis," NASA CR-177985, September 1985.

102. *UNICOS© Math and Scientific Library Reference Manual*, Vol. 3, Publication SR–2086 6.0, Cray Research, Inc., January 1991, pp.233–244.

103. Eleshaky, M. E. and Baysal, O., "Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains," *Proceedings of 4th AIAA/NASA/OAI Multidisciplinary Analysis and Optimization Conference*, Cleveland, Ohio, September 1992.

104. Burgreen, G. W., Baysal, O., and Eleshaky, M. E., "Improving the Efficiency of Aerodynamic Shape Optimization Procedures," *Proceedings of 4th AIAA/NASA/OAI Multidisciplinary Analysis and Optimization Conference*, Cleveland, Ohio, September 1992.

105. McBride, B. J., Hiemel, S., Ehlers, J. G., and Gordon, S., "Thermodynamic Properties to 6000 °K for 210 Substances Involving the First 18 Elements," NASA SP-3001, 1963.

106. Svehla, R. A., "Estimated Viscosities and Thermal Conductivities of Gases at High Temperatures," NASA TR-132, 1962.

APPENDICES

Appendix A

PARAMETERS FOR THERMODYNAMIC AND TRANSPORT PROPERTIES

A.1 Tables For Prediction of Thermodynamic and Transport Properties

Table A.1 Coefficients‡ for the specific heat function, Cp(*T*), Eq.(3.11)

Substance	A	B	C	D	E	Temp. range,°K
Ar	2.5000	0.0000	0.0000	0.0000	0.0000	0 – 6000
Fr_{12}	1.6966	0.03460	$-0.428*10^{-4}$	$0.199*10^{-7}$	0.0000	100 – 600
N_2	3.6916	– 0.00133	$0.265*10^{-5}$	$-0.976*10^{-9}$	0.0000	270 – 775
O_2	3.7189	– 0.02516	$0.858*10^{-5}$	$-0.829*10^{-8}$	0.0000	270 – 760

‡Values are found by a curve fit of the data tabulated in Ref. [105]

Table A.2 Thermodynamic quantities

Substance	Enthalpy of formation⁺,*h*° (Joule/kg)	Molecular Weight‡,*w*
Ar	$-1.55146*10^5$	$0.39948*10^2$
Fr_{12}	$-4.18881*10^6$	$1.20914*10^2$
N_2	$3.02736*10^6$	$0.28016*10^2$
O_2	$-2.72918*10^6$	$0.31999*10^2$

⁺Values are evaluated at 298.25 °K.
‡Values are obtained from Ref.[31].

Table A.3 Sutherland Law viscosity parameters for Dilute-Gas

Substance	μ_o, Pa.sec	T_o, °K	S, °K	Temp. range,°K
$Ar^\ddagger$	$2.125*10^{-5}$	273.11_1	144.44	200 - 1500
Fr_{12}^+	$1.192*10^{-5}$	273.11_1	191.29	100 - 1000
$N_2^\ddagger$	$1.663*10^{-5}$	273.11_1	106.66	223 - 1500
$O_2^\ddagger$	$1.191*10^{-5}$	273.11_1	138.88	230 - 200

‡Data are obtained from Ref. [30].

+Data are obtained by a curve fit of the data of Ref. [106].

Table A.4 Sutherland Law thermal conductivity parameters for Dilute-Gas

Substance	k_o, watts $m^{-1}°K^{-1}$	T'_o, °K	S', °K	Temp. range,°K
$Ar^\ddagger$	$1.640*10^{-2}$	273.11_1	150.00	214 - 1500
Fr_{12}^+	$8.536*10^{-2}$	273.11_1	3728.93	100 - 1000
$N_2^\ddagger$	$2.420*10^{-2}$	273.11_1	166.66	208 - 1200
$O_2^\ddagger$	$2.455*10^{-2}$	273.11_1	222.22	217 - 600

‡Data are obtained from Ref. [30].

+Data are obtained by a curve fit of the data of Ref. [106].

Table A.5 Molecular parameters for Dilute-Gas‡

Substance	Collision diameter, σ (Å)	Molecular temperature, T_ε (°K)
Ar	3.542	93.3
Fr_{12}	5.250	253.0
N_2	3.798	71.4
O_2	3.467	106.7

‡Data are obtained from Ref. [106].

A.2 Universal Gas Constant

$R = 8.314_4*10^3$ Joule kg-mole^{-1} °K^{-1}

Appendix B

ADAPTIVE GRIDS

The self-adaptive grid procedure outlined by Nakahashi and Deiwert [39] has been adopted in this study to adapt and restructure the computational grids. The adaptive-grid methodology is based on variational principles. The procedure is analogous to applying tension and torsion spring forces proportional to the local flow gradient at every grid point and finding the equilibrium position of the resulting system of grid points. The 2-D problem of grid adaption is split into a series of one-dimensional problems along the computational coordinate lines (Fig. B.1). The reduced 1-D problem then requires a tridiagonal solver to find the location of grid points along a given coordinate line. The 2-D adaption is achieved by the sequential application of the method in each coordinate direction.

The tension force directs the redistribution of points to the high gradient regions. To maintain smoothness and a measure of orthogonality of grid lines, torsional forces, which relate information between the family of lines adjacent to one another, are introduced. It should be noted that the smoothness and orthogonality constraints are direction-dependent, since they relate only the coordinate lines that are being adapted to the neighboring lines that have already been adapted.

The concept of splitting the 2-D adaption into series of one directional adaptions is shown in Fig. B.1. In this procedure, a flowfield solution is obtained using the unadapted grid shown in Fig. B.1a, and the points in this grid are adapted to the flow solution as follows. The first adaption, as shown in Fig. B.1b, marches from left to right; adaption has already taken place along the first and second lines, and the third line is currently being adapted. The second adaption Fig. B.1c marches from bottom to top, "adapting" the adapted grid. This example is an arbitrary order of adaption and marching directions; any other order will also produce an adapted grid.

Shown in Fig. B.2 is the "current" adaption line in more detail. The equation controlling the redistribution of points along each line consists of two parts. The first part is the one directional approach utilizing tension spring constants. The second part is the addition of the torsion term. The tension spring constant (ω) have the effect of clustering the redistributed points into high gradient regions. The torsion term provides a correction to this redistribution to maintain continuity between sequentially adapted lines.

The "current" line to be adapted is *j*, and the marching is from bottom to top. This implies that lines *(j-1)* and *(j-2)* have already been adapted. In this analysis, *j* is being used as the marching direction and *i* is being used as the adaption direction. (*i*) varies from 1 to *n*, where *n* is the total number of grid points along *j*-line. The arc length at D, $s_{i,j}$, is defined (along *j*) as

$$s_{i+1} = s_i + \sqrt{\left(x_{i+1} - x_i\right)^2 + \left(y_{i+1} - y_i\right)^2} \tag{B.1}$$

A spring-like force is defined to act between each *i* and *i*+1 nodes such that

$$\omega_i \, \Delta \, s_i = K \tag{B.2}$$

where ω_i, the spring constant, is a weighting function based on the flow gradient, and *K* is the resultant force. In order to redistribute the points along a line with the minimum solution error, the weighting function is defined as

$$\omega_i = 1.0 + A \, \bar{f}_i^{B} \tag{B.3}$$

where $\bar{f}_i$ is a function of the gradient of the flow variable, q, i.e.,

$$\bar{f}_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \text{ and } f_i = \frac{\partial q_i}{\partial s} \tag{B.4}$$

Since the adaption is based on a scalar function q, this function can be evaluated as a specified combination of the conservative flow variables Q or any specified fluid properties that represent the flow field.

The A and B are constants directly related to the grid spacing and are chosen to maintain the grid intervals to within the limits $(\Delta s_{min}$ and $\Delta s_{max})$. The value of A is constant throughout the grid adaption and is given by:

$$A = \left(\Delta s_{min} / \Delta s_{max}\right) - 1.0 \tag{B.4}$$

The value of B is computed (by an iterative process) for each j -line to provide

$$specified\ \Delta s_{min} = computed\ \Delta s_{min} \tag{B.5}$$

That is, the computed minimum grid spacing is equal to the specified value. A detailed description of the derivation of B is given by Nakahshi and Deiwert [39].

For this set of equations, with no torsion term, It can be solved directly for Δs_i. Taking the sum of both sides of Eq. (B.2) gives

$$\sum_{l=1}^{np_i} \Delta s_l = s_{max} = K \sum_{l=1}^{np_i} 1/\omega_l \tag{B.6}$$

giving

$$K = s_{max} / \sum_{l=1}^{np_i} 1/\omega_l \tag{B.7}$$

where np_i is the total number of grid intervals along j constant line. Substituting back in Eq. (B.2), we obtain

$$\Delta s_i = s_{max} / \left(\omega_i \sum_{l=1}^{np_i} 1/\omega_l \right) \tag{B.8}$$

The above solution technique is used only along the initial adaption line. Continuing this approach for successive line-by-line adaptions will not necessarily create a mesh that is sufficiently smooth for flowfield analysis. Therefore, to provide a measure of orthogonality and smoothness, the connectivity between adjacent j lines needs to be existed. This is done by introducing the concept of torsion force which is a function of the j, j -1 and j -2 lines. The torsion force is evaluated as

$$F_t = -\tau_i \left(s_i - s'_i \right) \tag{B.9}$$

where the torsion parameter τ defines the magnitude of this torsion force and s' defines its inclination (i.e., orthogonality and smoothness). The derivation and the recommended value of the torsion parameter, τ, is given in Nakahshi and Deiwert [39].

Finally, Eq. (B.2) can be rewritten to represent the force balance, i.e.,

$$\omega_i \left(s_{i+1} - s_i \right) - \omega_{i-1} \left(s_i - s_{i-1} \right) = 0 \tag{B.10}$$

To introduce the torsion force to the system of equations, Eq. (B.9) is added to Eq. (B.10) to obtain,

$$\omega_i \left(s_{i+1} - s_i \right) - \omega_{i-1} \left(s_i - s_{i-1} \right) + \tau_i \left(s'_i - s_i \right) = 0 \tag{B.11}$$

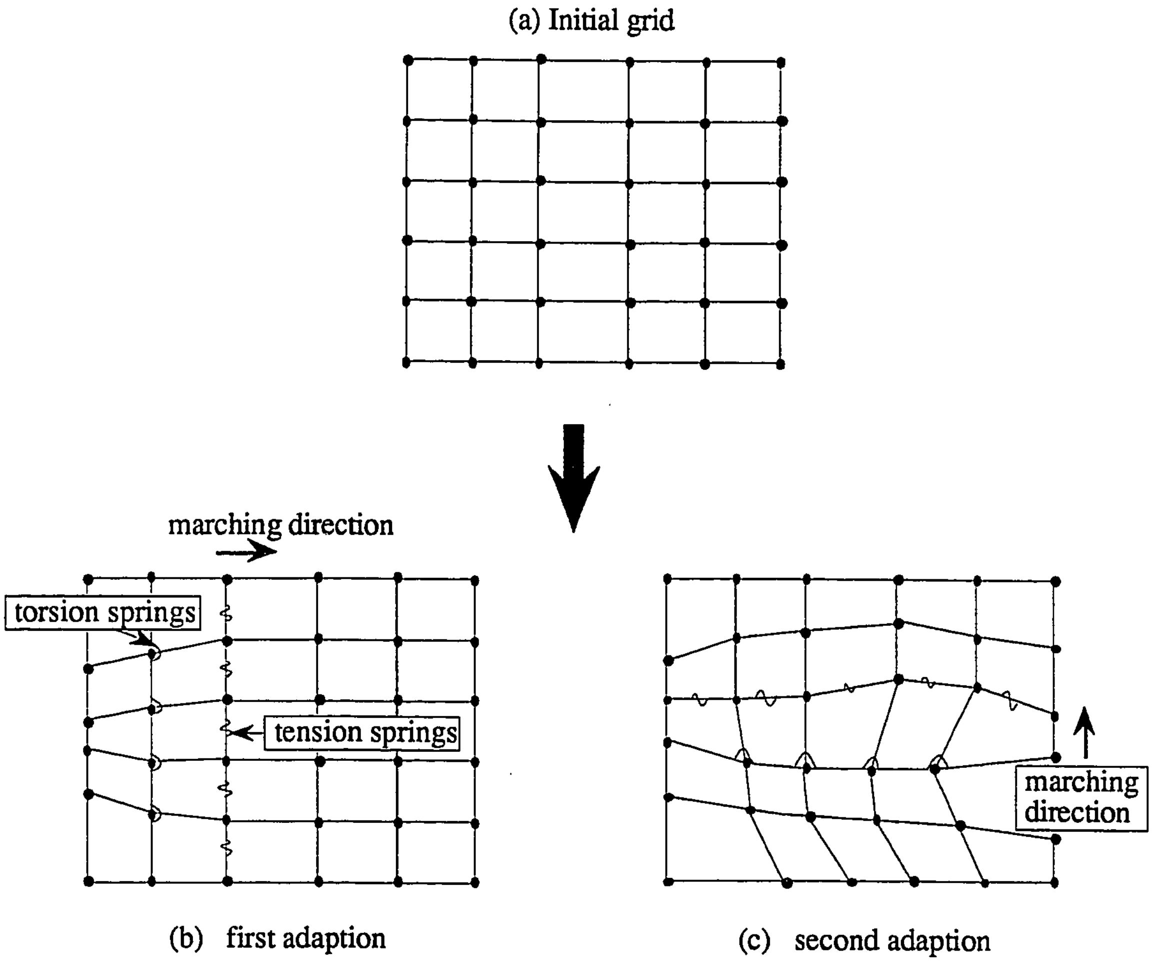
which is rearranged to produce the coefficients of the tridiagonal matrix used to solve for s_i, that is,

$$\omega_{i-1} s_{i-1} - s_i \left(\omega_i - \omega_{i-1} + \tau_i \right) + \omega_i \, s_{i+1} = \tau_i \, s'_i \qquad \text{(B.12)}$$

This equation is written for each interval along the adaption line, producing a system of $(np_i - 1)$ equations. Since s_1 and s_n are known, there are $(np_i - 2)$ row in the matrix. This equation is solved iteratively until,

$$\sum_{i=1}^{np_i} \left| s_i^{(t)} - s_i^{(t-1)} \right| < 10^{-3} * s_{max} \qquad \text{(B.13)}$$

The values of Δs_{min} and Δs_{max} used in the present study are 0.25 and 1.5, respectively. Once the 2-D grid is adapted in both directions to the flowfield gradients, the flowfield solution is interpolated onto the new grid using a second-order, one-dimensional Lagrange interpolation.


(a) Initial grid

(b) first adaption

(c) second adaption

Fig. B.1 Splitting of 2-D adaption into one-directional adaptions.

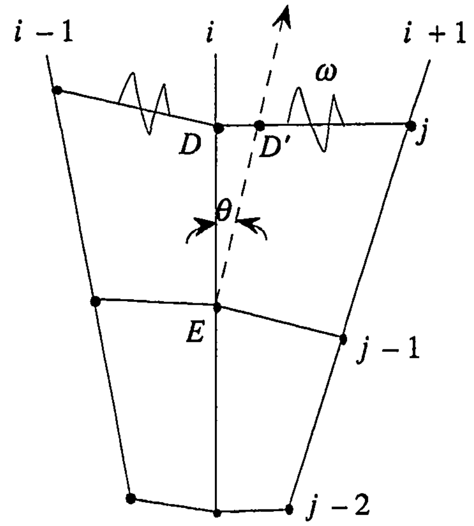


Fig. B.2 Adaption line j, showing tension and torsion springs.

Appendix C

ELEMENTS OF THE LHS OF SENSITIVITY EQUATION

C.1 Obtaining The Coefficients Matrices for The Sensitivity Equation

If the geometric terms ($M_{i\pm 1/2,\, j\pm 1/2}$) in Eq. (8.11) are kept constant, Eq. (8.11) can be rewritten in the following functional form,

$$R_{i,j} = R_{i,j}\left(Q^-_{i\pm 1/2,j},\, Q^+_{i\pm 1/2,j},\, Q^-_{i,j\pm 1/2},\, Q^+_{i,j\pm 1/2},\, Q_{i,j\pm 1/2},\, Q_{i,j}\right) \qquad \text{(C.1)}$$

Differentiating the above equation with respect to the conserved flowfield variables yields the ij^{th} row of the left hand side of Eq. (8.6) as follows:

$$\begin{aligned}
[LHS\ of\ Eq.\ (8.6)]_{i,j} = {} & \frac{\partial R_{i,j}}{\partial Q^-_{i+1/2,j}} \frac{\partial Q^-_{i+1/2,j}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q^-_{i-1/2,j}} \frac{\partial Q^-_{i-1/2,j}}{\partial \overline{X}_D} \\
& + \frac{\partial R_{i,j}}{\partial Q^-_{i,j+1/2}} \frac{\partial Q^-_{i,j+1/2}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q^-_{i,j-1/2}} \frac{\partial Q^-_{i,j-1/2}}{\partial \overline{X}_D} \\
& + \frac{\partial R_{i,j}}{\partial Q^+_{i+1/2,j}} \frac{\partial Q^+_{i+1/2,j}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q^+_{i-1/2,j}} \frac{\partial Q^+_{i-1/2,j}}{\partial \overline{X}_D} \\
& + \frac{\partial R_{i,j}}{\partial Q^+_{i,j+1/2}} \frac{\partial Q^+_{i,j+1/2}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q^+_{i,j-1/2}} \frac{\partial Q^+_{i,j-1/2}}{\partial \overline{X}_D} \\
& + \frac{\partial R_{i,j}}{\partial Q_{i,j+1}} \frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q_{i,j}} \frac{\partial Q_{i,j}}{\partial \overline{X}_D} + \frac{\partial R_{i,j}}{\partial Q_{i,j-1}} \frac{\partial Q_{i,j-1}}{\partial \overline{X}_D}
\end{aligned}$$

(C.2)

To obtain the derivatives of $R_{i,j}$ with respect to $Q_{i,j\pm 1}$,$Q_{i,j}$, and $Q^{\pm}{}_{i\pm 1/2,j}$ in the above equation, the discrete form of the residual given by Eq. (8.11) is used. This yields the following expressions:

$$\frac{\partial R_{i,j}}{\partial Q_{i,j-1}} = \frac{\partial\left[G_v\left(Q_{i,j-1}, Q_{i,j}, M_{i,j-1/2}\right)\right]_{i,j-1/2}}{\partial Q_{i,j-1}} = (A_v)_{i,j} \tag{C.3}$$

$$\frac{\partial R_{i,j}}{\partial Q_{i,j}} = \frac{\partial\left[G_v\left(Q_{i,j-1}, Q_{i,j}, M_{i,j-1/2}\right)\right]_{i,j-1/2}}{\partial Q_{i,j}} - \frac{\partial\left[G_v\left(Q_{i,j+1}, Q_{i,j}, M_{i,j+1/2}\right)\right]_{i,j+1/2}}{\partial Q_{i,j}} = (B_v)_{i,j} \tag{C.4}$$

$$\frac{\partial R_{i,j}}{\partial Q_{i,j+1}} = -\frac{\partial\left[G_v\left(Q_{i,j+1}, Q_{i,j}, M_{i,j+1/2}\right)\right]_{i,j+1/2}}{\partial Q_{i,j+1}} = (C_v)_{i,j} \tag{C.5}$$

$$\frac{\partial R_{i,j}}{\partial Q^-_{i+1/2,j}} = \frac{\partial\left[\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right)\right]}{\partial Q^-_{i+1/2,j}} \tag{C.6}$$

$$\frac{\partial R_{i,j}}{\partial Q^-_{i-1/2,j}} = -\frac{\partial\left[\hat{F}^+\left(Q^-_{i-1/2,j}, M_{i-1/2,j}\right)\right]}{\partial Q^-_{i-1/2,j}} \tag{C.7}$$

$$\frac{\partial R_{i,j}}{\partial Q^+_{i+1/2,j}} = \frac{\partial\left[\hat{F}^-\left(Q^+_{i+1/2,j}, M_{i+1/2,j}\right)\right]}{\partial Q^+_{i+1/2,j}} \tag{C.8}$$

$$\frac{\partial R_{i,j}}{\partial Q^+_{i-1/2,j}} = -\frac{\partial\left[\hat{F}^-\left(Q^+_{i-1/2,j}, M_{i-1/2,j}\right)\right]}{\partial Q^+_{i-1/2,j}} \tag{C.9}$$

$$\frac{\partial R_{i,j}}{\partial Q^-_{i,j+1/2}} = \frac{\partial\left[\hat{G}^+\left(Q^-_{i,j+1/2}, M_{i,j+1/2}\right)\right]}{\partial Q^-_{i,j+1/2}} \tag{C.10}$$

$$\frac{\partial R_{i,j}}{\partial Q^-_{i,j-1/2}} = -\frac{\partial\left[\hat{G}^+\left(Q^-_{i,j-1/2}, M_{i,j-1/2}\right)\right]}{\partial Q^-_{i,j-1/2}} \tag{C.11}$$

$$\frac{\partial R_{i,j}}{\partial Q^+_{i,j+1/2}} = \frac{\partial\left[\hat{G}^-\left(Q^+_{i,j+1/2}, M_{i,j+1/2}\right)\right]}{\partial Q^+_{i,j+1/2}} \tag{C.12}$$

$$\frac{\partial R_{i,j}}{\partial Q^+_{i,j-1/2}} = -\frac{\partial\left[\hat{G}^-\left(Q^+_{i,j-1/2}, M_{i,j-1/2}\right)\right]}{\partial Q^+_{i,j-1/2}} \tag{C.13}$$

The expressions given by Eqs. (C.6)—(C.13) are equivalent to the Jacobian matrices of the split fluxes $\hat{F}^{\pm}$ and $\hat{G}^{\pm}$. Denoting the Jacobian matrices of the split fluxes $\hat{F}^{\pm}$ and $\hat{G}^{\pm}$ by $A^{\pm}$ and $B^{\pm}$, respectively, yields the following form of Eq. (C.2),

$$\begin{aligned}
[LHS\ of\ Eq.\ (8.6)]_{i,j} = {} & A^+_{i+1/2,j}\frac{\partial Q^-_{i+1/2,j}}{\partial \overline{X}_D} - A^+_{i-1/2,j}\frac{\partial Q^-_{i-1/2,j}}{\partial \overline{X}_D} \\
& + B^+_{i,j+1/2}\frac{\partial Q^-_{i,j+1/2}}{\partial \overline{X}_D} - B^+_{i,j-1/2}\frac{\partial Q^-_{i,j-1/2}}{\partial \overline{X}_D} \\
& + A^-_{i+1/2,j}\frac{\partial Q^+_{i+1/2,j}}{\partial \overline{X}_D} - A^-_{i-1/2,j}\frac{\partial Q^+_{i-1/2,j}}{\partial \overline{X}_D} \\
& + B^-_{i,j+1/2}\frac{\partial Q^+_{i,j+1/2}}{\partial \overline{X}_D} - B^-_{i,j-1/2}\frac{\partial Q^+_{i,j-1/2}}{\partial \overline{X}_D} \\
& + (A_v)_{i,j}\frac{\partial Q_{i,j-1}}{\partial \overline{X}_D} + (B_v)_{i,j}\frac{\partial Q_{i,j}}{\partial \overline{X}_D} + (C_v)_{i,j}\frac{\partial Q_{i,j+1}}{\partial \overline{X}_D}
\end{aligned} \tag{C.14}$$

According to Eq. (6.19), the cell-face values of the conserved variables $Q^{\pm}_{i\pm l/2,j}$ and $Q^{\pm}_{i,j\pm l/2}$ can be viewed as functions of $Q_{i\pm l,j\pm l}$ and $Q_{i,j}$. For example, $Q^-_{i+l/2,j}$ and $Q^+_{i+1/2,j}$ can be rewritten in the following functional relationship:

$$Q^-_{i+1/2,j} = Q^-_{i+1/2,j}\left(Q_{i-1,j}, Q_{i,j}, Q_{i+1,j}\right) \tag{C.15}$$

$$Q^{+}_{i+1/2,j} = Q^{+}_{i+1/2,j}\left(Q_{i,j},\, Q_{i+1,j},\, Q_{i+2,j}\right) \tag{C.16}$$

Consequently,

$$\frac{\partial Q^{-}_{i+1/2,j}}{\partial \overline{X}_D} = \frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i-1,j}}\frac{\partial Q_{i-1,j}}{\partial \overline{X}_D} + \frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i,j}}\frac{\partial Q_{i,j}}{\partial \overline{X}_D} + \frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i+1,j}}\frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} \tag{C.17}$$

$$\frac{\partial Q^{+}_{i+1/2,j}}{\partial \overline{X}_D} = \frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i,j}}\frac{\partial Q_{i,j}}{\partial \overline{X}_D} + \frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i+1,j}}\frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} + \frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i+2,j}}\frac{\partial Q_{i+2,j}}{\partial \overline{X}_D} \tag{C.18}$$

Using Eq. (6.19), the derivatives of $Q^{\pm}_{i\pm 1/2,\,j}$ in Eqs. (C.17) and (C.18) are obtained as follows:

$$\begin{aligned}\frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i-1,j}} &= -\frac{\phi}{4}(1-\kappa)[I] \\ &= a_1\end{aligned} \tag{C.19}$$

$$\begin{aligned}\frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i,j}} &= \left(1-\frac{\phi}{2}\kappa\right)[I] \\ &= a_2\end{aligned} \tag{C.20}$$

$$\begin{aligned}\frac{\partial Q^{-}_{i+1/2,j}}{\partial Q_{i+1,j}} &= \frac{\phi}{4}(1+\kappa)[I] \\ &= a_3\end{aligned} \tag{C.21}$$

$$\begin{aligned}\frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i,j}} &= \frac{\phi}{4}(1+\kappa)[I] \\ &= a_3\end{aligned} \tag{C.22}$$

$$\begin{aligned}\frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i+1,j}} &= \left(1-\frac{\phi}{2}\kappa\right)[I] \\ &= a_2\end{aligned} \tag{C.23}$$

$$\frac{\partial Q^{+}_{i+1/2,j}}{\partial Q_{i+2,j}} = -\frac{\phi}{4}(1-\kappa)[I]$$
$$= a_1 \qquad \text{(C.24)}$$

Therefore, Eqs. (C.17) and (C.18) can be rewritten as

$$\frac{\partial Q^{-}_{i+1/2,j}}{\partial \overline{X}_D} = a_1 \frac{\partial Q_{i-1,j}}{\partial \overline{X}_D} + a_2 \frac{\partial Q_{i,j}}{\partial \overline{X}_D} + a_3 \frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} \qquad \text{(C.25)}$$

$$\frac{\partial Q^{+}_{i+1/2,j}}{\partial \overline{X}_D} = a_3 \frac{\partial Q_{i,j}}{\partial \overline{X}_D} + a_2 \frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} + a_1 \frac{\partial Q_{i+2,j}}{\partial \overline{X}_D} \qquad \text{(C.26)}$$

Similarly,

$$\frac{\partial Q^{-}_{i,j+1/2}}{\partial \overline{X}_D} = b_1 \frac{\partial Q_{i,j-1}}{\partial \overline{X}_D} + b_2 \frac{\partial Q_{i,j}}{\partial \overline{X}_D} + b_3 \frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} \qquad \text{(C.27)}$$

$$\frac{\partial Q^{+}_{i,j+1/2}}{\partial \overline{X}_D} = b_3 \frac{\partial Q_{i,j}}{\partial \overline{X}_D} + b_2 \frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} + b_1 \frac{\partial Q_{i,j+2}}{\partial \overline{X}_D} \qquad \text{(C.28)}$$

where,

$$b_1 = -\frac{\phi}{4}(1-\kappa)[I] \qquad \text{(C.29)}$$

$$b_2 = \left(1 - \frac{\phi}{2}\kappa\right)[I] \qquad \text{(C.30)}$$

$$b_3 = \frac{\phi}{4}(1+\kappa)[I] \qquad \text{(C.31)}$$

Similar expressions can be obtained for $\left(\partial Q^{\pm}_{i-1/2,j} / \partial \overline{X}_D\right)$ and $\left(\partial Q^{\pm}_{i,j-1/2} / \partial \overline{X}_D\right)$. Substituting these expressions as well as the expressions given by Eqs. (C.25)-(C.28) into Eq. (C.14) yields,

$$[LHS\ of\ Eq.\ (8.6)]_{i,j} = \overline{D}\frac{\partial Q_{i-2,j}}{\partial \overline{X}_D} + \overline{A}\frac{\partial Q_{i-1,j}}{\partial \overline{X}_D} + \overline{B}\frac{\partial Q_{i,j}}{\partial \overline{X}_D} + \overline{C}\frac{\partial Q_{i+1,j}}{\partial \overline{X}_D} + \overline{E}\frac{\partial Q_{i+2,j}}{\partial \overline{X}_D} + \overline{H}\frac{\partial Q_{i,j-2}}{\partial \overline{X}_D} + \overline{F}\frac{\partial Q_{i,j-1}}{\partial \overline{X}_D} + \overline{G}\frac{\partial Q_{i,j+1}}{\partial \overline{X}_D} + \overline{I}\frac{\partial Q_{i,j+2}}{\partial \overline{X}_D} \quad \text{(C.32)}$$

where,

$$\overline{D} = -a_1 A^+_{i-1/2,j} \quad \text{(C.33)}$$

$$\overline{A} = a_1 A^+_{i+1/2,j} - a_2 A^+_{i-1/2,j} - a_3 A^-_{i-1/2,j} \quad \text{(C.34)}$$

$$\overline{B} = a_2\left(A^+_{i+1/2,j} - A^-_{i-1/2,j}\right) - a_3\left(A^+_{i-1/2,j} - A^-_{i+1/2,j}\right) + b_2\left(B^+_{i,j+1/2} - B^-_{i,j-1/2}\right) - b_3\left(B^+_{i,j-1/2} - B^-_{i,j+1/2}\right) + (B_v)_{i,j} \quad \text{(C.35)}$$

$$\overline{C} = a_3 A^+_{i+1/2,j} + a_2\ A^-_{i+1/2,j} - a_1\ A^-_{i-1/2,j} \quad \text{(C.36)}$$

$$\overline{E} = a_1\ A^-_{i+1/2,j} \quad \text{(C.37)}$$

$$\overline{H} = -b_1\ B^+_{i,j-1/2} \quad \text{(C.38)}$$

$$\overline{F} = b_1\ B^+_{i,j+1/2} - b_2\ B^+_{i,j-1/2} - b_3\ B^-_{i,j-1/2} + (A_v)_{i,j} \quad \text{(C.39)}$$

$$\overline{G} = b_3\ B^+_{i,j+1/2} + b_2\ B^-_{i,j+1/2} - b_1\ B^-_{i,j-1/2} + (C_v)_{i,j} \quad \text{(C.40)}$$

$$\overline{I} = b_1\ B^-_{i,j+1/2} \quad \text{(C.41)}$$

The coefficients $\overline{A}$ through $\overline{I}$ are 4x4 blocks for two-dimensional equations and are functions of both the Jacobian matrices of the split fluxes, $A^{\pm}$ and $B^{\pm}$, and the Jacobians of the viscous fluxes. It is noteworthy that these coefficients are very similar to those used in the upwind relaxation algorithm of Thomas and Walters [98]. However, the Jacobians of the viscous fluxes in the present formulation are different than those appeared in Ref. [98]. This is due to the fact that in the present formulation the viscosities at the cell-interfaces are considered as functions of the conserved variables to maintain the consistency between both sides of the sensitivity equation (Eq. (8.6)).

C.2 An Example of The Boundary Conditions of The Sensitivity Equation

To provide a consistent treatment for the boundary conditions, the same boundary conditions used in the flow analysis is used in the sensitivity equations. For example, to apply inviscid wall boundary condition to the sensitivity equation, Eq. (6.35) is used. Hence, like the flow analysis, the boundary condition at $j = 1$ is applied directly to the cell-interface primitive variables at the wall as follows,

$$q^{\pm}_{i,1/2} = q_{i,o} \tag{C.42}$$

$$q^{-}_{i,3/2} = q_{i,1} + \frac{\phi}{4}\{2(1-\kappa)(q_{i,1} - q_{i,o}) + (1+\kappa)(q_{i,2} - q_{i,1})\} \tag{C.43}$$

$$q^{+}_{i,3/2} = q_{i,2} - \frac{\phi}{4}\{(1+\kappa)(q_{i,2} - q_{i,1}) + (1+\kappa)(q_{i,3} - q_{i,2})\} \tag{C.44}$$

where,

$$\rho_{i,o} = \rho_{i,1} \tag{C.45}$$

$$u_{i,o} = u_{i,1} - \hat{\eta}_{x_{i,1/2}} V_{i,1/2} \tag{C.46}$$

$$v_{i,o} = v_{i,1} - \hat{\eta}_{y_{i,1/2}} V_{i,1/2} \tag{C.47}$$

$$P_{i,o} = P_{i,1} \tag{C.48}$$

Consequently, the revised expression for Eq. (C.27) at $j = 1$ is given by

$$\frac{\partial Q^{-}_{i,3/2}}{\partial \overline{X}_D} = 2b_1 \frac{\partial Q_{i,o}}{\partial \overline{X}_D} + b_2 \frac{\partial Q_{i,1}}{\partial \overline{X}_D} + b_3 \frac{\partial Q_{i,2}}{\partial \overline{X}_D} \tag{C.49}$$

and at the wall,

$$\frac{\partial Q^-_{i,1/2}}{\partial \overline{X}_D} = \frac{\partial Q_{i,o}}{\partial \overline{X}_D} \tag{C.50}$$

Similarly,

$$\frac{\partial Q^+_{i,1/2}}{\partial \overline{X}_D} = \frac{\partial Q_{i,o}}{\partial \overline{X}_D} \tag{C.51}$$

Using Eqs. (C.49)—(C.51) in Eq. (C.14) yields the following revised expression for Eq. (C.32)

$$\begin{aligned}[LHS\ of\ Eq.\ (8.6)]_{i,1} = &\ \overline{D}\frac{\partial Q_{i-2,1}}{\partial \overline{X}_D} + \overline{A}\frac{\partial Q_{i-1,1}}{\partial \overline{X}_D} + \overline{C}\frac{\partial Q_{i+1,1}}{\partial \overline{X}_D} \\ &+ \overline{E}\frac{\partial Q_{i+2,1}}{\partial \overline{X}_D} + \left[\overline{F}\frac{\partial Q_{i,o}}{\partial Q_{i,1}} + \overline{B}\right]\frac{\partial Q_{i,1}}{\partial \overline{X}_D} \\ &+ \overline{G}\frac{\partial Q_{i,2}}{\partial \overline{X}_D} + \overline{I}\frac{\partial Q_{i,3}}{\partial \overline{X}_D}\end{aligned} \tag{C.52}$$

where the terms $\overline{H}$, $\overline{F}$, and $\overline{B}$ are revised using Eqs.(C.49)—(C.51) to yield the following forms,

$$\begin{aligned}\overline{B} = &\ a_2\left(A^+_{i+1/2,j} - A^-_{i-1/2,j}\right) - a_3\left(A^+_{i-1/2,j} - A^-_{i+1/2,j}\right) \\ &+ b_2 B^+_{i,j+1/2} + b_3 B^-_{i,j+1/2}\end{aligned} \tag{C.53}$$

$$\overline{F} = 2b_1\ B^+_{i,j+1/2} - B^+_{i,j-1/2} - B^-_{i,j-1/2} \tag{C.54}$$

$$\overline{H} = 0 \tag{C.55}$$

In Eq. (C.52), the term $\left(\partial Q_{i,o}/\partial Q_{i,1}\right)$ is given by

$$\frac{\partial Q_{i,o}}{\partial Q_{i,1}} = \frac{\partial Q_{i,o}}{\partial q_{i,o}}\frac{\partial q_{i,o}}{\partial q_{i,1}}\frac{\partial q_{i,1}}{\partial Q_{i,1}} \tag{C.56}$$

where the transformation matrices $\left[\partial Q_{i,o} / \partial q_{i,o}\right]$ and $\left[\partial q_{i,1} / \partial Q_{i,1}\right]$ are used to facilitate the conversion from conserved variables to primitive variables, and vice versa. Finally, the derivatives of $q_{i,o}$ with respect to $q_{i,1}$ in Eq. (C.56) is given by

$$\frac{\partial q_{i,o}}{\partial q_{i,1}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1-\left(\hat{\eta}_x\right)^2_{i,1/2} & -\left(\hat{\eta}_x\,\hat{\eta}_y\right)_{i,1/2} & 0 \\ 0 & -\left(\hat{\eta}_x\,\hat{\eta}_y\right)_{i,1/2} & 1-\left(\hat{\eta}_y\right)^2_{i,1/2} & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \tag{C.57}$$

Appendix D

THE DERIVATIVES OF THE SPLIT-FLUX $\hat{F}^+$ WITH RESPECT TO GEOMETRIC TERMS

In Eq. (8.14), $M_{i+1/2,j}$ represents a vector that contains the geometric terms $\hat{\xi}_x$,$\hat{\xi}_y$, and $\left(|\nabla\xi|/J\right)$ evaluated at the cell-interface $(i+1/2, j)$and it is given below.

$$M_{i+1/2,j} = \begin{Bmatrix} \dfrac{|\nabla\xi|}{J} \\ \hat{\xi}_x \\ \hat{\xi}_y \end{Bmatrix}_{i+1/2,j} \tag{D.1}$$

Therefore,

$$\frac{\partial\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial M_{i+1/2,j}} = \begin{Bmatrix} \dfrac{\partial\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(|\nabla\xi|/J\right)} \\ \dfrac{\partial\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\hat{\xi}_x} \\ \dfrac{\partial\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\hat{\xi}_y} \end{Bmatrix}^T_{i+1/2,j} \tag{D.2}$$

The elements of the above expression (Eq. (D.2)) are obtained for both subsonic and supersonic flows as follows: For subsonic flow when $M_\xi < 1$,

$$\hat{F}^+\left(Q^-_{i+1/2,j}, M_{i+1/2,j}\right) = \left(\frac{|\nabla\xi|}{J}\right)_{i+1/2,j} \begin{Bmatrix} f^+_{\text{mass}} \\ f^+_{\text{mass}}\left\{\left[\hat{\xi}_x(-\bar{u}+2a)/\gamma\right] + u\right\} \\ f^+_{\text{mass}}\left\{\left[\hat{\xi}_y(-\bar{u}+2a)/\gamma\right] + v\right\} \\ f^+_{\text{energy}} \end{Bmatrix}_{i+1/2,j} \tag{D.3}$$

where

$$f^{+}_{\text{mass}} = \rho a \left(M_{\xi} + 1\right)^{2} / 4 \qquad \text{(D.4)}$$

$$f^{+}_{\text{energy}} = f^{+}_{\text{mass}} \left[\frac{(1-\gamma)\bar{u}^{2} + 2(\gamma-1)\bar{u}a + 2a^{2}}{\left(\gamma^{2} - 1\right)} + \frac{\left(u^{2} + v^{2}\right)}{2} \right] \qquad \text{(D.5)}$$

$$\bar{u} = \hat{\xi}_{x} u + \hat{\xi}_{y} v \qquad \text{(D.6)}$$

$$M_{\xi} = \frac{\bar{u}}{a} \qquad \text{(D.7)}$$

and $\hat{\xi}_{x}$,$\hat{\xi}_{y}$ are the directional cosines. Consequently, the elements of Eq. (D.2) are ,

$$\frac{\partial \hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(|\nabla\xi| / J\right)_{i+1/2,j}} = \begin{pmatrix} f^{+}_{\text{mass}} \\ f^{+}_{\text{mass}} \left\{ \left[\hat{\xi}_{x}\left(-\bar{u} + 2a\right) / \gamma \right] + u \right\} \\ f^{+}_{\text{mass}} \left\{ \left[\hat{\xi}_{y}\left(-\bar{u} + 2a\right) / \gamma \right] + v \right\} \\ f^{+}_{\text{energy}} \end{pmatrix}_{i+1/2,j} \qquad \text{(D.8)}$$

$$\frac{\partial \hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(\hat{\xi}_{x}\right)_{i+1/2,j}} =$$

$$\left(\frac{|\nabla\xi|}{J}\right)_{i+1/2,j} \begin{pmatrix} \dfrac{\partial f^{+}_{\text{mass}}}{\partial\hat{\xi}_{x}} \\ \dfrac{\partial f^{+}_{\text{mass}}}{\partial\hat{\xi}_{x}} \left\{ \left[\hat{\xi}_{x}\left(-\bar{u} + 2a\right) / \gamma \right] + u \right\} + f^{+}_{\text{mass}} \left\{ \left(-\bar{u} + 2a\right) / \gamma + \hat{\xi}_{x}\left(-u\right) / \gamma \right\} \\ \dfrac{\partial f^{+}_{\text{mass}}}{\partial\hat{\xi}_{x}} \left\{ \left[\hat{\xi}_{y}\left(-\bar{u} + 2a\right) / \gamma \right] + v \right\} + f^{+}_{\text{mass}} \left\{ \hat{\xi}_{y}\left(-u\right) / \gamma \right\} \\ \dfrac{\partial f^{+}_{\text{energy}}}{\partial\hat{\xi}_{x}} \end{pmatrix}_{i+1/2,j}$$

(D.9)

$$\frac{\partial \widehat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(\widehat{\xi_y}\right)_{i+1/2,j}} =$$

$$\left(\frac{|\nabla\xi|}{J}\right)_{i+1/2,j} \left(\begin{array}{c} \dfrac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_y}} \\ \dfrac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_y}} \left\{\left[\widehat{\xi_x}\,(-\overline{u}+2a)/\gamma\right]+u\right\}+f^{+}_{\text{mass}}\left[\widehat{\xi_x}\,(-v)/\gamma\right] \\ \dfrac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_y}} \left\{\left[\widehat{\xi_y}\,(-\overline{u}+2a)/\gamma\right]+v\right\}+f^{+}_{\text{mass}}\left[(-\overline{u}+2a)/\gamma+\widehat{\xi_y}\,(-v)/\gamma\right] \\ \dfrac{\partial f^{+}_{\text{energy}}}{\partial\widehat{\xi_y}} \end{array} \right)_{i+1/2,j}$$

(D.10)

The derivatives in Eqs. (D.8)—(D.10) are given by,

$$\left(\frac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_x}}\right)_{i+1/2,j} = \left[\rho u\,\left(M_{\xi}+1\right)\ /2\right]_{i+1/2,j}$$

(D.11)

$$\left(\frac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_y}}\right)_{i+1/2,j} = \left[\rho v\,\left(M_{\xi}+1\right)\ /2\right]_{i+1/2,j}$$

(D.12)

$$\left(\frac{\partial f^{+}_{\text{energy}}}{\partial\widehat{\xi_x}}\right)_{i+1/2,j} = \left(\frac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_x}}\right)_{i+1/2,j} \left[\frac{(1-\gamma)\overline{u}^{2}+2\,(\gamma-1)\,\overline{u}\,a+2a^{2}}{\left(\gamma^{2}-1\right)}+\frac{\left(u^{2}+v^{2}\right)}{2}\right]_{i+1/2,j}$$
$$+\left\{f^{+}_{\text{mass}}\left[\frac{2\,(1-\gamma)\,\overline{u}\,u+2\,(\gamma-1)\,u\,a}{\left(\gamma^{2}-1\right)}\right]\right\}_{i+1/2,j}$$

(D.13)

$$\left(\frac{\partial f^{+}_{\text{energy}}}{\partial\widehat{\xi_y}}\right)_{i+1/2,j} = \left(\frac{\partial f^{+}_{\text{mass}}}{\partial\widehat{\xi_y}}\right)_{i+1/2,j} \left[\frac{(1-\gamma)\overline{u}^{2}+2\,(\gamma-1)\,\overline{u}\,a+2a^{2}}{\left(\gamma^{2}-1\right)}+\frac{\left(u^{2}+v^{2}\right)}{2}\right]_{i+1/2,j}$$
$$+\left\{f^{+}_{\text{mass}}\left[\frac{2\,(1-\gamma)\,\overline{u}\,v+2\,(\gamma-1)\,v\,a}{\left(\gamma^{2}-1\right)}\right]\right\}_{i+1/2,j}$$

(D.14)

Next, for supersonic flow when $M_\xi \geq 1$,

$$\hat{F}^{+}=\hat{F} \qquad \hat{F}^{-}=0$$

hence,

$$\hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)=\frac{1}{J}\begin{Bmatrix}\rho U\\ \rho U u + \xi_x P\\ \rho U v + \xi_y P\\ (e+P)U\end{Bmatrix}_{i+1/2,j} \tag{D.15}$$

Alternatively,

$$\hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)=\left(\frac{|\nabla \xi|}{J}\right)_{i+1/2,j}\begin{Bmatrix}\rho \bar{u}\\ \left(\rho\bar{u}u + \hat{\xi}_x P\right)\\ \left(\rho\bar{u}v + \hat{\xi}_y P\right)\\ (e+P)\bar{u}\end{Bmatrix}_{i+1/2,j} \tag{D.16}$$

Consequently, the elements of Eq. (D.2) are

$$\frac{\partial\hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(|\nabla\xi|/J\right)_{i+1/2,j}}=\begin{Bmatrix}\rho \bar{u}\\ \left(\rho\bar{u}u + \hat{\xi}_x P\right)\\ \left(\rho\bar{u}v + \hat{\xi}_y P\right)\\ (e+P)\bar{u}\end{Bmatrix}_{i+1/2,j} \tag{D.17}$$

$$\frac{\partial\hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(\hat{\xi}_x\right)_{i+1/2,j}}=\left(\frac{|\nabla\xi|}{J}\right)_{i+1/2,j}\begin{Bmatrix}\rho u\\ \rho u^2+P\\ \rho uv\\ (e+P)u\end{Bmatrix}_{i+1/2,j} \tag{D.18}$$

$$\frac{\partial\hat{F}^{+}\left(Q^{-}_{i+1/2,j}, M_{i+1/2,j}\right)}{\partial\left(\hat{\xi}_y\right)_{i+1/2,j}}=\left(\frac{|\nabla\xi|}{J}\right)_{i+1/2,j}\begin{Bmatrix}\rho v\\ \rho vu\\ \rho v^2+P\\ (e+P)v\end{Bmatrix}_{i+1/2,j} \tag{D.19}$$

BIOGRAPHY

Mohamed E. Eleshaky was born on October 7, 1960 in Alexandria, Egypt. He began his undergraduate studies in 1977. Throughout his undergraduate studies, he was awarded undergraduate scholarships and in the summer of 1980, he was awarded practical training in Braunau, Austria. He received his Bachelor of Science degree in Mechanical Engineering from Alexandria University, Egypt, in June 1982, graduating at the top of his class. He was honored in October 1982 by the Egyptian Engineering Syndicate as the top of 60 other students having graduated with the grade: distinction with degree of honor (Equivalent to 4.0 on a 4.0 scale). Immediately thereafter he was appointed as a teaching assistant in the Mechanical Engineering department in the same university while conducting his graduate studies. He received a Master of Science degree in Mechanical Engineering in June 1987 from Alexandria University, Egypt. On May 12, 1988 he married Dr. Maissa Zikry (DDS). On May 15, 1988, he began his Ph. D. studies in Mechanical Engineering Department of Old Dominion University. The author was involved in research work pertaining to applied CFD, at NASA Langley Research Center, Hampton, Virginia, U.S.A. He was honored in 1990 by the American Honor Society of Phi Kappa Phi. In April 1992, he received a certificate of recognition from NASA for the disclosure of an inventive contribution entitled, "Aerodynamic Design Optimization Using Sensitivity Analysis and Computational Fluid Dynamics." The author was involved in the publications given by Refs. [20,21,22,66,68,88,103,104].