

stellingen ontbreken

**TR diss
2863**

**MULTI-POINT AERODYNAMIC DESIGN
BY OPTIMIZATION**

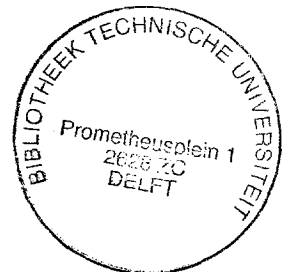
672641
3191479
TR diss 2863

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. ir. K.F. Wakker
in het openbaar te verdedigen ten overstaan van een commissie,
door het College van Dekanen aangewezen,
op woensdag 11 december 1996 te 16.00 uur

door

Bambang Irawan SOEMARWOTO
Sarjana Teknik Mesin Institut Teknologi Bandung
geboren te Yogyakarta, Indonesië



Dit proefschrift is goedgekeurd door de promotoren:

Prof. ir. J.W. Slooff

Prof. dr. ir. J.L. van Ingen

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter

Prof. ir. J.W. Slooff, promotor, TU Delft

Prof. dr. ir. J.L. van Ingen, promotor, TU Delft

Prof. dr. ir. H.W.M Hoeijmakers, TU Delft

Prof. dr. ir. J.W. Boerstoel, TU Delft

Prof. dr. A. Jameson, Princeton University (U.S.A.)

Prof. ir. H.O. Diran, Institut Teknologi Bandung (Indonesia)

Prof. dr. ir. H. Djodjodhardjo, Institut Teknologi Bandung (Indonesia)

The research described in this thesis has been performed by the author at the Department of Theoretical Aerodynamics of the National Aerospace Laboratory (NLR) in Amsterdam.

The research has been financially supported partly by the National Aerospace Laboratory (NLR), the Indonesian Ministry of Education and Culture, and the Dutch Ministry of Education and Science.

Copyright © by B.I. Soemarwoto, Delft, the Netherlands.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of the author or the publisher.

Publisher:

Delft University of Technology

Faculty of Aerospace Engineering

P.O. Box 5058

2600 GB Delft

the Netherlands

ISBN 90-5623-051-4

Summary

The thesis describes the development of a design methodology for obtaining the optimal solution of multi-point aerodynamic design problems of a single airfoil in two-dimensional flow. The design method treats the problem as a (constrained) optimization problem. This is formulated in terms of aerodynamic objective functionals to be minimized, subject to aerodynamic and/or geometric constraint function(al)s which have to be satisfied.

The flow models used in the development and application examples of the design methodology are the incompressible potential flow governed by the Laplace equation, inviscid compressible flow governed by the Euler equations, and viscous compressible flow governed by the Reynolds-averaged Navier-Stokes (RANS) equations. Advantage is taken of existing flow solvers for the analysis computation of airfoils in the abovementioned flow models.

The design method makes use of a gradient-based optimization algorithm for obtaining the optimal solution through employing an existing optimization routine. The gradient of an aerodynamic functional is computed in an efficient manner by means of the variational method. This implies that for each aerodynamic functional, a distinct adjoint problem is to be formulated. The solution of the adjoint problem provides the so-called (vector of) Lagrange multipliers, which are used for evaluating the gradient formula.

The formulation of the adjoint problem, the construction of an adjoint solver for obtaining the Lagrange multipliers and the gradient, and the integration of the adjoint solver with an existing flow solver and optimization routine form the main subject of this thesis.

Inverse problems are considered for aerodynamic design using the Laplace equation. A methodology is discussed for obtaining well-posedness by introducing free target velocity parameters. The well-posedness is defined in terms of the Lighthill constraints. The objective of the investigation is to construct an appropriate automatic procedure for modifying an unrealistic target velocity distribution such that a useable airfoil geometry can be found.

The feasibility of the optimization methodology is also investigated for aerodynamic design based on the Euler equations. A reconstruction type of inverse problem is addressed as a means to check the accuracy of the computed gradients. To demonstrate the capability of the methodology for solving practical aerodynamic problems, single-point test cases for (wave) drag reduction problems under aerodynamic and geometric constraints are considered. The constraints considered are:

- An equality constraint representing a specified (design) lift coefficient.
- Constraints with lower-bound on pitching moment, cross-sectional area, trailing edge included angle, and leading edge radius.

The feasibility of the optimization methodology is finally investigated for aerodynamic design based on the Reynolds-Averaged Navier-Stokes (RANS) equations. The investigation addresses the same test cases as that defined for design studies using the Euler equations. The results, obtained from design using the Euler and the RANS equations, are compared.

Based on the knowledge gained from the single-design-point studies using the Euler and the RANS Equations, a multi-point aerodynamic design method is developed. The suitability of two multi-objective optimization strategies is investigated, namely: (i) the method of the sum of weighted objectives, and (ii) the method of fuzzy optimization. The multi-point design test cases are defined by combining single-point test cases, with the same type of aerodynamic and geometric constraints.

From the results obtained in the investigation it is concluded that the optimization methodology described in this thesis is suitable for solving practical single-point and multi-point aerodynamic design problems of a single airfoil in two-dimensional flow. The importance of the methodology for practical design purposes is particularly due to the possibility of directly incorporating both aerodynamic and geometric constraints, and due to the possibility to search for compromises and to perform trade-offs for resolving design conflicts. Since the methodology itself is not restricted to two-dimensional flow, as a result of the investigation it is recommended to extend the methodology to three-dimensional flow.

Samenvatting in het Nederlands

Dit proefschrift beschrijft de ontwikkeling van een methode voor de bepaling van de optimale oplossing van meer-punts aerodynamische ontwerpproblemen voor een enkelvoudige profiel in twee-dimensionale stroming. Met de term meer-punts wordt bedoeld dat de ontwerpdoelen voor meerdere stromingscondities tegelijkertijd in beschouwing genomen worden. Het ontwerpprobleem wordt geformuleerd als een optimalisatie-probleem, waarbij een aerodynamische objectfunctie geminimaliseerd moet worden, rekening houdend met aerodynamische en geometrische restricties.

De stromingsmodellen, die worden gebruikt als basis voor het ontwerp, zijn de onsamendrukbare potentiaalstroming waarvoor de Laplace vergelijking geldt, de niet-viskeuze samendrukbare stroming waarvoor de Euler vergelijkingen gelden en de viskeuze samendrukbare stroming die wordt beschreven met behulp van de Reynolds-gemiddelde Navier-Stokes (RANS) vergelijkingen.

Bij de ontwikkeling van de ontwerpmethode is gebruik gemaakt van bestaande computer-programma's gebaseerd op de bovengenoemde stromingsmodellen voor de berekening van stromingen om een enkelvoudig profiel. De ontwerpmethode maakt bovendien gebruik van een beschikbare subroutine gebaseerd op een optimalisatie-algorithme, dat uitgaat van het bekend zijn van de gradient van de te minimaliseren objectfunctie.

De gradient van een aerodynamische objectfunctie wordt op efficiënte wijze berekend met behulp van een methode waarin gebruik gemaakt wordt van variatierekening. Dit houdt in dat, voor iedere aerodynamische objectfunctie, een zogenaamd toegevoegd probleem moet worden geformuleerd. De oplossing van het toegevoegde probleem wordt gevormd door de vector van Lagrange multiplicatoren met behulp waarvan de gradient kan worden berekend.

De formulering van het toegevoegde probleem, de constructie van een oplossingsmethode voor het toegevoegde probleem ter bepaling van de Lagrange multiplicatoren en de gradient, alsmede de integratie van de oplossingsmethode voor het toegevoegde probleem met een bestaande stromingsberekeningsmethode en de bovengenoemde optimalisatiesubroutine, vormen het hoofdonderwerp van dit proefschrift.

Bij de ontwikkeling van de ontwerpmethode gebaseerd op de Laplace vergelijking, is uitgegaan van het zogenaamde inverse ontwerpprobleem, waarbij de ontwerpdoelen geformuleerd worden in de vorm van voorgescreven snelheidsverdelingen. Om er zorg voor te dragen dat op deze wijze een goed gesteld probleem wordt verkregen, is een methode gevolgd waarbij door middel van het

toevoegen van vrije variabelen in de definitie van de gewenste snelheidsverdeling zodanige vrijheidsgraden in het probleem zijn geïntroduceerd, dat aan de zogenaamde Lighthill-restricties kan worden voldaan. Door de op deze wijze verkregen modificatie wordt de voorgeschreven snelheidsverdeling realiseerbaar gemaakt.

De haalbaarheid van de optimalisatie methodologie wordt onderzocht voor het aerodynamisch ontwerp gebaseerd op de Euler vergelijkingen. Teneinde de nauwkeurigheid van de berekende gradient te evalueren, wordt een invers ontwerp-probleem beschouwd, waarbij door middel van het voorschrijven van een bekende drukverdeling een gegeven profiel gereconstrueerd wordt. De toepasbaarheid van de ontwerp-methode voor het oplossen van praktische aerodynamisch problemen wordt gedemonstreerd door middel van testgevallen van één-punts (golf) weerstand reductie met aerodynamische en geometrische restricties. Deze restricties bestaan uit

- een voorgeschreven draagkrachtcoëfficiënt,
- een voorgeschreven ondergrens voor het aerodynamisch duikmoment, voor het ingesloten oppervlak, voor de staartheek, en voor de neusstraal van het profiel.

Dezelfde testgevallen worden beschouwd voor de ontwerp-methode gebaseerd op de RANS vergelijkingen. De resultaten, die door middel van beide methoden zijn verkregen, worden met elkaar vergeleken.

Op basis van de kennis opgebouwd in het onderzoek met de Euler en de RANS vergelijkingen, is een meer-punts ontwerp-methode ontwikkeld. Hierbij wordt gebruik gemaakt van twee verschillende strategieën voor optimalisatieproblemen met meerdere objectfuncties, namelijk: (i) de methode van de som van gewogen objectfuncties, en (ii) de methode van "fuzzy" optimalisatie. De testgevallen worden gedefinieerd als een combinatie van één-punts gevallen, met hetzelfde type aerodynamische en geometrische restricties.

Op grond van de resultaten van het onderzoek wordt de conclusie getrokken dat de beschreven optimalisatie methodologie geschikt is voor het oplossen van praktische één-punts en meer-punts aerodynamische ontwerp-problemen voor enkelvoudige profielen in tweedimensionale stromingen. Het belang van de methodologie voor praktische doeleinden ligt vooral in de mogelijkheid om zowel aerodynamische als geometrische restricties aan te brengen én in de mogelijkheid te zoeken naar compromissen en uitwisselingsmogelijkheden bij conflictuerende ontwerp-eisen. Daar de methodologie zelf niet beperkt is tot tweedimensionale stromingen, wordt op grond van de bereikte resultaten aanbevolen de methodologie uit te breiden naar driedimensionale stromingen.

Acknowledgments

The research presented in this thesis has been made possible by the generous support from the Dutch National Aerospace Laboratory (NLR). I am grateful to all the people at the Theoretical Aerodynamics Department for the support with helpful discussions, and also for the friendliness. These gave me a really pleasant and challenging working atmosphere.

I would like to express my gratitude to Professor J.W. Slooff, as the promotor, for giving me the freedom to study and to probe in diverse directions during the research, while at the same time I have enjoyed his kind support in the form of many discussions, critical comments, encouragement, and advice. To Professor J.L. van Ingen as the co-promotor, I would like to thank for the attention and encouragement in pursuing the research.

Thanks are due to Ir. Th. Labrujère for many helpful discussions and advice, in particular for letting me share his knowledge in the field of aerodynamic inverse problems, and also for the kind help in the Dutch translation of the thesis summary.

To Ir. R. Hagmeijer and Dr.Ir. F.J. Brandsma, I would like to give thanks for the kind help in the use and application of the Euler/RANS flow solver.

In the aspects of design with the Euler/RANS equations, thanks are due to Professor J.W. Boerstael for his critical comments and helpful discussions.

I am grateful to Dr. B. Oskam for his kind support and attention in almost all aspects of the research and also, in particular, for the strong encouragement for pursuing the research in the methodology based on the Navier-Stokes equations.

I would like to thank Ir. K.M.J. de Cock for many helpful discussions and his willingness to be a "vraagbaak", particularly in the subject of CFD and generally in other important aspects.

I would also like to thank Ir. Bima Prananta for the pleasant and helpful phone/e-mail discussions, in particular, on the subjects of CFD and \LaTeX .

To Ir. J.W. van der Burg, I am grateful for helpful discussions in the early stage of the thesis writing.

I am also grateful to Prof. A. Tits of the University of Maryland who has kindly provided the optimization routine FSQP.

I would also like to thank the people at the computing center of the NLR for the kind support for keeping the input/output process going smoothly. To H.L. Por, I would like to give thanks for letting me share the rush hours together in his car between Almere and Amsterdam.

Finally, I would like to thank my wife for her constant support and encourage-

ment in all these years. To my daughter, I greatly appreciated her understanding in "Papa moet straks weer achter de computer zitten" in many occasions. Special thanks are due to my parents for the critical comments on the proposition (belonging to this thesis) related to ecology.

Nomenclature

C_1, C_2, C_3	Lighthill constraints
C_d	drag coefficient
C_l	lift coefficient
C_m	pitching moment coefficient
C_p	pressure coefficient
C_f	skin-friction coefficient
E	total energy per unit mass
M	Mach number
Pr	Prandtl number
Re	Reynolds number
S_a	airfoil surface
S_∞	Far-field boundaries
T	absolute temperature
V	magnitude of the velocity vector \vec{V}
V_n	normal velocity component
V_s	tangential velocity component
a	speed of sound
c_p	specific heat at constant pressure
c_v	specific heat at constant volume
e	internal energy per unit mass
p	pressure
s	surface arc length
t	time variable
u	x -component of the velocity vector \vec{V}
v	y -component of the velocity vector \vec{V}
v_n	normal velocity component
w	weight factor
\mathbf{A}	vector of aerodynamic constraint functionals
\mathbf{G}	vector of geometric constraint functions
\mathbf{Q}	vector of flow variables

R	residuals of the flow equations
S	vector of the search direction
f	generalized flow equations
p	vector of free parameters of the target velocity
q	vector of heat flux
x	airfoil coordinates
z	vector of the design point parameters
\mathcal{F}	general objective functional
\mathcal{L}	Lagrangian
\mathcal{P}	aerodynamic objective functional
\vec{V}	velocity vector
\vec{n}	unit normal vector
\vec{s}	unit tangential vector
$\vec{\omega}$	deformation velocity vector
α	angle of attack
δ	deviation, variation
κ	thermal conductivity coefficient
ν	kinematic viscosity
ρ	density
$\tau_{xx}, \tau_{xy}, \tau_{yy}$	elements of the viscous stress tensor
θ	vector of geometric parameters
λ	vector of the Lagrange multipliers for the flow equation
Υ	vector of the Lagrange multipliers for the flow boundary condition
ξ, η	control curve coordinates (in Chapter 3), coordinates of the computational domain (in Chapters 4 and 5)
ϕ	total velocity potential
φ	perturbation velocity potential
σ, σ	(vector of) source strength
μ, μ	(vector of) doublet strength (in Chapter 3), (vector of) membership function (in Chapter 6)
μ	viscosity (in Chapter 5)
ϱ	perturbation velocity potential due to a source of unit strength
Ω	flow domain

Contents

Summary	iii
Samenvatting in het Nederlands	v
Acknowledgments	vii
Nomenclature	ix
1 Introduction	1
2 Computational Aerodynamic Design Methods: characteristics, possibilities and limitations	5
2.1 Introduction	5
2.2 Direct Optimization Methods	6
2.3 Inverse methods	16
2.4 Flow Optimization Methods	26
2.5 Requirement Specification for a Multi-Point Aerodynamic Design Methodology	27
3 Aerodynamic Optimization using the Laplace Equation	31
3.1 Introduction	31
3.2 Statement of the Inverse Problem	33
3.3 Parametric Representation of the Airfoil Shape	34
3.4 Parametric Representation of the Target Velocity Distribution . .	36
3.5 Flow Equation	37
3.6 Discretization	39
3.7 Formulation of the Adjoint and Gradient Equations	41
3.8 Optimization Routine	42
3.9 Results and Discussion	42
3.10 Conclusion	46
4 Aerodynamic Optimization using the Euler Equations	57
4.1 Introduction	57
4.2 Statement of the Design Problem	58
4.3 The Euler Equations	58
4.4 Formulation of The Adjoint and Gradient Equations	60
4.4.1 The Adjoint Equation	62

4.4.2	The Gradient Equation	64
4.4.3	Remarks related to Jameson's approach for formulating the adjoint equations and boundary conditions	69
4.5	Shape Parameterization and Deformation Velocity	70
4.6	Aerodynamic Scaling of the Design Variables	74
4.7	Numerical Implementation	76
4.7.1	Discretization of the Adjoint Equations	76
4.7.2	Implementation of the Adjoint Boundary Conditions	78
4.7.3	Treatment (of the Correction) of Flow Far-Field Boundary Conditions	80
4.7.4	Verification of the Gradient Equations	81
4.8	Optimization Routine	82
4.9	Test Cases for Single-Point Design	83
4.9.1	Description and Results for Case E-1	84
4.9.2	Description and Results for Case E-2	85
4.9.3	Description and Results for Case E-3	87
4.9.4	Description and Results for Case E-4	88
4.10	Concluding Remarks	88
5	Aerodynamic Optimization using the Reynolds-Averaged Navier- Stokes Equations	113
5.1	Introduction	113
5.2	Statement of the Design Problem	114
5.3	The Reynolds-Averaged Navier-Stokes Equations	114
5.4	Formulation of the Adjoint and Gradient Equations	117
5.4.1	The Adjoint Equation	118
5.4.2	The Gradient Equation	122
5.5	Numerical Implementation	124
5.5.1	Discretization of the Adjoint Equations	124
5.5.2	Implementation of the Adjoint Boundary Conditions	124
5.5.3	Discretization of the Gradient Equations	126
5.6	Test Cases for Single-Point Design	126
5.6.1	Description and Results of Case N-1	126
5.6.2	Description and Results of Case N-2	127
5.6.3	Description and Results of Case N-3	129
5.6.4	Description and Results of Case N-4	130
5.7	Concluding Remarks	130
6	Multi-Point Aerodynamic Design	159
6.1	Introduction	159
6.2	General Statement of the Multi-objective Optimization Problem	159
6.3	Scalarization by the Sum of Weighted Objectives	160
6.4	Scalarization by means of Fuzzy Set Theory	160
6.5	Test Cases for Multi-Point Design	164
6.5.1	Results of Case ME-1 (Sum of Equally Weighted Objectives)	167
6.5.2	Results of Case ME-2 (Fuzzy Optimization)	167

6.5.3	Results of Case MN-1 (Sum of Equally Weighted Objectives)	168
6.5.4	Results of Case MN-2 (Fuzzy Optimization)	168
6.6	Concluding Remarks	169
7	Conclusions and Recommendations	187
7.1	Direct Aerodynamic Optimization Methodology	187
7.2	Optimization with the Laplace Equation	188
7.3	Optimization with the Euler Equations	188
7.4	Optimization with the Reynolds-Averaged Navier-Stokes Equations	188
7.5	Multi-point Aerodynamic Design	189
7.6	Recommendations for Future Research	189
BIBLIOGRAPHY		195
A	Matrices of the Right and Left Eigenvectors of the Normal Flux Jacobian	197
B	Far-Field Boundary Conditions of the Adjoint Equations	199
C	Formulas Related to the Correction of the Far-Field Boundary Condition	201
D	Variation of the Viscous Term	205
E	Adjoint Convective Fluxes	213
F	Adjoint Viscous Fluxes	215
G	Variations in Distributed Parameter Systems	217
	Curriculum Vitae	221

Chapter 1

Introduction

The process of aircraft design can be divided into three phases. The first phase is that of conceptual design. This phase ends with the decision on the aircraft configuration, e.g. narrow- or wide-body, jet or propeller propulsion, etc. The second phase is that of preliminary design. This results in the determination of the aircraft global parameters, e.g. wing span and area, engine size, etc. The third phase, which is commonly known as the detailed design phase, is concerned with the detailed development of aircraft components where each component must be considered as an integrated part of the aircraft. During this phase, the task of an aerodynamic designer is to define the detailed aerodynamic shape of aircraft components conforming to the design criteria and constraints specified in the previous design phases. To fulfill the task, three approaches can be distinguished: (i) experimental, (ii) computational analysis, and (iii) computational design.

The experimental approach involves measurement of a model of an aircraft (or an aircraft component) in a wind tunnel. The model is configured so as to get close similarities between the flow around the model and the flow around the full-scale aircraft in real flight. If the wind tunnel test data are considered unsatisfactory with regard to the design criteria, one has to modify the model shape and a new wind tunnel entry is necessary. The process has the form of a design-loop. The number of design cycles to be performed for meeting the design requirements depends to a large extent on the knowledge and skill of the designer. The number of design cycles that can be performed within a given time or within a given budget is determined primarily by the time and cost of model design and manufacturing.

The computational analysis approach aims at a reduction of the time and the cost of the design cycle by replacing the wind tunnel experiment by a numerical flow simulation. The rapid developments in mathematical flow modeling and numerical algorithms, enabled by the rapid advances in computer technology, have led to a wide variety of flow analysis computer codes. Such codes provide the numerical simulation of the flow around an aircraft (or an aircraft component) with approximate discrete representation of mathematical models of the physics involved. The designer evaluates the data that, if unsatisfactory, may necessitate a shape modification which in turn would entail another computation for the new shape. Because of the limited validity of the mathematical models, experimental

verification, in the wind tunnel, is, at some stage, inevitable.

In both the experimental and the computational analysis approach, the role of the designer, as man-in-the-loop, is a crucial one. One has to decide if and what shape modification is needed.

The computational design approach addresses the aerodynamic design problem more directly. A design code is characterized by the ability to assist the designer by creating a shape which produces flows that conform in some approximate sense to the design criteria. The aim is to reduce the man-in-the-loop effort in a typical design loop. This offers benefits like

- Taking the burden of repetitious shape modifications away from the designer. This means that the designer can avoid dealing with the non-linear relationship between the aerodynamic characteristics and the geometry of the shape. Therefore, more attention can be given to the interpretation and the formulation of the design criteria.
- Reducing personal and historical biases in a design environment that may hinder innovations.

The starting point of any detailed aerodynamic design of an aircraft is the specification of the design criteria in relation to the mission that the aircraft has to fulfill. The task of the designer is not only to meet the specification with a minimum or at least acceptable risk, but also to provide an adequate level of competitiveness in the quality of his design product relative to any potential rival in the aircraft industry.

The mission is usually specified as a profile along which the aircraft has to operate (Figure 1.1). The flight condition is changing continuously along the profile. Taking simultaneously the whole mission profile into consideration in the design process can be a formidable task. Therefore, it is common practice to divide the mission profile into segments which are decisive for the overall quality of the design. Then, for each individual segment a design-point and the corresponding design criteria are identified, and the associated flight condition is characterized by physical parameters, such as Reynolds and Mach numbers.

An aircraft has, in general, quite a complex geometry and it is practically impossible to handle this as a single entity in the detailed design phase. Thus, it is common practice to decompose an aircraft into several components (Figure 1.2). Each component is then designed separately subject to the design criteria and subject to constraints associated with design integration that have been formulated in the preliminary design phase.

The above simplification permits one to deal with what is called the *single-point* design problem of an aircraft component. However, when putting it all together at the end of the design process, one must cope with interference between components and conflicting *multi-point* design criteria. This may then lead to a laborious trial-and-error design process.

One way of dealing with the interference problem is to take more components into account simultaneously. Following this school of thought, a number of multi-component and three-dimensional computational design methods have emerged.

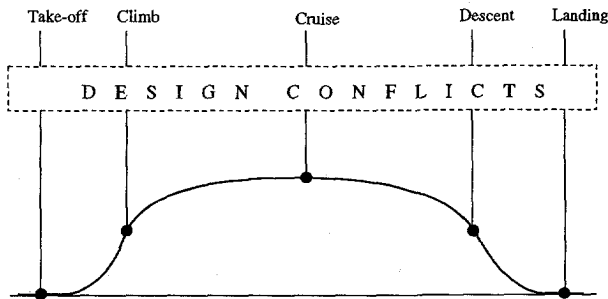


Figure 1.1: Typical mission profile and design points

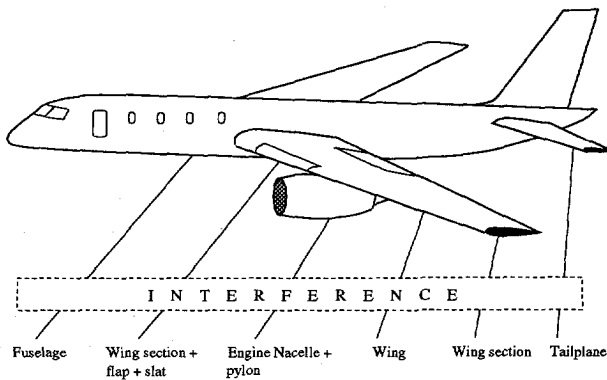


Figure 1.2: Break-down of aircraft components

By contrast, there seems to have been relatively little attention on how to cope with the conflicting multi-point design criteria. It is probably fair to state that multi-point design is a less-developed discipline in computational aerodynamics. Because of its potential for improving the efficiency of the design process and the quality of the design product, developments towards the establishment of multi-point aerodynamic design procedures, in the author's opinion, are desirable, if not essential.

The present thesis addresses multi-point aerodynamic design problems of a single airfoil in two-dimensional flow and proposes a computational design methodology for such problems. The development of the methodology is directed towards effectively resolving the multi-point design problem in an efficient manner. It is realized that accomplishment of this task requires an efficient and robust computational analysis method. However, bearing in mind that the rate of advancement in computational aerodynamic analysis methods is high, the dependency on the existing analysis methods should be kept low. This consideration is taken into account in the present development. It is also realized that, in a real design environment, an aerodynamic designer must inevitably interact with designers from other disciplines. The present development anticipates such

interaction by taking geometric constraints into consideration.

This thesis is organized as follows. Chapter 2 provides an overview of the current status of computational aerodynamic design. The chapter groups existing methods into different classes. The characteristics, possibilities, and limitations of each class of methods are assessed. The outcome of the assessment leads to general guidelines which direct the investigations described in the subsequent chapters in order to develop a multi-point aerodynamic design methodology.

Chapter 3 investigates the viability of utilizing the variational method in computing the gradient for aerodynamic optimization in inviscid incompressible flow. The flow model to be considered is the Laplace equation which is formulated as a boundary integral equation. The addressed optimization problem is the inverse problem of airfoils. This concerns the construction of an airfoil shape for a prescribed target surface velocity distribution. The technique for obtaining the well-posedness of the inverse problem is explained in terms of the Lighthill constraints.

Chapter 4 describes a general formulation of the optimization scheme for airfoils in inviscid compressible flow, governed by the Euler equations, utilizing the variational method for computing the gradient. This chapter addresses the (constrained) minimization of aerodynamic functionals taking such forms as surface integrations of the pressure and geometrical variables. The applicability for solving (reconstruction) inverse problems and a geometrically constrained drag reduction problem is demonstrated.

Chapter 5 presents an investigation of the feasibility of applying the variational method for (constrained) aerodynamic optimization using the Reynolds-averaged Navier Stokes equations. This chapter addresses the same design cases as in Chapter 4.

Chapter 6 describes the formulation of a multi-objective optimization scheme for solving multi-point aerodynamic design problems. It is assumed that an aerodynamic objective corresponds to a particular design point such that in multiple design points one has to deal with multiple objectives. The investigation concentrates on methods for incorporating compromises which may have to be imposed by the designer. To this end, two methods, the method based on the fuzzy set theory and the method of summing the objectives with weighting factors, are studied for their practicability. The techniques developed in Chapters 4 and 5 are adopted as elements of the scheme.

Finally, conclusions are drawn and presented in Chapter 7.

Chapter 2

Computational Aerodynamic Design Methods: characteristics, possibilities and limitations

2.1 Introduction

This chapter presents an assessment of currently existing aerodynamic design methods. The methods are grouped into a number of classes. Detailed descriptions of the methods will not be given, as these can be found in the literature. Also, the methods to be discussed are limited to those relevant for the present investigation. The list of cited references is not meant to be exhaustive, but is considered as representative for the methods discussed. References [51], [14], [38] should be consulted for a more complete picture of the current status of computational aerodynamic design methodology.

The assessment is meant to identify the strengths and the weaknesses of each class of method. This should then provide a basis for the conceptual requirements of a new design methodology capable of solving multi-point design problems. For the present assessment, the following classification has been chosen

- (a) *Direct optimization methods.* Methods of this category attempt to directly achieve the design criteria in terms of the global aerodynamic parameters, e.g. C_L , C_D , etc. The designer is assumed to be able to formulate the design problem as an optimization problem. The design problem becomes a minimization (or a maximization) problem of an objective functional subject to a number of design constraints. Aerodynamic constraints, e.g. C_M is not to exceed a certain value, or geometric constraints, e.g. the airfoil thickness should be greater than a specified value, can be incorporated directly in the formulation of the optimization problem.
- (b) *Inverse methods.* Methods of this type assist the designer by constructing an aerodynamic shape which generates a prescribed target velocity (or pressure) distribution on the surface of the shape. The designer is assumed to be able to prescribe the target velocity distribution in such a way

that it reflects required aerodynamic characteristics like lift, drag, pitching moment, and boundary-layer properties which determine the aerodynamic performance.

- (c) *Flow optimization methods.* In contrast with the previous methods where the aerodynamic shape is the design variable, flow optimization methods use flow characteristics (e.g. the velocity distribution on the surface) as the design variable(s), the optimal value of which is determined by means of an optimization algorithm. The objective functional to be minimized and the design constraints (of aerodynamic nature) are expressed in terms of aerodynamic performance quantities like C_L , C_D , etc. However, since the shape is the sought output of any aerodynamic design, methods in this category must be complemented with a shape-generating method.

The Assessment of characteristics, possibilities, and limitations of each class of methods must address the following aspects:

1. *Applicability of flow model.* This concerns the application range of the flow model(s) utilized in the design methodology. This is of importance, because the specified design points in a multi-point design problem may fall in a wide range of flow regimes which all should be adequately represented by the flow model.
2. *Completeness of problem representation.* This concerns the type of design objectives and constraints that can be incorporated in a design problem formulation. This aspect is relevant because in multi-point design different types of design objectives and constraints may be required for different design points.
3. *Suitability for multi-point design.* This aspect concerns the suitability of (extending) a design methodology for treating multi-point design problems.
4. *Computational cost.* This concerns the CPU time and the storage requirements of a design method when applied to a typical design problem.
5. *Code development effort.* This concerns the level of utilization of the available flow solvers. This aspect is addressed because higher utilization usually correlates with less effort (c.q. cost) for building a design code.
6. *Man-in-the-loop interaction.* This concerns the level of automation of the design process. That is, it has to be established where man-in-the-loop interaction is required and can be incorporated and where it can be and should be avoided.

2.2 Direct Optimization Methods

Aerodynamic design problems can be posed as an optimization problem, that is a minimization (or a maximization) problem for an (aerodynamic) objective

functional subject to aerodynamic and geometric constraints. The problem is solved by coupling an optimization algorithm with a flow solver with parameters describing the geometry of the shape as design variables. The work by Hicks et al. [24] may be regarded as pioneering this approach. Although this approach has a number of attractive features, especially with regard to the possibilities for implementing constraints, it had often been criticized for being too consumptive in terms of computational resources incurred by the computation of the objective gradients.

Another approach within the category of direct optimization methods is based on optimal control theory, which was investigated by Pironneau [46]. The potential of optimal control theory for treating practical aerodynamic design problems was recognized by Jameson [28], and has been demonstrated with computer results in subsequent works (Refs. [29], [47], [30], [32], [31]). Jameson's approach offers efficiency in the calculation of the objective gradients without sacrificing the attractive features of Hicks' approach.

The direct optimization problem is typically stated as follows,

$$\begin{aligned} &\text{Minimize} && \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}), \\ &\text{Subject to:} && \\ & && \mathbf{A}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}) \leq 0, \\ & && \mathbf{G}(\boldsymbol{\theta}) \leq 0, \end{aligned} \tag{2.1}$$

where \mathcal{P} is an aerodynamic objective functional, \mathbf{Q} is the vector of flow variables, $\boldsymbol{\theta}$ is the vector of design variables describing the aerodynamic shape, and \mathbf{z} is the vector of design point parameters (e.g., Mach number, Reynolds number, etc.). \mathbf{A} is a vector of aerodynamic constraint functionals, and \mathbf{G} is a vector of geometric inequality constraints. The flow variables \mathbf{Q} are implicitly dependent on $\boldsymbol{\theta}$ via a flow equation representing the flow model at the steady-state, the solution of which can be written as[†]

$$\mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}) = 0. \tag{2.2}$$

The term direct optimization refers to the aerodynamic objective and constraint functionals being expressed directly in terms of the global aerodynamic parameters: C_L , C_M , and C_D . These parameters are usually the (only) ones that appear in a design criteria specification. Geometric constraints are incorporated also in a direct manner into the problem formulation. Therefore, direct optimization methods offer attractive features with regard to the *completeness of problem representation*. However, in these methods, there is no direct control over the surface flow. Yet, the velocity distribution and the associated boundary-layer properties determine the values of the global aerodynamic parameters to a very large extent. Hence, it would seem desirable to exercise some control over the surface flow simultaneously with finding the optimal solution. A possible way of doing this would be to quantify a desirable pressure distribution and boundary-layer properties and to incorporate these into the optimization problem statement. This, however, is outside the context of the direct optimization methodology.

[†]The boundary conditions are supposed to be included in the notation of \mathbf{f} .

In direct optimization methods, there is virtually no restriction regarding the *applicability of flow models*. In principle, any flow analysis code can be coupled with an optimization routine in order to obtain a code with a design capability. Indeed, the simplest implementation of the methodology would be to consider both the flow analysis code and the optimization routine as a black box.

Optimization routines based on various optimization algorithms (e.g., Ref. [63]) are available on a commercial as well as public domain basis. In general, they can be classified into:

- *Zeroth-order methods* (e.g., Powell method). These methods require only the function values (of objective and constraints).
- *First-order methods* (e.g., Steepest Descents method). These methods require the function values and gradient with respect to the design variables.
- *Second-order methods* (e.g., Newton method). These methods require the function values, gradients, and Hessian matrices, assuming the existence of the second derivatives with respect to the design variables.
- *Quasi second-order methods* (e.g., Davidson-Fletcher-Powell method). These methods require the function values and gradients. In the course of optimization, the Hessian matrices are approximated sequentially based on the available information on the gradients. These methods also assume the existence of the second derivatives.

Zeroth-order methods are easy to implement, but convergence to the optimal solution is usually rather slow implying many function evaluations. Provided the gradient can be calculated, first-order methods offer significant improvement in convergence to the optimal solution. Convergence can be improved further by using a second-order method (if the Hessian matrix does exist). However, the computation of the Hessian matrix can be a formidable task. Nowadays, quasi second-order methods seem to be the most popular, because the approximation of the Hessian matrix does not incur high computational cost, while convergence is, in general, better than that of first-order methods. Accordingly, the following discussion assumes the use of a gradient based optimization method.

Algorithm I (Figure 2.1) outlines the design procedure of the direct optimization method. Gradients are required for determining the search direction vector \mathbf{S} in order to achieve the optimality condition \mathcal{C} . For example, for unconstrained problems, the steepest descents method defines $\mathbf{S} = -d\mathcal{P}/d\boldsymbol{\theta}$, whereas the optimality condition \mathcal{C} is expressed as $\|d\mathcal{P}/d\boldsymbol{\theta}\| = 0$. Thus, optimization algorithms require the gradient of \mathcal{P} , \mathbf{A} , and \mathbf{G} with respect to the design variables $\boldsymbol{\theta}$ at a current value of $\boldsymbol{\theta}$ (i.e., at $\boldsymbol{\theta} = \boldsymbol{\theta}^k$).

Computing the gradients of the aerodynamic objective and constraint functionals (\mathcal{P} and \mathbf{A}) is not at all trivial because of the implicit dependency of the flow variables \mathbf{Q} on the design variables $\boldsymbol{\theta}$ in the flow equation (2.2). Three methods for calculating the gradients can be distinguished. The description is given for the functional \mathcal{P} , but it applies also to the constraint functionals \mathbf{A} . For the

-
- (1) Let $k = 0$, and select initial design variables θ^k .
 - (2) Solve the flow equation $f(\mathbf{Q}^k, \theta^k, \mathbf{z}) = 0$ to obtain \mathbf{Q}^k .
 - (3) Evaluate the objective $\mathcal{P}(\mathbf{Q}^k, \theta^k, \mathbf{z})$, and the constraints $\mathbf{A}(\mathbf{Q}^k, \theta^k, \mathbf{z})$ and $\mathbf{G}(\theta^k)$.
 - (4) Evaluate an optimality condition \mathcal{C} . If it is satisfied then stop the procedure, otherwise proceed to the next step.
 - (5) Calculate a search direction \mathbf{S} along which the design variables can be updated:

$$\theta^{k+1} = \theta^k + \beta \mathbf{S},$$

where β is a scalar representing the optimal distance in the direction \mathbf{S} along which θ is moved in the design space.

- (6) Let $k = k + 1$, and return to step (2).
-

Figure 2.1: Algorithm I: Direct Optimization Method

sake of brevity the vector of design point parameters \mathbf{z} has been dropped[†]. Now, a gradient operator of \mathcal{P} with respect to the design variables θ is defined as

$$\frac{d\mathcal{P}}{d\theta} = \frac{\partial \mathbf{Q}}{\partial \theta} \frac{\partial \mathcal{P}}{\partial \mathbf{Q}} + \frac{\partial \mathcal{P}}{\partial \theta}, \quad (2.3)$$

where $\partial \mathbf{Q} / \partial \theta$ is the Jacobian matrix of \mathbf{Q} with respect to θ as implied by the flow equation (2.2). The three methods for computing the gradients are described as follows:

- (a) *Difference method.* Each component of the vector of design variables is perturbed in order to provide a perturbed value of the functional, and subsequently, the associated component of the gradient is calculated by a divided difference (Refs. [25], [34], [20]). The vector of design variables θ is

[†]The gradient with respect to θ is required for a fixed set of design point parameters \mathbf{z} .

perturbed in the following way:

$$\begin{pmatrix} \theta_1 \\ \vdots \\ \theta_{i-1} \\ \theta_i \\ \theta_{i+1} \\ \vdots \\ \theta_n \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \epsilon \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \tilde{\theta}_i,$$

then, the flow equation (2.2),

$$\mathbf{f}(\tilde{\mathbf{Q}}, \tilde{\boldsymbol{\theta}}_i) = 0,$$

is solved for the perturbed flow variables $\tilde{\mathbf{Q}}$. The perturbed functional value can be obtained as $\mathcal{P}(\tilde{\mathbf{Q}}, \tilde{\boldsymbol{\theta}}_i)$. Then, the i -th component of the gradient is determined as

$$\left(\frac{d\mathcal{P}}{d\theta_i} \right) = \frac{\mathcal{P}(\tilde{\mathbf{Q}}, \tilde{\boldsymbol{\theta}}_i) - \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta})}{\epsilon},$$

The most important disadvantage of this approach is that it requires at least $(n+1)$ flow analyses for obtaining the gradient vector of dimension n .

- (b) *Variational method.* In this method an analytical expression for the gradient is obtained by means of so-called Lagrange multipliers. The vector of Lagrange multipliers is the solution of a so-called adjoint problem, the nature of which is dictated by the functional and the flow equation (Refs. [28], [19], [50]). The method can be outlined as follows. A *Lagrangian* \mathcal{L} is defined as

$$\mathcal{L}(\mathbf{Q}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta}) + \boldsymbol{\lambda} \cdot \mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}), \quad (2.4)$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers. The dimension of $\boldsymbol{\lambda}$ equals the dimension of \mathbf{f} and, accordingly, the dimension of \mathbf{Q} . Indeed, $\boldsymbol{\lambda}$ is commonly referred to as the *adjoint* variables of the (discretized) *state* variables \mathbf{Q} and is defined in the same domain as \mathbf{Q} . The variation $\delta\mathcal{L}$ of \mathcal{L} which is due to the variations $\delta\boldsymbol{\theta}$, $\delta\mathbf{Q}$, and $\delta\boldsymbol{\lambda}$ can be written as

$$\delta\mathcal{L} = \left(\frac{\partial\mathcal{P}}{\partial\boldsymbol{\theta}} + \left(\frac{\partial\mathbf{f}}{\partial\boldsymbol{\theta}} \right)^T \boldsymbol{\lambda} \right) \cdot \delta\boldsymbol{\theta} + \left(\frac{\partial\mathcal{P}}{\partial\mathbf{Q}} + \left(\frac{\partial\mathbf{f}}{\partial\mathbf{Q}} \right)^T \boldsymbol{\lambda} \right) \cdot \delta\mathbf{Q} + \mathbf{f} \cdot \delta\boldsymbol{\lambda}. \quad (2.5)$$

In the formulations (2.4) and (2.5), \mathbf{Q} , $\boldsymbol{\theta}$, and $\boldsymbol{\lambda}$ are in principle independent variables.

From Equation (2.5) it follows that the optimality condition $\delta\mathcal{L} = 0$ is satisfied when

$$\begin{aligned} \frac{\partial\mathcal{P}}{\partial\boldsymbol{\theta}} + \left(\frac{\partial\mathbf{f}}{\partial\boldsymbol{\theta}} \right)^T \boldsymbol{\lambda} &= 0, \\ \frac{\partial\mathcal{P}}{\partial\mathbf{Q}} + \left(\frac{\partial\mathbf{f}}{\partial\mathbf{Q}} \right)^T \boldsymbol{\lambda} &= 0, \\ \mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}) &= 0. \end{aligned}$$

Two observations can now be made:

- (i) For a given θ , the contribution of $\delta\lambda$ to $\delta\mathcal{L}$ (i.e. the third term in the right hand side of Equation (2.5)) vanishes when the flow equation

$$\mathbf{f}(\mathbf{Q}, \theta) = 0, \quad (2.6)$$

is satisfied. This provides the values of \mathbf{Q} and, by the definition (2.4) of \mathcal{L} , implies that

$$\mathcal{L} = \mathcal{P}.$$

- (ii) As θ and \mathbf{Q} are available from (i), the contribution of $\delta\mathbf{Q}$ to $\delta\mathcal{L}$ (i.e. the second term in the the right hand side of Equation (2.5)) can be eliminated by solving the so-called adjoint equation:

$$\frac{\partial \mathcal{P}}{\partial \mathbf{Q}} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{Q}} \right)^T \lambda = 0. \quad (2.7)$$

This provides the values of λ and implies that

$$\delta\mathcal{L} = \delta\mathcal{P} = \left(\frac{\partial \mathcal{P}}{\partial \theta} + \left(\frac{\partial \mathbf{f}}{\partial \theta} \right)^T \lambda \right) \cdot \delta\theta.$$

This gives directly the gradient of \mathcal{P} with respect to θ :

$$\frac{d\mathcal{P}}{d\theta} = \lim_{\delta\theta \rightarrow 0} \frac{\delta\mathcal{L}}{\delta\theta} = \frac{\partial \mathcal{P}}{\partial \theta} + \left(\frac{\partial \mathbf{f}}{\partial \theta} \right)^T \lambda. \quad (2.8)$$

It should be noted that for unconstrained problems, the flow equation (2.6), the adjoint equation (2.7), and the condition that $\|d\mathcal{P}/d\theta\| = 0$ constitute the optimality condition for the Lagrangian \mathcal{L} and, equivalently, that for the functional \mathcal{P} .

Equation (2.4) is written in terms of generalized forms of the functional \mathcal{P} and flow equation (2.6), which applies to both continuous and discrete forms:

- In the continuous formulation, \mathcal{P} usually takes the form of an integral over (part of) the domain boundary,

$$\mathcal{P} = \int_S \psi(\mathbf{Q}, \theta) dS.$$

while the flow equation (2.6) is applied in each point $\mathbf{x} \in \Omega$, where Ω is the flow domain, while \mathbf{Q} and λ are distributed over the domain. This implies that in eq.(2.4) the dot product appears as an integral over the domain. The expression for \mathcal{L} then takes the form

$$\mathcal{L}(\mathbf{Q}, \theta, \lambda) = \int_S \psi(\mathbf{Q}, \theta) dS + \int_{\Omega} \lambda \cdot \mathbf{f}(\mathbf{Q}, \theta) d\Omega.$$

where

$$\left. \begin{aligned} \mathbf{Q} &= \mathbf{Q}(\mathbf{x}) \\ \lambda &= \lambda(\mathbf{x}) \end{aligned} \right\}, \quad \mathbf{x} \in \Omega.$$

- In a discrete formulation, the flow equation is defined in each point of a computational grid, and so are the vectors \mathbf{Q} and $\boldsymbol{\lambda}$. In a two-dimensional domain, \mathcal{L} would typically appear in the form

$$\mathcal{L}(\mathbf{Q}, \boldsymbol{\theta}, \boldsymbol{\lambda}) = \sum_i \psi(\mathbf{Q}_{i,0}, \boldsymbol{\theta}) + \sum_i \sum_j \lambda_{i,j} \cdot \mathbf{f}(\mathbf{Q}_{i,j}, \boldsymbol{\theta}).$$

The partial derivatives of the discrete quantities in the expressions for the adjoint equation (2.7) and gradient (2.8) are obtained by direct differentiation.

The adjoint equation (2.7) is linear in $\boldsymbol{\lambda}$ irrespective of whether the flow equation is linear or non-linear. The effort of solving the adjoint equation is generally comparable to that of solving the flow equation. The effort of evaluating the gradient (2.8) is generally comparable to the effort for the post-processing of a flow analysis. Thus, it can be expected that, in variational methods, the computational cost of obtaining the gradient is roughly equal to twice that of obtaining a single flow solution. It must also be noted that each functional (of the objective or constraints) is related to a distinct adjoint problem. Therefore, the number of (aerodynamic) functionals involved in the optimization problem is an important factor in determining the overall computational cost.

(c) *Sensitivity method.* The flow equation

$$\mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}) = 0$$

implies that

$$\frac{\partial \mathbf{f}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}} = 0.$$

This can be interpreted as that for any change in $\boldsymbol{\theta}$ there will be a corresponding change in \mathbf{Q} such that the combination $(\mathbf{Q}, \boldsymbol{\theta})$ always lies on the (hyper)plane defined by the flow equation $\mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}) = 0$. Letting

$$\begin{aligned} A &= \frac{\partial \mathbf{f}}{\partial \mathbf{Q}}, \\ B &= \frac{\partial \mathbf{Q}}{\partial \boldsymbol{\theta}}, \\ C &= \frac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}, \end{aligned}$$

and assuming that \mathbf{Q} and $\boldsymbol{\theta}$ have dimension m and n , respectively; the matrix A of dimension $(m \times m)$ is the Jacobian of the flow equation with respect to the flow variables; the matrix B of dimension $(m \times n)$ is the Jacobian of the flow variables with respect to the design variables; and C of dimension $(m \times n)$ is the Jacobian matrix of the flow equation with respect to the design variables. Rearranging the equation yields

$$A.B = -C.$$

Given current values of \mathbf{Q} and $\boldsymbol{\theta}$, which determine A and C , the above equation is solved for the elements of the unknown matrix B . Each column of B is obtained by solving a linear system of equations with the corresponding column of C as the right hand side. Thus, obtaining B requires the effort of solving a linear system of m equations with n different right-hand sides. The term "sensitivity" refers to the fact that the matrix B represents the *sensitivity* of the flow variables \mathbf{Q} to a small change of $\boldsymbol{\theta}$. As B is available, the gradient of \mathcal{P} with respect to $\boldsymbol{\theta}$ can be determined from equation (2.3). The partial derivatives appearing in the matrices A and C and in equation (2.3) are computed by means of direct analytical differentiation.

One should be aware that, in sensitivity methods, the number of design variables is an important factor in determining the computational cost implied by the computation of B . Compared with the variational method, for which the number of functionals is the important factor, the sensitivity method may be advantageous in terms of computational cost if the optimization problem involves many functionals.

In direct optimization methods the *code development effort* and the *computational cost* are strongly interrelated aspects. If the difference method is chosen for obtaining the gradient, the code development effort will be minimal at the expense of a high computational cost. On the other hand, both variational and sensitivity methods offer remarkably lower computational costs at the expense of a larger effort in developing the design code.

There are *variants of direct optimization methods* that do not strictly follow Algorithm I (Fig. 2.1), such as the "*simultaneous*" approaches described in Refs. [48], [58], and [26]. A distinct feature of these approaches is that the computation of the search direction vector \mathbf{S} (for updating the design variables $\boldsymbol{\theta}$) is computed based on the last available estimate of the flow variables. Thus, in the search for the optimal solution no "converged" flow solution is required, at any stage. At the end of the design procedure, the optimal solution for $\boldsymbol{\theta}$ is simultaneously obtained with the converged flow solution \mathbf{Q} , satisfying the flow equation $\mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}) = 0$. Such approaches offer a potential for a significant reduction in the computational cost, but seem to require a large code development effort because the flow solver and the optimizer have to be closely integrated in the design code.

An *all-at-once* approach, which may also be considered to be within the category of direct optimization methods, is proposed in Ref. [15]. The distinct feature of this approach is that the flow equation is directly incorporated in the optimization problem formulation as an equality constraint. Thus, the optimization problem (2.1) is reformulated as

$$\begin{aligned} &\text{Minimize} && \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}), \\ &\text{Subject to:} && \\ & && \mathbf{A}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}) \leq 0, \\ & && \mathbf{G}(\boldsymbol{\theta}) \leq 0, \\ & && \mathbf{f}(\mathbf{Q}, \boldsymbol{\theta}, \mathbf{z}) = 0. \end{aligned}$$

Here, the vector of design variables consist of both θ and Q , and the objective and constraints are treated as ordinary functions of θ and Q . This approach is also a simultaneous approach in the sense that the flow solution will not be satisfied until the optimal (θ, Q) is achieved at the end of the optimization procedure. Although this method seems to offer certain benefits in terms of computational cost, it is rather ambitious since the optimization algorithm has now not only the task of locating the optimal solution (in terms of the usual objective and constraints) but also of solving the flow equation. Hence, a very large-scale optimization problem is implied—after discretization, Q has the dimension proportional to the number of mesh points of the flow domain—, while optimization algorithms tend to perform worse when the number of design variables increases.

Adopting a gradient-based optimization algorithm requires the pre-assumption that objectives and constraints are smooth. In addition to that, it has the consequence that the global optimum (i.e. the "best" possible design solution) may not be found for a given initial estimate. This situation will not occur if an algorithm commonly referred to as a "global optimizer" is used. Such an algorithm, capable of locating the global optimum, does not assume knowledge of the topology of the design space, and therefore gradient information is not required. The algorithm only needs the function(al) values in the search of the global optimum. An example is a design method based on the *genetic algorithm* (e.g., Ref. [44]). This method simulates the process of natural selection based on the principle of survival of the fittest. In this terminology, an objective function represents the fitness level. An individual, representing an aerodynamic shape, is evaluated based on its level of fitness. The global optimum corresponds to the individual with the highest level of fitness.

The *suitability of direct optimization methods for multi-point design* can be described as follows. Multi-point design problems can be posed as a *multi-objective optimization problem*. Then, one has to deal with a vector of objective functionals:

$$\text{Minimize} \quad \begin{cases} P_1(Q, \theta, z_1) \\ \vdots \\ P_m(Q, \theta, z_m) \end{cases} \quad (2.9)$$

where m is the number of design points. P_k is associated with the objective on the k -th design point specified by the vector of design point parameters z_k . P_k , $k = 1, \dots, m$ constitute a vector of objectives. It should be noted that there is also a distinct vector of aerodynamic constraints A_k associated with the design points.

If the level of importance of one design point relative to others can be defined, then the vector of objectives can be *scalarized* by transforming the multi-objective optimization problem into the usual single-objective optimization problem:

$$\text{Minimize} \quad \tilde{P}(P_1, \dots, P_m), \quad (2.10)$$

where \tilde{P} is the scalar objective which incorporates all multi-point design criteria contained in P_k , $k = 1, \dots, m$. A typical method of scalarization is to form the

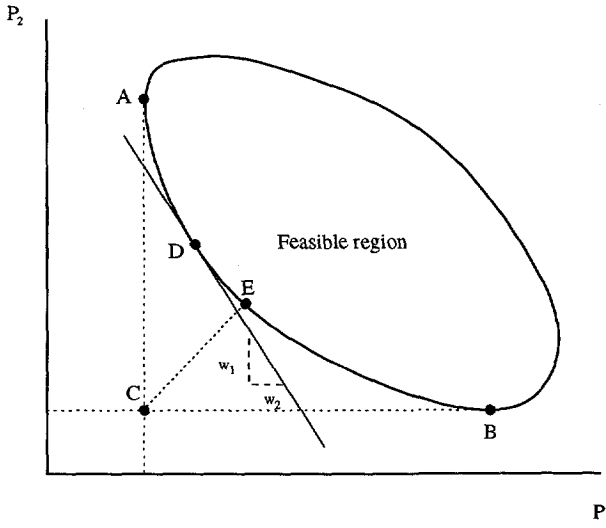


Figure 2.2: Pareto optimal solutions

sum of weighted objectives

$$\tilde{\mathcal{P}} = \sum_k w_k \mathcal{P}_k,$$

which has been applied in Ref. [59], [12], and [39]. This assumes that the weighting factors w_k are known *a priori*. Unfortunately, such an assumption may not always apply in practical situations. The nature of the multi-objective optimization problem is illustrated in Figure 2.2. Two objectives are assumed, namely \mathcal{P}_1 and \mathcal{P}_2 . The design space is drawn in the \mathcal{P}_1 – \mathcal{P}_2 plane. Point C represents an “ideal” solution. It is the point that corresponds to the minimum values of \mathcal{P}_1 and \mathcal{P}_2 if these were minimized individually. Point A represents the minimum of $\tilde{\mathcal{P}}$ if \mathcal{P}_2 is completely ignored ($w_2 = 0$). Similarly, Point B represents the minimum of $\tilde{\mathcal{P}}$ if \mathcal{P}_1 is ignored ($w_1 = 0$). Feasible optimal solutions, which are commonly known as the *Pareto optimal solutions* (Ref. [56]), lie on the segment between A and B of the boundary of the feasible domain. Point D represents one of the Pareto optimal solutions, which is located at the point where the line with the slope of $-w_1/w_2$ is tangent to the segment A – B . A “balanced” optimal solution is represented by E , which corresponds to the minimum distance between Point C and the segment A – B . This also illustrates that the balanced optimal solution, for which \mathcal{P}_1 and \mathcal{P}_2 are assumed to be “equally” compromised, does not necessarily correspond to the solution that results from specifying $w_1 = w_2$.

The multi-objective optimization approach seems to offer ample flexibility in solving multi-point design problems. It allows the designer to formulate compromises in a formal way. Indeed, compromise and trade-off are in the nature of multi-point design problems.

The discussion above could give the impression that direct optimization meth-

ods imply a high level of design automation with minor *man-in-the-loop interaction*. The interaction with the designer comes down to properly translating the design criteria into an optimization problem. Although this may seem simple in certain design cases, it is, in general, far from trivial. If it is not done properly, an ill-posed optimization problem can result. For example, the problem can be over-determined in the sense that there are too many constraints such that the feasible design space is null. On the other hand it can also be under-determined, implying multiple design solutions which might be of equal quality (in the sense of objectives and constraints) but are not (all) practically useful. Reformulation of the optimization problem is inevitable in order to obtain well-posedness. In the former situation, this can be done by relaxing the constraints, while in the latter situation, new constraints should be introduced.

The success of optimization depends also on whether the problem is well-scaled or not (Ref.[63]). This is determined by

- the nature of the optimization problem, and
- the way in which the aerodynamic shape is represented by the vector θ .

For example, the nature of the optimization problem might be such that a particular constraint "dominates" the objective which makes it difficult to locate the minimum. Also, unfortunate (relative) scaling of functional values and design variables may create deep and narrow "valleys" in the design space, in which case optimization algorithms would have to take many steps before arriving at the bottom of the valley.

In general, man-in-the-loop interaction must be anticipated in the form of reformulating the optimization problem, such that it is well-posed and well-scaled, and also in the form of feeding-in a number of different initial estimates so as to provide higher confidence that the best design solution is found.

From the preceding discussion it follows that a practical multi-point direct optimization methodology should contain the following features:

- (a) A multi-objective optimization scheme for solving multi-point design problems.
- (b) A scheme for relaxing the constraints in such a way that the adapted design problem will not differ too much from the one originally formulated by the designer.

2.3 Inverse methods

An aerodynamic design problem is often posed as an inverse problem. The term inverse refers to the construction of an aerodynamic shape generating a prescribed or "target" pressure (or velocity) distribution over the surface of the aerodynamic shape. The target pressure distribution implies lift, drag, pitching moment, and boundary-layer properties which determine the aerodynamic performance.

The development of inverse methods can be traced back to the works of Betz [5], Mangler [43], and Lighthill [41]. Utilizing conformal transformation principles, they showed that for a given condition of the onset flow, a target velocity distribution can be realized by an airfoil shape only when the velocity distribution satisfies three compatibility constraints. One constraint dictates compatibility between the target velocity and the onset flow. The other two constraints express compatibility with the closure condition of the airfoil contour. Although these constraints were derived for two-dimensional incompressible potential flows, it is generally accepted that for more general flow models such constraints exist as well. The notion of well-posedness of an inverse problem is usually connected to the fulfillment of the compatibility constraints. In spite of the fact that many existing inverse methods do not consider all the compatibility conditions, advancements in inverse methods have reached a level where problems in three-dimensional compressible flow are being addressed.

Prescribing a target pressure distribution over the surface of a body has the consequence that the surface boundary condition of the boundary value problem for solving the flow equation becomes non-linear. From the manner in which this non-linearity is treated, two approaches in solving the inverse problem can be distinguished:

- (a) *Indirect approach.* This refers to methods in which an iterative sequence of boundary-value problems with linearized boundary condition is solved in such a way that it converges to the solution(s) of the one with the non-linear boundary condition. Within this indirect approach, three classes of method can be distinguished:
 - (i) *Neumann-type methods* (Refs. [9], [53], [18]). These are frequently also referred to as the *residual-correction methods*. Imposing the *Neumann* boundary condition of zero normal velocity on the surface, the solution of the flow equation provides the surface pressure distribution. The deviation of the latter from the target pressure distribution is used as a driver for a shape modification. The shape modification algorithm ("design rule") is usually based on some relatively simple, linearized, relation between the shape modification and resulting change of the pressure distribution. It is essential that this linear approximate relation contains a representation of the main physical mechanisms of the flow. Algorithm II (Figure 2.3) illustrates the process.
 - (ii) *Dirichlet-type methods* (Refs. [60], [66], [23]). This type of methods is applicable for potential flow models. In these methods, the target pressure distribution (which can be translated into a tangential velocity distribution) is imposed as a *Dirichlet* boundary condition to the flow equation. The solution of the Dirichlet problem exhibits in general a finite non-zero normal velocity at the surface. Shape modifications are applied to correct this physically meaningless solution. The extent of the modification is determined by the magnitude of the resulting surface normal velocity. Algorithm III (Figure 2.4) illustrates

-
- (1) Let $k = 0$, and select an initial shape described by θ^k .
 - (2) Solve the flow equation $f(\mathbf{Q}^k, \theta^k, \mathbf{z}) = 0$ to obtain \mathbf{Q}^k .
 - (3) Calculate the deviation of the (actual) pressure distribution $p^k = p^k(\mathbf{Q}^k)$ from the target pressure distribution p_t :

$$\delta p = p^k - p_t.$$
 - (4) Evaluate a design rule \mathcal{G}_N to obtain the shape modification $\delta\theta$:

$$\delta\theta = \mathcal{G}_N(\delta p).$$
 - (5) For a small positive number ϵ , if $|\delta p| \leq \epsilon$ then stop the procedure, otherwise proceed to the next step.
 - (6) Update θ by

$$\theta^{k+1} = \theta^k + r\delta\theta.$$

where r is a relaxation factor.
 - (7) Let $k = k + 1$, and return to step (2).
-

Figure 2.3: Algorithm II: Neumann-type indirect inverse method

the process.

- (iii) *Inverse-by-optimization methods* (Refs. [36], [58]). Here, the inverse problem is posed as an optimization problem. The problem involves the search for an aerodynamic shape which gives a pressure distribution—the "actual" one that results from the flow equation—with a minimum deviation from a prescribed target pressure distribution. A common way of formulating this optimization problem takes the form

$$\text{Minimize } \mathcal{P} = \frac{1}{2} \int_S (p - p_t)^2 ds, \quad (2.11)$$

where \mathcal{P} is an aerodynamic functional that represents the deviation of the actual pressure distribution $p = p(\mathbf{Q})$ from the prescribed target pressure p_t ; S is the surface of the shape to be (re-)designed. The algorithm and techniques described in Section 2.2 can be used to solve this problem. Algorithm IV (Figure 2.5), which is derived from the more general Algorithm I, illustrates the process. It should be noted that, although it is not indicated in the above algorithm, geometric constraints can be incorporated easily in the optimization problem formulation in the same manner as described in the case of direct optimization methods (cf. Section 2.2).

-
- (1) Let $k = 0$, and select an initial shape described by θ^k .
 - (2) Obtain \mathbf{Q}^k from the flow equation $f(\mathbf{Q}^k, \theta^k, \mathbf{z}) = 0$ where the target pressure distribution (or its corresponding tangential velocity distribution) is imposed as a Dirichlet boundary condition, i.e. $p^k = p_t$ where $p^k = p^k(\mathbf{Q}^k)$.
 - (3) The resulting normal velocity distribution $v_n = v_n(\mathbf{Q}^k)$ is used to determine the shape modification following a design rule \mathcal{G}_D :

$$\delta\theta = \mathcal{G}_D(\delta v_n).$$
 - (5) For a small positive number ϵ , if $|v_n| \leq \epsilon$ stop the procedure, otherwise proceed to the next step.
 - (6) Update θ by

$$\theta^{k+1} = \theta^k + r\delta\theta.$$

where r is a relaxation factor.
 - (7) Let $k = k + 1$, and return to step (2).
-

Figure 2.4: Algorithm III: Dirichlet-type indirect inverse method

- (b) *Direct approach.* This refers to methods which directly address the boundary value problem with the non-linear boundary condition. This approach involves a non-linear system of equations that must be solved for a set of dependent variables consisting of both flow and geometric variables. Newton's method can be used for solving the non-linear system of equations, in which case Algorithm V (Figure 2.6) illustrates the process. Within the direct approach, two classes of method can be distinguished:
- (i) *Methods based on transformation of the flow equations* (Refs. [11], [13], [4]). In these methods, the flow equation is transformed into a special form based on the (initially unknown) streamline positions. Most methods within this category take advantage, in one way or another, of the fact that the velocity vector is tangent to the streamlines such that in the resulting system of non-linear equations the total number of dependent variables is reduced remarkably. The target pressure distribution is specified along an initial estimate of the streamline that corresponds to the surface to be (re-)designed. The solution of the transformed flow equation then provides both the flow variables and the actual position of all the streamlines, including the streamline that determines the aerodynamic shape.
 - (ii) *Least-squares-type methods* (Refs. [37], [7]). In these methods, the in-

-
- (1) Let $k = 0$, and select an initial set of design variables θ^k representing a geometrical shape.
 - (2) Solve the flow equation $f(\mathbf{Q}^k, \theta^k, \mathbf{z}) = 0$ to obtain \mathbf{Q}^k .
 - (3) For the resulting pressure $p = p(\mathbf{Q}^k)$ evaluate

$$\mathcal{P} = \frac{1}{2} \int_S (p - p_t)^2 dS$$

- (4) Evaluate an optimality condition \mathcal{C} representing the deviation $|p - p_t|$. If it is satisfied then stop the procedure, otherwise proceed to the next step.
- (5) Calculate a search direction \mathbf{S} by which the design variables can be updated:

$$\theta^{k+1} = \theta^k + \beta \mathbf{S},$$

where β is a scalar representing the optimal distance in which θ is moved in the design space in the direction \mathbf{S} .

- (6) Let $k = k + 1$, and return to step (2).
-

Figure 2.5: Algorithm IV: Inverse-by-optimization method

verse problem is formulated as a minimization problem of a *quadratic* functional representing the deviation of the actual pressure distribution from the target pressure distribution. Then, the stationary condition for this minimization problem is derived in a least-squares sense. This leads to a system of non-linear equations which is solved for both the flow and the geometric variables.

Regarding the *aspect of the applicability of the flow model*, the indirect approach, in particular the residual-correction method, seems to be the most flexible as it can make use of virtually all available types of flow solver. By contrast, the direct approach is somewhat restricted in this respect; not because of fundamental reasons, but rather due to practical limitations, as applications based on a complex flow model can easily result in a prohibitively large non-linear system of equations.

Considering the *completeness of problem representation aspect* it appears that inverse methods can be very effective as far as aerodynamics is concerned. Although there is no mechanism in the inverse procedure that directly controls all the aerodynamic properties, the possibility for the designer to prescribe a target pressure distribution is very valuable. The designer has control, though indirectly, over the global aerodynamic parameters of interest (C_L , etc.) and also over the

-
- (1) Let $k = 0$, and select an initial shape θ^k and an initial estimate of the flow variables Q^k .
 - (2) Calculate the residual of the system of non-linear equations:

$$R = R(\theta^k, Q^k(p_t), z).$$

This notation means that the pressure $p = p(Q^k)$, corresponding to a streamline described by θ , is set to the value of the target pressure distribution p_t .

- (3) For a small positive number ϵ , if $\|R\| \leq \epsilon$ stop the procedure, otherwise proceed to the next step.
- (4) For θ^k and Q^k calculate the Jacobians $\partial R / \partial \theta$ and $\partial R / \partial Q$, to obtain $\delta\theta$ and δQ :

$$\begin{pmatrix} \delta\theta \\ \delta Q \end{pmatrix} = - \begin{bmatrix} \partial R / \partial \theta & \partial R / \partial Q \end{bmatrix} R,$$

where $\partial R / \partial \theta$ and $\partial R / \partial Q$ are the Jacobians of the residual.

- (5) Obtain new estimates of θ and Q :

$$\begin{pmatrix} \theta^{k+1} \\ Q^{k+1} \end{pmatrix} = \begin{pmatrix} \theta^k + r\delta\theta \\ Q^k + r\delta Q \end{pmatrix}$$

where r is a relaxation factor.

- (5) Let $k = k + 1$, and return to step (2).
-

Figure 2.6: Algorithm V: Direct inverse method

surface flow which determines the boundary-layer properties. However, there is a fundamental difficulty in applying inverse methods, at least those which aim at exactly satisfying a specified target pressure distribution ("pure" inverse methods). This is related to the fact that there is little knowledge—with the exception of the compatibility constraints addressed by Lighthill [41] for 2-D airfoils—on what compatibility constraints there are in a more general 3-D case and on how to specify a target pressure distribution that conforms to such conditions. The situation becomes more complicated when practical geometric constraints, such as those related to structural requirements, must be taken into account also.

Indeed, it is very likely that the inverse problem (actually imposed by the designer) in general is ill-posed in the sense that the target pressure distribution is not completely realizable or the geometry is over-constrained. As a consequence,

convergence (in terms of vanishing $|\delta p|$, $|v_n|$, \mathcal{P} and $\|\mathbf{R}\|$ in Algorithms II-V) may not be realized. Although for certain ill-posed cases (pure) inverse codes may, at a certain stage, still give a useful output (in the sense that the actual pressure distribution "is close to" the target), one should anticipate that the ill-posedness can be so severe as to cause total failure of the inverse computation.

Three possibilities may be distinguished as a means to obtain well-posedness:

- *Relaxation of the geometric constraints.* By relaxing the constraints, it would be possible to find a geometry, which may possibly still be near the boundary of the region of feasible geometries (e.g., in a least-squares sense), and generates the target pressure distribution.
- *Relaxation or parameterization of the target pressure distribution.* By introducing free parameters into the prescribed target pressure distribution, the prescribed target pressure distribution can be adapted until a realizable one is found for which there exists a corresponding (constrained) geometry.
- *Relaxation of the design point parameters.* By adapting the design point parameters, it would be possible to fulfill the compatibility conditions such that a geometry can be found that generates the target pressure distribution. However, since the design point parameters are usually fixed as a result of the preliminary design phase, relaxing their values is not attractive from a practical point of view.

The strategy of relaxing both the target pressure distribution and the geometric constraints can be recognized in the method by Fray et al. (Refs. [16] and [17]). This method involves a weighted least-squares formulation which compromises between the geometric constraints with the aerodynamic requirements as implied by the target pressure distribution. In this method, the dependent variables are the strengths of the singularity distributions (sources, doublets, and vortices) of a linearized potential flow model which governs the shape modifications. It should be noted that this method does not guarantee the achievement of well-posedness in the strict sense of convergence with vanishing δp , etc. However, the least-squares formulation should be appreciated for its capability in avoiding undesirable geometries while maintaining a pressure distribution which is close to the "target"; thus preventing failure of the computation and producing useful results.

The methods described in Refs. [57], [11], and [66] use the strategy of adapting the target pressure distribution and/or the design point parameters. Obviously, it is desirable to keep the adaptation within a minimal extent. The method by Strand [57] for two-dimensional potential flow takes this into consideration by formulating a minimization problem for the deviation between a prescribed target pressure distribution (the original one) and the adapted target for which the Lighthill constraints are imposed explicitly. The adapted target is then fed into an inverse procedure for obtaining the shape. Such an approach can be applied only if closed-form compatibility conditions (in this case, the Lighthill constraints) are available. Unfortunately, as indicated earlier, knowledge on such conditions for more general flow situations is not available.

For practical use, it is desirable that possibilities for obtaining well-posedness are implemented in an automated scheme. The consequence of not doing so may be an intensive *man-in-the-loop interaction*, i.e. the designer must go through a, possibly, cumbersome process trying several possibilities in order to obtain a useful result (or even a result at all).

Regarding the *code development effort*, the residual-correction approach seems to require the least effort. That is, by simply attaching a (consistent) shape modification procedure to an existing flow analysis code, one can readily obtain an inverse code. The Dirichlet-type indirect approach needs more effort. This is associated with the necessary replacement of the Neumann boundary condition in an analysis code by the Dirichlet boundary condition. The implementation of an inverse method based on the direct approach is rather laborious. Although use of an existing flow analysis code might still be made, for example with regard to the discretization scheme, the construction of the system of non-linear equations usually demands considerable effort.

The trade-off between *code development effort* and *computational cost* observed when discussing direct optimization methods (Section 2.2) also applies to inverse methods. The direct approach, in which the flow variables and the shape may be considered to be obtained "simultaneously", generally require relatively modest CPU time compared to indirect approaches. In the direct approach, the overall computational cost for solving an inverse problem is in general of the same order of magnitude as that of a single flow analysis computation, but at the expense of a large code development effort. Regarding the storage requirements, the indirect approach seems to be the best choice. For the direct approach, the storage requirement can be prohibitively high if a large system non-linear of equations has to be dealt with.

With respect to the *suitability for multi-point design* two approaches are reported in the literature, namely:

- The surface segmentation approach.
- The inverse-by-optimization approach with weighted sum of objectives.

In the surface segmentation approach (Refs. [49], [36]), the body surface is divided into a number of segments. For each design point a specific body segment is designated on which a local target pressure distribution is imposed. This assumes the existence of a unique relation between the design point and the body segment. It also assumes that the designer is able to identify such a relation. Both assumptions are highly questionable, from a physical as well as a practical point of view. As a matter of fact, by strict multi-point terminology, the surface segmentation approach cannot be regarded as a multi-point design approach. The reason is that, given an inverse problem formulated in such manner, one can also arrive at the same solution by performing inverse calculations for each design point, where in each calculation the segment which is not of interest is kept fixed. This, as a whole, is identical with the process of solving a number of single-point inverse problems individually.

In the inverse-by-optimization approach (Ref. [39]), the multi-point inverse problem is formulated as a multi-objective optimization problem (cf. Section 2.2). For each design point an objective functional of the type (2.11) is defined. This results in a vector of objective functionals:

$$\text{Minimize} \quad \begin{cases} \mathcal{P}_1 = \frac{1}{2} \int_S (p_1 - p_{t_1})^2 dS \\ \vdots \\ \mathcal{P}_m = \frac{1}{2} \int_S (p_m - p_{t_m})^2 dS \end{cases}$$

where the subscript refers to the design point. The above multi-objective optimization problem is transformed into

$$\text{Minimize} \quad \sum_{k=1}^m w_k \mathcal{P}_k.$$

As p_{t_k} , for $k = 1 \dots m$, is the target pressure distribution that represents the aerodynamic requirement in design point k , the inverse-by-optimization approach is suitable for a multi-point design. From the computational results presented in Ref. [39] one can learn that even in the situation in which p_{t_k} fulfills the compatibility conditions in each individual design point k , the well-posedness of the whole multi-point inverse problem is not guaranteed.

The multi-point inverse problem is, in essence, a problem of constructing a single shape that generates, as closely as possible, two or more target pressure distributions, each of which is prescribed for a different design point. In order to understand the nature of the problem, a "one-dimensional" model equation is considered which is written as

$$f(p, x, z) = 0, \quad (2.12)$$

where f , p , x and z are assumed to represent the flow equation, the pressure, the shape, and the design point parameter, respectively. Although this may seem oversimplified, it facilitates understanding the concept. The contour $f = 0$ is illustrated in Figure 2.7 for different values of z . The shape is assumed to be constrained by

$$x_l \leq x \leq x_u.$$

The feasible region is indicated in the figure. In the analysis problem, a feasible x is given on a specified z . The corresponding p can then be obtained for the specific x and z .

In the single-point inverse problem, a target p is specified for a given z , and x is to be determined. If the inverse problem is to be well-posed then p should be specified such that the point (p, z) falls within the feasible region, in which case the shape x can be determined immediately. For an ill-posed problem where the specified point is located outside the feasible region, adapting the target pressure and/or relaxing the geometric constraint (allowing "leeway") is necessary, as described earlier. In addition to that, one can also adapt z such that the combination (p, x, z) falls within the (expanded) feasible region.

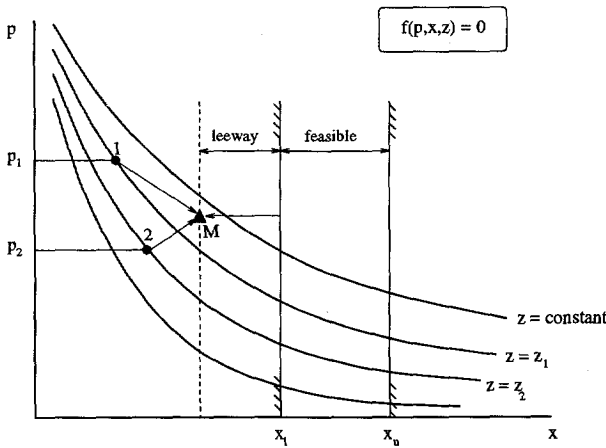


Figure 2.7: Domain of a model inverse problem

The multi-point inverse problem can be considered as a generalization of the single-point inverse problem in the sense that, instead of one pair of (p, z) , two or more pairs are specified, e.g. (p_1, z_1) and (p_2, z_2) , for which one value of x is to be determined. It is obvious that an exact solution exists only if (p_1, z_1) and (p_2, z_2) are located along a single line $x = \text{constant}$ in fig. 2.7. Since this is not the case in general, adaption of p and/or in z and/or leeways in the geometric constraints is inevitable. In Figure 2.7 point M illustrates a multi-point inverse solution that could be obtained in this manner, where the dotted line indicate the "leeway" that has been accepted.

Recalling the three possibilities for obtaining well-posedness of inverse problems, it is now clear that a practical multi-point inverse design methodology should contain the following features:

- (a) a scheme for relaxing to the geometric constraints.
- (b) a scheme for relaxing or adapting the prescribed target pressure distributions.
- (c) a scheme for relaxing or adapting the design point parameters.
- (d) a scheme for minimizing the extent of the adaptations/relaxations indicated in (a)–(c).

In essence, this means that the different requirements for specific design points are relaxed (in terms of the geometric constraints, the target velocities and the design point parameters) so as to become compatible with each other. It implies that some degrees of freedom have to be introduced into the problem with (d) serving to avoid multiple solutions due to the increased degrees of freedom.

2.4 Flow Optimization Methods

Flow optimization methods deal with the problem of determining flow characteristics that in some optimal sense satisfy specified design criteria. The optimization problem is usually posed as a minimization (or a maximization) problem of an aerodynamic objective functional subject to some aerodynamic constraints. The geometric shape is not involved as a variable in this problem. Instead, the objective and the constraint functionals are expressed solely in terms of the flow variables associated with some assumed geometric shape. Thus, in this approach, the design variables, of which the optimal value has to be determined, are parameters describing the flow variables. Usually, the flow variables of interest are those associated with the surface of the aerodynamic shape because of the direct relationship with forces, (pitching) moments, and the boundary-layer properties. A common choice for the design variables is a set of parameters describing the surface velocity (or pressure) distribution (Refs. [27], [40], [45], [61]).

Obviously, this methodology needs some form of surface flow or boundary-layer analysis, assuming that this is applicable (e.g. without excessive flow separation). The boundary-layer equations can be written in a symbolic form, such as

$$h(\mathbf{d}, \mathbf{b}, \mathbf{z}) = 0, \quad (2.13)$$

where \mathbf{d} and \mathbf{b} denote the vectors of the dependent and independent variables, respectively. The vector \mathbf{b} refers to a set of parameters describing the *driving* flow variables (i.e. the design variables), while \mathbf{d} refers to the *driven* flow variables. It is important to note that in attached flows, allowing direct boundary-layer analysis, the driving variable is the velocity distribution at the edge of the boundary-layer, while the driven variables are the boundary-layer parameters such as the skin friction, displacement thickness, etc. For separated flows, requiring inverse boundary-layer analysis (Ref. [64]), the opposite applies.

A typical way of stating the optimization problem is given below,

$$\text{Minimize} \quad \mathcal{P}(\mathbf{d}, \mathbf{b}, \mathbf{z}), \quad (2.14)$$

$$\text{Subject to:} \quad \mathbf{A}(\mathbf{d}, \mathbf{b}, \mathbf{z}) \leq 0 \quad (2.15)$$

The objective \mathcal{P} and the vector of constraints \mathbf{A} represent such global aerodynamic parameters as C_L , C_M , and C_D . It should be noted that the values of C_L , C_M , and C_D can only be approximate since at this stage the precise shape required for satisfying the aerodynamic objective is not involved or known. \mathcal{P} and \mathbf{A} can also represent aerodynamic requirements related to the boundary-layer properties such as the transition point location, velocity (pressure) gradient, and alike.

From the *completeness of problem representation* point of view, flow optimization methods have the distinct feature that they are capable of controlling the surface flow. However, the design problem is not fully addressed since the geometric shape is not a design variable.

Flow optimization methods require only modest *computational cost* in terms of CPU time, because a boundary-layer analysis can be performed rather quickly, at least for two-dimensional cases.

Since the aerodynamic shape is, eventually, the sought output of any aerodynamic design, flow optimization methods must be complemented with a shape-generating design method (e.g. an inverse method). At this point, *man-in-the-loop interaction* is expected to be necessary or desirable, because of all aspects discussed in the preceding section.

The algorithms and techniques (such as for the gradient calculation, multi-objective approach, etc.) described in Section 2.2 and the assessment related to these, are also relevant for the flow optimization methods. Thus, a multi-objective strategy can be adopted in order to deal with multi-point design problems, bearing in mind that in principle the compatibility of solutions in all design points must also be satisfied. The means described in Section 2.2 for coping with ill-posedness of the optimization problem related to the over-constrained or under-constrained situations can also be used here.

2.5 Requirement Specification for a Multi-Point Aerodynamic Design Methodology and Scope of the Thesis

The assessment presented in the previous sections shows that there is no single existing design methodology which is equally favorable in all aspects. When the development of a (new) multi-point aerodynamic design methodology is the objective, it is convenient to, first, specify the requirements, in terms of the aspects addressed in the preceding sections, that must be fulfilled by the methodology. These requirements can be divided into *primary* and *secondary* ones. Primary requirements are those considered as compulsory in the sense that not fulfilling them would imply that a useful design solution cannot be obtained. Secondary requirements are less strict in the sense that, even if these are not fulfilled, a design solution can still be obtained, but, depending on the design environment one is working in, some unfavorable consequences (e.g. high computational cost, etc.) may be implied. The primary requirements can be specified as follows:

1. With regard to the *applicability of flow model*, the design methodology should be based on flow models adequately covering the regimes of interest within which an aircraft must operate. This implies that, at least for transport aircraft, the flow models to be adopted should model compressibility effects and viscous phenomena related to the boundary-layer flow. This would mean that, as a minimum, a combination of an inviscid flow solver based on the potential flow model or the Euler equations and a compressible boundary-layer solver should suffice. However, it is obvious that use of a flow solver based on the (Reynolds-averaged) Navier-Stokes (RANS) equations to be preferred, in view of the wider class of flows modeled by RANS and the type of design points to be considered.

The discussion of the preceding sections indicates that direct optimization methods and "indirect"-type of inverse methods do not impose restrictions

regarding the flow models and are therefore to be preferred.

It is expected that one may wish to employ different flow solvers, depending on the nature of the design problem, the physics involved, and the design environment (e.g. computer resources). Therefore, investigations with different types of flow models are useful.

2. With regard to the *completeness of problem representation*, the design methodology should incorporate possibilities for dealing with design criteria comprising aerodynamic requirements (objectives and constraints) and geometric constraints. The geometric constraints are considered to be the result of interactions with other disciplines. The preceding discussions suggest that constraints are most easily incorporated within an optimization methodology. As discussed in Section 2.4 flow optimization methods do not address the complete design problem. Therefore, investigations in this thesis focus on direct optimization and inverse-by-optimization methods.
3. With regard to the *suitability for multi-point design*, the design methodology should provide a facility to search for compromises and perform trade-offs in a flexible way in order to resolve conflicts in the design criteria that may occur in a multi-point design problem. This implies that use of a multi-objective optimization algorithm should be explored. The incorporation of compromises and trade-offs should not be limited to a sum-of-weighted-objectives type of approach (cf. Section 2.2) because such an approach is capable of incorporating compromises in terms of objectives only and not, in principle, in terms of (acceptable) relaxation of constraints. Accordingly, this thesis also anticipates cases in which some of the constraints may have to be violated to a certain acceptable degree. For such cases, leeways can be introduced, but with the condition that they should be minimal in the optimal solution.

The secondary requirements are the followings:

1. With regard to the *computational cost*, it is desirable for practical reasons, that an optimal design solution, or at least a solution representing a significant improvement over an existing design, can be obtained in a quick turn-around time. In this respect, a gradient-based (quasi second-order) optimization algorithm seems to be most suitable and therefore is adopted. For computing the gradients, the difference method must be avoided. Anticipating a moderate number of (aerodynamic) functionals involved in the optimization problem, the variational method should be preferred over the sensitivity method. A large part of this thesis will therefore concern the formulation of adjoint equations and gradient formulae.
2. With regard to the *code development effort*, utilization of existing flow analysis codes and numerical subroutine libraries should be pursued. The design code should support easy maintenance and future extensions. This implies that a modular approach has to be employed in developing the code, with existing analysis codes and optimization routines treated as black boxes.

3. With regard to the *man-in-the-loop interaction*, the design code should, obviously, provide the designer with user-friendly man-machine interfaces such that necessary (or desirable) interactions can be performed efficiently. This implies, amongst others, that an optimal default status (or set of parameters), with the highest probability of giving useful solutions, has to be defined. At the same time, users should be allowed to alter the status or parameter set in order to explore other possible design solutions.

In summary, this thesis will concentrate on direct optimization and inverse-by-optimization methods, utilizing gradient based (quasi second-order) optimization algorithms with the gradients determined through the variational method. Methods of this kind will be investigated for incompressible potential flows (based on the Laplace equation), inviscid compressible flows (governed by the Euler equations) and viscous compressible flows (governed by the Reynolds-Averaged Navier-Stokes equations).

In view of time and effort limitations the investigations address airfoil design problems in two-dimensional steady flow only. However, the methodology chosen is expected to be applicable also in three-dimensional flow problems.

Chapter 3

Aerodynamic Optimization using the Laplace Equation

3.1 Introduction

The contents of this chapter is based on Ref. [55].

For the aerodynamic design of airfoils in incompressible potential flow, inverse methods seem to represent the only useful design problem formulation. Formulations in terms of the drag coefficient C_d are not possible due to the d'Alembert paradox[†].

In inverse design problems one has to deal with the construction of an airfoil geometry that produces a prescribed target velocity distribution along its arc length^{††}. The prescribed target velocity distribution usually satisfies criteria regarding the airfoil performance. Lift, drag, pitching moment, and terms associated with boundary-layer properties (e.g. laminar flow, transition location) that determine the aerodynamic performance are reflected within the velocity distribution.

The design process usually begins with a designer defining a target velocity distribution for a specified design point which serves as input to an inverse design code. In doing so, he may be assisted by a routine specially developed for such task (see for example van den Dam [61]). The inverse design code has the role of searching for the corresponding airfoil geometry. The effectiveness of many inverse codes depends on whether or not the target falls within (or close to) a class of realizable velocity distributions. When it is far from being realizable, some codes totally fail, while others may still yield a geometry albeit with a velocity distribution which differs appreciably from the target. Normally at this point, the designer must redefine the target velocity distribution, sometimes in a rather speculative manner, and feed it back into the code. Such procedure may have to be repeated several times, depending on the expertise of the designer. This seems

[†] Airfoils do not exhibit a drag force in two-dimensional incompressible potential flow.

^{††} Without loss of generality, the inverse problems in this chapter are assumed to be stated in terms of velocities instead of pressure, as the two quantities are related by the Bernoulli equation $p + \frac{1}{2}\rho V^2 = \text{constant}$.

to be a reason why inverse codes are mainly used for redesign purposes. Although this has proved to be very useful in practice, ignoring the starting-from-scratch approach may inhibit innovations.

The development of inverse methods can be traced back to the works by Betz [5], Mangler [43], and Lighthill [41]. Based on conformal transformation principles, they have shown that, for a given free stream condition, realizable target velocity distributions must satisfy three compatibility constraints. One constraint dictates the compatibility between the target velocity and the free stream. The other two express the condition that the airfoil contour must be closed. Nowadays, there are a considerable number of inverse methods that have been developed from many different approaches. A survey by Slooff [51] and Labrujère & Slooff [38] can be consulted for details.

One way of taking the constraints into account is to introduce free parameters in the velocity distribution. These parameters have the role of modifying a prescribed target velocity distribution in an appropriate way such that it becomes well-posed in terms of the constraints. This is the approach taken by Volpe & Melnik [66], and later also by Drela [11]. Of particular significance is the numerical observation performed by Volpe [65]. Volpe demonstrated that the parameters connected with the location of the forward stagnation point and the one with the pressure loading along the airfoil surface can effectively influence the constraints. In this chapter, investigations are carried out in which a similar approach is adopted, but it differs from Volpe's because an optimization technique is employed for driving the velocity parameters in order to obtain well-posedness in terms of the compatibility constraints. The significance of the velocity parameters will be investigated for cases in which the (initially prescribed) target velocity distribution is far from being realizable.

The inverse problem discussed hereafter is posed as an optimization problem where an objective functional is minimized. The objective functional expresses the deviation between the target velocity and the actual velocity distribution. An optimization algorithm provides a sequence of intermediate airfoil shapes with (actual) velocity distributions that reduce the objective functional until the deviation vanishes. This is what is referred to as the inverse-by-optimization method in Chapter 2.

In the current approach the airfoil shape is represented by a number of geometric parameters which are treated as the design variables. In order to assure well-posedness in terms of the compatibility constraints, a set of target velocity parameters is introduced, allowing modification of the target velocity distribution. The parameters are defined such that they can effectively alter the state of the constraints, and are treated as additional design variables. The optimization procedure updates the geometric and velocity parameters simultaneously while minimizing the objective functional. At the optimal solution, the obtained airfoil should produce a velocity distribution that matches the parameterized target velocity distribution, with a minimal deviation from the original target distribution.

The optimization algorithm used in this investigation requires information on the gradient of the objective functional with respect to the design variables. The objective functional is expressed in terms of the velocity distribution that in turn

is implicitly dependent on the geometric parameters through the governing flow equation. In order to obtain the gradient, the implicit dependency is treated by means of the variational method. The present study considers an incompressible potential flow. A boundary-integral equation expresses the flow equation. In order to satisfy the external Neumann boundary condition, which defines the zero normal component of the velocity, an internal Dirichlet boundary condition is imposed. A low-order panel method is employed to solve the discretized integral equation.

This chapter is organized in the following order. Firstly, the inverse problem is formulated. Secondly, the formulation of the adjoint and gradient equations are described. Finally, results of computations are presented and discussed, and conclusions are drawn.

3.2 Statement of the Inverse Problem

The inverse problem being addressed is that of the construction of the airfoil shape, defined by the coordinates $\mathbf{x}^*(s)$, which produces a prescribed target velocity distribution $V_o(s)$ along its arc length s . The arc length is to be measured from the lower trailing edge clockwise up to the upper trailing edge. It is normalized such that $s \in [0, 1]$. The sign convention for the velocity is such that it is negative on the lower surface and positive on the upper surface. Lower and upper refers to the arc sector before and after the stagnation point, respectively. The contour of the airfoil is required to be closed. The airfoil is immersed in an incompressible steady potential flow where the onset flow speed has been normalized such that $V_\infty = 1$.

The inverse problem is subject to the equation that governs the incompressible potential flow with a unit onset flow speed, expressed here, in a generalized form as[†]

$$f(\mathbf{x}^*(s), V_o(s)) = 0. \quad (3.1)$$

Because the airfoil contour is required to be closed and the onset flow is of unit speed, $V_\infty = 1$, a realizable target velocity distribution must satisfy the three compatibility constraints formulated by Lighthill [41], expressed here symbolically as

$$C_1(V_o(s)) = 0, \quad (3.2)$$

$$C_2(V_o(s)) = 0, \quad (3.3)$$

$$C_3(V_o(s)) = 0. \quad (3.4)$$

If the prescribed target velocity distribution $V_o(s)$ does not satisfy the constraints (3.2)-(3.4) then it has to be modified. This can be accomplished by introducing free parameters, represented by a vector \mathbf{p} , into the definition of $V_o(s)$. Since there are three equations to be satisfied, in order to have a determined system of equations, \mathbf{p} must be a vector with three elements. The

[†]The boundary condition is assumed to be included in the notation.

notation $V_p(s, \mathbf{p})$ is introduced for the parameterized target velocity distribution. For reconstruction type of inverse problems, i.e. cases in which one tries to reconstruct an existing airfoil, $V_o(s)$ already satisfies the constraints. In such cases, the solution will be $\mathbf{p} = \mathbf{p}^* = \mathbf{p}_o$ with $V_p(s, \mathbf{p}_o) \equiv V_o(s)$.

The solution of the inverse problem satisfies

$$f(\mathbf{x}^*(s), V_p(s, \mathbf{p})) = 0, \quad (3.5)$$

$$C_1(V_p(s, \mathbf{p})) = 0, \quad (3.6)$$

$$C_2(V_p(s, \mathbf{p})) = 0, \quad (3.7)$$

$$C_3(V_p(s, \mathbf{p})) = 0. \quad (3.8)$$

In order to solve equations (3.5)-(3.8), the inverse problem is posed as an optimization problem with the objective functional to be minimized taking the form

$$\mathcal{F} = \frac{1}{2} \int_0^1 (V(s) - V_p(s, \mathbf{p}))^2 ds, \quad (3.9)$$

where $V(s)$ is the velocity distribution obtained by solving the flow equation

$$f(\mathbf{x}(s), V(s)) = 0,$$

for an intermediate closed contour airfoil shape, $\mathbf{x}(s)$. The coordinates of the airfoil shape are a function of a vector of geometric parameters $\boldsymbol{\theta}$ (to be described in the next section):

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\theta}).$$

The optimal solution sought for is

$$\boldsymbol{\theta} = \boldsymbol{\theta}^*,$$

$$\mathbf{p} = \mathbf{p}^*,$$

such that $\mathcal{F} = 0$. Since by definition $V(s)$ satisfies the Lighthill constraints, the condition $\mathcal{F} = 0$ implies that $V_p(s, \mathbf{p}^*)$ satisfies the Lighthill constraints.

The optimization problem is solved by a gradient-based optimization algorithm requiring the computation of the gradient of \mathcal{F} with respect to $\boldsymbol{\theta}$ and \mathbf{p} .

3.3 Parametric Representation of the Airfoil Shape

Consistent treatment of the inverse problem requires a geometric parameterization that guarantees closure of the trailing edge. Figure 3.1 shows a control curve defined by the coordinates $(\xi(\tau), \zeta(\tau))$, where $\tau \in [0, 1]$. The horizontal and vertical gaps between the end-points of the curve can be expressed as, respectively,

$$\Delta\xi = \xi(1) - \xi(0) = \int_0^1 \frac{\partial\xi(\tau)}{\partial\tau} d\tau, \quad (3.10)$$

$$\Delta\zeta = \zeta(1) - \zeta(0) = \int_0^1 \frac{\partial\zeta(\tau)}{\partial\tau} d\tau. \quad (3.11)$$

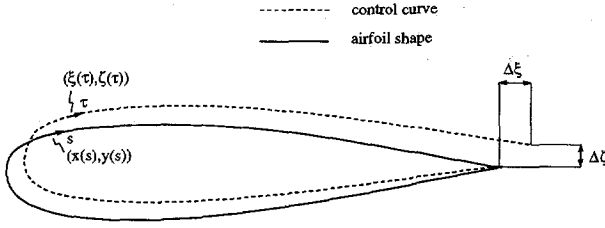


Figure 3.1: Control curve and airfoil shape

The coordinates of an airfoil shape with a closed trailing edge can then be expressed as

$$x(\tau) = x_{TE} + \frac{1}{S} \int_0^\tau \left(\frac{\partial \xi(\tau)}{\partial \tau} - \Delta \xi \right) d\tau, \quad (3.12)$$

$$y(\tau) = y_{TE} + \frac{1}{S} \int_0^\tau \left(\frac{\partial \zeta(\tau)}{\partial \tau} - \Delta \zeta \right) d\tau. \quad (3.13)$$

The scaling factor S is introduced to assure $s \in [0, 1]$,

$$S = \int_{\tau=0}^{\tau=1} \left(\left(\frac{\partial \xi(\tau)}{\partial \tau} - \Delta \xi \right)^2 + \left(\frac{\partial \zeta(\tau)}{\partial \tau} - \Delta \zeta \right)^2 \right)^{1/2} d\tau. \quad (3.14)$$

The trailing edge is fixed as follows,

$$(x, y)_{TE} = (1, 0). \quad (3.15)$$

The reason for choosing a unit total arc length ($s \in [0, 1]$) and a fixed trailing coordinate is to simplify the treatment of the problem. A local arc segment of the airfoil and that of the control curve are related by

$$ds(\tau) = \frac{1}{S} \left(\left(\frac{\partial \xi(\tau)}{\partial \tau} - \Delta \xi \right)^2 + \left(\frac{\partial \zeta(\tau)}{\partial \tau} - \Delta \zeta \right)^2 \right)^{1/2} d\tau. \quad (3.16)$$

The arc length s on the airfoil at a parameter value τ can be determined from

$$s(\tau) = \frac{1}{S} \int_{\tau=0}^{\tau} ds(\tau). \quad (3.17)$$

The parameters $\partial \xi(\tau)/\partial \tau$ and $\partial \zeta(\tau)/\partial \tau$ determine the airfoil shape. These can be comprised into one parameter, representing the slope of the control curve, which is defined by

$$\theta(\tau) = \arctan \frac{\partial \zeta(\tau)}{\partial \xi(\tau)}. \quad (3.18)$$

For each distinct value of $\tau \in [0, 1]$, the slope $\theta(\tau)$ can be considered as representing a "component" of the vector θ . Since $\theta(\tau)$ is continuously defined over the control curve, θ is a vector of infinite dimensions[†].

[†]This will be reduced to a finite dimension by a discretization described in Section 3.6.

3.4 Parametric Representation of the Target Velocity Distribution

To enable an effective control over the compatibility constraints, the three velocity parameters contained in \mathbf{p} should influence V_o in the following sense (Ref. [66]):

- (a) The overall surface velocity level relative to the onset flow speed.
- (b) The difference between the upper and lower surface velocity levels.
- (c) The stagnation point location.

An essential feature of the target velocity parameterization must be that, even for a wide variation of \mathbf{p} , there is a close resemblance between $V_p(\mathbf{p})$ and V_o , in order that the (supposedly) important features of V_o are preserved best. It is further required that the presence of \mathbf{p} does not introduce any discontinuity in the target velocity distribution as a function of the arc length s . Following these requirements, the parameterization of the target velocity distribution is expressed in a general form as

$$V_p(s, \mathbf{p}) = V_p(s, p_1, p_2, p_3) = f_1(s, p_1) (f_2(s, p_2, V_o(f_3(s, p_3))) + V_o(f_3(s, p_3))), \quad (3.19)$$

where

$$\begin{aligned} f_1 &= p_1, \quad \text{for } 0 \leq s \leq 1, \\ f_2 &= \begin{cases} -p_2 \frac{V_o(f_3(s, p_3)) - V_o(s_1)}{V_o(0) - V_o(s_1)}, & \text{for } 0 \leq s \leq s_1 + p_3, \\ 0, & \text{for } s_1 + p_3 < s \leq s_2 + p_3, \\ p_2 \frac{V_o(s_2) - V_o(f_3(s, p_3))}{V_o(s_2) - V_o(1)}, & \text{for } s_2 + p_3 < s \leq 1, \end{cases} \\ f_3 &= \begin{cases} \frac{s_1}{s_1 + p_3} s, & \text{for } 0 \leq s \leq s_1 + p_3, \\ s - p_3, & \text{for } s_1 + p_3 < s \leq s_2 + p_3, \\ \frac{1 - s_2}{1 - s_2 - p_3} (s - s_2 - p_3) + s_2, & \text{for } s_2 + p_3 < s \leq 1. \end{cases} \end{aligned}$$

The parameters s_1 and s_2 are obtained from the conditions

$$\begin{aligned} \frac{\partial V_o(s_1)}{\partial s} &= 0, \quad s_1 \text{ is on the lower surface} \\ \frac{\partial V_o(s_2)}{\partial s} &= 0, \quad s_2 \text{ is on the upper surface.} \end{aligned}$$

It can easily be verified that $V_p(s, \mathbf{p}) \equiv V_o(s)$ when $\mathbf{p} = \mathbf{p}_o = (1 \ 0 \ 0)^T$.

It should be noted that V_o as a function of s is usually given in a tabular form. In this case, $V_o(s)$ is constructed by means of a piecewise cubic spline.

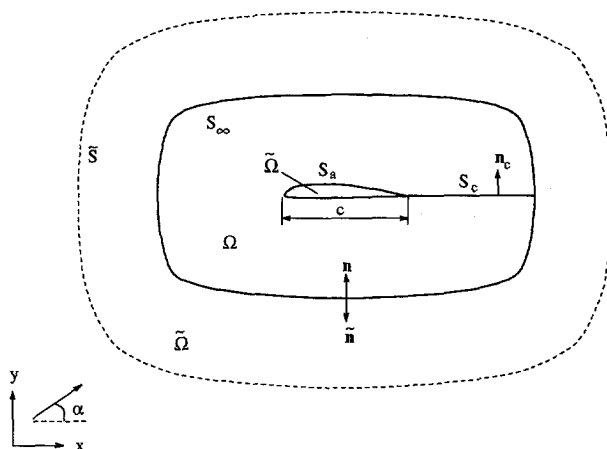


Figure 3.2: Domain definition for the incompressible potential flow

3.5 Flow Equation

Figure 3.2 provides the nomenclature. Incompressible potential flows are governed by the Laplace equation:

$$\nabla^2 \phi = 0 \quad \text{in } \Omega, \quad (3.20)$$

where ϕ is a velocity potential defined by

$$\vec{\nabla} \phi = \vec{V}. \quad (3.21)$$

A perturbation velocity potential φ is introduced, such that

$$\phi = \phi_\infty + \varphi, \quad (3.22)$$

where

$$\phi_\infty = V_\infty x. \quad (3.23)$$

This implicitly states that the free stream is parallel to the x -axis. The Laplace equation can also be written in terms of φ :

$$\nabla^2 \varphi = 0. \quad (3.24)$$

Equation (3.20) is subject to the Neumann (solid-wall) boundary condition on the airfoil surface[†]:

$$\vec{\nabla} \phi \cdot \vec{n} = 0 \quad \text{on } S_a. \quad (3.25)$$

The slit S_c is introduced to model the circulation, Γ , for lifting cases. Additional conditions associated with the presence of the slit follow from the conservation

[†]The boundary-layer is not included so that the normal component of the surface velocity is zero.

of mass and momentum jump conditions across S_c :

$$[\vec{\nabla}\varphi \cdot \vec{n}]_c = 0, \quad (3.26)$$

$$[\varphi]_c = \text{constant}. \quad (3.27)$$

The Kutta condition, representing the requirement for a finite and continuous velocity at the trailing edge, fixes the circulation:

$$\Gamma = [\varphi]_c. \quad (3.28)$$

This has the consequence that on the far-field boundaries,

$$\phi = \phi_\infty + \varphi_\infty \quad \text{on } S_\infty, \quad (3.29)$$

where

$$\varphi_\infty = \left(\frac{1}{4} \text{sign}(y_\infty - y_{TE}) + \frac{1}{2\pi} \arctan \left(\frac{x_\infty - x_{TE}}{y_\infty - y_{TE}} \right) \right) \Gamma. \quad (3.30)$$

The perturbation velocity at S_∞ is zero, such that

$$\vec{\nabla}\varphi \cdot \vec{n} = 0 \quad \text{on } S_\infty. \quad (3.31)$$

In order to obtain the solution of the Laplace equation (3.24), a boundary integral equation representing a source and a doublet distribution along the surface is considered (Ref. [52]). In order to satisfy the external Neumann boundary condition (3.25), which defines the zero normal component of the velocity, an internal Dirichlet boundary condition is imposed, i.e.

$$\varphi = 0 \quad \text{on } P \in \tilde{\Omega}.$$

The domain $\tilde{\Omega}$ is defined as the domain outside Ω , which includes the domain inside the airfoil. The boundary integral equation to be solved takes the form

$$\int_S (\varrho \vec{\nabla}\varphi \cdot \vec{n} - \varphi \vec{\nabla}\varrho \cdot \vec{n}) dS = 0 \quad \text{on } P \in \tilde{S}, \quad (3.32)$$

where ϱ and $\vec{\nabla}\varphi \cdot \vec{n}$ have the physical interpretation as the perturbation velocity potentials in \tilde{P} induced by, respectively, a source and a doublet density of unit strength, situated in a point $Q \in S$, where the point Q is moving along with the integral variable. The expressions for ϱ and $\vec{\nabla}\varphi \cdot \vec{n}$ are known to be (Ref. [2]):

$$\begin{aligned} \varrho &= \frac{1}{2\pi} \ln \|\vec{r}\|, \\ \vec{\nabla}\varphi \cdot \vec{n} &= -\frac{1}{2\pi} \frac{\vec{n} \cdot \vec{r}}{\|\vec{r}\|^2}. \end{aligned}$$

The first term in the integral of equation (3.32) represents a source distribution with the strength per unit length:

$$\sigma = \vec{\nabla}\varphi \cdot \vec{n}. \quad (3.33)$$

The second term represents a doublet distribution with the strength per unit length:

$$\mu = -\varphi. \quad (3.34)$$

On S_a , the Neumann boundary condition (3.25) and the definition of φ in Equation (3.22) fix the value of σ :

$$\sigma = \vec{\nabla}\varphi \cdot \vec{n} = -\vec{\nabla}\phi_\infty \cdot \vec{n} \quad \text{on } S_a. \quad (3.35)$$

On S_c , the jump condition (3.26) implies that there is no source distribution, i.e. $\sigma = 0$, whereas the condition (3.27) dictates a constant doublet distribution with the strength $\mu_c = -[\varphi]_c$ per unit length. The Kutta condition is formulated as the relation $\mu_c = \mu_{TE}$. For points on the airfoil, the contribution from the integration over S_∞ vanishes due to the large distance. The integral equation then becomes

$$\int_{S_a} (\sigma \varrho + \mu \vec{\nabla}\varrho \cdot \vec{n}) dS + \mu_{TE} \int_{S_c} \vec{\nabla}\varrho \cdot \vec{n} dS = 0. \quad \text{on } P \in \tilde{S}_a \quad (3.36)$$

From Equation (3.35), it follows that the airfoil shape also explicitly defines σ . Hence, for a given airfoil, μ and μ_{TE} are the unknowns to be determined from Equation (3.36). Recalling Equations (3.34), (3.25), (3.22), and (3.21), the velocity V on the airfoil surface can be obtained from

$$V = \vec{\nabla}\phi \cdot \vec{s} = \vec{\nabla}\phi_\infty \cdot \vec{s} + \vec{\nabla}\varphi \cdot \vec{s} = \vec{\nabla}\phi_\infty \cdot \vec{s} - \nabla\mu \cdot \vec{s}, \quad (3.37)$$

with \vec{s} the unit tangential vector along the airfoil. $\vec{\nabla}\phi_\infty \cdot \vec{s}$ is explicitly dependent on the airfoil shape. The objective functional (3.9) can now be written in the following form,

$$\mathcal{F} = \frac{1}{2} \int_{S_a} (\vec{\nabla}\phi_\infty \cdot \vec{s} + \vec{\nabla}\varphi \cdot \vec{s} - V_p(s, \mathbf{p}))^2 dS. \quad (3.38)$$

3.6 Discretization

The control curve in Figure 3.1 is discretized into N straight segments, implying an N -dimensional vector θ :

$$\theta = (\theta_1 \cdots \theta_N)^T.$$

The vector component θ_i is the slope of the i -th segment of the control curve. The parameter vector θ controls the shape of the airfoil consisting of N planar panels. A collocation point is defined at the center of each panel. The airfoil coordinates are given by

$$x_i = x_{TE} + \frac{1}{S} \sum_{j=1}^{i-1} \Delta\tau_j (\cos \theta_j - \Delta\xi), \quad (3.39)$$

$$y_i = y_{TE} + \frac{1}{S} \sum_{j=1}^{i-1} \Delta\tau_j (\sin \theta_j - \Delta\zeta), \quad (3.40)$$

where θ_j and $\Delta\tau_j$ are the slope and the width of the j -th segment of the control curve, respectively, and:

$$\Delta\xi = \sum_{j=1}^N \Delta\tau_j \cos \theta_j,$$

$$\Delta\zeta = \sum_{j=1}^N \Delta\tau_j \sin \theta_j.$$

The length of the i -th panel is

$$\Delta s_i = \frac{1}{S} \left((\cos \theta_i - \Delta\xi)^2 + (\sin \theta_i - \Delta\zeta)^2 \right)^{1/2} \Delta\tau_i,$$

and its slope, β_i , can be determined from

$$\cos \beta_i = \frac{\cos \theta_i - \Delta\xi}{\Delta s_i},$$

$$\sin \beta_i = \frac{\sin \theta_i - \Delta\zeta}{\Delta s_i}.$$

The parameter S is defined by

$$S = \sum_{i=1}^N \left((\cos \theta_i - \Delta\xi)^2 + (\sin \theta_i - \Delta\zeta)^2 \right)^{1/2} \Delta\tau_i.$$

On each panel, the source and doublet distributions are approximated to be constant. The doublet strength over the slit S_c is approximated as

$$\mu_{TE} = \mu_N - \mu_1. \quad (3.41)$$

The integration over S_a in Equation (3.36) is approximated by a summation of integrals over each panel. Taking the collocation points for the points $\tilde{P} \in \tilde{S}_a$, then the discrete form of the flow equation is obtained as

$$A\sigma + B\mu = 0, \quad (3.42)$$

where σ and μ are N -dimensional vectors of the source and doublet strengths, respectively. The i -th component of these vectors is associated with the i -th panel. A and B are $N \times N$ matrices of aerodynamic influence coefficients due to the source and doublet distribution, respectively. A , B , and σ are functions of the vector of design variables θ .

Solving equation (3.42) provides the doublet strengths μ . The perturbation velocity $\vec{\nabla}\varphi \cdot \vec{s}$ at each collocation point is calculated by finite differencing,

$$(\vec{\nabla}\varphi \cdot \vec{s})_i = \left(\frac{\varphi_{i+1} - \varphi_{i-1}}{s_{i+1} - s_{i-1}} \right) = - \left(\frac{\mu_{i+1} - \mu_{i-1}}{s_{i+1} - s_{i-1}} \right),$$

where s_i denotes the arc length of the i -th collocation point on the airfoil:

$$s_i = \frac{1}{2} \Delta s_i + \sum_{j=1}^{i-1} \Delta s_j.$$

Finally, equation (3.37) gives the total velocity:

$$V_i = (\vec{\nabla} \phi_\infty \cdot \vec{s})_i + (\vec{\nabla} \varphi \cdot \vec{s})_i.$$

The objective functional (3.9) is discretized in the form

$$\mathcal{F} = \frac{1}{2} \sum_{i=1}^N \left((\vec{\nabla} \phi_\infty \cdot \vec{s})_i + (\vec{\nabla} \varphi \cdot \vec{s})_i - V_p(s_i, \mathbf{p}) \right)^2 \Delta s_i. \quad (3.43)$$

3.7 Formulation of the Adjoint and Gradient Equations

Employing a gradient-based optimization algorithm for minimizing the functional (3.43) requires the computation of the gradient of \mathcal{F} with respect to the design variables θ and \mathbf{p} . The gradient of \mathcal{F} with respect to \mathbf{p} can be obtained by direct analytical differentiation.

The gradient of \mathcal{F} with respect to θ is obtained by the variational method. This means that an adjoint problem must be formulated in order to obtain the gradient. A Lagrangian is defined as

$$\mathcal{L} = \mathcal{F} + \lambda \cdot (A\sigma + B\mu), \quad (3.44)$$

where λ is the vector of Lagrange multipliers, each component of which is associated with a panel. Explicit dependencies in the above equation are indicated as follows[†]

$$\begin{aligned} \mathcal{L} &= \mathcal{L}(\lambda, \mu, \theta), \\ \mathcal{F} &= \mathcal{F}(\mu, \theta), \\ A &= A(\theta), \\ B &= B(\theta), \\ \sigma &= \sigma(\theta), \end{aligned}$$

The partial derivatives are obtained by direct analytical differentiation. The variation of \mathcal{L} is expressed as

$$\delta \mathcal{L} = (A\sigma + B\mu) \cdot \delta \lambda + \left(\frac{\partial \mathcal{F}}{\partial \mu} + B^\top \lambda \right) \cdot \delta \mu + \left(\frac{\partial \mathcal{F}}{\partial \theta} + J_\theta^\top \lambda \right) \cdot \delta \theta,$$

where J_θ is the Jacobian of the flow equation with respect to θ :

$$J_\theta = \frac{\partial}{\partial \theta} (A\sigma + B\mu).$$

Flow equation (3.42) cancels the contribution in $\delta \mathcal{L}$ due to $\delta \lambda$. The adjoint equation is obtained by setting the contribution of $\delta \mu$ to zero, i.e.

$$\frac{\partial \mathcal{F}}{\partial \mu} + B^\top \lambda = 0. \quad (3.45)$$

[†]For brevity the vector \mathbf{p} is dropped.

which yields the column vector λ . With both the flow and adjoint equations satisfied, the gradient of \mathcal{F} with respect to the design variables θ is obtained as

$$\frac{d\mathcal{F}}{d\theta} = \frac{\partial \mathcal{F}}{\partial \theta} + J_{\theta}^T \lambda \quad (3.46)$$

The partial derivatives in the right-hand of equation (3.46) side are obtained by direct analytical differentiation.

3.8 Optimization Routine

The present investigation employs the optimization routine E04UCF of the NAG numerical libraries available on NLR's computer system. This routine is based on the Sequential Quadratic Programming (SQP) algorithm which is widely known to be the most cost-effective method for non-linear optimization. For each iteration, the algorithm requires the information on the function value of the objective and its gradient for determining a new iterate in order to reduce the value of the objective. A detailed description of the algorithm is given in Ref. [1].

3.9 Results and Discussion

Three cases are considered, namely:

1. Case L-1: Exact reconstruction problem. The prescribed target velocity distribution is taken from the panel method calculation of a NACA 4418 airfoil at zero angle of attack. The prescribed target is fully produceable.
2. Case L-2: Redesign problem. The prescribed target velocity distribution is the result of a *viscous* analysis calculation defined in [62]. Since the inverse methodology described in this paper is based on the inviscid flow model for closed airfoil contours, the prescribed target is only close to being produceable.
3. Case L-3: Total design problem. The prescribed target velocity distribution is defined by a B-spline control polygon. The prescribed target has a very low degree of produceability.

For all cases, a NACA 0012 airfoil is taken as the initial airfoil geometry. The design space consists of 64 geometric parameters (components of θ) and 3 velocity parameters (components of p).

The convergence history of the three cases is given in Figure 3.3. Each function call requires one flow analysis. Observing the slope of the curves, the rate of convergence for the three cases are in general of the same order of magnitude.

For a reconstruction problem in subcritical flow, the residual-correction inverse method of Fray et al. [17] requires about 3 iterations to reach a deviation in C_p of about 10^{-2} . Recalling that in incompressible flow,

$$C_p = 1 - V^2,$$

if V is of order one, i.e. $V = O(1)$, one obtains

$$\Delta C_p \approx 2\Delta V$$

where Δ denotes a small deviation. This implies that $\Delta C_p \approx 10^{-2}$ is roughly comparable to $\mathcal{F} \approx 1.25 \times 10^{-5}$. To reach this, the present method needs 21 iterations, which is rather high compared to the method of Fray. This could partly be attributed to an unfavorable design space topology implied by the chosen set of design variables[†]. However, the present method has the advantage of the possibility of obtaining well-posedness in a consistent way by means of the three free parameters of the velocity distribution.

Figure 3.4 shows the result of case L-1 in terms of velocity distribution. The corresponding pressure coefficient and airfoil shape are given in Figure 3.5. The prescribed target, the parameterized target and the actual velocity distributions are coincident at all locations. This confirms that the velocity parameters are not significant for cases in which the compatibility constraints are already satisfied. As can be seen, the NACA 4418 airfoil geometry has been closely reconstructed.

The results of case L-2 are shown in Figure 3.6 and 3.7. There are slight differences between the prescribed and the parameterized targets. However, the parameterized target is matched by the actual velocity distribution. The corresponding pressure coefficient and airfoil shape are shown in Figure 3.7.

The prescribed target velocity distribution of case L-3 can be seen in Figure 3.8. An inverse code without target velocity parameterization based on the residual-correction method developed at the NLR, which works fine for reconstruction cases, completely failed for case L-3. The present method, on the other hand, is still able to give an acceptable result as shown in Figure 3.8. Although there is a significant deviation from the prescribed target, the actual velocity matches the parameterized target. One should also observe the strong resemblance between the prescribed and the parameterized targets, especially on the upper surface. The corresponding pressure coefficient and airfoil shape are presented in Figure 3.9.

Admittedly, case L-3 is rather unrealistic from the viewpoint of an experienced designer. The stagnation point location and the low velocity level on the lower surface are really uncommon. Figure 3.10 shows the original and modified target C_p distribution drawn as a function of x/c on the initial airfoil (NACA 0012). It is unlikely that an experienced designer would conceive such a target C_p distribution (the original nor the (realizable) modified one) for redesigning the initial airfoil. Nevertheless, such prescribed target suits the purpose of examining the robustness of the present method.

Confronted with the result of case L-3 where a prescribed target velocity distribution was modified considerably, one might suspect that the inverse procedure has ignorantly varied the velocity parameters to find whatever geometry in an easy way. Essential, in this respect, is the precise form of the parameterization (Equation (3.19) of the target velocity distribution. To investigate this,

[†]I.e., different (geometric and target velocity distribution) parameterization schemes may give different rates of convergence.

a non-linear constrained minimization problem is formulated where the Lighthill constraints are imposed explicitly. The constraints are expressed in the form formulated by Strand [57]. The problem is stated below:

$$\begin{aligned}
 &\text{Minimize} && \|\mathbf{p} - \mathbf{p}_o\|^2 \\
 &\text{subject to:} && \\
 &C_1 = && \int_0^{2\pi} \ln |V_p(\omega, \mathbf{p})| d\omega = 0, \\
 &C_2 = && \int_0^{2\pi} \ln |V_p(\omega, \mathbf{p})| \cos \omega d\omega + 2\pi \sin^2 \alpha = 0, \\
 &C_3 = && \int_0^{2\pi} \ln |V_p(\omega, \mathbf{p})| \sin \omega d\omega - \pi \sin 2\alpha = 0,
 \end{aligned} \tag{3.47}$$

where α is the angle of attack and ω is the polar angle on the circle plane. The problem is, essentially, to satisfy the Lighthill constraints with the smallest possible modification to the prescribed target velocity distribution. The design variable is \mathbf{p} , which is initially set equal to \mathbf{p}_o . The airfoil geometry is not involved here. In the constraints C_1 , C_2 , and C_3 , the parameterized velocity distribution V_p is expressed in terms of ω instead of the arc length s as given. Hence, a relationship between s and ω must be determined first. Figure 3.11 shows a circle plane and its conformally mapped airfoil plane. For consistency with the orientation of ω , an arc length variable $\eta \in [0, 1]$ on the airfoil is introduced, such that

$$\eta = 1 - s,$$

and accordingly, the velocity along the airfoil in terms of η is

$$V_a(\eta, \mathbf{p}) = -V_p(1 - s, \mathbf{p}).$$

The velocity along the circle, $V_c(\omega)$, is known to be:

$$V_c(\omega) = -2(\sin(\omega - \alpha) + \sin \alpha), \tag{3.48}$$

For a conformal transformation between the two planes, the following applies:

$$V_a(\eta, \mathbf{p}) d\eta = V_c(\omega) R d\omega, \tag{3.49}$$

where R is the circle radius. This defines a general relationship between η and ω :

$$\int_0^\eta V_a(\eta, \mathbf{p}) d\eta = \int_0^\omega V_c(\omega) R d\omega = -2R(\cos \alpha - \cos(\omega - \alpha) + \omega \sin \alpha). \tag{3.50}$$

Two quantities, namely R and α , have still to be determined. Using Equation (3.48), and applying Equation (3.50) for $\eta = 1$ and $\omega = 2\pi$ yield the circulation:

$$\begin{aligned}
 \Gamma &= - \int_0^1 V_a(\eta) d\eta \\
 &= - \int_0^{2\pi} V_c(\omega) R d\omega \\
 &= 4\pi R \sin \alpha.
 \end{aligned} \tag{3.51}$$

Similarly, for the forward stagnation point, where $\eta = \eta_0$ and $\omega = \pi + 2\alpha$:

$$\begin{aligned} K &= - \int_0^{\eta_0} V_a(\eta) d\eta \\ &= - \int_0^{\pi+2\alpha} V_c(\omega) R d\omega \\ &= -2R(2 \cos \alpha + (\pi + 2\alpha) \sin \alpha). \end{aligned}$$

Eliminating R by combining the last two equations results in

$$K + \frac{\Gamma}{2\pi \sin \alpha} (2 \cos \alpha + (\pi + 2\alpha) \sin \alpha) = 0. \quad (3.52)$$

Solving this equation provides α . Substituting α into Equation (3.51) yields R . As α and R are available, for a specified η —or equivalently, s —the corresponding ω can be determined by solving Equation (3.50). In the numerical implementation, a piecewise linear representation is used for both $V_p(s, \mathbf{p})$ and $V_p(\omega, \mathbf{p})$. This circumvents singularities at the stagnation point associated with the natural logarithm terms of the constraints. Equations (3.52) and (3.50) are solved by means of the Newton-Raphson method.

The prescribed target velocity distribution of case L-3 is fed into the minimization problem (3.47). The initial state is

$$\begin{aligned} \mathbf{p} &= \mathbf{p}_0 = (1 \quad 0 \quad 0)^T, \\ C_1 &= -9.890 \times 10^{-1}, \\ C_2 &= -8.136 \times 10^{-1}, \\ C_3 &= -6.150 \times 10^{-1}. \end{aligned}$$

E04UCF NAG routine is also used to solve this problem. The required partial derivatives of the constraints with respect to \mathbf{p} are calculated by finite differences. At the optimal solution:

$$\begin{aligned} \mathbf{p} &= (0.964 \quad 0.560 \quad 0.005)^T, \\ C_1 &= -1.054 \times 10^{-7}, \\ C_2 &= 5.346 \times 10^{-8}, \\ C_3 &= 9.564 \times 10^{-8}. \end{aligned}$$

The resulting velocity distribution is shown in Figure 3.12, where it is compared with the result of case L-3. Reasonable agreement is shown. Ideally, the three curves should be coincident. However, since the numerical modeling in the inverse procedure and in solving problem (3.47) are completely different, the small discrepancies do not undermine the significance of the result: the modification of the prescribed target velocity distribution that has taken place in the case L-3 is minimal in the sense of problem (3.47), confirming that the target velocity parameterization is a suitable one.

3.10 Conclusion

An optimization type method has been formulated for solving the inverse problem of aerodynamics for incompressible potential flow. Well-posedness of the inverse problem is guaranteed through the introduction of free velocity parameters.

The gradient of the objective functional required by the optimization method is computed by means of the variational method. The adjoint and gradient equations are formulated based on the discretized flow equation. With the low-order panel method used for the discretization of the flow equation, the adjoint and gradient equations can be formulated in a rather simple manner.

The computational results have demonstrated the robustness of the present methodology in dealing with general inverse airfoil problems. The introduction of free target velocity parameters as design variables, in addition to the geometric parameters, in the optimization formulation has shown to be a viable way for obtaining well-posedness of the inverse problem. However, in comparison with other inverse methods, the method does not seem to be very efficient for cases in which the specified target velocity distribution is (close to) a feasible one in terms of well-posedness.

The present method has a unique feature in the sense that the designer is not required to have experience in specifying feasible target velocity distributions. From computational results, the designer will learn whether his input is a proper one. In this way, expertise is gained instead of being required.

The approach of formulating the inverse problem as an optimization problem and of introducing velocity parameters in obtaining well-posedness offers several advantages:

- (a) *Extendability with regard to the flow model.* The Lighthill constraints are indirectly taken into account in the optimization formulation through the introduction of the free velocity parameters. Extending the approach to more complex flow models —where closed-form expressions of such constraints are not (yet) available—is relatively straightforward.
- (b) *Extendability with regard to the type of constraint.* The Lighthill constraints requiring closure of the trailing edge are special cases of geometric constraints. If other types of geometric constraints are imposed (e.g. thickness constraints, volume constraints), additional velocity parameters may be introduced. Of course, the additional parameterization must be defined in such a way that the state of the constraints can be effectively be altered.

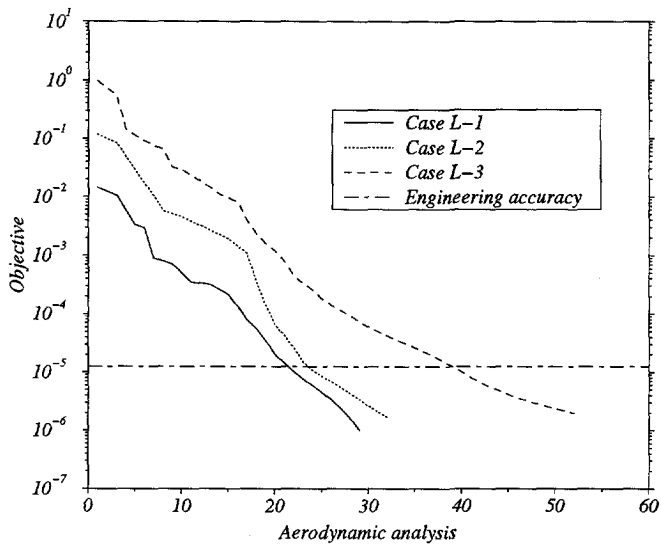


Figure 3.3: Design iteration history

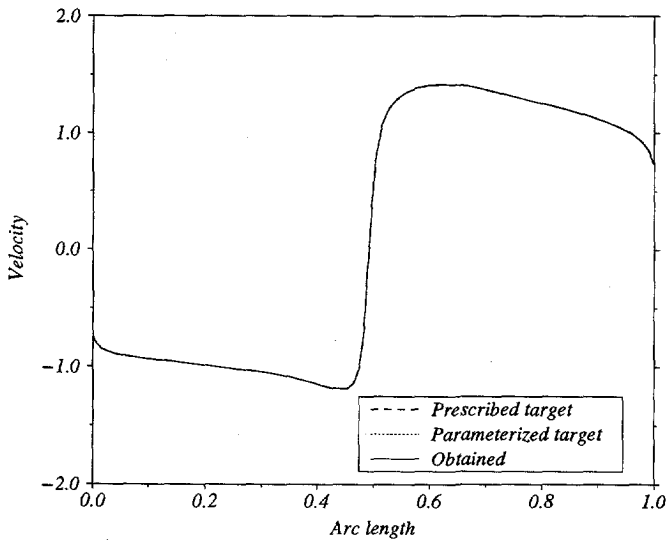


Figure 3.4: Velocity distributions of Case L-1

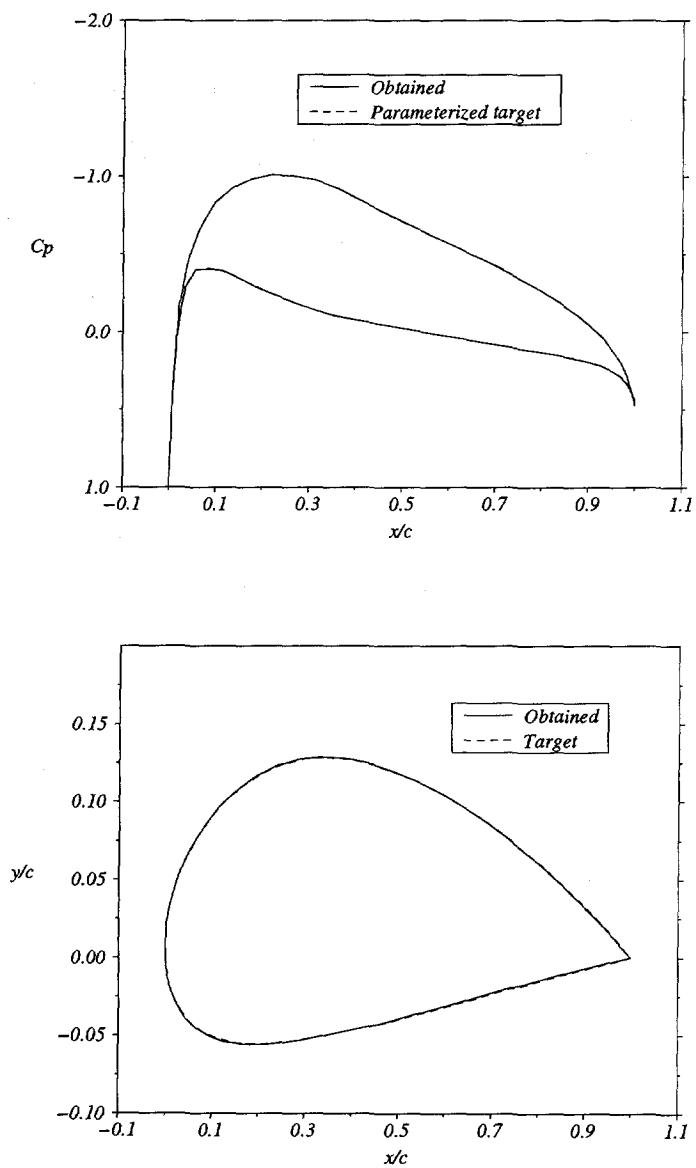


Figure 3.5: C_p distributions and airfoil geometries ($\alpha = 0.$) of Case L-1.

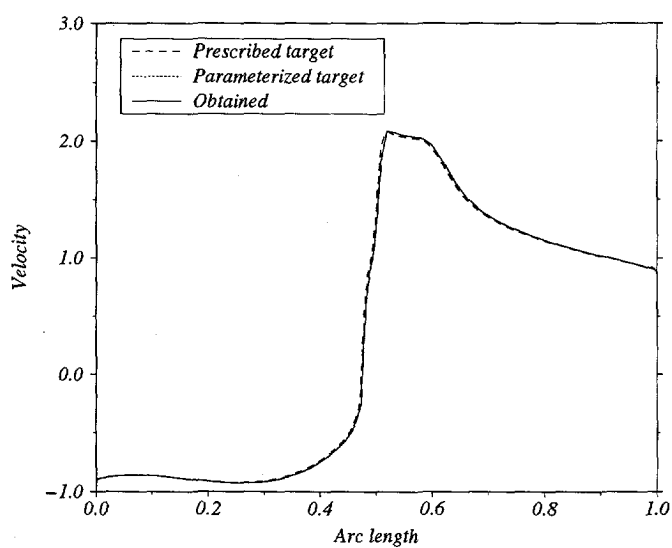


Figure 3.6: Velocity distributions of Case L-2

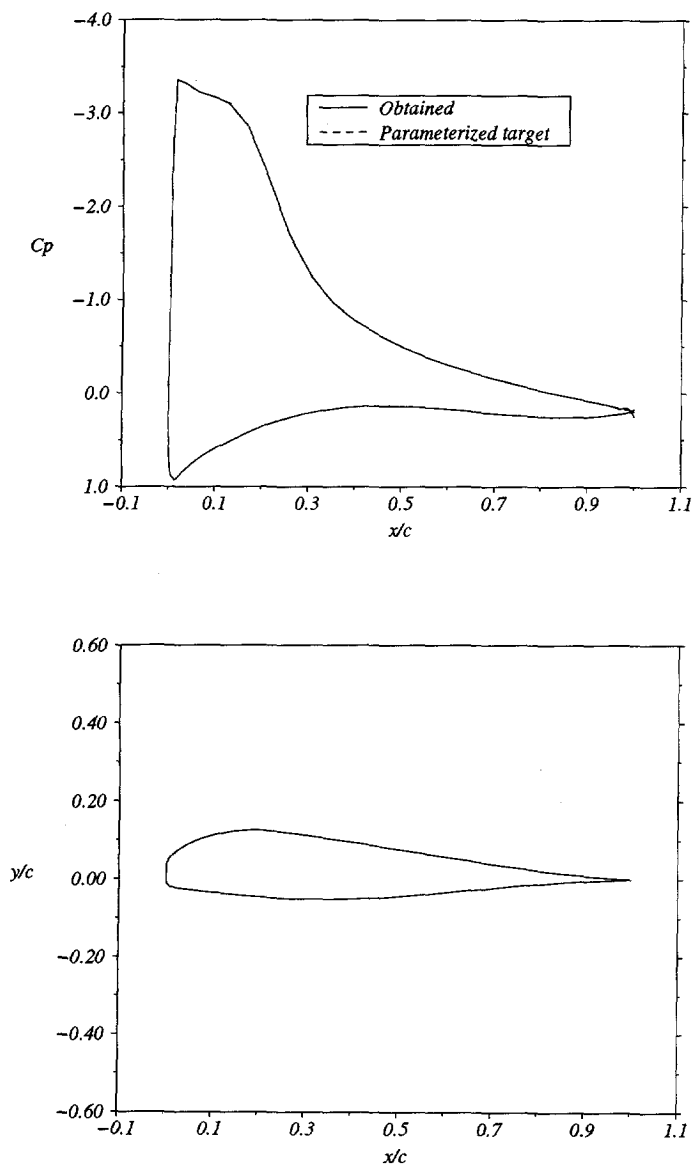


Figure 3.7: C_p distributions and airfoil geometry ($\alpha = 8.62^\circ$) of Case L-2

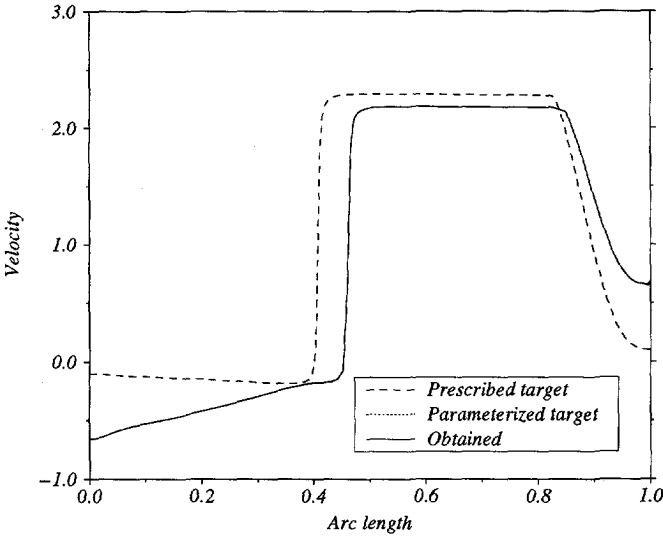


Figure 3.8: Velocity distributions of Case L-3

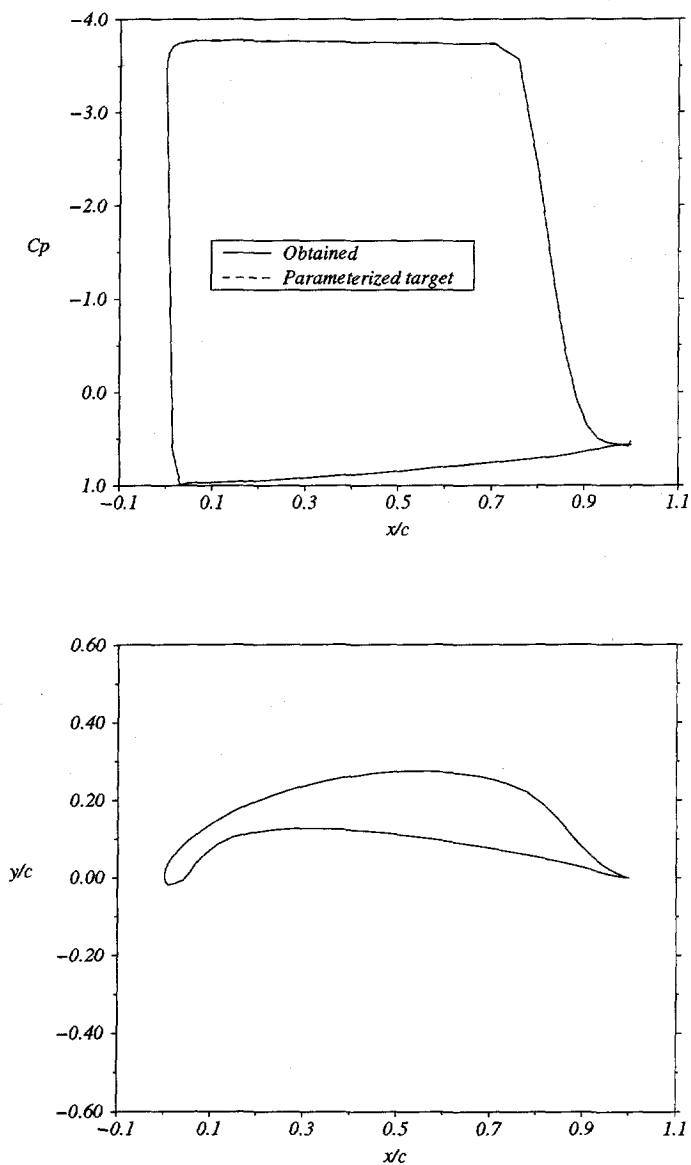


Figure 3.9: C_p distributions and airfoil geometry ($\alpha = 12.3^\circ$) of Case L-3

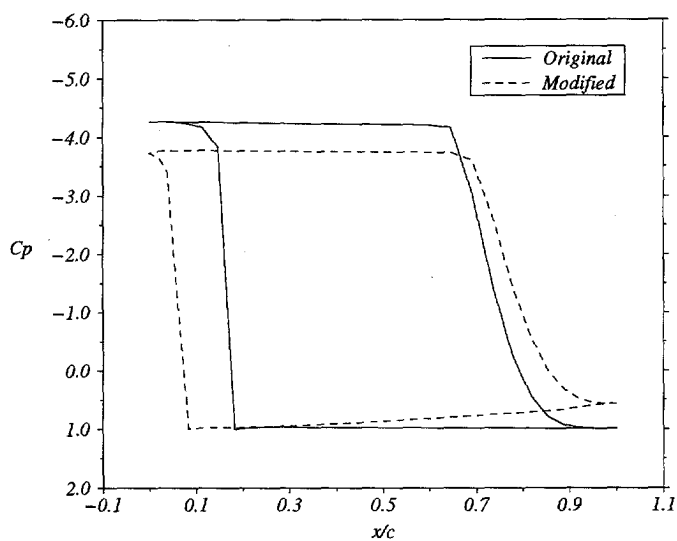


Figure 3.10: Target C_p of Case L-3 as a function of x/c of the initial airfoil

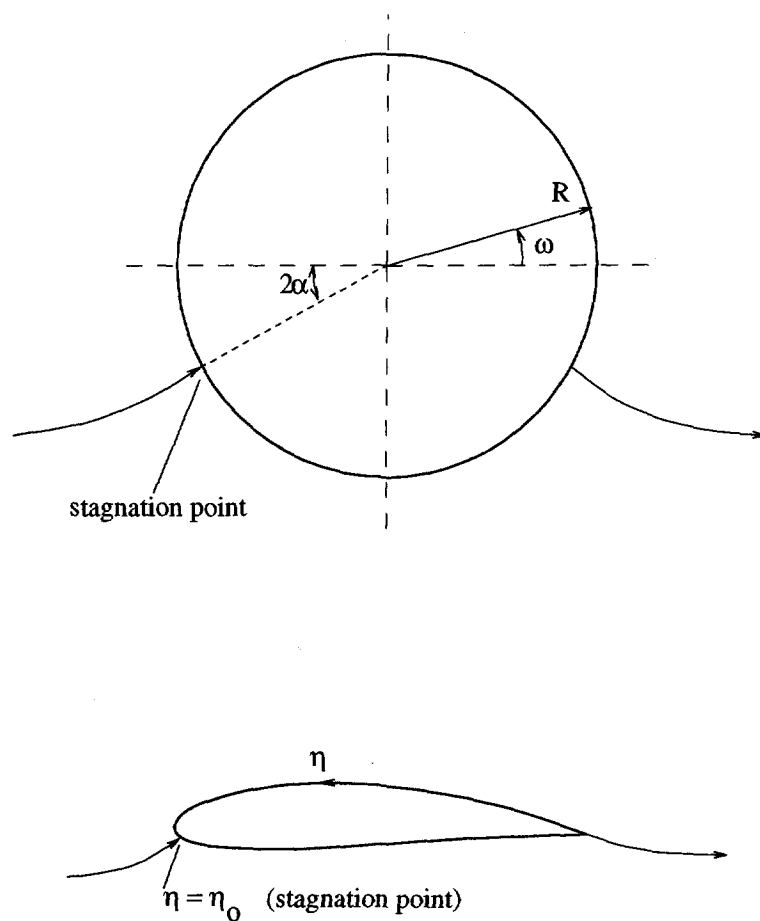


Figure 3.11: Circle and its conformally mapped airfoil

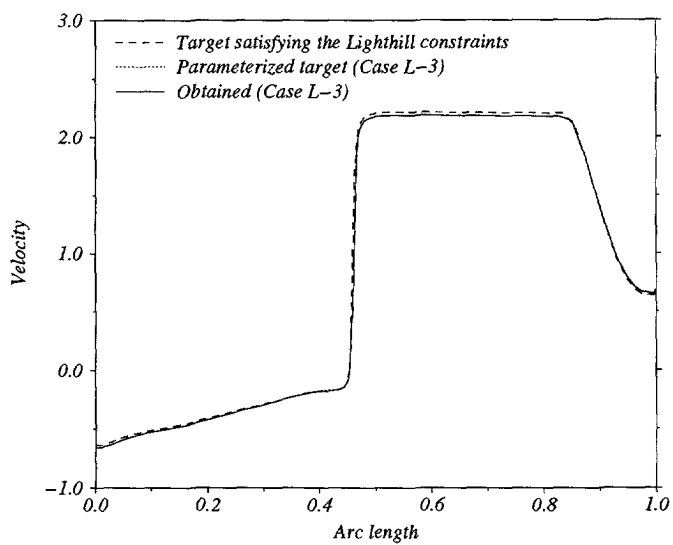


Figure 3.12: Comparison of velocity distributions

Chapter 4

Aerodynamic Optimization using the Euler Equations

4.1 Introduction

This chapter describes experiences obtained with applications of the variational method to aerodynamic design problems employing the Euler equations for inviscid compressible flow as the mathematical flow model. The main motive for this is, that it represents a logical intermediate step towards the further goal of aerodynamic design based on the (Navier-Stokes) equations for viscous compressible flow. The logic being that the numerical treatment of the Euler equations and those of the Navier-Stokes equations are closely connected. Apart from this, the compressible inviscid flow model can be useful in dealing with many aerodynamic design problems. For example, in transonic cruise design points, the performance of airfoils can be improved by reducing the drag due to shock waves. In practice, designing for reduced wave drag with an inviscid flow model can lead to significant improvements for the real viscous flow situation.

This chapter concerns transonic airfoil design problems. The design problems are posed as optimization problems involving aerodynamic functionals representing the lift (C_l), drag (C_d), and pitching moment (C_m) coefficients. These are defined in terms of the pressure coefficient C_p obtained from the flow variables Q which satisfy the Euler equations.

The design variables consist of geometric parameters θ (defining the airfoil geometry) and an angle of attack α (defining the orientation of the free stream with respect to the airfoil geometry). The optimal values of θ and α are obtained by means of a gradient-based optimization algorithm. The gradients of the aerodynamic functionals with respect to the design variables θ and α are computed by means of the variational method.

The methodology presented in this chapter takes advantage of the availability of a flow solver (Ref. [6]) and optimization routine (Ref. [69]). The main task to be done is then to develop an adjoint solver and a gradient evaluator, and to integrate these with the existing flow solver and optimization routine into a design code.

This chapter is organized as follows. A statement of the design problem is given in Section 4.2. A brief description of the flow model based on the Euler equations is given in Section 4.3. The adjoint and gradient equations are formulated in Section 4.4. Sections 4.5 and 4.6 are related to the airfoil shape parameterization. A description of the numerical implementation of the adjoint and gradient equations is given in Section 4.7. The optimization routine used in the investigation is briefly described in Section 4.8. Test cases and computational results are discussed in Section 4.9. Finally, some conclusions are drawn in Section 4.10.

4.2 Statement of the Design Problem

The optimization problem being addressed can be written as follows[†],

$$\begin{aligned} &\text{Minimize} && \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta}, \alpha), \\ &\text{Subject to:} && \\ & && \mathcal{A}(\mathbf{Q}, \boldsymbol{\theta}, \alpha) \leq 0, \\ & && \mathbf{G}(\boldsymbol{\theta}) \leq 0, \end{aligned} \tag{4.1}$$

where \mathcal{P} and \mathcal{A} represent such aerodynamic functionals as lift, drag and pitching moment coefficients, while the vector \mathbf{G} represents geometric constraints. The geometric parameters $\boldsymbol{\theta}$ and the angle of attack α are treated as the design variables, of which the optimal values are to be determined.

Problem (4.1) is subject to the Euler equations for a given (fixed) value of the Mach number. The Euler equations impose an implicit dependency of the flow variables \mathbf{Q} upon $\boldsymbol{\theta}$ and α .

Problem (4.1) is to be solved by means of a gradient-based optimization algorithm. The gradient of \mathbf{G} can be obtained rather easily by direct analytical differentiation, whereas the gradients of the aerodynamic functionals \mathcal{P} and \mathcal{A} will be computed by means of the variational method. This implies that an adjoint problem must be formulated, the solution of which is used for evaluating the gradients.

4.3 The Euler Equations

Figure (4.1) illustrates the nomenclature. Assuming adiabatic flow and no external forces, the time-dependent Euler equations in the conservative form are written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \vec{\nabla} \cdot \vec{\mathbf{F}} = 0, \quad \text{in } \Omega, \tag{4.2}$$

[†]For the sake of brevity, the Mach number does not appear because this is fixed.

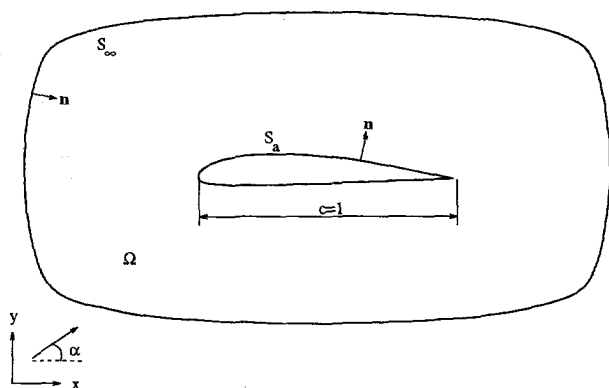


Figure 4.1: Domain definition for the compressible inviscid flow

where \mathbf{Q} is the vector of conservative flow variables:

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad (4.3)$$

which have been non-dimensionalized with respect to the free stream values of the flow variables. At the steady-state, equation (4.2) becomes

$$\vec{\nabla} \cdot \vec{\mathbf{F}} = 0. \quad (4.4)$$

The convective flux vector $\vec{\mathbf{F}}$ is defined as

$$\vec{\mathbf{F}} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \quad (4.5)$$

where

$$\mathbf{f} = \begin{pmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ (\rho E + p)u \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v v + p \\ (\rho E + p)v \end{pmatrix}. \quad (4.6)$$

The total energy E per unit mass is defined as

$$E = e + \frac{1}{2}(u^2 + v^2),$$

where e is the internal energy per unit mass. The system of equations is closed by the calorically perfect gas relation,

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)). \quad (4.7)$$

On the airfoil surface, S_a , the zero normal-velocity boundary condition holds[†]. This is expressed as

$$\rho \vec{V} \cdot \vec{n} = n_x \rho u + n_y \rho v = 0. \quad (4.8)$$

The boundary conditions on S_∞ are formulated based on the characteristic variables (Riemann invariants) of the one-dimensional Euler equation (taken in the direction normal to S_∞). The Riemann invariants can be obtained as

$$W_1 = \mathcal{S} \quad (4.9)$$

$$W_2 = \vec{V} \cdot \vec{s} \quad (4.10)$$

$$W_3 = \vec{V} \cdot \vec{n} + \frac{2a}{\gamma - 1} \quad (4.11)$$

$$W_4 = \vec{V} \cdot \vec{n} - \frac{2a}{\gamma - 1} \quad (4.12)$$

where \mathcal{S} is the entropy and a is the speed of sound. For *subsonic inflow*, the incoming Riemann invariants, W_+ , and the outgoing ones, W_- , can be identified as

$$W_+ = \begin{pmatrix} W_1 \\ W_2 \\ W_4 \end{pmatrix}, \quad W_- = W_- = W_3,$$

whereas for *subsonic outflow*:

$$W_+ = W_+ = W_4, \quad W_- = \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix}.$$

The incoming Riemann invariants follow from the specification of the free stream conditions, while the outgoing Riemann invariants are obtained from inside the domain by an extrapolation procedure. As the Riemann invariants are available, the flow variables on S_∞ are determined by solving equations (4.9)–(4.12) for Q .

4.4 Formulation of The Adjoint and Gradient Equations

For the design cases to be considered, a general form of aerodynamic functionals is assumed as follows

$$\mathcal{F} = \int_{S_a} \psi(p, \theta, \alpha) dS. \quad (4.13)$$

[†]The boundary-layer is not included so that the normal component of the surface velocity is zero.

where ψ is a function of the pressure p , the design variables θ , and the angle of attack α . The notation \mathcal{F} applies to both \mathcal{P} and \mathcal{A} of the problem statement (4.1).

It should be noted that global aerodynamic properties, such as wave drag, can be formulated as domain integrals and/or far-field surface integrals (Ref. [67]). The reason for choosing the formulation of pressure integrals over the airfoil surface, equation (4.13), is that the available flow solver does not have provisions for the other formulations.

Employing the variational method means that the adjoint equation must be formulated, the solution of which provides the Lagrange multipliers. These in turn are used in evaluating the gradient of the functional with respect to the design variables θ and α .

As p is obtained from \mathbf{Q} which satisfies the steady-state Euler equations, the functional \mathcal{F} is independent of the transient state. Therefore, it is sufficient to consider the steady-state Euler equations (4.4) and boundary condition (4.8) in the definition of a Lagrangian \mathcal{L} as follows,

$$\mathcal{L} = \int_{S_a} \psi \, dS + \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) \, d\Omega + \int_{S_a} \Upsilon (\rho \vec{V} \cdot \vec{n}) \, dS, \quad (4.14)$$

where $\boldsymbol{\lambda}$ and Υ are the Lagrange multipliers. $\boldsymbol{\lambda}$ is a vector with four components defined in Ω , each component of which may be considered as corresponding to a component of the conservative flow variables \mathbf{Q} . The Lagrange multiplier Υ is a scalar function defined on S_a .

In order to derive the adjoint and gradient equations, one must evaluate the variation of \mathcal{L} , denoted as $\delta\mathcal{L}$, due to the independent variables $\boldsymbol{\lambda}$, Υ , \mathbf{Q} , θ and α :

$$\delta\mathcal{L} = \delta\mathcal{L}_{\boldsymbol{\lambda}} + \delta\mathcal{L}_{\Upsilon} + \delta\mathcal{L}_{\mathbf{Q}} + \delta\mathcal{L}_{\theta} + \delta\mathcal{L}_{\alpha}.$$

The notation $\delta\mathcal{L}_{\boldsymbol{\lambda}}$ refers to the variation of $\delta\mathcal{L}$ implied by the variation of $\boldsymbol{\lambda}$ while the other variables are kept fixed, and similarly for $\delta\mathcal{L}_{\Upsilon}$, etc. The variations $\delta\mathcal{L}_{\boldsymbol{\lambda}}$, $\delta\mathcal{L}_{\Upsilon}$, and $\delta\mathcal{L}_{\mathbf{Q}}$ are evaluated with θ and α kept fixed. Keeping θ fixed implies a fixed domain Ω . For the variation of $\boldsymbol{\lambda}$, Υ , and \mathbf{Q} with a fixed domain an accent notation is introduced as $\boldsymbol{\lambda}'$, Υ' , and \mathbf{Q}' , respectively. The variation $\boldsymbol{\lambda}'$ contributes to $\delta\mathcal{L}$ with

$$\delta\mathcal{L}_{\boldsymbol{\lambda}} = \int_{\Omega} \boldsymbol{\lambda}' \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) \, d\Omega. \quad (4.15)$$

Satisfying the Euler equations (4.4) implies

$$\delta\mathcal{L}_{\boldsymbol{\lambda}} = 0. \quad (4.16)$$

The variation Υ' contributes with

$$\delta\mathcal{L}_{\Upsilon} = \int_{S_a} \Upsilon' (\rho \vec{V} \cdot \vec{n}) \, dS. \quad (4.17)$$

Satisfying the zero normal-velocity boundary condition (4.8) implies

$$\delta\mathcal{L}_{\Upsilon} = 0. \quad (4.18)$$

4.4.1 The Adjoint Equation

As the Euler equations (4.4) and zero normal-velocity boundary condition (4.8) are satisfied, giving equations (4.16) and (4.18), the variation of \mathcal{L} becomes

$$\delta\mathcal{L} = \delta\mathcal{L}_Q + \delta\mathcal{L}_\theta + \delta\mathcal{L}_\alpha.$$

The variation $\delta\mathcal{L}_Q$ can be expressed as

$$\delta\mathcal{L}_Q = \int_{S_a} \left(\frac{\partial\psi}{\partial p} \frac{\partial p}{\partial \mathbf{Q}} \right) \cdot \mathbf{Q}' dS + \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}') d\Omega + \int_{S_a} \gamma(\rho\vec{V})' \cdot \vec{n} dS. \quad (4.19)$$

Introducing \vec{A} as the Jacobian of the flux vector $\vec{\mathbf{F}}$ with respect to the conservative flow variables \mathbf{Q} ,

$$\vec{A} = \frac{\partial \vec{\mathbf{F}}}{\partial \mathbf{Q}},$$

C as the Jacobian of the normal flux defined on the boundaries

$$C = \frac{\partial(\vec{\mathbf{F}} \cdot \vec{n})}{\partial \mathbf{Q}} = \vec{A} \cdot \vec{n},$$

and integrating the domain integral in equation(4.19) by parts gives

$$\begin{aligned} \delta\mathcal{L}_Q = & \int_{S_a} \left(\left(\frac{\partial\psi}{\partial p} \frac{\partial p}{\partial \mathbf{Q}} - C^T \boldsymbol{\lambda} \right) \cdot \mathbf{Q}' + \gamma(\rho\vec{V})' \cdot \vec{n} \right) dS \\ & - \int_{S_\infty} C^T \boldsymbol{\lambda} \cdot \mathbf{Q}' dS - \int_{\Omega} (\vec{A}^T \cdot \vec{\nabla} \boldsymbol{\lambda}) \cdot \mathbf{Q}' d\Omega. \end{aligned} \quad (4.20)$$

The adjoint equation and boundary condition follow from the condition

$$\delta\mathcal{L}_Q = 0.$$

This can be satisfied by setting the integrals equal to zero.

The adjoint equation is obtained by setting the domain integral equal to zero, which is

$$\vec{A}^T \cdot \vec{\nabla} \boldsymbol{\lambda} = 0 \quad \text{in } \Omega. \quad (4.21)$$

The surface integral over S_∞ can be made equal to zero as follows. The time-dependent Euler equations are considered in the quasi-linear form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \vec{A} \cdot \vec{\nabla} \mathbf{Q} = 0. \quad (4.22)$$

The characteristic wave propagation on S_∞ is determined by the eigenvalues of the matrix $C = \vec{A} \cdot \vec{n}$. Diagonalization of C provides

$$C = X \Lambda X^{-1}, \quad (4.23)$$

where the matrix X and its inverse, X^{-1} , consist of the right and left eigenvectors of C , respectively (cf. Appendix A), while Λ is a diagonal matrix consisting of the eigenvalues of C ,

$$\Lambda = \begin{bmatrix} \vec{V} \cdot \vec{n} & 0 & 0 & 0 \\ 0 & \vec{V} \cdot \vec{n} & 0 & 0 \\ 0 & 0 & \vec{V} \cdot \vec{n} + a & 0 \\ 0 & 0 & 0 & \vec{V} \cdot \vec{n} - a \end{bmatrix}. \quad (4.24)$$

The variation \mathbf{Q}' and the variation of the characteristic variables, \mathbf{W}' , are related by

$$\mathbf{Q}' = X \mathbf{W}'. \quad (4.25)$$

It is important to note that, on the boundary, *positive eigenvalues* correspond to *incoming* characteristics, for which the associated characteristic variables, \mathbf{W}_+ , are *fixed*. Accordingly, *negative eigenvalues* correspond to *outgoing* characteristics, where the associated characteristic variables, \mathbf{W}_- , may *vary* depending on the state inside the domain. Assuming that N is the dimension of \mathbf{W}_+ , for the flow problem there are N analytical boundary conditions fixing the values of \mathbf{W}_+ . Whereas, for obtaining \mathbf{W}_- , $(4 - N)$ additional equations must be formulated based upon the state inside the domain. The fact that \mathbf{W}_+ is fixed implies

$$\mathbf{W}'_+ = 0. \quad (4.26)$$

Taking this into account, one obtains

$$\mathbf{Q}' = X_- \mathbf{W}'_-, \quad (4.27)$$

where X_- (cf. Appendix A) is a matrix of dimension $(4 \times (4 - N))$ consisting of the elements of X that multiply \mathbf{W}_- in equation (4.25). The integral over S_∞ in equation (4.20) can now be expressed in terms of \mathbf{W}'_- as follows,

$$\int_{S_\infty} C^T \boldsymbol{\lambda} \cdot \mathbf{Q}' dS = \int_{S_\infty} (CX_-)^T \boldsymbol{\lambda} \cdot \mathbf{W}'_- dS. \quad (4.28)$$

This becomes zero if the following condition is satisfied,

$$(CX_-)^T \boldsymbol{\lambda} = 0. \quad (4.29)$$

Equation (4.29) takes the role of the boundary condition on S_∞ of the adjoint problem. Since the matrix product (CX_-) is of dimension $(4 \times (4 - N))$, the above condition provides only $(4 - N)$ equations, and N additional equations must be formulated (to be described in Section 4.7.2). This is a situation opposite to one that occurs in the problem of solving the flow equation itself where the boundary condition provides N equations.

Setting the surface integral over S_a equal to zero gives

$$\left(\frac{\partial \psi}{\partial p} \frac{\partial p}{\partial \mathbf{Q}} - C^T \boldsymbol{\lambda} \right) \cdot \mathbf{Q}' + \gamma ((\rho \vec{V})' \cdot \vec{n}) = 0.$$

This can be worked out to give

$$n_x \lambda_2 + n_y \lambda_3 = \frac{\partial \psi}{\partial p}, \quad (4.30)$$

and

$$\gamma = \lambda_1 + u \lambda_2 + v \lambda_3 + \left(\frac{u^2 + v^2}{2} + \frac{a^2}{\gamma - 1} \right) \lambda_4, \quad (4.31)$$

where a is the speed of sound. Equation (4.30) takes the role of the boundary condition on S_a for the adjoint problem. It can be observed that if λ_2 and λ_3 are interpreted as the Cartesian components of an adjoint velocity vector $\bar{\lambda}$, equation (4.30) can be interpreted as a transpiration boundary condition, i.e. as a complement of the zero normal-velocity boundary condition (4.8).

The adjoint problem can be summarized as follows. The adjoint equation (4.21) is solved for $\bar{\lambda}$, subject to the boundary conditions (4.30) on S_a and (4.29) on S_∞ . As $\bar{\lambda}$ is available, the Lagrange multiplier γ is determined from equation (4.31).

4.4.2 The Gradient Equation

After solving the Euler and the adjoint equations, providing the values of \mathbf{Q} , $\bar{\lambda}$ and γ , the variation of \mathcal{L} becomes

$$\delta \mathcal{L} = \delta \mathcal{L}_\theta + \delta \mathcal{L}_\alpha. \quad (4.32)$$

The variation $\delta \mathcal{L}_\theta$ is dealt with as follows. Since θ is a parameter that describes the shape of S_a , which is part of the flow domain boundary, the variation $\delta \theta$ implies also a variation of the flow domain Ω . As a result of this, and recognizing that

$$\begin{aligned} \mathbf{Q} &= \mathbf{Q}(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \bar{\lambda} &= \bar{\lambda}(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \gamma &= \gamma(\mathbf{x}), & \mathbf{x} \in S_a, \end{aligned}$$

the variation of Ω also implies a variation of \mathbf{Q} , $\bar{\lambda}$, and γ in the form of, respectively,

$$\delta \mathbf{Q}_\Omega = \frac{\partial \mathbf{Q}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right), \quad \mathbf{x} \in \Omega, \quad (4.33)$$

$$\delta \bar{\lambda}_\Omega = \frac{\partial \bar{\lambda}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right), \quad \mathbf{x} \in \Omega, \quad (4.34)$$

$$\delta \gamma_\Omega = \frac{\partial \gamma}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right), \quad \mathbf{x} \in S_a. \quad (4.35)$$

This leads to the introduction of the notion of the *deformation velocity* $\vec{\omega}$, defined as (cf. Appendix G)

$$\vec{\omega}(\mathbf{x}) = \vec{\chi} \cdot \delta \theta,$$

where

$$\vec{\chi} = \begin{pmatrix} \chi_x \\ \chi_y \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial \theta} \end{pmatrix}.$$

The Cartesian components of $\vec{\omega}$ can be expressed as

$$\begin{aligned} \omega_x &= \chi_x \cdot \delta \theta, \\ \omega_y &= \chi_y \cdot \delta \theta. \end{aligned}$$

Accordingly, the normal and tangential components of $\vec{\omega}$ on the boundary S_a can be written as

$$\begin{aligned} \omega_n &= \chi_n \cdot \delta \theta, \\ \omega_s &= \chi_s \cdot \delta \theta, \end{aligned}$$

where

$$\begin{pmatrix} \chi_n \\ \chi_s \end{pmatrix} = \begin{pmatrix} n_x & n_y \\ n_y & -n_x \end{pmatrix} \begin{pmatrix} \chi_x \\ \chi_y \end{pmatrix}.$$

Expressions (4.33)–(4.35) can now be written in the form

$$\begin{aligned} \frac{\partial \mathbf{Q}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right) &= \vec{\nabla} \mathbf{Q} \cdot \vec{\omega}, \quad \mathbf{x} \in \Omega, \\ \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right) &= \vec{\nabla} \boldsymbol{\lambda} \cdot \vec{\omega}, \quad \mathbf{x} \in \Omega, \\ \frac{\partial \Upsilon}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta \theta \right) &= \left(\frac{\partial \nu}{\partial s} \right) \omega_s, \quad \mathbf{x} \in S_a. \end{aligned}$$

They represent the so-called *convective variation* of \mathbf{Q} , $\boldsymbol{\lambda}$ and Υ , respectively. The term convective refers to the variation implied by the domain variation, where the domain variation is interpreted as a deformation with the speed $\vec{\omega}$. As a complement to the convective variation, the notion *local variation* can be introduced for the variations \mathbf{Q}' , $\boldsymbol{\lambda}'$, and Υ' . The local and convective variations constitute the *total variation* represented by a *material derivative*[†], defined as

$$\begin{aligned} \dot{\mathbf{Q}} &= \mathbf{Q}' + \vec{\nabla} \mathbf{Q} \cdot \vec{\omega}, \\ \dot{\boldsymbol{\lambda}} &= \boldsymbol{\lambda}' + \vec{\nabla} \boldsymbol{\lambda} \cdot \vec{\omega}, \\ \dot{\nu} &= \Upsilon' + \left(\frac{\partial \nu}{\partial s} \right) \omega_s, \end{aligned}$$

where the first and second terms in the right-hand sides are the local and convective variations, respectively.

[†]This is also commonly referred to as the substantial derivative.

The convective variations of \mathbf{Q} , λ and γ contribute to $\delta\mathcal{L}_\theta$ as $\delta\mathcal{L}_o$ defined as

$$\begin{aligned}\delta\mathcal{L}_o = & \int_{\Omega} (\vec{\nabla}\lambda \cdot \vec{\omega}) \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\Omega + \int_{S_a} \left(\frac{\partial \nu}{\partial s} \right) \omega_s (\rho \vec{V} \cdot \vec{n}) dS \\ & + \int_{S_a} \left(\frac{\partial \psi}{\partial p} \frac{\partial p}{\partial \mathbf{Q}} - C^T \lambda \right) \cdot (\vec{\nabla} \mathbf{Q} \cdot \vec{\omega}) dS + \int_{S_a} \gamma (\vec{\nabla}(\rho \vec{V}) \cdot \vec{\omega}) \cdot \vec{n} dS \\ & - \int_{S_\infty} C^T \lambda \cdot (\vec{\nabla} \mathbf{Q} \cdot \vec{\omega}) dS - \int_{\Omega} (\vec{A}^T \cdot \lambda) \cdot (\vec{\nabla} \mathbf{Q} \cdot \vec{\omega}) d\Omega.\end{aligned}$$

This has the same form as $\delta\mathcal{L}_\lambda + \delta\mathcal{L}_\gamma + \delta\mathcal{L}_Q$, implied by the local variations, with $\delta\mathcal{L}_\lambda$ given by equation (4.15), $\delta\mathcal{L}_\gamma$ by (4.17) and $\delta\mathcal{L}_Q$ by (4.20). Hence, $\delta\mathcal{L}_o$ also vanishes if $\delta\mathcal{L}_\lambda$, $\delta\mathcal{L}_\gamma$, and $\delta\mathcal{L}_Q$ are set equal to zero as in the preceding sections. It can be concluded that the solution of the flow and adjoint problems eliminate the contribution from the total variations of λ , γ , and \mathbf{Q} . The variation $\delta\mathcal{L}_\theta$ is thus solely due to the (total) geometric variation of the boundary S_a .

Two approaches can be distinguished for obtaining the gradients of \mathcal{F} with respect to θ and α . These are described as follows:

- (i) *Material derivative approach.* The variation of the relevant geometric properties are given as the material derivative formulae, equations (G.10)–(G.15), in Appendix G. The variation $\delta\mathcal{L}_\theta$ can be written as

$$\delta\mathcal{L}_\theta = \int_{S_a} \left(\frac{\partial \psi}{\partial \theta} \cdot \delta\theta + \gamma(\rho \vec{V} \cdot \vec{n}) \cdot \dot{\vec{n}} \right) dS + \int_{S_a} (\psi + \gamma(\rho \vec{V} \cdot \vec{n})) d\dot{S} + \int_{\Omega} \lambda \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\dot{\Omega}.$$

Substituting the Euler equations and zero normal-velocity boundary condition yields

$$\delta\mathcal{L}_\theta = \int_{S_a} \left(\frac{\partial \psi}{\partial \theta} \cdot \delta\theta + \gamma(\rho \vec{V} \cdot \vec{n}) \cdot \dot{\vec{n}} \right) dS + \int_{S_a} \psi d\dot{S}.$$

Substituting equations (G.10)–(G.15) gives

$$\begin{aligned}\delta\mathcal{L}_\theta &= \int_{S_a} \left(\frac{\partial \psi}{\partial \theta} \cdot \delta\theta + \psi(\omega_{s,s} + H\omega_n) - \gamma(\rho \vec{V} \cdot \vec{s})(\omega_{n,s} + H\omega_s) \right) dS \\ &= \int_{S_a} \left(\frac{\partial \psi}{\partial \theta} + \psi(\chi_{s,s} + H\chi_n) - \gamma(\rho \vec{V} \cdot \vec{s})(\chi_{n,s} + H\chi_s) \right) dS \cdot \delta\theta,\end{aligned}\tag{4.36}$$

where H is the surface curvature, and the subscripts s, s and n, s refer to the tangential derivative of the tangential and normal component, respectively, of ω and χ . The gradient of the general functional \mathcal{F} with respect to θ is given by

$$\frac{d\mathcal{F}}{d\theta} = \lim_{\delta\theta \rightarrow 0} \frac{\delta\mathcal{L}_\theta}{\delta\theta},$$

with $\left(\frac{\delta\mathcal{L}_\theta}{\delta\theta} \right)$ obtained directly from equation (4.36), such that

$$\frac{d\mathcal{F}}{d\theta} = \int_{S_a} \left(\frac{\partial \psi}{\partial \theta} + \psi(\chi_{s,s} + H\chi_n) - \gamma(\rho \vec{V} \cdot \vec{s})(\chi_{n,s} + H\chi_s) \right) dS. \tag{4.37}$$

The gradient of \mathcal{F} with respect to α can be obtained in a similar manner. Noting that

$$\delta \mathcal{L}_\alpha = \left(\int_{S_a} \frac{\partial \psi}{\partial \alpha} dS - \int_{S_\infty} (CX_+)^T \lambda \cdot \frac{\partial \mathbf{W}_+}{\partial \alpha} dS \right) \delta \alpha, \quad (4.38)$$

one obtains the gradient of \mathcal{F} with respect to α as follows,

$$\frac{d\mathcal{F}}{d\alpha} = \lim_{\delta\alpha \rightarrow 0} \frac{\delta \mathcal{L}_\alpha}{\delta \alpha} = \int_{S_a} \frac{\partial \psi}{\partial \alpha} dS - \int_{S_\infty} (CX_+)^T \lambda \cdot \frac{\partial \mathbf{W}_+}{\partial \alpha} dS. \quad (4.39)$$

- (ii) *Discrete approach.* In this approach, it is used that the discretization of certain expressions of \mathcal{F} (such as those for C_l and C_d) must satisfy the *static condition*. This means that, for a constant surface pressure distribution over a closed-contoured airfoil, \mathcal{F} is invariant (i.e., the airfoil does not experience a force). Provided the airfoil contour remains closed for any variation of θ , the static condition can be stated as

$$\left. \frac{\partial \mathcal{F}}{\partial \theta} \right|_{p=\text{constant}} = 0. \quad (4.40)$$

Functionals requiring the static condition usually take the form

$$\mathcal{F} = \int_{S_a} p(n_y \psi_1(\alpha) + n_x \psi_2(\alpha)) dS,$$

which in 2-D can be worked out to give

$$\mathcal{F} = \psi_1(\alpha) \int_{S_a} p dx - \psi_2(\alpha) \int_{S_a} p dy.$$

The discretized form of \mathcal{F} , denoted by $\hat{\mathcal{F}}$, can be written as

$$\hat{\mathcal{F}} = \psi_1(\alpha) \sum_{i=1}^{n-1} \frac{1}{2} (p_{i+1} + p_i) (x_{i+1} - x_i) - \psi_2(\alpha) \sum_{i=1}^{n-1} \frac{1}{2} (p_{i+1} + p_i) (y_{i+1} - y_i),$$

where

$$x_i = x_i(\theta), \quad y_i = y_i(\theta), \quad i = 1, \dots, n,$$

n is the number of grid points, while $i = 1$ and $n - 1$ corresponds with the lower and upper trailing edge, respectively. The partial derivative of $\hat{\mathcal{F}}$ with respect to θ is obtained by direct analytical differentiation of the above discrete expression. It can easily be verified that the static condition (4.40) is always satisfied by $\hat{\mathcal{F}}$ as long as the airfoil contour remains closed. In contrast, in the alternative approach with the material derivative formulae, the numerical evaluation of the partial derivatives of \mathcal{F} with respect to θ do not, in general, imply the static condition. This is partly due to the truncation error in the approximation of the tangential derivative and the surface curvature occurring in the material derivative formulae.

Hence, in the discrete approach, after solving the flow and adjoint problems, providing the values of \mathbf{Q} , λ and Υ , the surface integration over S_a of the Lagrangian (4.14),

$$\mathcal{L}_a = \int_{S_a} (\psi + \Upsilon(\rho \vec{V} \cdot \vec{n})) dS,$$

is discretized into

$$\hat{\mathcal{L}}_a = \hat{\mathcal{F}} - \sum_{i=1}^{n-1} \frac{1}{2} (I_{i+1} + I_i)(y_{i+1} - y_i) + \sum_{i=1}^{n-1} \frac{1}{2} (J_{i+1} + J_i)(x_{i+1} - x_i),$$

where

$$I = \Upsilon(\rho u), \quad J = \Upsilon(\rho v).$$

The gradient of \mathcal{F} with respect to θ is obtained as

$$\frac{d\mathcal{F}}{d\theta} = \frac{\partial \hat{\mathcal{L}}_a}{\partial \theta}. \quad (4.41)$$

The gradient with respect to α is given by

$$\frac{d\mathcal{F}}{d\alpha} = \frac{\partial \hat{\mathcal{L}}_a}{\partial \alpha} - \int_{S_\infty} (CX_+)^T \lambda \cdot \frac{\partial \mathbf{W}_+}{\partial \alpha} dS. \quad (4.42)$$

The choice between the material derivative and discrete approach will be made based on numerical experiments, which are to be described in Section 4.7.4.

A fixed (design) lift coefficient can be considered as an equality constraint, which means that

$$\left(\frac{dC_l}{d\alpha} \right) \delta\alpha + \left(\frac{dC_l}{d\theta} \right) \cdot \delta\theta = 0.$$

This implies that one variable out of the set of the components of θ and α can be chosen to be dependent. A convenient choice is to take α as the dependent variable, giving

$$\delta\alpha = - \left(\frac{dC_l}{d\alpha} \right)^{-1} \left(\frac{dC_l}{d\theta} \right) \cdot \delta\theta. \quad (4.43)$$

The variation of the Lagrangian can be expressed as

$$\begin{aligned} \delta\mathcal{L} &= \delta\mathcal{L}_\theta + \delta\mathcal{L}_\alpha \\ &= \left(\frac{d\mathcal{F}}{d\theta} \right) \cdot \delta\theta + \left(\frac{d\mathcal{F}}{d\alpha} \right) \delta\alpha. \end{aligned}$$

Substitution of equation (4.43) gives

$$\delta\mathcal{L} = \left(\frac{d\mathcal{F}}{d\theta} \right) \cdot \delta\theta - \left(\frac{d\mathcal{F}}{d\alpha} \right) \left(\frac{dC_l}{d\alpha} \right)^{-1} \left(\frac{dC_l}{d\theta} \right) \cdot \delta\theta.$$

Hence, for a fixed C_l the gradient of \mathcal{F} with respect to the design variables θ can be obtained as

$$\left(\frac{d\mathcal{F}}{d\theta}\right)_{C_l} = \frac{d\mathcal{F}}{d\theta} - \left(\frac{d\mathcal{F}}{d\alpha}\right) \left(\frac{dC_l}{d\alpha}\right)^{-1} \left(\frac{dC_l}{d\theta}\right), \quad (4.44)$$

with $\left(\frac{d\mathcal{F}}{d\theta}\right)$ given by equation (4.37), or (4.41), and $\left(\frac{d\mathcal{F}}{d\alpha}\right)$ by equation (4.39), or (4.42). The expressions for $\left(\frac{dC_l}{d\alpha}\right)$ and $\left(\frac{dC_l}{d\theta}\right)$ are obtained by substituting C_l into \mathcal{F} , where

$$C_l = \int_{S_a} 2(p - p_\infty)(n_x \sin \alpha - n_y \cos \alpha) dS,$$

giving the function ψ as

$$\psi = 2(p - p_\infty)(n_x \sin \alpha - n_y \cos \alpha).$$

4.4.3 Remarks related to Jameson's approach for formulating the adjoint equations and boundary conditions

This section discusses the principal differences between the approach used by Jameson (Refs. [28], [29], [47], [32], [30], [31]) and the approach described in the preceding section.

In Jameson's approach, the adjoint equations and boundary conditions are formulated on the basis of the flow equations which are transformed into a computational domain. This implies that the variational analysis is done on a fixed domain. Because the domain is fixed, a boundary movement analysis (such as that described in Appendix G) is not needed in obtaining the expression for the gradient with respect to the geometric design variables (i.e. θ). The gradient expression consists of contour and domain integrals taken along the boundary and over the volume of the computational domain, respectively. In these integrals the variation of the design variables (i.e. $\delta\theta$) is involved through the variation of the Jacobian of the coordinate transformation.

In Jameson's approach, the design problem is mapped to a fixed domain on which the flow and adjoint solutions are computed with a finite mesh. The contribution from the mesh variations, both on the airfoil surface and in the flow field, is taken into account in the computation of the gradient. This allows an accurate definition of the gradient and implies a consistent treatment of the design problem throughout the optimization process, provided that the sensitivity of the transformation Jacobian (i.e. the metrics) with respect to the design variables can be determined accurately. Unfortunately, computing the sensitivity of the transformation Jacobian is not easy and can be quite costly, unless a simple analytic or algebraic grid generation procedure with an explicit relationship between

the mesh and the design variables is employed. Therefore, in general, one cannot take advantage of existing grid generation procedures, especially those based on partial differential equations. Furthermore, the expression for the gradient is not readily suitable for unstructured mesh. Indeed, this is the main reason that Jameson's approach has not been adopted in this thesis; the author of this thesis had at his disposal a flow solver with a grid generation and a grid adaption procedure based on partial differential equations.

In the present approach, the adjoint equations and boundary conditions are formulated based on the flow equations described in the physical domain. This leads to a general expression for the gradient, which is independent of the type of discretization, but requires a boundary movement analysis. The main disadvantage of the present approach is probably that the mesh variations are not taken into account in the computation of the gradient. In other words, one must accept the assumption that the discrete flow and adjoint solutions are independent of the mesh, which is only true in the case of infinitely fine mesh.

4.5 Shape Parameterization and Deformation Velocity

The shape parameterization gives the relation between the airfoil coordinates $\mathbf{x} \mid \mathbf{x} \in S_a$ and the design variables $\boldsymbol{\theta}$. This also implies the specification of the deformation velocity.

A shape parameterization should satisfy the following requirements,

- (i) In principle there should be no restriction on the possible number of design variables. This is desirable in order not to restrict the design space to a certain family of airfoil shapes.
- (ii) The surface curvature must be continuous to ensure smoothness of the airfoil surface. This is desirable so as to avoid numerical irregularities that could be implied by surface discontinuities.
- (iii) The design variables should preferably have (in an approximate sense) a linear relationship with the surface curvature. This would allow scaling of the design variables (to be described in Section 4.6). Also, this is expected to have an effect of reducing the non-linearity of the optimization problem, considering that, for local subsonic flow, a local variation of the surface curvature is proportional (in an approximate sense) with the local variation of pressure (Refs.[9], [54]), while the aerodynamic functionals of interest are defined in terms of pressure. This consideration can be expected also to be relevant for transonic conditions because the flow is locally subsonic over a large part of the airfoil surface.

There are a number of parameterization schemes known in the literature (e.g., Ref. [59]) which satisfy the first two requirements. However, since it is not easy to examine whether these schemes also satisfy the third requirement, it was preferred

to develop a custom parameterization scheme. This is described in the following paragraphs.

A parametric curve with the coordinates $(\tau, \hat{y}(\tau))$ is introduced, where

$$0 \leq \tau \leq 1.$$

This has a one-to-one correspondence with the airfoil arc length measured clockwise from the lower trailing edge ($\tau = 0$) to the upper trailing edge ($\tau = 1$). The leading edge corresponds to $\tau = \tau_{LE} = 0.5$. The coordinates (x, y) of an airfoil are defined by

$$x(\tau) = (2\tau - 1)^2, \quad 0 \leq \tau \leq 1, \quad (4.45)$$

$$y(\tau) = \begin{cases} \int_0^\tau \tilde{y}_l d\tau - \frac{\tau}{\tau_{LE}} \epsilon_l, & 0 \leq \tau \leq \tau_{LE}, \\ \int_{\tau_{LE}}^\tau \tilde{y}_u d\tau - \frac{\tau - \tau_{LE}}{1 - \tau_{LE}} \epsilon_u, & \tau_{LE} \leq \tau \leq 1, \end{cases} \quad (4.46)$$

where

$$\begin{aligned} \epsilon_l &= \int_0^{\tau_{LE}} \tilde{y}_l d\tau, \\ \epsilon_u &= \int_{\tau_{LE}}^1 \tilde{y}_u d\tau, \\ \tilde{y}_l &= \tau \tilde{y}_o + \int_0^\tau \hat{y}'' d\tau, \\ \tilde{y}_u &= \tau \tilde{y}_o + \int_{\tau_{LE}}^\tau \hat{y}'' d\tau. \end{aligned}$$

where the accent denotes the derivative with respect to τ . The terms with ϵ_l and ϵ_u ensure that the trailing and leading edges are fixed at $(0, 0)$ and $(1, 0)$, respectively. The parameter \tilde{y}_o is introduced as a means to satisfy the continuity of the curvature at the leading edge. This is given as

$$\tilde{y}_o = 2 \left[\frac{1}{\tau_{LE}} \int_0^{\tau_{LE}} \left(\int_0^\tau \hat{y}'' d\tau \right) d\tau - \int_0^{\tau_{LE}} \hat{y}'' d\tau - \frac{1}{1 - \tau_{LE}} \int_{\tau_{LE}}^1 \left(\int_{\tau_{LE}}^\tau \hat{y}'' d\tau \right) d\tau \right].$$

The function $\hat{y}'' = \hat{y}''(\tau)$, which is the second derivative of \hat{y} with respect to τ , takes the role as the design variables θ . Because $\hat{y}''(\tau)$ is continuously distributed over $0 \leq \tau \leq 1$, the parameterization scheme represents, in principle, an infinite dimensional design space, i.e. the vector θ consists of the components $\theta(\tau) = \hat{y}''(\tau)$.

The surface curvature can be expressed as

$$H = \frac{x''y' - x'y''}{(x' + y')^{3/2}}. \quad (4.47)$$

The variation δH due to the variation $\delta \hat{y}''$ can be evaluated as follows. Equation (4.45) implies that x' and x'' are invariant with respect to \hat{y}'' , i.e.

$$\delta x' = \delta x'' = 0.$$

The Taylor series expansions of y'' and y' around $\tau = \tau_o$ give

$$\begin{aligned}y''(\tau) &= y''(\tau_o) + y'''(\tau_o)\Delta\tau + \dots \\y'(\tau) &= y'(\tau_o) + y''(\tau_o)\Delta\tau + \dots\end{aligned}$$

Since the analysis concerns only the local variation of curvature at $\tau = \tau_o + \Delta\tau$, the parameters ϵ_l , ϵ_u , and \tilde{y}_o can be assumed to be invariant with \hat{y}'' . Then, equation (4.46) implies

$$\begin{aligned}\delta y''(\tau) &= \delta \hat{y}''(\tau_o), \\ \delta y'(\tau) &= \delta \hat{y}''(\tau_o)\Delta\tau.\end{aligned}$$

As $\tau \rightarrow \tau_o$, one has $\delta y'(\tau) \rightarrow 0$, and accordingly the variation of the surface curvature can be written as

$$\delta H \approx c \delta \hat{y}'',$$

which implies

$$H \approx c \theta(\tau), \quad (4.48)$$

where

$$c = -\frac{x'}{(x' + y')^{3/2}}.$$

Equation (4.48) shows an approximate linear dependency between the curvature and the design variables, thus satisfying the third requirement stated above.

For computer implementation, the infinite dimensional design space described by equation (4.46) must inevitably be reduced to a finite one. This is done by representing \hat{y}'' with N piecewise linear functions of τ , with continuities in the values of \hat{y}'' (C_0 -continuous). The airfoil shape itself is then C_2 -continuous. Thus, an N -dimensional design space is constructed by an N -dimensional vector of design variables θ as follows,

$$\theta = (\theta_1 \quad \dots \quad \theta_N)^\top = (\hat{y}''(\tau_1) \quad \dots \quad \hat{y}''(\tau_N))^\top, \quad (4.49)$$

with $\hat{y}''(\tau)$ varying linearly in a segment bounded by τ_k and τ_{k+1} , $k = 1, \dots, N-1$. Equation (4.46) is evaluated on this basis and the resulting expression is differentiated analytically with respect to θ . The deformation velocity at a grid point i due to a component θ_k can then be obtained as

$$\begin{aligned}\omega_y(i)_k &= \chi_y(i)_k \delta \theta_k \\ &= \left(\frac{\partial y(\tau_i)}{\partial \theta_k} \right) \delta \theta_k, \quad k = 1, \dots, N\end{aligned}$$

It should be noted that τ_i refers to the value that corresponds to a surface grid point, while τ_k refers to the value for which \hat{y}'' is specified on the parametric curve.

The validity of equation (4.48) can be examined by comparing the exact partial derivative given by

$$H(\tau_k)_k = \frac{\partial H(\tau_k)}{\partial \theta_k},$$

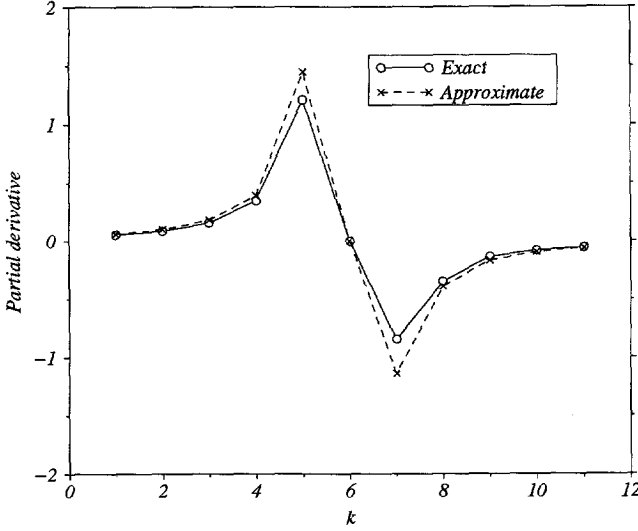


Figure 4.2: Comparison of the exact and approximate partial derivatives of the curvature.

with the approximate expression derived from equation (4.48)

$$\tilde{H}(\tau_k)_k = -\frac{x'(\tau_k)}{(x'(\tau_k) + y'(\tau_k))^{3/2}}.$$

Figure 4.2 shows that these two expressions agree reasonably well for the shape-fitting of the RAE 2822 airfoil.

It should be noted that the airfoil arc length s increases monotonously with τ . This implies that if $\hat{y}''(\tau)$ is monotonous with τ then, from equation (4.48), the curvature is also monotonous with τ and accordingly with s . Because of the linear representation of $\hat{y}''(\tau)$, this also means that the curvature is monotone on each segment between τ_k and τ_{k+1} , $k = 1, \dots, N-1$. Hence, the number of design variables, N , determines the wave-length of "wiggles" in the airfoil surface.

The shape parameterization has been tested by fitting the airfoils RAE 2822, NACA 4418, and Liebeck LA203A. The coordinates of these airfoils are given in a tabular form of 101 points. The shape-fitting is performed in the least-square sense with increasing numbers of design variables:

$$N = 11, 21, 31, 41, 51, 61$$

Figure 4.3 shows the L_2 -norm taken over the y -values, i.e.

$$L_2 = \left[\frac{\sum_{i=1}^N (y - y_{fit})_i^2}{N} \right]^{\frac{1}{2}}.$$

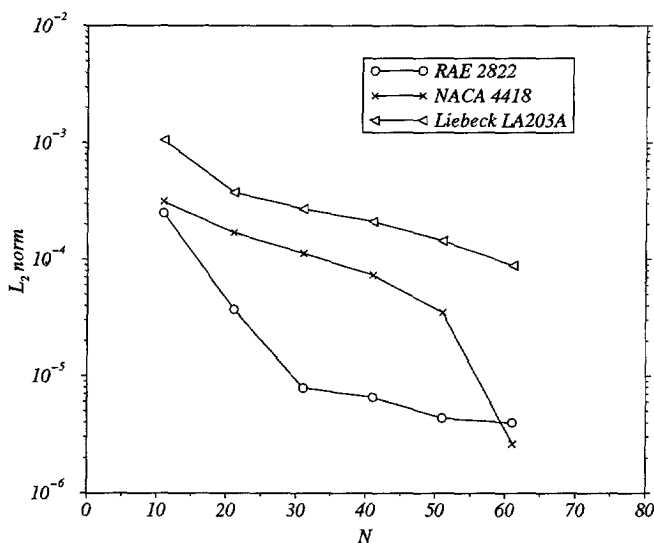


Figure 4.3: L_2 -norms of the shape-fitting.

As can be seen, 11 to 21 variables are usually sufficient for obtaining a reasonable fit with an L_2 -norm of about 10^{-4} to 10^{-3} .

4.6 Aerodynamic Scaling of the Design Variables

The success of an optimization depends to a large extent on whether the problem formulation is well-scaled or not. Vanderplaats [63] presented a clear illustration of the advantage of a well-scaled optimization problem (in terms of the number of objective function evaluations), and suggested a method for determining the scaling factors.

In this method the scaling factors, comprised in a vector \mathbf{d} , consist of the diagonal elements of the Hessian matrix \mathcal{H}^\dagger . The vector of scaled design variables $\tilde{\boldsymbol{\theta}}$ is defined as

$$\tilde{\boldsymbol{\theta}} = \mathbf{d}^\top \boldsymbol{\theta}.$$

The optimization proceeds in the scaled design space. The function and gradients of \mathcal{P} in the scaled design space are determined as follows,

$$\mathcal{P}(\tilde{\boldsymbol{\theta}}) = \mathcal{P}(\boldsymbol{\theta}),$$

[†]The Hessian matrix consists of the second partial derivatives of the objective/constraint functional with respect to the design variables.

$$\frac{d\mathcal{P}}{d\boldsymbol{\theta}} = \mathbf{d}^{-1} \cdot \left(\frac{d\mathcal{P}}{d\boldsymbol{\theta}} \right),$$

where \mathbf{d}^{-1} is a vector the components of which are the reciprocals of the corresponding components of \mathbf{d} .

In order to study the suitability of scaling for aerodynamic optimization, a model aerodynamic functional is considered of the form[†]

$$\mathcal{P} = \frac{1}{2} \int_S (C_p - C_{p,t})^2 dS. \quad (4.50)$$

The procedure of scaling requires the computation of the Hessian matrix of \mathcal{P} . This can be approximated by making use of an "inverse aerodynamic formula" that relates changes in C_p to changes in the curvature H (Ref. [54]) in the form

$$\delta H \approx \delta C_p (1 + H^2)^{0.25},$$

This is rearranged to give

$$\delta C_p \approx \delta H (1 + H^2)^{-0.25}.$$

It is assumed that the term $(1 + H^2)$ is invariant with $\boldsymbol{\theta}$. The variation of \mathcal{P} with respect to the variation of $\theta_k \in \boldsymbol{\theta}$ can be obtained as follows,

$$\begin{aligned} \delta \mathcal{P} &= \int_S (C_p - C_{p,t}) \delta C_p dS \\ &= \int_S (1 + H^2)^{-0.25} (C_p - C_{p,t}) \delta H dS \\ &= \int_S (1 + H^2)^{-0.25} (C_p - C_{p,t}) \left(\frac{\partial H}{\partial \theta_k} \right) \delta \theta_k dS, \end{aligned}$$

which gives the partial derivative

$$\frac{\partial \mathcal{P}}{\partial \theta_k} = \int_S (1 + H^2)^{-0.25} (C_p - C_{p,t}) \left(\frac{\partial H}{\partial \theta_k} \right) dS.$$

In case of a linear relationship between H and $\boldsymbol{\theta}$, one obtains

$$\begin{aligned} \delta \left(\frac{\partial \mathcal{P}}{\partial \theta_k} \right) &= \int_S (1 + H^2)^{-0.25} \left(\frac{\partial H}{\partial \theta_k} \right) \delta C_p dS \\ &= \int_S (1 + H^2)^{-0.5} \left(\frac{\partial H}{\partial \theta_k} \right)^2 \delta \theta_k dS, \end{aligned}$$

The k -th diagonal element of the Hessian matrix is defined as

$$\mathcal{H}_{kk} = \frac{\partial^2 \mathcal{P}}{\partial \theta_k^2},$$

[†]This represents a functional used to describe inverse problems (cf. Section 4.9).

which can be approximated as

$$\mathcal{H}_{kk} = \int_S \frac{1}{\sqrt{1+H^2}} \left(\frac{\partial H}{\partial \theta_k} \right)^2 dS,$$

The scaling factors can finally be determined as

$$\mathbf{d} = \left(\frac{1}{\sqrt{\mathcal{H}_{11}}} \quad \cdots \quad \frac{1}{\sqrt{\mathcal{H}_{NN}}} \right)^\top. \quad (4.51)$$

The possibility of applying the design variable scaling in this manner emphasizes the advantage of using a shape parameterization scheme that satisfies such linear condition as equation (4.48).

4.7 Numerical Implementation

Since the adjoint equation exhibits a strong similarity with the Euler equations, the numerical implementation should also be similar to the largest possible extent. This is directly related with the choice of the flow solver that will be used in the design procedure. The investigation carried out in this chapter makes use of the inviscid (Euler) mode of HI-TASK code of the National Aerospace Laboratory NLR (Refs. [6] and [21]). The discretization of the adjoint equation and the computer implementation have been chosen such that maximum advantage is taken of the existing features of HI-TASK. The philosophy is to treat the flow solver as a black box.

4.7.1 Discretization of the Adjoint Equations

HI-TASK, which stands for **H**ighly-**I**ntegrated **T**urbulent **A**irflow **S**imulation **K**ernel, is basically a 2-D flow solver for single-element airfoil applications based on the Reynolds-averaged Navier-Stokes (RANS) equations. The Euler mode is obtained by dropping the viscous fluxes. The discretization is based on a cell-vertex finite volume scheme equivalent with a central difference scheme. For each grid point, a primary cell consisting of 9 vertices shown in Figure 4.4 is used as the control volume. The discretized time-dependent Euler equations takes the form

$$\Delta\Omega_{i,j} \frac{d\mathbf{Q}}{dt} + \int_{S_{i,j}} \vec{\mathbf{F}} \cdot \vec{\mathbf{n}} dS + \mathbf{D}_{i,j} = 0, \quad (4.52)$$

where $\Delta\Omega_{i,j}$ is the volume of the primary cell associated with a grid point i, j . \mathbf{D} consists of a 4-th and 2-nd order artificial dissipation term:

$$\mathbf{d} = \frac{\partial}{\partial \xi} \left(\epsilon_2, \frac{\partial \mathbf{Q}}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\epsilon_2, \frac{\partial \mathbf{Q}}{\partial \eta} \right) + \frac{\partial}{\partial \xi} \left(\epsilon_4, \frac{\partial^3 \mathbf{Q}}{\partial \xi^3} \right) + \frac{\partial}{\partial \eta} \left(\epsilon_4, \frac{\partial^3 \mathbf{Q}}{\partial \eta^3} \right). \quad (4.53)$$

where ϵ_2 and ϵ_4 are adaptable parameters depending on the local flow condition, with the subscripts i and j indicating the direction parallel to ξ and η , respectively. ξ and η refer to the coordinates in the computational domain. The 4-th

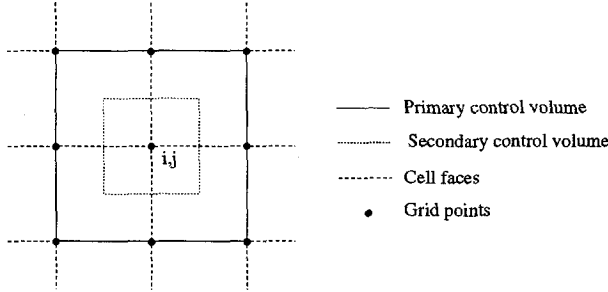


Figure 4.4: Finite Volume Stencil

order term is required for the stability of the central-difference scheme. The 2-nd order term is added for capturing shock waves, formulated based on the work of Jameson et al. [33].

Equation (4.52) is a system of ordinary differential equations which is integrated in time towards the steady-state in order to obtain the flow variables \mathbf{Q} . The time integration employs a five-stage Runge-Kutta scheme with a V -cycle multigrid procedure. The spatial discretization is on the basis of a C-type grid. In HI-TASK, a grid adaption module is integrated (Ref. [21]) and can optionally be activated.

In order to take advantage of the numerical schemes of the flow solver, the adjoint equation to be considered should also be time-dependent. Referring back to the situation on the domain boundary reflected by equation (4.29), it is noted that the number of conditions that can be provided analytically for the adjoint problem is complementary with that for the flow problem itself. This implies that the characteristic propagations of the adjoint problem are opposite to that of the flow problem. This means that the sign of the eigenvalues of the matrix coefficients of the flow problem (\tilde{A}) and those of the adjoint problem (\tilde{A}^T) should be opposite. Hence, the time-dependent adjoint equation should be formulated as

$$\frac{\partial \lambda}{\partial t} - A_x^T \frac{\partial \lambda}{\partial x} - A_y^T \frac{\partial \lambda}{\partial y} = 0. \quad (4.54)$$

which, subject to appropriate boundary condition, is to be integrated in time towards the steady-state to obtain the adjoint variables λ .

Equation (4.54) is not in divergence form. In order to adopt the finite volume scheme employed by HI-TASK for solving the adjoint equation, a divergence form is needed. For this purpose, the Jacobians A_x and A_y are taken constant within a control volume associated with a grid point i, j where the flow variables are defined. This approximation requires that the control volumes do not overlap so as to avoid multiple definitions of A_x and A_y . Because the primary control volumes overlap, the secondary control volume shown in Figure 4.4 is used for discretizing the adjoint equations.

The divergence form for a secondary control volume can be obtained as follows,

$$-A_x^\top \frac{\partial \lambda}{\partial x} - A_y^\top \frac{\partial \lambda}{\partial y} = -\frac{\partial A_x^\top \lambda}{\partial x} - \frac{\partial A_y^\top \lambda}{\partial y} = -\vec{\nabla} \cdot \vec{A}^\top \lambda.$$

This implies that one may introduce an adjoint flux \vec{Z} , with its Cartesian components defined as

$$\begin{aligned} Z_x &= -A_x^\top \lambda, \\ Z_y &= -A_y^\top \lambda. \end{aligned}$$

The adjoint flux formulas are given in Appendix E. The adjoint equation (4.54) is discretized into

$$\Delta \Omega_{i,j} \frac{d\lambda}{dt} + \int_{S_{i,j}} \vec{Z} \cdot \vec{n} dS + \widetilde{D}_{i,j} = 0 \quad (4.55)$$

The term $\widetilde{D}_{i,j}$ also consists of a 4-th and a 2-nd order artificial dissipations. These are formulated in the same way as in the flow solver, i.e. λ replace Q in equation (4.53):

$$d = \frac{\partial}{\partial \xi} \left(\epsilon_2 \frac{\partial \lambda}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\epsilon_2 \frac{\partial \lambda}{\partial \eta} \right) + \frac{\partial}{\partial \xi} \left(\epsilon_4 \frac{\partial^3 \lambda}{\partial \xi^3} \right) + \frac{\partial}{\partial \eta} \left(\epsilon_4 \frac{\partial^3 \lambda}{\partial \eta^3} \right).$$

The parameters ϵ_2 and ϵ_4 are identical with those used in the flow solver. Equation (4.55) is integrated in time towards the steady state with the same scheme as employed by the flow solver.

4.7.2 Implementation of the Adjoint Boundary Conditions

The values of λ on the far-field boundary S_∞ are obtained as follows. The boundary condition (4.29) provides $(4 - N)$ equations, where N is the number of outgoing characteristics of the adjoint problem. N additional equations are required in order to determine λ . These are formulated based on the characteristic relations of equation (4.54). The nature of this equation is determined by the eigenvalues of a matrix \tilde{C} defined as

$$\tilde{C} = -\vec{A}^\top \cdot \vec{n}.$$

Since $\tilde{C} = -C^\top$, it has real eigenvalues which are of the same magnitude, but of the opposite sign, as the eigenvalues of the matrix C defined by equation (4.23). One may then write

$$\tilde{C} = X^{-\top} \tilde{A} X^\top,$$

where $\tilde{A} = -A$. The characteristic variables of the adjoint equation are identified as

$$R = X^\top \lambda. \quad (4.56)$$

One should notice that the above equation is not in the variational form because \tilde{C} is not a function of λ and the flow variables are assumed constant along the

normal direction of S_∞ . Now, the components of \mathbf{R} are distinguished into the incoming ones,

$$\mathbf{R}_+ = X_-^\top \boldsymbol{\lambda},$$

and the outgoing ones,

$$\mathbf{R}_- = X_+^\top \boldsymbol{\lambda}. \quad (4.57)$$

According to the theory of Kreiss [35], the incoming characteristics \mathbf{R}_+ which is of dimension $(4 - N)$ imply that $(4 - N)$ conditions may be analytically specified. These are provided in the form of $(4 - N)$ equations given by condition (4.29). The remaining equations are obtained by extrapolating (towards the boundary) the values of \mathbf{R}_- from inside the domain. The values of $\boldsymbol{\lambda}$ on S_∞ can then be determined by solving the following equations,

- For *subsonic inflow*, condition (4.29) provides one equation:

$$(CX_-)_{i1} \lambda_i = 0, \quad (4.58)$$

while the extrapolation of the outgoing characteristic variables,

$$\mathbf{R}_- = \begin{pmatrix} R_1 \\ R_2 \\ R_4 \end{pmatrix},$$

provides the additional three equations:

$$X_{i1} \lambda_i = R_1, \quad (4.59)$$

$$X_{i2} \lambda_i = R_2, \quad (4.60)$$

$$X_{i4} \lambda_i = R_4. \quad (4.61)$$

In the above equations summations are taken over $i = 1, \dots, 4$.

- For *subsonic outflow*, condition (4.29) provides three equations:

$$(CX_-)_{i1} \lambda_i = 0, \quad (4.62)$$

$$(CX_-)_{i2} \lambda_i = 0, \quad (4.63)$$

$$(CX_-)_{i3} \lambda_i = 0, \quad (4.64)$$

while the extrapolation of the outgoing characteristic variable,

$$R_- = R_4,$$

provides the additional equation

$$X_{i4} \lambda_i = R_4. \quad (4.65)$$

Also, in the above expressions summations are taken over $i = 1, \dots, 4$.

The related formulae are given in Appendix B.

The boundary condition (4.30) on the airfoil surface S_a is implemented together with the characteristic relations that correspond to the non-positive eigenvalues of the matrix $\vec{A} \cdot \vec{n}$ on S_a . This leads to a linear system of equations to be solved for λ on S_a ,

$$X_{i1} \lambda_i = R_1, \quad (4.66)$$

$$X_{i2} \lambda_i = R_2, \quad (4.67)$$

$$X_{i3} \lambda_i = R_3, \quad (4.68)$$

$$n_x \lambda_2 + n_y \lambda_3 = \frac{\partial \psi}{\partial p}, \quad (4.69)$$

where $i = 1 \dots 4$. The characteristic variables R_1 , R_2 , and R_3 are obtained by extrapolation from inside the domain towards the boundary S_a .

4.7.3 Treatment (of the Correction) of Flow Far-Field Boundary Conditions

In order to gain accuracy in the finite-flow domain, HI-TASK employs a correction for the circulation around the airfoil in the far-field boundary condition of the Euler equations. The correction scheme is formulated in terms of the characteristic variables, and affects only the incoming characteristic variables containing the velocity components and the speed of sound.

The Riemann invariants of the one-dimensional Euler equations across S_∞ are given in equations (4.9)–(4.12). For *subsonic inflow*, the corrected Riemann invariants, \mathbf{W}_+^* , are identified as

$$\mathbf{W}_+^* = \begin{pmatrix} W_2^* \\ W_4^* \end{pmatrix} = \begin{pmatrix} \vec{V}_\infty^* \cdot \vec{s} \\ \vec{V}_\infty^* \cdot \vec{n} - \frac{2a_\infty^*}{\gamma - 1} \end{pmatrix}.$$

The superscript '*' indicates the corrected values. For *subsonic outflow*, there is only one incoming Riemann invariant, which is W_4 , such that in this situation:

$$\mathbf{W}_+^* = W_4^* = \vec{V}_\infty^* \cdot \vec{n} - \frac{2a_\infty^*}{\gamma - 1}.$$

The correction is based on the circulation produced by a vortex located at the quarter chord point of the airfoil. This is formulated in terms of the lift coefficient, C_l . The variation of \mathbf{W}_+^* due to the variation of C_l can be expressed as

$$\mathbf{W}_+^{*'} = \left(\frac{\partial \mathbf{W}_+^*}{\partial C_l} \right) \delta C_l. \quad (4.70)$$

The correction formulas and the related partial derivatives are described in Appendix C. Equation (4.70) replaces equation (4.26). Hence, equation (4.27) must be modified into

$$\mathbf{Q}' = (X_- \mathbf{W}_-') + (X_+^* \mathbf{W}_+^{*'}), \quad (4.71)$$

where X_+^* is a matrix consisting of the elements of X that multiply \mathbf{W}_+^* in equation (4.25) (cf. Appendix A). Consequently, the integral over S_∞ in equation (4.19) becomes

$$\int_{S_\infty} C^\top \boldsymbol{\lambda} \cdot \mathbf{Q}' dS = \int_{S_\infty} \left((CX_-)^\top \boldsymbol{\lambda} \cdot \mathbf{W}'_- + (CX_+^*)^\top \boldsymbol{\lambda} \cdot \mathbf{W}'_+ \right) dS, \quad (4.72)$$

The variation δC_l can be expressed as

$$\delta C_l = \int_{S_a} 2(n_x \sin \alpha - n_y \cos \alpha) \left(\frac{\partial p}{\partial \mathbf{Q}} \right) \cdot \mathbf{Q}' dS. \quad (4.73)$$

Substituting this into equation (4.70) and, subsequently, into (4.72) gives

$$\begin{aligned} \int_{S_\infty} C^\top \boldsymbol{\lambda} \cdot \mathbf{Q}' dS &= \int_{S_\infty} (CX_-)^\top \boldsymbol{\lambda} \cdot \mathbf{W}'_- dS \\ &\quad + C_\infty \int_{S_a} (n_x \sin \alpha - n_y \cos \alpha) \left(\frac{\partial p}{\partial \mathbf{Q}} \right) \cdot \mathbf{Q}' dS, \end{aligned} \quad (4.74)$$

where

$$C_\infty = 2 \int_{S_\infty} (CX_+^*)^\top \boldsymbol{\lambda} \cdot \left(\frac{\partial \mathbf{W}_+^*}{\partial C_l} \right) dS.$$

It can be verified that incorporating expression (4.74) into the variation $\delta \mathcal{L}_Q$ (4.20) has the consequence that the boundary condition (4.30) on S_a has to be corrected into

$$n_x \lambda_2 + n_y \lambda_3 = \frac{\partial \psi}{\partial p} - C_\infty (n_x \sin \alpha - n_y \cos \alpha). \quad (4.75)$$

The correction procedure given above can be interpreted as follows. For the flow problem, the boundary condition on S_∞ is corrected based on the (flow) state on S_a , while on the other hand, for the adjoint problem, the boundary condition on S_a is corrected based on the (adjoint) state on S_∞ .

4.7.4 Verification of the Gradient Equations

The suitability of the two approaches (cf. Section 4.4.2) for determining the gradients has been assessed as follows. Because no data is available which can be used as a reference, gradient computations for different types of objective functions have also been performed by central differencing flow solver results for perturbed geometries, and these have been compared with the gradient computed by the variational method described in the preceding sections. In doing so one should keep in mind that the gradient obtained by central difference is not necessarily the more accurate one.

The results, for 11 design variables, are given below

- *The gradient of a functional representing the deviation between an actual and a prescribed target C_p distribution.* This concerns the objective functional of Case E-1 to be described in Section 4.9. The design example

results in Figure 4.5 demonstrate close agreement between the three approaches, in particular between the results of the material derivative and discrete approaches.

- *The gradient of the lift coefficient C_l .* The gradient is computed for a fixed α . The results are presented in Figure 4.6. Reasonable agreement is shown between the gradients computed by the central difference method and the discrete approach. The result of the material derivative approach differs significantly, but it still follows the same trend which means that it generally gives the same descent direction.
- *The gradient of the drag coefficient C_d .* This concerns the objective functional of Case E-2 described in Section 4.9. Again, reasonable agreement is shown (Figure 4.7) between the gradients computed by the central difference method and the discrete approach, while the material derivative approach only shares the same trend.
- *Gradient of the pitching moment coefficient C_m .* This concerns the constraint functional of Case E-3 described in Section 4.9. The results are shown in Figure 4.8. Close agreement is shown between the results of the three approaches.

The discrete approach seems to produce the most consistent result. Therefore, this will be employed for solving all of the design cases discussed hereafter.

It should be mentioned that the central-difference method required a total of 23 flow analyses for each gradient shown in Figures 4.5–4.8. This illustrates the advantage of the variational method, which required the computational effort of only about 2 flow analyses for each gradient. There is also the difficulty with the central-difference method in choosing the magnitude of the design variable perturbation. In the present study, this has been determined by a trial-and-error procedure.

4.8 Optimization Routine

The present investigation employs the optimization routine FSQP, which stands for **F**easible **S**equential **Q**uadratic **P**rogramming. This routine is based on a modified Sequential Quadratic Programming (SQP) algorithm capable of generating feasible iterates. The detailed description of the algorithm used in FSQP is given in Ref. ([69]). For each iterate, of which the function values and the gradient of the objective/constraint are known, the algorithm determines a trajectory along which the vector of design variables is updated so as to reduce the objective while satisfying the constraints. In a generalized form, the update procedure can be expressed as

$$\theta^{j+1} = \theta^j + \beta \mathbf{S}$$

where \mathbf{S} represents the direction of the trajectory, and β is a scalar that determines the distance along \mathbf{S} which moves the current θ^j towards θ^{j+1} . The process of determining β is referred to as the *line search*.

Apart from the consideration that SQP is generally known to be the most cost-effective method for non-linear constrained optimization, the reasons for selecting FSQP are the followings,

- (i) FSQP generates feasible iterates with respect to constraints. This has a practical advantage that,—if the optimization process should be stopped at an intermediate stage—, the last iterate would still be useful in the sense that it would represent an improvement over the initial design while the constraints are satisfied. The situations in which an optimization process should be stopped are:
 - The last iterates correspond to small variations of the airfoil shape giving only marginal improvement in the objective. Stopping the process would not degrade the quality of the result, but obviously would avoid unnecessary additional expensive flow analyses.
 - In unfavorable cases[†], the line search might imply many flow analyses in order to find a new iterate.
 - The number of flow analyses exceeds a maximum number allowed (for example, limited by the available computational resources).
- (ii) If the initial design provided by the designer is infeasible for some inequality constraints, FSQP first generates a feasible iterate before minimizing the objective. This offers convenience if the designer is primarily concerned about the constraints. In this case, with no objective being specified, given an infeasible design the routine can provide the designer with a feasible one.
- (iii) FSQP has the capability of solving multi-objective optimization problems in a *min-max* sense. This is suitable for dealing with multi-point aerodynamic design to be described in Chapter 6.

4.9 Test Cases for Single-Point Design

Four test cases are considered designated as Case E-1, E-2, E-3, and E-4. The following conditions are common to all cases:

- The airfoil is defined by the parameterization scheme described in Section 4.5 with 11 design variables.
- The Euler computations have been performed by the inviscid mode of HI-TASK (Ref. [6]) on a 128×32 grid with the grid adaption procedure (Ref. [21]) activated. Figure 4.9 shows a typical grid produced by HI-TASK.
- The gradient is computed based on the discrete formulation of the variation of geometric properties (i.e., the *discrete approach* described in Section 4.4.2).

[†]For instance, the optimization process has reached a point in the design space where there are strong non-linearities.

A remark must be made with respect to accuracy. It is to be expected that for the grid mentioned above the numerical accuracy of the flow solver results will be quite modest in particular for the pressure drag; probably of the order of 0.01 in C_p and of the order of 0.001 to 0.01 in C_d .

4.9.1 Description and Results for Case E-1

Case E-1 is a reconstruction type inverse problem. The design point is specified as follows,

$$M = 0.72, \quad C_l = 1.$$

The target C_p is obtained from a flow analysis of a best-fit of the RAE 2822 airfoil. The target C_p distribution is defined on the airfoil chord with proper distinction between the lower and upper surface of the airfoil. The NACA 0012 airfoil is used as the starting airfoil geometry.

The objective functional to be minimized has the form

$$\mathcal{P} = \frac{1}{2} \int_0^1 (C_p - C_{p,t})_l^2 dx + \frac{1}{2} \int_0^1 (C_p - C_{p,t})_u^2 dx,$$

where x is coincident with the airfoil chord, and $C_{p,t}$ is the target value. The subscripts l and u refer to the lower and upper surface, respectively. \mathcal{P} can also be expressed in the form

$$\mathcal{P} = \frac{1}{2} \int_{S_a} (C_p - C_{p,t})^2 |n_y| dS,$$

In terms of the definition of the general functional (4.13), the function ψ in this case takes the form

$$\psi = \frac{1}{2} (C_p - C_{p,t})^2 |n_y|.$$

Recalling that

$$C_p = 2(p - p_\infty),$$

with p, p_∞ non-dimensionalized by $\rho_\infty V_\infty^2$, one has

$$\frac{\partial \psi}{\partial p} = 2(C_p - C_{p,t}) |n_y|.$$

This is substituted into the adjoint boundary condition (4.75). Then, the adjoint equation (4.21) is solved for λ , and ν is obtained from (4.31). The gradient is obtained by substituting \mathcal{P} into \mathcal{F} in equation (4.44).

One purpose of selecting this test case is to investigate the accuracy level of the computed gradient, because the optimal solution (i.e., the target airfoil) is known beforehand. The computed gradient is considered to be of sufficient accuracy if the optimal solution can be obtained. The optimal solution is assumed to be obtained if $\mathcal{P} \leq 10^{-4}$. This means that the difference between the actual and target C_p distributions is roughly within 0.01 (engineering accuracy).

Figure 4.10 shows the design iteration history. An empty circle, referred to as "Evaluated", represents one geometry and one flow analysis. A cross, referred

to as "New iterate", indicates that the corresponding geometry and flow analysis are used by the optimization algorithm as a basis for finding the next iterate. The gradient needs to be computed only for the crossed circles. The process from one crossed circle to another represents a line search (cf. Section 4.8).

As can be seen, engineering accuracy has been achieved in the 24th iteration. One may thus conclude that the adjoint formulation and gradient evaluations are correct[†]. The corresponding C_p distribution and airfoil geometry are shown in Figures 4.11, which demonstrate that the best-fit of the RAE 2822 has been closely recovered.

The results have been obtained with the design variables scaled by the procedure described in Section 4.6. In order to study the effect of the scaling, the same case was run again with (i) the unscaled design variables and (ii) the design variables scaled by a constant ($\mathbf{d} = 0.1$). The latter implies larger shape modifications than the former. Figure 4.12 gives a comparison in terms of the number of aerodynamic analyses. This clearly demonstrates the effectiveness of the aerodynamic scaling.

It should be noted that the convergence in the case with aerodynamic scaling can be accelerated by multiplying the scaling factor \mathbf{d} by a positive constant number $r < 1$. This may be considered as an over-relaxation factor which gives larger shape modifications in each iteration. It has been found that with $r = 0.75$, the engineering accuracy for the same case can be reached in about 19 iterations (Figure 4.12).

As a final remark, a comparison with a residual-correction inverse method of Fray et al. [17] is made. It was concluded by Fray et al. that for transonic flow cases the residual-correction type inverse method can reach engineering accuracy in 10-15 iterations. The present method requires a slightly larger computational effort. However, it offers much more flexibility in the choice of objective functionals and in dealing with (geometric) constraints.

4.9.2 Description and Results for Case E-2

Case E-2 concerns a wave drag reduction problem with geometric constraints for

$$M = 0.72, \quad C_l = 1.$$

The optimization starts with an initial airfoil specified as a best-fit of the RAE 2822 airfoil. Geometric constraints are imposed on the trailing edge included angle and on the cross-sectional area. These are limited to be not less than their initial values (given by the initial airfoil). Thus, the optimization begins with all constraints active.

The objective functional to be minimized represents the pressure drag coefficient C_d ,

$$\mathcal{P} = C_d = - \int_{S_a} C_p (n_x \cos \alpha + n_y \sin \alpha) dS,$$

[†]In particular, this also confirms that the far-field boundary condition for λ has been formulated correctly, recalling that equation (4.39) contains an integral taken over the far-field boundary S_∞ .

which, in inviscid, transonic flow is equal to the (shock) wave drag. In terms of the definition of the general functional (4.13), the function ψ in this case takes the form

$$\psi = -C_p(n_x \cos \alpha + n_y \sin \alpha),$$

giving

$$\frac{\partial \psi}{\partial p} = -2(n_x \cos \alpha + n_y \sin \alpha).$$

This is substituted into the adjoint boundary condition (4.75). Subsequently, the adjoint equation (4.21) is solved for λ , and ν is obtained from (4.31). The gradient is determined by substituting \mathcal{P} into \mathcal{F} in equation (4.44).

The same aerodynamic scaling as in the previous case is applied. Although the scaling was formulated for the functional (4.50), numerical experiments showed that it is also effective for drag reduction problems.

The final iterate corresponds to the 19th flow analysis. The optimization process was stopped after the line search spent 5 flow analyses (the 20th to 24th) without finding a new (improved) iterate.

Figure 4.13 shows how the drag is reduced as the optimization proceeds, with a final drag reduction of about 0.0140 (140 "counts"). In order to determine the significant amount of the drag reduction that results from the optimization, grid refinement studies need to be performed. This gives an estimate of the numerical error-bandwidth of the computation of drag. A drag reduction which is larger than the error-bandwidth can be considered to be significant.

The grid refinement studies are described as follows. A flow analysis is done for both the initial and the final airfoil on a (256×64) - and (512×128) -grid. If the grid-point density of 512×128 corresponds to a mesh-size of h , the grid-point density of 256×64 and 128×32 corresponds to the mesh size of $2h$ and $4h$, respectively. The computed drag value as a function of the squared value of the mesh size is shown in Figure (4.14). The straight-looking curves shown in the figure demonstrate that the numerical scheme is of second-order accuracy.

The drag value corresponding to the mesh-size of zero is obtained through extrapolation based on a linear-fit of the curve (see Table 4.1). Subtracting this from the drag value obtained from the (128×32) -grid, used for the optimization, yields the absolute error of the computed drag. The difference between the absolute error of the initial airfoil and that of the final airfoil gives an estimate of the numerical error-bandwidth of the drag computation occurring in the optimization, which is 18 counts. Thus, of the 140 counts of drag reduction resulting from the optimization, only 122 counts should be considered as significant.

A qualitative assessment of the optimization result can also be done by means of the contour-plot of the value of the Mach number and the total pressure in the flow field. These are shown in Figure 4.15 and 4.16, respectively. These figures illustrate that the shock wave has been eliminated around the final airfoil.

The states of geometric constraints corresponding to each iterate are shown in Figure 4.19. The trailing edge included angle constraint was not active during the optimization. In contrast, the cross-sectional area constraint was always active, meaning that the optimization proceeded along this constraint boundary.

The resulting C_p distribution and airfoil geometry are shown in Figure 4.17. As can be seen, the shock wave has practically been removed from the surface pressure distribution. However, the drag as indicated in Figure 4.13 is still quite high and indeed very high for a shockless flow. This is probably to be attributed to the numerical errors in the flow solution. Indeed, for a typical subsonic flow analysis for which the drag should be zero, the flow solver appears to yield about the same level of drag counts. The error could of course be reduced by increasing the grid resolution. A more principal aspect demonstrated by the result is that the method is capable of removing the shock wave from the C_p distribution.

Interesting phenomena occurred between the 10th and 12th analysis. Small geometric modifications from the 8th to the 10th analysis were followed by a rather drastic change in the geometry from the 10th to the 12th analysis as shown in Figure 4.18. The optimization moved the solution from a conventionally looking family of C_p distributions (or airfoils) up to the 10th analysis to a rather peculiar one given by the supposedly optimal solution at the 19th analysis. This is accompanied by a steep decrease in the pitching moment coefficient (stronger nose-down moment). The phenomena also suggest that the optimization has proceeded from a well-behaved part of the design space to a highly non-linear one.

One can not exclude the possibility that both families of C_p distributions represent two physically shockless solutions. In other words, subtracting the numerical errors from the computed drag could possibly yield zero drag in both cases. If that is the case, the optimal solution would not be unique and additional constraints, e.g. on pitching moment, would be needed for obtaining uniqueness. This suggests that each family of C_p distributions could belong to a distinct local minimum which happens also to be a global minimum, because drag cannot be less than zero.

4.9.3 Description and Results for Case E-3

Case E-3 is a drag reduction problem with geometric as well as aerodynamic constraints. The case specification is the same as Case E-2, except that

- A constraint on the pitching moment coefficient is introduced.
- A leading edge radius constraint is added into the set of geometric constraints.

These constraints must not be less than their initial values. Hence, like in the previous case, the optimization starts with all constraints active.

The constraint functional C_m is formulated as

$$C_m = - \int_{S_a} C_p (n_x y - n_y (x - 0.25)) dS.$$

which can be substituted into \mathcal{F} in order to obtain the gradient. The function ψ for C_m takes the form

$$\psi = -C_p (n_x y - n_y (x - 0.25)),$$

giving

$$\frac{\partial \psi}{\partial p} = -2 (n_x y - n_y (x - 0.25)).$$

The optimization process was stopped after the maximum number of 25 flow analyses was exceeded. The drag reduction history is shown in Figure 4.20. The final iterate corresponds to the 26th flow analysis with a drag reduction of about 112 counts. Grid refinement studies for this case, like those applied to Case E-2, have been performed and summarized in Figure 4.21 and Table 4.2. The error-bandwidth is 3 counts. This means that the whole drag reduction of 112 resulting from the optimization can practically be considered to be significant.

The resulting C_p distribution and airfoil geometry are shown in Figure 4.22. As can be seen, the shock wave has been weakened significantly.

The pitching moment constraint (Figure 4.20) is active in all optimization stages. On the other hand, the geometric constraints are not active at most stages of optimization (Figure 4.23) including the final stage. Hence, the pitching moment may be considered as the determining factor.

4.9.4 Description and Results for Case E-4

The specification of this case is the same as Case E-3, except for the flow conditions

$$M = 0.78, \quad C_l = 0.5$$

The reason for specifying this case is to provide a basis of comparison with the multi-point aerodynamic design to be described in Chapter 6.

The optimization process was stopped after the line search spent 5 flow analyses (the 11th to 15th) without finding a new (improved) iterate. The drag reduction history is shown in Figure 4.24. The final iterate corresponds to the 10th flow analysis and a drag reduction of 101 counts. Grid refinement studies for this case, like those applied to Case E-2, have been performed and summarized in Figure 4.25 and Table 4.3. The error-bandwidth is 1 count. This means that the whole drag reduction of 101 resulting from the optimization can practically be considered to be significant.

The resulting C_p distribution and airfoil geometry are shown in Figure 4.26. As can be seen, the shock wave has, again, practically been removed. The cross-sectional area constraint is the only active constraint.

4.10 Concluding Remarks

The results given above indicate that the present method represents a viable approach for solving constrained single-point transonic aerodynamic design (drag reduction) problems, based on the compressible inviscid flow model described by the Euler equations. The possibility for incorporating both aerodynamic and geometric constraints is of great practical value, since in real design practice one is always confronted with such design constraints.

Compared with direct finite difference type of optimization methods, the variational method has proved to be a very cheap method for computing the gradient. The computation of the gradient has been validated by comparison with alternative methods in Section 4.7.4 and the reconstruction of an "optimal" solution in Section 4.9. It suggests that the computed gradient is of sufficient accuracy to be used for general aerodynamic optimization problems.

The shape parameterization formulated in Section 4.5 was found to be able to describe a large variation of airfoil shapes. That the curvature is approximately linear with the design variables opens the possibility of aerodynamic scaling. The aerodynamic scaling has proved to be effective in reducing the number of function evaluations in the case of inverse as well as drag reduction problems.

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	211	71	140
2	2	163	37	126
3	1	151	28	123
4	0	147	25	122
Error: (1)-(4)		64	46	18

Table 4.1: Grid Refinement Studies for Drag of Case E-2

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	211	99	112
2	2	163	49	114
3	1	151	36	115
4	0	147	32	115
Error: (1)-(4)		64	67	-3

Table 4.2: Grid Refinement Studies for Drag of Case E-3

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	164	63	101
2	2	134	31	103
3	1	124	22	102
4	0	122	20	102
Error: (1)-(4)		42	43	-1

Table 4.3: Grid Refinement Studies for Drag of Case E-4

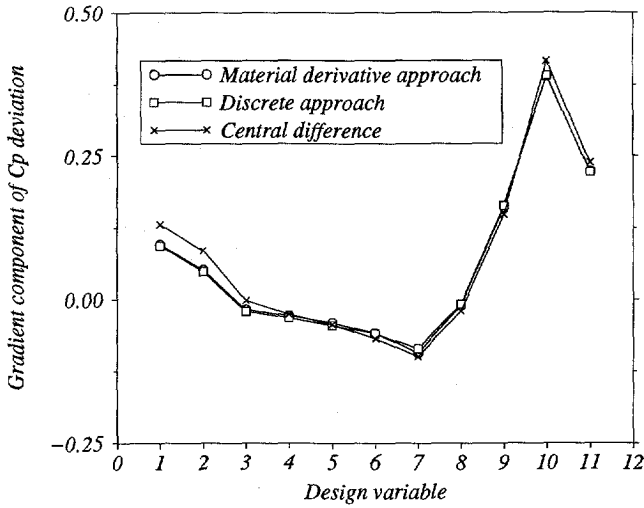


Figure 4.5: Comparison of the gradients of the functional representing an inverse problem (*Flow condition:* $M = 0.72$, $C_l = 1$. *Airfoil:* RAE2822).

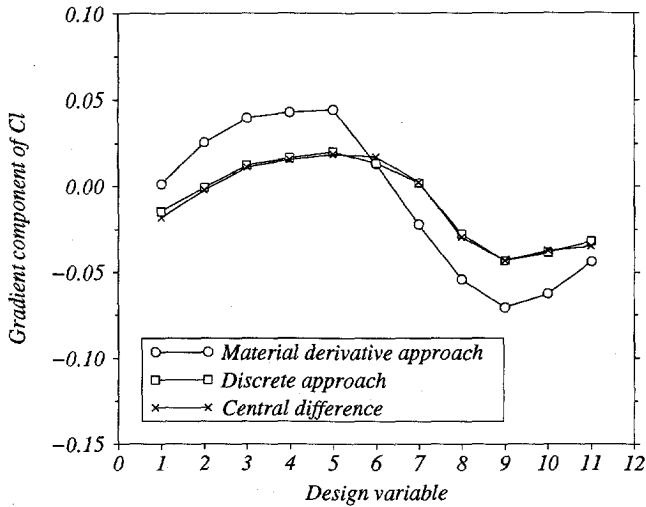


Figure 4.6: Comparison of the gradients of the lift coefficient C_l (*Flow condition:* $M = 0.72$, $\alpha = 2.5^\circ$. *Airfoil:* RAE2822).

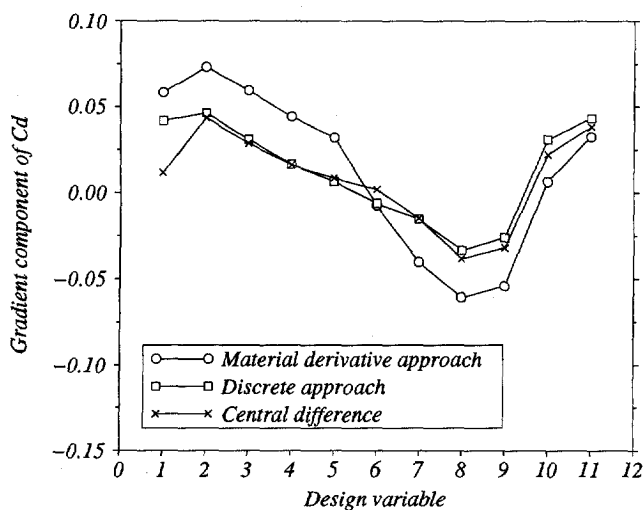


Figure 4.7: Comparison of the gradients of the drag coefficient C_d (Flow condition: $M = 0.72$, $C_l = 1$. Airfoil: RAE2822).

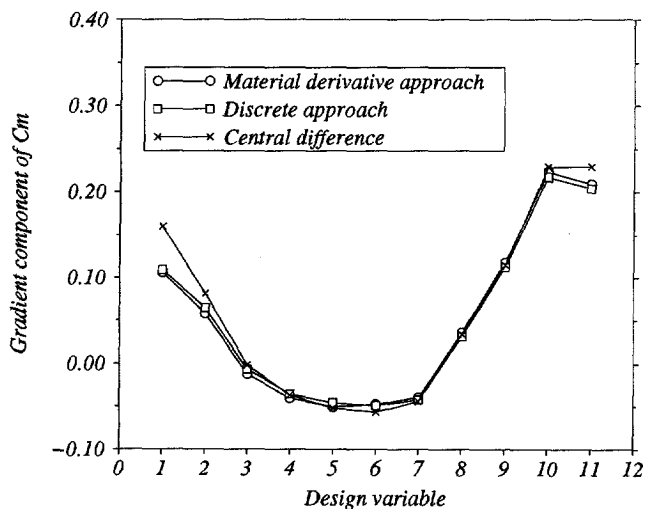


Figure 4.8: Comparison of the gradients of the pitching moment coefficient C_m (Flow condition: $M = 0.72$, $C_l = 1$. Airfoil: RAE2822).

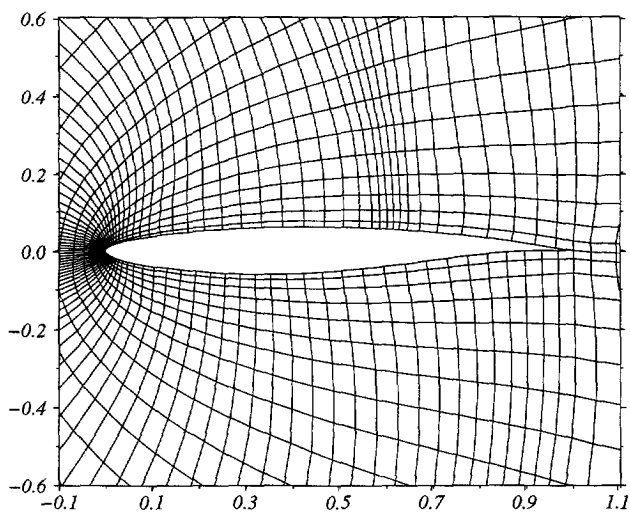


Figure 4.9: Typical 128×32 Euler grid after adaption.

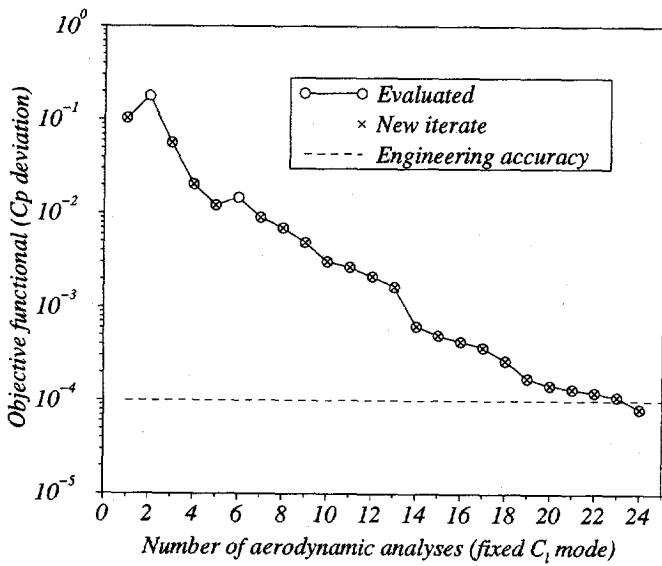


Figure 4.10: Objective for Case E-1 (Transonic reconstruction inverse design. Flow model: Euler. Design point: $M = 0.72$, $C_l = 1$. Initial airfoil: NACA 0012. Target airfoil: RAE2822).

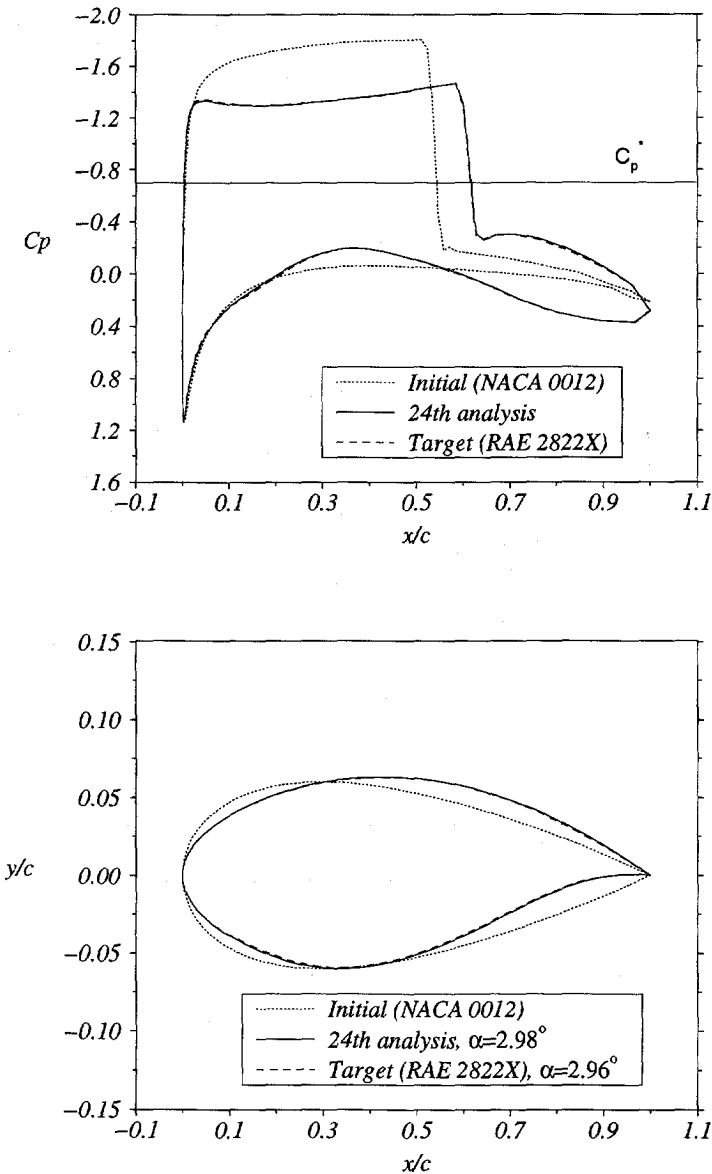


Figure 4.11: C_p distribution and airfoil geometry for Case E-1 (*Subsonic reconstruction inverse design*. Flow model: Euler equations. Design point: $M = 0.72$, $C_l = 1$. Initial airfoil: NACA 0012. Target airfoil: RAE2822).

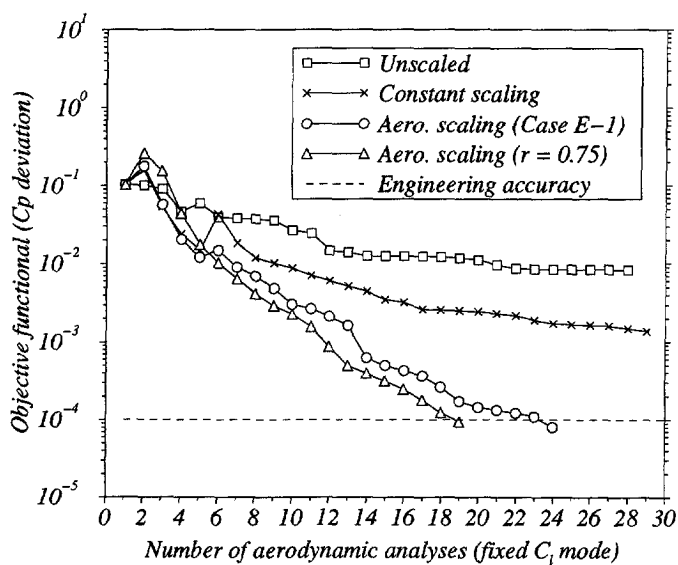


Figure 4.12: Effect of the aerodynamic scaling.

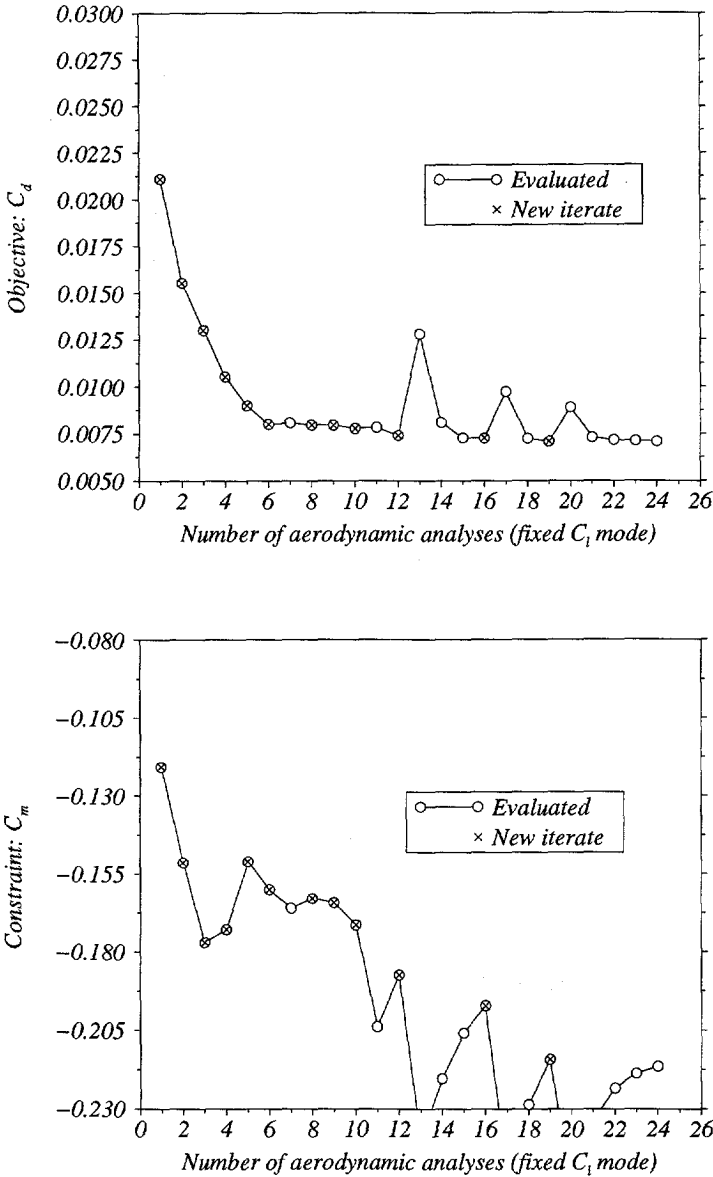


Figure 4.13: Wave drag coefficients for Case E-2 (Transonic constrained wave drag reduction. Flow model: Euler equations. Design point: $M = 0.72$, $C_l = 1$. Initial airfoil: RAE2822. Constraints: trailing edge included angle and cross sectional area).

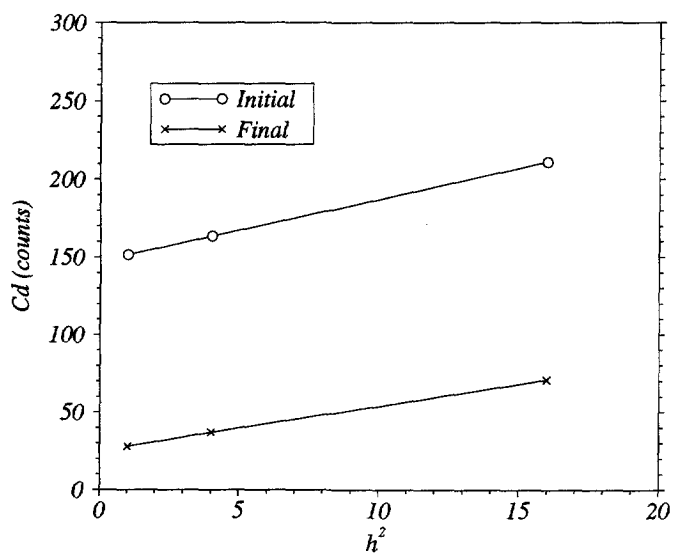


Figure 4.14: Grid refinement studies for Drag of Case E-2

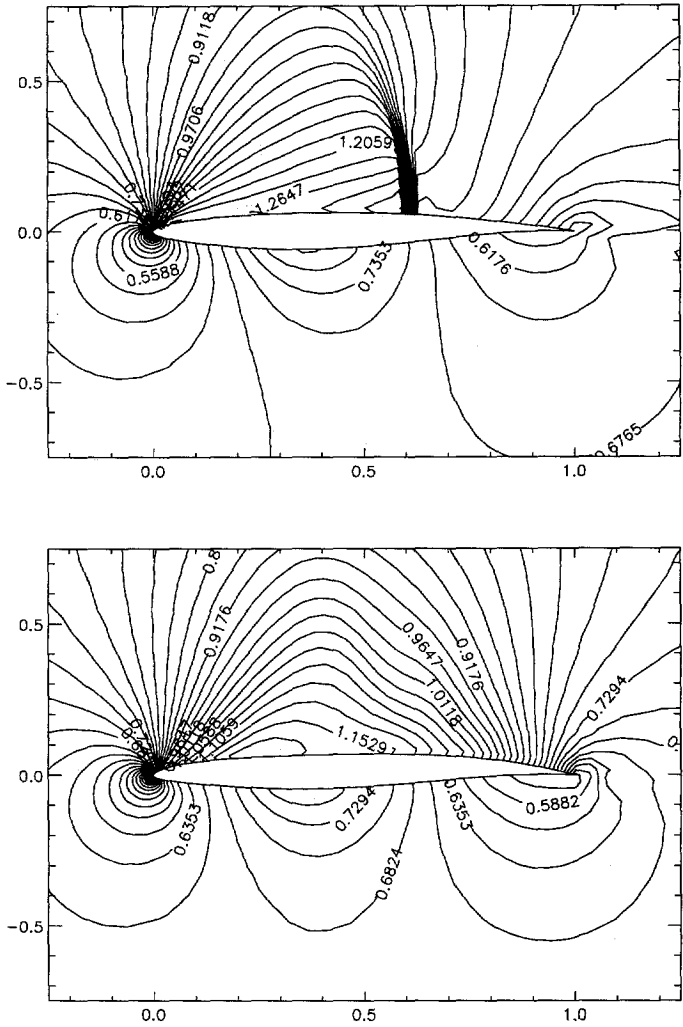


Figure 4.15: Mach contour around the initial airfoil (top) and the final airfoil (bottom) for Case E-2.

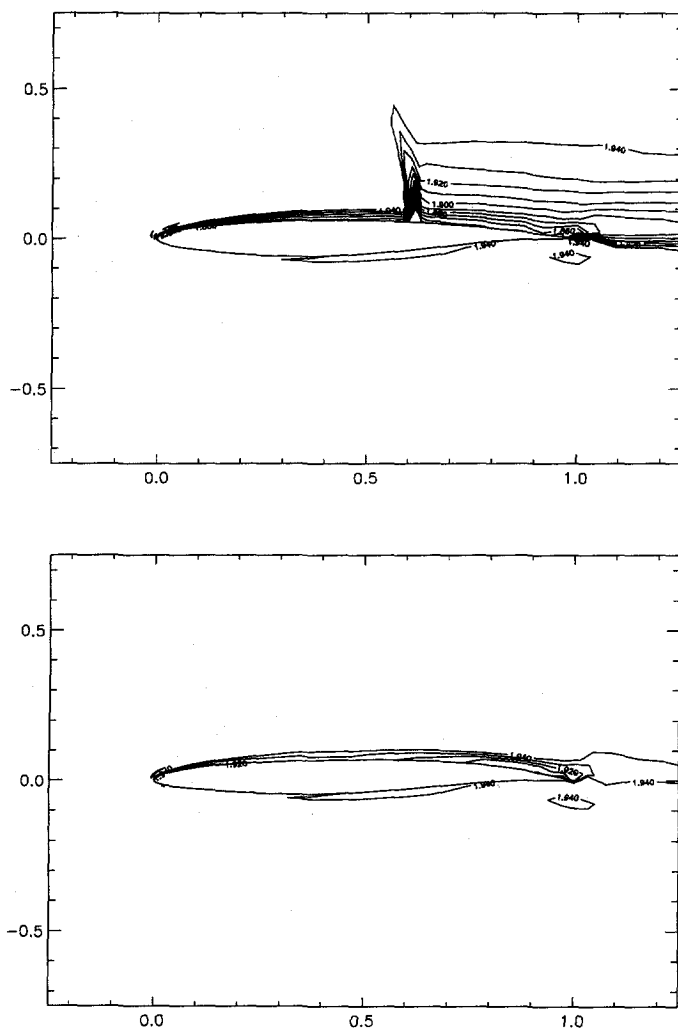


Figure 4.16: Total pressure contour around the initial airfoil (top) and the final airfoil (bottom) for Case E-2.

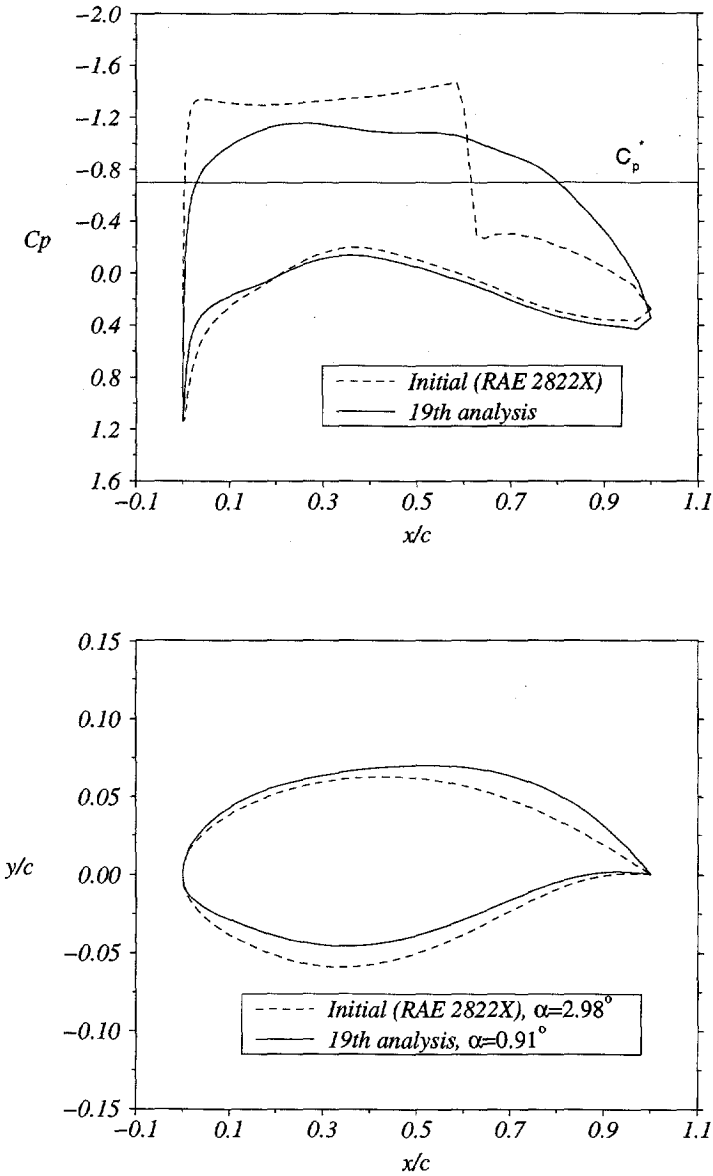


Figure 4.17: C_p distribution and airfoil geometry for Case E-2 (Transonic constrained wave drag reduction). Flow model: Euler. Design point: $M = 0.72$, $C_l = 1$. Initial airfoil: RAE2822. Constraints: trailing edge included angle and cross sectional area).

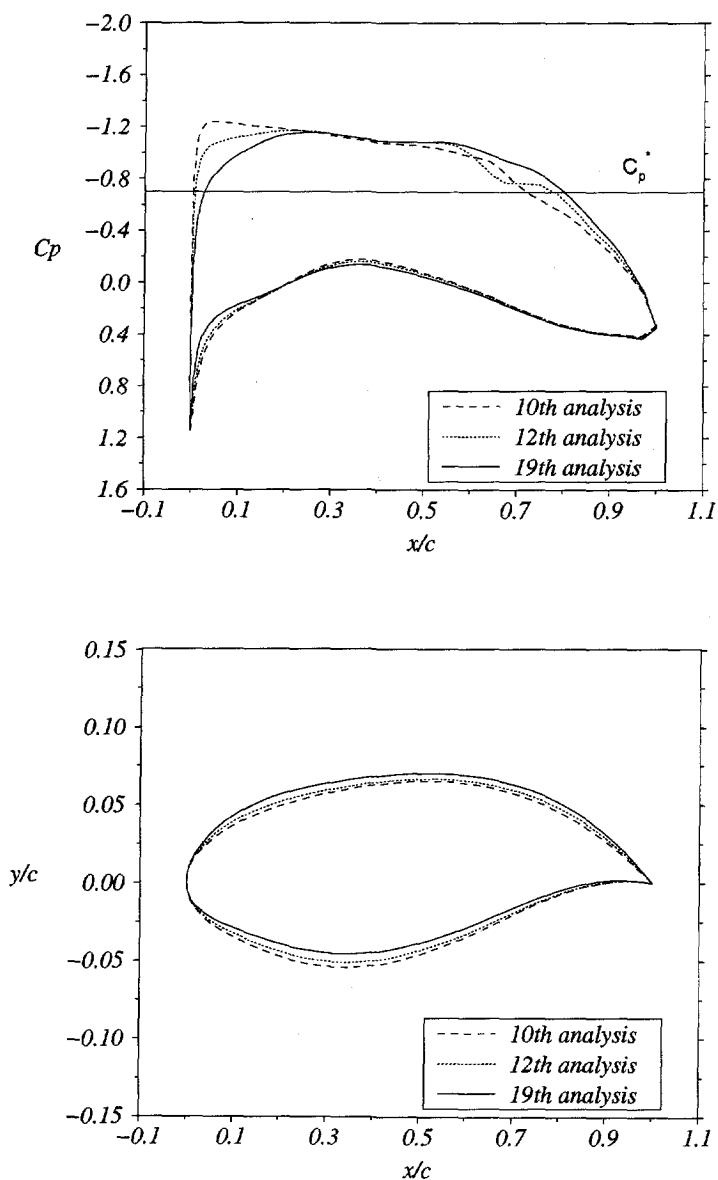


Figure 4.18: Two families of C_p distribution and airfoil geometry discovered in Case E-2.

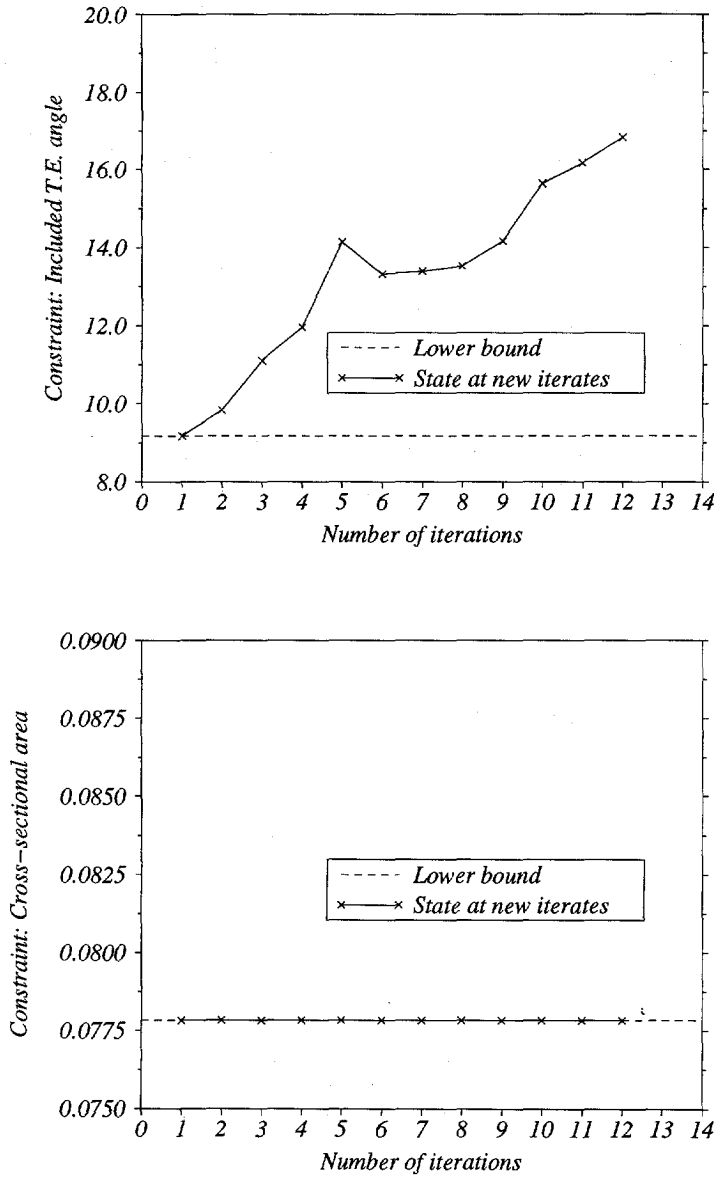


Figure 4.19: State of geometric constraints for Case E-2.

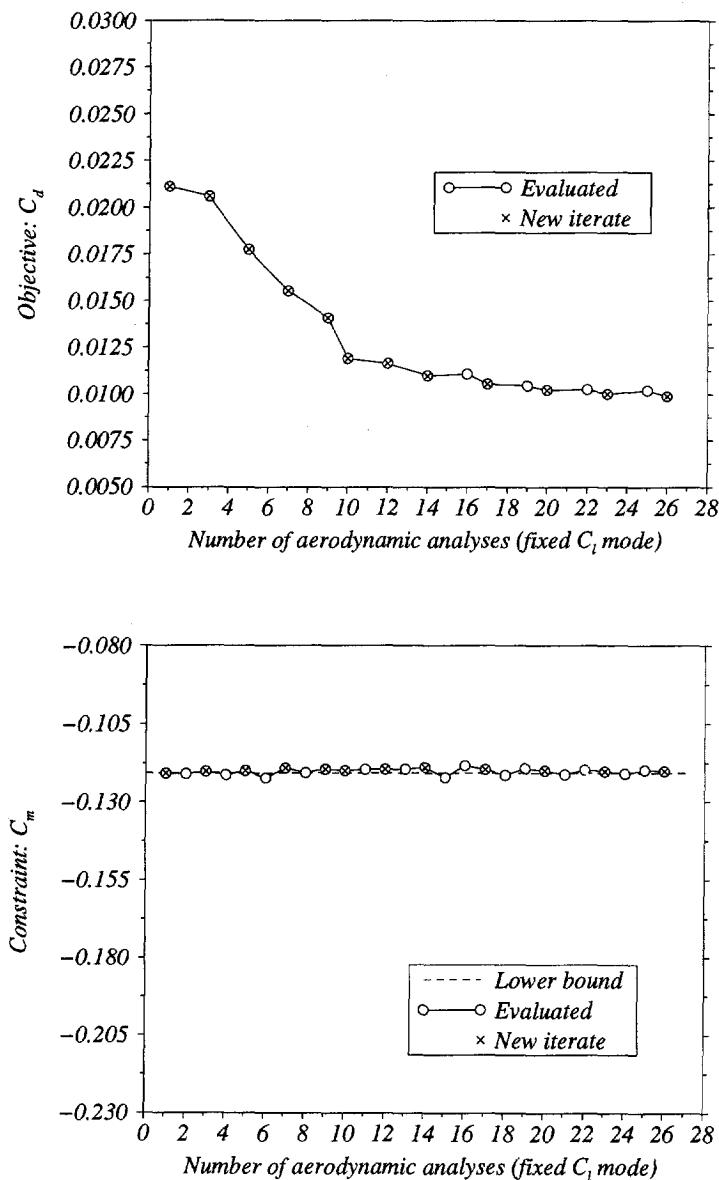


Figure 4.20: Wave drag and pitching moment coefficients for Case E-3 (Transonic constrained wave drag reduction). *Flow model:* Euler. *Design point:* $M = 0.72$, $C_l = 1$. *Initial airfoil:* RAE2822. *Constraints:* pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

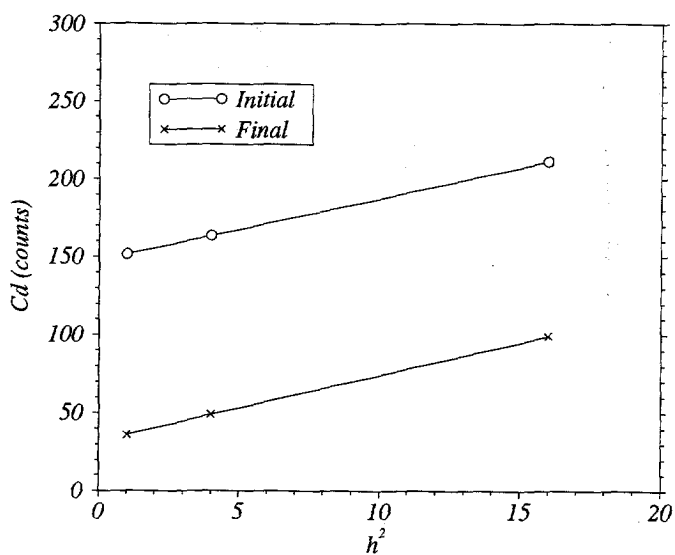


Figure 4.21: Grid refinement studies for Drag of Case E-3

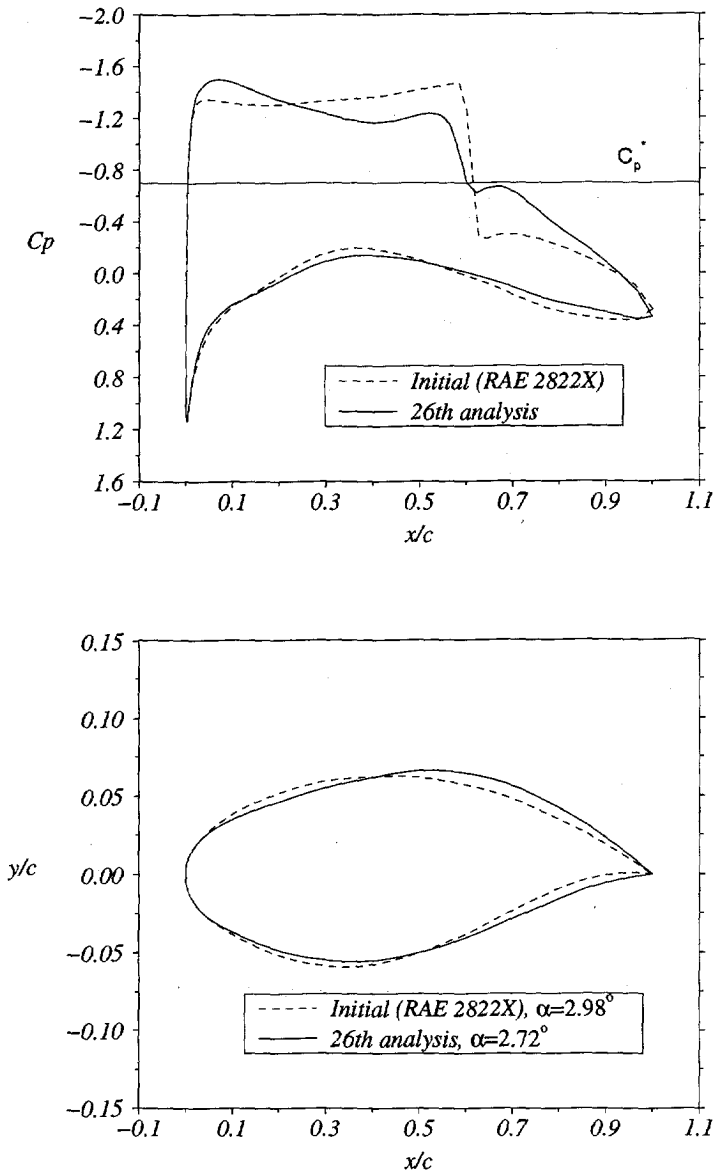


Figure 4.22: C_p distribution and airfoil geometry for Case E-3 (Transonic constrained wave drag reduction. Flow model: Euler. Design point: $M = 0.72$, $C_l = 1$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

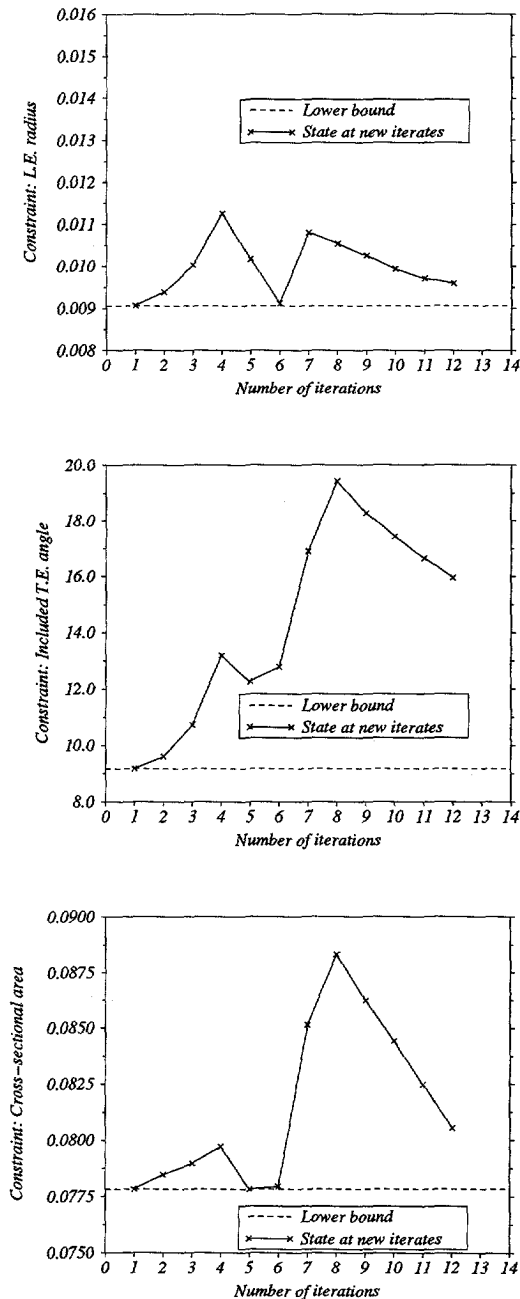


Figure 4.23: State of geometric constraints for Case E-3.

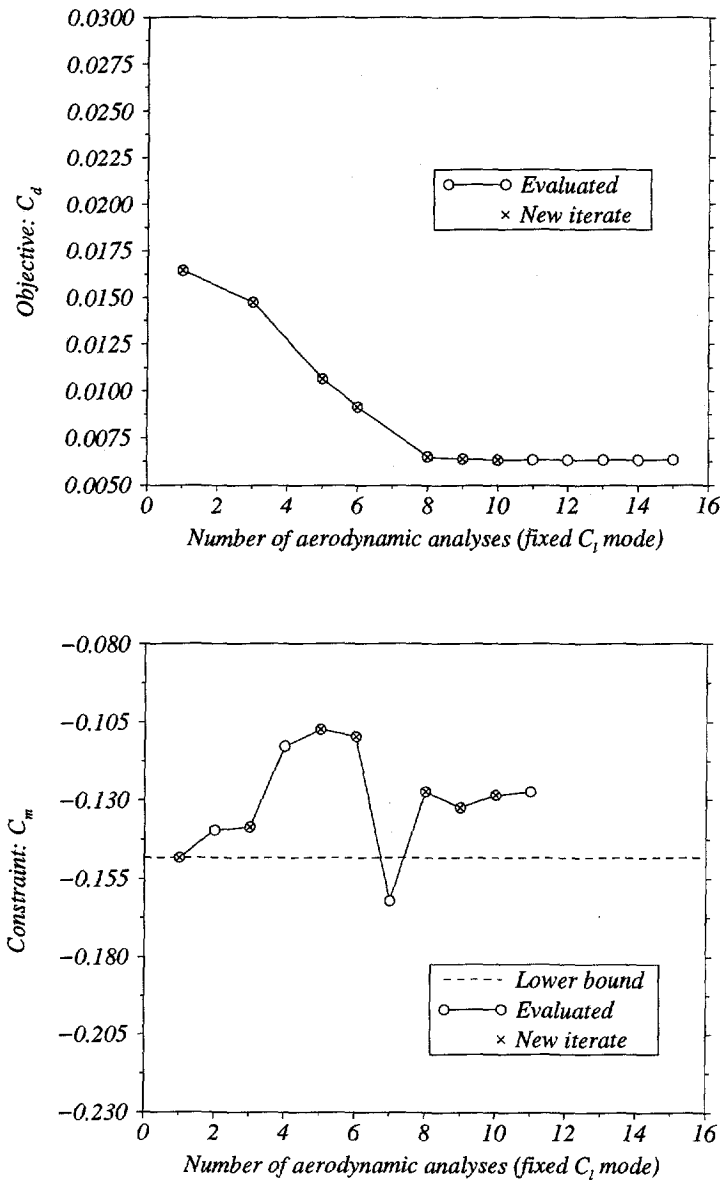


Figure 4.24: Wave drag and pitching moment coefficients for Case E-4 (Transonic constrained wave drag reduction). *Flow model: Euler. Design point: $M = 0.78$, $C_l = 0.5$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area.*

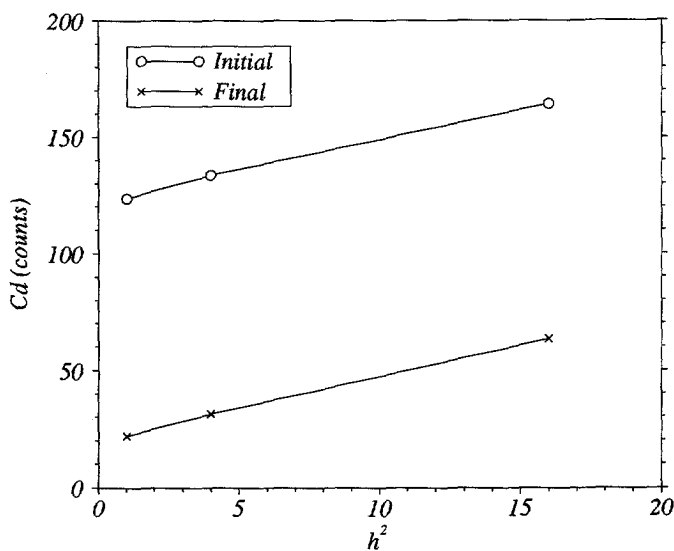


Figure 4.25: Grid refinement studies for Drag of Case E-4

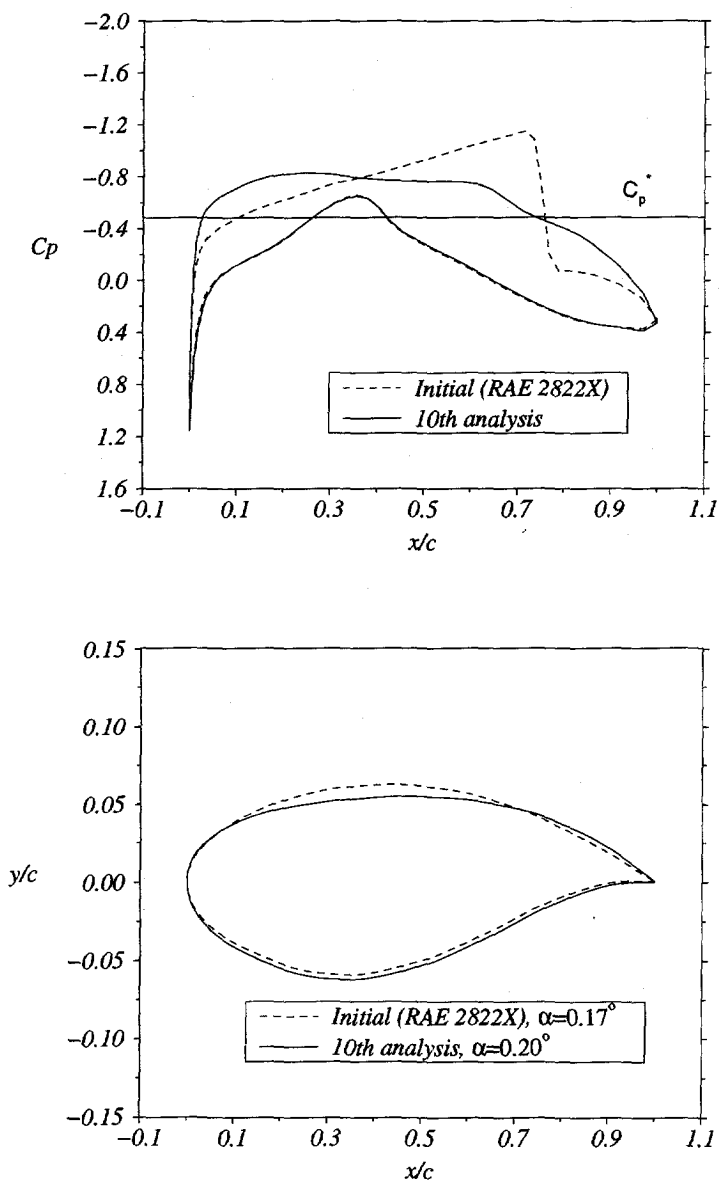


Figure 4.26: C_p distribution and airfoil geometry for Case E-4 (Transonic constrained wave drag reduction). Flow model: Euler. Design point: $M = 0.78$, $C_l = 0.5$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

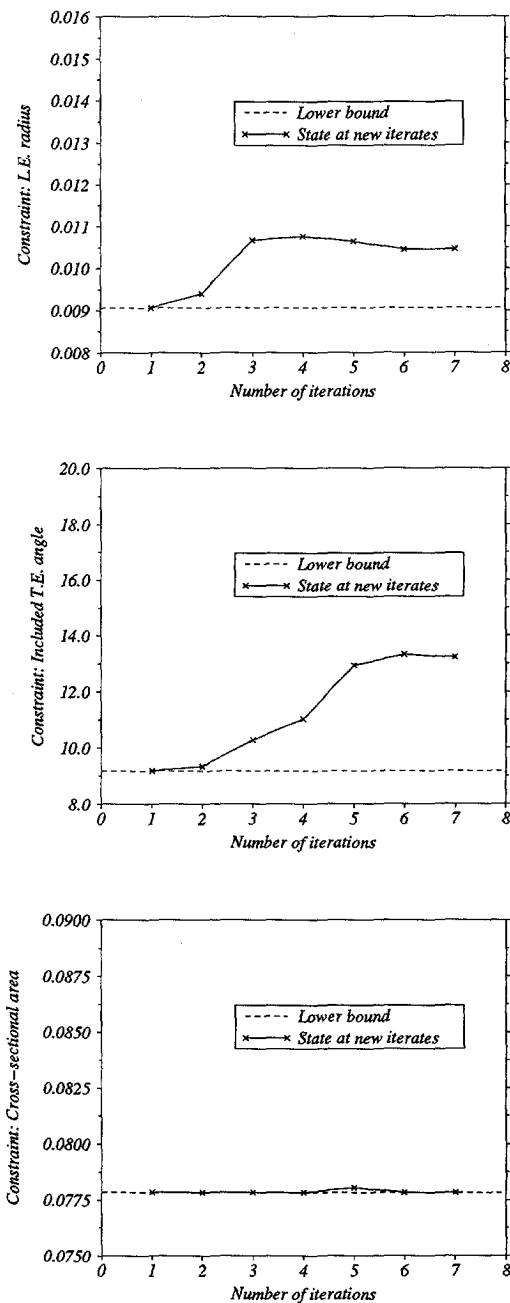


Figure 4.27: State of geometric constraints for Case E-4.

Chapter 5

Aerodynamic Optimization using the Reynolds-Averaged Navier-Stokes Equations

5.1 Introduction

Although aerodynamic optimization based on the Euler equations can be of practical significance, it is of course desirable to be able to perform design studies on the basis of a more complete model of the flow physics such as the Reynolds-Averaged Navier-Stokes (RANS) equations. Relatively few efforts have been reported in the literature (e.g., [20], [42], [8]). Reasons might be: (i) the relatively high amount of computational resources required, (ii) the complexity in constructing a design process with a consistent treatment of the viscous phenomena and, possibly, (iii) lack of suitability of the flow solver.

In the present chapter, The same transonic airfoil design problems as in Chapter 4 are addressed, but now with viscous effects included. This permits a comparative study between the "inviscid" and "viscous" design solutions. The design problems are posed as optimization problems involving aerodynamic functionals representing the lift, drag, and pitching moment coefficients. These are defined in terms of the pressure coefficient C_p obtained from the flow variables \mathbf{Q} , where \mathbf{Q} is the solution of the RANS equations.

As in Chapter 4, the design variables are geometric parameters θ which define the airfoil geometry and the angle of attack α . The optimal values of θ and α are to be obtained by means of a gradient-based optimization algorithm. The gradient of the aerodynamic functionals with respect to the design variables θ and α are computed by means of the variational method.

This chapter is organized as follows. A statement of the design problem is given in Section 5.2. A brief description of the Navier-Stokes equations is presented in Section 5.3. The adjoint and gradient equations are formulated in Section 5.4, followed by a description of the numerical implementation in Section 5.5. Test cases and computational results are discussed in Section 5.6. Finally, some conclusions are drawn in Section 5.7.

5.2 Statement of the Design Problem

The optimization problem being addressed can be written as follows,

$$\begin{aligned} &\text{Minimize} \quad \mathcal{P}(\mathbf{Q}, \boldsymbol{\theta}, \alpha), \\ &\text{Subject to:} \quad \mathcal{A}(\mathbf{Q}, \boldsymbol{\theta}, \alpha) \leq 0, \\ &\quad \quad \quad \mathbf{G}(\boldsymbol{\theta}) \leq 0, \end{aligned} \tag{5.1}$$

where \mathcal{P} and \mathcal{A} represent such aerodynamic functionals as lift, drag and pitching moment coefficients, while the vector \mathbf{G} represents geometric constraints. The geometric parameters $\boldsymbol{\theta}$ and the angle of attack α are treated as the design variables, of which the optimal values are to be determined.

Problem (5.1) is subject to the RANS equations for a given value of Mach number and Reynolds number. The RANS equations impose an implicit dependency of the flow variables \mathbf{Q} upon $\boldsymbol{\theta}$ and α .

Since, as in Chapter 4, problem (5.1) is to be solved by means of a gradient-based optimization algorithm, the gradient of \mathcal{P} , \mathcal{A} , and \mathbf{G} with respect to the design variables $\boldsymbol{\theta}$ must be computed. The gradient of \mathbf{G} can be obtained rather easily by direct analytical differentiation. The gradient of the aerodynamic functionals \mathcal{P} and \mathcal{A} will be computed by means of the variational method. This means that an adjoint problem must be formulated, the solution of which is used for evaluating the gradient.

The methodology presented in this chapter takes advantage of the availability of the HI-TASK flow solver and the optimization routine already introduced in Chapter 4. The main task to be done is to develop an adjoint solver and gradient evaluator, and to integrate these with the existing flow solver and optimization routine into a design code.

5.3 The Reynolds-Averaged Navier-Stokes Equations

Figure (4.1) in Chapter 4 gives the nomenclature. Assuming adiabatic flow and no external forces, the time-dependent RANS equations in the conservative form are written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \vec{\nabla} \cdot \vec{\mathbf{F}} = 0, \tag{5.2}$$

where \mathbf{Q} is the vector of conservative time-averaged flow variables:

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \tag{5.3}$$

which have been non-dimensionalized with respect to the free stream. At the steady-state, equation (5.2) becomes

$$\vec{\nabla} \cdot \vec{\mathbf{F}} = 0. \tag{5.4}$$

The flux $\vec{\mathbf{F}}$ consists of the convective, $\vec{\mathbf{F}}_c$, and viscous, $\vec{\mathbf{F}}_v$, flux vectors,

$$\vec{\mathbf{F}} = \vec{\mathbf{F}}_c - \vec{\mathbf{F}}_v. \quad (5.5)$$

The convective flux vector is the same as that given in equation (4.6). The viscous flux vector is defined as

$$\vec{\mathbf{F}}_v = \begin{pmatrix} f_v \\ g_v \end{pmatrix}, \quad (5.6)$$

where f_v and g_v are the Cartesian components given by

$$\mathbf{f}_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xx}u + \tau_{xy}v - q_x \end{pmatrix}, \quad \mathbf{g}_v = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{xy}u + \tau_{yy}v - q_y \end{pmatrix}. \quad (5.7)$$

Assuming that air behaves like a Newtonian fluid, the elements τ_{xx} , τ_{xy} , and τ_{yy} of the viscous stress tensor are expressed as

$$\tau_{xx} = l_d(\vec{\nabla} \cdot \vec{V}) + 2\mu_d \frac{\partial u}{\partial x} - \overline{\rho u' u'}, \quad (5.8)$$

$$\tau_{yy} = l_d(\vec{\nabla} \cdot \vec{V}) + 2\mu_d \frac{\partial v}{\partial y} - \overline{\rho v' v'}, \quad (5.9)$$

$$\tau_{xy} = \mu_d \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) - \overline{\rho u' v'}, \quad (5.10)$$

where μ_d is the dynamic viscosity. The last term on the right-hand side of equations (5.8)-(5.10) are the Reynolds stresses. The Stokes hypothesis provides

$$l_d = -\frac{2}{3}\mu_d.$$

The heat flux vector \mathbf{q} consists of a laminar part and a turbulent part, the components of which are defined by

$$q_x = -\kappa \frac{\partial T}{\partial x} + \overline{\rho h' u'}, \quad (5.11)$$

$$q_y = -\kappa \frac{\partial T}{\partial y} + \overline{\rho h' v'}, \quad (5.12)$$

where the thermal conductivity coefficient κ is related to the Prandtl number by

$$\text{Pr} = \frac{c_p \mu_d}{\kappa}, \quad (5.13)$$

c_p is the specific heat at constant pressure, T is the absolute temperature and h is the mass specific enthalpy. The Prandtl number is assumed to have a constant value, $\text{Pr} = 0.72$, throughout the flow.

The total energy E per unit mass is defined as

$$E = e + \frac{1}{2}(u^2 + v^2),$$

where e is the internal energy per unit mass. The RANS equations are closed by the equation of state of a calorically perfect gas, given as follows

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho(u^2 + v^2)\right), \quad (5.14)$$

$$T = \frac{1}{c_v}\left(E - \frac{1}{2}(u^2 + v^2)\right), \quad (5.15)$$

where $\gamma = c_p/c_v$, with c_v the specific heat at constant volume. The dynamic viscosity is obtained from the Sutherland's law,

$$\frac{\mu_d}{\mu_\infty} = \left(\frac{T}{T_\infty}\right)^{3/2} \frac{T_\infty + 110}{T + 110}. \quad (5.16)$$

The turbulent terms occurring in equations (5.8)-(5.10) and (5.11)-(5.12) are modeled by the concept of "eddy viscosity". The Reynolds stress terms are assumed to satisfy the same expression as the laminar parts, except that μ_d is replaced by a turbulent or eddy viscosity, μ_t . The same applies for the heat flux, where a turbulent conductivity κ_t can be introduced. μ_t and κ_t are related by a turbulent Prandtl number Pr_t , i.e.

$$Pr_t = c_p \mu_t / \kappa_t.$$

which is also assumed constant, $Pr_t = 0.9$.

With the definition of the turbulent viscosity, equations (5.8)-(5.10) can be expressed as follows,

$$\tau_{xx} = l(\vec{\nabla} \cdot \vec{V}) + 2\mu \frac{\partial u}{\partial x}, \quad (5.17)$$

$$\tau_{yy} = l(\vec{\nabla} \cdot \vec{V}) + 2\mu \frac{\partial v}{\partial y}, \quad (5.18)$$

$$\tau_{xy} = \mu \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right), \quad (5.19)$$

where

$$l = -\frac{2}{3}(\mu_d + \mu_t)$$

$$\mu = \mu_d + \mu_t$$

Using equations (5.13) and (5.15), the heat fluxes can be written as

$$q_x = -\gamma \frac{\mu}{Pr} \frac{\partial e}{\partial x}, \quad (5.20)$$

$$q_y = -\gamma \frac{\mu}{Pr} \frac{\partial e}{\partial y}, \quad (5.21)$$

where

$$\gamma \frac{\mu}{Pr} = \gamma \left(\frac{\mu_d}{Pr} + \frac{\mu_t}{Pr_t} \right).$$

The eddy viscosity μ_t is obtained by means of a turbulence model. In the flow solver HI-TASK, the Baldwin-Lomax turbulence model is implemented. This is formulated in terms of the kinematic viscosity,

$$\nu_t = \frac{\mu_t}{\rho},$$

based on a two-layer model,

$$\nu_t = \begin{cases} \nu_{ti}, & \text{for } n \leq n_c \\ \nu_{to}, & \text{for } n > n_c \end{cases}$$

where n is the normal distance to the wall and n_c is the smallest value of n where the values from the *inner* and *outer* formulations are identical. A detailed descriptions of the turbulence model for determining ν_t is given in Ref. [3].

On the airfoil surface, S_a , the no-slip and adiabatic boundary conditions are applied. The no-slip boundary condition can be expressed as

$$\rho \vec{V} \cdot \vec{n} = 0, \quad (5.22)$$

$$\rho \vec{V} \cdot \vec{s} = 0. \quad (5.23)$$

The adiabatic wall boundary condition is expressed as

$$\vec{\nabla} T \cdot \vec{n} = 0, \quad (5.24)$$

which, in terms of the internal energy, is equivalent to

$$\vec{\nabla} e \cdot \vec{n} = 0. \quad (5.25)$$

The boundary conditions can be combined into a vector \mathbf{B} :

$$\mathbf{B} = \begin{pmatrix} \rho \vec{V} \cdot \vec{n} \\ \rho \vec{V} \cdot \vec{s} \\ \vec{\nabla} e \cdot \vec{n} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (5.26)$$

The boundary conditions on the far-field boundaries S_∞ are formulated in the same way as in the case of the Euler equations[†] (cf. Section 4.3).

5.4 Formulation of the Adjoint and Gradient Equations

For the design cases to be considered, a general form of the aerodynamic functional is assumed as follows,

$$\mathcal{F} = \int_{S_a} \psi(p, \tau_w, \theta, \alpha) dS. \quad (5.27)$$

[†]The viscous terms are neglected in the far-field.

where ψ is a function of the pressure p , the wall shear stress τ_w and the design variables (θ and the angle of attack α). The notation \mathcal{F} applies to both \mathcal{P} and \mathcal{A} in the problem statement (5.1).

Employing the variational method implies that the adjoint equation must be formulated the solution of which provides the Lagrange multipliers. These in turn are used in evaluating the gradient of the functional with respect to the design variables θ and α .

As p and τ_w are obtained from \mathbf{Q} , which satisfies the steady-state RANS equations and the boundary conditions (5.26), the functional \mathcal{F} is independent of the transient state. Therefore, it is sufficient to consider the steady-state RANS equations (5.4) and the boundary conditions (5.26) in the definition of a Lagrangian \mathcal{L} as follows,

$$\mathcal{L} = \int_{S_a} \psi \, dS + \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) \, d\Omega + \int_{S_a} \boldsymbol{\Upsilon} \cdot \mathbf{B} \, dS, \quad (5.28)$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\Upsilon}$ are the vectors of Lagrange multipliers. $\boldsymbol{\lambda}$ has four components considered as corresponding to the components of \mathbf{Q} . The Lagrange multiplier $\boldsymbol{\Upsilon}$ has three components defined over S_a .

The variation of \mathcal{L} is due to independent variations of $\boldsymbol{\lambda}$, $\boldsymbol{\Upsilon}$, τ_w , \mathbf{Q} , θ and α :

$$\delta \mathcal{L} = \delta \mathcal{L}_{\boldsymbol{\lambda}} + \delta \mathcal{L}_{\boldsymbol{\Upsilon}} + \delta \mathcal{L}_{\tau} + \delta \mathcal{L}_{\mathbf{Q}} + \delta \mathcal{L}_{\theta} + \delta \mathcal{L}_{\alpha}.$$

Although the wall shear stress τ_w is dependent upon \mathbf{Q} in the form of a partial derivative,

$$\tau_w = \mu \frac{\partial V_s}{\partial n},$$

its variation, τ'_w , cannot be expressed in terms of \mathbf{Q}' , but in term of the normal derivative of the tangential velocity. Hence, τ_w is treated as an independent variable in the definition of the Lagrangian.

5.4.1 The Adjoint Equation

Like in the Euler case, it can be shown that $\delta \mathcal{L}_{\boldsymbol{\lambda}}$ and $\delta \mathcal{L}_{\boldsymbol{\Upsilon}}$ vanish if the RANS equations (5.4) and boundary conditions (5.26) are satisfied. The variation of \mathcal{L} then becomes

$$\delta \mathcal{L} = \delta \mathcal{L}_{\tau} + \delta \mathcal{L}_{\mathbf{Q}} + \delta \mathcal{L}_{\theta} + \delta \mathcal{L}_{\alpha}.$$

The adjoint equation and its boundary conditions follow from the condition

$$\begin{aligned} \delta \mathcal{L}_{\mathbf{Q}} &= 0, \\ \delta \mathcal{L}_{\tau} &= 0. \end{aligned}$$

In order to evaluate $\delta \mathcal{L}_{\mathbf{Q}}$, the terms associated with the inviscid, $\vec{\mathbf{F}}_c$, and viscous, $\vec{\mathbf{F}}_v$, fluxes are treated separately, giving

$$\begin{aligned} \delta \mathcal{L}_{\mathbf{Q}} &= \int_{S_a} \left(\frac{\partial \psi}{\partial p} \frac{\partial p}{\partial \mathbf{Q}} \cdot \mathbf{Q}' + \frac{\partial \psi}{\partial \tau_w} \tau'_w \right) dS \\ &\quad + \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}'_c) \, d\Omega - \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}'_v) \, d\Omega + \int_{S_a} \boldsymbol{\Upsilon} \cdot \mathbf{B}' \, dS. \end{aligned}$$

At this stage it is convenient to introduce convective and viscous variations, $\delta\mathcal{I}$ and $\delta\mathcal{J}$, defined as

$$\begin{aligned}\delta\mathcal{I} &= \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}'_c) d\Omega, \\ \delta\mathcal{J} &= \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}'_v) d\Omega.\end{aligned}$$

The inviscid term $\delta\mathcal{I}$ is treated in a similar way as in the Euler case. This gives

$$\begin{aligned}\delta\mathcal{I} &= - \int_{S_a} (\gamma - 1) (n_x \lambda_2 + n_y \lambda_3) (\rho E)' dS \\ &\quad - \int_{S_a} \left(\lambda_1 + \frac{a^2}{\gamma - 1} \lambda_4 \right) \rho (n_x u' + n_y v') dS \\ &\quad - \int_{S_{\infty}} (C^T \boldsymbol{\lambda}) \cdot \mathbf{Q}' dS - \int_{\Omega} (\vec{A}^T \cdot \vec{\nabla} \boldsymbol{\lambda}) \cdot \mathbf{Q}' d\Omega.\end{aligned}$$

In obtaining the viscous term $\delta\mathcal{J}$, the variation of the viscosity, μ' , due to the variation of the flow variables, \mathbf{Q}' , is assumed negligible. This assumption is also driven by practical reasons, because otherwise one would have to deal with complicated mathematical expressions in formulating the adjoint equations. The viscous fluxes on the far-field boundary are also assumed negligible. This assumption is in agreement with the treatment of the far-field boundary conditions for the RANS equations, where use is made of the characteristic relations of the Euler equations.

The procedure for obtaining $\delta\mathcal{J}$ is described in detail in Appendix D. The final result, equation (D.25), is rewritten here for convenience,

$$\begin{aligned}\delta\mathcal{J} &= - \int_{S_a} \left[(\vec{\lambda} \cdot \vec{s}) \tau'_w + \lambda_4 \left(\tau_w (\vec{V}' \cdot \vec{s}) + \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} e' \cdot \vec{n}) \right) \right. \\ &\quad \left. - \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right) (\vec{V}' \cdot \vec{n}) \right. \\ &\quad \left. - \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) (\vec{V}' \cdot \vec{s}) - \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' \right] dS \\ &\quad + \int_{\Omega} Y^T \mathbf{K} \cdot \mathbf{Q}' d\Omega.\end{aligned}$$

The variation $\delta\mathcal{L}_Q + \delta\mathcal{L}_\tau$ can now be obtained as

$$\begin{aligned}\delta\mathcal{L}_Q + \delta\mathcal{L}_\tau &= \int_{S_a} \left(\frac{\partial \psi}{\partial p} (\gamma - 1) (\rho E)' + \frac{\partial \psi}{\partial \tau_w} \tau'_w \right) dS + \delta\mathcal{I} - \delta\mathcal{J} \\ &= \int_{S_a} \left(\frac{\partial \psi}{\partial p} (\gamma - 1) (\rho E)' + \frac{\partial \psi}{\partial \tau_w} \tau'_w \right) dS \\ &\quad - \int_{S_a} (\vec{\lambda} \cdot \vec{n}) (\gamma - 1) (\rho E)' dS - \int_{S_a} \left(\lambda_1 + \frac{a^2}{\gamma - 1} \lambda_4 \right) (\rho \vec{V}' \cdot \vec{n}) dS \\ &\quad + \int_{S_a} \left[(\vec{\lambda} \cdot \vec{s}) \tau'_w + \lambda_4 \left(\tau_w (\vec{V}' \cdot \vec{s}) + \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} e' \cdot \vec{n}) \right) \right.\end{aligned}$$

$$\begin{aligned}
& - \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right) (\vec{V}' \cdot \vec{n}) \\
& - \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) (\vec{V}' \cdot \vec{s}) - \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' \Big] dS \\
& - \int_{S_\infty} (C^\top \lambda) \cdot \mathbf{Q}' dS - \int_\Omega (\vec{A}^\top \cdot \vec{\nabla} \lambda + Y^\top \mathbf{K}) \cdot \mathbf{Q}' d\Omega \\
& + \int_{S_a} (\Gamma_1(\rho \vec{V}' \cdot \vec{n}) + \Gamma_2(\rho \vec{V}' \cdot \vec{s}) + \Gamma_3(\vec{\nabla} e' \cdot \vec{n})) dS, \tag{5.29}
\end{aligned}$$

where $\vec{\lambda}$ denotes an adjoint velocity vector with the Cartesian components λ_2 and λ_3 given by

$$\vec{\lambda} = \begin{pmatrix} \lambda_2 \\ \lambda_3 \end{pmatrix},$$

and

$$\begin{aligned}
\lambda_n &= \vec{\lambda} \cdot \vec{n}, \\
\lambda_s &= \vec{\lambda} \cdot \vec{s}.
\end{aligned}$$

The matrix \vec{A} is the Jacobian of the convective flux vector $\vec{\mathbf{F}}_c$ with respect to the conservative flow variables \mathbf{Q} . C is the Jacobian of the normal component of the convective flux. Y is the Jacobian of the primitive flow variables $\mathbf{U} = (\rho, u, v, p)^\top$ with respect to the conservative flow variables \mathbf{Q} , so that

$$\mathbf{U} = Y\mathbf{Q}.$$

The vector \mathbf{K} is defined in equation (D.24) as

$$\mathbf{K} = \begin{pmatrix} \frac{a^2(\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) 1}{(\gamma - 1)\text{Pr} \rho} \\ \frac{\partial \Gamma_{xx}}{\partial x} + \frac{\partial \Gamma_{xy}}{\partial y} - \tau_{xx} \frac{\partial \lambda_4}{\partial x} - \tau_{xy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xx}}{\partial x} + \frac{\partial \Psi_{xy}}{\partial y} \\ \frac{\partial \Gamma_{xy}}{\partial x} + \frac{\partial \Gamma_{yy}}{\partial y} - \tau_{xy} \frac{\partial \lambda_4}{\partial x} - \tau_{yy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xy}}{\partial x} + \frac{\partial \Psi_{yy}}{\partial y} \\ \frac{a^2(\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) 1}{(\gamma - 1)\text{Pr} p} \end{pmatrix}.$$

The parameters Γ_{xx} , Γ_{yy} , and Γ_{xy} are given by equations (D.13), (D.15), and (D.14) as follows,

$$\begin{aligned}
\Gamma_{xx} &= l\vec{\nabla} \cdot \vec{\lambda} + 2\mu \frac{\partial \lambda_2}{\partial x}, \\
\Gamma_{yy} &= l\vec{\nabla} \cdot \vec{\lambda} + 2\mu \frac{\partial \lambda_3}{\partial y}, \\
\Gamma_{xy} &= \mu \left(\frac{\partial \lambda_3}{\partial x} + \frac{\partial \lambda_2}{\partial y} \right).
\end{aligned}$$

which may be considered as corresponding to the elements of the stress tensor τ_{xx} , τ_{xy} , and τ_{yy} of equations (5.8), (5.9), and (5.10). The parameters Ψ_{xx} , Ψ_{yy} , and Ψ_{xy} are defined in equations (D.17), (D.18), and (D.19) as

$$\begin{aligned}\Psi_{xx} &= (l + 2\mu)u \frac{\partial \lambda_4}{\partial x} + lv \frac{\partial \lambda_4}{\partial y}, \\ \Psi_{yy} &= (l + 2\mu)v \frac{\partial \lambda_4}{\partial y} + lu \frac{\partial \lambda_4}{\partial x}, \\ \Psi_{xy} &= \mu \left(u \frac{\partial \lambda_4}{\partial y} + v \frac{\partial \lambda_4}{\partial x} \right).\end{aligned}$$

Setting the domain integral in equation (5.29) to zero leads to the adjoint equation:

$$\vec{A}^T \cdot \vec{\nabla} \lambda + Y^T \mathbf{K} = 0 \quad \text{in } \Omega. \quad (5.30)$$

Three adjoint boundary conditions to be applied on S_a are obtained by setting equal to zero the coefficients of $(\rho E)'$, τ'_w and e' , which gives

$$\vec{\lambda} \cdot \vec{n} = \frac{\partial \psi}{\partial p}, \quad (5.31)$$

$$\vec{\lambda} \cdot \vec{s} = -\frac{\partial \psi}{\partial \tau_w}, \quad (5.32)$$

$$\vec{\nabla} \lambda_4 \cdot \vec{n} = 0. \quad (5.33)$$

Equations (5.31) and (5.32) may be considered as corresponding to the no-slip boundary conditions (5.22) and (5.23), while equation (5.33) corresponds to the adiabatic wall boundary condition (5.25).

The Lagrange multipliers \mathbf{Y} follow from the elimination of the terms containing $(\vec{V}' \cdot \vec{n})$, $(\vec{V}' \cdot \vec{s})$ and $(\vec{\nabla} e' \cdot \vec{n})$, i.e.

$$\gamma_1 = \left(\lambda_1 + \frac{a^2}{\gamma - 1} \lambda_4 \right) + \frac{1}{\rho} \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right), \quad (5.34)$$

$$\gamma_2 = -\frac{1}{\rho} \left(\lambda_4 \tau_w - \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) \right), \quad (5.35)$$

$$\gamma_3 = -\lambda_4 \left(\gamma \frac{\mu}{P_r} \right). \quad (5.36)$$

The surface integral over S_∞ is dealt with in the same way as in the Euler case. Thus, equation (4.29) is also used as the boundary condition for the adjoint equation (5.30).

The adjoint problem can be summarized as follows. The adjoint equation (5.30) is solved for λ , subject to the boundary conditions (5.31)–(5.33) on S_a and (4.29) on S_∞ . As λ is available, the Lagrange multipliers \mathbf{Y} are determined from equations (5.34)–(5.36).

5.4.2 The Gradient Equation

As the flow and adjoint equations are satisfied, the variation of \mathcal{L} becomes

$$\delta\mathcal{L} = \delta\mathcal{L}_\theta + \delta\mathcal{L}_\alpha.$$

As in the Euler case, two approaches can be distinguished for obtaining the gradient: (i) the material derivative approach, and (ii) the discrete approach. However, as it will turn out that physical considerations are needed in obtaining suitable expressions of the gradients, it is more convenient to give the explanation in terms of the material derivatives.

Following the same procedure as in the Euler case, one obtains

$$\begin{aligned}\delta\mathcal{L}_\theta &= \int_{S_a} \frac{\partial\psi}{\partial\theta} \cdot \delta\theta \, dS + \int_{S_a} \psi \, d\dot{S} \\ &\quad + \int_{S_a} [\gamma_1 \rho (\dot{n}_x u + \dot{n}_y v) + \gamma_2 \rho (\dot{n}_y u - \dot{n}_x v)] \, dS \\ &\quad + \int_{S_a} \gamma_3 (\vec{\nabla} e \cdot \dot{\vec{n}}) \, dS.\end{aligned}$$

Taking the no-slip boundary condition into account, i.e.

$$u = 0, \quad v = 0,$$

gives

$$\delta\mathcal{L}_\theta = \int_{S_a} \frac{\partial\psi}{\partial\theta} \cdot \delta\theta \, dS + \int_{S_a} \psi \, d\dot{S} + \int_{S_a} \gamma_3 (\vec{\nabla} e \cdot \dot{\vec{n}}) \, dS,$$

with the material derivative formulae given in Appendix G. However, it turns out that this procedure does not lead to a correct expression for the gradient[†]. Also, the resulting expression does not approach that of the Euler case for $\text{Re} \rightarrow \infty$.

In order to obtain a suitable expression for the gradient, it is helpful to consider some aspects of the physics of viscous flow. The structure of viscous flow at high Reynolds number with a thin boundary-layer along the solid surface implies an indirect control mechanism between the airfoil shape and the (inviscid) external flow. If the streamlines across the boundary-layer (from the airfoil surface up to the edge of boundary-layer) are assumed to be parallel with the airfoil surface, one may write

$$\frac{\partial(\rho \vec{V} \cdot \vec{n})}{\partial\eta} = 0, \quad \eta_o \leq \eta \leq \eta_\infty, \quad (5.37)$$

where η is normal to the surface, while η_o and η_∞ refer to the airfoil surface and boundary-layer edge, respectively. Two subdomains Ω_1 and Ω_2 are introduced as follows,

$$\begin{aligned}\Omega_1 &= \int_{S_a} \int_{\eta_o}^{\eta_\infty} d\eta \, dS, \\ \Omega_2 &= \Omega - \Omega_1.\end{aligned}$$

[†]This conclusion is based on numerical observations suggesting that the steepest descent direction was not obtained.

Referring to the Lagrangian (5.28), the term associated with the Navier-Stokes equations can be written as:

$$\begin{aligned}\mathcal{L}_\Omega &= \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\Omega \\ &= \int_{\Omega_1 + \Omega_2} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\Omega \\ &= \int_{S_a} \int_{\eta_0}^{\eta_\infty} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\eta dS + \int_{\Omega_2} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}) d\Omega.\end{aligned}$$

For evaluating the variation $\delta\mathcal{L}_\theta$ of \mathcal{L} due to the variation of the airfoil shape, only the integral over Ω_1 is considered. The integrand $(\vec{\nabla} \cdot \vec{\mathbf{F}})$ can be expressed in terms of partial derivatives with respect to η and s . There appear terms which are identical with the left-hand side of equation (5.37). Numerical observations indicate that this term is the only one which has a significant variation with respect to θ . Dropping the other terms, the variation $\delta\mathcal{L}_\theta$ can be obtained as

$$\delta\mathcal{L}_\theta = \int_{S_a} \frac{\partial\psi}{\partial\boldsymbol{\theta}} \cdot \delta\boldsymbol{\theta} dS + \int_{S_a} \int_{\eta_0}^{\eta_\infty} U(\eta) \left(\frac{\partial(\rho\vec{V} \cdot \dot{\vec{n}})}{\partial\eta} \right) d\eta dS,$$

where

$$U(\eta) = \left(\lambda_1 + u\lambda_2 + v\lambda_3 + \left(\frac{u^2 + v^2}{2} + \frac{a^2}{\gamma - 1} \right) \lambda_4 \right) \Big|_{\eta}.$$

Substitution of equation (G.10) for $\dot{\vec{n}}$ gives

$$\begin{aligned}\delta\mathcal{L}_\theta &= \int_{S_a} \frac{\partial\psi}{\partial\boldsymbol{\theta}} \cdot \delta\boldsymbol{\theta} dS \\ &\quad - \int_{S_a} \int_{\eta_0}^{\eta_\infty} U(\eta) \left((\boldsymbol{\chi}_{n,s} + H\boldsymbol{\chi}_s) \frac{\partial(\rho\vec{V} \cdot \vec{s})}{\partial\eta} \right) \cdot \delta\boldsymbol{\theta} d\eta dS,\end{aligned}$$

so that

$$\frac{d\mathcal{F}}{d\boldsymbol{\theta}} = \int_{S_a} \left(\frac{\partial\psi}{\partial\boldsymbol{\theta}} - \int_{\eta_0}^{\eta_\infty} U(\eta) (\boldsymbol{\chi}_{n,s} + H\boldsymbol{\chi}_s) \frac{\partial(\rho\vec{V} \cdot \vec{s})}{\partial\eta} d\eta \right) dS. \quad (5.38)$$

In the limit of $\text{Re} \rightarrow \infty$ the boundary-layer becomes infinitesimally thin, so that the Navier-Stokes equations approach the Euler equations, which implies that the gradient equation (5.38) approaches equation (4.37). This is confirmed by the fact that, for an infinitesimally thin boundary-layer,

$$\int_{\eta_0}^{\eta_\infty \rightarrow \eta_0} U(\eta) (\boldsymbol{\chi}_{n,s} + H\boldsymbol{\chi}_s) \frac{\partial(\rho\vec{V} \cdot \vec{s})}{\partial\eta} d\eta \rightarrow \Upsilon(\rho\vec{V} \cdot \vec{s}) (\boldsymbol{\chi}_{n,s} + H\boldsymbol{\chi}_s),$$

where Υ is given by equation (4.31). The expression for the gradient with respect to the angle of attack is the same as that of the Euler case, i.e.

$$\frac{d\mathcal{F}}{d\alpha} = \lim_{\delta\alpha \rightarrow 0} \frac{\delta\mathcal{L}_\alpha}{\delta\alpha} = \int_{S_a} \frac{\partial\psi}{\partial\alpha} dS - \int_{S_\infty} (C\mathbf{X}_+)^T \boldsymbol{\lambda} \cdot \frac{\partial\mathbf{W}_+}{\partial\alpha} dS. \quad (4.39)$$

Also, for a fixed C_l the same procedure as in the Euler case can be followed, and equation (4.44) is also relevant for the Navier-Stokes case, i.e.

$$\left(\frac{d\mathcal{F}}{d\theta}\right)_{C_l} = \frac{d\mathcal{F}}{d\theta} - \left(\frac{d\mathcal{F}}{d\alpha}\right) \left(\frac{dC_l}{d\alpha}\right)^{-1} \left(\frac{dC_l}{d\theta}\right). \quad (4.44)$$

5.5 Numerical Implementation

As already indicated, the investigation carried out in this chapter also makes use of the HI-TASK code of the National Aerospace Laboratory NLR. The main features of this code have already been described in Section 4.7.

5.5.1 Discretization of the Adjoint Equations

In order to take advantage of the numerical schemes of the HI-TASK flow solver, the adjoint equation (5.30) should be considered as the steady-state of a time-dependent system. The time-dependent adjoint equation is formulated as

$$\frac{\partial \lambda}{\partial t} - A_x^T \frac{\partial \lambda}{\partial x} - A_y^T \frac{\partial \lambda}{\partial y} - Y^T \mathbf{K} = 0. \quad (5.39)$$

In order to adopt the finite-volume scheme employed by HI-TASK for solving the adjoint equation, a divergence form of equation (5.39) is needed. This is obtained with the same procedure as described for the Euler case. The adjoint solver employs the secondary control volume indicated in Figure 4.4, in which the Jacobian \tilde{A} , the viscosity μ , and the elements of the stress tensor (τ_{xx} , τ_{yy} , τ_{xy}) are assumed constant. This leads to the definition of the adjoint convective and viscous fluxes as given in Appendices E and F.

The adjoint equation (5.39) is discretized into

$$\Delta \Omega_{i,j} \frac{d\lambda}{dt} + \int_{S_{i,j}} \tilde{\mathbf{Z}} \cdot \tilde{\mathbf{n}} dS + \tilde{\mathbf{D}}_{i,j} = 0, \quad (5.40)$$

where $\tilde{\mathbf{Z}}$ is the adjoint flux and the term $\tilde{\mathbf{D}}_{i,j}$ represents a 4-th and 2-nd order artificial dissipation term. These are formulated in the same way as in the flow solver. Equation (5.40) is integrated in time towards the steady-state with the same scheme as employed by the flow solver.

5.5.2 Implementation of the Adjoint Boundary Conditions

The boundary conditions on the far-field boundary S_∞ are implemented in the same way as in the Euler case.

The boundary conditions on the airfoil surface S_a are implemented as follows. Equations (5.31)–(5.32) fix the value of λ_2 and λ_3 . With the no-slip boundary conditions applied, the fourth adjoint equation on S_a reduces to

$$\frac{\partial \lambda_4}{\partial t} - (\gamma - 1) \left(\frac{\partial \lambda_2}{\partial x} + \frac{\partial \lambda_3}{\partial y} \right) - \frac{\partial}{\partial x} \left(\frac{a^2 \mu}{\rho \text{Pr}} \frac{\partial \lambda_4}{\partial x} \right) - \frac{\partial}{\partial y} \left(\frac{a^2 \mu}{\rho \text{Pr}} \frac{\partial \lambda_4}{\partial y} \right) = 0. \quad (5.41)$$

With the adjoint boundary condition (5.33) applied, equation (5.41) can be integrated in time for each grid point on S_a [†].

A relation between λ_1 and λ_4 on S_a is obtained as follows. With the no-slip boundary conditions applied, and approaching S_a from within the domain the continuity equation reduces to

$$\vec{\nabla} \cdot \vec{F}_1 \rightarrow \rho \frac{\partial V_n}{\partial n} = 0, \quad (5.42)$$

while the energy equation gives

$$\vec{\nabla} \cdot \vec{F}_4 \rightarrow \left(\rho \frac{a^2}{\gamma - 1} \frac{\partial V_n}{\partial n} \right) - \left(\tau_w \frac{\partial V_s}{\partial n} - \vec{\nabla} \cdot \vec{q} \right) = 0. \quad (5.43)$$

Combining the above equations gives on S_a

$$\frac{\partial V_n}{\partial n} = 0, \quad (5.44)$$

$$\tau_w \frac{\partial V_s}{\partial n} - \vec{\nabla} \cdot \vec{q} = 0. \quad (5.45)$$

For these equations two Lagrange multipliers are needed, introduced as b_1 and b_2 . Recalling that λ_1 and λ_4 are the Lagrange multipliers for the continuity and energy equations, respectively, one obtains

$$b_2 = \lambda_4,$$

because these multiply the same term in equations (5.45) and (5.43). Similarly, it also follows that

$$b_1 = \rho \lambda_1 = \rho \frac{a^2}{\gamma - 1} \lambda_4,$$

because these multiply the same term in equations (5.44), (5.42) and (5.43). Hence, the relation between λ_1 and λ_4 can be obtained as

$$\lambda_1 = \frac{a^2}{\gamma - 1} \lambda_4. \quad (5.46)$$

[†]However, numerical observations suggest that the viscous terms in equation (5.41) do not imply any significant effect on the computed gradient, such that it may be dropped.

5.5.3 Discretization of the Gradient Equations

Equation (5.38) is evaluated based on the discrete approach described in Section (4.4.2).

The integration in the s - and η -directions are performed using the trapezoidal rule. The factor η_∞ corresponds to the grid point index by $j = (3/8)j_{max}$, with $j = 0$ and $j = j_{max}$ being the airfoil surface and outer boundary, respectively. Numerical experiments suggest that taking $j > (3/8)j_{max}$ does not imply any significant difference in the computed gradients.

5.6 Test Cases for Single-Point Design

In order to make a comparative study, the specification of the test cases follows that given in Section 4.9. Four test cases are considered designated as Case N-1, N-2, N-3, and N-4. The following conditions are common in all cases:

- The airfoil is defined by the parameterization scheme described in Section 4.5 with 11 design variables.
- The aerodynamic scaling of the design variables described in Section 4.6 is applied.
- The Navier-Stokes (N-S) computations are performed by HI-TASK on a 256×64 grid. Figure 5.1 shows a typical grid produced by HI-TASK.
- The optimization procedure employs the FSQP routine.
- The gradients are computed based on the discrete formulation of the variations of geometric properties (i.e., the *discrete approach* described in Section 4.4.2).

5.6.1 Description and Results of Case N-1

Case N-1 is a reconstruction type inverse problem. The design point is specified by

$$M = 0.72, \quad C_l = 0.5, \quad Re = 6.5 \times 10^6.$$

The target C_p is obtained from an N-S flow analysis of a best-fit of the RAE 2822 airfoil. The target C_p distribution is defined on the airfoil chord with proper distinction between the lower and upper surface of the airfoil. The NACA 0012 airfoil is used as the initial airfoil geometry. The reason for selecting $C_l = 0.5$, instead of $C_l = 1$ like in the Euler case E-1, is that for $C_l = 1$ there is a strong flow separation over the initial geometry which causes failure of the flow solver. The objective functional is formulated in the same way as in Case E-1, which gives

$$\begin{aligned} \frac{\partial \psi}{\partial p} &= 2(C_p - C_{p,t})|n_y|, \\ \frac{\partial \psi}{\partial \tau_w} &= 0. \end{aligned}$$

These are substituted into the adjoint boundary conditions (5.31) and (5.32). Then, the adjoint equation (5.30) is solved for λ , and the gradient is obtained as described in Section 5.4.

Like in the Euler case, the main purpose of selecting this test case is to have an (indirect) check on the accuracy level of the computed gradient, because the optimal solution is known beforehand. The engineering accuracy corresponds to $\mathcal{P} \leq 10^{-4}$.

Figure 5.2 shows the design iteration history. An empty circle, referred to as "Evaluated", represents one flow analysis. A crossed circle, referred to as "New iterate", indicates that the corresponding geometry and flow analysis are used by the optimization algorithm as a basis for finding the next iterate. The process from one crossed circle to another represents a line search.

As can be seen, the engineering accuracy has been achieved at the 16th iteration. This indicates that the adjoint formulation and gradients are correct. The corresponding C_p distributions and airfoil geometries are shown in Figures 5.3, which demonstrate that the target geometry has been reconstructed. The rate of convergence is faster than that of the Euler case E-1. This is probably due to the shockless target C_p distribution of the N-S case.

5.6.2 Description and Results of Case N-2

The specification of case N-2 is the same as the Euler case E-2. This concerns a (pressure) drag reduction problem with geometric constraints for

$$M = 0.72, \quad C_l = 1., \quad \text{Re} = 6.5 \times 10^6.$$

The optimization starts with an initial airfoil specified as a best-fit of the RAE 2822 airfoil. Geometric constraints are imposed on the trailing edge included angle and cross-sectional area. These are limited to be not less than their initial values (given by the initial airfoil). The optimization begins with all constraints active.

Initially, Case N-2 was defined as a minimization of the total drag (pressure drag + skin friction drag):

$$C_d = - \int_{S_a} C_p (n_x \cos \alpha + n_y \sin \alpha) dS + \int_{S_a} \tau_w (n_y \cos \alpha - n_x \sin \alpha) dS.$$

This gives

$$\begin{aligned} \frac{\partial \psi}{\partial p} &= -2(n_x \cos \alpha + n_y \sin \alpha), \\ \frac{\partial \psi}{\partial \tau_w} &= 2(n_y \cos \alpha - n_x \sin \alpha), \end{aligned}$$

which are substituted into the adjoint boundary conditions (5.31) and (5.32) in order to solve the adjoint equation (5.30) for λ . Figure 5.4 shows the total drag reduction history. The high (off-line) values in the figure (corresponding to the iterations 8, 11, 16, 21, and 25) indicates failure of the grid generation (grid folding, etc.), while the flow solver failed at the iterations 20 and 24.

Case N-2 has also been investigated for the minimization of pressure drag. In this case, the objective functional to be minimized represents the pressure drag coefficient C_d ,

$$C_{d_p} = - \int_{S_a} C_p (n_x \cos \alpha + n_y \sin \alpha) dS.$$

This is formulated in the same way as in the Euler case E-2, which gives

$$\begin{aligned} \frac{\partial \psi}{\partial p} &= -2(n_x \cos \alpha + n_y \sin \alpha), \\ \frac{\partial \psi}{\partial \tau_w} &= 0. \end{aligned}$$

These are again substituted into the adjoint boundary conditions (5.31) and (5.32).

Figure 5.5 shows the pressure drag reduction history. The figure also shows the corresponding reduction of the total drag, which is slightly less than the reduction in pressure drag due to a slight increase in the skin friction drag (the difference between the two lines shown in the figure). Figure 5.4 shows that the total drag minimization (with the skin-friction drag included) could not achieve the total drag level given by the pressure drag minimization indicated in Figure 5.5. This is an unexpected result, which suggests that the gradient of the total drag functional is not fully correct. Further research is needed for investigating this problem.

The final iterate corresponds to the 6th flow analysis. The optimization process was stopped after the line search spent 5 flow analyses (the 7th to 11th) without finding a new (improved) iterate. Grid refinement studies for this case, like those applied to the Euler case, have been performed and summarized in Figure 5.6 and Table 5.1. The error-bandwidth is 5 counts. This means that the drag reduction of $(153 - 5) = 148$ counts can practically be considered to be significant.

The resulting C_p distribution and airfoil geometry are shown in Figure 5.7. As can be seen, the shock wave has been weakened significantly. The skin friction coefficient distribution is shown in Figure 5.8 with the convention that the tangential velocity follows the arc length measured clockwise from the lower trailing edge.

The states of geometric constraints corresponding to each iterate are shown in Figure 5.9. Both constraints were active up to the 5th analysis, after which the cross-sectional area moved away from its limiting boundary.

The airfoil shape resulting from case E-2 has also been analyzed using the Navier-Stokes solver. The viscous flow solution is compared with the result of case N-2 in Figure 5.10 and 5.11. The N-S design (from case N-2) appears to give a weaker shock wave (corresponding to 0.0138 of total C_d) than the one in the Euler design (corresponding to 0.0161 of total C_d), when both designs (i.e. airfoils) are analyzed in the viscous flow using the Navier-Stokes solver.

5.6.3 Description and Results of Case N-3

Case N-3 is a pressure drag reduction problem with aerodynamic and geometric constraints. The case specification is the same as Case N-2, except that

- A constraint on the pitching moment coefficient is introduced.
- A leading edge radius constraint is added into the set of geometric constraints.

These constraints must also not be less than their initial values. Again, the optimization starts with all constraints active.

The pressure drag reduction history is shown in Figure 5.12. The final iterate corresponds to the 9th flow analysis. The optimization process was stopped after the line search failed to find a new (improved) iterate with 5 flow analyses. Grid refinement studies for this case have been performed and summarized in Figure 5.13 and Table 5.2. The error-bandwidth is 7 counts. This means that the drag reduction of $(59 - 7) = 52$ counts can practically be considered to be significant.

The resulting C_p distribution and airfoil geometry are shown in Figure 5.14. As can be seen, the shock wave has again been weakened significantly.

The pitching moment constraint (Figure 5.12) is active during the whole iteration process. The states of geometric constraints are shown in Figure 5.16. The trailing edge included angle and cross-sectional area constraints are active with most iterates.

The airfoil shape resulting from case E-3 has also been analyzed using the Navier-Stokes solver. The C_p and C_f distributions of the viscous flow solutions of the airfoil of case E-3 are compared with the result of case N-3 in Figure 5.17 and 5.18, respectively. Surprisingly, the Euler design gives a lower total drag (0.0205) than that given by the N-S design (0.0232), but the C_p distribution of the Euler design indicates a stronger shock wave. Both Euler and N-S are feasible solutions, but the pitching moment constraint is not active for the Euler design ($C_m = -0.0796$).

The main difference between the two airfoils is in the trailing edge angle and leading edge radius (Figure 5.17). The N-S design has a smaller trailing edge angle and larger leading edge radius. Without knowing the topology of the design space, it is quite difficult to explain these differences. Nevertheless, one may try to find an explanation as follows,

- In the Navier-Stokes case, the optimization algorithm seems to avoid airfoils with a large trailing edge angle and a small leading edge radius. It is recognized that the boundary-layer tends to separate as the trailing edge angle gets larger (and/or the leading edge radius gets smaller for a finite angle of attack). The flow separation would increase the pressure drag. Because the flow separation is not recognized by the inviscid flow model, the Euler design would be less restricted in this respect. This is in agreement with the non-active constraints shown in Figures 4.23, and with the fact

that there is a larger flow separation with the Euler design when computed with the Navier-Stokes solver, as indicated in Figure 5.18.

- Both designs start with the same airfoil and share the same geometric constraint boundaries, but the aerodynamic constraint boundary and objective contours are different because of the different flow models. Considering this, in the Navier-Stokes design space, the Euler design may belong to a local minimum which does not happen to be the same local minimum as for the Navier-Stokes design.

5.6.4 Description and Results of Case N-4

The specification of this case is the same as Case N-3, but for

$$M = 0.78, \quad C_l = 0.5, \quad Re = 6.5 \times 10^6.$$

The reason for specifying this case is to provide a basis for comparison with the results of multi-point aerodynamic design to be described in Chapter 6.

The pressure drag reduction is shown in Figure 5.19. The final iterate corresponds to the 24th flow analysis. The pitching moment constraint is active for most iterates including the final iterate. Grid refinement studies for this case have been performed and summarized in Figure 5.20 and Table 5.3. The error-bandwidth is 7 counts. This means that the drag reduction of $(80 - 9) = 71$ counts can practically be considered to be significant.

The resulting C_p distribution and airfoil geometry are shown in Figure 5.21. As can be seen, the shock wave has practically been eliminated from the flow solution.

Figure 5.23 shows the states of geometric constraints. The cross-sectional area is active with all iterates, the trailing edge angle is close to its bound, and the leading edge radius is totally non-active.

The result of this case is also compared with the viscous analysis of the Euler design obtained from case E-4 in Figure 5.24 and 5.25. The total drags of the Euler and Navier-Stokes design are 0.0112 and 0.0119, respectively. Both are feasible solutions, but again the pitching moment constraint is not active for the Euler design ($C_m = -0.0974$).

5.7 Concluding Remarks

The results presented in the preceding sections indicate that the present method represents a viable approach for solving single-point transonic aerodynamic design problems based on the compressible viscous flow model governed by the Reynolds-Averaged Navier-Stokes (RANS) equations.

Application of the present method to a reconstruction type inverse problem suggests that the adjoint and gradient formulation are correct and that the computed gradient is of sufficient accuracy to be used for general aerodynamic optimization problems.

A significant problem is identified during the investigation: the method failed to give a useful result in the case of total drag reduction. Also, the assumption used in formulating the gradient, that the streamlines across the boundary-layer are parallel to the airfoil surface, is obviously not valid for cases with extensive flow separations. Further investigation for the case of total drag reduction is desirable.

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	301	174	127
2	2	291	138	153
3	1	283	126	157
4	0	282	124	158
	Error: (2)-(4)	9	14	-5

Table 5.1: Grid Refinement Studies for Drag of Case N-2

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	301	284	17
2	2	291	232	59
3	1	283	220	63
4	0	282	216	68
	Error: (2)-(4)	9	16	-7

Table 5.2: Grid Refinement Studies for Drag of Case N-3

No.	Mesh size (h)	Initial (counts)	Final (counts)	Initial - Final (counts)
1	4	212	153	59
2	2	199	119	80
3	1	197	110	87
4	0	196	107	89
	Error: (2)-(4)	3	12	-9

Table 5.3: Grid Refinement Studies for Drag of Case N-4

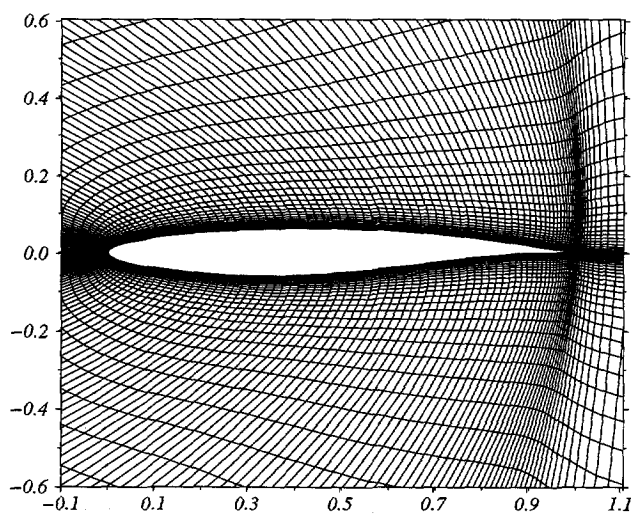


Figure 5.1: Typical 256×64 Navier-Stokes grid.

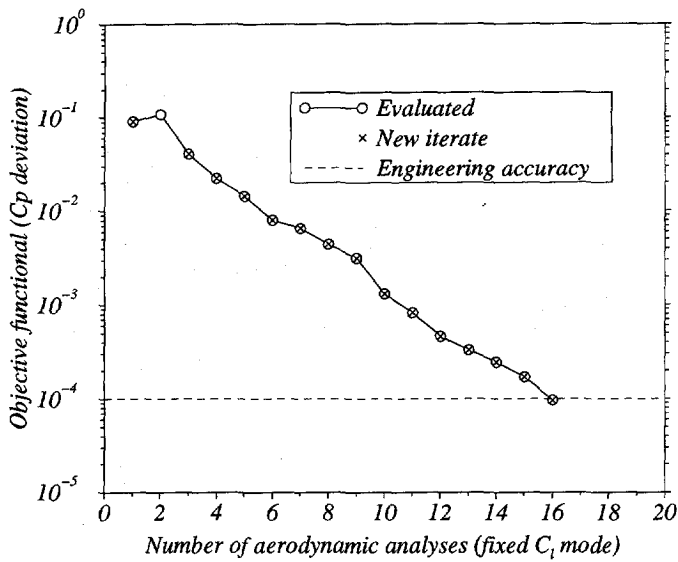


Figure 5.2: Objective for Case N-1 (*Transonic reconstruction inverse design. Flow model: Navier-Stokes. Design point: $M = 0.72$, $C_l = 0.5$, $Re = 6.5 \times 10^6$. Initial airfoil: NACA 0012. Target airfoil: RAE2822*).

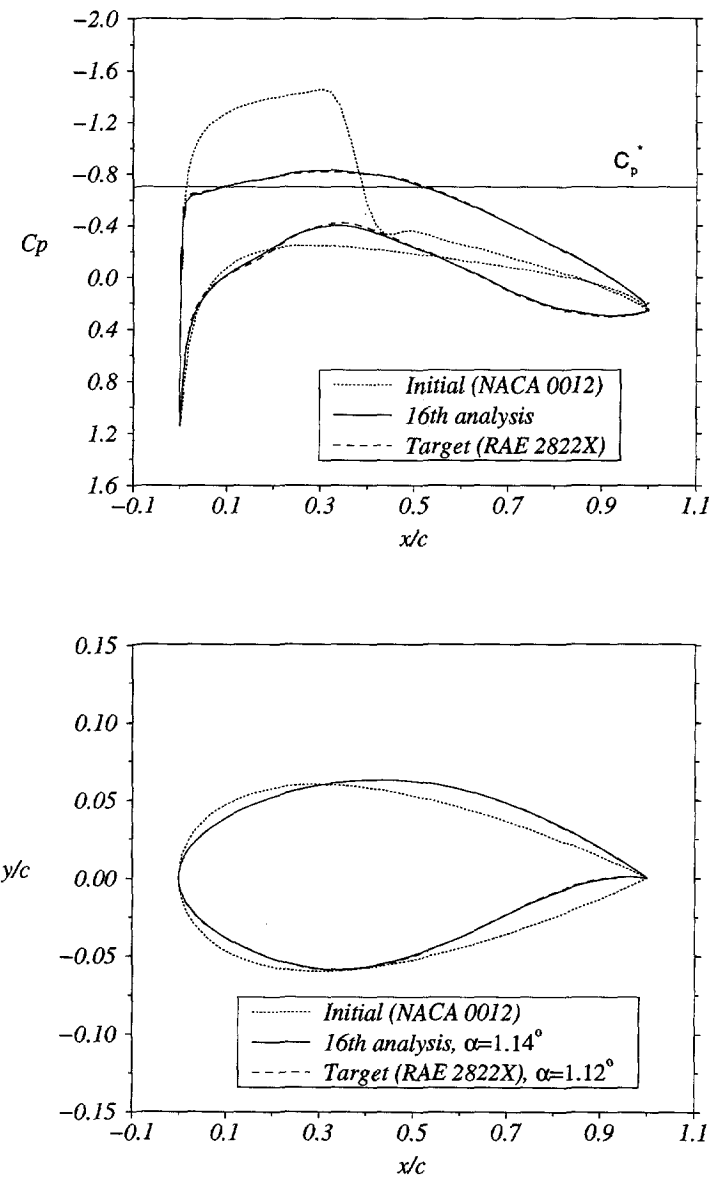


Figure 5.3: C_p distribution and airfoil geometry for Case N-1 (*Transonic reconstruction inverse design*. Flow model: Navier-Stokes. Design point: $M = 0.72$, $C_l = 0.5$, $Re = 6.5 \times 10^6$. Initial airfoil: NACA 0012. Target airfoil: RAE2822).

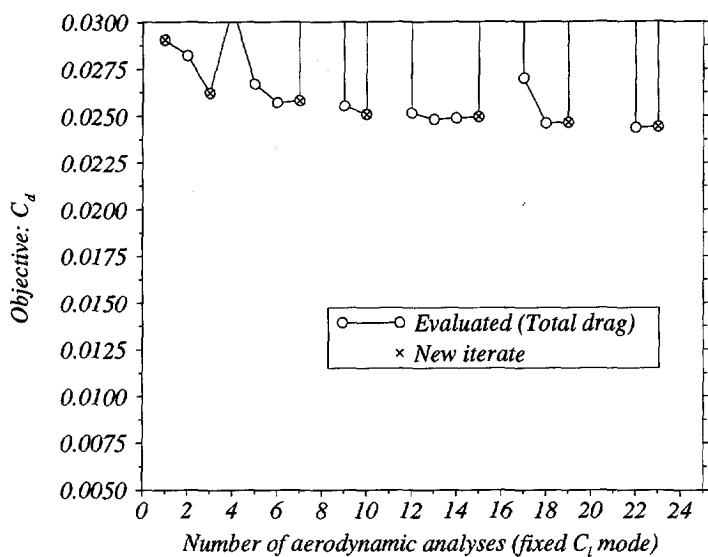


Figure 5.4: Total drag coefficients (*Flow model*: Navier-Stokes. *Design point*: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. *Initial airfoil*: RAE2822. *Constraints*: trailing edge included angle and cross sectional area).

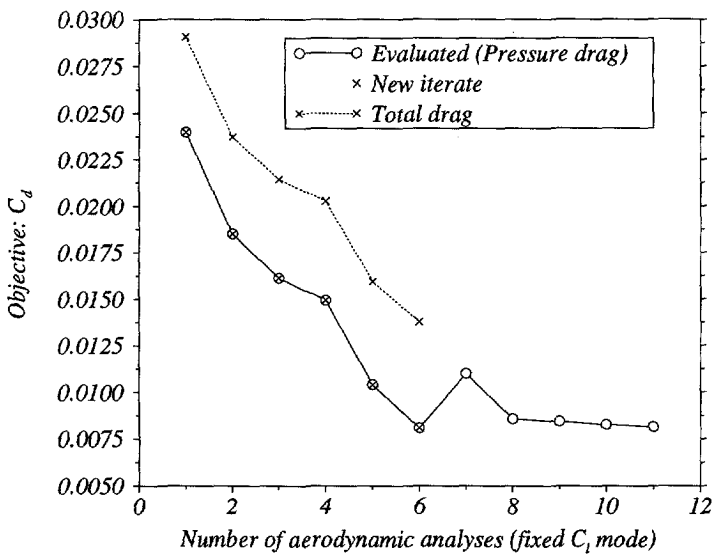


Figure 5.5: Pressure drag coefficients for Case N-2 (*Transonic constrained pressure drag reduction*). *Flow model*: Navier-Stokes. *Design point*: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. *Initial airfoil*: RAE2822. *Constraints*: trailing edge included angle and cross sectional area).

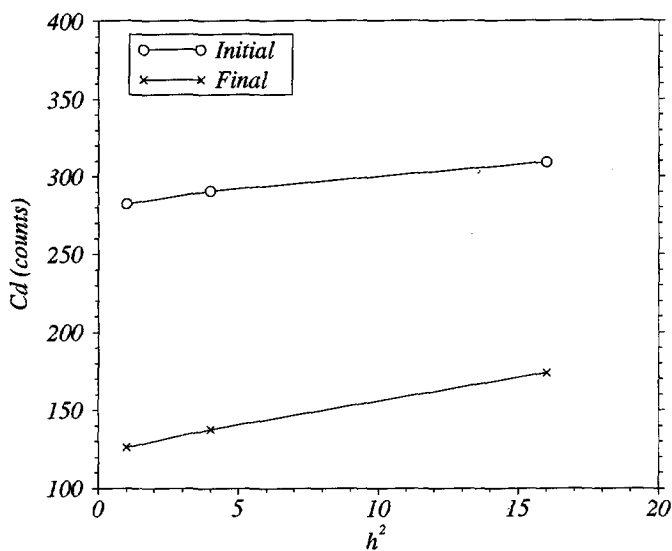


Figure 5.6: Grid refinement studies for the Total Drag of Case N-2

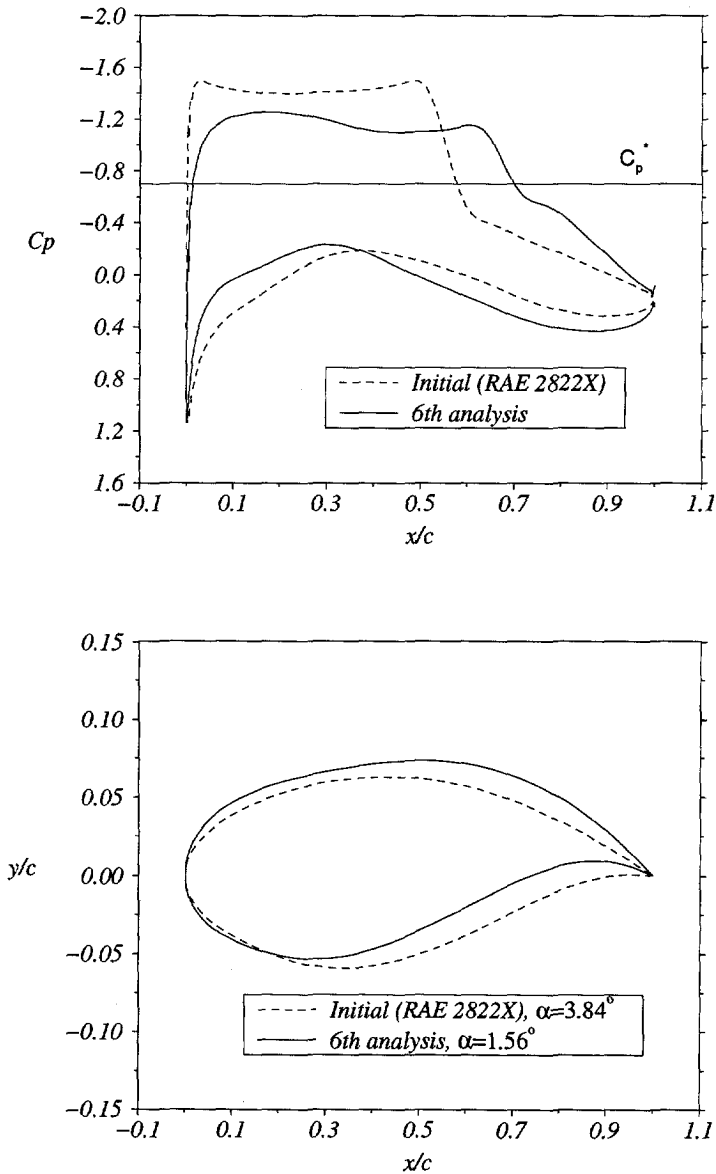


Figure 5.7: C_p distribution and airfoil geometry for Case N-2 (Transonic constrained pressure drag reduction). Flow model: Navier-Stokes. Design point: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: trailing edge included angle and cross sectional area).

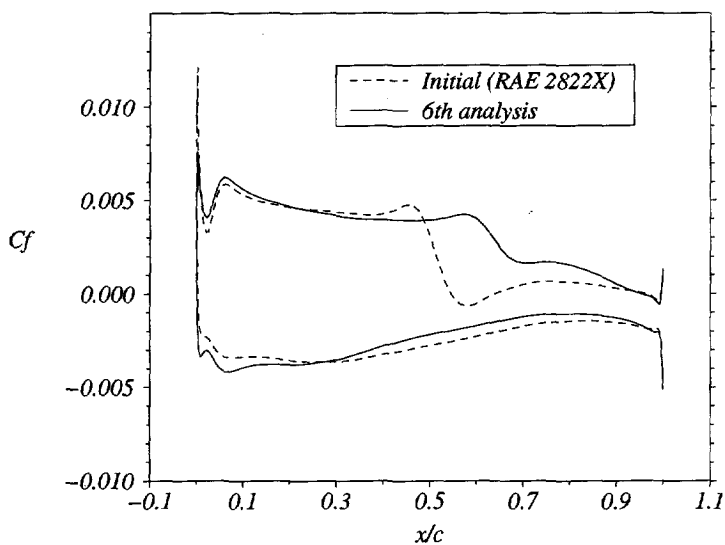


Figure 5.8: C_f distribution for Case N-2 (Transonic constrained pressure drag reduction. Flow model: Navier-Stokes. Design point: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: trailing edge included angle and cross sectional area).

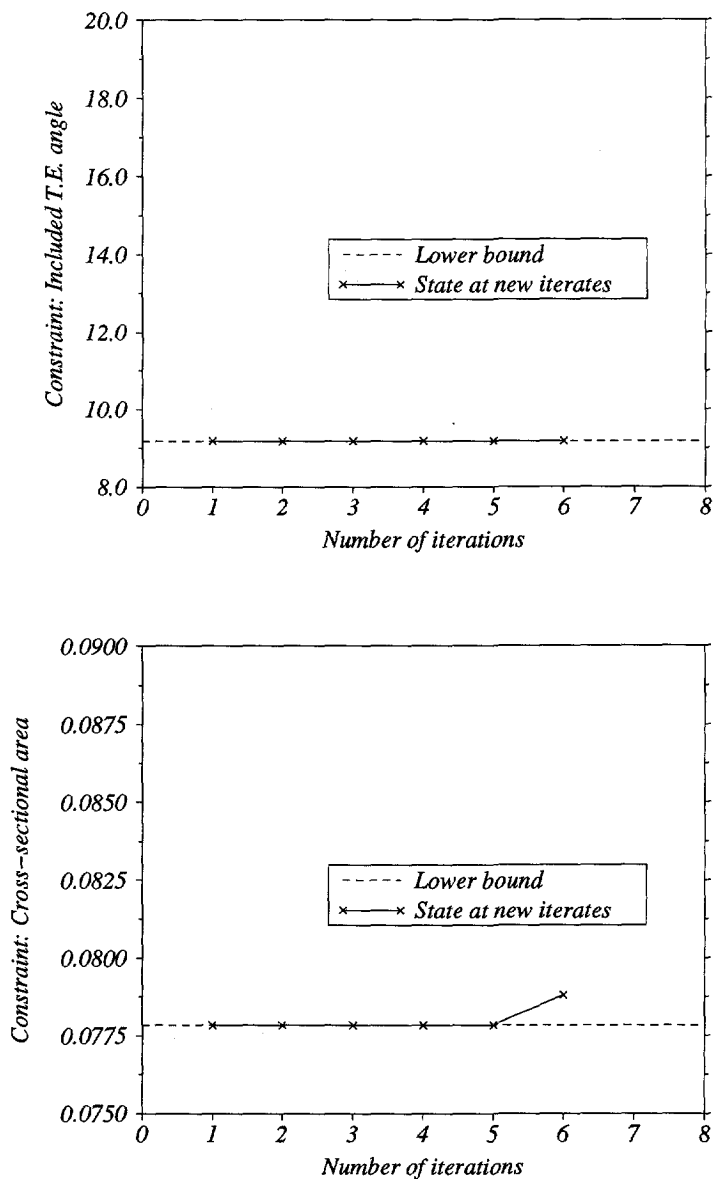


Figure 5.9: State of geometric constraints for Case N-2.

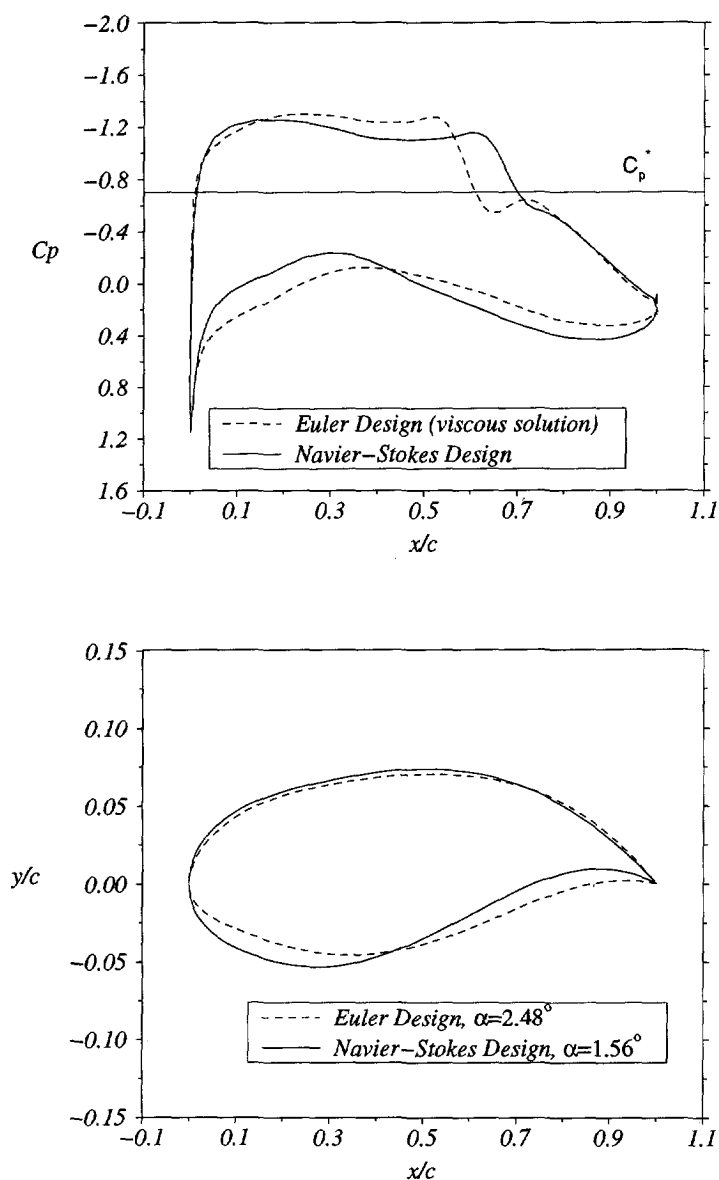


Figure 5.10: Comparison of C_p distribution and airfoil geometry of resulting from Case E-2 and N-2

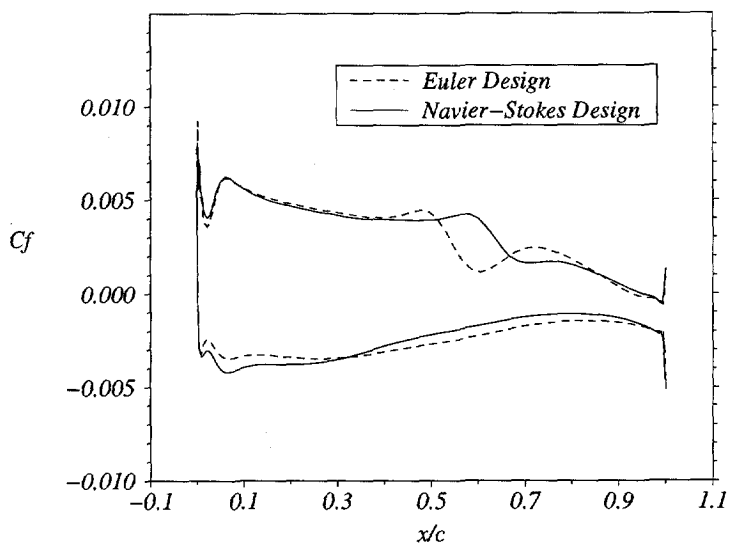


Figure 5.11: Comparison of C_f distribution resulting from Case E-2 and N-2

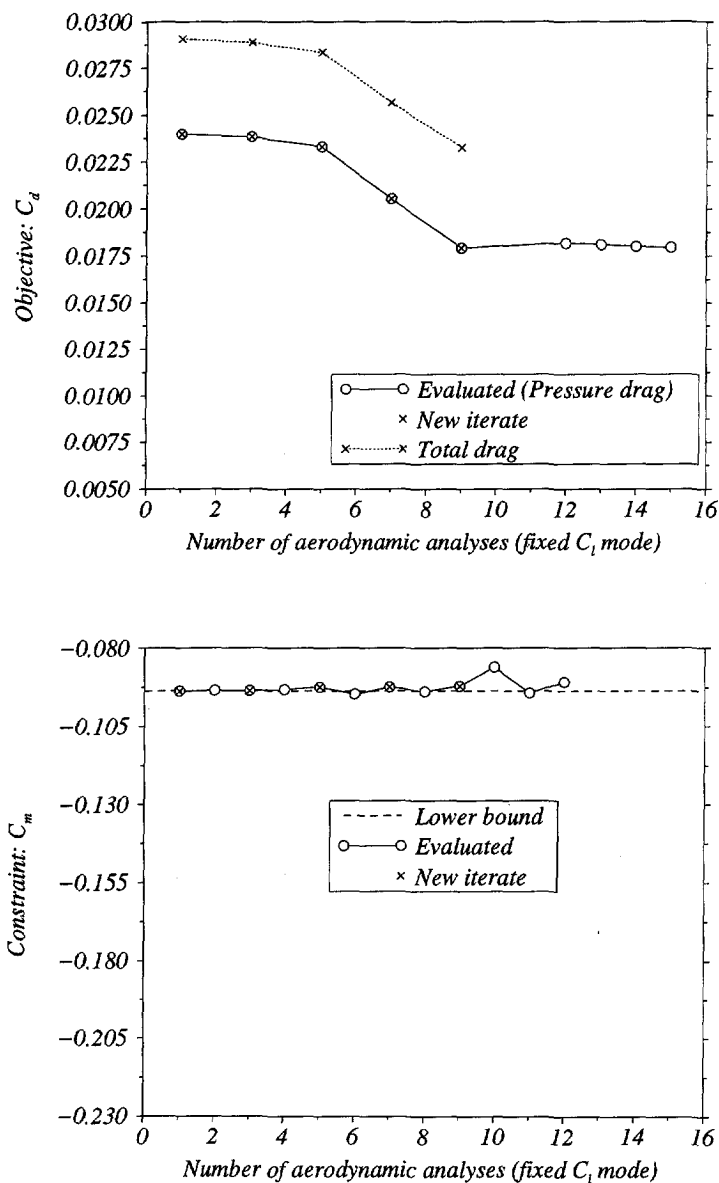


Figure 5.12: Pressure drag and pitching moment coefficients for Case N-3 (*Transonic constrained pressure drag reduction. Flow model: the Navier-Stokes. Design point: $M = 0.72$, $C_l = 1.$, $Re \approx 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area*).

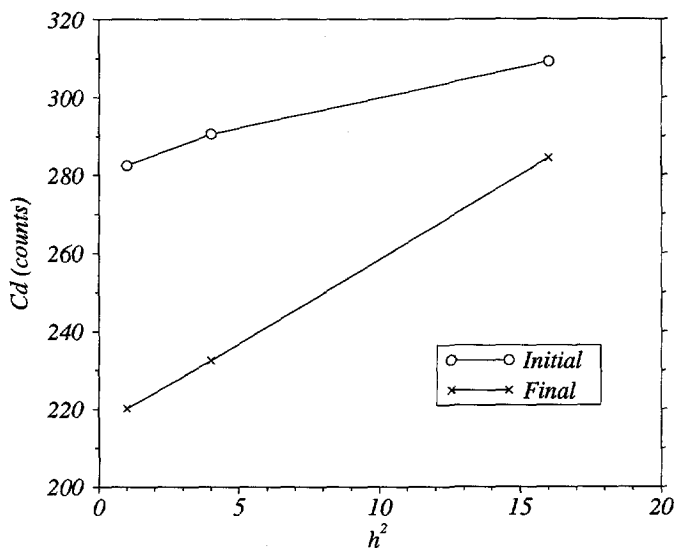


Figure 5.13: Grid refinement studies for the Total Drag of Case N-3

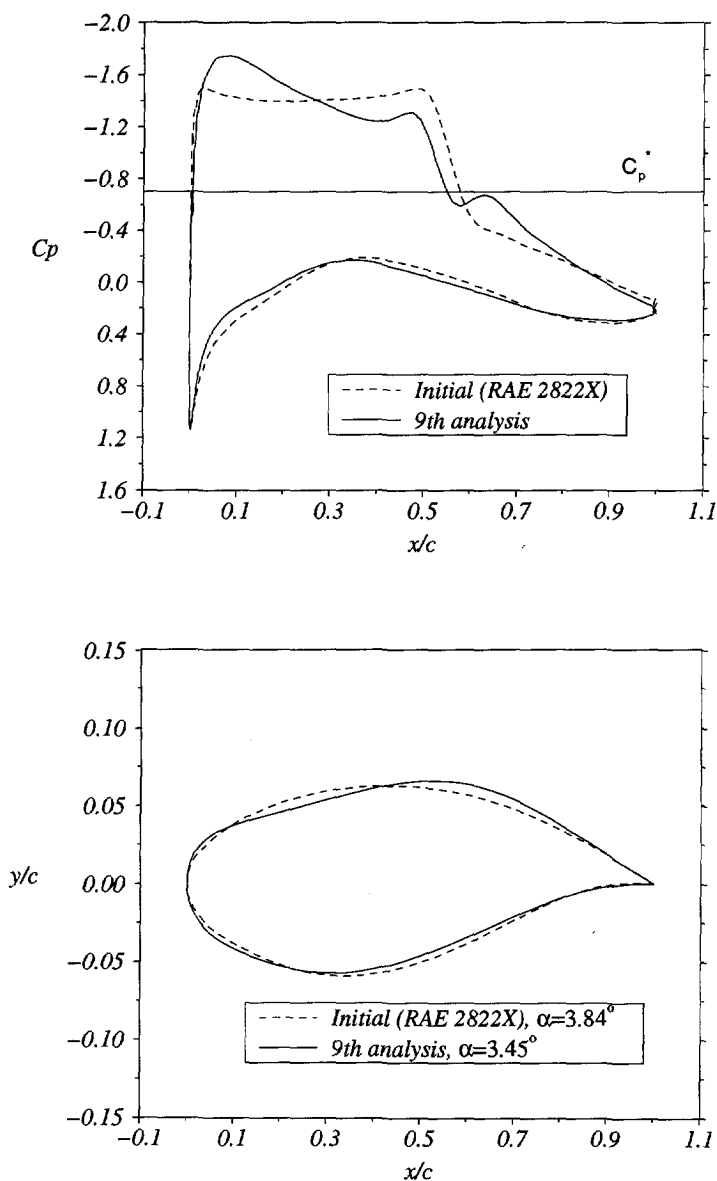


Figure 5.14: C_p distribution and airfoil geometry for Case N-3 (Transonic constrained pressure drag reduction. *Flow model*: Navier-Stokes. *Design point*: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. *Initial airfoil*: RAE2822. *Constraints*: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

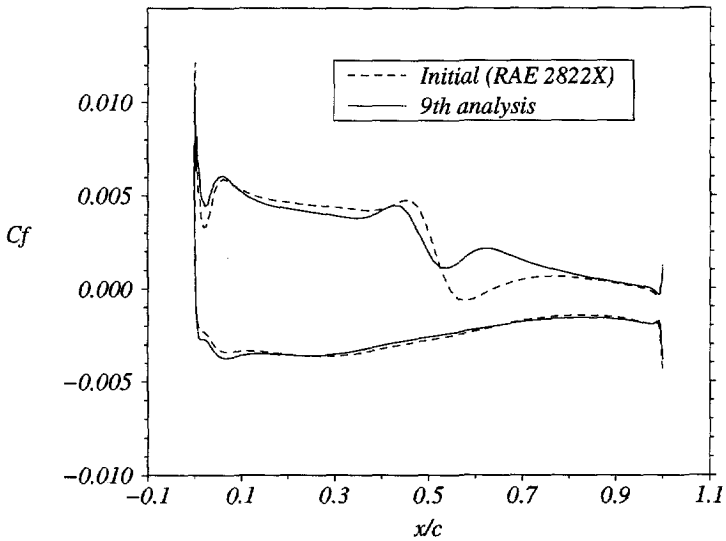


Figure 5.15: C_f distribution for Case N-3 (Transonic constrained pressure drag reduction). Flow model: Navier-Stokes. Design point: $M = 0.72$, $C_l = 1.$, $Re = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

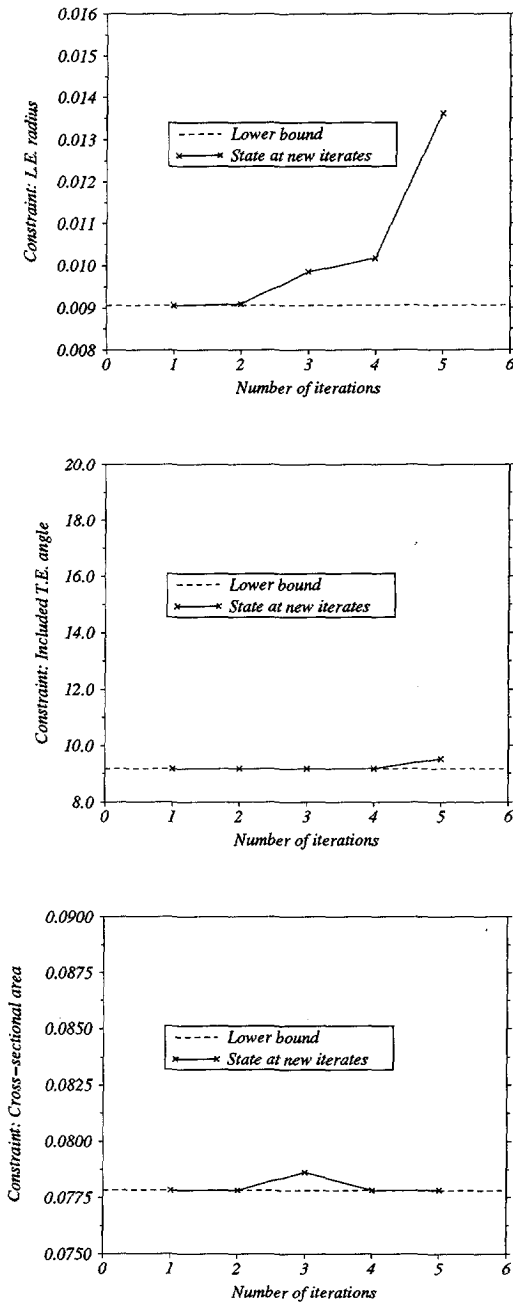


Figure 5.16: State of geometric constraints for Case N-3.

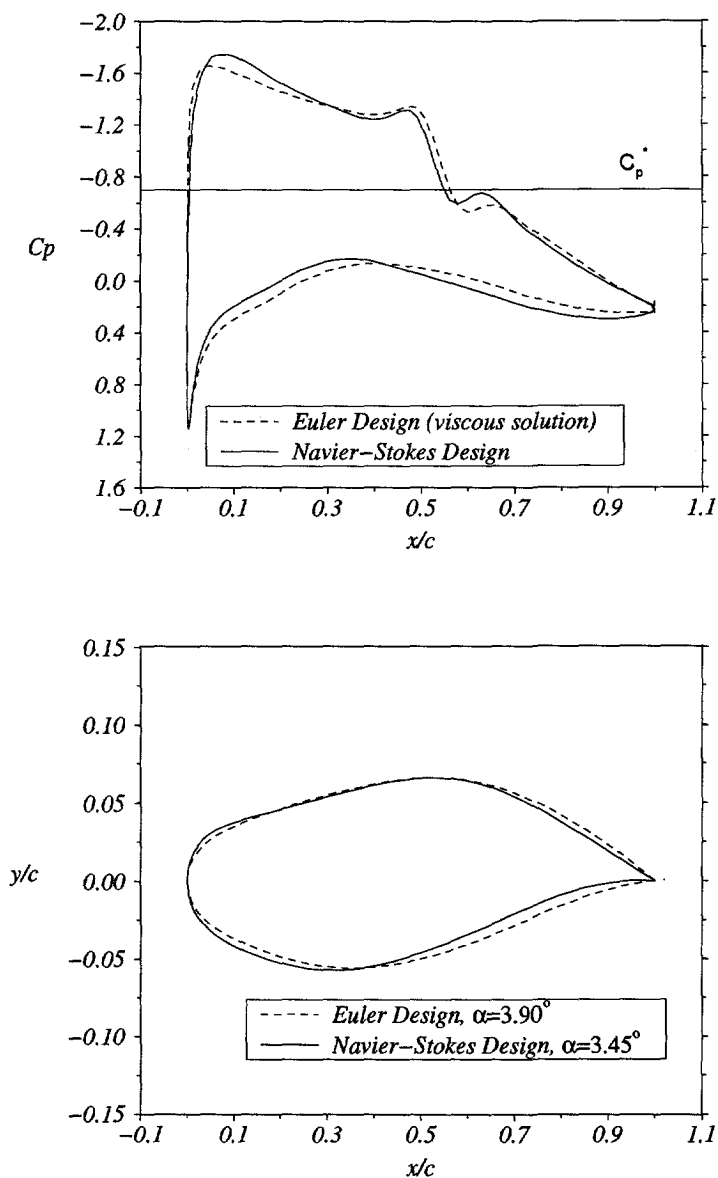


Figure 5.17: Comparison of C_p distribution and airfoil geometry of resulting from Case E-3 and N-3

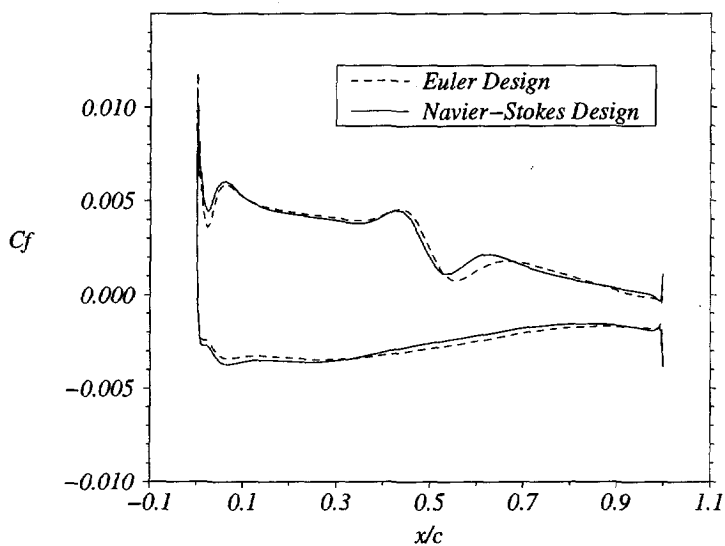


Figure 5.18: Comparison of C_f distribution resulting from Case E-3 and N-3

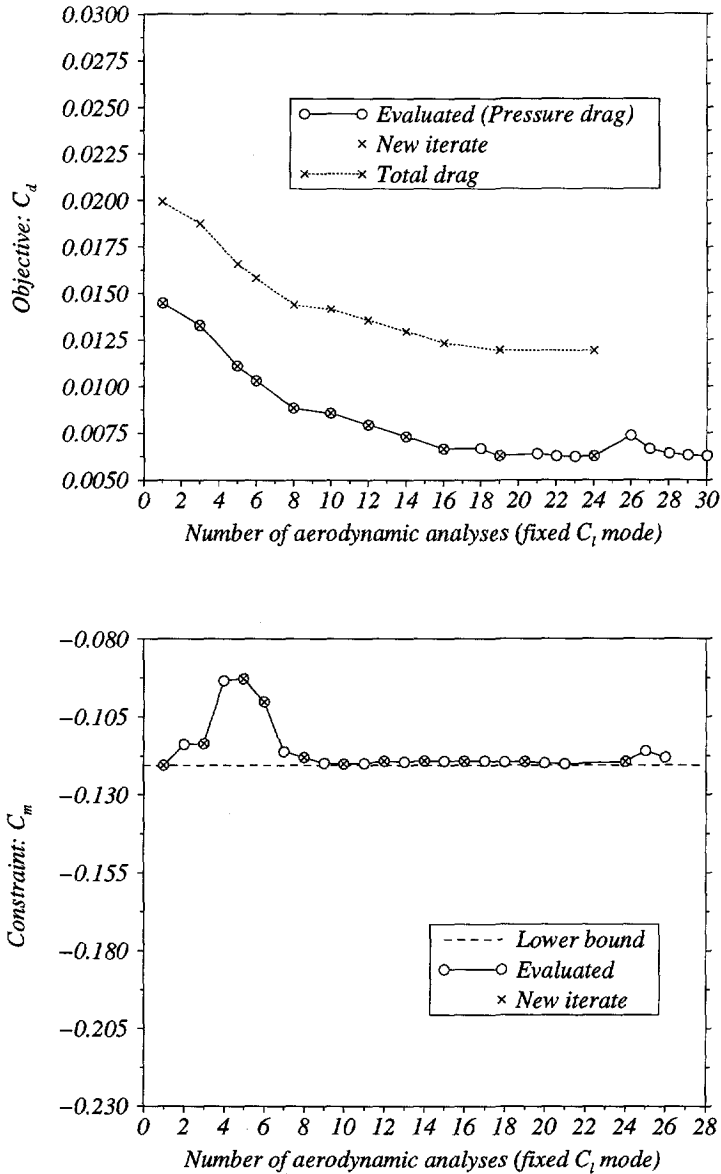


Figure 5.19: Pressure drag and pitching moment coefficients for Case N-4 (Transonic constrained pressure drag reduction. Flow model: the Navier-Stokes. Design point: $M = 0.78$, $C_l = 0.5$, $Re = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

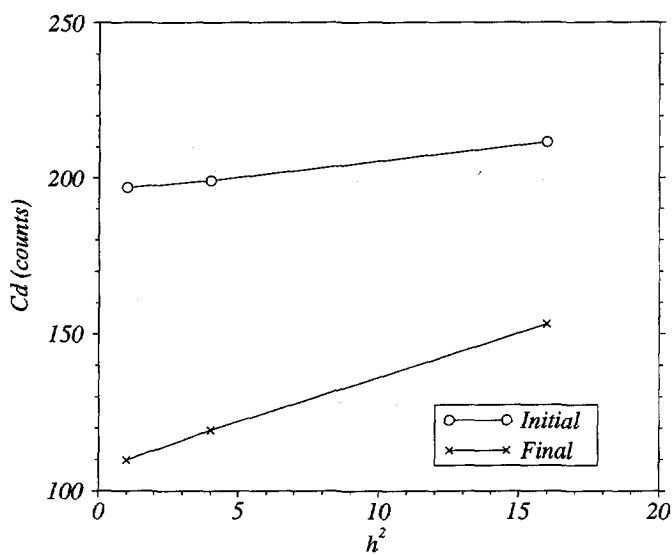


Figure 5.20: Grid refinement studies for the Total Drag of Case N-4

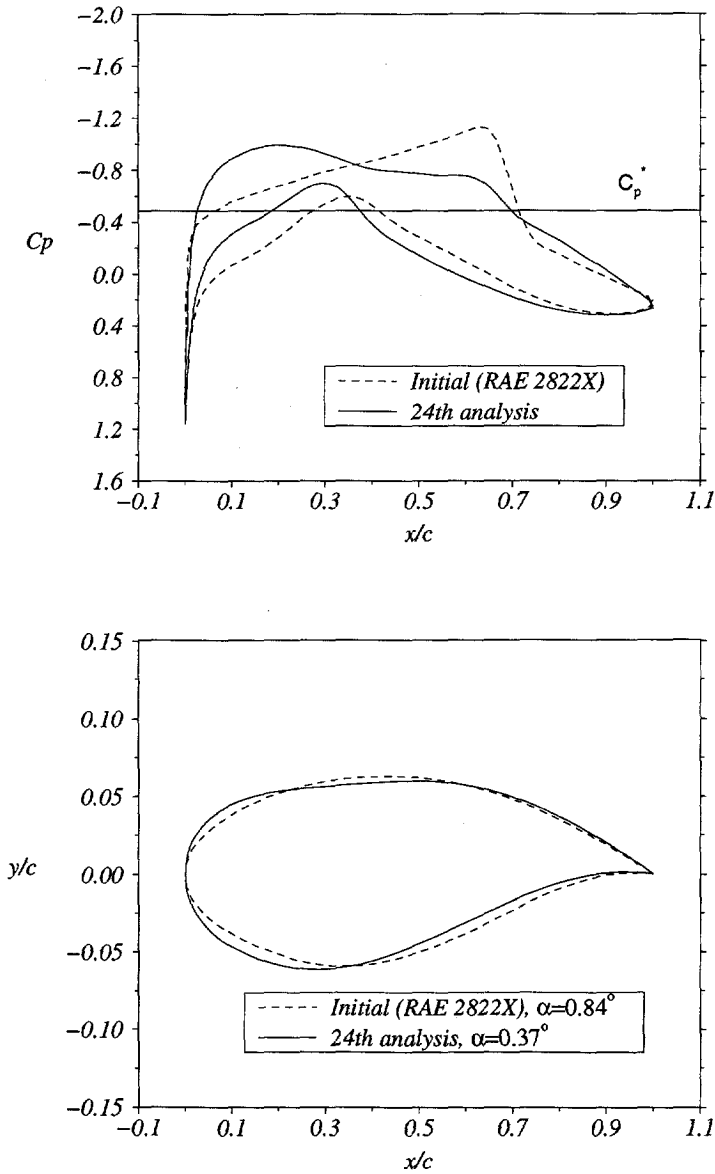


Figure 5.21: C_p distribution and airfoil geometry for Case N-4 (Transonic constrained pressure drag reduction. *Flow model:* Navier-Stokes. *Design point:* $M = 0.78$, $C_l = 0.5$, $Re = 6.5 \times 10^6$. *Initial airfoil:* RAE2822. *Constraints:* pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

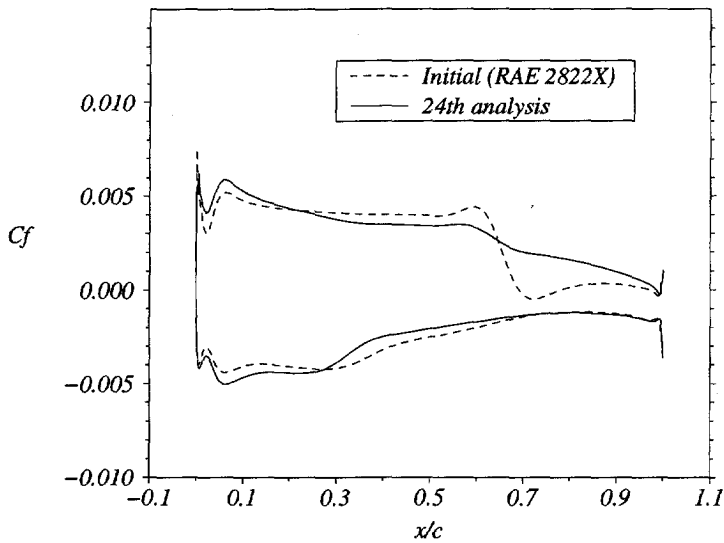


Figure 5.22: C_f distribution for Case N-4 (Transonic constrained pressure drag reduction. *Flow model*: Navier-Stokes. *Design point*: $M = 0.78$, $C_l = 0.5$, $Re = 6.5 \times 10^6$. *Initial airfoil*: RAE2822. *Constraints*: pitching moment, leading edge radius, trailing edge included angle, and cross sectional area).

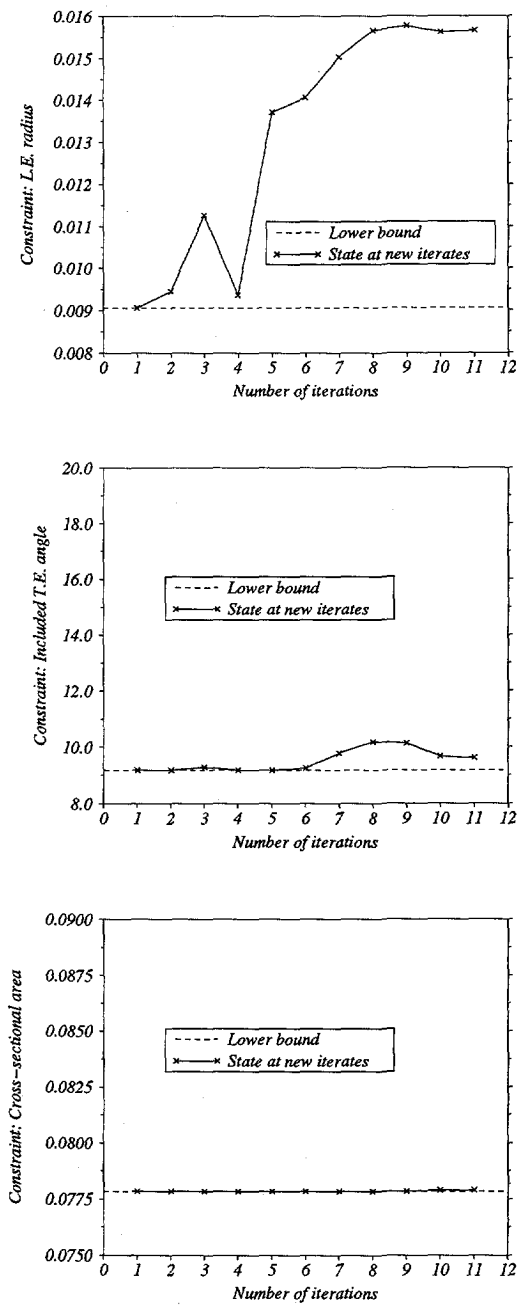


Figure 5.23: State of geometric constraints for Case N-4.

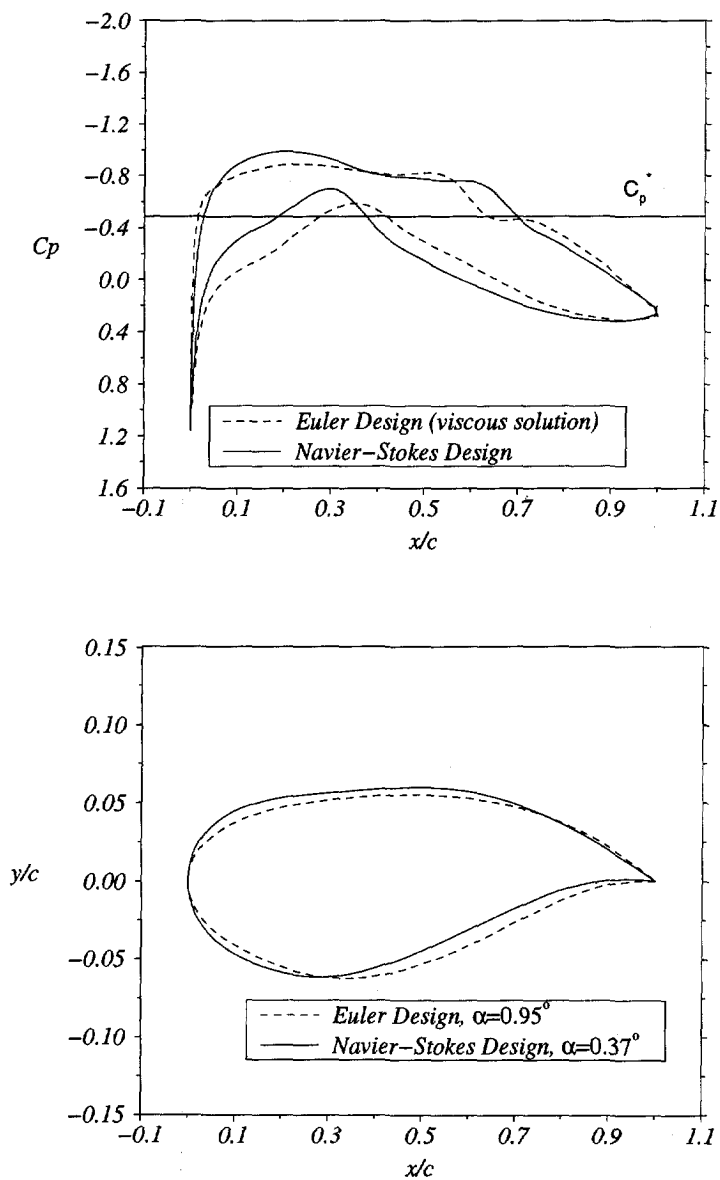


Figure 5.24: Comparison of C_p distribution and airfoil geometry of resulting from Case E-4 and N-4

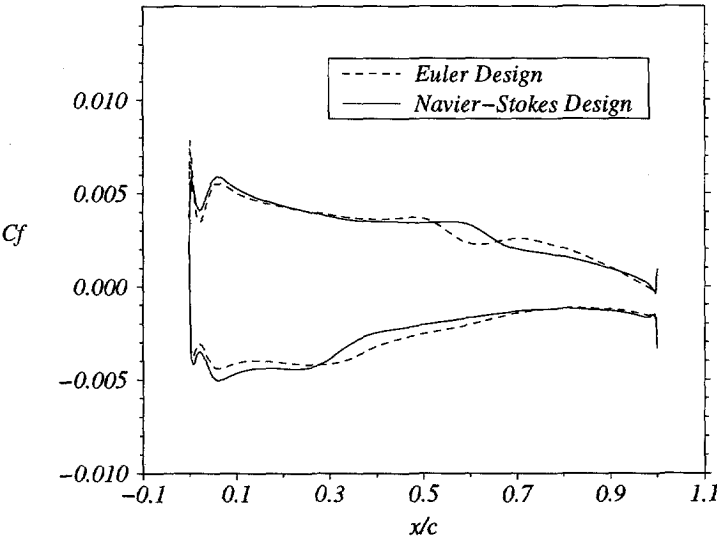


Figure 5.25: Comparison of C_f distribution resulting from Case E-4 and N-4

Chapter 6

Multi-Point Aerodynamic Design

6.1 Introduction

The preceding chapters deal with single-point aerodynamic design by means of optimization. However, as already discussed in Section 2.2, the optimization method can also be applied in multi-point design problems. This is done by formulating a multi-point design problem as a multi-objective optimization problem. The resulting vector of objectives can be scalarized into a single-objective. In this way, the multi-objective problem is transformed into a single-objective problem. The latter is then solved by the method described in the previous chapters.

In this chapter, two methods of scalarization are discussed, namely: (i) scalarization by summing the objectives using weight factors, and (ii) scalarization utilizing fuzzy-set theory. This chapter is organized as follows. First, in Section 6.2 a general statement of the multi-point design problem is given. This appears basically as a (constrained) multi-objective optimization problem formulation. The first method of scalarization is explained in Section 6.3. Fuzzy-set theory and its application to optimization is described in Section 6.4. To investigate the applicability of both approaches, test cases are discussed in Section 6.5. Finally, concluding remarks are given in Section 6.6.

6.2 General Statement of the Multi-objective Optimization Problem

It is assumed that there are m design points to be considered. A design point k is identified by the vector of design point parameters \mathbf{z}_k (e.g., Mach numbers, design lift coefficients, etc.). With each $k = 1, \dots, m$, an aerodynamic objective, \mathcal{P}_k , and an aerodynamic constraint \mathcal{A}_k is associated[†]. A vector of geometric constraints \mathbf{G} is also supposed to be imposed. The multi-objective optimization problem is

[†]For brevity, it is assumed that there is only one aerodynamic constraint in each design point. However, multiple constraints are equally possible.

stated as follows,

$$\begin{aligned}
 &\text{For } k = 1 \dots m: \\
 &\quad \text{Minimize} \quad \mathcal{P}_k(\mathbf{Q}_k, \boldsymbol{\theta}, \alpha_k, \mathbf{z}_k) \\
 &\quad \text{subject to:} \\
 &\qquad \mathcal{A}_k(\mathbf{Q}_k, \boldsymbol{\theta}, \alpha_k, \mathbf{z}_k) \leq 0, \\
 &\qquad \mathbf{G}(\boldsymbol{\theta}) \leq 0,
 \end{aligned} \tag{6.1}$$

where \mathbf{Q}_k is the vector of flow variables associated with a design point k , and $\boldsymbol{\theta}$ and α are the vector of design variables. The optimal values of $\boldsymbol{\theta}$ and α_k are to be determined, which corresponds to an optimal geometry of the airfoil shape.

6.3 Scalarization by the Sum of Weighted Objectives

In this approach the objectives are added together to form a single-objective by means of the weighting factors w_k , $k = 1, \dots, m$. In this way, the multi-objective problem (6.1) is transformed into

$$\begin{aligned}
 &\text{Minimize} \quad \sum_{k=1}^m w_k \mathcal{P}_k \\
 &\quad \text{subject to:} \\
 &\qquad \mathcal{A}_k \leq 0, \quad k = 1, \dots, m \\
 &\qquad \mathbf{G} \leq 0,
 \end{aligned}$$

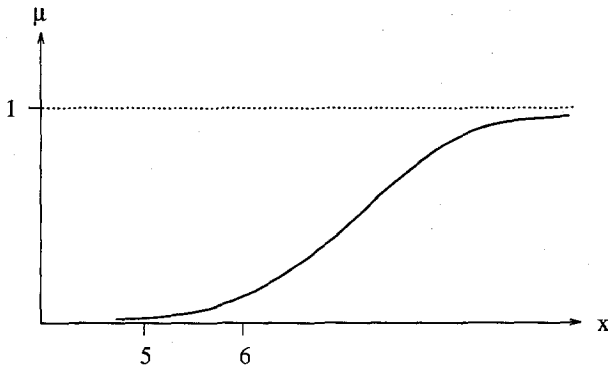
This approach might seem to be simple to apply, but it has the disadvantage that it may not be known a priori what combination of weight factors is appropriate for the design problem at hand. For instance, as already indicated in Section 2.2, equal weight factors do not necessarily lead to with a "balanced" design solution. A trial-and-error process is likely needed in specifying the weight factors.

6.4 Scalarization by means of Fuzzy Set Theory

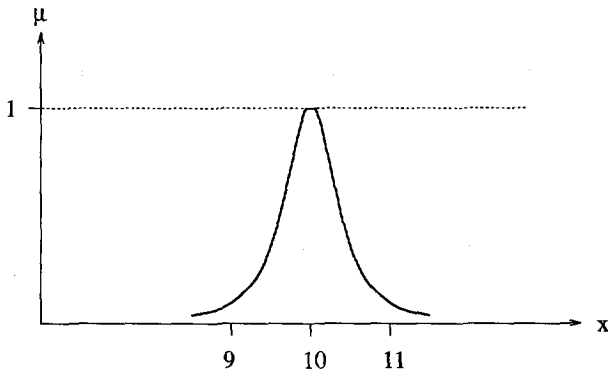
Fuzzy set theory was initially proposed by Zadeh [68] who defined it as follows,

Definition 1 Let X be a space of points (objects), with a generic element of X denoted by x . Thus, $X = x$. A fuzzy set (class) A in X is characterized by a membership (characteristic) function $\mu_A(x)$ which associates with each point in X a real number in the interval $[0, 1]$, with the value of $\mu_A(x)$ at x representing the "grade of membership" of x in A .

For example, a fuzzy set of real numbers *greater than 5* can be characterized by a membership function as shown in Figure 6.1a. Another example (Figure 6.1b)



(a)



(b)

Figure 6.1: Examples of Fuzzy Sets

illustrates a membership function that characterizes a fuzzy set of real numbers close to 10. Since a membership function can take a value in the interval $[0, 1]$, it may be regarded as a generalization of the so-called table of truth, i.e., the table of truth can only take the value of either 0 (false) or 1 (true). The application of fuzzy set theory to optimization problems (commonly referred to as *fuzzy optimization*) needs a characterization of each of the objectives and constraints in terms of a membership function. Of importance in fuzzy optimization is the concept of the intersection of two or more fuzzy sets which is defined in Ref. [68] as follows:

Definition 2 The intersection of two fuzzy sets A and B with respective membership functions $\mu_A(x)$ and $\mu_B(x)$ is a fuzzy set C , written as $C = A \cap B$, whose membership function is related to those of A and B by

$$\mu_C(x) = \min[\mu_A(x), \mu_B(x)], \quad x \in X,$$

or, in abbreviated form

$$\mu_C = \mu_A \wedge \mu_B.$$

As an example, the following simple (fuzzy) optimization problem from Ref. [70] can be considered:

Objective: x should be substantially larger than 10
Constraint: x should be in the vicinity of 11

The objective is characterized by the membership function:

$$\mu_f(x) = \begin{cases} 0, & x < 10, \\ (1 + (x - 10)^{-2})^{-1}, & x \geq 10, \end{cases}$$

while the constraint is characterized by:

$$\mu_c(x) = ((1 + (x - 11)^4)^{-1}).$$

These are depicted in Figure 6.2. The (fuzzy) optimal solution, x_d , lies on a curve that belongs to μ_d defined by

$$\begin{aligned} \mu_d(x) &= \mu_f(x) \wedge \mu_c(x) \\ &= \min(\mu_f(x), \mu_c(x)). \end{aligned}$$

x_d takes a value that corresponds to the maximum of the intersection of the membership functions:

$$x_d = \max(\min(\mu_f(x), \mu_c(x))),$$

or, equivalently, if one prefers formulating it as a minimization problem:

$$x_d = \min(\max(-\mu_f(x), -\mu_c(x))).$$

This shows that a fuzzy optimization problem is equivalent with a min-max problem of the membership functions. The characterization by membership functions has the implication that the objectives and constraints become of the same nature. It implies also that there is no longer any difference in the treatment of a single-objective and a multi-objective optimization problem.

Extending the concept to general constrained multi-objective multivariate optimization problems is straightforward. This requires a "fuzzification" of the optimization problem statement. An objective, F , to be minimized, can be fuzzified by associating with it a monotonously descending membership function like those shown in Figure 6.3a, whereas a general constraint, C , with its respective lower and upper bounds, c_l and c_u , can be fuzzified by a membership function like that in Figure 6.3b. The thresholds, F_{min} and F_{max} , and the leeway δc , are reasonably specified values which, however, may be problem dependent.

The *steeper* part of the curves indicates *more tendency* towards/against the membership, or in other words, the *flatter* part indicates *indifference* regarding the grade of membership. In multi-objective optimization problems, this means

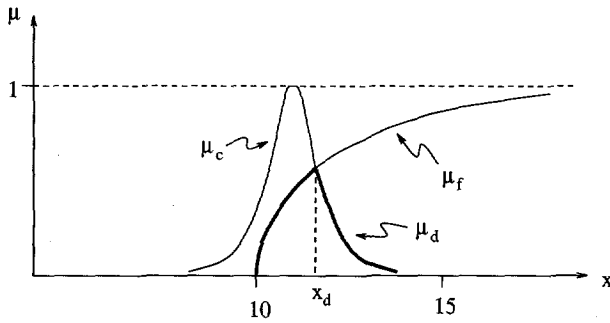


Figure 6.2: Example of A Fuzzy Optimization

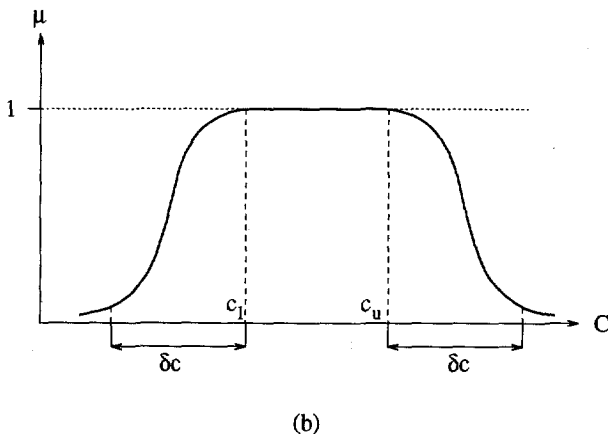
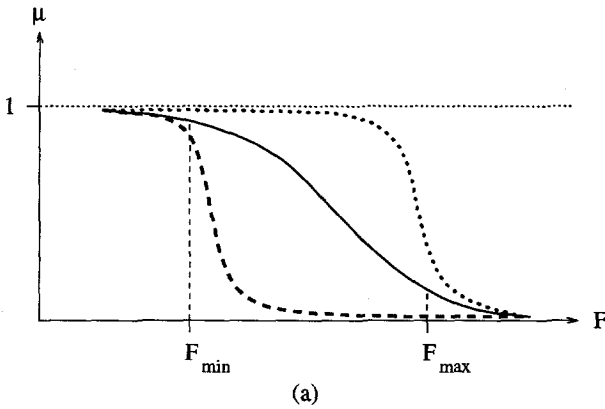


Figure 6.3: Fuzzification of Objectives (a) and Constraints (b)

that an objective value which corresponds to a higher membership function value can be compromised with respect to the other objectives.

Returning to the multi-objective optimization problem (6.1), a number of m membership functions must be defined for the objectives \mathcal{P}_k , $k = 1, \dots, m$. In addition to that, a number of $m + n$ membership functions need to be defined for the constraints \mathcal{A}_k and \mathbf{G} . Here, n is the dimension of \mathbf{G} . This results in N membership functions, where

$$N = 2m + n.$$

Since in fuzzy optimization, the optimal solution corresponds to the maximum of the intersection of the membership functions, the equivalent fuzzy form of the optimization problem of (6.1) is stated as

$$\text{Minimize} \quad \max(-\mu_1, \dots, -\mu_N). \quad (6.2)$$

Problem (6.2) can be solved by means of an optimization algorithm like FSQP (cf. Section 4.8). The required gradients of the membership functions with respect to the design variables $\boldsymbol{\theta}$ are obtained by the chain rule of differentiation. For example, the gradient of the i -th membership function μ_i associated with the k -th objective is obtained as

$$\frac{d\mu_i}{d\boldsymbol{\theta}} = \frac{\partial\mu_i}{\partial\mathcal{P}_k} \frac{d\mathcal{P}_k}{d\boldsymbol{\theta}},$$

where $\frac{d\mathcal{P}_k}{d\boldsymbol{\theta}}$ can be obtained by the variational method discussed in the preceding chapters.

The above discussion assumes that some degree of constraint violation is allowed, as indicated by the leeway in the membership functions. If the constraints must be satisfied strictly, then one should not characterize the constraints by the membership functions. Instead, they should be directly incorporated in the optimization. For example, in case of strict geometric constraints, only the objectives need to be characterized by the membership functions, and the fuzzy optimization problem should be formulated as follows,

$$\begin{aligned} &\text{Minimize} \quad \max(-\mu_1, \dots, -\mu_{2m}) \\ &\text{subject to:} \quad \mathbf{G} \leq 0. \end{aligned}$$

It should be noticed that the \max operator is now taken over $2m$ instead of N membership functions.

6.5 Test Cases for Multi-Point Design

Four test cases have been considered for constrained pressure drag reduction in two design points based on the Euler and Navier-Stokes equations. The design points are taken as a combination of the Mach numbers and design lift coefficients defined in case E-3 (case N-3) and case E-4 (case N-4) in Chapter 4 (Chapter 5).

- Design Point 1 (DP-1): $C_l = 1.$, $M_\infty = 0.72$, $Re = 6.5 \times 10^6$.
- Design point 2 (DP-2): $C_l = 0.5$, $M_\infty = 0.78$, $Re = 6.5 \times 10^6$.

The Reynolds number Re is not applicable in the Euler case. The objective is to reduce the pressure drag while satisfying the pitching moment constraint in both design points. The same geometric constraints are imposed, i.e. the leading-edge radius, the trailing-edge angle, and the cross-sectional area of the airfoil. The optimization starts with an initial airfoil specified as a best-fit of the RAE 2822 airfoil. The constraints are formulated in the same way as in the single-point design cases. The optimization begins with all constraints active.

The test cases are identified as Case ME-1, ME-2, MN-1, and MN-2. The letters E and N stand for Euler and Navier-Stokes, respectively. The digit 1 refers to the method of sum of weighted objectives, with the weight factors specified as

$$w_1 = w_2 = 0.5.$$

The digit 2 refers to the fuzzy optimization method, where the geometric constraints are assumed to be strict, while the pressure drag objectives and pitching moment constraints are characterized by a membership function of the type depicted in Figure 6.3. A suitable form is

$$\mu = \frac{\mu_{max}}{2} \left(1 + \frac{1}{\arctan(\pi/2)} \arctan(\arctan(2hf^*)) \right).$$

μ_{max} is a scaling factor (not to be confused with \mathbf{d} in Section 4.6) which has the effect of reducing the magnitude of the gradient of μ and, thus, avoiding too large shape modifications. The factor h determines the type of membership function:

$$h = \begin{cases} -1, & \mu \text{ descends monotonously with } f, \\ 1, & \mu \text{ ascends monotonously with } f. \end{cases}$$

A descending form of μ is applied to an objective to be minimized or an inequality constraint with an upper-bound. An ascending form of μ is applied to an objective to be maximized or an inequality constraint with a lower-bound. Figure (6.4) shows a typical membership function μ which descends monotonously. The factor f^* is defined as

$$f^* = \frac{f - \frac{1}{2}(f_u + f_l)}{\frac{1}{2}(f_u - f_l)},$$

with f the current value of objective or constraint. The upper, f_u , and lower, f_l , values of f are defined in terms of the percentage (or fraction) of the initial value of f .

The following choices apply in all cases:

- Scaling factor $\mu_{max} = 0.1$.
- For the objective C_d to be minimized (applied to both design points):

$$\begin{aligned} h &= -1, \\ f_l &= 0.25C_{d,initial}, \\ f_u &= C_{d,initial}. \end{aligned}$$

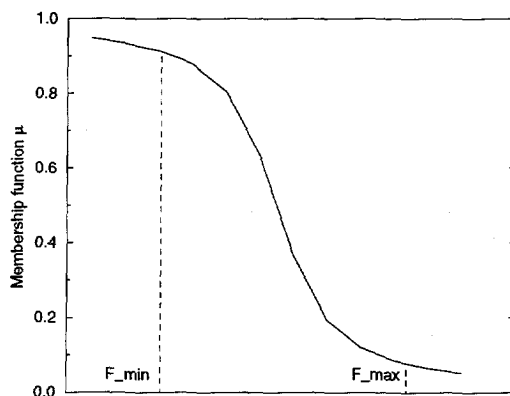


Figure 6.4: Monotonously descending membership function

- For a C_m constraint with a lower-bound (applied to both design points):

$$\begin{aligned} h &= 1, \\ f_l &= C_{m,initial} - 0.1|C_{m,initial}|, \\ f_u &= C_{m,initial}. \end{aligned}$$

This means that a 75% reduction of drag is considered desirable while allowing a 10% violation of the pitching moment constraints. It should be noted that the initial values of C_m correspond to a high value of the membership function, while the initial values of C_d correspond to a low value of μ . This means that at the initial stage a compromise in C_m may readily be taken in favor of a reduction in C_d . It should also be noted that the membership functions of C_d in both design points are initially equal, which means that the two drag values are considered as equally "bad".

It is further noted that in all cases:

- The airfoil is defined by the parameterization scheme described in Section 4.5 with 11 design variables.
- The aerodynamic scaling of the design variables described in Section 4.6 is applied.
- The Euler computations (for ME cases) are performed by the inviscid mode of HI-TASK (Ref. [6]) on a 128×32 grid with the grid adaption procedure (Ref. [21]) activated. The Navier-Stokes computations (for MN cases) are performed by the viscous mode of HI-TASK on a 256×64 grid without grid adaption.
- The optimization procedure employs the FSQP routine.

6.5.1 Results of Case ME-1 (Sum of Equally Weighted Objectives)

The pressure drag and pitching moment results are shown in Figures 6.5 and 6.6. The obtained airfoil geometry and the sum of the weighted pressure drag coefficients are shown in Figure 6.7. The optimization process was stopped after a maximum number of 20 flow analyses in a design point was reached[†].

As can be seen, the drag has been reduced in both design points, with the pitching moment constraints active. There is much more reduction of drag in the second design point ($\approx 45\%$) than in the first design point ($\approx 12\%$). This suggests that the requirement in the first design point is a much more difficult one. This also illustrates that choosing equal weight factors does not necessarily imply a well-balanced solution.

The pressure distributions are shown in Figure 6.8. As can be seen, the shock waves have been weakened in both design points, although, in agreement with the difference in drag reduction, this is more pronounced in the second design point. The state of the geometric constraints are shown in Figure 6.9. The trailing edge angle and leading edge radius constraints are not active, while the cross sectional area constraint is active in most iterates.

6.5.2 Results of Case ME-2 (Fuzzy Optimization)

The drag and pitching moment coefficients are shown in Figure 6.10 and 6.11. The obtained airfoil geometry and the membership functions are shown in Figure 6.12. The optimization process was stopped after a maximum number of 20 flow analyses in a design point was exceeded.

Figure 6.12 shows the history of the membership functions which have been rescaled to the range $[0, 1]$. In fuzzy optimization, the minimum amongst the membership function values corresponds to the most important criterion. The figure indicates that the membership function value of the pitching moment constraints is not the minimum for most iterates. Hence, for most iterates, the pitching moment constraints are not considered as important by the method. This can be interpreted as that the 10% "leeway" provides ample room for the reduction of drag. The figure also indicates that, in the early stages of optimization, the drag in DP-2 was considered more important than in the other design point. This role is switched at the fifth iteration, where the first design point became more important.

Figure 6.12 also demonstrates that taking one criterion (objective or constraint) as the most important one, does not necessarily mean that the other criteria must be compromised (e.g., from the 3rd to the 4th iteration all criteria are improved). As a matter of fact, the change in membership function from one iteration to another gives an indication of the "sensitivity" of the associated criterion with respect to the most important one.

In the final result a balanced reduction of drag is obtained, i.e. about 25%

[†]The terms "Evaluated" and "New iterate" are explained in Section 4.9.1.

and 28% reduction in the first and the second design point, respectively. This is accompanied by about 3% and 5% violation of the pitching moment constraints in the respective design points. The balanced reduction of drag is an expected result, because the drag values in the two design points are initially considered as equally bad, while it is desired to obtain an equal amount of improvement (75% reduction of drag).

The pressure distributions are shown in Figure 6.13. As can be seen, the shock waves are indeed weaker in the final iterate. The states of geometric constraints are shown in Figure 6.14.

6.5.3 Results of Case MN-1 (Sum of Equally Weighted Objectives)

The optimization method failed in this case. The line search in the first iterate could not find any improvement. During the line search, C_{d_2} was indeed decreased but this was accompanied by an increase in C_{d_1} , with the overall effect of increasing the sum of the weighted objectives.

In order to find the culprit in this situation, case MN-1 was modified by specifying $w_2 = 0$, thus ignoring the second objective. It is noted that the modified case is still a two-point design case, because the C_m constraint is still considered in the second design point (which distinguishes it from case N-3). The computer run for the modified case failed too.

There are two possible explanations for the failure of the optimization in both the original and the modified case:

- The search vector is a null vector or, in other words, the initial point is a stationary point. However, this is not recognized by the optimization algorithm. This is probable because the gradients could not reach the accuracy level assumed in the stationary point criteria (internally defined in the optimization routine).
- The initial point is not a stationary point, but the feasible space is too narrow for the accuracy level of the computed gradients.

Whatever has caused the failure, it seems inevitable to accept some constraint violations in order to reduce the drag.

6.5.4 Results of Case MN-2 (Fuzzy Optimization)

The drag and pitching moment coefficients are shown in Figure 6.15 and 6.16. The membership functions and airfoil geometries are given in Figure 6.17. The optimization process was stopped after the line search failed to find a new iterate after 5 flow analyses.

Figure 6.17 shows the history of the membership functions which have been rescaled to be within the range $[0, 1]$. Like in the Euler case, a balanced reduction of drag values has again been obtained, but with a smaller amount than for the inviscid case. About 12% and 10% drag reduction in the first and the second

design point is achieved, respectively. This is accompanied by about 7% and 7.5% violation of the pitching moment constraints in the respective design points.

The C_p distributions are presented in Figure 6.18. In agreement with the drag reduction, the shock waves appear to be slightly weaker in the flow solution of the final iterate. The corresponding C_f distribution is shown in Figure 6.18. The states of geometric constraints are given in Figure 6.20.

6.6 Concluding Remarks

The results presented above seem to indicate that the present method represents a viable approach for dealing with transonic multi-point design problems based on either the Euler or Navier-Stokes equations.

The computational results indicate that (both aerodynamic and geometric) constraints tend to be more stringent in case of the viscous (RANS) flow model. This, of course, is to be expected, since in the Euler case there are no restrictions on the local pressure gradient while in the RANS case these are limited by the ability of the boundary-layer to cope with them without (significant) separation.

The complexity of multi-point design problems is not only incurred by conflicting aerodynamic objectives, but also by restrictions imposed by aerodynamic constraints. The fuzzy optimization method appears to be effective in alleviating the level of problem complexity, because the objectives and constraints are treated in the same way as the set of criteria which have to be achieved. This is measured in terms of membership functions. The results demonstrate that allowing relatively small constraint violations can lead to significant improvements in the objectives.

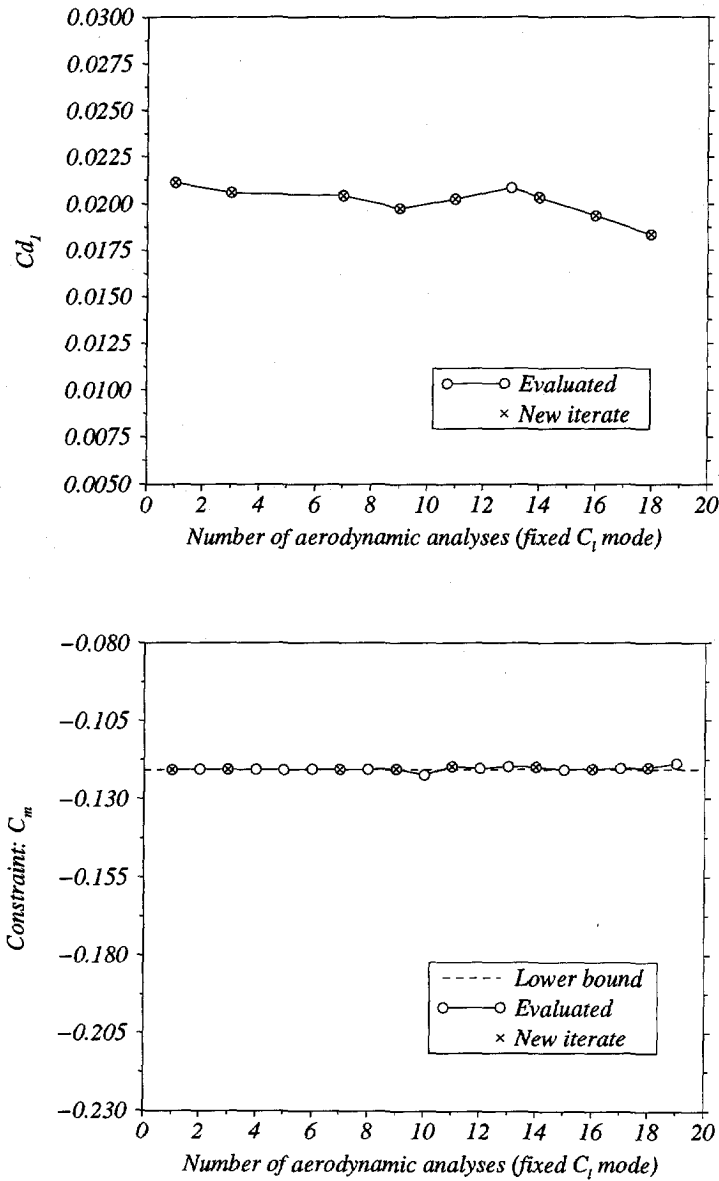


Figure 6.5: Wave drag and pitching moment coefficients in DP-1 of Case ME-1 (Two-point transonic constrained wave drag reduction by sum of weighted objectives. Flow model: Euler. Weight factor: $w_1 = 0.5$. Design point: $M_1 = 0.72$, $C_{l_1} = 1$. Initial airfoil: RAE2822. Constraints: pitching moments, leading edge radius, trailing edge included angle, and cross sectional area).

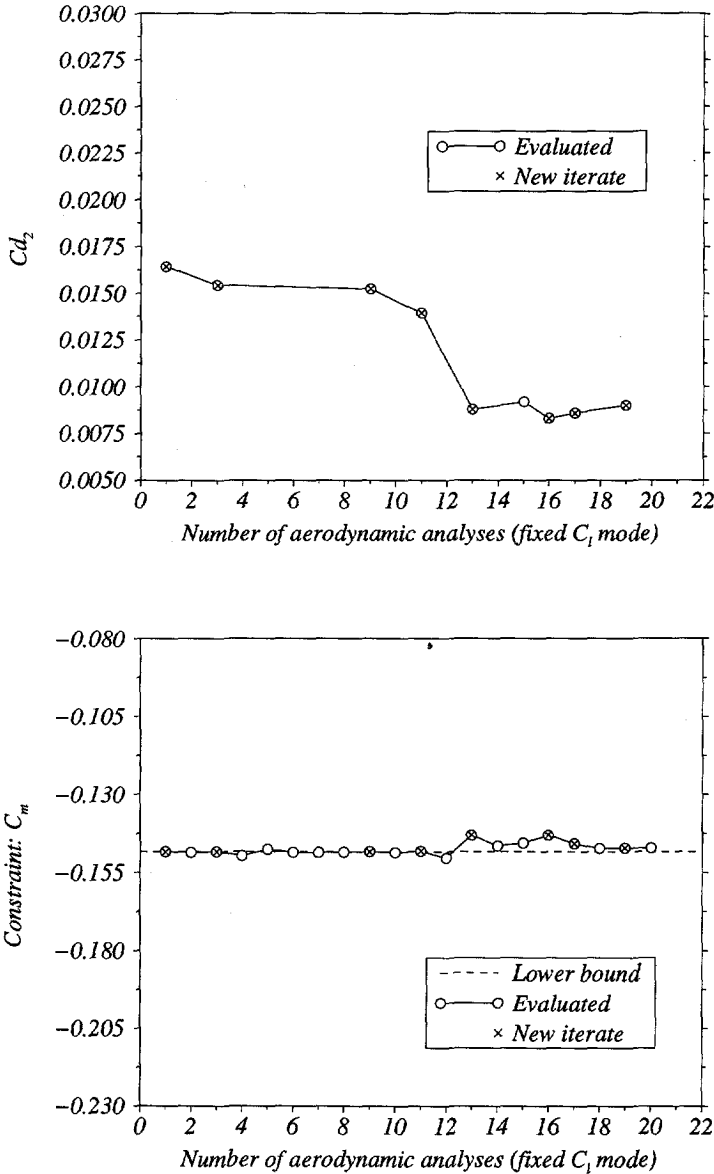


Figure 6.6: Wave drag and pitching moment coefficients in DP-2 of Case ME-1 (Two-point transonic constrained wave drag reduction by sum of weighted objectives. Flow model: Euler. Weight factor: $w_2 = 0.5$. Design point: $M_2 = 0.78$, $C_{l_2} = 0.5$. Initial airfoil: RAE2822. Constraints: pitching moments, leading edge radius, trailing edge included angle, and cross sectional area).

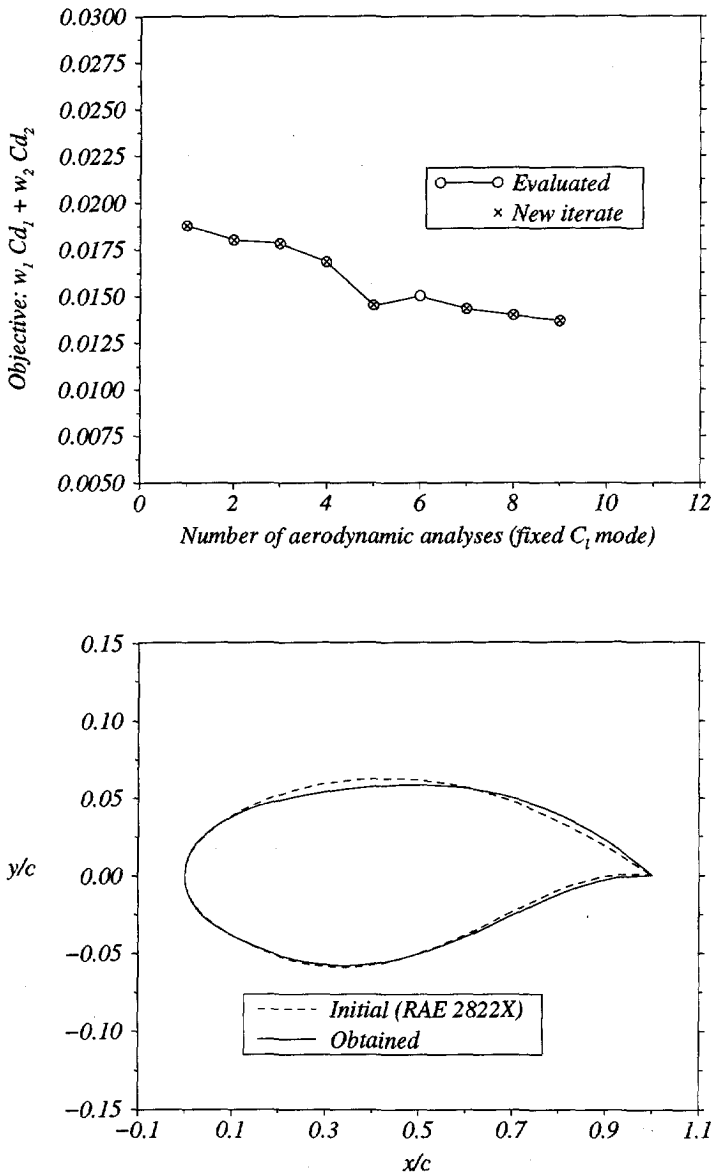


Figure 6.7: Sum of weighted objectives and airfoil geometries of Case ME-1 (*Two-point transonic constrained wave drag reduction by sum of weighted objectives. Flow model: Euler. Weight factors: $w_1 = 0.5$, $w_2 = 0.5$. Design points: $M_1 = 0.72$, $C_{l_1} = 1$, $M_2 = 0.78$, $C_{l_2} = 0.5$. Initial airfoil: RAE2822. Constraints: pitching moments, leading edge radius, trailing edge included angle, and cross sectional area*).

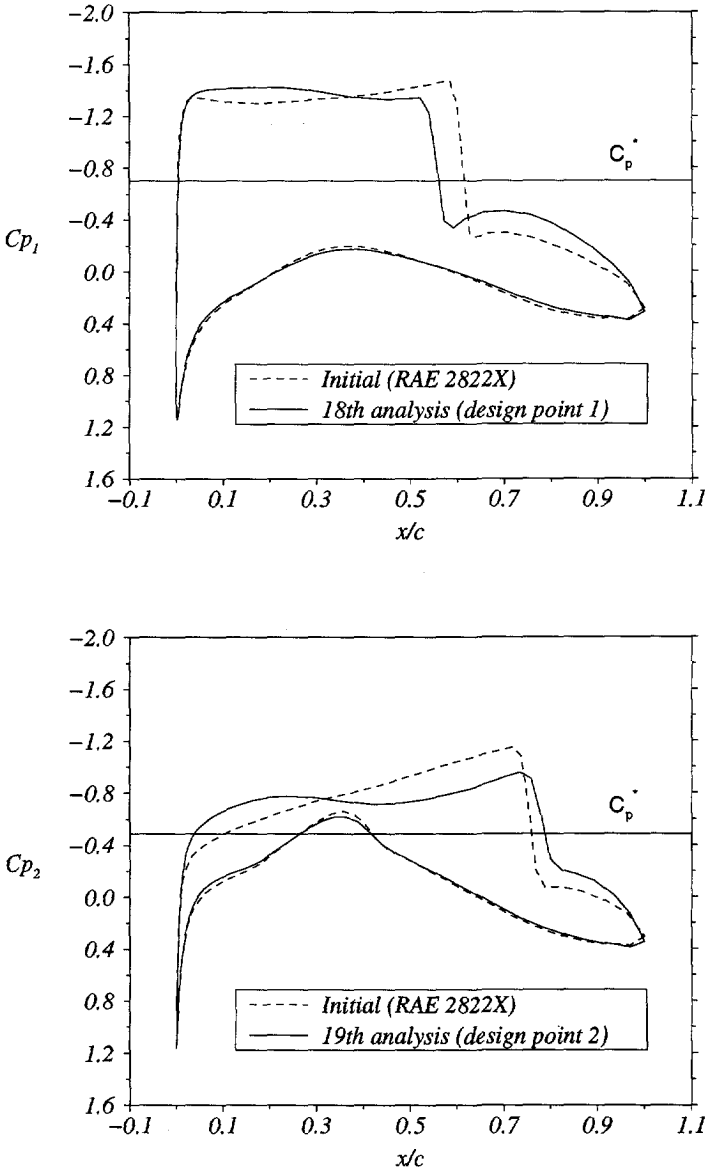


Figure 6.8: C_p distributions of Case ME-1 (*Two-point transonic constrained wave drag reduction by sum of weighted objectives. Flow model: Euler. Weight factors: $w_1 = 0.5$, $w_2 = 0.5$. Design points: $M_1 = 0.72$, $C_{i1} = 1$, $M_2 = 0.78$, $C_{i2} = 0.5$. Initial airfoil: RAE2822. Constraints: pitching moments, leading edge radius, trailing edge included angle, and cross sectional area*).

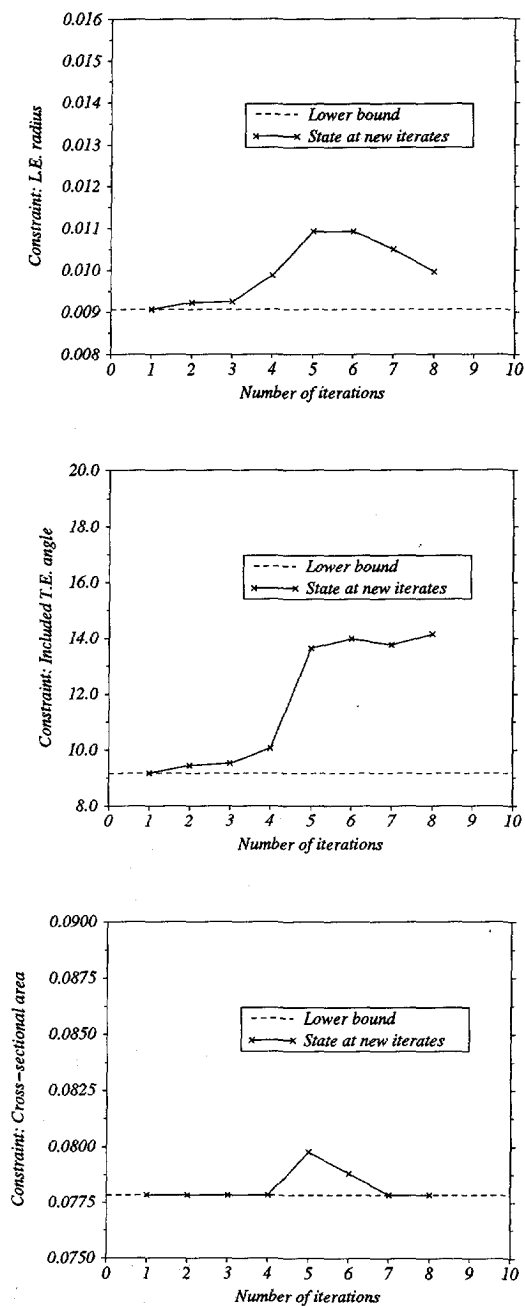


Figure 6.9: State of geometric constraints of Case ME-1.

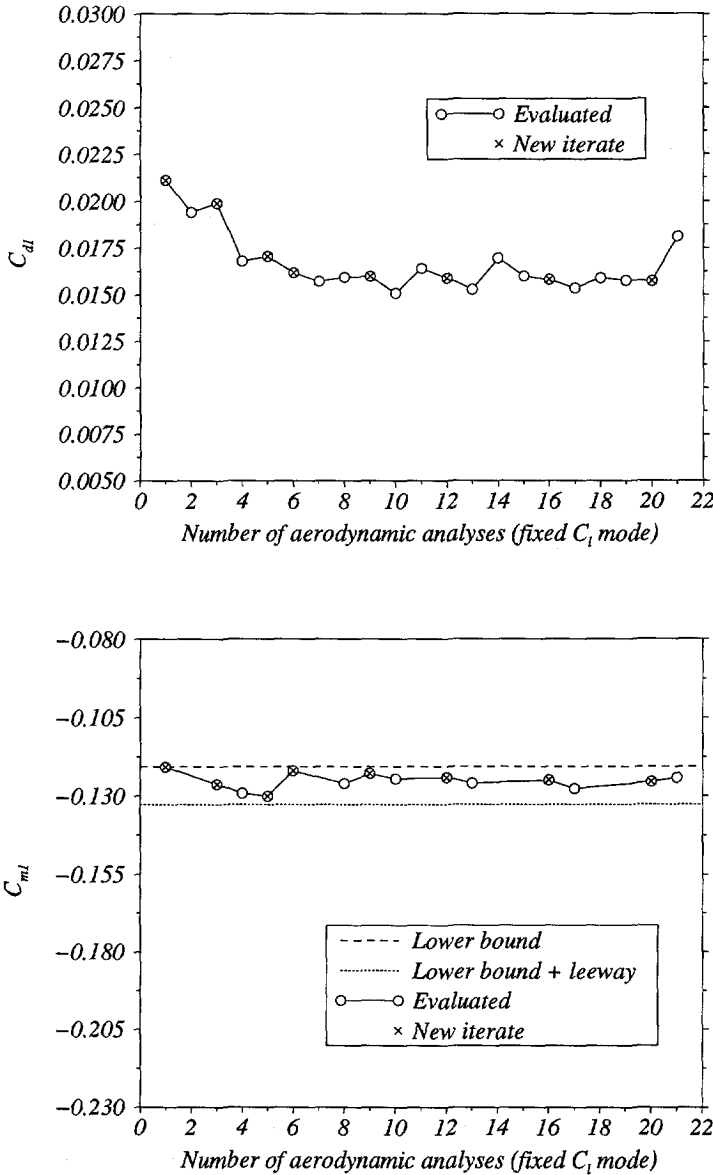


Figure 6.10: Wave drag and pitching moment coefficients in DP-1 of Case ME-2 (Two-point transonic constrained wave drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Wave drag and pitching moment. Flow model: Euler. Design point: $M_1 = 0.72$, $C_{l_1} = 1$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

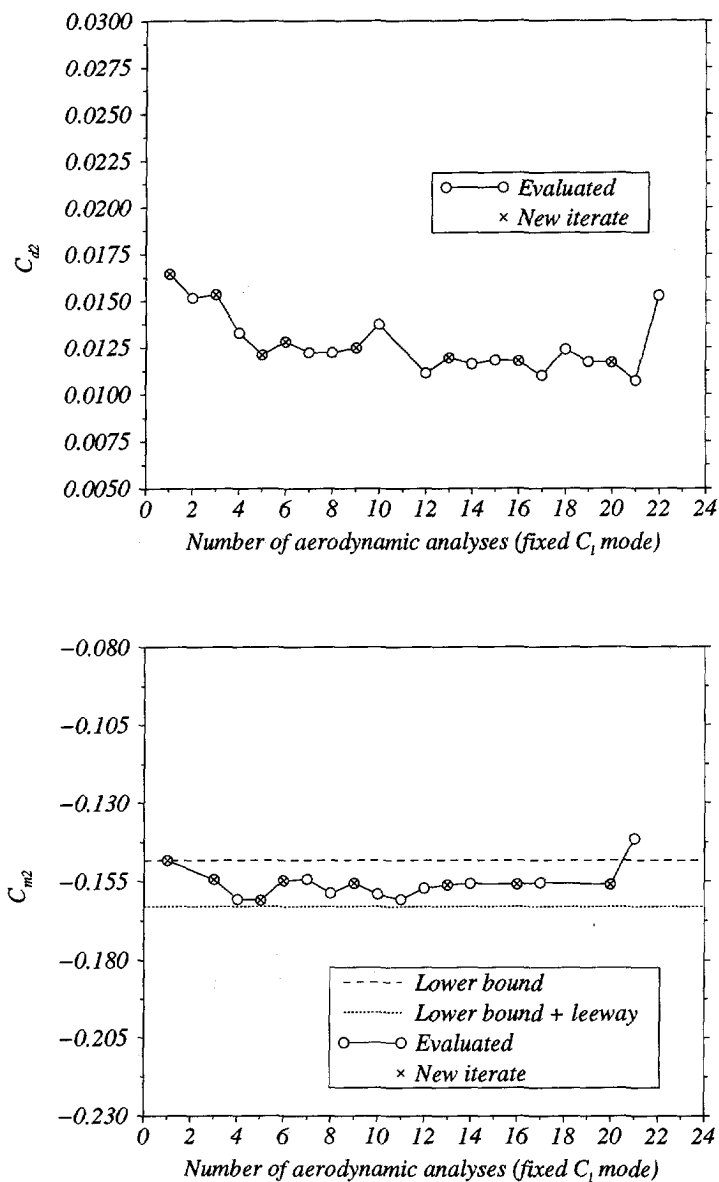


Figure 6.11: Wave drag and pitching moment coefficients in DP-2 of Case ME-2 (Two-point transonic constrained wave drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Wave drag and pitching moment. Flow model: Euler. Design point: $M_2 = 0.78$, $C_{l_2} = 0.5$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

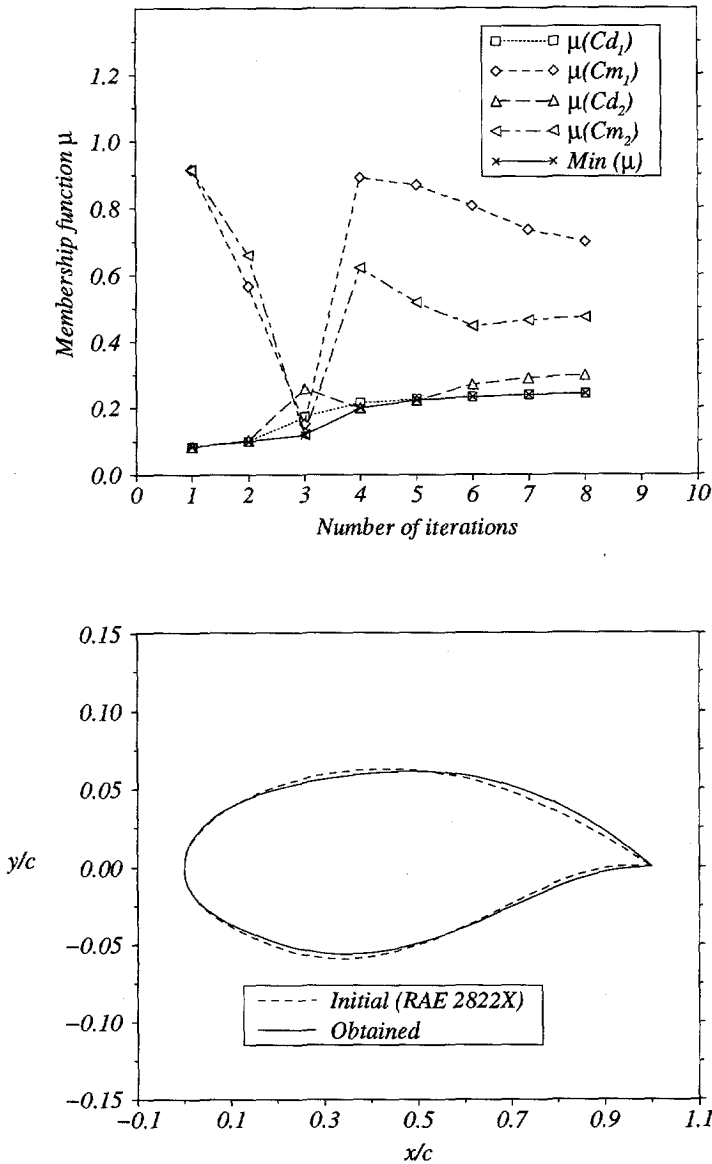


Figure 6.12: Membership functions and airfoil geometries of Case ME-2 (Two-point transonic constrained wave drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Wave drag and pitching moment. Flow model: Euler. Design points: $M_1 = 0.72$, $C_{l_1} = 1$, $M_2 = 0.78$, $C_{l_2} = 0.5$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

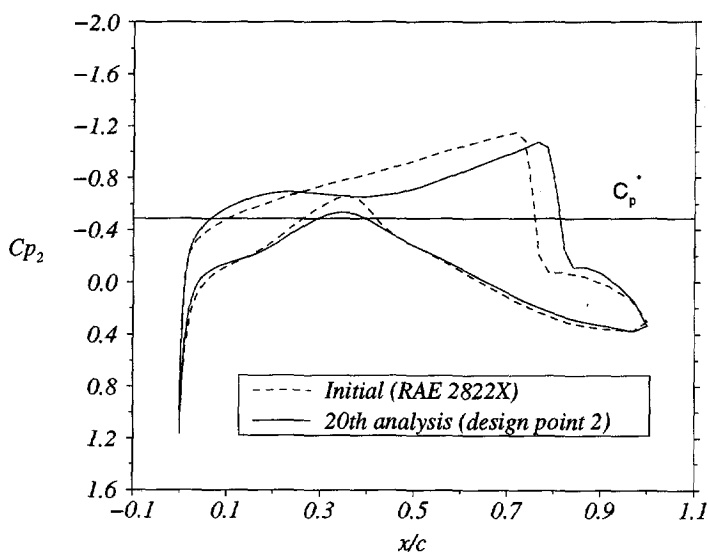
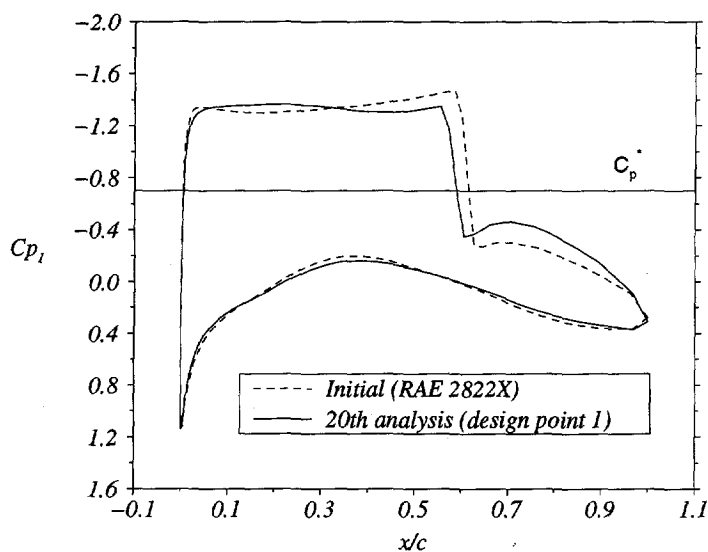


Figure 6.13: C_p distributions of Case ME-2 (Two-point transonic constrained wave drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Wave drag and pitching moment. Flow model: Euler. Design points: $M_1 = 0.72$, $C_{l1} = 1$, $M_2 = 0.78$, $C_{l2} = 0.5$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

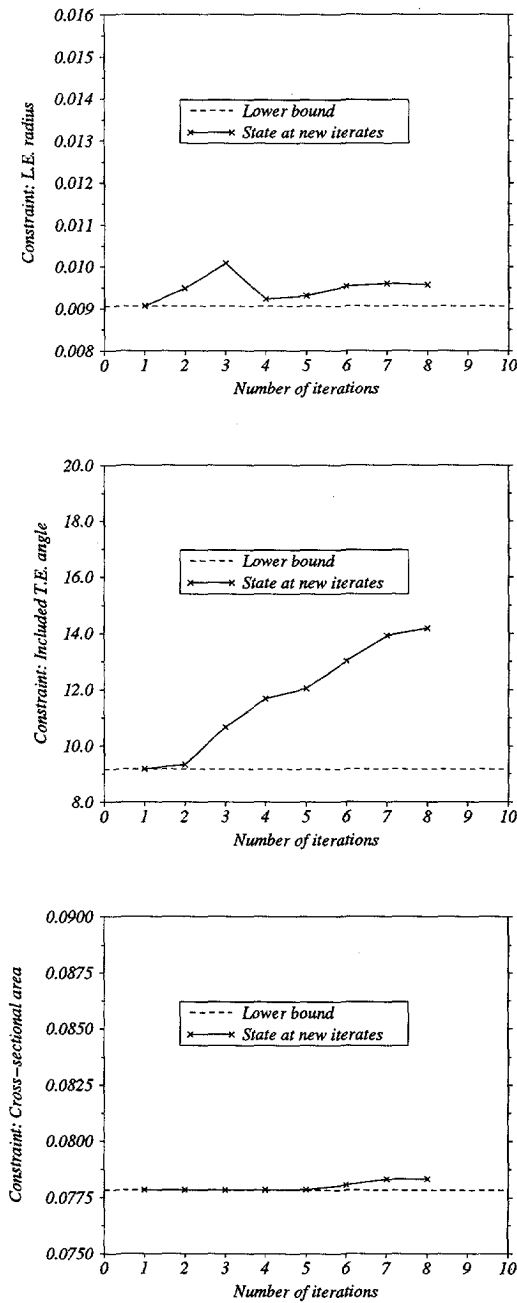


Figure 6.14: State of geometric constraints of Case ME-2.

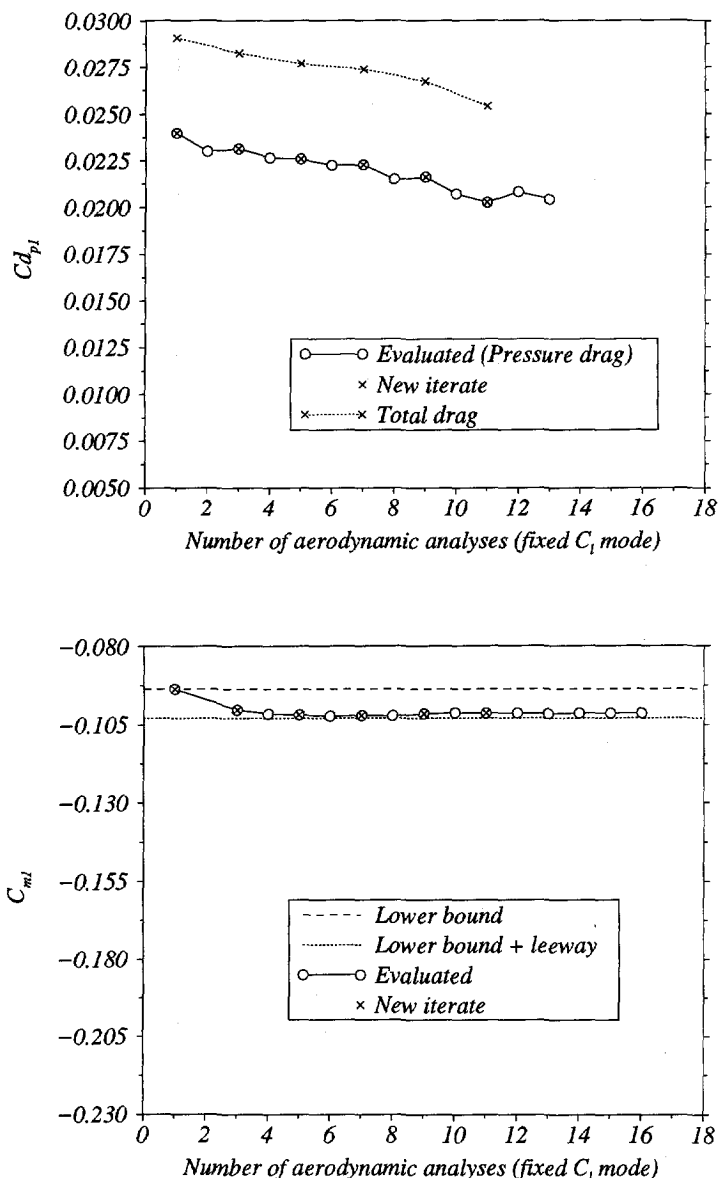


Figure 6.15: Pressure drag and pitching moment coefficients in DP-1 of Case MN-2 (Two-point transonic constrained pressure drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Pressure drag and pitching moment. Flow model: Navier-Stokes. Design point: $M_1 = 0.72$, $C_{l_1} = 1$, $Re_1 = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

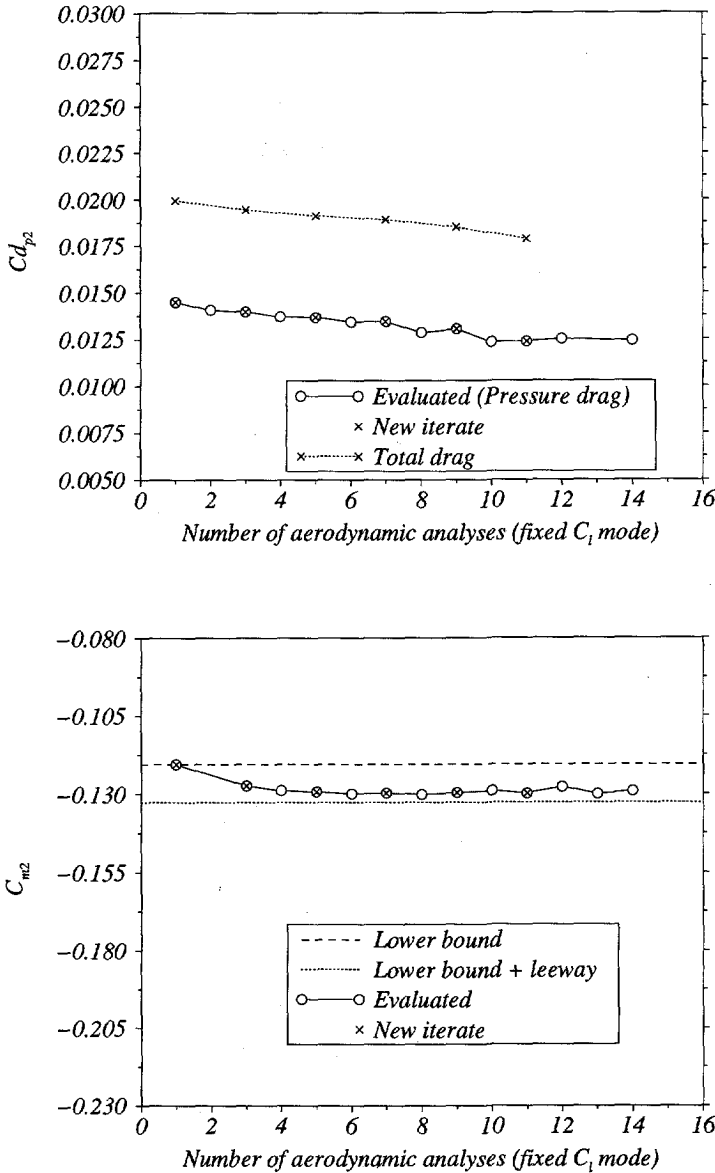


Figure 6.16: Pressure drag and pitching moment coefficients in DP-2 of Case MN-2 (Two-point transonic constrained pressure drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Pressure drag and pitching moment. Flow model: Navier-Stokes. Design points: $M_2 = 0.78$, $C_{l_2} = 0.5$, $Re_2 = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

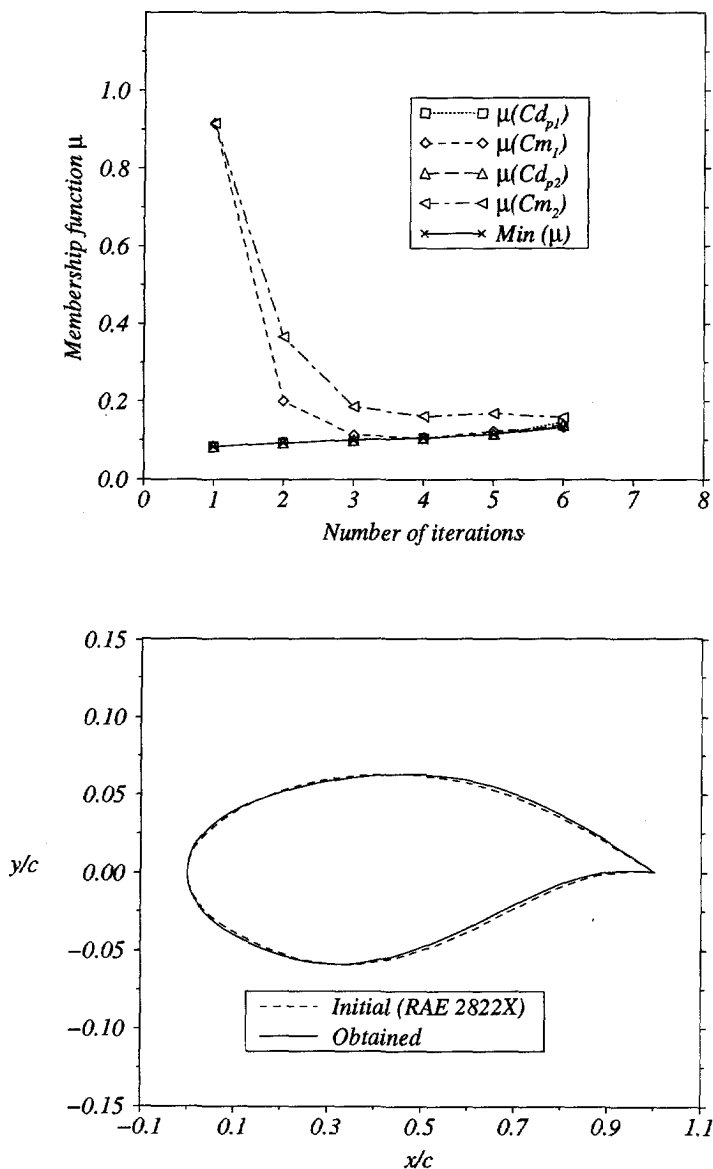


Figure 6.17: Membership functions and airfoil geometries of Case MN-2 (Two-point transonic constrained pressure drag reduction by fuzzy optimization. Fuzzy objectives/constraints: Pressure drag and pitching moment. Flow model: Navier-Stokes. Design points: $M_1 = 0.72$, $C_{l_1} = 1$, $M_2 = 0.78$, $C_{l_2} = 0.5$, $Re_1 = Re_2 = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and airfoil volume).

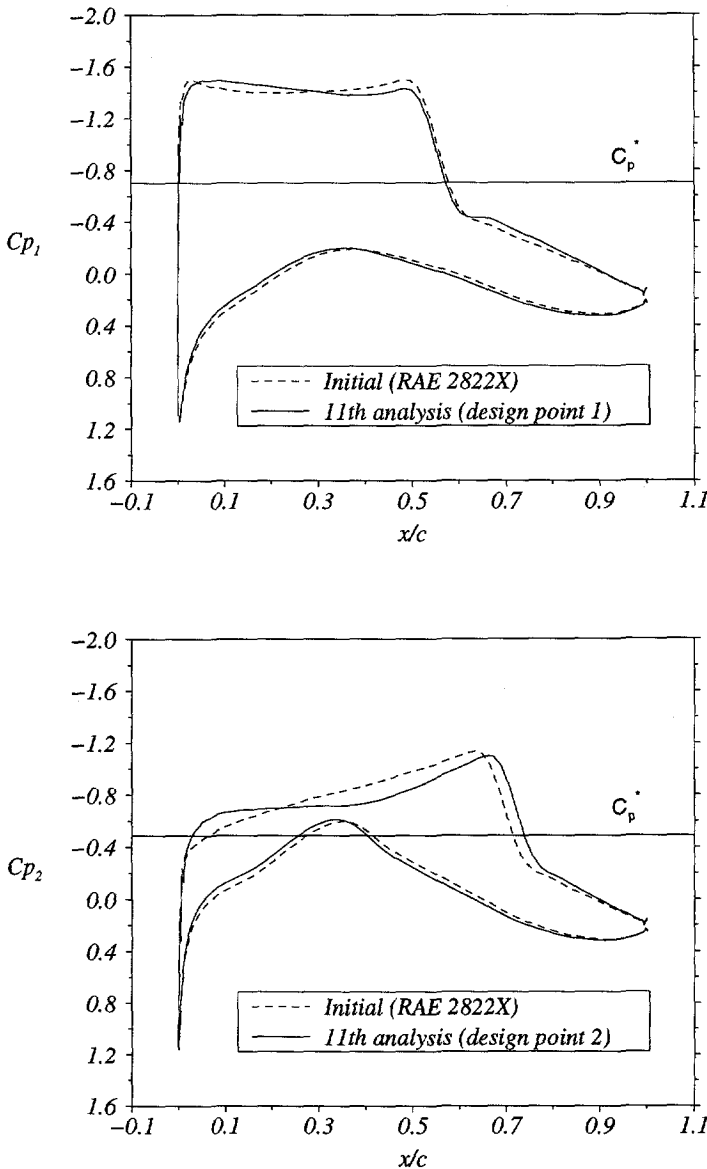


Figure 6.18: C_p distributions of Case MN-2 (Two-point transonic constrained pressure drag reduction by fuzzy optimization. Fuzzy objectives/ constraints: Pressure drag and pitching moment. Flow model: Navier-Stokes. Design points: $M_1 = 0.72$, $C_{l1} = 1$, $M_2 = 0.78$, $C_{l2} = 0.5$, $Re_1 = Re_2 = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

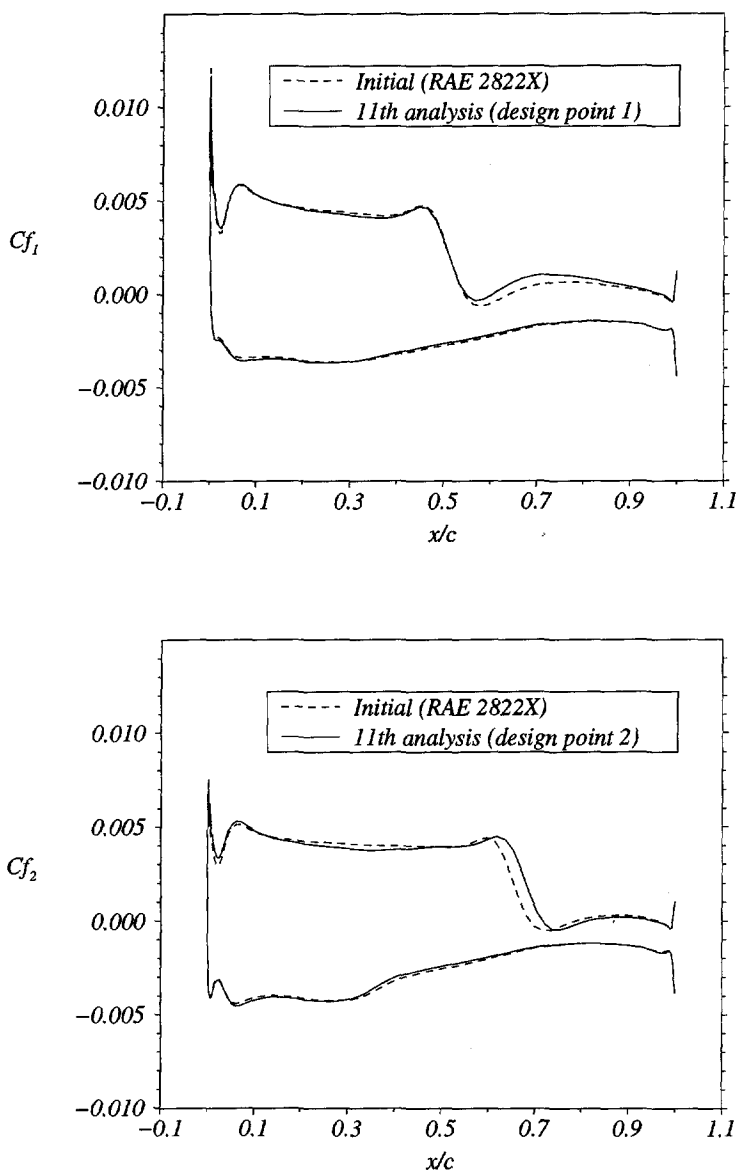


Figure 6.19: C_f distributions of Case MN-2 (Two-point transonic constrained pressure drag reduction by fuzzy optimization. Fuzzy objectives/ constraints: Pressure drag and pitching moment. Flow model: Navier-Stokes. Design points: $M_1 = 0.72$, $C_{l1} = 1$, $M_2 = 0.78$, $C_{l2} = 0.5$, $Re_1 = Re_2 = 6.5 \times 10^6$. Initial airfoil: RAE2822. Constraints: leading edge radius, trailing edge included angle, and cross sectional area).

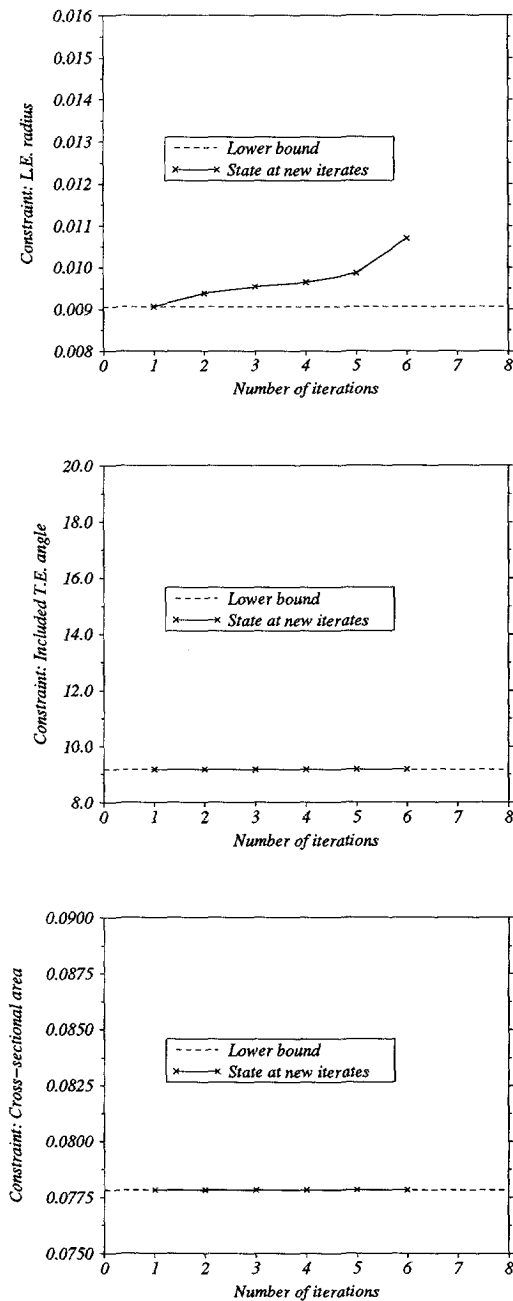


Figure 6.20: State of geometric constraints of Case MN-2.

Chapter 7

Conclusions and Recommendations

7.1 Direct Aerodynamic Optimization Methodology

The objective of the investigation presented in this thesis is to construct an aerodynamic design optimization methodology that is capable of dealing with general constrained multi-point aerodynamic design problems in two-dimensional inviscid and viscous flow. It has been argued that for this purpose the methodology should satisfy the following requirements:

- The methodology should offer the possibility to choose between different flow models. The feasibility of applying the present methodology for the flow models described by the Laplace, Euler and Reynolds-Averaged Navier Stokes equations has been demonstrated.
- The method should be capable of handling aerodynamic design problems defined in terms of global aerodynamic coefficients with aerodynamic and geometric constraints. This is realized by adopting the direct optimization methodology.
- The method should be capable of dealing with multi-point design problems. This is realized by incorporating a multi-objective optimization strategy. It has been demonstrated that the method is effective in resolving problems resulting from conflicting design criteria.
- The computational cost associated with application of the method should be at least an order of magnitude lower than those of aerodynamic design methods based on *numerical* optimization. This is realized by the combination of a quasi-second-order feasible optimization algorithm with the variational method for computing the gradient. This allows useful results to be obtained at a computational cost that is comparable to that of obtaining 18 to 50 flow solutions. This is to be compared to the many hundreds of flow

solutions required in the direct numerical optimization approach employing alternative methods for obtaining the gradient.

- In order to limit the code development effort, the design code has been set up modularly, consisting of an existing flow solver, an adjoint solver based on the flow solver, a gradient evaluator, and existing optimization routine.

7.2 Optimization with the Laplace Equation

The Laplace equation describes an incompressible potential flow model. Because drag is not represented in this model, it is most appropriate to formulate an aerodynamic design problem as an inverse problem. This requires expertise in prescribing a realizable target velocity (or pressure) distribution, related to the question of the well-posedness of the inverse problem.

It is known that the velocity distribution over a closed contoured airfoil immersed in an incompressible potential flow must satisfy three compatibility conditions. For this purpose free parameters have been introduced in the target velocity distribution. A prescribed (not realizable) target velocity distribution is automatically modified in such a way that an airfoil geometry can be found which corresponds to the modified distribution. Incorporating this idea into an optimization methodology has resulted in a robust inverse code.

For a typical reconstruction type of inverse problem, the method requires about 15-20 flow analyses before obtaining the optimal airfoil shape. This relatively high computational cost can be reduced, e.g. by using a better representation of the airfoil geometry.

7.3 Optimization with the Euler Equations

The Euler equations describe an inviscid compressible flow model. As compressibility effects like shock waves are represented, the Euler equations permit a direct treatment of transonic (wave) drag reduction problems.

The method has been applied successfully to a number of transonic wave drag reduction problems with geometric and aerodynamic constraints. This demonstrated its capability of reducing the shock strength or, in certain cases, removing the shock wave completely. Through grid refinement studies, it can be concluded that the drag reduction resulting from the optimization can be considered significant for practical situations.

7.4 Optimization with the Reynolds-Averaged Navier-Stokes Equations

The Reynolds-averaged Navier-Stokes equations describe a viscous compressible flow model. As compressibility and viscous effects are incorporated, the Navier-

Stokes equations permit in principle a complete direct treatment of a wide class of aerodynamic design problems.

The method has been applied to a number of transonic pressure drag reduction problems with geometric and aerodynamic constraints. This has shown to be able to produce airfoils with significantly lower drag values, where in certain cases the shock waves are practically removed. For a good assessment of the resulting drag reduction, the error level of the computed drag values should be known, e.g. by grid refinement studies.

A significant problem identified during the investigation is the problem of total drag reduction. It is not yet clear how the skin-friction term can have a major effect on the optimization process.

7.5 Multi-point Aerodynamic Design

The present method treats a multi-point aerodynamic design problem as a multi-objective optimization problem. The method has been applied successfully to a two-point aerodynamic design problem with aerodynamic as well as geometric constraints.

Two approaches for obtaining the solution have been investigated, namely:

- (i) The method of sum of weighted objectives. An investigation has been carried out for equal weight factors. The method is capable of reducing the drag values for both design points simultaneously by strictly satisfying the constraints.
- (ii) The method of fuzzy optimization. This method has shown to be suitable for resolving problems resulting from conflicting design criteria (in terms of objectives and constraints). It has been demonstrated that allowing a small violation of the (aerodynamic) constraints can lead to a significant improvement in the final value of the objectives.

7.6 Recommendations for Future Research

In the test cases considered in the present investigation, the design space was defined by 11 design variables. The sensitivity of the solutions to the number of design variables has not been investigated. In the search for a better design, the number of design variables should be increased in order to cover a larger design space. In addition to that, different initial airfoil shapes should be tried in order to investigate the sensitivity (if any) of the final result for the starting solution.

The multi-point design problems addressed in this thesis are two-point pressure drag reduction problems of single-element airfoils at transonic flow conditions. The design points have a different lift coefficient and Mach number. It would be desirable to investigate the suitability of the method for multi-point design problems involving other types of aerodynamic functionals, e.g. maximization of C_l with the purpose of improving the buffet boundary of an airfoil,

or minimization of C_d obtained from integration along the far-field boundaries. Also, for a complete treatment of the multi-point design problem, it is desirable to consider subsonic flow conditions.

The present methodology is not limited to two-dimensional single airfoil problems, but can be extended in order to deal with multi-component airfoils and three-dimensional design problems. This would imply a drastic increase in the number of design variables. Then, there might be a need for an optimization routine dedicated to large-scale optimization problems.

Bibliography

- [1] *NAG Fortran Library Manual (Mark 14)*, 1990.
- [2] J.D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, second edition, 1991.
- [3] B.A. Baldwin and H. Lomax. Thin-layer approximation and algebraic model for separated turbulent flows. *AIAA Paper 78-257*, 1978.
- [4] R.M. Barron and C.F. An. Analysis and design of transonic airfoils using streamwise coordinates. In G.S. Dulikravich, editor, *Third International Conference on Inverse Design and Optimization in Engineering Sciences (ICIDES-III)*, October 1991.
- [5] A. Betz. Änderung eines Profils zur Erzielung einer vorgegebenen Änderung der Druckverteilung. *Luftfahrtforschung*, (11):157-64, 1934.
- [6] F.J. Brandsma. Description of the method used by NLR. In Haase W. et.al, editor, *EUROVAL-A European Initiative on Validation of CFD Codes*, volume 42 of *Notes on Numerical Fluid Mechanics*. Vieweg, Brunswick, 1993.
- [7] D.R. Bristow. A solution to the inverse problem for incompressible axisymmetric potential flow. *AIAA Paper 74-520*, June 1974.
- [8] J. Huan and V. Modi. Optimum Design of Minimum Drag Bodies in Incompressible Laminar Flow Using a Control Theory Approach. *Inverse Problems in Engineering*, 1:1-25, 1994.
- [9] W.H. Davis Jr. Technique for developing design tools from the analysis methods of computational aerodynamics. *AIAA Paper 79-1529*, July 1979.
- [10] K. Dems and Z. Mróz. Variational approach by means of adjoint systems to structural optimization and sensitivity analysis-II. *Int. J. Solids Structures*, 20(6):527-552, 1984.
- [11] M. Drela. *Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations*. PhD thesis, Massachusetts Institute of Technology, 1985.
- [12] M. Drela. Viscous and inviscid inverse schemes using Newton's method. In *Special Course on inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD-R-780, May 1990.

- [13] G.S. Dulikravich. A stream-function-coordinate (SFC) concept in aerodynamic shape design. In *Special Course on inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications*, AGARD-R-780, May 1990.
- [14] G.S. Dulikravich. Aerodynamic shape design and optimization. *AIAA Paper 91-0476*, January 1991.
- [15] P.D. Frank and G.R. Shubin. *A Comparison of optimization-based Approaches for a Model Computational Aerodynamics Design Problem*. Boeing ECA-TR-136, 1990.
- [16] J.M.J. Fray and J.W. Slooff. A constrained inverse method for the aerodynamic design of thick wings with given pressure distribution in subsonic flow. AGARD CP No. 285, Paper 16, 1980.
- [17] J.M.J. Fray, J.W. Slooff, J.W. Boerstoeel, and A. Kassies. Inverse method with geometric constraints for transonic aerofoil design. *Intl. J. Numerical Method in Engineering*, 22:327-339, 1986.
- [18] P. Garabedian and G. McFadden. Design of supercritical swept wings. *AIAA Journal*, 20(3):289-291, 1982.
- [19] A. Iollo and M.D. Salas. *Contribution to The Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks*. ICASE Rep. no. 95-20, 1995.
- [20] J.O. Hager, S. Eyi, and K.D. Lee. Design efficiency evaluation for transonic airfoil optimization: A case for Navier-Stokes design. *AIAA Paper 93-3112*, July 1993.
- [21] R. Hagmeijer. Grid adaption based on modified anisotropic diffusion equations formulated in the parametric domain. *J. of Computational Physics*, 115(1):169-183, Nov 1994.
- [22] E.J. Haug, K.K. Choi, and V. Komkov. *Design Sensitivity Analysis of Structural Systems*. Academic Press, 1986.
- [23] P.A. Henne. An inverse transonic wing design method. *AIAA Paper 80-0330*, 1980.
- [24] R.M. Hicks, E.M. Murman, and G.N. Vanderplaats. *An Assessment of Airfoil Design by Numerical Optimization*. NASA TM X-3092, 1974.
- [25] R.M. Hicks and G.N. Vanderplaats. Application of numerical optimization to the design of supercritical airfoils without drag-creep. *SAE Paper 770440*, April 1977.
- [26] W.P. Huffman et al. Practical design and optimization in computational fluid dynamics. *AIAA Paper 93-3111*, July 1993.

- [27] S. Huo. Optimization based on boundary layer concept for compressible flows. *Journal of Engineering for Power*, pages 195–205, April 1975.
- [28] A. Jameson. *Aerodynamic Design via control theory*. ICASE Rep. no. 88-64, 1988.
- [29] A. Jameson. Optimum aerodynamic design via boundary control. In *Optimum Design Methods for Aerodynamics*, AGARD-R-803, November 1994.
- [30] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA Paper 95-1729*, June 1995.
- [31] A. Jameson. Optimum aerodynamic design using control theory. *CFD Review*, pages 495–528, 1995.
- [32] A. Jameson. The present status, challenges, and future developments in computational fluid dynamics. In *Fluid Dynamics Panel Symposium*, October 1995.
- [33] A. Jameson, W. Schmidt, and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using runge-kutta time-stepping schemes. *AIAA Paper 81-1259*, 1981.
- [34] R.A. Kennelly Jr. Improved method for transonic airfoil design-by-optimization. *AIAA Paper 83-1864*, July 1983.
- [35] H.O. Kreiss. Initial boundary value problems for hyperbolic systems. *Comm. Pure & Appl. Math.*, 23:277–298, 1976.
- [36] K. Kubrynski. Design of 3-dimensional complex airplane configurations with specified pressure distribution via optimization. In G.S. Dulikravich, editor, *Third International Conference on Inverse Design and Optimization in Engineering Sciences (ICIDES-III)*, October 1991.
- [37] Th.E. Labrujère. *Multi-Element Airfoil Design by Optimization*. NLR MP 78023 U, 1978.
- [38] Th.E. Labrujère and J.W. Slooff. Computational methods for the aerodynamic design of aircraft components. *Annu. Rev. Fluid Mech.*, (25):183–214, 1993.
- [39] Th.E. Labrujère and J.v.d. Vooren. Calculus of variations applied to 2d multi-point airfoil design. In *The First European Computational Fluid Dynamics Conference*, September 1992.
- [40] R.H. Liebeck. On the design of subsonic airfoils for high lift. *AIAA Paper 76-046*, July 1976.
- [41] M.J. Lighthill. *A New Method of Two-Dimensional Aerodynamic Design*. Aeronaut. Res. Counc. Rep. Memo. 2112, 1945.

- [42] J.B. Malone, J.C. Narramore, and L.N. Sankar. An efficient airfoil design method using the Navier-Stokes equations. In *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, May 1989.
- [43] W. Mangler. Die Berechnung eines Tragflügelprofils mit vorgeschriebener Druckverteilung. *Jahrb. Deutsche Luftfahrtforsch.*, pages I46-I53, 1938.
- [44] G. Mosetti and C. Poloni. Aerodynamic shape optimization by means of a genetic algorithm. In *5th Int. Symp. on Computational Fluid Dynamics, Sendai*, volume II, 1993.
- [45] J.C. Narramore and R.D. Yeary. Airfoil design and analysis using an information systems approach. *AIAA Paper 80-1444*, July 1980.
- [46] O. Pironneau. On optimum profile in stokes flow. *Journal of Fluid Mechanics*, (59):117-128, 1972.
- [47] J. Reuther and A. Jameson. Aerodynamic shape optimization of wing and wing-body configurations using control theory. *AIAA Paper 95-0123*, January 1995.
- [48] M.H. Rizk. Aerodynamic optimization by simultaneously updating flow variables and design parameters. In *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, May 1989.
- [49] M.S. Selg and M.D. Maughmer. A multi-point inverse airfoil design method based on conformal mapping. *AIAA Paper 91-0069*, January 1991.
- [50] G.R. Shubin and P.D. Frank. *A comparison of the implicit gradient approach and the variational approach to aerodynamic design optimization*. Boeing AMS-TR-163, 1991.
- [51] J.W. Slooff. *A Survey of Computational Methods for Subsonic and Transonic Aerodynamic Design*. NLR MP 84066 U, 1984.
- [52] J.W. Slooff. Numerical aircraft aerodynamics. Lecture Notes of the Faculty of Aerospace Engineering TU-Delft, 1988.
- [53] J.W. Slooff and N. Voogt. *Aerodynamic Design of Thick Supercritical Wings through the Concept of Equivalent Subsonic Pressure Distribution*. NLR MP 78011 U, 1978.
- [54] L.A. Smith and R.L. Campbell. *A Method for The Design for Transonic Flexible Wings*. NASA TP 3045, 1990.
- [55] B.I. Soemarwoto. Robust shape design in aerodynamics. *Inverse Problems in Engineering*, 1:153-177, 1994.

- [56] W. Stadler. *Fundamentals of Multicriteria Optimization*, pages 1–25. Multicriteria Optimization in Engineering and in the Sciences. Plenum Press, 1988.
- [57] T. Strand. Exact method for designing airfoils with given velocity distribution in incompressible flow. *J. Aircraft*, (11):651–659, 1973.
- [58] S. Ta'asan, G. Kuruwila, and M.D. Salas. Aerodynamic design and optimization in one shot. *AIAA Paper 92-0025*, January 1992.
- [59] J.J. Thibert. One point and multi-point design optimization for airplane and helicopter application. In *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, May 1989.
- [60] T.L. Tranen. A rapid computer aided transonic airfoil design method. *AIAA Paper 74-501*, 1974.
- [61] R.F. van den Dam, J.A. van Egmond, and J.W. Slooff. Optimization of target pressure distributions. In *Computational Methods for Aerodynamic Design (Inverse) and Optimization*, AGARD CP-463, May 1989.
- [62] J.A. van Egmond, Th.E. Labrujère, and J.v.d. Vooren. *Test cases for a work shop within the Brite Euram Project BE 1082*. NLR CR 90329 L, 1990.
- [63] G.N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design With Applications*. McGraw-Hill Book Company, 1984.
- [64] A.E.P. Veldman. The calculation of incompressible boundary layers with strong viscous-inviscid interaction. In *Computation of Viscous-Inviscid Interactions*, AGARD CP-291, October 1980.
- [65] G. Volpe. The inverse design of closed airfoils in transonic flow. *AIAA Paper 83-0504*, 1983.
- [66] G. Volpe and R.E. Melnik. The role of constraints in the inverse design problem for transonic airfoils. *AIAA Paper 81-1233*, 1981.
- [67] J. van der Vooren and J.W. Slooff. CFD-based drag prediction: State-of-the-art, theory, prospect. AIAA Professional Studies, Course on Drag Prediction and Measurement, August 1990.
- [68] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
- [69] J.L. Zhou and A.L. Tits. *User's Guide for FSQP Version 3.3b: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints*. Electrical Engineering Department and Institute for Systems Research, University of Maryland.
- [70] H.J. Zimmermann. Description and optimization of fuzzy systems. *Int. J. General Systems*, 2:209–215, 1976.

Appendix A

Matrices of the Right and Left Eigenvectors of the Normal Flux Jacobian

The right eigenvectors of the matrix $C = \vec{A} \cdot \vec{n}$ can be obtained as

$$X = \begin{bmatrix} 1 & 0 & \frac{\rho}{2a} & -\frac{\rho}{2a} \\ u & n_y \rho & \frac{\rho}{2a}(u + n_x a) & -\frac{\rho}{2a}(u - n_x a) \\ v & -n_x \rho & \frac{\rho}{2a}(v + n_y a) & -\frac{\rho}{2a}(v - n_y a) \\ \frac{V^2}{2} & \rho \vec{V} \cdot \vec{s} & \frac{\rho \vec{V} \cdot \vec{n}}{2} + \frac{\rho V^2}{4a} + \frac{\rho a}{2(\gamma - 1)} & \frac{\rho \vec{V} \cdot \vec{n}}{2} - \frac{\rho V^2}{4a} - \frac{\rho a}{2(\gamma - 1)} \end{bmatrix}$$

where $V^2 = u^2 + v^2$. The left eigenvectors is the inverse of the right eigenvectors, that is

$$X^{-1} = \begin{bmatrix} 1 - \frac{(\gamma - 1)V^2}{2a^2} & \frac{(\gamma - 1)u}{a^2} & \frac{(\gamma - 1)v}{a^2} & \frac{1 - \gamma}{a^2} \\ \frac{-\vec{V} \cdot \vec{s}}{\rho} & \frac{n_y}{\rho} & \frac{-n_x}{\rho} & 0 \\ \frac{(\gamma - 1)V^2 - 2a\vec{V} \cdot \vec{n}}{2\rho a} & \frac{(1 - \gamma)u + n_x a}{\rho a} & \frac{(1 - \gamma)v + n_y a}{\rho a} & \frac{\gamma - 1}{\rho a} \\ \frac{(1 - \gamma)V^2 - 2a\vec{V} \cdot \vec{n}}{2\rho a} & \frac{(\gamma - 1)u + n_x a}{\rho a} & \frac{(\gamma - 1)v + n_y a}{\rho a} & \frac{1 - \gamma}{\rho a} \end{bmatrix}$$

The definitions of the matrices X_- , X_+ , and X_+^* depend on whether an inflow or an outflow occurs:

- For subsonic inflow,

$$X_- = \begin{bmatrix} X_{13} \\ X_{23} \\ X_{33} \\ X_{43} \end{bmatrix}, \quad X_+ = \begin{bmatrix} X_{11} & X_{12} & X_{14} \\ X_{21} & X_{22} & X_{24} \\ X_{31} & X_{32} & X_{34} \\ X_{41} & X_{42} & X_{44} \end{bmatrix}, \quad X_+^* = \begin{bmatrix} X_{12} & X_{14} \\ X_{22} & X_{24} \\ X_{32} & X_{34} \\ X_{42} & X_{44} \end{bmatrix}.$$

- For subsonic outflow,

$$X_- = \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ X_{31} & X_{32} & X_{33} \\ X_{41} & X_{42} & X_{43} \end{bmatrix}, \quad X_+ = X_+^* = \begin{bmatrix} X_{14} \\ X_{24} \\ X_{34} \\ X_{44} \end{bmatrix}.$$

Appendix B

Far-Field Boundary Conditions of the Adjoint Equations

For subsonic inflow, the boundary values of λ are obtained as follows:

$$\begin{aligned}\lambda_1 &= \frac{2a^2 - (\gamma - 1)V^2}{2a^2} R_1 - \frac{\vec{V} \cdot \vec{s}}{\rho} R_2 - \frac{(\gamma - 1)V^2 - 2a\vec{V} \cdot \vec{n}}{2\rho a} R_4, \\ \lambda_2 &= \frac{(\gamma - 1)u}{a^2} R_1 + \frac{n_y}{\rho} R_2 + \frac{n_x a + (\gamma - 1)u}{\rho a} R_4, \\ \lambda_3 &= \frac{(\gamma - 1)v}{a^2} R_1 - \frac{n_x}{\rho} R_2 + \frac{n_y a + (\gamma - 1)v}{\rho a} R_4, \\ \lambda_4 &= \frac{1 - \gamma}{a^2} R_1 - \frac{\gamma - 1}{\rho a} R_4.\end{aligned}$$

Similarly, for subsonic outflow,

$$\begin{aligned}\lambda_1 &= -\frac{(\gamma - 1)V^2 - 2a\vec{V} \cdot \vec{n}}{2\rho a} R_4, \\ \lambda_2 &= \frac{n_x a + (\gamma - 1)u}{\rho a} R_4, \\ \lambda_3 &= \frac{n_y a + (\gamma - 1)v}{\rho a} R_4, \\ \lambda_4 &= -\frac{\gamma - 1}{\rho a} R_3.\end{aligned}$$

In the above equations, R_1 , R_2 , and R_4 are computed from the characteristic relations, such that

$$\begin{aligned} R_1 &= \lambda_{\Omega_1} + u \lambda_{\Omega_2} + v \lambda_{\Omega_3} + \frac{V^2}{2} \lambda_{\Omega_4}, \\ R_2 &= \rho(n_y \lambda_{\Omega_2} - n_x \lambda_{\Omega_3}) + (\rho \vec{V} \cdot \vec{s}) \lambda_{\Omega_4}, \\ R_4 &= -\frac{\rho}{2a} \lambda_{\Omega_1} - \frac{\rho}{2a}(u - n_x a) \lambda_{\Omega_2} - \frac{\rho}{2a}(v - n_y a) \lambda_{\Omega_3}, \\ &\quad - \left(\frac{\rho V^2}{4a} - \frac{\rho \vec{V} \cdot \vec{n}}{2} + \frac{\rho a}{2(\gamma - 1)} \right) \lambda_{\Omega_4}, \end{aligned}$$

where λ_{Ω} are obtained by extrapolating the values of λ from inside the domain towards the far-field boundaries.

Appendix C

Formulas Related to the Correction of the Far-Field Boundary Condition

In HITASK, the corrected free-stream velocity components are determined according to

$$u_{\infty}^* = u_{\infty} + f \sin \theta,$$

$$v_{\infty}^* = v_{\infty} - f \cos \theta,$$

where θ is the angle between the free stream direction and the line connecting the far-field boundary point, (x, y) , with the quarter chord point, $(0.25, 0)$,

$$\cos \theta = \frac{x - 0.25}{r_o},$$

$$\sin \theta = \frac{y}{r_o},$$

assuming a unit chord. The factor r_o is defined as

$$r_o = \sqrt{(x - 0.25)^2 + y^2}.$$

The factor f is defined as

$$f = \frac{C_l V_{\infty} \beta}{4\pi r_o (1 - M_{\infty}^2 \sin^2(\theta - \alpha))},$$

with $\beta = \sqrt{1 - M_{\infty}^2}$. The speed of sound is corrected according to

$$a_{\infty}^* = \sqrt{H_{\infty} - \frac{1}{2}(u_{\infty}^{*2} + v_{\infty}^{*2})},$$

where the enthalpy H_{∞} is defined as

$$H_{\infty} = \frac{\gamma p_{\infty}}{\rho_{\infty}} + \frac{1}{2}(u_{\infty}^2 + v_{\infty}^2).$$

From the above formulas, one can derive

$$\begin{aligned}\frac{\partial u_{\infty}^*}{\partial C_l} &= \frac{\sin \theta V_{\infty} \beta}{4\pi r_o(1 - M_{\infty}^2 \sin^2(\theta - \alpha))}, \\ \frac{\partial v_{\infty}^*}{\partial C_l} &= \frac{-\cos \theta V_{\infty} \beta}{4\pi r_o(1 - M_{\infty}^2 \sin^2(\theta - \alpha))}, \\ \frac{\partial a_{\infty}^*}{\partial C_l} &= \frac{-1}{2a_{\infty}^*} \frac{V_{\infty} \beta (u_{\infty}^* \sin \theta - v_{\infty}^* \cos \theta)}{4\pi r_o(1 - M_{\infty}^2 \sin^2(\theta - \alpha))},\end{aligned}$$

so that the partial derivatives of the corrected Riemann invariants with respect to C_l can be obtained,

$$\begin{aligned}\frac{\partial W_2^*}{\partial C_l} &= n_y \frac{\partial u_{\infty}^*}{\partial C_l} - n_x \frac{\partial v_{\infty}^*}{\partial C_l}, \\ \frac{\partial W_4^*}{\partial C_l} &= n_x \frac{\partial u_{\infty}^*}{\partial C_l} + n_y \frac{\partial v_{\infty}^*}{\partial C_l} - \frac{2}{(\gamma - 1)} \frac{\partial a_{\infty}^*}{\partial C_l}.\end{aligned}$$

The matrix product (CX_+^*) needed for evaluating C_{∞} in is obtained as follows. Distinction must be made between inflow or outflow:

- For subsonic inflow, (CX_+^*) is a matrix of dimension (4×2) , of which the elements are

$$\begin{aligned}(CX_+^*)_{11} &= 0, \\ (CX_+^*)_{21} &= \rho(n_x \vec{V} \cdot \vec{s} - v), \\ (CX_+^*)_{31} &= -\rho n_x \vec{V} \cdot \vec{n}, \\ (CX_+^*)_{41} &= \rho n_x (u \vec{V} \cdot \vec{s} - v \vec{V} \cdot \vec{n}) + \rho uv, \\ (CX_+^*)_{12} &= -\frac{\rho(\vec{V} \cdot \vec{n} - a)}{2a}, \\ (CX_+^*)_{22} &= -\frac{\rho(u - n_x a)(\vec{V} \cdot \vec{n} + a)}{2a}, \\ (CX_+^*)_{32} &= -\frac{\rho(v \vec{V} \cdot \vec{n} - n_x a \vec{V} \cdot \vec{s} - a(2v - n_y a))}{2a}, \\ (CX_+^*)_{42} &= -\frac{\rho a(\vec{V} \cdot \vec{n} - a)}{2(\gamma - 1)} - \frac{\rho V^2 \vec{V} \cdot \vec{n}}{4a} + \frac{\rho n_x^2 (u^2 - v^2)}{2} \\ &\quad + \frac{\rho(u^2 + 3v^2 + 4n_x n_y uv)}{4}.\end{aligned}$$

- For subsonic outflow, (CX_+^*) is a matrix of dimension (4×1) , of which the elements are

$$\begin{aligned}
 (CX_+^*)_{11} &= -\frac{\rho(\vec{V} \cdot \vec{n} - a)}{2a}, \\
 (CX_+^*)_{21} &= -\frac{\rho(u - n_x a)(\vec{V} \cdot \vec{n} + a)}{2a}, \\
 (CX_+^*)_{31} &= -\frac{\rho(v\vec{V} \cdot \vec{n} - n_x a\vec{V} \cdot \vec{s} - a(2v - n_y a))}{2a}, \\
 (CX_+^*)_{41} &= -\frac{\rho a(\vec{V} \cdot \vec{n} - a)}{2(\gamma - 1)} - \frac{\rho V^2 \vec{V} \cdot \vec{n}}{4a} + \frac{\rho n_x^2(u^2 - v^2)}{2} \\
 &\quad + \frac{\rho(u^2 + 3v^2 + 4n_x n_y uv)}{4}.
 \end{aligned}$$

Appendix D

Variation of the Viscous Term

The variations considered here are due only to the variation \mathbf{Q}' . The assumptions used are

- The variation of μ with respect to the variation of \mathbf{Q} is neglected.
- The viscous terms on the far-field boundary are dropped.

The viscous term is written as

$$\mathcal{J} = \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}_v) d\Omega.$$

The variation of \mathcal{J} due to \mathbf{Q}' can be expressed as

$$\delta \mathcal{J} = \int_{\Omega} \boldsymbol{\lambda} \cdot (\vec{\nabla} \cdot \vec{\mathbf{F}}'_v) d\Omega.$$

Integration by parts yields

$$\delta \mathcal{J} = - \int_{S_a} \boldsymbol{\lambda} \cdot (\vec{\mathbf{F}}'_v \cdot \vec{n}) dS - \int_{\Omega} \vec{\mathbf{F}}'_v \cdot \vec{\nabla} \boldsymbol{\lambda} d\Omega. \quad (\text{D.1})$$

It is noted that the unit normal vector points toward the flow domain. Recalling equation (5.7) and introducing $\delta \mathcal{J}_1$, $\delta \mathcal{J}_2$, $\delta \mathcal{J}_3$, and $\delta \mathcal{J}_4$ as follows,

$$\delta \mathcal{J}_1 = \int_{S_a} \left(\lambda_2 (n_x \tau'_{xx} + n_y \tau'_{xy}) + \lambda_3 (n_x \tau'_{xy} + n_y \tau'_{yy}) \right) dS, \quad (\text{D.2})$$

$$\begin{aligned} \delta \mathcal{J}_2 = \int_{S_a} \lambda_4 \left(n_x (\tau'_{xx} u + \tau'_{xy} v + \tau_{xx} u' + \tau_{xy} v' - q'_x) \right. \\ \left. + n_y (\tau'_{xy} u + \tau'_{yy} v + \tau_{xy} u' + \tau_{yy} v' + q'_x) \right) dS, \end{aligned} \quad (\text{D.3})$$

$$\delta \mathcal{J}_3 = \int_{\Omega} \left[\tau'_{xx} \frac{\partial \lambda_2}{\partial x} + \tau'_{xy} \frac{\partial \lambda_2}{\partial y} + \tau'_{xy} \frac{\partial \lambda_3}{\partial x} + \tau'_{yy} \frac{\partial \lambda_3}{\partial y} \right] d\Omega, \quad (\text{D.4})$$

$$\begin{aligned} \delta \mathcal{J}_4 = \int_{\Omega} \left[(\tau'_{xx} u + \tau_{xx} u' + \tau'_{xy} v + \tau_{xy} v' - q'_x) \frac{\partial \lambda_4}{\partial x} \right. \\ \left. + (\tau'_{xy} u + \tau_{xy} u' + \tau'_{yy} v + \tau_{yy} v' - q'_y) \frac{\partial \lambda_4}{\partial y} \right] d\Omega. \end{aligned} \quad (\text{D.5})$$

equation (D.1) can be written as

$$\delta \mathcal{J} = -\delta \mathcal{J}_1 - \delta \mathcal{J}_2 - \delta \mathcal{J}_3 - \delta \mathcal{J}_4. \quad (\text{D.6})$$

Now, the continuity equation is considered:

$$\vec{\nabla} \cdot \rho \vec{V} = 0.$$

Taking the no-slip boundary condition on S_a into account, the continuity equation can be expressed as

$$\vec{\nabla} \cdot \vec{V} = \frac{\partial V_n}{\partial n} = 0 \quad \text{on } S_a,$$

Substituting this into the definition for τ_{xx} , τ_{xy} , and τ_{yy} of equation (5.8), (5.10), and (5.9), gives the following relations

$$\tau_{xx} = 2 n_x n_y \tau_w, \quad (\text{D.7})$$

$$\tau_{yy} = -2 n_x n_y \tau_w, \quad (\text{D.8})$$

$$\tau_{xy} = (n_y^2 - n_x^2) \tau_w. \quad (\text{D.9})$$

Firstly, the variation $\delta \mathcal{J}_1$ can be worked out by substituting equations (D.7)–(D.9) into (D.2), giving

$$\begin{aligned} \delta \mathcal{J}_1 &= \int_{S_a} (n_y \lambda_2 - n_x \lambda_3) \tau'_w dS \\ &= \int_{S_a} (\vec{\lambda} \cdot \vec{s}) \tau'_w dS, \end{aligned} \quad (\text{D.10})$$

where $\vec{\lambda}$ is an adjoint velocity vector with the Cartesian components λ_2 and λ_3 :

$$\vec{\lambda} = \begin{pmatrix} \lambda_2 \\ \lambda_3 \end{pmatrix}.$$

Secondly, the variation $\delta \mathcal{J}_2$ is worked out by substituting the no-slip boundary conditions (5.22)–(5.23), equations (5.20)–(5.21), and equations (D.7)–(D.9) into (D.3), giving

$$\delta \mathcal{J}_2 = \int_{S_a} \lambda_4 \left(\tau_w (\vec{V}' \cdot \vec{s}) + \gamma \frac{\mu}{\text{Pr}} \vec{\nabla} \epsilon' \cdot \vec{n} \right) dS. \quad (\text{D.11})$$

Next, the variation of $\delta \mathcal{J}_3$ is obtained as follows. Equations (5.8)–(5.9) are substituted into equation (D.4) which yields

$$\begin{aligned} \delta \mathcal{J}_3 &= \int_{\Omega} \left[\left(\frac{\partial \lambda_2}{\partial x} + \frac{\partial \lambda_3}{\partial y} \right) l (\vec{\nabla} \cdot \vec{V}') \right. \\ &\quad \left. + \mu \left[2 \left(\frac{\partial u'}{\partial x} \frac{\partial \lambda_2}{\partial x} + \frac{\partial v'}{\partial y} \frac{\partial \lambda_3}{\partial y} \right) + \left(\frac{\partial \lambda_2}{\partial y} + \frac{\partial \lambda_3}{\partial x} \right) \left(\frac{\partial v'}{\partial x} + \frac{\partial u'}{\partial y} \right) \right] \right] d\Omega. \end{aligned}$$

Rearranging the coefficients of μ gives

$$\begin{aligned} \delta \mathcal{J}_3 = \int_{\Omega} \left[\left(\frac{\partial \lambda_2}{\partial x} + \frac{\partial \lambda_3}{\partial y} \right) l(\vec{\nabla} \cdot \vec{V}') \right. \\ + \mu \left(\frac{\partial \lambda_2}{\partial x} \frac{\partial u'}{\partial x} + \frac{\partial \lambda_2}{\partial y} \frac{\partial u'}{\partial y} \right) + \mu \left(\frac{\partial \lambda_3}{\partial x} \frac{\partial v'}{\partial x} + \frac{\partial \lambda_3}{\partial y} \frac{\partial v'}{\partial y} \right) \\ \left. + \mu \left(\frac{\partial \lambda_2}{\partial x} \frac{\partial u'}{\partial x} + \frac{\partial \lambda_3}{\partial x} \frac{\partial u'}{\partial y} \right) + \mu \left(\frac{\partial \lambda_2}{\partial x} \frac{\partial v'}{\partial x} + \frac{\partial \lambda_3}{\partial y} \frac{\partial v'}{\partial y} \right) \right] d\Omega. \end{aligned}$$

This can be written in a compact form as

$$\begin{aligned} \delta \mathcal{J}_3 = \int_{\Omega} \left[(\vec{\nabla} \cdot \vec{\lambda}) l(\vec{\nabla} \cdot \vec{V}') \right. \\ \left. + \mu \left(\vec{\nabla} \lambda_2 \cdot \vec{\nabla} u' + \vec{\nabla} \lambda_3 \cdot \vec{\nabla} v' + \frac{\partial \vec{\lambda}}{\partial x} \cdot \vec{\nabla} u' + \frac{\partial \vec{\lambda}}{\partial y} \cdot \vec{\nabla} v' \right) \right] d\Omega. \end{aligned}$$

Using the following vector identities,

$$\begin{aligned} (\vec{\nabla} \cdot \vec{\lambda}) l(\vec{\nabla} \cdot \vec{V}') &= \vec{\nabla} \cdot (l(\vec{\nabla} \cdot \vec{\lambda}) \vec{V}') - \vec{\nabla} (l(\vec{\nabla} \cdot \vec{\lambda})) \cdot \vec{V}', \\ \mu \vec{\nabla} \lambda_2 \cdot \vec{\nabla} u' &= \vec{\nabla} \cdot \mu u' \vec{\nabla} \lambda_2 - u' (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_2), \\ \mu \vec{\nabla} \lambda_3 \cdot \vec{\nabla} v' &= \vec{\nabla} \cdot \mu v' \vec{\nabla} \lambda_3 - v' (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_3), \\ \mu \frac{\partial \vec{\lambda}}{\partial x} \cdot \vec{\nabla} u' &= \vec{\nabla} \cdot \mu u' \frac{\partial \vec{\lambda}}{\partial x} - u' \left(\vec{\nabla} \cdot \mu \frac{\partial \vec{\lambda}}{\partial x} \right), \\ \mu \frac{\partial \vec{\lambda}}{\partial y} \cdot \vec{\nabla} v' &= \vec{\nabla} \cdot \mu v' \frac{\partial \vec{\lambda}}{\partial y} - v' \left(\vec{\nabla} \cdot \mu \frac{\partial \vec{\lambda}}{\partial y} \right), \end{aligned}$$

one obtains

$$\begin{aligned} \delta \mathcal{J}_3 = \int_{\Omega} \left[\vec{\nabla} \cdot (l(\vec{\nabla} \cdot \vec{\lambda}) \vec{V}') + \vec{\nabla} \cdot \mu u' \vec{\nabla} \lambda_2 + \vec{\nabla} \cdot \mu v' \vec{\nabla} \lambda_3 \right. \\ \left. + \vec{\nabla} \cdot \mu u' \frac{\partial \vec{\lambda}}{\partial x} + \vec{\nabla} \cdot \mu v' \frac{\partial \vec{\lambda}}{\partial y} \right] d\Omega \\ - \int_{\Omega} \left[\left(\frac{\partial (l \vec{\nabla} \cdot \vec{\lambda})}{\partial x} + \vec{\nabla} \cdot \mu \vec{\nabla} \lambda_2 + \vec{\nabla} \cdot \mu \frac{\partial \vec{\lambda}}{\partial x} \right) u' \right. \\ \left. + \left(\frac{\partial (l \vec{\nabla} \cdot \vec{\lambda})}{\partial y} + \vec{\nabla} \cdot \mu \vec{\nabla} \lambda_3 + \vec{\nabla} \cdot \mu \frac{\partial \vec{\lambda}}{\partial y} \right) v' \right] d\Omega. \quad (\text{D.12}) \end{aligned}$$

The terms having the divergence form, which are collected under the first integral, are worked out using the Gauss theorem (with the integral over S_{∞} neglected):

$$\int_{\Omega} \vec{\nabla} \cdot (l(\vec{\nabla} \cdot \vec{\lambda}) \vec{V}') d\Omega = - \int_{S_a} l(\vec{\nabla} \cdot \vec{\lambda}) (\vec{V}' \cdot \vec{n}) dS,$$

$$\begin{aligned}
\int_{\Omega} \vec{\nabla} \cdot \mu u' \vec{\nabla} \lambda_2 d\Omega &= - \int_{S_a} \mu u' \vec{\nabla} \lambda_2 \cdot \vec{n} dS, \\
\int_{\Omega} \vec{\nabla} \cdot \mu v' \vec{\nabla} \lambda_3 d\Omega &= - \int_{S_a} \mu v' \vec{\nabla} \lambda_3 \cdot \vec{n} dS, \\
\int_{\Omega} \vec{\nabla} \cdot \mu u' \frac{\partial \vec{\lambda}}{\partial x} d\Omega &= - \int_{S_a} \mu u' \frac{\partial \vec{\lambda}}{\partial x} \cdot \vec{n} dS, \\
\int_{\Omega} \vec{\nabla} \cdot \mu v' \frac{\partial \vec{\lambda}}{\partial y} d\Omega &= - \int_{S_a} \mu v' \frac{\partial \vec{\lambda}}{\partial y} \cdot \vec{n} dS.
\end{aligned}$$

The following notations are introduced[†]

$$\Gamma_{xx} = l \vec{\nabla} \cdot \vec{\lambda} + 2\mu \frac{\partial \lambda_2}{\partial x}, \quad (\text{D.13})$$

$$\Gamma_{yy} = l \vec{\nabla} \cdot \vec{\lambda} + 2\mu \frac{\partial \lambda_3}{\partial y}, \quad (\text{D.14})$$

$$\Gamma_{xy} = \mu \left(\frac{\partial \lambda_3}{\partial x} + \frac{\partial \lambda_2}{\partial y} \right). \quad (\text{D.15})$$

After some manipulations, equation (D.12) can be written as

$$\begin{aligned}
\delta \mathcal{J}_3 = & - \int_{S_a} \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right) (\vec{V}' \cdot \vec{n}) \\
& + \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) (\vec{V}' \cdot \vec{s}) dS \\
& - \int_{\Omega} \left[\left(\frac{\partial \Gamma_{xx}}{\partial x} + \frac{\partial \Gamma_{xy}}{\partial y} \right) u' + \left(\frac{\partial \Gamma_{xy}}{\partial x} + \frac{\partial \Gamma_{yy}}{\partial y} \right) v' \right] d\Omega,
\end{aligned}$$

where H is the surface curvature.

Next, the variation $\delta \mathcal{J}_4$ (D.5) is worked out using equations (5.8)–(5.12). This gives

$$\begin{aligned}
\delta \mathcal{J}_4 = & \int_{\Omega} \left[\left(u \frac{\partial \lambda_4}{\partial x} + v \frac{\partial \lambda_4}{\partial y} \right) l(\vec{\nabla} \cdot \vec{V}') \right. \\
& + 2\mu \left(u \frac{\partial u'}{\partial x} \frac{\partial \lambda_4}{\partial x} + v \frac{\partial v'}{\partial y} \frac{\partial \lambda_4}{\partial y} \right) + \mu \left(\frac{\partial v'}{\partial x} + \frac{\partial u'}{\partial y} \right) \left(v \frac{\partial \lambda_4}{\partial x} + u \frac{\partial \lambda_4}{\partial y} \right) \\
& + \left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} \right) u' + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} \right) v' \\
& \left. + \gamma \frac{\mu}{Pr} \left(\frac{\partial e'}{\partial x} \frac{\partial \lambda_4}{\partial x} + \frac{\partial e'}{\partial y} \frac{\partial \lambda_4}{\partial y} \right) \right] d\Omega.
\end{aligned}$$

After some manipulations, one obtains

$$\delta \mathcal{J}_4 = \int_{\Omega} \left[(\vec{V} \cdot \vec{\nabla} \lambda_4) l(\vec{\nabla} \cdot \vec{V}') \right]$$

[†]These may be considered as elements of an adjoint stress tensor because of the close resemblance with τ_{xx} , τ_{xy} , and τ_{yy} .

$$\begin{aligned}
& +\mu u \left(\frac{\partial u'}{\partial x} \frac{\partial \lambda_4}{\partial x} + \frac{\partial u'}{\partial y} \frac{\partial \lambda_4}{\partial y} \right) + \mu v \left(\frac{\partial v'}{\partial x} \frac{\partial \lambda_4}{\partial x} + \frac{\partial v'}{\partial y} \frac{\partial \lambda_4}{\partial y} \right) \\
& + \mu \frac{\partial \lambda_4}{\partial x} \left(u \frac{\partial u'}{\partial x} + v \frac{\partial u'}{\partial y} \right) + \mu \frac{\partial \lambda_4}{\partial y} \left(u \frac{\partial v'}{\partial x} + v \frac{\partial v'}{\partial y} \right) \\
& + \left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} \right) u' + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} \right) v' \\
& + \gamma \frac{\mu}{\text{Pr}} \left(\frac{\partial e'}{\partial x} \frac{\partial \lambda_4}{\partial x} + \frac{\partial e'}{\partial y} \frac{\partial \lambda_4}{\partial y} \right) \Big] d\Omega.
\end{aligned}$$

A compact form can be written as follows,

$$\begin{aligned}
\delta \mathcal{J}_4 = \int_{\Omega} \Big[& (\vec{V} \cdot \vec{\nabla} \lambda_4) l(\vec{\nabla} \cdot \vec{V}') \\
& + \mu u \vec{\nabla} \lambda_4 \cdot \vec{\nabla} u' + \mu v \vec{\nabla} \lambda_4 \cdot \vec{\nabla} v' \\
& + \mu \frac{\partial \lambda_4}{\partial x} \vec{V} \cdot \vec{\nabla} u' + \mu \frac{\partial \lambda_4}{\partial y} \vec{V} \cdot \vec{\nabla} v' \\
& + \left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} \right) u' + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} \right) v' \\
& + \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} e' \cdot \vec{\nabla} \lambda_4) \Big] d\Omega.
\end{aligned} \tag{D.16}$$

The following vector identities are considered,

$$\begin{aligned}
(\vec{V} \cdot \vec{\nabla} \lambda_4) l(\vec{\nabla} \cdot \vec{V}') &= \vec{\nabla} \cdot (l(\vec{V} \cdot \vec{\nabla} \lambda_4) \vec{V}') - \vec{V}' \cdot \vec{\nabla} l(\vec{V} \cdot \vec{\nabla} \lambda_4), \\
\mu u \vec{\nabla} \lambda_4 \cdot \vec{\nabla} u' &= \vec{\nabla} \cdot \mu u u' \vec{\nabla} \lambda_4 - u' (\vec{\nabla} \cdot \mu u \vec{\nabla} \lambda_4), \\
\mu v \vec{\nabla} \lambda_4 \cdot \vec{\nabla} v' &= \vec{\nabla} \cdot \mu v v' \vec{\nabla} \lambda_4 - v' (\vec{\nabla} \cdot \mu v \vec{\nabla} \lambda_4), \\
\mu \frac{\partial \lambda_4}{\partial x} \vec{V} \cdot \vec{\nabla} u' &= \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial x} u' \vec{V} - u' \left(\vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial x} \vec{V} \right), \\
\mu \frac{\partial \lambda_4}{\partial y} \vec{V} \cdot \vec{\nabla} v' &= \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial y} v' \vec{V} - v' \left(\vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial y} \vec{V} \right), \\
\gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} e' \cdot \vec{\nabla} \lambda_4) &= \frac{\gamma}{\text{Pr}} (\vec{\nabla} \cdot \mu e' \vec{\nabla} \lambda_4 - e' (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4)),
\end{aligned}$$

for obtaining

$$\begin{aligned}
\delta \mathcal{J}_4 = \int_{\Omega} \Big[& \vec{\nabla} \cdot (l(\vec{V} \cdot \vec{\nabla} \lambda_4) \vec{V}') + \vec{\nabla} \cdot \mu u u' \vec{\nabla} \lambda_4 + \vec{\nabla} \cdot \mu v v' \vec{\nabla} \lambda_4 \\
& + \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial x} u' \vec{V} + \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial y} v' \vec{V} + \frac{\gamma}{\text{Pr}} \vec{\nabla} \cdot \mu e' \vec{\nabla} \lambda_4 \Big] d\Omega \\
& + \int_{\Omega} \Big[\left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial}{\partial x} (l \vec{V} \cdot \vec{\nabla} \lambda_4) \right. \\
& \left. - \vec{\nabla} \cdot \mu u \vec{\nabla} \lambda_4 - \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial x} \vec{V} \right) u' \\
& \left. - \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial}{\partial y} (l \vec{V} \cdot \vec{\nabla} \lambda_4) \right. \right. \\
& \left. \left. - \vec{\nabla} \cdot \mu v \vec{\nabla} \lambda_4 - \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial y} \vec{V} \right) v' \right] d\Omega.
\end{aligned}$$

$$\begin{aligned}
& + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial}{\partial y} (l \vec{V} \cdot \vec{\nabla} \lambda_4) \right. \\
& \quad \left. - \vec{\nabla} \cdot \mu v \vec{\nabla} \lambda_4 - \vec{\nabla} \cdot \mu \frac{\partial \lambda_4}{\partial y} \vec{V} \right) v' \\
& \quad - \frac{\gamma}{\text{Pr}} (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) e' \Big] d\Omega.
\end{aligned}$$

Applying the Gauss theorem and the no-slip boundary condition for the first integral, and introducing Ψ_{xx} , Ψ_{xy} , and Ψ_{yy} as

$$\Psi_{xx} = (l + 2\mu)u \frac{\partial \lambda_4}{\partial x} + lv \frac{\partial \lambda_4}{\partial y}, \quad (\text{D.17})$$

$$\Psi_{yy} = (l + 2\mu)v \frac{\partial \lambda_4}{\partial y} + lu \frac{\partial \lambda_4}{\partial x}, \quad (\text{D.18})$$

$$\Psi_{xy} = \mu \left(u \frac{\partial \lambda_4}{\partial y} + v \frac{\partial \lambda_4}{\partial x} \right), \quad (\text{D.19})$$

gives

$$\begin{aligned}
\delta \mathcal{J}_4 = & - \int_{S_a} \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' dS \\
& + \int_{\Omega} \left[\left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial \Psi_{xx}}{\partial x} - \frac{\partial \Psi_{xy}}{\partial y} \right) u' \right. \\
& \quad + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial \Psi_{xy}}{\partial x} - \frac{\partial \Psi_{yy}}{\partial y} \right) v' \\
& \quad \left. - \frac{\gamma}{\text{Pr}} (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) e' \right] d\Omega.
\end{aligned} \quad (\text{D.20})$$

Recalling that

$$p = (\gamma - 1)\rho e,$$

and the speed of sound

$$a = \left(\gamma \frac{p}{\rho} \right)^{\frac{1}{2}},$$

one may write

$$\frac{\gamma}{\text{Pr}} (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) e' = \left(\frac{a^2 (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4)}{(\gamma - 1) \text{Pr}} \right) \left(\frac{p'}{p} - \frac{\rho'}{\rho} \right).$$

Substituting this into (D.20) gives

$$\begin{aligned}
\delta \mathcal{J}_4 = & - \int_{S_a} \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' dS \\
& + \int_{\Omega} \left[\left(\tau_{xx} \frac{\partial \lambda_4}{\partial x} + \tau_{xy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial \Psi_{xx}}{\partial x} - \frac{\partial \Psi_{xy}}{\partial y} \right) u' \right. \\
& \quad + \left(\tau_{xy} \frac{\partial \lambda_4}{\partial x} + \tau_{yy} \frac{\partial \lambda_4}{\partial y} - \frac{\partial \Psi_{xy}}{\partial x} - \frac{\partial \Psi_{yy}}{\partial y} \right) v' \\
& \quad \left. + \left(\frac{a^2 (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4)}{(\gamma - 1) \text{Pr}} \right) \left(\frac{p'}{p} - \frac{\rho'}{\rho} \right) \right] d\Omega.
\end{aligned} \quad (\text{D.21})$$

Substitution of equations (D.10), (D.11), (D.16), and (D.21) into (D.6) yields

$$\begin{aligned}
 \delta \mathcal{J} = & - \int_{S_a} \left[(\vec{\lambda} \cdot \vec{s}) \tau'_w + \lambda_4 \left(\tau_w (\vec{V}' \cdot \vec{s}) + \gamma \frac{\mu}{\text{Pr}} \vec{\nabla} e' \cdot \vec{n} \right) \right. \\
 & - \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right) (\vec{V}' \cdot \vec{n}) \\
 & \left. - \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) (\vec{V}' \cdot \vec{s}) - \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' \right] dS \\
 & + \int_{\Omega} \left[\left(\frac{\partial \Gamma_{xx}}{\partial x} + \frac{\partial \Gamma_{xy}}{\partial y} - \tau_{xx} \frac{\partial \lambda_4}{\partial x} - \tau_{xy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xx}}{\partial x} + \frac{\partial \Psi_{xy}}{\partial y} \right) u' \right. \\
 & + \left(\frac{\partial \Gamma_{xy}}{\partial x} + \frac{\partial \Gamma_{yy}}{\partial y} - \tau_{xy} \frac{\partial \lambda_4}{\partial x} - \tau_{yy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xy}}{\partial x} + \frac{\partial \Psi_{yy}}{\partial y} \right) v' \\
 & \left. + \left(\frac{a^2 (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4)}{(\gamma - 1) \text{Pr}} \right) \left(\frac{p'}{p} - \frac{\rho'}{\rho} \right) \right] d\Omega. \quad (\text{D.22})
 \end{aligned}$$

The domain integral can be expressed in terms of the conservative flow variables by using the transformation

$$\mathbf{U}' = Y \mathbf{Q}',$$

where Y is the Jacobian of the primitive flow variables, $\mathbf{U} = (\rho \ u \ v \ p)^\top$, with respect to \mathbf{Q} ,

$$Y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -u & \frac{1}{\rho} & 0 & 0 \\ \frac{\rho}{-v} & \rho & \frac{1}{\rho} & 0 \\ \frac{\rho}{(\gamma - 1)(u^2 + v^2)} & 0 & \rho & \gamma - 1 \\ \frac{(\gamma - 1)(u^2 + v^2)}{2} & -(\gamma - 1)u & -(\gamma - 1)v & \gamma - 1 \end{bmatrix}. \quad (\text{D.23})$$

The coefficients of \mathbf{U}' in equation (D.22) can be collected into a vector \mathbf{K} defined as

$$\mathbf{K} = \begin{pmatrix} -\frac{a^2 (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) 1}{(\gamma - 1) \text{Pr} \ \rho} \\ \frac{\partial \Gamma_{xx}}{\partial x} + \frac{\partial \Gamma_{xy}}{\partial y} - \tau_{xx} \frac{\partial \lambda_4}{\partial x} - \tau_{xy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xx}}{\partial x} + \frac{\partial \Psi_{xy}}{\partial y} \\ \frac{\partial \Gamma_{xy}}{\partial x} + \frac{\partial \Gamma_{yy}}{\partial y} - \tau_{xy} \frac{\partial \lambda_4}{\partial x} - \tau_{yy} \frac{\partial \lambda_4}{\partial y} + \frac{\partial \Psi_{xy}}{\partial x} + \frac{\partial \Psi_{yy}}{\partial y} \\ \frac{a^2 (\vec{\nabla} \cdot \mu \vec{\nabla} \lambda_4) 1}{(\gamma - 1) \text{Pr} \ p} \end{pmatrix}. \quad (\text{D.24})$$

Finally, equation (D.22) is written as

$$\delta \mathcal{J} = - \int_{S_a} \left[(\vec{\lambda} \cdot \vec{s}) \tau'_w + \lambda_4 \left(\tau_w (\vec{V}' \cdot \vec{s}) + \gamma \frac{\mu}{\text{Pr}} \vec{\nabla} e' \cdot \vec{n} \right) \right]$$

$$\begin{aligned}
& - \left(l(\vec{\nabla} \cdot \vec{\lambda}) + 2\mu \frac{\partial \lambda_n}{\partial n} \right) (\vec{V}' \cdot \vec{n}) \\
& - \mu \left(\frac{\partial \lambda_s}{\partial n} + \frac{\partial \lambda_n}{\partial s} - H(\vec{\lambda} \cdot \vec{s}) \right) (\vec{V}' \cdot \vec{s}) - \gamma \frac{\mu}{\text{Pr}} (\vec{\nabla} \lambda_4 \cdot \vec{n}) e' \Big] dS \\
& + \int_{\Omega} Y^T \mathbf{K} \cdot \mathbf{Q}' d\Omega.
\end{aligned} \tag{D.25}$$

Appendix E

Adjoint Convective Fluxes

The inviscid adjoint fluxes follow from the product of $A^T \lambda$. The x -component of the fluxes are obtained as

$$\mathbf{Z}_x = - \begin{bmatrix} \left(\frac{(\gamma-1)V^2}{2} - u^2 \right) \lambda_2 - uv\lambda_3 + \left(\frac{(\gamma-2)V^2}{2} - \frac{a^2}{\gamma-1} \right) u\lambda_4 \\ \lambda_1 - (\gamma-3)u\lambda_2 + v\lambda_3 + \left(\frac{3u^2+v^2}{2} + \frac{a^2}{\gamma-1} - \gamma u^2 \right) \lambda_4 \\ (1-\gamma)v\lambda_2 + u\lambda_3 - (\gamma-1)uv\lambda_4 \\ (\gamma-1)\lambda_2 + \gamma u\lambda_4 \end{bmatrix}$$

while the y -component is

$$\mathbf{Z}_y = - \begin{bmatrix} -uv\lambda_2 + \left(\frac{(\gamma-1)V^2}{2} - v^2 \right) \lambda_3 + \left(\frac{(\gamma-2)V^2}{2} - \frac{a^2}{\gamma-1} \right) v\lambda_4 \\ v\lambda_2 - (\gamma-1)u\lambda_3 - (\gamma-1)uv\lambda_4 \\ \lambda_1 + u\lambda_2 - (\gamma-3)v\lambda_3 + \left(\frac{u^2+3v^2}{2} + \frac{a^2}{\gamma-1} - \gamma v^2 \right) \lambda_4 \\ (\gamma-1)\lambda_3 + \gamma v\lambda_4 \end{bmatrix}$$

Appendix F

Adjoint Viscous Fluxes

The x -component of the adjoint viscous fluxes can be derived by collecting the terms under the operator $\partial/\partial x$ from the product of $Y^T \mathbf{K}$ in Appendix D. This gives

$$\mathbf{Z}_x = \begin{bmatrix} -\frac{1}{\rho}(u\Phi_{xx} + v\Phi_{xy}) + \frac{a^2\mu}{(\gamma-1)\text{Pr}} \left(\frac{(\gamma-1)V^2}{2p} - \frac{1}{\rho} \right) \frac{\partial\lambda_4}{\partial x} \\ \frac{1}{\rho}\Phi_{xx} - \frac{a^2\mu u}{p\text{Pr}} \frac{\partial\lambda_4}{\partial x} \\ \frac{1}{\rho}\Phi_{xy} - \frac{a^2\mu v}{p\text{Pr}} \frac{\partial\lambda_4}{\partial x} \\ \frac{a^2\mu}{p\text{Pr}} \frac{\partial\lambda_4}{\partial x} \end{bmatrix}$$

where

$$\Phi_{xx} = \Gamma_{xx} - \tau_{xx}\lambda_4 + \Psi_{xx}$$

$$\Phi_{xy} = \Gamma_{xy} - \tau_{xy}\lambda_4 + \Psi_{xy}$$

$$\Phi_{yy} = \Gamma_{yy} - \tau_{yy}\lambda_4 + \Psi_{yy}$$

The y -component is obtained as follows,

$$\mathbf{Z}_y = \begin{bmatrix} -\frac{1}{\rho}(u\Phi_{xy} + v\Phi_{yy}) + \frac{a^2\mu}{(\gamma-1)\text{Pr}} \left(\frac{(\gamma-1)V^2}{2p} - \frac{1}{\rho} \right) \frac{\partial\lambda_4}{\partial y} \\ \frac{1}{\rho}\Phi_{xy} - \frac{a^2\mu u}{p\text{Pr}} \frac{\partial\lambda_4}{\partial y} \\ \frac{1}{\rho}\Phi_{yy} - \frac{a^2\mu v}{p\text{Pr}} \frac{\partial\lambda_4}{\partial y} \\ \frac{a^2\mu}{p\text{Pr}} \frac{\partial\lambda_4}{\partial y} \end{bmatrix}$$

Appendix G

Variations in Distributed Parameter Systems

This appendix is based on Refs. [22] and [10], which should be consulted for the detailed descriptions as these are not presented here. Figure G.1 shows a domain Ω which is mapped onto Ω_t following a transformation $T(\mathbf{x}, t)$ defined by ([22]):

$$T : \mathbf{x} \rightarrow \mathbf{x}_t(\mathbf{x}), \quad \mathbf{x} \in \Omega,$$

such that

$$\mathbf{x}_t = T(\mathbf{x}, t), \tag{G.1}$$

$$\Omega_t \equiv T(\Omega, t). \tag{G.2}$$

The mapping can be considered as a scheme for modifying the shape of the boundary S to one of S_t due to the deformation of Ω following a certain path in time t . The parameter t is thus conceived as taking the role of time with which the notion of *deformation velocity* $\vec{\omega}$ at a point \mathbf{x}_t can be defined:

$$\vec{\omega}(\mathbf{x}_t, t) = \frac{d\mathbf{x}_t}{dt} = \frac{\partial T(\mathbf{x}, t)}{\partial t}. \tag{G.3}$$

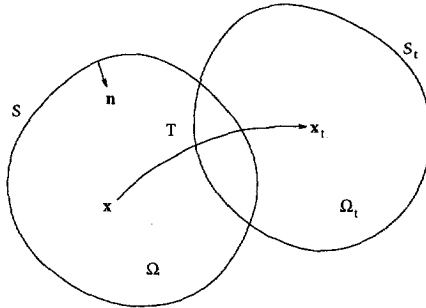


Figure G.1: Transformation of Domain Ω

The Taylor series expansion for T around $t = 0$ can be written as

$$T(\mathbf{x}, t) = T(\mathbf{x}, 0) + t \frac{\partial T}{\partial t}(\mathbf{x}, 0) + \dots,$$

which, after neglecting the higher order terms, becomes

$$T(\mathbf{x}, t) = \mathbf{x} + t \vec{\omega}(\mathbf{x}, 0). \quad (\text{G.4})$$

Substituting this into equation (G.1) yields

$$\mathbf{x}_t = \mathbf{x} + t \vec{\omega}(\mathbf{x}, 0).$$

Now, it is supposed that a vector variable θ , which varies with time, controls the deformation of the domain, such that at a time t :

$$\mathbf{x}_t = \mathbf{x}_t(\theta_t).$$

If the change of θ per unit time is defined as $\delta\theta$, then

$$\theta_t = \theta + t \delta\theta.$$

Combining the last three equations gives

$$\vec{\omega}(\mathbf{x}) = \frac{\partial \mathbf{x}}{\partial \theta} \cdot \delta\theta, \quad \text{at } t = 0.$$

This can be expressed in the form of

$$\vec{\omega}(\mathbf{x}) = \vec{\chi} \cdot \delta\theta,$$

where

$$\vec{\chi} = \begin{pmatrix} \chi_x \\ \chi_y \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial \theta} \end{pmatrix}.$$

Thus, the Cartesian components of $\vec{\omega}$ are defined as

$$\omega_x = \chi_x \cdot \delta\theta, \quad (\text{G.5})$$

$$\omega_y = \chi_y \cdot \delta\theta, \quad (\text{G.6})$$

while the normal and tangential components on the boundary S are determined by

$$\omega_n = \chi_n \cdot \delta\theta, \quad (\text{G.7})$$

$$\omega_s = \chi_s \cdot \delta\theta, \quad (\text{G.8})$$

where

$$\begin{pmatrix} \chi_n \\ \chi_s \end{pmatrix} = \begin{pmatrix} n_x & n_y \\ n_y & -n_x \end{pmatrix} \begin{pmatrix} \chi_x \\ \chi_y \end{pmatrix}.$$

For a scalar field u defined in Ω , the material derivative is defined as

$$\dot{u} = u' + \vec{\nabla}u \cdot \vec{\omega}. \quad (G.9)$$

The material derivative \dot{u} represents the total variation of u which consists of the local variation u' , where Ω is assumed fixed, and the convective variation $(\vec{\nabla}u \cdot \vec{\omega})$ due to the deformation of Ω with the speed $\vec{\omega}$. Accordingly, the material derivative of a function f of u defined in Ω can be written as

$$\dot{f} = \frac{\partial f}{\partial u} \cdot (u' + \vec{\nabla}u \cdot \vec{\omega}).$$

The material derivative also applies for geometric properties. Letting the Jacobian matrix of the deformation velocity be J ,

$$J = \begin{pmatrix} \frac{\partial \omega_x}{\partial x} & \frac{\partial \omega_x}{\partial y} \\ \frac{\partial \omega_y}{\partial x} & \frac{\partial \omega_y}{\partial y} \end{pmatrix} = J(\delta\theta)$$

the following material derivative formulas can be derived:

- The material derivative of a unit normal vector \vec{n} on the boundary is defined as

$$\dot{\vec{n}} = (\vec{n} \cdot J^T \cdot \vec{n})\vec{n} - J^T \cdot \vec{n},$$

which can be elaborated to yield

$$\dot{\vec{n}} = -(\omega_{n,s} + H\omega_s)\vec{s}, \quad (G.10)$$

where H is the surface curvature, \vec{s} is the unit tangential, and $\omega_{n,s}$ denotes the tangential derivative of ω_n :

$$\omega_{n,s} = \frac{\partial \chi_n}{\partial s} \cdot \delta\theta. \quad (G.11)$$

- The material derivative of a unit tangential vector \vec{s} on the boundary can be obtained from equation (G.10):

$$\dot{\vec{s}} = \begin{pmatrix} \dot{n}_y \\ -\dot{n}_x \end{pmatrix} = (\omega_{n,s} + H\omega_s)\vec{n}. \quad (G.12)$$

- The material derivative of a surface element ds is defined as

$$d\dot{S} = (\vec{\nabla} \cdot \vec{\omega} - \vec{n} \cdot J^T \cdot \vec{n}) dS,$$

which can be worked out to give

$$d\dot{S} = (\omega_{s,s} + H\omega_n) dS, \quad (G.13)$$

where $\omega_{s,s}$ denotes the tangential derivative of ω_s :

$$\omega_{s,s} = \frac{\partial \chi_s}{\partial s} \cdot \delta\theta. \quad (G.14)$$

- The material derivative of a volume element $d\Omega$ is defined as

$$d\dot{\Omega} = (\vec{\nabla} \cdot \vec{\omega}) d\Omega. \quad (\text{G.15})$$

Applying these, the material derivative of a domain functional defined as an integral of f over Ω , i.e.

$$\Phi_1 = \int_{\Omega} f d\Omega,$$

can be obtained as

$$\begin{aligned} \dot{\Phi}_1 &= \int_{\Omega} \dot{f} d\Omega + \int_{\Omega} f d\dot{\Omega} \\ &= \int_{\Omega} (\dot{f} + f(\vec{\nabla} \cdot \vec{\omega})) d\Omega \\ &= \int_{\Omega} (f' + \vec{\nabla} f \cdot \vec{\omega} + f(\vec{\nabla} \cdot \vec{\omega})) d\Omega \\ &= \int_{\Omega} (f' + \vec{\nabla} \cdot f\vec{\omega}) d\Omega, \end{aligned}$$

which, by the divergence theorem, becomes

$$\dot{\Phi}_1 = \int_{\Omega} f' d\Omega - \int_S f \omega_n dS. \quad (\text{G.16})$$

Also, the variation of a functional defined as an integral of f over S , i.e.

$$\Phi_2 = \int_S f dS,$$

can be written as

$$\dot{\Phi}_2 = \int_S \dot{f} dS + \int_S f d\dot{S},$$

which can be expanded as

$$\dot{\Phi}_2 = \int_S (f' + \vec{\nabla} f \cdot \vec{\omega} + f(\omega_{s,s} + H\omega_n)) dS. \quad (\text{G.17})$$

Curriculum Vitae

The author of this thesis was born in Yogyakarta, Indonesia, on July 27, 1964. After finishing Senior High School at "SMA Taruna Bakti Bandung" in 1982, he entered the Institute of Technology Bandung (ITB) to study in the sub-Department of Aeronautical Engineering of the Department of Mechanical Engineering. In August 1988 he graduated for the S1 degree in mechanical engineering from ITB after performing the "ingenieur"'s thesis work at the Faculty of Aerospace Engineering of TU Delft in the period of May 1987 - May 1988 under the supervision of Prof.ir. J.W. Slooff. After graduating, he joined the academic staff of the Department of Mechanical Engineering of ITB at the Laboratory of Aeronautical Engineering, with the task of giving several undergraduate courses in aerodynamics, until March 1991. In the period from March 1991 until August 1996, he was doing research on computational aerodynamic design methods at the Department of Theoretical Aerodynamics of the National Aerospace Laboratory (NLR) in Amsterdam, in the framework of Ph.D research resulting in this thesis.