DEVELOPMENT OF A DISCRETE ADJOINT-BASED AERODYNAMIC SHAPE OPTIMIZATION TOOL FOR NATURAL LAMINAR FLOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HALİL KAYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
AEROSPACE ENGINEERING

AUGUST 2020

Approval of the thesis:

DEVELOPMENT OF A DISCRETE ADJOINT-BASED AERODYNAMIC SHAPE OPTIMIZATION TOOL FOR NATURAL LAMINAR FLOWS

submitted by **HALİL KAYA** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Aerospace Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. İsmail Hakkı Tuncer
Head of the Department, **Aerospace Engineering**

Prof. Dr. İsmail Hakkı Tuncer
Supervisor, **Aerospace Engineering, METU**

Assoc. Prof. Dr. Hamdullah Yücel
Co-Supervisor, **Institute of Applied Mathematics, METU**

Examining Committee Members:

Prof. Dr. Yusuf Özyörük
Aerospace Engineering, METU

Prof. Dr. İsmail Hakkı Tuncer
Aerospace Engineering, METU

Prof. Dr. Sinan Eyi
Aerospace Engineering, METU

Prof. Dr. Ünver Kaynak
Aerospace Engineering, Yıldırım Beyazıt Uni.

Prof. Dr. Kürşad Melih Güleren
Flight Training, Eskişehir Technical Uni.

Date: 13.08.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Halil Kaya

Signature :

ABSTRACT

DEVELOPMENT OF A DISCRETE ADJOINT-BASED AERODYNAMIC SHAPE OPTIMIZATION TOOL FOR NATURAL LAMINAR FLOWS

Kaya, Halil
Doctor of Philosophy, Aerospace Engineering
Supervisor: Prof. Dr. İsmail Hakkı Tuncer
Co-Supervisor: Assoc. Prof. Dr. Hamdullah Yücel

August 2020, 130 pages

An adjoint-based aerodynamic shape optimization framework for natural laminar flows is developed. A Reynolds-Averaged Navier-Stokes flow solver with the Spalart-Allmaras turbulence model is coupled with the recently developed Bas-Cakmakcioglu transition model in order to predict laminar to turbulent transition onset. In the gradient-based optimization process, the sensitivity derivatives required by the optimization algorithm is obtained by the discrete adjoint method, which is developed for the in-house flow solver and implemented for natural laminar flow airfoils and wings. In the development of the discrete adjoint method, an automatic differentiation tool is employed to take the discrete derivative of the modules in the in-house flow solver heavily modified. The parametrization of the aerodynamic surface is realized by the Free-Form Deformation technique. The sensitivity derivatives with respect to design parameters, which are computed by the adjoint method, are validated with the finite-difference method. The success of the adjoint-based aerodynamic shape optimization methodology developed in this study is then demonstrated by optimizing aerodynamic characteristics of several airfoils and wings for compressible turbulent and natural laminar flows.

Keywords: Discrete Adjoint Method, Natural Laminar Flow, Aerodynamic Shape Optimization, Laminar to Turbulent Transition Model, Computational Fluid Dynamics

ÖZ

DOĞAL LAMİNAR AKIŞLAR İÇİN AYRIK ADJOINT TABANLI AERODİNAMİK ŞEKİL ENİYİLEME ARACI GELİŞTİRİLMESİ

Kaya, Halil
Doktora, Havacılık ve Uzay Mühendisliği
Tez Yöneticisi: Prof. Dr. İsmail Hakkı Tuncer
Ortak Tez Yöneticisi: Doç. Dr. Hamdullah Yücel

Ağustos 2020, 130 sayfa

Doğal laminar akışlar için adjoint tabanlı aerodinamik şekil eniyileme yapısı geliştirilmiştir. Laminar akıştan türbülanslı akışa geçiş başlangıcını tahmin etmek için, Spalart-Allmaras türbülans modeli içeren bir Reynolds-Ortalama Navier-Stokes akış çözücü yakın zamanda geliştirilmiş Baş-Çakmakçıoğlu geçiş modeli ile birleştirilmiştir. Gradyan tabanlı eniyileme işleminde, eniyileme algoritmasının gerektirdiği hassasiyet türevleri, kurum içi akış çözü için geliştirilen ve doğal laminar akış kanat kesitleri ve kanatları için uygulanan ayrık bir adjoint yöntem ile elde edilmiştir. Ayrık bitişik yöntemin geliştirilmesinde yeniden yazılan akış çözücüdeki modüllerin ayrık türevini almak için bir otomatik türev aracı kullanılmıştır. Aerodinamik dış yüzeyin parametrizasyonu serbest şekil deformasyon yöntemi ile gerçekleştirilmiştir. Adjoint yöntemle hesaplanan şekil parametrelerine göre hassasiyet türevleri, ilk olarak sonlu farklar yöntemi ile doğrulanmıştır. Bu çalışmada geliştirilen adjoint tabanlı aerodinamik şekil eniyileme yönteminin başarısı, sıkıştırılabilir türbülanslı ve doğal laminar akışlar için kanat ve kanat kesitlerinin eniyilenmesi ile gösterilmiştir.

Anahtar Kelimeler: Ayrık Adjoint Yöntem, Doğal Laminar Akış, Aerodinamik Şekil Eniyilemesi, Laminardan Türbülansa Geçiş Modeli, Hesaplamalı Akışkanlar Dinamiği

To My Mother

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Prof. Dr. İsmail Hakkı Tuncer for his encouragement, support, criticism, advice, and guidance throughout the research. I also wish to express my gratitude to my co-supervisor Assoc. Prof. Dr. Hamdullah Yücel for his guidance and technical assistance regarding the theory and mathematical background of the topic.

I would like to offer my special thanks to Hakan Tiftikçi for his time, guidance, and effort in the thesis study. He made many crucial contributions to the study, by both helping with the theory and implementation issues, and simply helping to track down the coding problems from time to time. His willingness to spare time for my questions and problems improved the study in many aspects.

My special thanks are extended to Dr. Emre Özkaya and Prof. Dr. Nicholas R. Gauger for their hospitality, aids, and the opportunity to work together at the Chair for the Scientific Computing at the Technical University of Kaiserslautern. I would also like to thank Tim Albring for his time to answer my questions.

I am particularly grateful to Dr. Erhan Tarhan and Ethem Hakan Orhan for their guidance and for making it possible for this study to be funded. I also wish to thank Hüseyin Yağcı. The possibilities and the assistance given by him has been a great help in accomplishing the study.

Assistance regarding the theory and implementation issues of the Bas-Cakmakcioglu transition model provided by Dr. Samet Çaka Çakmakcıoğlu and Dr. Onur Baş is greatly appreciated.

I owe my deepest gratitude to my close friends Orkun Şimsek, Senem Ayşe Haser, Dr. Erdem Ayan ve Hande Akalan for their support and friendship. I wish to thank Hüdai Alayli for his contributions and encouragement. I also would like to offer my special thanks to my brother Anıl Kaya for providing me with the delicious dinners and making life easier throughout the thesis.

I owe more than thanks to my family for their support and encouragement throughout my life.

Last but not least important, I would like to thank all my colleagues and the R&D management department of Turkish Aerospace for their support.

This work is partially funded by Turkish Aerospace and the Presidency of Defence Industries in the framework of the Researcher Training Program for Defence Industry.

TABLE OF CONTENTS

ABSTRACT v

ÖZ.. vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xii

LIST OF TABLES xv

LIST OF FIGURES.. xvi

LIST OF ABBREVIATIONS xx

LIST OF SYMBOLS.. xxii

CHAPTERS

1 INTRODUCTION.. 1

1.1 Adjoint Methods and Automatic Differentiation 3

1.2 Natural Laminar Flows.. 6

1.3 Motivation 8

1.4 Thesis Objective 10

1.5 Thesis Outline.. 10

2 METHOD 13

2.1 Reynolds-Averaged Navier-Stokes Flow Solver.. 13

2.1.1 Governing Equations 13

2.1.2 Spatial Discretization... 16

2.1.3 Geometrical Properties of a Control Volume 16

2.1.4 Discretization of Convective Fluxes... 18

2.1.5 Variable Reconstruction 20

2.1.6 Evaluation of Gradients of Flow Variables 20

2.1.7 Limiter Function for Variable Reconstruction 23

2.1.8 Discretization of Viscous Fluxes 25

2.1.9 Temporal Discretization 26

2.1.10 Time Step and Local Time Stepping 27

2.1.11 Turbulence Modelling 28

2.1.12 Spalart-Allmaras Turbulence Model 29

2.1.13 Bas-Cakmakcioglu Transition Model 31

2.1.14 Numerical Implementation and Parallelization 33

2.2 Sensitivity Analysis 36

2.2.1 Adjoint Equations 36

2.2.2 Automatic Differentiation 40

2.2.3 Computation of Flux Jacobian Matrices 47

2.2.4 Computation of Partial Sensitivities of Objective Function 50

2.2.5 Solution of the Adjoint System of Equations 51

2.3 Optimization Framework 53

2.3.1 Surface Parametrization and Grid Deformation 54

2.3.2 Grid Sensitivity 56

2.3.3 Optimization Algorithm 56

3 RESULTS AND DISCUSSION 59

3.1 Validation of RANS Solver 59

3.1.1 Flow over a Flat Plate 59

3.1.2 Flow over RAE 2822 Airfoil 62

3.1.3 Flow over Natural Laminar Airfoil 65

3.1.4 Flow over Onera M6 Wing 68
3.1.5 Flow over Low Aspect Ratio Laminar Wing 72
3.1.6 Flow over High Aspect Ratio Laminar Wing 77
3.2 Validation of Adjoint Solver 80
3.2.1 Drag Sensitivity at Fixed Lift: RAE 2822 80
3.2.2 Range Parameter Sensitivity: NLF(1)-0416 82
3.2.3 Drag Sensitivity at Fixed Lift: ONERA M6 84
3.2.4 Endurance Parameter Sensitivity: Low Aspect Ratio Wing 86
3.2.5 Endurance Parameter Sensitivity: High Aspect Ratio Wing 87
3.3 Optimization Studies 89
3.3.1 Drag Minimization at a Fixed Lift: RAE 2822 Airfoil 90
3.3.2 Maximization Range Parameter Maximization: NLF(1)-0.416 92
3.3.3 Drag Minimization at a Fixed Lift: ONERA M6 Wing 95
3.3.4 Endurance Parameter Maximization: Low Aspect Ratio Wing 98
3.3.5 Endurance Parameter Maximization: High Aspect Ratio Wing 103
4 CONCLUSION 107
4.1 Further Studies 109
REFERENCES 111
CURRICULUM VITAE 129

LIST OF TABLES

TABLES

Table 2.1 Typical latency values in modern computer architecture (Ajwani & Meyerhenke, 2010) 35

Table 3.1 The corrected flow conditions for test case 6 and test case 9 62

Table 3.2 Comparison of aerodynamic coefficients of ONERA M6 wing............. 71

Table 3.3 Geometrical properties of MDO505 wing 73

Table 3.4 The flow conditions of the cases.. 74

Table 3.5 Comparison of the transition onset at mid-span 76

Table 3.6 Geometrical properties of the experimental wing................................... 77

LIST OF FIGURES

FIGURES

Figure 2-1 A triangular and a quadrilateral element 17
Figure 2-2 Effect of limiter function 25
Figure 2-3 Parallel Efficiency 36
Figure 2-4 Sample Fortran code 42
Figure 2-5 Subroutine generated by *TAPENADE* in the forward mode 44
Figure 2-6 Subroutine generated by *TAPENADE* in reverse mode 46
Figure 2-7 Optimization flow chart 53
Figure 2-8 A sample deformation carried out by SU2_DEF employing FFD approach 55
Figure 3-1 Flat plate computational grid 60
Figure 3-2 Skin friction coefficient (Tu∞ 0.18%) 60
Figure 3-3 Skin friction coefficient (Tu∞ 3.0%) 61
Figure 3-4 Computational grid for RAE 2822 airfoil 63
Figure 3-5 Pressure distributions for case 6 (left), and case 9 (right) 63
Figure 3-6 Mach contours 64
Figure 3-7 Convergence history for case 6 64
Figure 3-7 Computational grid for NLF(1)-0416 airfoil 65
Figure 3-8 Variation of lift (left) and drag coefficients (right) 66
Figure 3-9 Transition locations of the boundary layer from laminar to turbulent flow; open symbols indicate orifices on which flow is laminar, and closed symbols indicate orfices on which flow is turbulent 67
Figure 3-10 Convergence history for NLF(1)-0416 airfoil at the angle of attack 6° 68
Figure 3-10 Computational grid for Onera M6 wing 69
Figure 3-11 Pressure coefficients at different wingspan stations 70
Figure 3-12 Mach contours 71
Figure 3-13 Convergence history for ONERA M6 analysis 72

Figure 3-13 The wing section of the MDO505 Morphing Wing........................... 73
Figure 3-14 Computational grid for MDO 505 wing.. 74
Figure 3-15 Pressure coefficient and turbulent working variable distribution at the midspan wing section (case 3) 75
Figure 3-16 Skin friction coefficients. The IR image of the experiment (Koreanschi, et al., 2017) (left) and numerical results illustrated on the IR image (right) 75
Figure 3-17 Convergence history for case 3 76
Figure 3-17 Computational grid for experimental wing 78
Figure 3-18 Transition locations along the wing span (Permission for the oil-flow image is not granted).. 78
Figure 3-19 The turbulent working variable ν and x-velocity u distribution at the midspan wing section.. 79
Figure 3-20 Convergence history; normalized continuity residual (left), change in the lift and drag coefficient after the iteration of 50,000 (right)............................ 79
Figure 3-21 FFD box and the control points generated on RAE 2822 airfoil 81
Figure 3-22 Sensitivity derivatives computed by the finite difference method and the adjoint method.. 82
Figure 3-23 FFD box and control points generated on NLF(1)-0416 airfoil.......... 83
Figure 3-24 Sensitivity derivatives computed by the finite difference and the adjoint method 83
Figure 3-25 FFD box and control points (blue spheres depict active control points) 84
Figure 3-26 Comparison of sensitivity derivatives computed by the finite difference and the adjoint methods at the root chord (red), at the mid-span chord (blue), at the tip chord (green)... 85
Figure 3-27 FFD box and control points (blue spheres depict active control points) 86

Figure 3-28 Comparison of sensitivity derivatives computed by the finite difference and the adjoint method at the root chord (red), at the mid-span chord (blue), at the tip chord (green) 87
Figure 3-29 FFD box and control points (blue spheres depict active control points) 88
Figure 3-30 Comparison of sensitivity derivatives computed by the finite difference and the adjoint derivatives at the root chord (red), at the mid-span chord (blue), at the tip chord (green) 89
Figure 3-31 The evolution of the objective function value along the optimization steps 90
Figure 3-32 The baseline and optimized profile 91
Figure 3-33 The pressure coefficient distributions of the baseline and optimized profile 91
Figure 3-34 Mach distributions 92
Figure 3-35 Evolution of cl/cd along the optimization steps 93
Figure 3-36 The baseline and optimized airfoils 93
Figure 3-37 Pressure coefficient distributions on the baseline and optimized airfoils 94
Figure 3-38 The pressure coefficient distributions of the baseline and optimized profile 94
Figure 3-39 Evolution of objective function value along optimization steps 96
Figure 3-40 The wing sections of the baseline wing and the optimized wing (at η=0.33 and 0.66) 96
Figure 3-41 Pressure coefficients for baseline geometry and optimized geometry 97
Figure 3-42 Mach contours at different wingspan sections 98
Figure 3-43 Evolution of endurance parameter value throughout optimization steps 99
Figure 3-44 Maximum thickness and maximum camber to chord ratio distribution on the baseline and optimized wing along the span 100
Figure 3-45 Pressure coefficient contours for the baseline and optimized wing .. 101

Figure 3-46 The wing sections of the baseline wing and the optimized wing...... 102
Figure 3-47 Skin friction distribution on the upper surfaces of the baseline and optimized wing.. 102
Figure 3-48 Evolution of endurance parameter value along optimization steps... 104
Figure 3-49 Maximum thickness and maximum camber to chord ratio distribution on the baseline and optimized wing along the span... 104
Figure 3-50 The lift distribution along the span 105
Figure 3-51 Skin friction coefficient distribution on the upper surfaces of the baseline and the optimized wing.. 105

LIST OF ABBREVIATIONS

ABBREVIATIONS

AD	Automatic Differentiation
ASO	Aerodynamic Shape Optimization
B-C	Bas-Cakmakcioglu
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CPU	Central Processing Unit
CRIAQ	Consortium for Research and Innovation in Aerospace in Quebec
CS	Clean Sky Programme
CS2	Clean Sky Programme 2
DAKOTA	Design Analysis Kit for Optimization and Terascale Applications
DDES	Delayed-Detached Eddy Simulation
DLR	German Aerospace Center
DNS	Direct Numerical Simulation
FFD	Free-Form Deformation
FV	Finite Volume
GMRES	Generalized Minimum Residual
HALE	High Altitude Long Endurance
IR	Infra-Red
JAXA	Japan Aerospace Exploration Agency
LamAir	Laminar Aircraft Project
LU	Lower-Upper
LS	Least-Square
MALE	Medium Altitude Long Endurance
MDO	Multi-Disciplinary Optimization
MPI	Message Passing Interface
MUMPS	MUltifrontal Massively Parallel sparse direct Solver

NACA	National Advisory Committee for Aeronautics
NASA	National Aeronautics and Space Administration
NLF	Natural Laminar Flow
PDE	Partial Differential Equations
PETSc	Portable, Extensible Toolkit for Scientific Computation
RANS	Reynolds-Averaged Navier-Stokes
S-A	Spalart-Allmaras
TAI	Turkish Aerospace Industries
TuLam	Toughen up Laminar Technology
UAV	Unmanned Aerial Vehicle
URANS	Unsteady Reynolds-Averaged Navier-Stokes

LIST OF SYMBOLS

SYMBOLS

$\boldsymbol{A}$	Area vector
c	Chord
c_l, C_L	Lift coefficient
c_d, C_D	Drag coefficient
c_p, C_p	Pressure coefficient
c_f, C_F	Skin friction coefficient
d_w	Distance to the closest wall boundary
$\boldsymbol{e}$	Unit vector
E	Total Energy per unit mass
$\boldsymbol{F_c}$	The convective flux tensor
$\boldsymbol{F_v}$	The viscous flux tensor
h	Finite difference step size
I	Objective function
k	Conductivity, Turbulent kinetic energy
$\boldsymbol{l}$	Length vector
m	meter
M	Mach number
$\boldsymbol{n}$	The face normal vector
N_{cell}	Number of cells
N_{dim}	Dimension size
N_{node}	Number of nodes
Pr_L, Pr_T	Prandtl number and turbulent Prandtl number, respectively
$\boldsymbol{R}$	Residual vector, upper triangular matrix
Re	Reynolds number
Re_v	Vorticity Reynolds number

Re_{θ}	Momentum thickness Reynolds number
Re_{θ_c}	Critical momentum thickness Reynolds number
$\boldsymbol{Q}$	Orthogonal matrix
S	Area
t	time
t	Thickness
T	The static temperature
$\boldsymbol{T}$	The transformation matrix
T_{u_∞}	Free-stream turbulence intensity
u	x component of the velocity vector
$\boldsymbol{U}$	The flow variable vector
$\widehat{\boldsymbol{U}}$	The rotated flow variable vector, a subset of flow variable vector
V	Velocity magnitude
$\boldsymbol{V}$	Velocity vector
v	y component of the velocity vector
w	z component of the velocity vector, weights of the LS approach
$\boldsymbol{X}$	Grid node coordinates
$\widehat{\boldsymbol{X}}$	Subset of grid node coordinates
y^{+}	Non-dimesnional wall distance
α	Angle of attack
$\boldsymbol{\alpha}$	Design variable vector
γ	The ratio of specific heat coefficient
γ_{BC}	Bas-Cakmakcioglu intermittency distribution function
δ	Kronecker delta
$\boldsymbol{\epsilon}$	The rate of the strain tensor
η	Half span ratio
$\boldsymbol{\Psi}$	Limiter function
θ	The volumetric dilatation rate
$\boldsymbol{\Theta}$	The work done by the fluid

κ	Von Karman constant
λ	The second viscosity coefficient
$\boldsymbol{\lambda}$	Adjoint vector
μ, μ_L	The dynamic viscosity
μ_T	Turbulent dynamic viscosity
$\tilde{\nu}$	Turbulent working variable
ν_t	turbulent kinematic viscosity
ν_{BC}	Bas-Cakmakcioglu turbulent kinematic viscosity like term
ρ	Density
σ	CFL number
$\boldsymbol{\tau}$	Stress tensor
χ_1, χ_2	The first and second constants of B-C transition model
$\boldsymbol{\Psi}$	Limiter function
$\mathit{\Omega}$	Volume, vorticity magnitude
°	degree

CHAPTER 1

INTRODUCTION

Aerodynamic shape optimization plays a significant role in the efficient design of aerial vehicles. It enables designing an aerial vehicle that has lower fuel consumption and CO_2 emission, longer range, higher endurance, higher payload, lower noise level, etc. As an example, decreasing the drag coefficient of a subsonic civil transport aircraft by one drag count through a better aerodynamic design results in about 90 kg more payload (Basha & Ghaly, 2007). Similarly, reducing the drag coefficient of a Lockheed C-5 airplane one drag count (which is less than 0.5% of the total drag coefficient of the Lockheed C-5), the payload capacity of the airplane at the cruise condition can be improved by around 450 kg (Basha & Ghaly, 2007).

In the early days of the aerospace industry, a design was mainly driven by trial-error and wind tunnel testing. However, wind tunnel testing is, in general, an expensive, elaborated, and time-consuming process. Moreover, it is not feasible to assess numerous design options by conducting wind tunnel tests. Nowadays, thanks to modern computers, the aerodynamic characteristics of an aerial vehicle can be estimated within a short time and in a much more economical way by computational fluid dynamics (CFD) analyses. More importantly, to conduct comprehensive design optimization studies are only feasible by employing numerical tools coupled with CFD, and wind tunnel tests are generally performed at the final design stage for validation and verification.

In the design optimization of competitive modern aerial vehicles, a design engineer should utilize a large number of design variables that provide sufficient design flexibility. The optimization algorithm instrumented in the study should, therefore, be able to deal with the large number of design variables required by the optimum aerodynamic shape design efficiently.

Gradient-based optimization algorithms are widely employed in aerodynamic shape optimization studies. Although gradient-free and surrogate-based optimization algorithms are both successful in achieving an optimum design, they both face the *curse of dimensionality* when dealing with a large number of design variables. That means the analyses required by the method grow exponentially with the increasing number of design variables. Even though the modern computers and CFD software packages enable conducting a high-fidelity analysis within hours, gradient-free algorithms may still be infeasible to utilize in an optimization problem with a large number of design variables. On the other hand, gradient-based algorithms do not scale with the dimension of the problem exponentially and are more feasible in a large dimensional aerodynamic shape optimization problem (Lyu, Xu, & Martins, 2014), (Yu, Lyu, Xu, & Martins, 2018).

Traditionally, the estimation of the sensitivity derivatives required in an optimization study is performed by the finite difference approximation. In the case of employing the finite difference approximation in a gradient-based algorithm, the computational cost of the algorithm is still much lower than the gradient-free and the surrogate-based optimization algorithms. Nevertheless, the dependence of the dimensionality abides, and an optimization problem still scales linearly with the number of design variables. There are analytic methods in the context of partial differential equations (PDE) constrained optimization (Martins & Hwang, 2013), namely the direct method and the adjoint method, as well. They are both suitable to use in aerodynamic shape optimization problems to get sensitivity derivatives. In these methods, the governing equations that are the constraints of the problem are differentiated, and the differentiated equations are made use of computing the sensitivity derivatives. Similar to the finite difference method, the computational cost of the direct methods scales linearly with the dimension of the problem. On the other hand, the computational cost of the adjoint method has been exhibited to be essentially independent of the number of design variables. That makes them a powerful tool in computing the sensitivity derivatives required in a gradient-based aerodynamic shape optimization problem.

1.1 Adjoint Methods and Automatic Differentiation

The adjoint methods that have been already instrumented in the context of the PDE constrained optimization over decades were firstly applied in fluid dynamics by Pironneau (1973) for Stokes flow. The method was further extended by Pironneau (1974) for laminar flows at higher Reynolds numbers. However, both studies were studied in an analytical sense. The method was firstly used by Glowinski & Pironneau (1975) in a CFD context to verify the previous works of Pironneau. Moreover, adjoint methods may be treated at either the continuous or the discrete level, known as *optimize-then-discretize* and *discretize-then-optimize* approach, respectively. In the context of adjoint methods, if the governing equations are firstly optimized and then discretized, hence the *optimize-then-discretize* approach, it is called a continuous adjoint method. Moreover, if the governing equations are firstly discretized and then optimized, hence the *discretize-then-optimize* approach, it is called a discrete adjoint method.

A continuous adjoint formulation to develop a design methodology for the compressible potential flow equations and the Euler equations using conformal mapping is stated by Jameson (1988). Subsequently, Jameson and his colleagues published their optimum design studies using a continuous adjoint approach for flows governed by compressible potential flow equations, Euler equations, and finally, Reynolds-Averaged Navier-Stokes equations (Jameson & Reuther, 1994), (Jameson, 1995), (Reuther, Jameson, Farmer, Martinelli, & Saunders, 1996), (Jameson, Pierce, & Martinelli, 1997), (Jameson, 1999), (Reuther, Jameson, Alonso, Rimlinger, & Sauders, 1997). On the other hand, an application of discrete adjoint methods for a compressible potential flow was conducted by Angrad (1983). Moreover, Elliot & Peraire (1996) used a discrete adjoint method to compute the sensitivity gradient for two-dimensional and three-dimensional Euler equations.

Nielsen & Anderson (2000) developed a discrete adjoint method for aerodynamic design optimization on unstructured grids using the Navier-Stokes equations.
In general, the discrete approach is considered more accurate since the gradients computed by a discrete adjoint method is consistent with the discrete objective function that is computed using the discrete form of the governing equations (Thomas, Hall, & Dowell, 2005). However, the gradients computed by the continuous adjoint approach are not necessarily consistent with the discrete objective function. Thus, it is more convenient to utilize discrete adjoint methods to validate the gradient with the finite-difference and to utilize them in a gradient-based optimization study. Furthermore, compared to discrete adjoint methods, continuous adjoint methods require noteworthy mathematical formulations, especially in the presence of viscous fluxes and turbulence models. Besides their disadvantages, it is also worthy of mentioning the advantages of continuous adjoint methods. They allow treating different discretization schemes in the adjoint form of the governing equations. They also provide an analytical adjoint form of the governing equations enabling us to study the characteristics of the equations. Finally, detailed comparisons of both methods can be found in the studies of Nadarajah & Jameson (2000), (2001), Carnarius, Thiele, Özkaya, & Gauger (2010), and Evgrafov, Gregersen, & Sorensen (2011).

Although, in general, developing a discrete formulation is simpler than developing a continuous adjoint formulation, it is also an elaborating, tedious, error-prone, and challenging task for a complex CFD solver. However, recently, a technique that highly automates the development of sensitivity analysis codes has been devised. In the technique, the existing code is differentiated by systematically applying the chain rule line-by-line to generate a new code that computes the sensitivity. The technique is known as automatic differentiation (AD) (Griewank & Walther, 2008), (Naumann, 2011) (also known as algorithmic differentiation or computational differentiation). The technique also employed in the development of a discrete adjoint solver. Automatic differentiation enables to develop a discrete adjoint solver either by differentiating the entire CFD code or by selectively using AD to compute the elements of the Jacobian matrices appearing in the adjoint method with much less effort.

There are two different approaches to develop a discrete adjoint method by using AD. In the first approach, AD is used to differentiate the entire flow solver (Albring, Sagebaum, & Gauger, 2016), (Djeddi & Ekici, 2019). In theory, the method requires minimum effort to develop an adjoint solver. However, due to the requirement of storing intermediate variables of the flow solver in the reverse mode of AD, employing AD to differentiate the entire flow solver is generally inefficient for large three-dimensional problems. Furthermore, the computational cost and memory requirement of the method are hindering. In order to alleviate the memory problem, advanced techniques such as checkpointing and local pre-accumulation by compromising from speed are employed although the problem size to use an adjoint method is still limited (Kenway, Mader, He, & Martins, 2019). On the other hand, in the latter approach, AD is selectively performed to construct the flux Jacobian matrices required by adjoint methods (Dilgen, Dilgen, Fuhrman, Sigmund, & Lazarov, 2018), (He, Mader, Martins, & Maki, 2020). In the approach, it is possible to compute each element of the Jacobian matrix using flow variables and the grid information. Therefore, it does not require to store the intermediate variables, and the computational cost and memory requirement of the method are not hindering.

Finally, over decades adjoint methods have gained increasing popularity thanks to its usefulness in computing sensitivities. It is still a hot topic on aerodynamic shape optimization studies (Nemili, Özkaya, Gauger, Kramer, & Thiele, 2017), (Lozano, 2018), (Cheylan, Fritz, Ricott, & Sagaut, 2019), (Albring, Beckett, & Gauger, Challenges in Sensitivity Computations for (D)DES and URANS, 2019), (Mengze, Qi, & Tamer, 2019), (Mader, Martins, & Maki, 2020), (Djeddi & Ekici, 2020). In addition to aerodynamic shape optimization studies, in the context of CFD, the adjoint methods have also been used in grid adaptation studies (Shei & Wang, 2016), (Balan, Park, & Anderson, 2019), in computing stability derivatives (Mader & Martins, 2014), etc.

1.2 Natural Laminar Flows

Another topic in aerodynamic shape optimization is the Natural Laminar Flow (NLF) technology that has been shown over decades to be able to reduce drag, thus fuel consumption and emissions. The technology primarily aims to stabilize instabilities arising from boundary layer disturbances and pressure gradient by proper design. The researches in this area burgeoned thanks to initial implementations of linear stability theory that was used as an instrument to study the stabilization of the boundary layer by favorable pressure gradients. These researches led to the design of NACA (National Advisory Committee for Aeronautics) 6-series airfoils that are natural laminar flow airfoils (Eastman, Kenneth, & Pinkerton, 1933). The superiority of the NACA 6-series airfoils held out around twenty years. By the advent of the computers, the theory could be utilized to assess more complex flow fields even occurring around simple airfoils. Moreover, although stabilization or destabilization of a viscous boundary layer could be estimated by conducting stability analyses on the Orr-Sommerfeld and Squire equations, the ability to estimate the transition onset came with very useful semi-empirical e^n method (Smith & Gamberoni, 1956), (van Ingen, 1956). Thus, airfoils with better performance were designed, such as airfoils of Wortman FX-series (Althaus, 1972), airfoils provided by McMasters (1974), or by Viken, Watson-Viken, Pfenninger, Morgan & Campell (1987). Subsequently, thanks to the relentless improvement in computational power, engineers could evaluate more complex flow fields for three-dimension as well as two-dimension, such as three-dimensional studies performed by Harris, Iyer, & Radwan (1987), Cummings & Garcia (1993), Iyer, Spall, & Dagenhart (1994), and Duck & Hall (1995). Around that time, some studies to couple transition methodologies with the existing turbulence models used in RANS equations were introduced. As an example, Warren & Hassan (Alternative to the e^n Method for Determining Onset of Transition, 1998), (Transition Closure Model for Predicting Transition Onset, 1998) developed a methodology to combine transition onset prediction methods with a turbulence model that is $k - \zeta$ model (Robinson & Hassan, 1997). Subsequently, many researchers suggested different laminar to turbulent transition models (Suzen & Huang, 2000), (Menter F. R., et al., 2004), (Walters & Leylek, 2004), (Walters & Cokljat, 2008), as well. Thus, the coupling of transition methodologies with the existing turbulence models used in RANS equations has allowed engineers to consider more complex geometries and have significantly contributed to the improvement of the NLF technology.

Although, in the past, the NLF technology was primarily used on experimental aircraft and sailplanes; nowadays, thanks to the incorporation of laminar to turbulent transition models into RANS solver, NLF is finding its way onto long-endurance unmanned air vehicles (UAV), commercial aircraft, and business jets (Fujino, Yoshizaki, & Kawamura, 2003), (Fujino, 2005), (Campell, Campell, & Streit, 2011), (Crouch, 2015). Moreover, there are still numerous recently completed researches and projects, as well as ongoing researches and projects regarding NLF technology. For instance, the DLR (German Aerospace Center) project LamAiR (Laminar Aircraft Project) dealt with designing a laminar wing for short-medium range transport aircraft (Seitz, Kruse, Wunderlich, & Bold, 2011). Its successor project TuLam (Toughen up Laminar Technology) aimed to increase further the Technology Readiness Level of laminar flow technologies at the institute (Seitz, Hübner, & Risse, 2019). Moreover, as another example, researchers at NASA (National Aerospace Agency) and the Japan Aerospace Exploration Agency (JAXA) collaboratively studied on implementation of the NLF design method to a supersonic transport aircraft configuration called as the NASA/JAXA wing-body (NJWB.) (Lynde & Campell, 2016). NLF was also one of the critical topics of the Clean Sky Programme (CS) and is of importance in the Clean Sky 2 Programme (CS2). Just as NLF technology, laminar to turbulent transition modeling is a still ongoing and attractive issue (Wauters, Degroote, & Vierendeels, 2019), (Kaynak, Bas, Cakmakcioglu, & Tuncer, 2019), (Diakakis, Papadakis, & Voutsinas, 2019).

1.3 Motivation

The usefulness of NLF technology and adjoint methods in an aerodynamic shape optimization enables high motivation to combine the methods for the design optimization of NLF air vehicles. Hence, in this study, it is aimed to combine these two essential and highly-interesting topics in aerodynamic shape optimization studies by the implementation of adjoint methods for a RANS flow solver coupled with a turbulence model and a laminar to turbulent transition model.

In literature, there are some studies that combine adjoint methods with turbulence models employing a transition model to predict transition onset. For instance, Khayatzadeh & Nadarajah (2011), (2012) developed a discrete adjoint solver for the $k - w\ SST$ turbulence model coupled with $\gamma - Re_\theta$ transition model (Menter F. R., et al., 2004). In their studies, they also carried out optimization studies to minimize the drag coefficients of S809, NACA 0012, and NLF(1)-0416 airfoils by minimizing turbulent kinetic energy k in the domain at a constant lift. Moreover, they optimized the c_l/c_d ratio of NACA 0012 and NLF(1)-0416 aifoils. Rashad & Zingg (2016) developed a two-dimensional discrete adjoint solver for the Spalart-Allmaras turbulence model coupled with an e^n transition model. They conducted single-point and multi-point optimization studies on NLF airfoils by computing necessary gradients with their discrete adjoint solver. More recently, Yang & Mavriplis (2019) published an implementation of the adjoint solver for fully coupled turbulence-transition equations. They optimized a NACA 0012 airfoil in viscous flow with free-transition for decreasing drag and increasing c_l/c_d ratio. In the study, the turbulence model was the Spalart-Allmaras model and the transition model to predict transition onset was AFT2 model (Coder & Maughmer, 2014).

Although there are studies that combine NLF and adjoint methods, they are limited to two-dimensional problems. Thus, another motivation for the present implementation is to evaluate the applicability of adjoint methods for RANS equations coupled with a turbulence model and a laminar to turbulent transition model in a three-dimensional problem.

In this context, it is aimed to develop an in-house adjoint solver. The adjoint solver is developed in a discrete manner. The reasons for employing the discrete approach are the consistency of the discrete form of the governing equations with the objective function and the convenience provided by AD in the development phase, in coding, and in adapting to changes in the flow solver.

In order to develop a discrete adjoint solver, an aerodynamic flow solver to be differentiated is required. Moreover, since AD is taken advantage of the development of the discrete adjoint solver, it is better to have enough knowledge about the low-level implementation of the solver to manage the automatic differentiation process smoothly. Therefore, it is decided to develop an aerodynamic flow solver, which would be mainly based on a predecessor in-house solver detailed in the reference (Gökhan, 2014). Moreover, developing an in-house flow solver allows having full knowledge of the algorithm, data structures, subroutines, etc. of the solver. In this way, the development of a discrete adjoint solver that requires to differentiate discretized flow equations is performed very easily. Furthermore, that enables to integrate easily different schemes, different turbulence models, transition models etc. in the flow and adjoint solver whose low-level implementations are mastered.

In the solver, in order to estimate the transition onset, Bas-Cakmakcioglu (B-C) transition model (Cakmakcioglu, Bas, & Kaynak, 2017) is incorporated into the solver. The incorporated laminar to turbulent transition model is a relatively new model developed. The model is an algebraic transition model. Since it is an algebraic model, it does not introduce a new equation to solve. Hence, the size of the flux Jacobian matrix appearing in the adjoint method does not grow with the incorporation of the transition model, and that makes the model a favorable choice to implement in a RANS adjoint solver that considers the laminar to turbulent transition. Moreover, another motivation to implement the B-C transition model is that it will be the first study that considers the implementation of the B-C transition model in an adjoint approach.

1.4 Thesis Objective

The present study aims at developing an adjoint-based aerodynamic shape optimization tool for three-dimensional natural laminar flows and implementing it for natural laminar flow airfoils and wings. An in-house, finite volume-based flow solver is adopted for further development and the addition of a discrete adjoint solver by means of automatic differentiation. The main objectives of the study may be itemized as follows:

- To modify the in-house flow solver by rewriting it in *Fortran 95* with more efficient schemes and data structures;
- To couple the RANS solver with the Spalart-Allmaras and the correlation-based Bas-Cakmakcioglu transition model;
- To validate the flow solver for both turbulent and natural laminar flows;
- To develop the discrete adjoint flow equations for natural laminar flows by means of an automatic differentiation tool and validate the sensitivity derivatives computed by the adjoint solver against the finite difference method;
- To develop an aerodynamic shape optimization framework that enables to employ the sensitivity derivatives computed by the discrete adjoint solver and an open-source optimization tool;
- To perform aerodynamic shape optimization studies for both turbulent and natural laminar flows over airfoils and wings.

1.5 Thesis Outline

In the following chapter, the methodology of the flow solver, and the adjoint solver are introduced. The structure of the optimization framework is explained. In the subchapter 2.1, the governing equations, the discretization, and the numerical implementation of the RANS flow solver are detailed. In the subchapter 2.2, the adjoint equations, and automatic differentiation are described. Subsequently, the development of the adjoint method is mentioned. The solution of the adjoint system is explained. In the subchapter 2.3, the surface parametrization and the grid deformation method are described. The computation of the grid sensitivities and the algorithm employed in the optimization framework are presented.

In chapter 3, the validation of the flow solver, the validation of the adjoint solver, and the optimization studies are delivered. In the subchapter 3.1, the validation of the RANS solver is performed through turbulent and natural laminar flow airfoil and wings. In subchapter 3.2, the validation studies of the adjoint solver are presented and discussed. In subchapter 3.3, optimization studies that are conducted to demonstrate the functionality of the adjoint method in ASO are described, and the results of the optimization studies are presented.

Finally, in chapter 4, the summary of the thesis, concluding remarks and future studies are discussed.

CHAPTER 2

METHOD

In this chapter, the Reynolds-Averaged Navier-Stokes (RANS) flow solver, the adjoint solver, and the optimization framework developed in this study are introduced in detail, and the functionality of the adjoint solver in an aerodynamic shape optimization study is demonstrated.

2.1 Reynolds-Averaged Navier-Stokes Flow Solver

In this section, the newly developed flow solver that inherits many features from its predecessor solver is introduced and detailed.

The flow solver is a finite volume (FV) based solver on unstructured grids. The integral forms of the governing equations are discretized in a cell-centered FV method with a second-order accurate flux evaluation.

The temporal discretization scheme is a third-order explicit Runga-Kutta scheme. Furthermore, when a steady-state solution is concerned, in order to accelerate the convergence rate, a local time-stepping is employed.

In the following subsections, the governing equations of the fluid flow, spatial and temporal discretization, turbulence modeling, transition modeling, and some implementation issues are detailed.

2.1.1 Governing Equations

The mathematical definition of a compressible fluid flow includes the conservation equations for mass, momentum, and energy, which are known as the Navier-Stokes equations, and the equation of state together with the proper boundary conditions.

The integral form of the conservation equations can be stated in vector notation as in Eq 2-1.

$$\frac{\partial}{\partial t}\int_{\Omega}\boldsymbol{U}d\Omega+\oint_{\partial\Omega}(\boldsymbol{F_c}-\boldsymbol{F_v}).\boldsymbol{n}\,ds=0 \qquad \text{Eq 2-1}$$

In Eq 2-1, $\boldsymbol{U}$, $\boldsymbol{F_c}$ and $\boldsymbol{F_v}$ stand for the conservative flow variables, the tensor of the convective fluxes, and the tensor of the viscous fluxes, respectively. Furthermore, the vector of the conservative flow variables and the convective fluxes are defined in Eq 2-2.

$$\boldsymbol{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix}, \qquad \boldsymbol{F_c}=\boldsymbol{F}(\boldsymbol{U})+\boldsymbol{G}(\boldsymbol{U})+\boldsymbol{H}(\boldsymbol{U}) \qquad \text{Eq 2-2}$$

In the equation, ρ denotes the density, u, v, w are the components of the velocity vector $\boldsymbol{V}$, and E is the total energy per unit mass. The components of the convective flux tensor $\boldsymbol{F, G, H}$ are given in Eq 2-3.

$$\boldsymbol{F}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho(E+p)u\end{bmatrix}, \boldsymbol{G}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ \rho(E+p)v\end{bmatrix}, \boldsymbol{H}=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ \rho(E+p)w\end{bmatrix} \qquad \text{Eq 2-3}$$

The assumption of calorically perfect gas makes sense for practical aerodynamics problems. Hence, pressure p in the Eq 2-3 can be estimated by the equation of the state (Eq 2-4).

$$p=(\gamma-1)\rho\left[E-\frac{u^2+v^2+w^2}{2}\right] \qquad \text{Eq 2-4}$$

In the equation of the state, γ denotes the ratio of specific heat coefficients.

The components of the viscous fluxes are given in the following equation.

$$\boldsymbol{F_v} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ \tau_{zx} \\ \Theta_x \end{bmatrix} \boldsymbol{i} + \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{zy} \\ \Theta_y \end{bmatrix} \boldsymbol{j} + \begin{bmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ \Theta_z \end{bmatrix} \boldsymbol{k} \qquad \text{Eq 2-5}$$

In the equation, $\tau_{xx}, \tau_{yy}, \tau_{zz}$ stand for the normal stresses and $\tau_{xy}, \tau_{yz}, \tau_{xz}$ stand for the shear stresses. They are the components of the symmetric stress tensor stated in Eq 2-6.

$$\bar{\bar{\boldsymbol{\tau}}} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} \qquad \text{Eq 2-6}$$

For a Newtonian fluid, the stress tensor is linearly correlated with the strain rate tensor. Moreover, the stress tensor of a Newtonian fluid is defined as follows,

$$\tau_{ij} = \lambda\, \theta \delta_{ij} + 2\mu \epsilon_{ij} \qquad \text{Eq 2-7}$$

where μ, λ are the dynamic viscosity coefficient and the second viscosity coefficient, respectively. The value of λ is estimated by the help of Stoke’s hypothesis as given in Eq 2-8.

$$\lambda = -\frac{2}{3}\mu \qquad \text{Eq 2-8}$$

Moreover, ϵ_{ij} is the rate of the strain tensor,

$$\epsilon_{ij} = \frac{1}{2}(V_{i,j} + V_{j,i}) \qquad \text{Eq 2-9}$$

and θ is the volumetric dilatation rate, which is defined as stated in Eq 2-10.

$$\theta = V_{i,i} \qquad \text{Eq 2-10}$$

$\boldsymbol{\Theta}$ denotes the work done by the fluid and the heat conduction in the fluid. The relation regarding $\boldsymbol{\Theta}$ is given in Eq 2-11, where k, and T represent the thermal conductivity and the static temperature, respectively.

$$\Theta_i = V_i \tau_{ij} + k \frac{\partial T}{\partial x_i} \qquad \text{Eq 2-11}$$

Finally, the equations above are non-dimensionalized with freestream reference values, $\rho_\infty, V_\infty, T_\infty$ and a length scale L_∞ that is, in general, mean aerodynamic chord length. That makes variables the same order of magnitude, and therefore, during numerical computations, loss of accuracy due to numerical round-off error is minimized.

2.1.2 Spatial Discretization

The spatial discretization of the flow solver is based on FV method. Thus, it is necessary to decompose the domain into finite volumes.

In a cell-centered scheme, the finite volumes are the grid cells of the computational domain. In order to discretize the domain, a variety of cell types are used in CFD. The solver can operate on triangular, quadrilateral, tetrahedral, pyramid, prismatic, and hexahedral cells.

2.1.3 Geometrical Properties of a Control Volume

In a two-dimensional problem, the area vector of a cell is calculated by making use of vector cross products. The Euclidean norm of the cross product equals the area of a parallelogram with vectors formed by the side of the parallelogram. Since the faces are planar, the area evaluated by the method is exact.

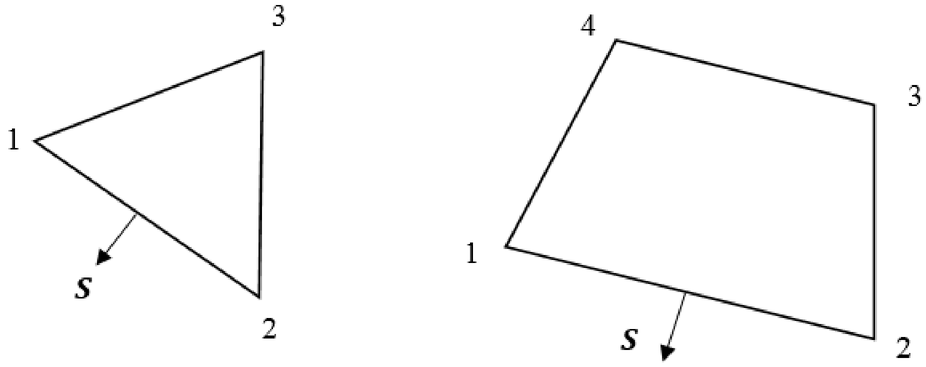


Figure 2-1 A triangular and a quadrilateral element

Hence, area vector $\boldsymbol{A}_{\boldsymbol{triangular}}$ of a triangle cell and area vector $\boldsymbol{A}_{\boldsymbol{triangular}}$ of a quadrilateral cell illustrated in Figure 2-1 are calculated by Eq 2-12 and Eq 2-13, respectively.

$$\boldsymbol{A}_{\boldsymbol{triangular}} = \frac{\boldsymbol{l}_{\mathbf{12}} \times \boldsymbol{l}_{\mathbf{13}}}{2} \qquad \text{Eq 2-12}$$

$$\boldsymbol{A}_{\boldsymbol{triangular}} = \boldsymbol{l}_{\mathbf{12}} \times \boldsymbol{l}_{\mathbf{14}} \qquad \text{Eq 2-13}$$

In the above equation, the vector $\boldsymbol{l}$, e.g. $\boldsymbol{l}_{\mathbf{12}}$ stands for the vector connecting the vertices 1-2. Moreover, the edge vector of a cell always points outward as given in Figure 2-1. The edge vector $\boldsymbol{S}$ of a cell, e.g., at the side 1-2, is computed by Eq 2-14.

$$\boldsymbol{S} = (y_2 - y_1)\boldsymbol{i} + (x_1 - x_2)\boldsymbol{j} \qquad \text{Eq 2-14}$$

In a three-dimensional problem, the volume Ω of a cell is calculated based on the divergence theorem stated in Eq 2-15,

$$\Omega = \frac{1}{3}\sum_{i=1}^{N_F}(\boldsymbol{r}_{mid} \cdot \boldsymbol{A})_i \qquad \text{Eq 2-15}$$

where N_F refers to the number of the faces constituting the control volume, and $\boldsymbol{r}_{mid}$ is the vector from the center of the face i to the cell center. The outward-pointing face vector $\boldsymbol{A}$ of a cell is estimated as given in Eq 2-12 and Eq 2-13. The method gives the exact volume for a tetrahedral element. Furthermore, it also gives the exact volume for a prism and a hexahedral cell, unless the quadrilateral faces are bi-planar.

2.1.4 Discretization of Convective Fluxes

The convective fluxes are discretized, such that they are the sum of the normal fluxes across the boundary segments of the control volume,

$$\oint \boldsymbol{F_c} \cdot \boldsymbol{n}\, ds = \sum_{j=1}^{N_F} (\boldsymbol{F}(\boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U}) + \boldsymbol{H}(\boldsymbol{U}))_{\boldsymbol{j}} \cdot \boldsymbol{n}_j \Delta S_j\,. \qquad \text{Eq 2-16}$$

Similar to the predecessor solver, the multi-dimensional Euler equations are not solved by a dimensional-splitting scheme that carries out one-dimensional methods in each dimension, but by the *unsplit finite volume method* (Toro, 2009). In this scheme, the rotational invariance property of the Euler equations is exploited. Accordingly, by using a transformation matrix, the multi-dimensional Euler equations are rearranged, such that they are defined in an augmented one-dimensional system that is normal to the face boundary. The rearranged convective flux at a face may be written as follows,

$$\left(\boldsymbol{F}(\boldsymbol{U}) + \boldsymbol{G}(\boldsymbol{U}) + \boldsymbol{H}(\boldsymbol{U})\right)_{\boldsymbol{j}} \cdot \boldsymbol{n}_j \Delta S_j \;= \boldsymbol{T}^{-\boldsymbol{1}} \widehat{\boldsymbol{F}}(\boldsymbol{T}\boldsymbol{U}) \cdot \boldsymbol{n}_j \Delta S_j \qquad \text{Eq 2-17}$$

where $\widehat{\boldsymbol{F}}$ is the convective flux corresponding to the augmented one-dimensional system, and $\boldsymbol{T}$ is the three-dimensional rotation matrix defined in Eq 2-18.

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & cos\theta^{xy}cos\theta^{xz} & cos\theta^{xy}sin\theta^{xz} & sin\theta^{xz} & 0 \\ 0 & -sin\theta^{xz} & cos\theta^{xz} & 0 & 0 \\ 0 & -sin\theta^{xy}cos\theta^{xz} & -sin\theta^{xy}sin\theta^{xz} & cos\theta^{xy} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq 2-18}$$

In Eq 2-18, the angle θ^{xy} stands for the angle between the face normal vector $\boldsymbol{n}_j$ and x-y plane, akin to that, the angle θ^{xz} stand for the angle between the face normal vector $\boldsymbol{n}_j$ and x-z plane. When a two-dimensional case is considered, the rotation matrix is defined as in Eq 2-19.

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos\theta^y & sin\theta^y & 0 \\ 0 & -sin\theta^y & cos\theta^y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{Eq 2-19}$$

Moreover, in the augmented one-dimensional system, the vector of rotated flow variables $\widehat{\boldsymbol{U}}$ defined in Eq 2-20 is considered.

$$\widehat{\boldsymbol{U}} = \boldsymbol{TU} \qquad \text{Eq 2-20}$$

In the solver, the convective flux vector corresponding to the augmented one-dimensional system $\widehat{\boldsymbol{F}}$ can be computed by either the Roe's approximate Riemann solver (Roe, 1981) or another Riemann solver (Nishikawa & Kitamura, 2008) that is a combination of the Roe's solver and the Rusanov/HLL (hereafter referred to as Rotated-Roe-HLL). Numerical experiments performed show that the Rotated-Roe-HLL scheme is more robust and to have a higher convergence rate comparing Roe's scheme. In the original paper (Nishikawa & Kitamura, 2008) of the Rotated-Roe-HLL scheme, the robustness and the high-convergence rate of the scheme are also emphasized and depicted by examples. Due to its robustness and high convergence rate, the Rotated-Roe-HLL scheme is preferred as the default convective flux scheme of the solver. In addition, in order to get a physically relevant non-unique solution, Harten's entropy fix (Harten, Lax, & van Leer, 1983) is imposed.

To sum up, the computation of the convective fluxes through a face is conducted by following the algorithm given in the study (Toro, 2009). The algorithm is

I. Calculate the rotated left and right flow variables $\widehat{\boldsymbol{U}}_L$, $\widehat{\boldsymbol{U}}_R$ by the transformation given in Eq 2-20,

II. Compute the convective fluxes corresponding to the augmented one-dimensional system $\widehat{\boldsymbol{F}}$ by either the Roe's scheme or the Rotated-Roe-HLL scheme,

III. Rotate $\widehat{\boldsymbol{F}}$ back to the inertial frame by multiplying $\boldsymbol{T^{-1}}$.

Finally, the Roe scheme and the Rotated-Roe-HLL scheme have been implemented by modifying corresponding subroutines given on the website (Masatsuka, 2020) in terms of computational efficiency.

2.1.5 Variable Reconstruction

In the upwind schemes, face fluxes are computed using the left and right values. In the case of the cell-centered scheme, a first-order scheme reads the left and right state as the scalar flow field value at the cell center. Nevertheless, first-order schemes are too diffusive, which results in weaker, smeared shocks and excessive growth of the shear layer, hence inaccurate solutions. Therefore, a second-order scheme that assumes a linear variation in a cell is incorporated into the solver.

In the solver, a second-order accuracy is achieved by the piecewise linear reconstruction given in the equations below.

$$U_L = U_i + \boldsymbol{\Psi_i}\,(\boldsymbol{\nabla} U_i \cdot \boldsymbol{r}_L) \qquad \text{Eq 2-21}$$

$$U_R = U_j + \boldsymbol{\Psi_j}\,\left(\boldsymbol{\nabla} U_j \cdot \boldsymbol{r}_R\right) \qquad \text{Eq 2-22}$$

where U_L and U_R denote the left and right state, respectively. $\boldsymbol{\Psi}$ denotes for a limiter function detailed in the subchapter 2.1.7. $\boldsymbol{\nabla} U$ is the gradient of the flow variables that are calculated by the methods given in the subchapter 2.1.6. $\boldsymbol{r}_L$ and $\boldsymbol{r}_R$ represent the vector from the cell-center to the mid point of the face, and the neighbor cell center to the midpoint of the face, respectively. Finally, in the above equations and the following equations, the subscripts i, and j represent the cell of interest, and the neighbouring cells of it, respectively.

2.1.6 Evaluation of Gradients of Flow Variables

In a finite volume solver, the gradients of the scalar flow variables are necessary for the implementation of high-resolution advection schemes, and the computation of viscous fluxes dependent on velocity gradients. There are two basic strategies to estimate gradients on an unstructured grid. The first one is based on Green-Gauss approaches, and the second one is based on least-square approaches. A variety of different implementations of Green-Gauss approaches (Barth & Jespersen, 1989), (Deka, Brahmachary, Thirumalaisamy, & Dalal, 2018), (Nishikawa, 2019) and Least-Squares approaches exist (Haider, Croisille, & Courbet, 2009), (Sozer, Brehm, & Kiris, 2014). Moreover, there are also hybrid methods to compute gradients, e.g., the study of Shima, Kitamura, & Haga (2013). The accuracy and robustness of the methods are assessed in many studies (Mavriplis, 2003), (Diskin & Thomas, 2008), (Diskin & Thomas, 2011), (Syrakos, Varchanis, Dimakopoulos, Goulas, & Tsamopoulos, 2017).

In the solver, both approaches are implemented to estimate gradient values. The basic version of Green-Gauss approach is implemented. The approach is mathematically defined as given below in Eq 2-23.

$$\boldsymbol{\nabla} U = \frac{1}{\Omega} \oint_{\partial\Omega} U\, \boldsymbol{n}\, ds \qquad \text{Eq 2-23}$$

Moreover, the discretized form of Eq 2-23 was formulated.

$$\boldsymbol{\nabla} U_{\boldsymbol{i}} = \frac{1}{\Omega} \sum_{j=1}^{N_F} \frac{1}{2} (U_i + U_j) \boldsymbol{n_j} \Delta S_j \qquad \text{Eq 2-24}$$

The scalar flow field value U at the face is computed by taking the arithmetic average of the flow field value of the cell itself U_i and the neighbor cell U_j.

As stated before, a least-square approach to compute the gradients is also implemented to the solver. The approach is the weighted least square approach. The method is based on a linear fit over neighboring cells. Since the number of faces of an element is always greater than the dimension, the linear fit requires solving an overdetermined linear system $\boldsymbol{Ax} = \boldsymbol{b}$, where the matrix $\boldsymbol{A}$ and the vectors $\boldsymbol{x}, \boldsymbol{b}$ are defined in the following equations.

$$\boldsymbol{A} = \begin{bmatrix} w_1(x_1 - x_i) & w_1(y_1 - y_i) & w_1(z_1 - z_i) \\ w_2(x_1 - x_i) & w_2(y_2 - y_i) & w_2(z_2 - z_i) \\ \vdots & \vdots & \vdots \\ w_{N_F}(x_{N_F} - x_i) & w_{N_F}(y_{N_F} - y_i) & w_{N_F}(z_{N_F} - z_i) \end{bmatrix} \qquad \text{Eq 2-25}$$

$$\boldsymbol{x} = \boldsymbol{\nabla} U_{\boldsymbol{i}} \qquad \text{Eq 2-26}$$

$$\boldsymbol{b} = \begin{bmatrix} w_1(U_1 - U_i) \\ w_2(U_2 - U_i) \\ \vdots \\ w_{N_F}(U_{N_F} - U_i) \end{bmatrix} \qquad \text{Eq 2-27}$$

In the implementation, the inverse-distance weighting is applied. The weights w may be written as is given in Eq 2-28,

$$w_j = \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}} \qquad \text{Eq 2-28}$$

where x, y, z denote the coordinates of the cell centers. The subscript i refer to the cell itself and j refer to the corresponding neighbor cell. The solution of the linear system requires the inversion of the matrix $\boldsymbol{A}$. In order to avoid the possible problems with ill-conditioning, the pseudo-inverse of the matrix $\boldsymbol{A}$ is computed by taking advantage of QR factorization. So, the matrix is decomposed into an orthogonal matrix $\boldsymbol{Q}$ and an upper triangular matrix $\boldsymbol{R}$. Thus, the pseudo-inverse of the $\boldsymbol{A}$ matrix reads

$$\boldsymbol{A}^{-1} = \boldsymbol{R}^{-1}\boldsymbol{R}^{T}\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{b}. \qquad \text{Eq 2-29}$$

Moreover, the entries of the upper triangular matrix $\boldsymbol{R}$ are computed as given in the study of Haselbacher & Blazek (2000). In the paper, QR factorization is performed utilizing the Gram-Shmidt orthogonalization. The entries r of the upper triangular matrix $\boldsymbol{R}$ read Eq 2-30.

$$\begin{gathered} r_{11} = \sqrt{\sum_{j=1}^{N_F} {w_j}^2 (x_j - x_i)^2}, r_{12} = \frac{1}{r_{11}} \sum_{j=1}^{N_F} w_j (x_j - x_i) w_j (y_j - y_i) \\ r_{22} = \sqrt{\sum_{j=1}^{N_F} \left({w_j}^2 (y_j - y_i)^2\right) - {r_{12}}^2}, r_{23} = \frac{1}{r_{22}} \left(r_{23\_1} - \frac{r_{12}}{r_{11}} r_{23\_2}\right) \\ r_{23\_1} = \sum_{j=1}^{N_F} w_j (y_j - y_i) w_j (z_j - z_i) \end{gathered} \qquad \text{Eq 2-30}$$

$$r_{23\_2} = \sum_{j=1}^{N_F} w_j\left(y_j - y_i\right) w_j\left(z_j - z_i\right)$$

$$r_{33} = \sqrt{\sum_{j=1}^{N_F}\left(w_j{}^2(z_j - z_i)^2\right) - \left(r_{13}{}^2 + r_{23}{}^2\right)}$$

In a two-dimensional problem, the components of the matrix $\boldsymbol{S} = \boldsymbol{R}^{-1}\boldsymbol{R}^T$ reads,

$$\begin{gathered} s_{11} = (r_{12}r_{12} + r_{22}r_{22})/\det(\boldsymbol{R}),\ s_{12} = -r_{11}r_{12}/\det(\boldsymbol{R}) \\ s_{21} = s_{12}, s_{22} = r_{12}r_{12}/\ \det\,(\boldsymbol{R}). \end{gathered} \qquad \text{Eq 2-31}$$

In a three-dimensional problem, the components of the matrix $\boldsymbol{S}$ reads,

$$\begin{gathered} s_{11} = ((r_{22}r_{33})^2 + (r_{12}r_{33})^2 + (r_{12}r_{23} - r_{13}r_{22})^2)/\det(\boldsymbol{R}) \\ s_{12} = -\,((r_{12}r_{23} - r_{13}r_{22})r_{11}r_{23} + r_{12}r_{23}r_{13}r_{11})/\det(\boldsymbol{R}) \\ s_{13} = (r_{12}r_{23} - r_{13}r_{22})r_{11}r_{23}/\det(\boldsymbol{R}) \\ s_{22} = ((r_{11}r_{33})^2 + (r_{11}r_{23})^2)/\det(\boldsymbol{R}) \\ s_{23} = -r_{11}r_{23}r_{11}r_{22}/\det(\boldsymbol{R}) \\ s_{12} = s_{21}, s_{31} = s_{13}, s_{32} = s_{23}. \end{gathered} \qquad \text{Eq 2-32}$$

At the cells neighboring a boundary, length between cell centers is assumed as twice the length between the cell center and the mid of the corresponding boundary face. The flow variables of the neighbor cell are computed from the corresponding ghost cell.

Note that in the implementation, the robustness of the least-square is low. Therefore, in the analyses, gradient computations are performed by using the Green-Gauss approach for a second-order accuracy.

2.1.7 Limiter Function for Variable Reconstruction

Second and higher-order spatial discretization schemes require a flux limiter function to suppress the oscillations appearing in the vicinity of high gradients (e.g., shocks). The limiter functions achieve this goal by reducing or removing the gradients employed in the reconstruction, so they guarantee the monotonicity.

In order to prevent such oscillations and spurious solutions, a limiter function is implemented to the solver. The implemented limiter function is the function suggested by Venkatakrishnan (1993), (1995) because the function is considered to have a better convergence rate and to be more robust comparing the other limiter functions thanks to its smoothness.

The Venkatakrishnan limiter function is stated in Eq 2-33,

$$\boldsymbol{\Psi}_{\boldsymbol{i}} = min_j \begin{cases} \dfrac{1}{\Delta_2}\left[\dfrac{\left(\Delta_{1,max}^2 + \varepsilon^2\right)\Delta_2 + 2\Delta_2^2\Delta_{1,max}}{\Delta_{1,max}^2 + 2\Delta_2^2 + \ \Delta_{1,max}\Delta_2 + \varepsilon^2}\right] & if\ \Delta_2 > 0 \\ \dfrac{1}{\Delta_2}\left[\dfrac{\left(\Delta_{1,min}^2 + \varepsilon^2\right)\Delta_2 + 2\Delta_2^2\Delta_{1,min}}{\Delta_{1,min}^2 + 2\Delta_2^2 + \ \Delta_{1,min}\Delta_2 + \varepsilon^2}\right] & if\ \Delta_2 < 0 \\ 1 & if\ \Delta_2 = 0, \end{cases} \qquad \text{Eq 2-33}$$

where

$$\Delta_{1,max} = U_{max} - U_i \qquad \text{Eq 2-34}$$

$$\Delta_{1,min} = U_{min} - U_i \qquad \text{Eq 2-35}$$

$$\Delta_2 = \boldsymbol{\nabla} U_i \cdot \boldsymbol{r}_L \qquad \text{Eq 2-36}$$

$$\varepsilon^2 = \left(K\widehat{\Delta x}\right).^3 \qquad \text{Eq 2-37}$$

In the equations above, U denotes the scalar flow field variables. U_{max} and U_{min} refer to the maximum and minimum scalar flow field variables of the neighboring cells and the cell itself, respectively. $\widehat{\Delta x}$ denotes an average grid size. It is also possible to employ a local length scale $\widehat{\Delta x}_i$ rather than employing a global length scale $\widehat{\Delta x}$. The parameter of K represents a threshold. Oscillations below the threshold value are allowed to occur and not treated by the limiter. To set the value to 0 means fully limiting. However, that may deteriorate the convergence of the solution. On the other hand, an increase in K reduces the effectiveness of the limiter, but it ameliorates the convergence of the solution.

In the solver, rather than employing a local length scale, a global value that is 0.01 non-dimensional length for $\widehat{\Delta x}$ is employed. Moreover, in the solver, the default value of K is 5.

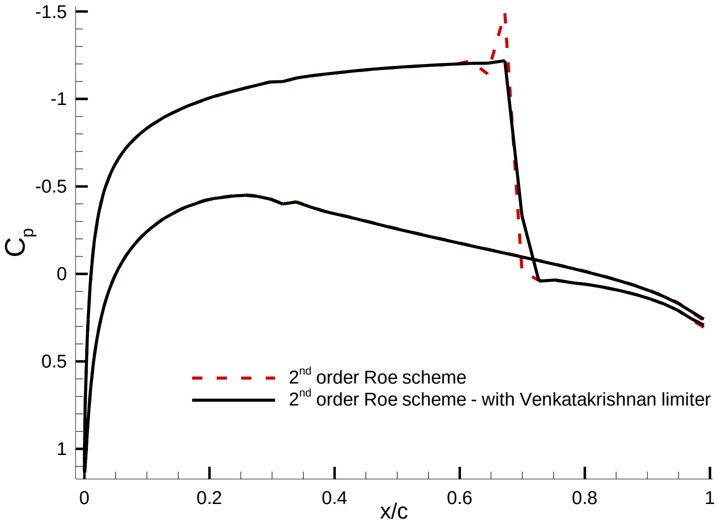


Figure 2-2 Effect of limiter function

Finally, the effect of the limiter function is represented for an inviscid flow around RAE 2822 airfoil in Figure 2-2. The figure shows that the limiter remedies the solution by removing the spurious kink appearing in the vicinity of the shock.

2.1.8 Discretization of Viscous Fluxes

In order to compute viscous fluxes defined by Eq 2-5 in a discrete manner, velocities, velocity gradients, and dynamic viscosities at cell faces are required. Unlike convective fluxes that have a hyperbolic nature, viscous fluxes have an elliptic nature. Thus, the cell face values are simply obtained by averaging the neighboring cell, as expressed in the following equations.

$$U_{\boldsymbol{ij}} = \frac{1}{2}(U_i + U_j) \tag{Eq 2-38}$$

$$\overline{\boldsymbol{\nabla} U}_{ij} = \frac{1}{2}(\boldsymbol{\nabla} U_i + \boldsymbol{\nabla} U_j) \tag{Eq 2-39}$$

$$\mu_{ij} = \frac{1}{2}(\mu_i + \mu_j) \tag{Eq 2-40}$$

Besides, the gradient values at faces are corrected by the directional derivative along the line connecting the cell centers. The correction prevents decoupling on quadrilateral and hexahedral grids, where the decoupling may increase the truncation error and reduce the convergence rate significantly. The modified gradient at the cell faces may be written as given in Eq 2-41.

$$\boldsymbol{\nabla} U_{\boldsymbol{ij}} = \overline{\boldsymbol{\nabla} U}_{ij} - \left(\overline{\boldsymbol{\nabla} U}_{ij} \cdot \boldsymbol{t}_{ij} - \left(\frac{\partial U}{\partial l} \right)_{ij} \right) \boldsymbol{t}_{\boldsymbol{ij}} \qquad \text{Eq 2-41}$$

In Eq 2-41, $\left(\frac{\partial U}{\partial l}\right)_{ij}$ term is defined as the change in the flow variable U, form the center of the cell *i* to the center of the cell *j*. It is defined in Eq 2-42,

$$\left(\frac{\partial U}{\partial l} \right)_{ij} \approx \frac{U_j - U_i}{l_{ij}} \qquad \text{Eq 2-42}$$

where l_{ij} is the distance from the center of the cell *i* to the center of the cell *j*. $\boldsymbol{t}_{ij}$ appearing in Eq 2-42 is a unit vector, which is defined as in Eq 2-43.

$$\boldsymbol{t}_{ij} = \frac{\boldsymbol{r}_{ij}}{l_{ij}} \qquad \text{Eq 2-43}$$

In the equation above, $\boldsymbol{r}_{ij}$ is the vector connecting the center of the cell *i* to the center of the cell *j*.

2.1.9 Temporal Discretization

The flow solver applies different discretization schemes in space and time. The spatial discretization is detailed in the previous subchapters. In this subchapter, the temporal discretization is explained.

The conservation equation stated in Eq 2-1 is discretized formulation as followed.

$$\Omega_i \frac{\partial \boldsymbol{U_i}}{\partial t_{\boldsymbol{i}}} + \underbrace{\sum_{j=1}^{n_{face}} (\boldsymbol{F_c}_j - \boldsymbol{F_v}_j) \cdot \boldsymbol{n_j} \Delta S_{\boldsymbol{j}}}_{Residual, \boldsymbol{R_i}} = 0 \qquad \text{Eq 2-44}$$

In Eq 2-44, the part defined as $Residual$ is formed by the summation of convective and viscous fluxes. The time integration is then performed.

The time integration is employed by the third-order Runga-Kutta explicit scheme. Hence, the flow variables $\boldsymbol{U_i}$ of a cell are updated as expressed in Eq 2-45.

$$\begin{aligned} \boldsymbol{U}_i{}^{(1)} &= \boldsymbol{U}_i{}^{\boldsymbol{n}} - \propto_1 \frac{\Delta t_i}{\Omega_i} \boldsymbol{R}(\boldsymbol{U}_i{}^{n}) \\ \boldsymbol{U}_i{}^{(2)} &= \boldsymbol{U}_i{}^{(1)} - \propto_2 \frac{\Delta t_i}{\Omega_i} \boldsymbol{R}\big(\boldsymbol{U}_i{}^{(1)}\big) \\ \boldsymbol{U}_i{}^{n+1} &= \boldsymbol{U}_i{}^{(2)} - \propto_3 \frac{\Delta t_i}{\Omega_i} \boldsymbol{R}\boldsymbol{U}_i{}^{(2)} \end{aligned} \qquad \text{Eq 2-45}$$

In Eq 2-45, $\propto_1, \propto_2, \propto_3$ are the stage coefficients of the third-order Runga-Kutta scheme and are set to 0.333, 0.5, and 1, respectively.

Finally, for a steady-state solution, to accelerate the convergence rate of the solver, a local time-stepping scheme is employed. The local time-stepping scheme employed in the solver is detailed in the following subchapter.

2.1.10 Time Step and Local Time Stepping

In the flow solver, the time step of a cell i is estimated Courant–Friedrichs–Lewy (CFL) criterion expressed in Eq 2-46. The relation is given by Blazek (2001) that refers to (Vijayan & Kallinderis, 1994) for the implementation.

$$\Delta t_i = \sigma \frac{\Omega_I}{(\widehat{\Lambda}_c^x + \widehat{\Lambda}_c^x + \widehat{\Lambda}_c^x)_I + C(\widehat{\Lambda}_\nu^x + \widehat{\Lambda}_\nu^x + \widehat{\Lambda}_\nu^x)_I} \qquad \text{Eq 2-46}$$

In Eq 2-46, σ denotes CFL number. The spectral radii are calculated as given in Eq 2-47.

$$\begin{aligned} \widehat{\Lambda}_c^x &= (|u| + c)\Delta \hat{S}^x \\ \widehat{\Lambda}_c^y &= (|v| + c)\Delta \hat{S}^y \\ \widehat{\Lambda}_c^z &= (|w| + c)\Delta \hat{S}^z \end{aligned} \qquad \text{Eq 2-47}$$

The viscous spectral radii are computed as follows,

$$\widehat{\Lambda}_v^x = \max\left(\frac{4}{3}, \frac{\gamma}{\rho}\right)\left(\frac{\mu_L}{Pr_L} + \frac{\mu_T}{Pr_T}\right)\frac{(\Delta S^x)^2}{\Omega}, etc. \qquad \text{Eq 2-48}$$

where ΔS^x, ΔS^y, ΔS^z, respectively, corresponds to the projected area of the control volume on the *y-z*, *x-z*, and *x-y* plane. Moreover, C is set to 4. Finally, if a time-accurate solution is aimed, the global time step is set to the minimum time step among all cells in the domain, as given in Eq 2-49.

$$\Delta t = \min_i(\Delta t_i) \qquad \text{Eq 2-49}$$

2.1.11 Turbulence Modelling

The Navier-Stokes equations provide a mathematical definition of compressible fluid flow, including turbulent flows. However, in the presence of turbulence, the simulation of fluid flow is problematic. The numerical simulation of turbulence, which is called Direct Numerical Simulation (DNS), requires a discretization in the Kolmogorov scale both in time and space. Therefore, it requires a very large number of cells in the flow domain that scales with $Re^{9/4}$ for a proper spatial resolution. Furthermore, the time step should be small enough to resolve even the smallest scale motion. Despite the high performance of the modern supercomputers, DNS is only feasible for relatively simple flows at low Reynolds numbers. In practical engineering problems, the average quantitative effects of turbulence play a more significant role rather than instantaneous quantitative effects of turbulence that are difficult or infeasible to compute. Therefore, the flow field variables in the Navier-Stokes equation are expressed as the sum of the mean and the fluctuating part that has a zero mean, which is called the Reynolds-Averaged Navier-Stokes (RANS) equations. RANS equations are mostly identical to the Navier-Stokes equations. In RANS equations, the variables are the mean values except the additional time-averaged term $-\rho\overline{u_i u_j}$.

The additional term $-\rho\overline{u_i u_j}$ is called the Reynolds stress because its effect on the mean flow is similar to that of stress term. Although the term is the mean of multiplication of two fluctuating velocities, it is not equal to zero due to the existence of an auto-correlation between the fluctuating velocities. The presence of the Reynolds stress term causes the well-known closure problem in the RANS equations. In order to close the equations, the Reynolds stress term needs to be modeled in terms of the mean flow variables. These models required to close the RANS equations are called turbulence models. Turbulence models may be coupled with a laminar to turbulent transition submodel in order to predict the transition onset automatically. Since the present study considers natural laminar flows, both a turbulent and transition model are implemented into the RANS solver. The turbulence and a laminar to turbulent transition model implemented in this study are discussed in detail in the next subchapter.

2.1.12 Spalart-Allmaras Turbulence Model

In the RANS solver, the closure of the equations is achieved by using the Spalart-Allmaras turbulence model (Spalart & Allmaras, 1992). The model is based on Boussinesq's eddy viscosity hypothesis (Boussinesq, 1877), (Boussinesq, 1896). The hypothesis suggests relating the Reynolds stress tensor linearly with the mean strain rate tensor, as in the viscous stress tensor. In the relation given in Eq 2-50, the proportionality factor is named as the eddy (or turbulent) viscosity, μ_T,

$$-\rho\overline{u_i u_j} = 2\mu_T\overline{\epsilon_{ij}} - \frac{2}{3}\rho k\delta_{ij} \qquad \text{Eq 2-50}$$

where k is the turbulent kinetic energy computed as given in Eq 2-51.

$$k = \frac{1}{2}\overline{u_i u_j} \qquad \text{Eq 2-51}$$

Now, the only unknown variable is the eddy viscosity, and unlike the kinematic viscosity, it does not represent any physical characteristics of the fluid. However, it depends on the local flow field variables. As it is stated earlier to model the eddy viscosity and to achieve closure in RANS equations, the Spalart-Allmaras turbulence model is utilized. This model solves a transport equation for a turbulent working variable, $\check{\nu}$, which is related to the eddy viscosity through,

$$\mu_T = \rho \check{\nu} f_{v_1} \qquad \text{Eq 2-52}$$

where

$$f_{v_1} = \frac{\chi^3}{\chi^3 + C_{v_1}} \qquad \text{Eq 2-53}$$

and

$$\chi = \frac{\check{\nu}}{\nu} \qquad \text{Eq 2-54}$$

The integral form of the equation is given in Eq 2-55.

$$\frac{\partial}{\partial t}\int_{\Omega} \tilde{\nu} d\Omega + \oint_{\partial \Omega} (\boldsymbol{F_{c,T}} - \boldsymbol{F_{v,T}})\, \boldsymbol{n}\, dS = \int_{\Omega} Q_T d\Omega \qquad \text{Eq 2-55}$$

$\boldsymbol{F_{c,T}}$ is the advection of the turbulent working variable,

$$\boldsymbol{F_{c,T}} = \tilde{\nu} V \qquad \text{Eq 2-56}$$

where V is the contravariant velocity. In the solver, the advection of the turbulent working variable is discretized using a scalar upwind scheme.

$\boldsymbol{F_{v,T}}$ is the diffusive flux given by the relation,

$$\boldsymbol{F_{v,T}} = \left(\frac{1}{\sigma}(\nu_L + \check{\nu})\right)\left(n_x \frac{\partial \check{\nu}}{\partial x} + n_y \frac{\partial \check{\nu}}{\partial y} + n_z \frac{\partial \check{\nu}}{\partial z}\right). \qquad \text{Eq 2-57}$$

Q_T is the source term of the turbulent equation. The source term is given in Eq 2-58,

$$Q_T = C_{b1}\tilde{S}\tilde{\nu} + \frac{C_{b2}}{\sigma}\left(\left(\frac{\partial \tilde{\nu}}{\partial x}\right)^2 + \left(\frac{\partial \tilde{\nu}}{\partial y}\right)^2 + \left(\frac{\partial \tilde{\nu}}{\partial z}\right)^2\right) - (C_{w1} f_w)\left(\frac{\tilde{\nu}}{d}\right)^2, \qquad \text{Eq 2-58}$$

where

$$\tilde{S} = \Omega + \frac{\tilde{\nu} f_{v2}}{\kappa^2 d^2}, \Omega = \sqrt{2\epsilon_{ij}\epsilon_{ij}}, f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \qquad \text{Eq 2-59}$$

$$f_w = g\left(\frac{g^{-6} + C_{w3}{}^{-6}}{1 + C_{w3}{}^{-6}}\right)^{-6}, g = r + C_{w2}(r^6 - r),\ r = \frac{\tilde{v}}{\tilde{S}\kappa^2 d^2}, \qquad \text{Eq 2-60}$$

and the constants are given as in Eq 2-61.

$$\begin{gathered} C_{b1} = 0.1355 \qquad \sigma = \frac{2}{3} \qquad C_{b2} = 0.622 \quad C_{v1} = 7.1 \\ C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{(1 + C_{b2})}{\sigma}\ ,\ C_{w2} = 0.3\ ,\ C_{w2} = 2.0 \end{gathered} \qquad \text{Eq 2-61}$$

Furthermore, at solid walls, the value of the turbulent working variable is assumed 0, hence μ_T. The initial dimensionless value of $\check{v}$ is set 3.0, which corresponds to $\mu_T \approx 0.21\mu$. At outflow boundaries, it is extrapolated. Finally, the turbulent working variable $\check{v}$ is added into the flow variable vector.

2.1.13 Bas-Cakmakcioglu Transition Model

The laminar-to-turbulent transition onset prediction plays an essential role in the simulation of flows over NLF airfoils and wings. The original Spalart-Allmaras turbulence models do not predict for laminar-to-turbulent transition in boundary layers. In the present study, the correlation-based algebraic B-C transition model (Cakmakcioglu, Bas, & Kaynak, 2017) is implemented in the flow solver.

In the transition model, an intermittency factor γ_{BC} multiplied with the production term that appears in the source term of the SA turbulence model in order to suppress the production of turbulence in laminar and transition regions, as stated in Eq 2-62.

$$\gamma_{BC} C_{b1}\tilde{S}\tilde{v} + \frac{C_{b2}}{\sigma}\left(\left(\frac{\partial\tilde{v}}{\partial x}\right)^2 + \left(\frac{\partial\tilde{v}}{\partial y}\right)^2 + \left(\frac{\partial\tilde{v}}{\partial z}\right)^2\right) - (C_{w1}f_w)\left(\frac{\tilde{v}}{d}\right)^2 \qquad \text{Eq 2-62}$$

The intermittency factor γ_{BC} is allowed to get a value between 0 and 1. In the model, until a defined transition onset criterion is satisfied, the model suppresses the production of turbulence by setting the intermittency factor to 0. Once the defined criterion is exceeded, the intermittency factor gets a value greater than 0, and the transition from laminar to turbulent flow starts.

The defined criterion is whether locally calculated momentum thickness Reynolds number Re_{θ} exceeds the critical momentum thickness Reynolds number $Re_{\theta c}$ or not. The value of $Re_{\theta c}$ is determined by some empirical correlations related to turbulent intensity value. In the original paper, three relations, each of which corresponds to a different turbulence intensity range, are considered. The first transition onset correlation, which is utilized when the turbulence intensity is below 1%, is as presented in the study of Menter et al. (2004). The correlation is stated in Eq 2-63.

$$Re_{\theta c} = 803.73(T_{U_{\infty}} + 0.6067)^{-1.027} \qquad \text{Eq 2-63}$$

When the turbulence intensity is between 1% and 3%, correlation is similar to the relation delivered in the study of Abu-Ghannam & Shaw (1980),

$$Re_{\theta c} = 163 + e^{(6.91 - T_{U_{\infty}})} \qquad \text{Eq 2-64}$$

and for the turbulent intensity values higher than 3%, the correlation given by Mayle (1991) is employed. The relation is stated in Eq 2-65.

$$Re_{\theta c} = 400\, {T_{U_{\infty}}}^{-0.625} \qquad \text{Eq 2-65}$$

The value of the intermittency factor is determined by the relation given in Eq 2-66.

$$\gamma_{BC} = 1 - e^{\left(-\sqrt{Term_1} - \sqrt{Term_2}\right)} \qquad \text{Eq 2-66}$$

$Term_1$ appears in Eq 2-66 to check the transition criterion. The relation regarding $Term_1$ is given in Eq 2-67.

$$Term_1 = \frac{\max\,(Re_{\theta} - Re_{\theta c}, 0{,}0)}{\chi_1} \qquad \text{Eq 2-67}$$

In the model, Re_{θ} is correlated with the locally calculated vorticity Reynolds number Re_{v} as stated in Eq 2-68.

$$Re_{\theta} = \frac{Re_{v}}{2.193} \qquad \text{Eq 2-68}$$

The vorticity Reynolds number Re_{v} is calculated by Eq 2-69,

$$Re_v = \frac{\rho {d_w}^2}{\mu}\Omega \qquad \text{Eq 2-69}$$

where d_w is the closest wall distance, and Ω is vorticity. In the original paper of the transition model, it is stated that $Term_1$ is not enough to model turbulence generation inside the boundary layer. That is because Re_v is proportional to the closest wall distance, so it gets quite a low value in the vicinity of the wall. To alleviate this issue a new term, $Term_2$, is defined. The term allows intermittency produced by the $Term_1$ to penetrate the boundary layer. $Term_2$ is defined by the relation in Eq 2-70,

$$Term_2 = \frac{\max(\nu_{BC} - \chi_2, 0.0)}{\chi_2} \qquad \text{Eq 2-70}$$

where ν_{BC} is given as

$$\nu_{BC} = \frac{\nu_t}{U d_w}. \qquad \text{Eq 2-71}$$

χ_1 and χ_2 are calibration constants, and the values are presented in Eq 2-72.

$$\chi_1 = 0.002, \chi_2 = 5.0/Re \qquad \text{Eq 2-72}$$

Moreover, when the transition model is enabled, the initial dimensionless value of $\check{\nu}$ is set 0.015 instead of the value of 3.0 set in the turbulence model.

2.1.14 Numerical Implementation and Parallelization

In this subchapter, the implementation issues regarding the solution algorithm, Fortran data structures, and the parallelization of the modified flow solver are considered.

The original flow solver is written in *Fortran 77*. In the present study, the solver converted into *Fortran 95* with the use of proper modules, data structures, and allocatable arrays. In addition, the dimension of the problem is parameterized, and now the developed solver may be employed for the simulation of 2D and 3D flows.

The efficiency of the solution algorithm is also improved by looping over all the cell faces rather than over the cells. Such an approach removes the ‘if’ statements needed to eliminate the computation of edge fluxes twice and makes the implementation of boundary conditions easier.

In accordance with the new solution algorithm, the cell-based data structure is changed to a face-based data structure. The face-based data structure *face_type* encapsulates the left and right cell indices, the normal vector, and the local face indices in the left and right cells. Moreover, all the cell faces are stored in a data structure that stores the array of interior faces, and the arrays of the boundary faces grouped with respect to the boundary type. Thus, the queries required to determine the face properties are avoided.

In order to easily manage the data regarding the grid-related information, a data structure *cell_type* is also introduced. The structure encapsulates the type of the cell, the array of nodes, neighbors, face indices, and the direction of normal vector of faces (either outwards or inwards). All the array sizes are now allocatable to the extent that the size of a variable of *cell_type* changes with the type of the cell.

In the grid input file, the nodes constituting a cell is given in a specific order compatible with the Visual Toolkit (VTK) format. The details of VTK format can be found in the references (Schroeder, Martin, & Lorensen, 2006), (Schroeder, 2010).

In the numerical implementation, another data structure *cellgeom_type* is introduced that encapsulates geometric properties of cells, such as volume, face normals, tangent vectors, center coordinates, the distance to the closest wall. The vector or matrix components of the structure are similarly defined as allocatable arrays. Furthermore, in order to reduce arithmetic operations, some additional terms needed frequently in the computations are also encapsulated by the data structure.

The computation of convective and viscous fluxes are now achieved by procedure pointers, which are easily set depending on the selected scheme. Thus, the queries performed before each call of the flux function regarding the type, the order of the selected scheme, and the dimension of the problem are avoided, and the readability of the code is improved significantly. The use of modern tools in programming with Fortran 95 improves computational efficiency significantly.

In the development of the solver, it is intended to write a *cache-friendly* code by considering memory latency. Memory latency is the time until the processor retrieves the requested data. The latency depends on the location of the data. The typical latency values of modern computers are given in Table 2.1.

Table 2.1 Typical latency values in modern computer architecture (Ajwani & Meyerhenke, 2010)

Cache Type	Latency (nanoseconds)
Register	~1
Caches	~10
Main Memory	~5-70
Hard Disk	~10000

In order to reduce the memory latency, frequently accessed data appearing in large arrays are assigned to local variables, in this way, the temporal locality is considered. Moreover, the spatial locality is also essential in a *cache-friendly* code. Hence, loops are constructed by considering column-major order (used by *Fortran*) that is each column is juxtaposed one after the other in contiguous memory locations.

Most of the present study is accomplished on a shared-memory rack server system having 32 cores. Therefore, the solver is easily parallelized by using OpenMP version 5.0 (OpenMP Architecture Review Board, 2018) compiler directives. The algorithm preserves the program structure of the serial version, and the parallel version of the solver is generated by compiler directives added before the main loops. Most of the loops at the top level of the solution algorithm are parallelized. Since most of the loop sizes are much larger than the number of cores, the nested level parallelism is not preferred. The speedup ratios of the computations are plotted in Figure 2-3 in terms of the elapsed time per iteration for a three-dimensional problem with 760×10^3 cells.

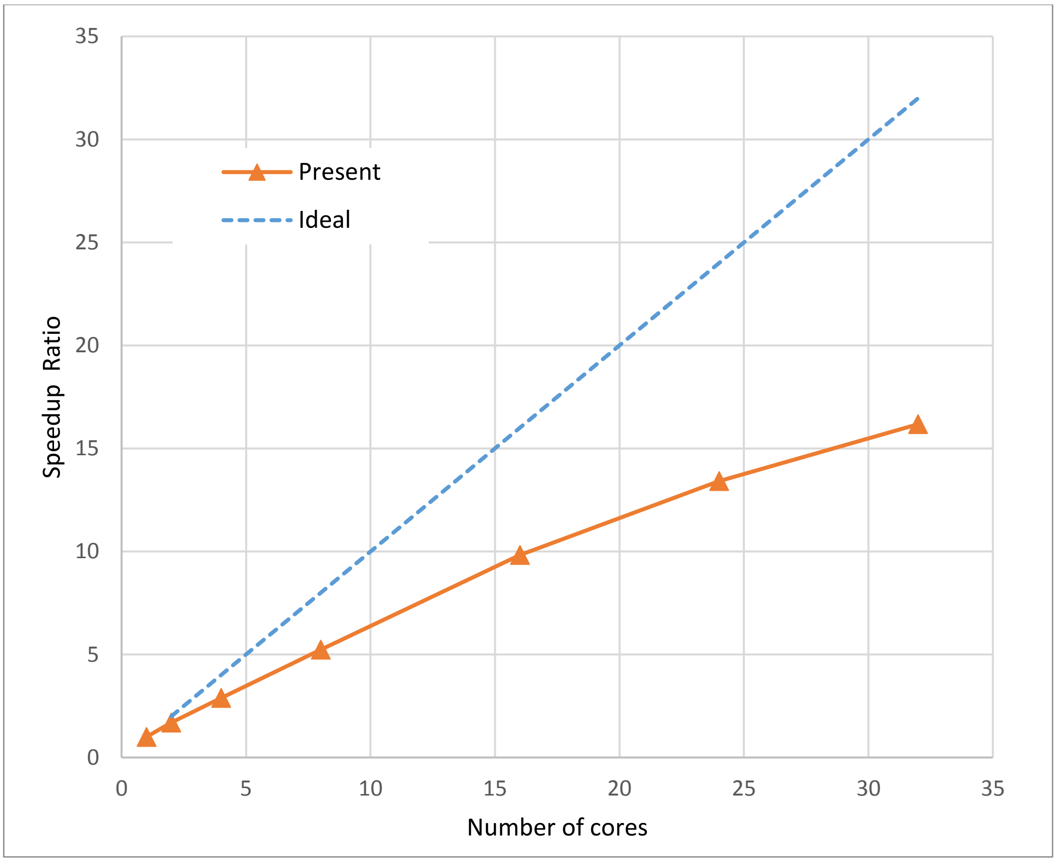


Figure 2-3 Parallel Efficiency

As seen in Figure 2-3 the parallel efficiency rapidly deteriorates, and 32 cores provide about 51% efficiency. Although the scalability of the computations may further be improved by the use of a profiler to detect the bottlenecks, it is left for a future study.

2.2 Sensitivity Analysis

2.2.1 Adjoint Equations

In an aerodynamic shape optimization problem, objective function I, such as C_L or C_D, depend on the geometric variables that define the aerodynamic surfaces and the flow field $\boldsymbol{U}$.

$$I = I\big(\boldsymbol{\alpha}, \boldsymbol{U}(\boldsymbol{\alpha})\big) \qquad \text{Eq 2-73}$$

A subset of geometric variables $\boldsymbol{\alpha}$, which define the aerodynamic surface to be optimized, is taken as the design variables. Since all the flow variables in the solution domain, $\boldsymbol{U}$, depend on the geometric variables through the boundary conditions, the governing equations of flow, in the study the RANS equations, now become the constraints of the aerodynamic shape optimization problem.

$$\boldsymbol{R} = \boldsymbol{R}\big(\boldsymbol{\alpha}, \boldsymbol{U}(\boldsymbol{\alpha})\big)=0 \qquad \text{Eq 2-74}$$

The sensitivity of the objective functions to the design variables may be expressed as

$$\frac{d\boldsymbol{I}}{d\boldsymbol{\alpha}} = \frac{\partial \boldsymbol{I}}{\partial \boldsymbol{\alpha}} + \frac{\partial \boldsymbol{I}}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\boldsymbol{\alpha}} \qquad \text{Eq 2-75}$$

On the other hand, the sensitivity of the governing fluid flow equations is similarly given as in Eq 2-76. Since the governing equations should always be satisfied, they are equal to zero.

$$\frac{d\boldsymbol{R}}{d\boldsymbol{\alpha}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\boldsymbol{\alpha}} = 0 \qquad \text{Eq 2-76}$$

Note that the total derivative term, $d\boldsymbol{U}/d\boldsymbol{\alpha}$, which is present in both expressions is the most challenging term to evaluate due to the inherent coupling between the flow variables and the boundary conditions. If it is computed by a one-sided finite difference method, as many flow solutions as the design variables are needed.

Note that Eq 2-76 leads to the following expression

$$\frac{d\boldsymbol{U}}{d\boldsymbol{\alpha}} = -\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{-1}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} \qquad \text{Eq 2-77}$$

Moreover, by substituting this statement into the sensitivity equation, the following statement is attained.

$$\frac{dI}{d\boldsymbol{\alpha}} = \frac{\partial I}{\partial \boldsymbol{\alpha}} - \underbrace{\frac{\partial I}{\partial \boldsymbol{U}} \overbrace{\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{-1} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}}^{-d\boldsymbol{U}/d\boldsymbol{\alpha}}}_{\boldsymbol{\lambda}^{T}} \qquad \text{Eq 2-78}$$

Now, there are two ways to solve the linear system given in Eq 2-78, depending on which right-hand side is chosen. In the first method, the total derivative $d\boldsymbol{U}/d\boldsymbol{\alpha}$ is calculated directly, the linear system given in Eq 2-79. Therefore, it is called the direct method.

$$-\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\frac{d\boldsymbol{U}}{d\boldsymbol{\alpha}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} \qquad \text{Eq 2-79}$$

The alternative method is the adjoint method. In the adjoint method, a vector called the adjoint vector is introduced, and the linear system in Eq 2-78 is defined only by partial derivatives as,

$$\frac{dI}{d\boldsymbol{\alpha}} = \frac{\partial I}{\partial \boldsymbol{\alpha}} + \boldsymbol{\lambda}^{T}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} \qquad \text{Eq 2-80}$$

where the adjoint vector $\boldsymbol{\lambda}$ is

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{\boldsymbol{T}} \boldsymbol{\lambda} = -\left[\frac{\partial I}{\partial \boldsymbol{U}}\right]^{\boldsymbol{T}} \qquad \text{Eq 2-81}$$

Thus, in the direct method, the computational cost to solve the linear system given in Eq 2-79 scales with the number of design variables, $\boldsymbol{\alpha}$, however, in the adjoint method, the computational cost to solve the linear system given in Eq 2-81 scales with the number of objective functions. Furthermore, typically, aerodynamic shape optimization problems consider a single or a few objective functions like C_L/C_D, ${C_L}^{3/2}/C_D$ etc. However, the shape is generally defined and controlled by tens or hundreds of design variables. Hence, the adjoint method provides a prominent computational advantage in dealing with aerodynamic shape optimization problems.

The adjoint formulation may also be defined by employing a Lagrange multiplier. By using a Lagrange multiplier to enforce the governing equation as constraints, an augmented objective function may be introduced.

$$J = I(\boldsymbol{\alpha}, \boldsymbol{U}(\boldsymbol{\alpha})) + \boldsymbol{\lambda}^T \boldsymbol{R}(\boldsymbol{\alpha}, \boldsymbol{U}(\boldsymbol{\alpha})) \qquad \text{Eq 2-82}$$

The sensitivity of the augmented objective function to the design variables may be expressed as given in the following equation.

$$dJ = \frac{\partial I}{\partial \boldsymbol{U}} d\boldsymbol{U} + \frac{\partial I}{\partial \boldsymbol{\alpha}} d\boldsymbol{\alpha} + \boldsymbol{\lambda}^T \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} d\boldsymbol{U} + \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} d\boldsymbol{\alpha} \right) \qquad \text{Eq 2-83}$$

When the equation is grouped considering the perturbations, the following statement appears.

$$dJ = \left(\frac{\partial I}{\partial \boldsymbol{U}} + \boldsymbol{\lambda}^{\boldsymbol{T}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \right) d\boldsymbol{U} + \left(\frac{\partial \mathrm{I}}{\partial \boldsymbol{\alpha}} + \boldsymbol{\lambda}^{\boldsymbol{T}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} \right) d\boldsymbol{\alpha} \qquad \text{Eq 2-84}$$

When $\boldsymbol{\lambda}^{\boldsymbol{T}}$ is chosen to satisfy the adjoint equation,

$$\frac{\partial I}{\partial \boldsymbol{U}} + \boldsymbol{\lambda}^{\boldsymbol{T}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} = 0 \Rightarrow \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}} \right]^{\boldsymbol{T}} \boldsymbol{\lambda} = - \left[\frac{\partial I}{\partial \boldsymbol{U}} \right]^{\boldsymbol{T}} \qquad \text{Eq 2-85}$$

the sensitivities are then obtained by the following equation.

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{dI}{d\boldsymbol{\alpha}} = \frac{\partial I}{\partial \boldsymbol{\alpha}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}} \qquad \text{Eq 2-86}$$

Finally, in the present study, due to having a domain discretized through finite volumes, residual equations and flow variables are also a function of grid coordinates $\boldsymbol{X}$. Thus, Eq 2-80 (or Eq 2-86) may be rewritten, as depicted in Eq 2-87.

$$\frac{dI}{d\boldsymbol{X}} \frac{d\boldsymbol{X}}{d\boldsymbol{\alpha}} = \left(\frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}} \right) \frac{d\boldsymbol{X}}{d\boldsymbol{\alpha}} \qquad \text{Eq 2-87}$$

Accordingly, in the present study, firstly, the sensitivity derivatives of the objective function with respect to grid nodes, and the sensitivity derivatives of the grid nodes with respect to design variables are computed separately. Afterward, the sensitivities of the objective function with respect to design variables are computed by performing the dot product given in Eq 2-87.

2.2.2 Automatic Differentiation

The adjoint approach requires the evaluation of the partial derivative terms of $\partial \boldsymbol{R}/\partial \boldsymbol{U}$, $\partial \boldsymbol{R}/\partial \boldsymbol{\alpha}$, $\partial I/\partial \boldsymbol{U}$, $\partial I/\partial \boldsymbol{\alpha}$ which appear in Eq 2-80, Eq 2-81. There are quite a few ways to compute these partial derivatives, such as by finite difference or complex-steps; however, the computational cost of these methods is prohibitive (Lyu, Kenway, Paige, & Martins, 2013). Another way is to derive these partial derivatives by differentiating all the related terms in the solver code by hand. However, this method is error-prone, requires lengthy development time, and is non-trivial for complex CFD solvers.

In the present study, an automatic differentiation (AD) tool is employed to evaluate these partial derivatives. AD is a method based on the systematic application of the differentiation by chain rule to a computer program (Griewank & Walther, 2008). The method is as accurate as of the analytical differentiation by hand. Moreover, since the method automatically differentiates the computer program, the development requires much less effort comparing to differentiating residual and objective function routines by hand.

A computer program consists of a finite set of elementary operations. The derivatives of these elementary functions are known. An AD tool scans the computer program line by line, identifies the expressions with the variables to be differentiated, and systematically applies the chain rule to these elementary functions. It finally produces a new code that evaluates the partial derivatives described by the user. There are two possible modes of AD. The first one is the *forward (or tangent) mode*. In this case, the chain rule is applied from bottom to up in the order of computations. The variable is differentiated as soon as it is evaluated. In this mode, the total derivatives of functions with respect to an independent variable are computed.

The other mode is the *reverse (or adjoint) mode*. In this mode, the algorithm firstly computes all the intermediate variables. Subsequently, it performs a reverse sweep, accumulating the derivatives from output to input. In other words, the chain rule is applied from top to bottom. In this way, by performing one sweep, the total derivatives of an output function with respect to all independent variables are computed.

There are two different implementations for an AD tool. The first one is the source transformation, and the latter is the operator overloading. The source transformation approach introduces new variables for derivatives and adds new source lines computing derivatives to the original code. On the other hand, the operator overloading approach does not make any changes in source code, but it changes the definition of operations and variable type. The overloaded operations now evaluate the derivatives of the function as well as the function itself, and the types of the variables are modified in such a way that they also contain the corresponding derivative values, as well as their original values (Martins & Hwang, 2013). There are numerous AD tools available for most of the programming languages. As examples of AD tools using source-code transformation approach, there are *ADIC* (Bischof, Roh, & Mauer, 1997), *OpenAD* (Utke, 2004) for *C/C++*; *OpenAD/F* (Utke, et al., 2008), *TAPENADE* (Hascoet & Pascual, 2013), *TAF* (Giering & Kaminski, 1998), *ADIFOR* (Bischof, Khademi, Mauer, & Carle, 1996), *ADF95* (Straka, 2005) for *Fortran*. *CoDiPack* (Sagebaum, Albring, & Gauger, 2019), *ADOL-C* (Griewank, Juedes, & Utke, 1996), *ADOL-F* (Shiriaev & Griewank, 1996) are some examples of AD tools using operator overloading approach. Furthermore, there are also AD tools available for *Java, Julia, Python*, *MATLAB*, R, *OpenCL*, *.NET*, *C#*, *Delphi*, etc. In the present study, the AD tool *TAPENADE* (Hascoet & Pascual, 2013) that utilizes the source-code transformation approach is employed.

In order to provide a better understanding of the source-code transformation approach, the numerical example is given below.

$$f(u_3, u_2, u_1, x) = x + u_1 u_3 + u_2 \sin(u_1)$$

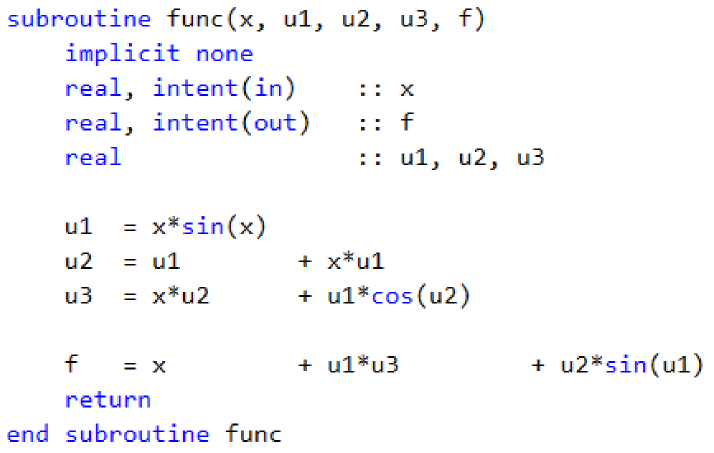
where Eq 2-88

$$u_3 = x u_2 + u_1 \cos(u_2)$$

$$u_2 = u_1 + xu_1$$

$$u_1 = xsin(x).$$

In this example, x is the input variable, which is an independent variable. u_1, u_2 and u_3 are the intermediate dependent variables. f is the output variable. The corresponding computer program in *Fortran* may be written as follows:



```
subroutine func(x, u1, u2, u3, f)
    implicit none
    real, intent(in)    :: x
    real, intent(out)   :: f
    real                :: u1, u2, u3

    u1  = x*sin(x)
    u2  = u1        + x*u1
    u3  = x*u2      + u1*cos(u2)

    f   = x         + u1*u3         + u2*sin(u1)
    return
end subroutine func
```

Figure 2-4 Sample Fortran code

AD tool takes all the variables as a vector, $\boldsymbol{v} = [x\ u_1\ u_2\ u_3\ f]^T$=$[v_1\ v_2\ v_3\ v_4\ v_5]^T$, and treats each line of the source code as a function of $\boldsymbol{V}$.

$$\boldsymbol{V} = \begin{bmatrix} x \\ xsin(x) \\ u_1 + xu_1 \\ xu_2 + u_1 cos(u_2) \\ x + u_1 u_3 + u_2 sin(u_1) \end{bmatrix} \qquad \text{Eq 2-89}$$

The forward mode of AD applies the chain rule.

$$\frac{dv_i}{dv_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial V_i}{\partial v_k} \frac{dv_k}{dv_j} \qquad \text{Eq 2-90}$$

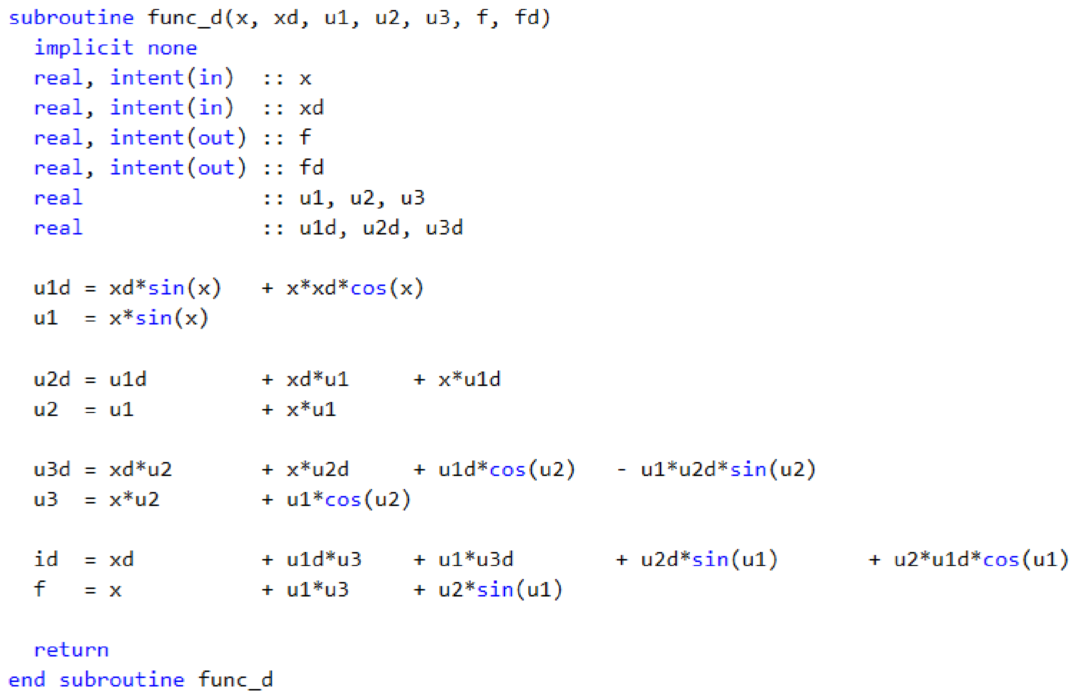
The system generated by the forward mode of AD that computes df/dx $(\mathrm{i.e.},\ dv_5/dv_1)$ may be illustrated in matrix form, as in Eq 2-91.

$$\begin{bmatrix} 1 & & & & \\ -\frac{\partial V_2}{\partial v_1} & 1 & & & \\ -\frac{\partial V_3}{\partial v_1} & -\frac{\partial V_3}{\partial v_2} & 1 & & \\ -\frac{\partial V_4}{\partial v_1} & -\frac{\partial V_4}{\partial v_2} & -\frac{\partial V_4}{\partial v_3} & 1 & \\ -\frac{\partial V_5}{\partial v_1} & -\frac{\partial V_5}{\partial v_2} & -\frac{\partial V_5}{\partial v_3} & -\frac{\partial V_5}{\partial v_4} & 1 \end{bmatrix} \begin{bmatrix} \frac{dv_1}{dv_1} \\ \frac{dv_2}{dv_1} \\ \frac{dv_3}{dv_1} \\ \frac{dv_4}{dv_1} \\ \frac{dv_5}{dv_1} \end{bmatrix} = \boldsymbol{I} \qquad \text{Eq 2-91}$$

Note that each variable can only depend on the independent variables and the intermediate variables computed earlier. Therefore, the system always forms a lower triangular matrix. Hence, it is possible to compute the total derivatives with respect to *x* by using forward substitution. In Eq 2-92, the system illustrated in Eq 2-91 is given by equations.

$$\begin{gathered} dx = d\mathrm{x} \\ du_1 = \sin(x)dx + xcos(x)dx \\ du_2 = du_1 + u_1dx + xdu_1 \\ du_3 = u_2dx + xdu_2 + cos(u_2)du_1 - u_1sin(u_2)du_2 \\ df = dx + u_3du_1 + u_1du_3 + sin(u_1)du_2 + u_2\cos(u_1)du_1 \end{gathered} \qquad \text{Eq 2-92}$$

In order to generate a subroutine that computes the total derivatives with respect to *x*, AD adds each equation to the original subroutine line by line as a statement. Each statement is placed before the statement that computes the corresponding variable. The subroutine generated by *TAPENADE* for the example given above is depicted in Figure 2-5.



```
subroutine func_d(x, xd, u1, u2, u3, f, fd)
  implicit none
  real, intent(in)  :: x
  real, intent(in)  :: xd
  real, intent(out) :: f
  real, intent(out) :: fd
  real              :: u1, u2, u3
  real              :: u1d, u2d, u3d

  u1d = xd*sin(x)   + x*xd*cos(x)
  u1  = x*sin(x)

  u2d = u1d         + xd*u1     + x*u1d
  u2  = u1          + x*u1

  u3d = xd*u2       + x*u2d     + u1d*cos(u2)   - u1*u2d*sin(u2)
  u3  = x*u2        + u1*cos(u2)

  id  = xd          + u1d*u3    + u1*u3d        + u2d*sin(u1)       + u2*u1d*cos(u1)
  f   = x           + u1*u3     + u2*sin(u1)

  return
end subroutine func_d
```

Figure 2-5 Subroutine generated by *TAPENADE* in the forward mode

dx (*xd* in the subroutine) is the input variable of the subroutine that *TAPENADE* generates. Hence, when it is intended to compute the total derivatives with respect to *x*, the input *xd* is given as 1, and the output variable *fd* will be the value of df/dx. That means in a forward mode by performing one forward substitution, the total derivatives with respect to one input variable can be computed. So, in order to compute all total derivatives with respect to input variables in a system, the forward mode requires solving as many systems as in the number of input variables. Thus, it is analogous to the direct method explained in the previous subchapter.

On the other hand, the reverse mode of AD applies the chain rule from top to bottom, as is given below.

$$\frac{dv_i}{dv_j} = \delta_{ij} + \sum_{k=j+1}^{i} \frac{dv_i}{dv_k}\frac{\partial V_k}{\partial v_j} \qquad \text{Eq 2-93}$$

The system generated by the reverse mode of AD that computes df/dx (dv_5/dv_1) may be illustrated in matrix form, as in Eq 2-94.

$$\begin{bmatrix} 1 & -\frac{\partial V_2}{\partial v_1} & -\frac{\partial V_3}{\partial v_1} & -\frac{\partial V_4}{\partial v_1} & -\frac{\partial V_5}{\partial v_1} \\ & 1 & -\frac{\partial V_3}{\partial v_2} & -\frac{\partial V_4}{\partial v_2} & -\frac{\partial V_5}{\partial v_2} \\ & & 1 & -\frac{\partial V_4}{\partial v_3} & -\frac{\partial V_5}{\partial v_3} \\ & & & 1 & -\frac{\partial V_5}{\partial v_4} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} \frac{dv_5}{dv_1} \\ \frac{dv_5}{dv_2} \\ \frac{dv_5}{dv_3} \\ \frac{dv_5}{dv_4} \\ \frac{dv_5}{dv_5} \end{bmatrix} = \boldsymbol{I} \tag{Eq 2-94}$$

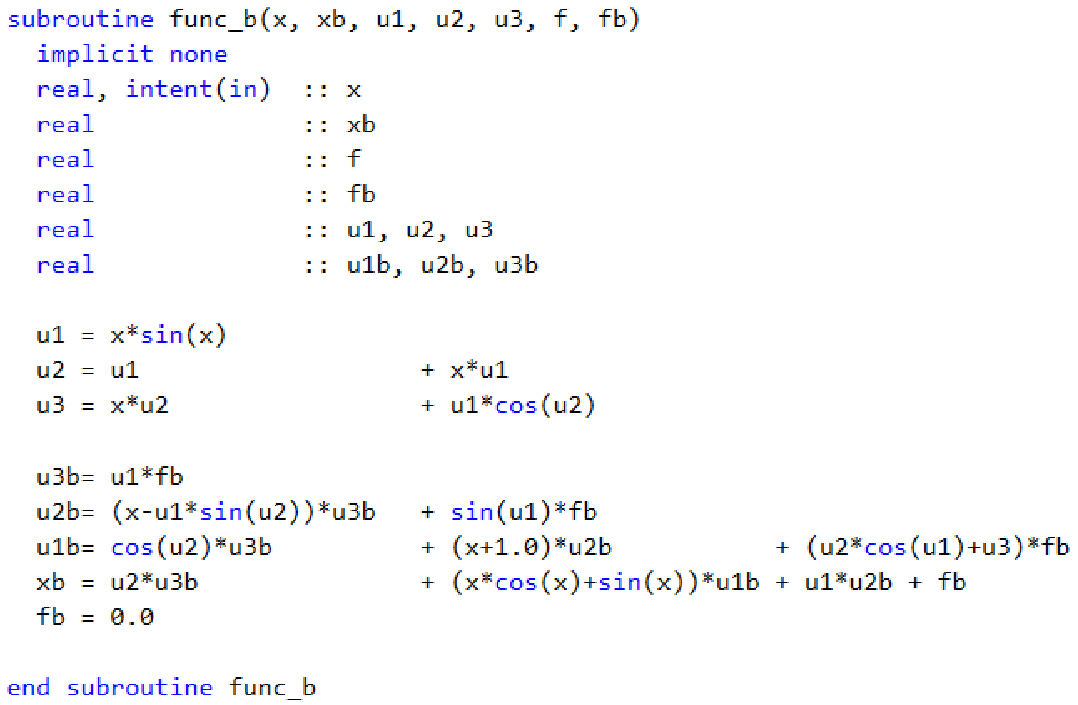
In Eq 2-95, the system to compute the total derivatives of dt_5 is given.

$$\begin{gathered} \bar{f} = 1 \\ \overline{u_3} = u_1\bar{f} \\ \overline{u_2} = \big(x - u_1 sin(u_2)\big)\overline{u_3} + sin(u_1)\bar{f} \\ \overline{u_1} = \ cos(u_2)\overline{u_3} + (x+1)\overline{u_2} + (u_2 cos(u_1) + u_3)\bar{f} \\ \bar{x} = u_2\overline{u_3} + \big(xcos(x) + sin(x)\big)\overline{u_1} + u_1\overline{u_2} + \bar{f}, \end{gathered} \tag{Eq 2-95}$$

where

$$\overline{(\quad)} = df/d(\quad).$$

In order to generate a subroutine that computes the total derivatives of *f*, AD firstly computes variables. AD then adds each equation to the original subroutine line by line as a statement. The subroutine generated for the numerical example is similarly depicted in Figure 2-6.



```
subroutine func_b(x, xb, u1, u2, u3, f, fb)
  implicit none
  real, intent(in)  :: x
  real              :: xb
  real              :: f
  real              :: fb
  real              :: u1, u2, u3
  real              :: u1b, u2b, u3b

  u1 = x*sin(x)
  u2 = u1                   + x*u1
  u3 = x*u2                 + u1*cos(u2)

  u3b= u1*fb
  u2b= (x-u1*sin(u2))*u3b   + sin(u1)*fb
  u1b= cos(u2)*u3b          + (x+1.0)*u2b           + (u2*cos(u1)+u3)*fb
  xb = u2*u3b               + (x*cos(x)+sin(x))*u1b + u1*u2b + fb
  fb = 0.0

end subroutine func_b
```

Figure 2-6 Subroutine generated by *TAPENADE* in reverse mode

In reverse mode, all variables need to be computed and stored since the algorithm computes the total derivatives in reverse order starting from the total derivative of *f* with respect to the variable that is computed the last. By using the intermediate variables stored, the total derivatives of the output function with respect to all the variables are evaluated in reverse order. Since a variable that is evaluated earlier can not be a function of a variable that is formed later, the reverse mode always forms an upper triangular matrix. Hence it is possible to compute the total derivatives of the output function with respect to all variables by performing one backward substitution. So, in order to compute all the derivatives of the output function with respect to all the input variables, reverse mode requires solving as many systems as in the number of output variables.

As a summary, an AD tool, by applying the chain rule systematically either from bottom to top or top to bottom, generates a new subroutine that computes the derivatives of functions coded in a computer program. Since the derivatives are exact, the AD tool is as accurate as an analytical method, and the implementation is straightforward. Further details of mathematical formulation, derivations, and examples of the forward and reverse modes can be found in References (Griewank & Walther, 2008), (Naumann, 2011), (Martins & Hwang, 2013).

In the study, AD is employed in a way that is similar to the approach proposed by Mader, Martins, Alonso, & van der Weide (2008), with some differences in the implementation. The details of the implementation are introduced in the following subchapters.

2.2.3 Computation of Flux Jacobian Matrices

The flux Jacobian matrix $\partial \boldsymbol{R}/\partial \boldsymbol{U}$ in the adjoint equation given in Eq 2-81 contains the sensitivity of cell residuals $\boldsymbol{R}_i$, with respect to all flow variables of the cells involved in its evaluation $\boldsymbol{U}_{j,k}$. The matrix is of the dimension $N_U N_{cell} \times N_U N_{cell}$, where N_U is the number of the flow variables and N_{cell} is the number of cells in the computational domain. Accordingly, the flux Jacobian matrix $\partial \boldsymbol{R}/\partial \boldsymbol{U}$ is given as is in Eq 2-96. Note that, in a turbulent flow case $\boldsymbol{R}_i$ includes the residual of the eddy viscosity model and $\boldsymbol{U}_j$ similarly includes the working variable of the turbulence model.

$$\begin{bmatrix}
\frac{\partial R_{1,1}}{\partial U_{1,1}} & \frac{\partial R_{1,1}}{\partial U_{2,1}} & \cdots & \frac{\partial R_{1,1}}{\partial U_{N_U,1}} & \frac{\partial R_{1,1}}{\partial U_{N_U,2}} & \cdots & \frac{\partial R_{1,1}}{\partial U_{N_U,N_{cell}}} \\
\frac{\partial R_{2,1}}{\partial U_{1,1}} & \frac{\partial R_{2,1}}{\partial U_{2,1}} & \cdots & \frac{\partial R_{2,1}}{\partial U_{N_U,1}} & \cdots & & \frac{\partial R_{2,1}}{\partial U_{N_U,N_{cell}}} \\
\vdots & \vdots & \cdots & \vdots & \cdots & & \vdots \\
\frac{\partial R_{N_U,1}}{\partial U_{1,1}} & \frac{\partial R_{N_U,1}}{\partial U_{2,1}} & \cdots & \frac{\partial R_{N_U,1}}{\partial U_{N_U,1}} & \cdots & & \frac{\partial R_{N_U,1}}{\partial U_{N_U,N_{cell}}} \\
\frac{\partial R_{1,2}}{\partial U_{1,1}} & \vdots & \vdots & \vdots & \ddots & & \frac{\partial R_{1,2}}{\partial U_{N_U,N_{cell}}} \\
\vdots & & & & & \ddots & \vdots \\
\frac{\partial R_{N_U,N_{cell}}}{\partial U_{1,1}} & \frac{\partial R_{N_U,N_{cell}}}{\partial U_{2,1}} & \cdots & \frac{\partial R_{N_U,N_{cell}}}{\partial U_{N_U,1}} & \frac{\partial R_{N_U,N_{cell}}}{\partial U_{N_U,2}} & \cdots & \frac{\partial R_{N_U,N_{cell}}}{\partial U_{N_U,N_{cell}}}
\end{bmatrix} \tag{Eq 2-96}$$

In Eq 2-96, the first subscript of R is numbered after the residual of the conservation of mass, momentum, and turbulence equations, respectively. The first subscript of U denotes the solution variable, and the second subscript denotes the cell index.

The flux Jacobian matrix $\partial\boldsymbol{R}/\partial\boldsymbol{X}$ in Eq 2-87 contains the partial derivatives of the cell residuals $\boldsymbol{R}_i$ with respect to the coordinates of the grid nodes $\boldsymbol{X}_j$. The dimension of the matrix is $N_U N_{cell} \times N_{dim} N_{node}$, where N_{dim} and N_{node} stand for the dimension of the problem, and the number of grid nodes in the computational domain, respectively. The flux Jacobian matrix $\partial\boldsymbol{R}/\partial\boldsymbol{X}$ is then expressed given as is in Eq 2-97.

$$\begin{bmatrix}
\frac{\partial R_{1,1}}{\partial X_{1,1}} & \cdots & \frac{\partial R_{1,1}}{\partial X_{N_{dim},1}} & \frac{\partial R_{1,1}}{\partial X_{1,2}} & \cdots \quad \cdots & \frac{\partial R_{1,1}}{\partial X_{N_{dim},N_{node}}} \\
\frac{\partial R_{2,1}}{\partial X_{1,1}} & \cdots & \frac{\partial R_{2,1}}{\partial X_{N_{dim},1}} & \frac{\partial R_{2,1}}{\partial X_{1,2}} & \cdots & \frac{\partial R_{2,1}}{\partial X_{N_{dim},N_{node}}} \\
\vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
\frac{\partial R_{N_U,1}}{\partial X_{1,1}} & \cdots & \frac{\partial R_{N_U,1}}{\partial X_{N_{dim},1}} & \frac{\partial R_{N_U,1}}{\partial X_{1,2}} & \cdots & \frac{\partial R_{N_U,1}}{\partial X_{N_{dim},N_{node}}} \\
\frac{\partial R_{1,2}}{\partial X_{1,1}} & \vdots & \vdots & \vdots & \ddots & \frac{\partial R_{1,2}}{\partial X_{N_{dim},N_{node}}} \\
\vdots & & & & \ddots & \vdots \\
\frac{\partial R_{N_U,N_{cell}}}{\partial X_{1,1}} & \cdots & \frac{\partial R_{N_U,N_{cell}}}{\partial X_{N_{dim},1}} & \frac{\partial R_{N_U,N_{cell}}}{\partial X_{1,2}} & \cdots \quad \cdots & \frac{\partial R_{N_U,N_{cell}}}{\partial X_{N_{dim},N_{node}}}
\end{bmatrix} \tag{Eq 2-97}$$

As in $\partial\boldsymbol{R}/\partial\boldsymbol{U}$ expression, the first subscript of R denotes the governing equations, and the first subscript of X denotes the x, y, and z coordinates, respectively, whereas the second subscripts denote the node index.

Prior to employing the AD tool to generate the modules, which provide the partial derivatives needed in the adjoint formulation, a *Fortran* routine that evaluates the cell residuals is recoded for compactness and numerical efficiency. As stated earlier (subchapter 2.1.14), the solution algorithm of the flow solver loops over the cell faces and computes all the face fluxes. The residuals of a cell $\boldsymbol{R}_i$ then evaluated by summing the face fluxes of the cell. Hence, a face-based loop rather than a cell-based loop is employed in the flow solver. However, it would be more convenient to have a cell-based loop to be used with the AD tool to evaluate the cell-based derivatives, $\partial \boldsymbol{R}/\partial \boldsymbol{U}$. Therefore, a stand-alone subroutine that computes the residuals of a single cell is formed. For a first-order scheme, this subroutine accepts the flow variables and the node coordinates of the cell and the neighboring cells as input arguments. For a second-order scheme, the flow variables and the coordinates of the cell nodes of the cells, which are neighbor to the immediate neighbor cells, are also included among the input arguments. The other input arguments of the subroutine are either constant values or not explicitly dependent on the flow variables and node coordinates. Therefore the subroutine recomputes all intermediate variables required to compute the cell residuals, such as turbulent viscosity, gradients of flow variables, face normals, the volume of cells. As a result, the stand-alone subroutine includes all the functional relations needed for the evaluation of the cell residual, $\boldsymbol{R}_i$.

The stand-alone subroutine is now used with the AD tool *TAPENADE* that applies the chain rule line by line and generates a subroutine that computes the partial derivatives of residuals of a cell *i* with respect to a flow variable *j* of a dependent cell *k*, $\partial \boldsymbol{R}_i/\partial \boldsymbol{U}_{j,k}$**.** By calling the subroutine for each flow variable of each dependent cells, the partial derivatives of residuals of a cell are computed. Similarly, when the subroutine is differentiated with respect to the grid node coordinates, *TAPENADE* applies the chain rule line by line and generates a subroutine that computes the partial derivatives of residuals of a cell i with respect to a coordinate *t* of a dependent node *l*, which is $\partial \boldsymbol{R}_i/\partial X_{t,l}$. By calling the subroutine for each coordinate of each dependent node, the partial derivatives of residuals of a cell are computed. Finally, by looping over all the cells in the computational domain, the Jacobian matrices $\partial \boldsymbol{R}/\partial \boldsymbol{U}$, $\partial \boldsymbol{R}/\partial \boldsymbol{X}$ are constructed. Since residuals of a cell only depend on a relatively small number of neighbor cells, the CPU time required to compute the Jacobian matrices is relatively short.

It should be noted that the S-A model equation and the intermittency factor appearing in the B-C transition model depend on wall distances, which creates an additional dependency of cell residuals on the wall distances. In order to simplify the construction of the Jacobian matrices, this dependence is removed by providing wall distances as constant arguments. Thus, the implicit relationship between the wall distances and the node coordinates is concealed. A similar approach is also implemented in the study of Lyu, Kenway, Paige, and Martins (2013). Furthermore, to reduce the error introduced, the grid in the vicinity of the wall boundaries is deformed as small as possible (detailed in subchapter 2.3.1).

In addition, since the limiters used in the second-order variable reconstruction do not continuously depend on the flow variables and therefore are not differentiable, they are removed from the residual computations in the stand-alone subroutine.

2.2.4 Computation of Partial Sensitivities of Objective Function

The computation of partial sensitivity of the objective function is accomplished by a similar approach given in the previous subchapter. In order to generate the subroutines computing the partial derivatives of an objective function by using *TAPENADE*, firstly, a subroutine that defines all explicit relations between the contribution of a cell to the objective function and flow variables and node coordinates $I\left(\widehat{\boldsymbol{U}},\widehat{\boldsymbol{X}}\right)$ is written, where $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{X}}$ refer to the flow variables and node coordinates of the cell, the neighbor cells, and the next-neighboring cells, respectively. The subroutine accepts flow variables and grid nodes of neighboring cells as input arguments. Moreover, the other input arguments are either constant values or not explicitly-dependent on flow variables or grid. The subroutine performs all corresponding computation of the contribution of a cell to the objective function. AD is then employed to differentiate the subroutine with respect to grid coordinates and the flow variables. In this way, the subroutines that evaluate the sensitivities of the objective function are created. Finally, the sensitivities to each node coordinate and flow variable are evaluated by calling these subroutines.

The dimension of the vector $\partial I/\partial \boldsymbol{U}$ is $N_U N_{cell}$. The appearance of the vector is given in Eq 2-98.

$$\left[\frac{\partial I}{\partial U_{1,1}} \quad \frac{\partial I}{\partial U_{2,1}} \quad \cdots \quad \frac{\partial I}{\partial U_{N_U,1}} \quad \frac{\partial I}{\partial U_{1,2}} \quad \cdots \quad \frac{\partial I}{\partial U_{N_{U-1},N_{cell}}} \quad \frac{\partial I}{\partial U_{N_U,N_{cell}}}\right] \qquad \text{Eq 2-98}$$

The partial sensitivity vector of an objective function with respect to grid node coordinates $\partial I/\partial \boldsymbol{X}$ is given in Eq 2-99. The vector is in the dimension of $N_{dim}N_{node}$.

$$\left[\frac{\partial I}{\partial X_{1,1}} \quad \cdots \quad \frac{\partial I}{\partial X_{N_{dim},1}} \quad \frac{\partial I}{\partial X_{1,2}} \quad \cdots \quad \frac{\partial I}{\partial U_{N_{dim-1},N_{node}}} \quad \frac{\partial I}{\partial U_{N_{dim},N_{node}}}\right] \qquad \text{Eq 2-99}$$

In the present study, the objective functions are functions of aerodynamic loads, such as C_L/C_D. However, it is possible to define any objective functions which are a function of flow variables or/and grid, e.g., total pressure recovery of an engine inlet.

2.2.5 Solution of the Adjoint System of Equations

The computation of the partial derivatives is followed by the solution of the adjoint system given in Eq 2-81. Once the adjoint variables are computed, the sensitivity derivatives $dI/d\boldsymbol{X}$ are easily evaluated by Eq 2-100.

$$\frac{dI}{d\boldsymbol{X}} = \frac{\partial I}{\partial \boldsymbol{X}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{X}} \qquad \text{Eq 2-100}$$

As mentioned before, the residual vector of a cell is only dependent on the cell itself and a small number of neighboring cells. Hence, the flux Jacobian matrix $\partial \boldsymbol{R}/\partial \boldsymbol{U}$ is very sparse. In order to take advantage of the sparsity and solve the system efficiently, a very instrumental and efficient parallel sparse direct solver MUltifrontal Massively Parallel Solver (*MUMPS*) (Amestoy, Duff, L'excellent, & Koster, 2001), (Amestoy, Buttari, L'excellent, & Mary, 2019) is utilized under the Portable Extensible Toolkit for Scientific computation (*PETSc*) environment (Balay, Gropp, McInnes, & Smith, 1997), (Balay, et al., 2019), (Balay, et al., 2020).

In order to solve the adjoint system and compute the sensitivity derivatives $dI/d\boldsymbol{X}$ by performing the operations given in Eq 2-100, a stand-alone program is written. It reads Jacobian matrices and partial derivatives of the objective function already evaluated. After reading matrices and vectors, the program stores them using *PETSc* parallel matrix and vector data structures as sparse entities. Once the matrices and vectors are filled and distributed corresponding memories, the adjoint system is solved using a multifrontal lower-upper (LU) factorization provided by *MUMPS*. Moreover, all internal processor communications managed by Message Passing Interface (MPI) libraries. Thus, the adjoint solver can be run parallel on distributed-memory systems, as well as shared-memory systems.

In the present study, due to its robustness and accuracy, a direct solver is employed to solve the adjoint system. However, the huge memory requirement due to fill-in during LU decomposition may preclude the use of a direct solver. Therefore, relatively large problems (such as matrices having rank larger than 10 million) may be solved using an iterative solver, e.g., GMRES (Saad & Schultz, 1986), with a proper preconditioner. Nevertheless, iterative solvers suffer robustness problems, especially in the presence of turbulence equations that significantly increases the stiffness of the matrix. There are some possible treatments to alleviate this problem, like using an approximate Jacobian matrix instead of the exact flow Jacobian matrix. A common treatment is to employ the frozen turbulence assumption. Because tightly coupling turbulence equation with the mean flow equations considerably increases the stiffness of the matrix. Another approach is to add a time-like derivative in order to increase the diagonal dominance of the system (Nielsen & Anderson, 2000). In this way, the system is solved by marching in time, much like the flow solver. The approach may be stated as given below,

$$\frac{\Omega}{\Delta t}\boldsymbol{I} + \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{\boldsymbol{T}} \Delta^n \boldsymbol{\lambda} = -\frac{\partial I}{\partial \boldsymbol{U}} - \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}\right]^{\boldsymbol{T}} \boldsymbol{\lambda}^n \qquad \text{Eq 2-101}$$

where

$$\boldsymbol{\lambda}^{n+1} = \boldsymbol{\lambda}^n + \Delta^n \boldsymbol{\lambda} \qquad \text{Eq 2-102}$$

The usage of this approach results in a more robust adjoint solver. In the present study, the main focus is on the implementation of adjoint methods for a RANS flow solver coupled with a turbulence model and a laminar to turbulent transition model. Therefore, the implementation of a robust iterative solver is left as a further study.

2.3 Optimization Framework

In the present study, the gradient-based optimization process is driven by the *DAKOTA* software environment (Adams, et al., 2014, updated 2019). *DAKOTA* is an open-source software developed by Sandia National Laboratories. It provides various gradient-based and gradient-free optimization algorithms. It also provides widely used algorithms for uncertainty quantification, parameter estimation, and sensitivity/variance analysis with design of experiment methods as well as parameter study methods. Due to its wide range of applicability and popular robust algorithms, *DAKOTA* is chosen as the main driver of the optimization study.

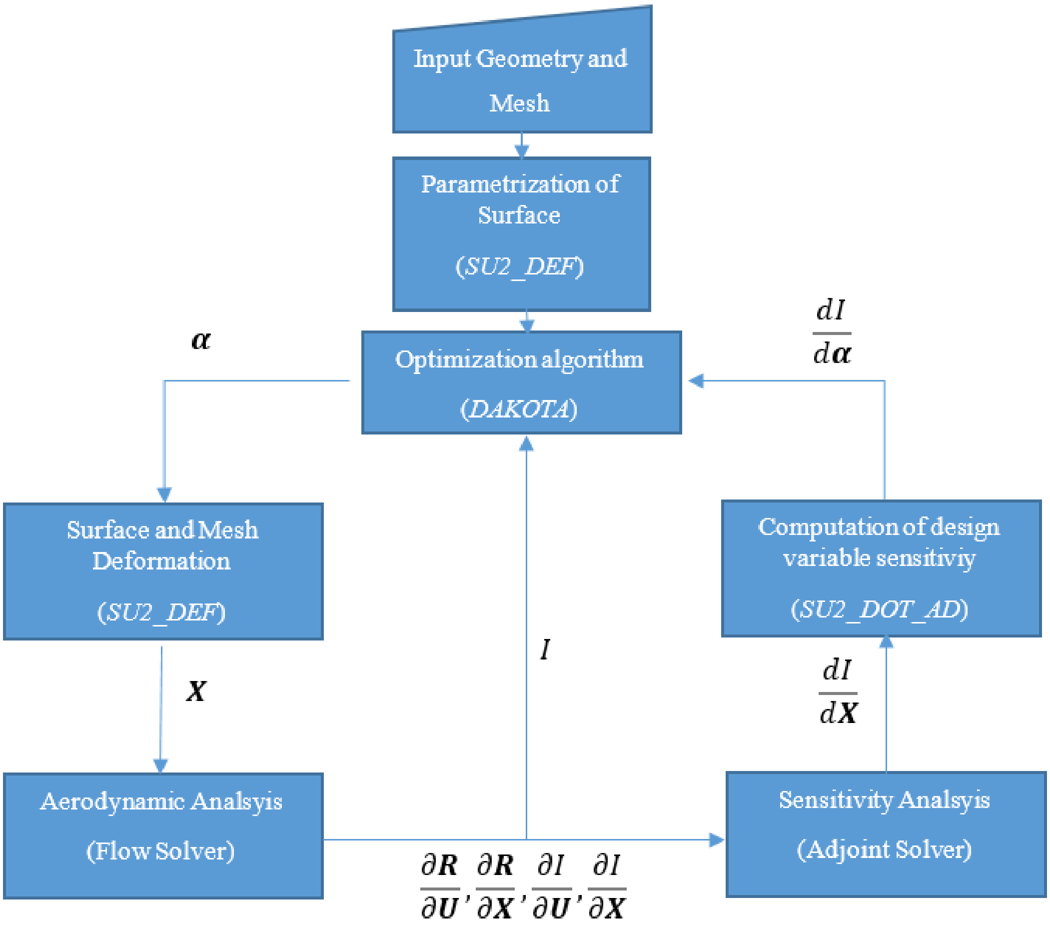


Figure 2-7 Optimization flow chart

The flow chart of the optimization process is depicted in Figure 2-7. As shown in the figure, in the optimization study, the baseline configuration and the corresponding grid are provided. The aerodynamic surface to be optimized is parameterized by using a Free-Form Deformation (FFD) technique. The optimization algorithm then generates a new set of design variables $\boldsymbol{\alpha}$ that are the node coordinates of the FFD box. The volume grid is then deformed through an approach based on the linear elasticity equation. The flow solver computes the value of the objective function I and the partial derivatives required by the adjoint solver. The adjoint solver then calculates the grid sensitivities of the objective function, $dI/d\boldsymbol{X}$. Finally, the sensitivities to the design variables, $dI/d\boldsymbol{\alpha}$, are evaluated by the dot product given in Eq 2-103.

$$\frac{dI}{d\boldsymbol{\alpha}} = \frac{dI}{d\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{\alpha}} \qquad \text{Eq 2-103}$$

The search for optimum design variables in the design space is then driven by *DAKOTA*, which uses the sensitivity derivatives and updates design variables. The optimization process continues until predefined convergence criteria are reached. The default stopping criteria are achieving a value of the $L_2\,norm$ of the gradient less than 0.0001 and value of normalized relative change less than 0.001.

2.3.1 Surface Parametrization and Grid Deformation

In the optimization process described in the previous subchapter, the initial surface geometry provided is parameterized by an FFD box, which provides the design variables to the optimization algorithm. Surface parameterization by an FFD box is achieved by an open-source *SU2* module, *SU2_DEF* (Economon, Palacios, Copeland, Lukaczyk, & Alonso, 2016). *SU2_DEF* includes several geometry parametrization techniques, including FFD. The FFD technique implemented in *SU2_DEF* is the one proposed by Samareh (2004). In general, FFD techniques employ a trivariate representation to parameterize a volume (Sederberg & Parry, 1986), (Coquillart, 1990), (Lamousin & Waggenspack, 1994). For the representation of a surface, Samareh (2004) proposes a technique that compresses a trivariate volume deformation approach to bivariate surface deformation. The number of design variables that define a surface is then reduced by order of magnitude, but it still provides the needed flexibility in surface representation. An example of the surface deformation performed by the FFD technique is illustrated in Figure 2-8.

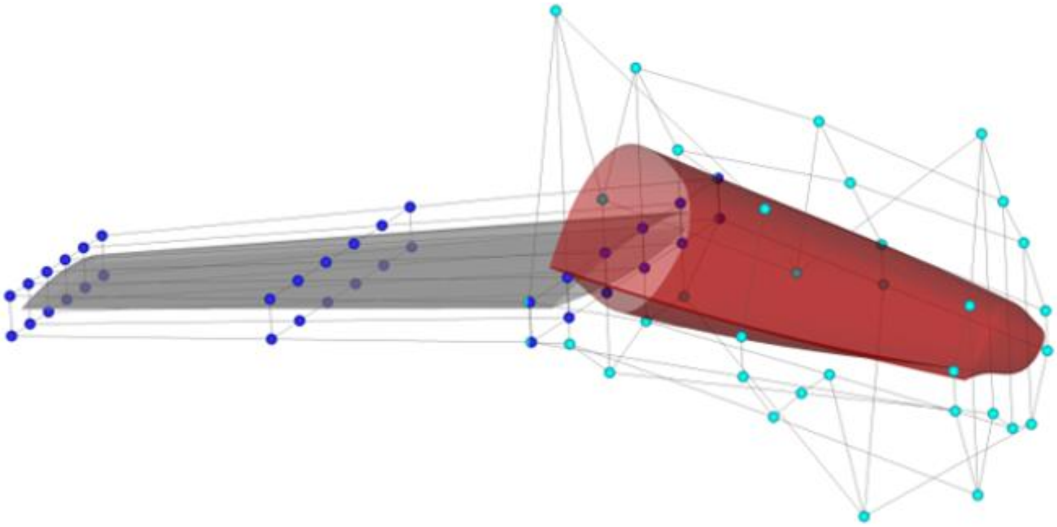


Figure 2-8 A sample deformation carried out by SU2_DEF employing FFD approach

The FFD approach utilized in *SU2_DEF* only deforms the surface nodes. The deformation of the volume grid is also accomplished by *SU2_DEF*. In order to deform the volume grid, *SU2_DEF* makes use of elastic deformations (Dwight, 2006). It sets the system of the linear elasticity equations and solves them using a preconditioned GMRES method (Saad & Schultz, 1986). In order to reduce the boundary cell deformations and to keep the wall distances about constant, the stiffness coefficients are made inversely proportional to the wall distance. The use of inverse wall distance as a stiffness coefficient effectively keeps the boundary layer grid intact and causes deformations to be absorbed in the volume grid.

2.3.2 Grid Sensitivity

The adjoint solver developed computes the sensitivities of the objective function to the grid coordinates, $dI/d\boldsymbol{X}$. The sensitivity of the objective function to the design variables, $dI/d\boldsymbol{\alpha}$, are obtained by $d\boldsymbol{X}/d\boldsymbol{\alpha}$ projecting the grid sensitivity of the objective function onto the design variables $dI/d\boldsymbol{\alpha}$. The projection of the grid sensitivity of the objective function onto the design variables is performed through the inner product given in Eq 2-103. In the present study, the inner product operation is similarly performed by using the *SU2_DOT_AD* module of the *SU2* software suit. The module is obtained by differentiating the FFD and grid deformation functions of *SU2_DEF* with respect to the design variables $\boldsymbol{\alpha}$ by using the AD tool *CoDiPack* (Sagebaum, Albring, & Gauger, 2019). *SU2_DOT_AD* evaluates $d\boldsymbol{X}/d\boldsymbol{\alpha}$ by the chain rule, and performs the inner product operation defined in Eq 2-103.

The adjoint solver developed computes the sensitivities of the objective function to grid coordinates $dI/d\boldsymbol{X}$. The sensitivity of the objective function to the design variables $dI/d\boldsymbol{\alpha}$ are obtained by $d\boldsymbol{X}/d\boldsymbol{\alpha}$ projecting the grid sensitivity of the objective function onto the design variables $dI/d\boldsymbol{\alpha}$. The projection of the grid sensitivity of the objective function onto the design variable sensitivity is performed through the inner product given in. In the present implementation, the computation of the grid sensitivity to design variables $d\boldsymbol{X}/d\boldsymbol{\alpha}$ and the inner product operation is performed by using *SU2_DOT_AD*.

2.3.3 Optimization Algorithm

In the present study, among the optimization algorithms available in *DAKOTA*, the quasi-Newton method with a Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation to the Hessians (Broyden, 1969), (Fletcher, 1970), (Goldfarb, 1970), (Shanno, 1980) is employed. In the present study, in order to perform the quasi-Newton algorithm, the implementation of the *OPT++* library (Meza, Olivia, Hough, & Williams, 2007) is utilized in the *DAKOTA* framework (referred to *DAKOTA*'s optpp_q_newton method). Furthermore, in the study, the optimization problems are either unconstrained problems or converted into an unconstrained problem by employing a penalty function.

CHAPTER 3

RESULTS AND DISCUSSION

In this chapter, the RANS solver developed for natural laminar flows are first validated. The validation of the flow solver is performed for both two- and three-dimensional, compressible, turbulent, and natural laminar flows over airfoils and wings. The validation of the flow solver is succeeded by the validation of the adjoint solver. In the validation studies, the sensitivities of an objective function with respect to both grid coordinates and design variables are compared against the sensitivities computed by the finite-difference approximation. Finally, full aerodynamic shape optimization studies are performed for the cases considered in the validation studies, including low and high aspect ratio natural laminar flow wings.

3.1 Validation of RANS Solver

In this subchapter, the validation studies regarding the developed RANS solver is presented. In order to validate the flow solver, several two-dimensional and three-dimensional cases are performed. In two-dimensional cases, flow over flat plate cases with different free stream turbulent intensities, transonic flow over an airfoil, and flow over a natural laminar airfoil are considered. In three-dimensional cases, transonic flow over the well-known Onera M6 wing, natural laminar flow over a low, and a high aspect ratio wing are studied.

3.1.1 Flow over a Flat Plate

As the first test case, the flow over a flat plate is considered. The flow has a zero pressure gradient with 0.18% turbulent intensity, and Re per meter is around

3.4×10^6 for 1 m long (Schubauer & Klebanoff, 1956). The test case is also the model calibration test case of the B-C transition model.

In the test case, the computational domain consists of 538×120 quadrilateral cells in the stream-wise and normal directions, respectively. There are 280 points on the flat plate. The size of the first layer is set to 2×10^{-6} unit that ensures $y^+ < 1$.

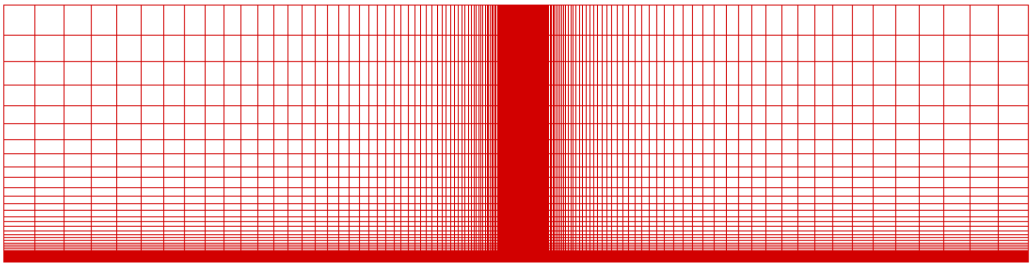


Figure 3-1 Flat plate computational grid

In the analysis, the convective fluxes for the mean flow equations are computed using a second-order Roe scheme. The turbulence is modeled using the S-A model with the B-C transition model.

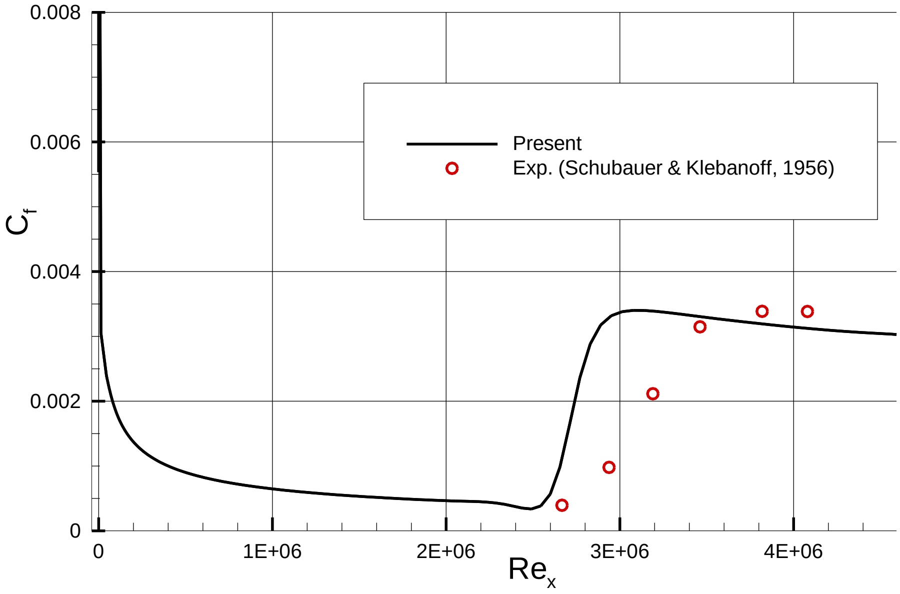


Figure 3-2 Skin friction coefficient (T_{u_∞} 0.18%)

In Figure 3-2, the skin friction coefficient computed by the flow solver is compared to the experimental data. Although the transition length is underestimated, there is a very good agreement on the transition onset location. Similar behavior is observed in the original paper of the B-C transition (Cakmakcioglu, Bas, & Kaynak, 2017). Although the transition model lacks a transition length correlation, the reference study (Cakmakcioglu, Bas, & Kaynak, 2017) concludes that the effect of such a correlation is minimal based on an additional comparison made with the two-equation $\gamma - Re_\theta$ model of Menter et al. (2004) that includes a transition length correlation.

Next, another simulation, which is based on the experimental study of Savill (1993) on flow over a flat plate with a turbulence intensity of 3.0%, is considered in order to validate the effect of turbulence intensity and Reynolds number on the transition onset. The Reynolds number of the flow per meter is 360×10^3. In the analysis, a grid similar to the one in the previous case is employed. The size of the first grid layer is now set to 2×10^{-5}. Again, the convective fluxes for the mean flow equations are computed using the second-order Roe scheme.

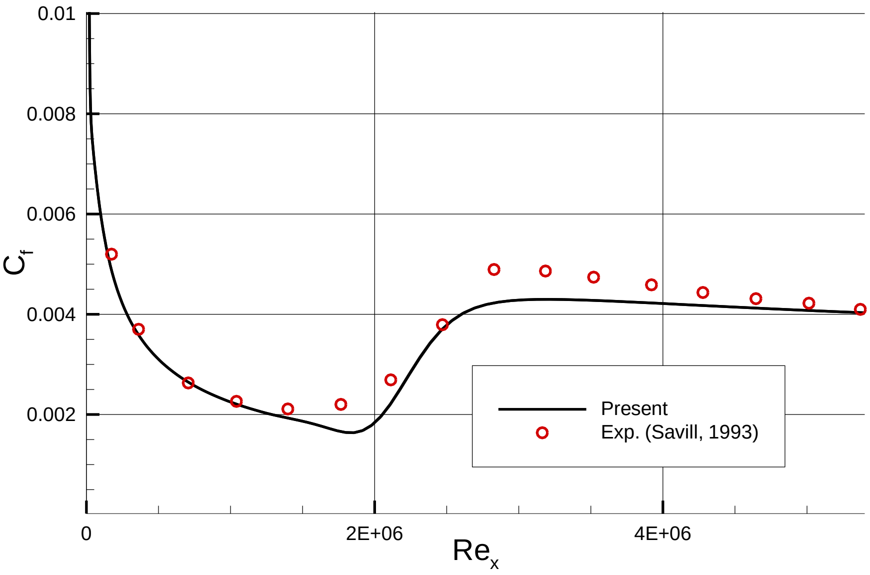


Figure 3-3 Skin friction coefficient (T_{u_∞} 3.0%)

As observed in Figure 3-3, the variation of the skin friction coefficient predicted agrees well with the experimental data. A similar prediction is also made in the reference study (Cakmakcioglu, Bas, & Kaynak, 2017). Based on these validation studies, it is concluded that the B-C transition model is successfully implemented in the flow solver.

3.1.2 Flow over RAE 2822 Airfoil

Transonic flow over RAE 2822 airfoil is a well-known test case that is commonly considered to validate flux schemes and turbulence models. In the current study, the 6th and 9th test cases of the AGARD report (Cook, McDonald, & Firmin, 1979) are considered. The flow conditions of test case 6 and test case 9, which are corrected by Slater, Dudek, & Tatum (2000) and by DLR (Rudnik, 1997) by accounting for wind tunnel influences respectively, are used. The corrected flow conditions are given in Table 3.1. It should be noted that in the experiments, a transition trip is applied near to the leading edge at 0.03 chord to fix laminar to turbulent boundary layer transition.

Table 3.1 The corrected flow conditions for test case 6 and test case 9

	Case 6	Case 9
M_∞	0.729	0.730
$\propto$	2.31°	2.80°
Re_c	6.5×10^{6}	6.5×10^{6}

A C-type computational grid that has 439×120 quadrilateral cells in the stream-wise and normal directions, respectively, is used in the numerical simulations (Figure 3-4). The size of the first grid layer is 1×10^{-6} unit that ensures $y^{+} < 1$, and there are 240 points over the airfoil surface.

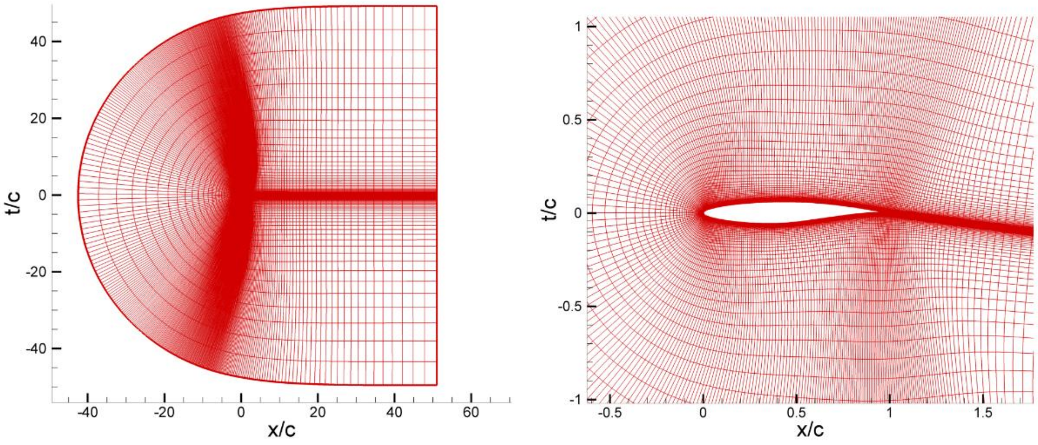


Figure 3-4 Computational grid for RAE 2822 airfoil

In the numerical simulations, the convective fluxes are again computed using the second-order Roe scheme. The gradients of the flow variables required by the reconstruction of the flow variables and the evaluation of the viscous fluxes are computed using the Green-Gauss approach. The Spalart-Allmaras turbulence model is similarly employed. However, the transition model is not employed, since a transition trip is placed near the leading edge in the experimental study.

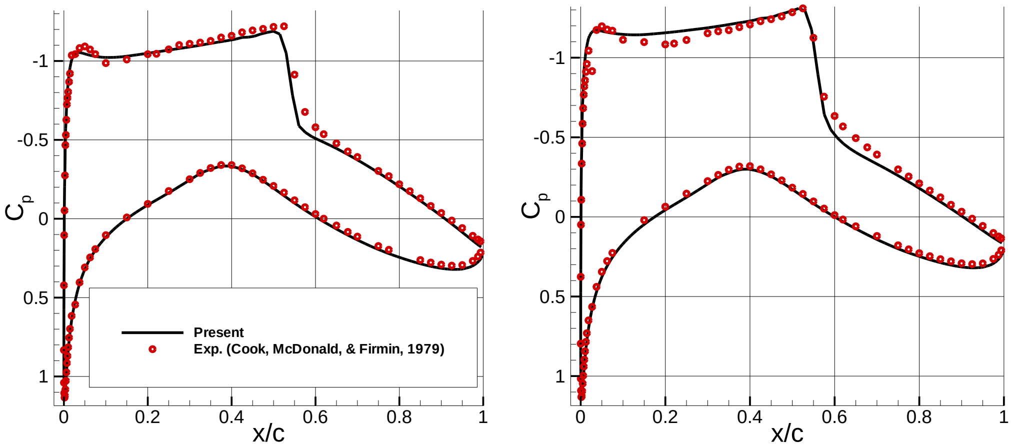


Figure 3-5 Pressure distributions for case 6 (left), and case 9 (right)

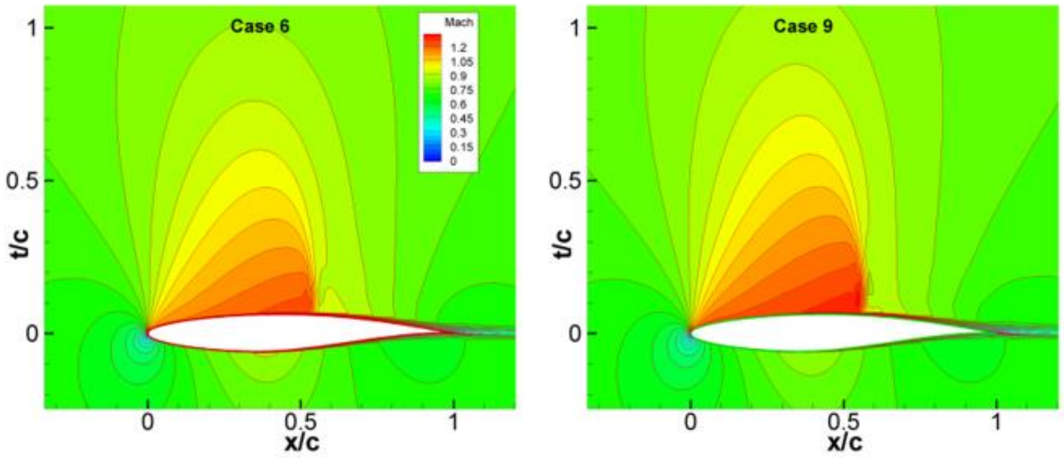


Figure 3-6 Mach contours

The pressure distributions predicted for both cases are compared to the experimental data in Figure 3-5. As observed, although the shock location for case 6 is underpredicted by about 3% chord, the predictions are, in general, in a very good agreement with the experimental data. Furthermore, the Mach contours for both cases are shown in Figure 3-6.

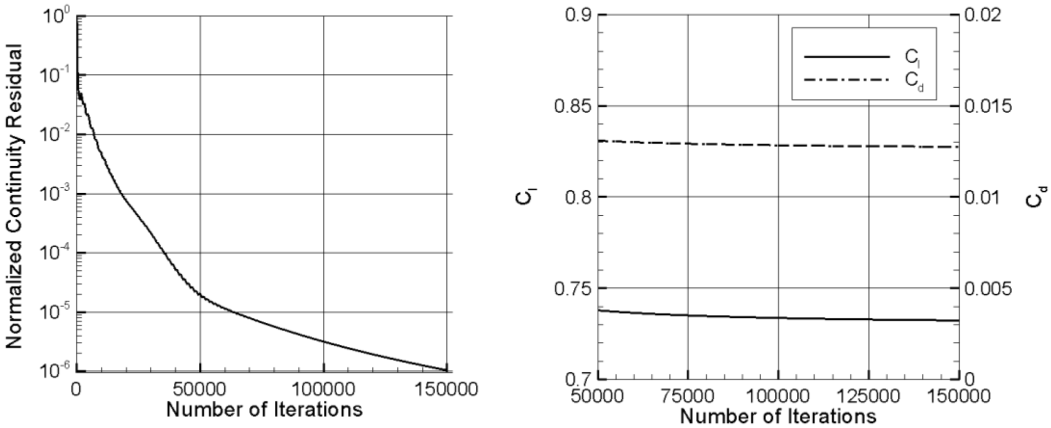


Figure 3-7 Convergence history for case 6

In the analyses regarding NLF(1)-0416 airfoil, it is aimed to decrease the continuity residuals more than six orders of magnitude for each angle of attack, and 300,000 iterations are found enough to decrease the continuity residuals more than six orders of magnitude (Eq 3-7). Furthermore, it is enough to get convergence in the lift and the drag coefficient. When all analyses regarding NLF(1)-0146 airfoil is considered, the changes in the lift and the drag coefficient are less than 0.0025 and 2 drag count in the last 50,000 iterations, respectively.

3.1.3 Flow over Natural Laminar Airfoil

The accuracy and the capability of the flow solver developed are next demonstrated by analyzing and then comparing the numerical and the experimental predictions for a natural laminar flow over the NLF(1)-0416 airfoil, for which there is an experimental study performed by Somers (1981). In the present study, flow over the NLF(1)-0416 airfoil at a Mach number of 0.1 and a Reynolds number of 2×10^6 is considered.

In the analysis, a C-type computational grid with 439×120 quadrilateral cells is employed (Figure 3-8) in the stream-wise and normal direction, respectively. The size of the first grid layer of 5×10^{-6} unit ensures $y^+ < 1$. There are similarly 240 grid nodes on the airfoil surface, and the farfield boundary is placed about 50 chord lengths away.

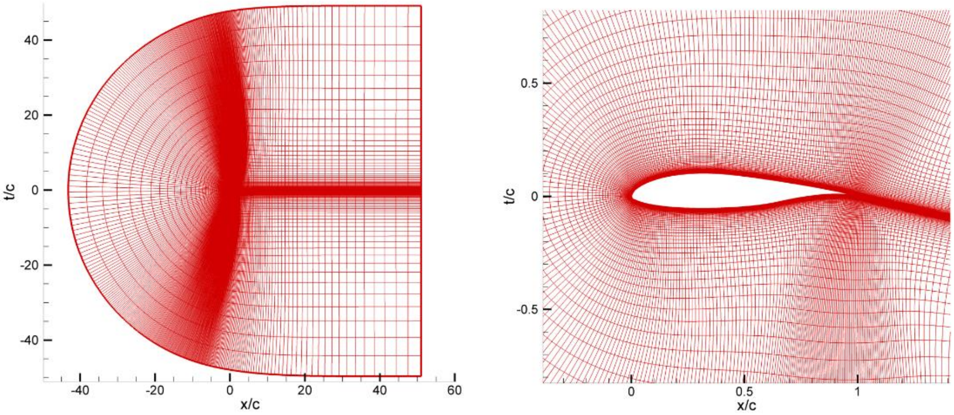


Figure 3-8 Computational grid for NLF(1)-0416 airfoil

In order to validate the results of the flow solver, the lift curve, the drag polar of the airfoil, and the transition onsets at different lift coefficients are compared to the experimental results.

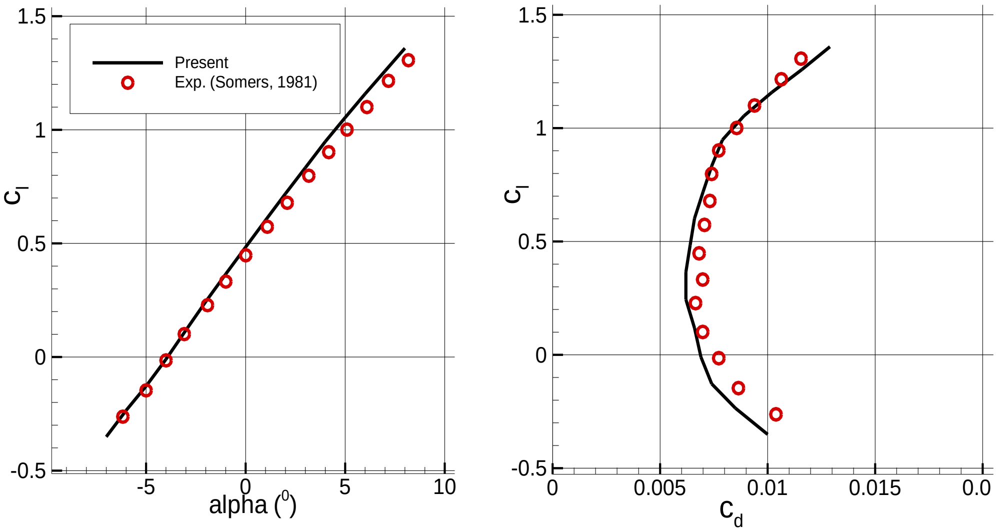


Figure 3-9 Variation of lift (left) and drag coefficients (right)

Figure 3-9 compares the aerodynamic coefficients of experimental and numerical results. It shows that the flow solver predicts slightly higher lift and lower drag coefficient. When the lift coefficient is greater than 1.0, the prediction of the drag coefficient is higher than measured in the experiment. In addition to possible turbulence and transition modeling errors, the reasons for such a discrepancy possibly stem from due to wall corrections or the presence of trailing edge with finite thickness used in the wind tunnel model. Although there is a slight discrepancy between experimental and numerical results, the results are in good agreement and encouraging. The deviation in the drag coefficient is not more than 10 drag counts, and the difference in the lift curve slope is less than 5%.

In the experiment, a microphone was connected to the orifices on the model. This microphone was utilized to locate the location of the boundary layer transition laminar to turbulent flow. The transition location is determined by measuring the noise level. In the laminar part, it is virtually silent. So the transition is detected as an increase in noise level. In the following figure (Figure 3-10), open symbols indicate the location of the last orifice on which flow is laminar, and closed symbols indicate the location of the first orifice on which flow is turbulent.

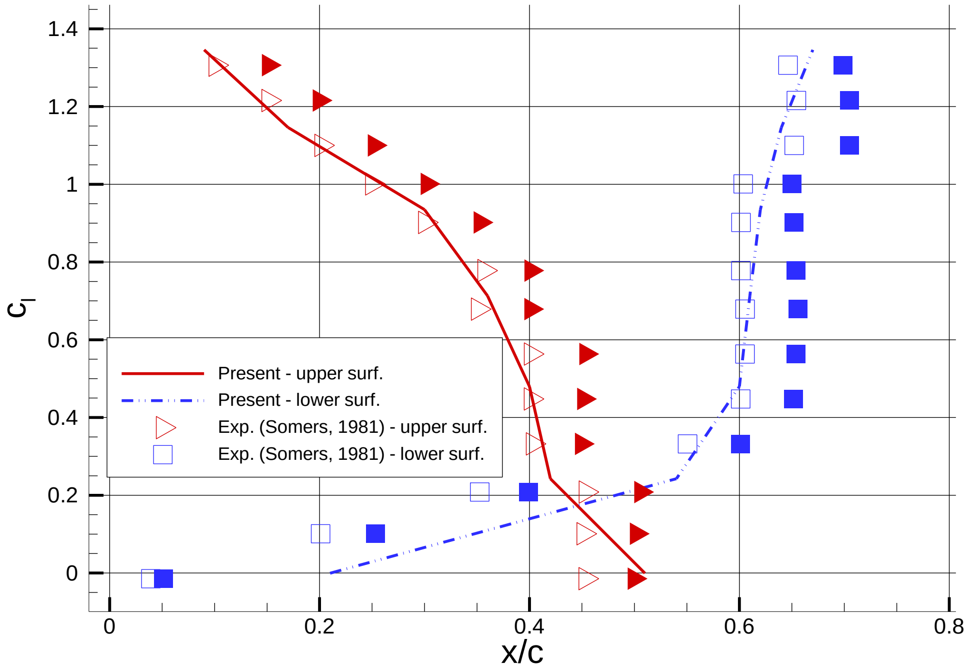


Figure 3-10 Transition locations of the boundary layer from laminar to turbulent flow; open symbols indicate orifices on which flow is laminar, and closed symbols indicate orfices on which flow is turbulent

Figure 3-10 depicts the comparison of the transition onsets between the locations predicted by the B-C transition model and measured by the experiment. The locations of the transition onsets predicted by the numerical method are in good agreement with the experimental results. As the angle of attack increases, transition onset on the upper surface moves steadily forward, and eventually at angles of attack above 0° lower transition onset sticks at around 60% chord. The change in the transition onset observed in the experiment is similar.

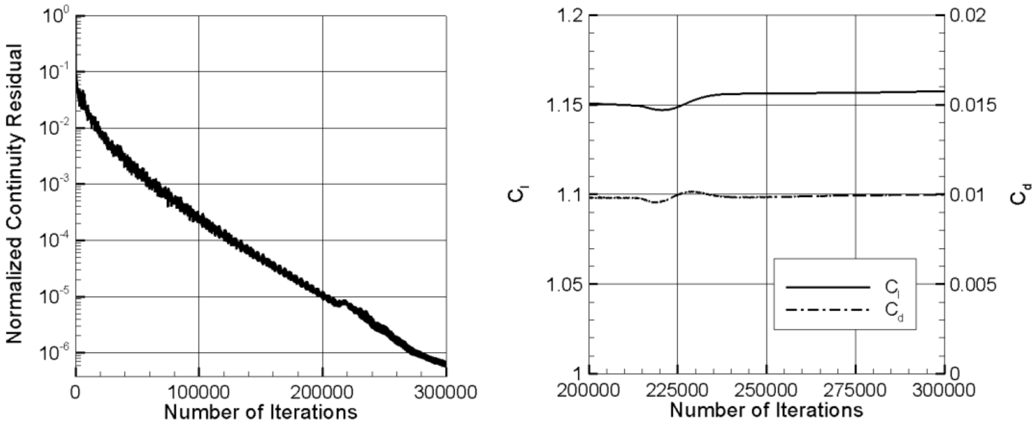


Figure 3-11 Convergence history for NLF(1)-0416 airfoil at the angle of attack 6°

When the flow velocity is small compared to the acoustic speed, the convective terms in the time-dependent equations become stiff, and the convergence rate decreases (Choi & Merkle, 1993). Thus, in order to achieve a satisfactory convergence, the number of iterations performed is increased to 300,000. As seen from Figure 3-11, by the iterations, the continuity residuals are reduced more than six orders of magnitude. Moreover, it is also found enough to get convergence in the lift and the drag coefficient. When all analyses regarding NLF(1)-0146 airfoil is considered, the changes in the lift and the drag coefficient are less than 0.0025 and 2 drag count in the last 50,000 iterations, respectively.

3.1.4 Flow over Onera M6 Wing

The transonic flows over the Onera M6 wing, which are studied experimentally by (Schmitt & Charpin, 1979), are almost the standard validation cases for external flows. Although the wing has a simple geometry, it has complex transonic flow characteristics (i.e., local supersonic flow, lambda shock). In some cases, there may a shock emerging from the leading edge as well as a normal shock further aft. In this section, the transonic flows are considered for further validation of the flow solver.

The three-dimensional, computational grid (Figure 3-12) includes around 200×10^3 hexahedral elements. The size of the first layer is 1×10^{-6} unit that ensures $y^+ < 1$.

There are around 4×10^3 quadrilateral elements that constitute the wing surface and the farfield boundary is placed around 20 mean aerodynamic chords long away.

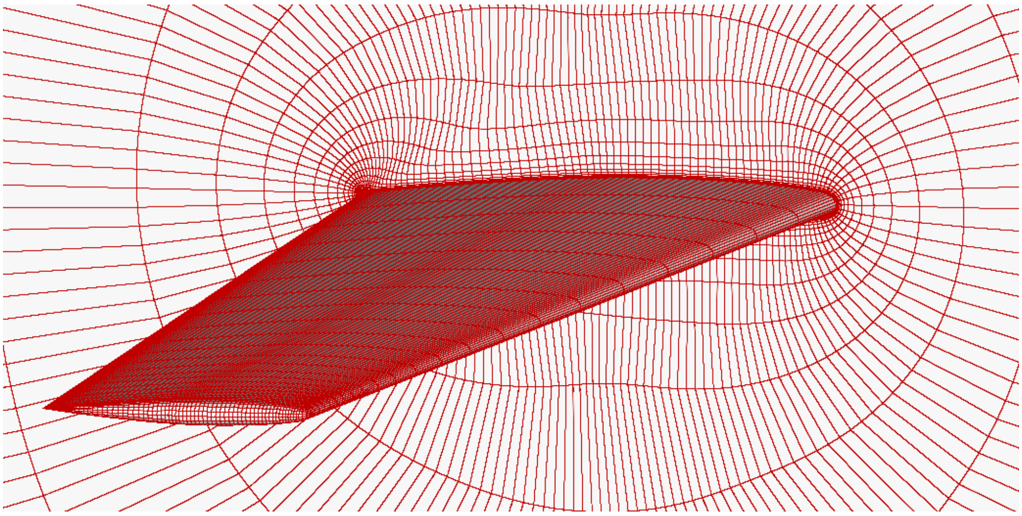


Figure 3-12 Computational grid for Onera M6 wing

In the test case, the free-stream Mach number is 0.8395, and the angle of attack is 3.06°. The Reynolds number based on the mean aerodynamic chord is 11.72×10^6. In the analysis, for the convective flux computation, a second-order Roe flux-difference splitting scheme is employed. In order to achieve second-order accuracy in space, conservative variables are reconstructed using the Green-Gauss approach. Turbulence effects are considered by the Spalart-Allmaras turbulence model.

The comparisons between the numerical and experimental results in terms of pressure coefficients are carried out at six different locations corresponded to η = 0.2, 0.44, 0.65, 0.8, 0.9, 0.95.

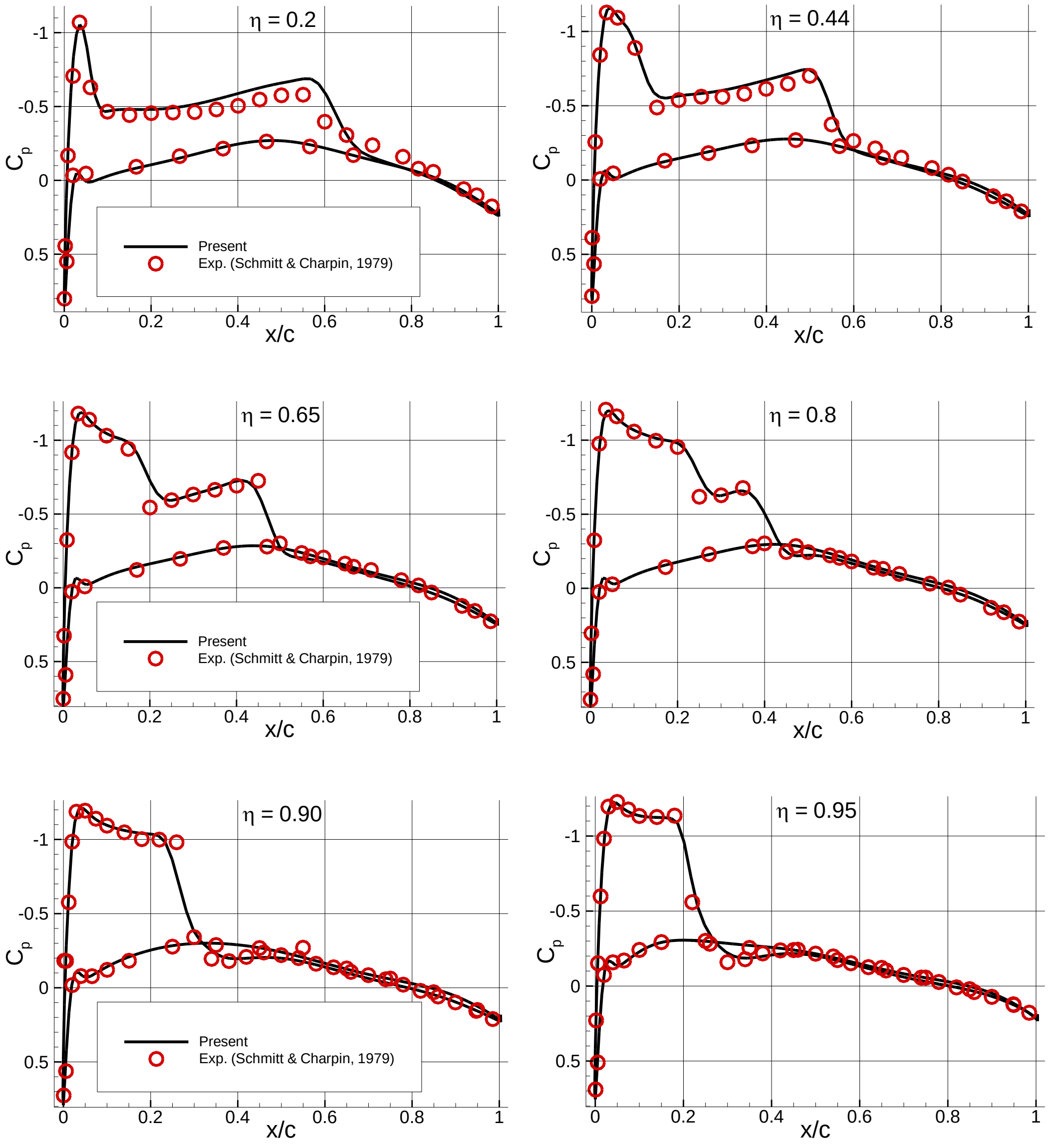


Figure 3-13 Pressure coefficients at different wingspan stations

Figure 3-13 depicts that the numerical method estimated the presence of lambda shock and the locations of the shock emerging from the leading edge as well as the normal shock further aft. As seen from the figure, the lambda shock spans up to 80% percent of the half-span. The figure also illustrates that it merges to a single shock at 90% of the half-span. This pattern agrees very well with the experimental results, and that is also illustrated through Mach contours in Figure 3-14.

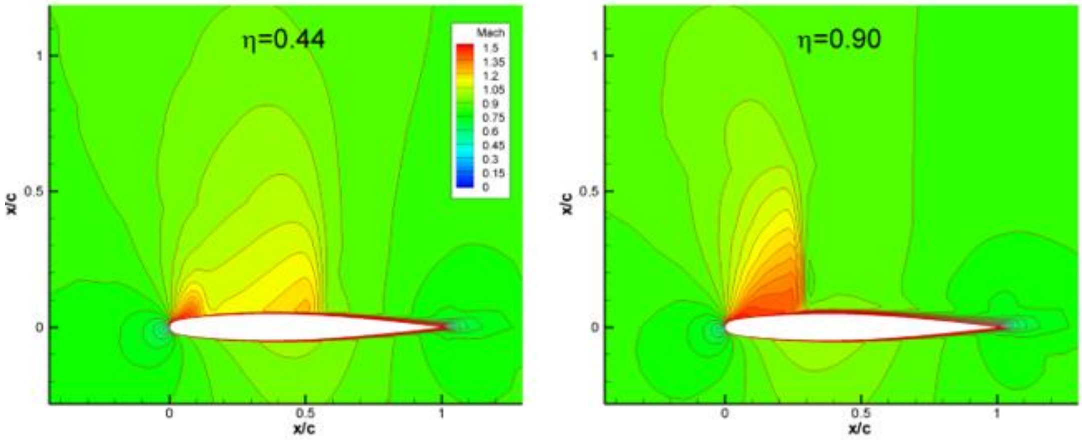


Figure 3-14 Mach contours

Due to the absence of the experimental values, the lift and drag coefficients are compared with the numerical studies of Kim & Rho (1998) and Radespiel et al. (1990) in Table 3.2. According to the results, there is a good agreement between the numerical studies. The maximum deviations in both the lift and drag coefficient are less than 1.5%.

Table 3.2 Comparison of aerodynamic coefficients of ONERA M6 wing

	C_L	C_D
Present result	0.2663	0.01819
Kim & Rho (1998)	0.2695	0.01817
Radespiel et al. (1990)	0.2677	0.01782

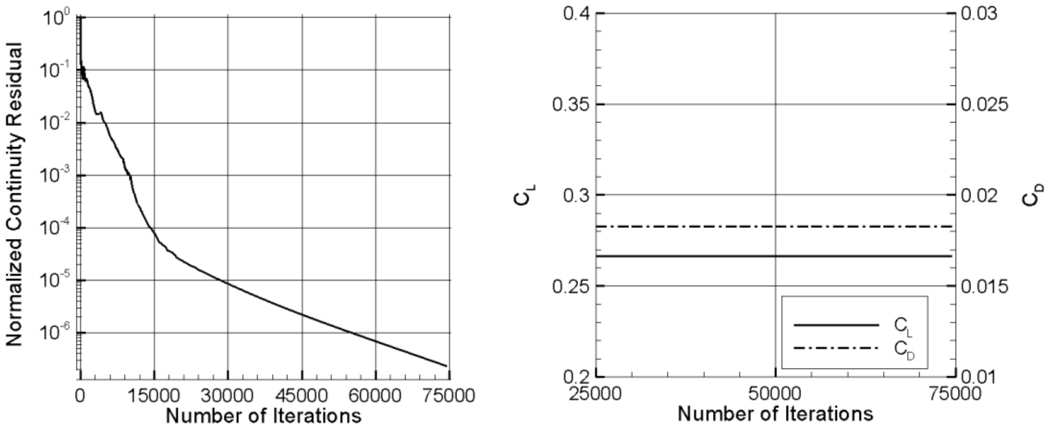


Figure 3-15 Convergence history for ONERA M6 analysis

In the analysis, around 75,000 iterations are performed, which results in more than six orders of magnitude decrease in continuity residual (Figure 3-15). Moreover, a satisfactory convergence in the lift and drag coefficient is achieved. The changes in the lift and the drag coefficient are less than 0.001 and 1 drag count in the last 50,000 iterations, respectively.

3.1.5 Flow over Low Aspect Ratio Laminar Wing

In this validation case, a low aspect ratio wing is considered. The wing model is based on the dimensions of a full-scale wingtip structure that is studied within the framework of the international CRIAQ MDO505 Morphing Wing Project both experimentally and numerically. In the experimental studies, the transition region is captured through infra-red (IR) thermography camera visualizations. More detailed information regarding the project and the wind tunnel test of the wing can be found in the references (Kammegne, Botez, Manou, & Mebarki, 2016), (Koreanschi, et al., 2017).

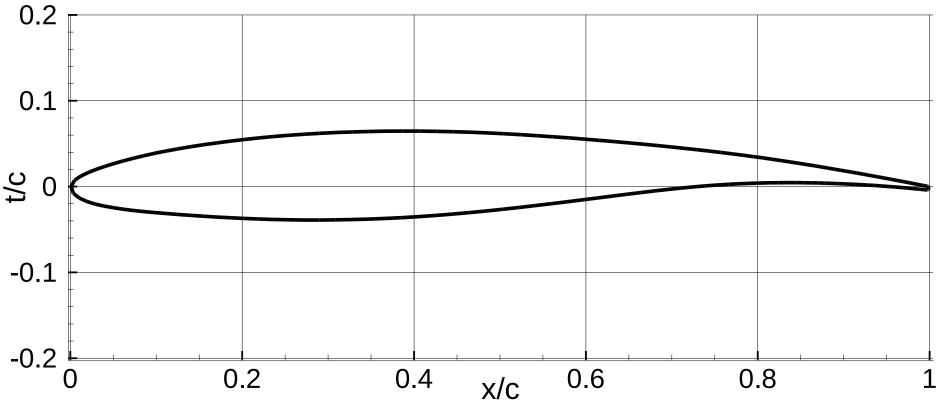


Figure 3-16 The wing section of the MDO505 Morphing Wing

The geometric properties of the wing are given in Table 3.3, and the baseline wing section is illustrated in Figure 3-16. Since the exact wingtip model is not provided, a rounded wingtip is utilized in the computations.

Table 3.3 Geometrical properties of MDO505 wing

Half span length	1.5 m
Length of root chord	1.5 m
Taper ratio	0.72
Aspect ratio	2.325
Leading edge sweep angle	8°
Trailing edge sweep angle	8°

The grid (Figure 3-17) generated for this case consists of around 210×10^3 hexahedral cells. The surface grid on the wing contains around 5×10^3 quadrilateral elements. The height of the first layer is 5×10^{-6} unit that ensures $y^+ < 1$, and the farfield boundary is placed around 50 mean aerodynamic chords length away.

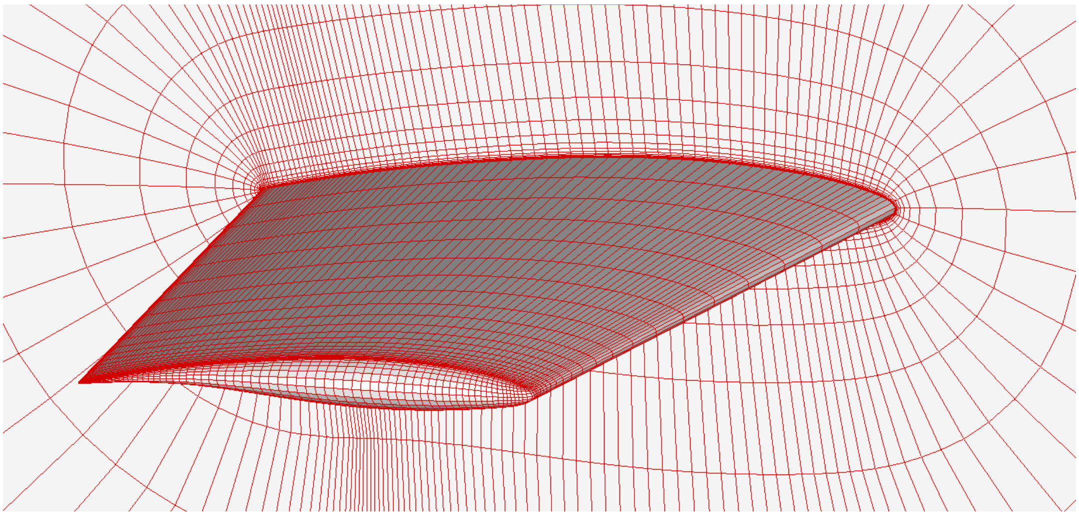


Figure 3-17 Computational grid for MDO 505 wing

The first three cases in the reference (Koreanschi, et al., 2017) are considered in the study. The flow conditions of the cases are given in Table 3.4.

Table 3.4 The flow conditions of the cases

Case No.	1	2	3
M_∞	0.15	0.15	0.15
$\propto$	0.68°	1.5°	2.1°
Re (per meter)	3.4×10^6	3.4×10^6	3.4×10^6

In the reference study, the IR visualization of the laminar-to-turbulent transition region for case 3 is given. The distribution of the transition location on the upper surface for case 3 is compared against the experimental data in Figure 3-19. The black line in Figure 3-19 denotes the average transition line on the upper surface. On the right-hand side, the predicted skin friction distribution over the wing surface is made transparent and overlaid on top of the experimental data shown on the left for a better comparison. As seen in the figure, the dark contour levels indicate the laminar flow region with low skin friction values, and its downstream boundary marks the transition, which is in good agreement with the experimental observation. Furthermore, the pressure coefficient and the turbulent working variable distribution at the midspan wing section is illustrated in Figure 3-18. The turbulent working variable distribution also indicates the transition onset.

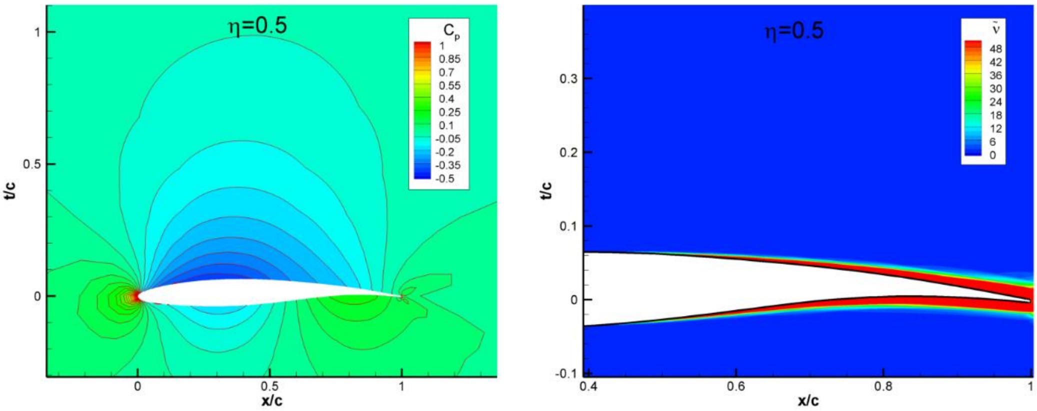


Figure 3-18 Pressure coefficient and turbulent working variable distribution at the midspan wing section (case 3)

It is also observed that, in the experiments, the transition onset close to the wingtip moves abruptly to the leading edge. However, such behavior is not observed in the numerical predictions, which may be attributed to the geometrical differences between the original and the rounded wingtip employed in the present study.

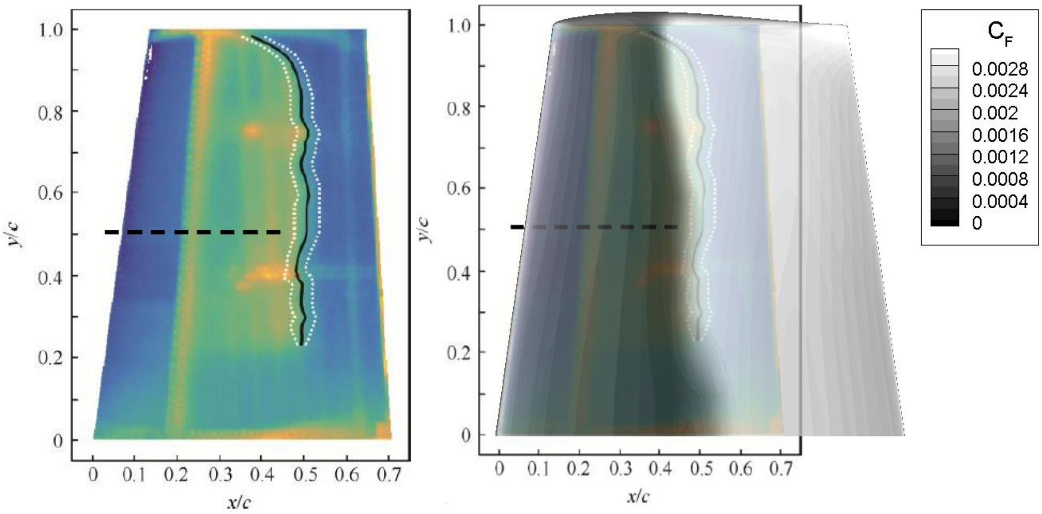


Figure 3-19 Skin friction coefficients. The IR image of the experiment (Koreanschi, et al., 2017) (left) and numerical results illustrated on the IR image (right)

The transition locations at the mid-span of the wing predicted by the solver are compared against the experimental values for all the cases in Table 3.5. The maximum difference between the experimental values and predictions is less than 3% of the mid chord length.

Table 3.5 Comparison of the transition onset at mid-span

Case No.	Numerical transition line (percentage of mid-span chord)	Experimental average transition line (percentage of mid-span chord)	Difference (percentage of mid-span chord)
1	47.7	50.57	2.9
2	46.7	47.91	1.2
3	46.4	48.26	1.9

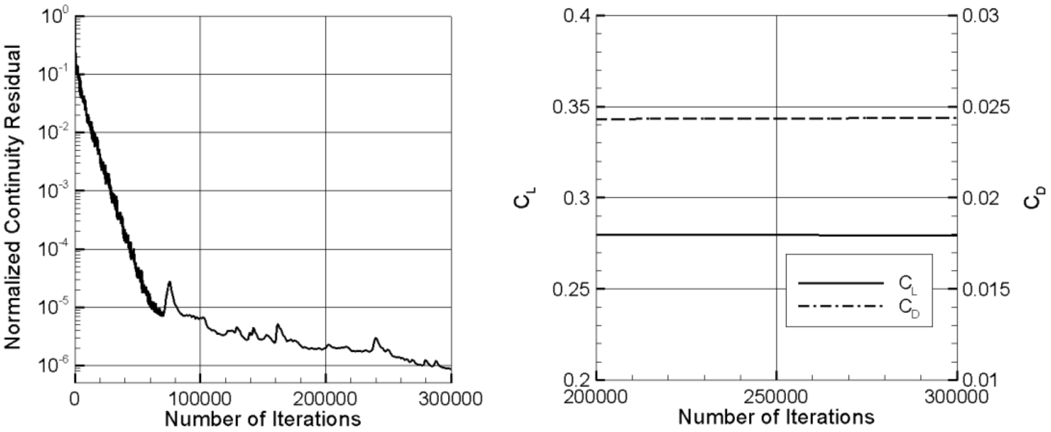


Figure 3-20 Convergence history for case 3

Figure 3-20 shows the convergence history for case 3. Since the Mach number is smaller than the previous case, the number of iterations to get a satisfactory convergence increases. Thus, the number of iterations is increased to 300,000 in the analysis. The number of iterations is the same for the analyses regarding case 1 and case 2. By 300,000 iterations, the continuity residuals are reduced more than six orders of magnitude for all three cases, and the changes in the lift and the drag coefficient are less than 0.001 and 1 drag count in the last 50,000 iterations, respectively.

3.1.6 Flow over High Aspect Ratio Laminar Wing

As the last validation case, a high aspect ratio wing that has a laminar flow section is considered. The wing is an experimental wing evaluated during a UAV development process at Turkish Aerospace Industries (TAI).

The geometrical properties of the wing are given in Table 3.6. It is based on a 16% thick airfoil.

Table 3.6 Geometrical properties of the experimental wing

Span length	16 m
Length of root chord	1.2 m
Taper ratio	0.405
Aspect ratio	18.8
Sweep angle at quarter chord	0°
Twist angle (linearly change)	1.5°

The wind tunnel tests of the experimental wing are performed at the Indonesian Low-Speed Wind Tunnel that is a 4 m × 3 m atmospheric closed-circuit subsonic wind tunnel. At the tests, a 1:6 scaled model is utilized. Flow visualization is performed using the surface oil-flow technique, which enables the observation of flow patterns on the wing surface, such as laminar separation and turbulent re-attachment. It is observed that the laminar-to-turbulent boundary layer transition on the upper surface of a wing at the angle of attack of interest is triggered by laminar separation and is followed by the reattachment of the turbulent boundary layer. The use of surface oil-flow visualization makes it possible to capture the laminar separation bubble and predict the average transition line on the surface.

The grid (Figure 3-21) employed in the case consists of around 240×10^3 hexahedral cells. The surface grid on the wing contains around 5.5×10^3 quadrilateral elements. The size of the first grid layer is 2×10^{-5} unit that ensures $y^+ < 1$, and the farfield boundary is placed around 30 mean aerodynamic chords length away.

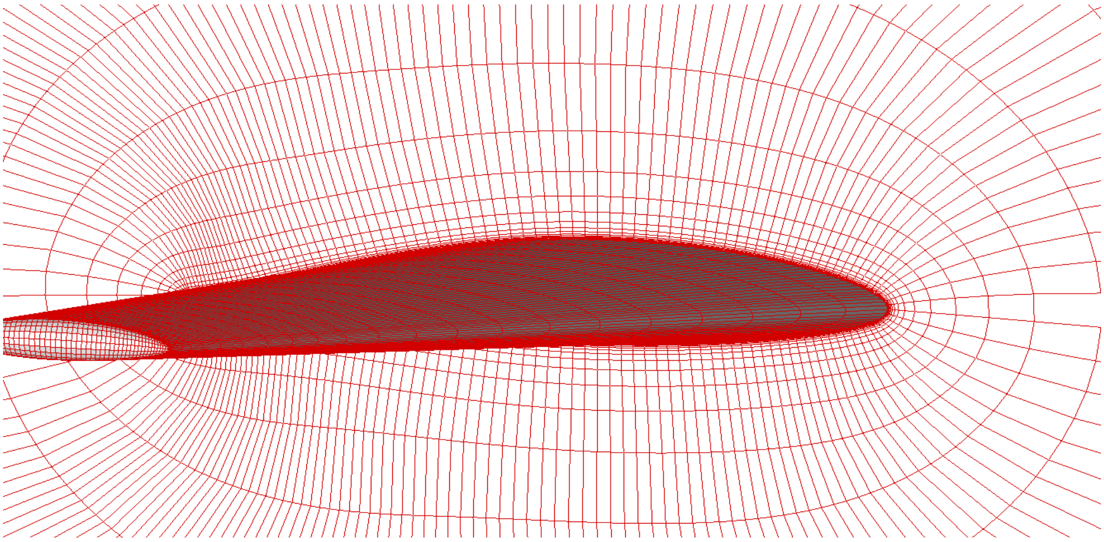


Figure 3-21 Computational grid for experimental wing

The experiments are conducted at a Mach number of 0.2, a Reynolds number of 600×10^3 based on the mean aerodynamic chord, and an angle of attack 6°. Similar to the previous case, the transition location on the upper wing surface is compared against the experimental data.

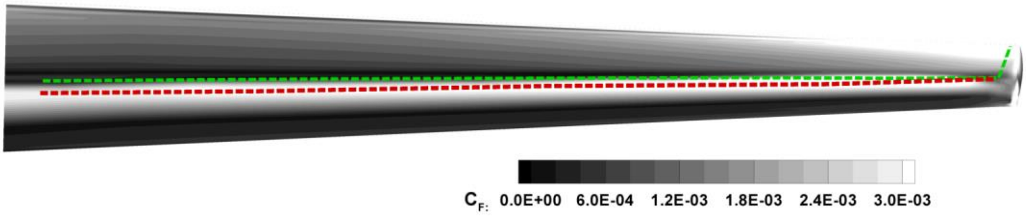


Figure 3-22 Transition locations along the wing span (Permission for the oil-flow image is not granted)

In Figure 3-22, the green dashed line indicates the laminar separation, and the red dashed line indicates the turbulent re-attachment along the span. The transition occurs between the laminar separation line and the turbulent re-attachment line.

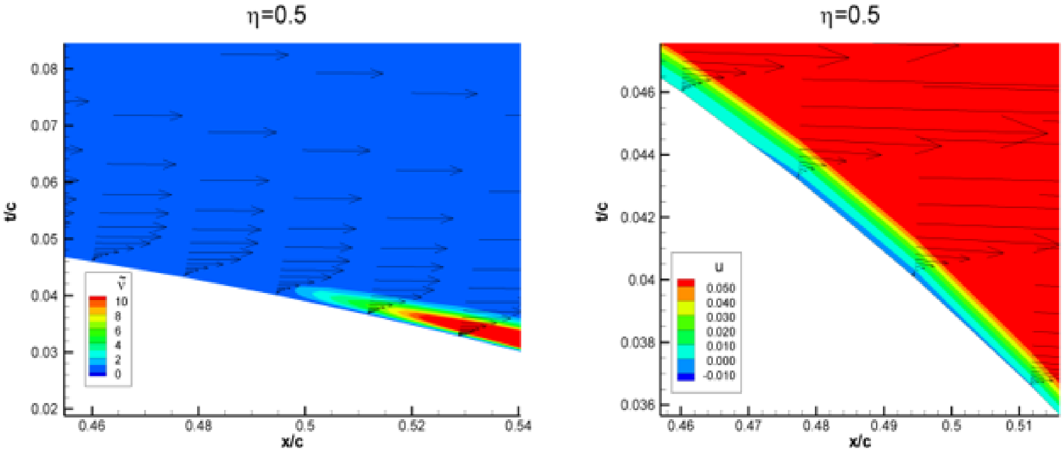


Figure 3-23 The turbulent working variable $\tilde{\nu}$ and x-velocity u distribution at the midspan wing section

Figure 3-23 shows that the transition begins around 50% chord, and the recirculation zone predicted by the flow solver is between 47% and 51% chord at the midspan. It is observed that the transition onset predicted by the present solver equipped with the B-C transition model fairly matches the experiment. Therefore, it is now concluded that the flow solver developed is able to predict the transition onset and the natural laminar flow fairly accurately.

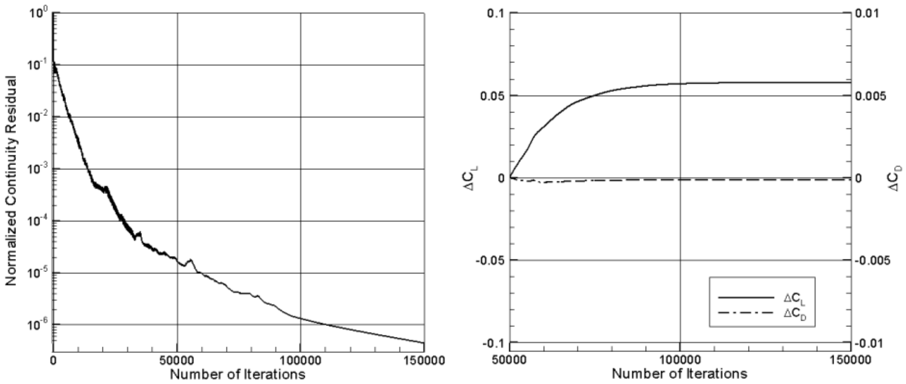


Figure 3-24 Convergence history; normalized continuity residual (left), change in the lift and drag coefficient after the iteration of 50,000 (right)

Finally, in the figure above, the convergence history is illustrated. As seen from the figure, the continuity residual is reduced more than six orders of magnitude in 150,000 iterations, and the change in the lift and the drag coefficient is less than 0.001 and 1 drag count in the last 50,000 iterations, respectively.

3.2 Validation of Adjoint Solver

In this section, the results of the validation studies of the developed adjoint solver are presented. The methodology to validate the adjoint solver is the commonly-used finite difference technique. In the study, the finite central difference approximation is used. The sensitivity derivatives are calculated by the technique, as given in Eq 3-1.

$$\frac{dI}{d\alpha_i} = \frac{I(\boldsymbol{\alpha}_0 + h\boldsymbol{e}_i) - I(\boldsymbol{\alpha}_0 - h\boldsymbol{e}_i)}{2h} + \mathcal{O}(h^2) \qquad \text{Eq 3-1}$$

where $\boldsymbol{e}_i$ is the i^{th} unit vector, and $\mathcal{O}(h^2)$ refers to a truncation error of approximation proportional to h^2. For an accurate approximation, the step size h should be chosen as small as possible to keep the truncation error low, but not so small that finite differences are in the noise. By the numerical experiments, the step size of 0.001 is deemed suitable. Since the method compels to perform two analyses (to compute the values of $I(\boldsymbol{\alpha}_0 + h\boldsymbol{e}_i)$, and $I(\boldsymbol{\alpha}_0 - h\boldsymbol{e}_i)$) for each design variable to get sensitivities with respect to each design variable, two new analyses are conducted for each design variable. The sensitivity derivatives calculated by the finite central difference approximation are compared against the sensitivity derivatives calculated by the adjoint solver. In order to validate the adjoint solver, the cases used in the validation of the flow solver are considered.

3.2.1 Drag Sensitivity at Fixed Lift: RAE 2822

In the first case, the transonic flow over an RAE 2822 airfoil at the flow conditions defined for case 9 studied in the subchapter 3.1.2 is examined.

The airfoil is parameterized by an FFD box with 22 control points, as illustrated in Figure 3-25. Since the control points at the leading and the trailing edges are fixed, the remaining 18 points are considered as the design variables. The design variables are only allowed to move in the y-direction.

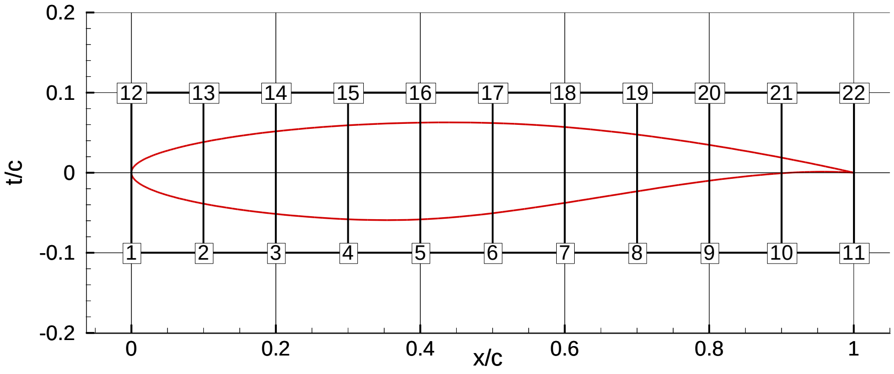


Figure 3-25 FFD box and the control points generated on RAE 2822 airfoil

The objective function is taken as the drag coefficient, and the lift coefficient is constrained. The constraint on the lift coefficient is imposed by a penalty function defined in Eq 3-2. Therefore, a single adjoint analysis is enough to compute the derivatives of the objective function with respect to all the design variables.

$$I = c_d + 2(c_l - c_l^{initial})^2 \qquad \text{Eq 3-2}$$

The gradients computed by the adjoint solver is validated against the gradients computed by the finite central difference approximation, as shown in Figure 3-26. As shown, the adjoint-based sensitivity derivatives are in very good agreement with the finite difference approximation. The maximum deviation is 2.1% of the $L_2\ norm$ of the sensitivity derivatives computed by finite difference approximation.

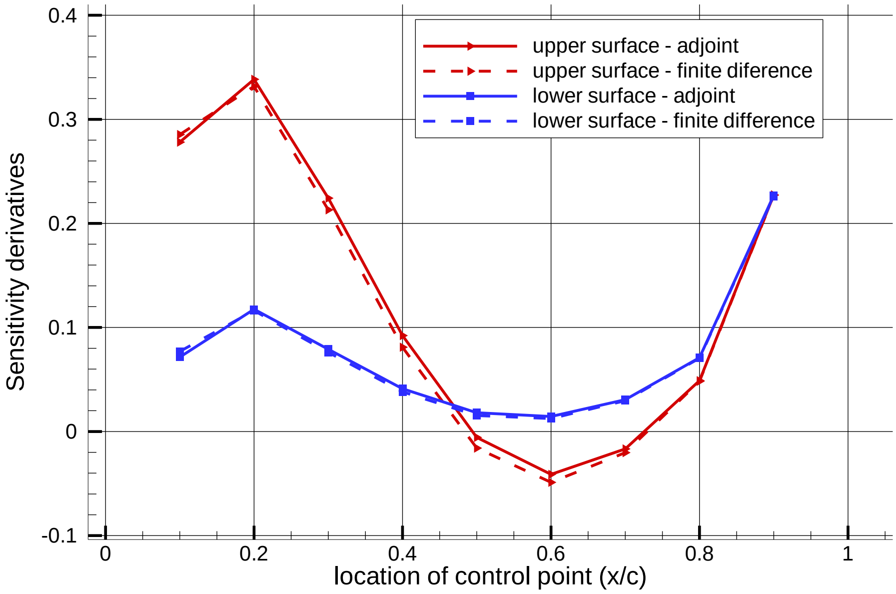


Figure 3-26 Sensitivity derivatives computed by the finite difference method and the adjoint method

3.2.2 Range Parameter Sensitivity: NLF(1)-0416

In the second validation study the flow over NLF(1)-0416 airfoil at $Re\ 2 \times 10^6$, M 0.1 is considered. The angle of attack is set as the angle of attack at which the lift coefficient is given as 1.0. The same computational grid used in the subchapter 3.1.3 is again employed. In this case, it is intended to validate the sensitivity derivatives estimated by the adjoint solver in the presence of the laminar-to-turbulent transition.

A similar parametrization, as is in the previous case, is applied. A similar FFD box employed in the previous case is again utilized, as illustrated in Figure 3-27.

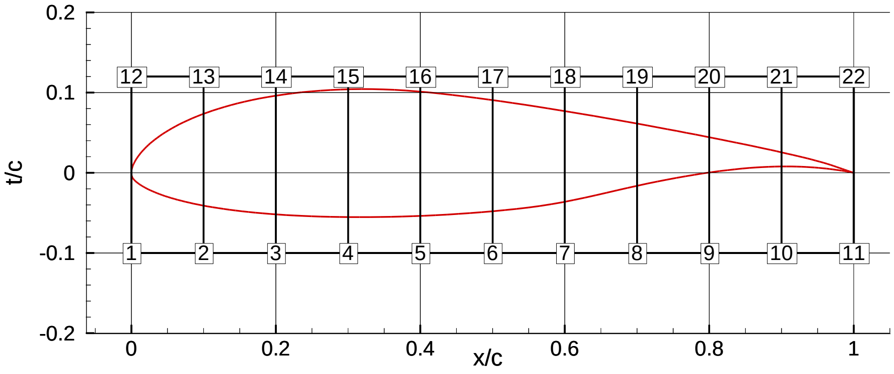


Figure 3-27 FFD box and control points generated on NLF(1)-0416 airfoil

In this case, the objective function is defined as the lift to drag ratio (i.e., range parameter).

$$I = c_l / c_d \qquad \text{Eq 3-3}$$

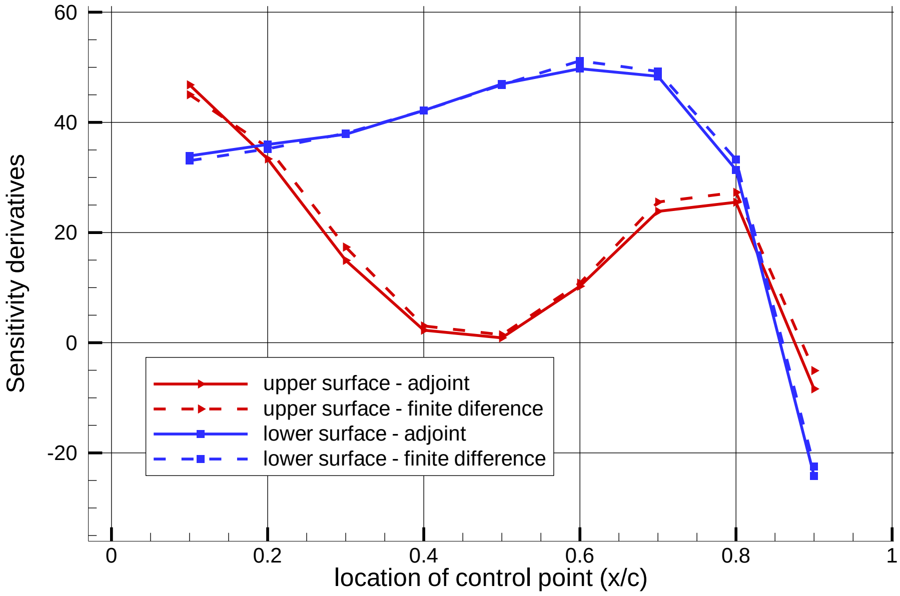


Figure 3-28 Sensitivity derivatives computed by the finite difference and the adjoint method

In the case of two-dimensional natural laminar flow, comparing with the finite difference estimations, the adjoint solver developed computes the sensitivity derivatives with high-accuracy (Figure 3-28). The presence of a natural laminar flow does not deteriorate the accuracy of gradient computation by the adjoint solver. The maximum deviation is only 2.3% of the $L_2\, norm$ of the sensitivity derivatives computed by finite difference approximation. Hence the results are encouraging to employ the adjoint solver to compute the sensitivity derivatives for natural laminar flows.

3.2.3 Drag Sensitivity at Fixed Lift: ONERA M6

Next, the three-dimensional adjoint solver developed is validated for three-dimensional flows. First, a transonic flow over an Onera M6 wing is considered. The flow conditions are the same as given in subchapter 3.1.4, which are the angle of attack of 3.06°, Re is 11.72×10^6 and M is 0.8395. In the validation study, the mesh defined in subchapter 3.1.4 is used.

Similar to the two-dimensional cases, a three-dimensional FFD box is utilized for surface deformations. The FFD box and the control points in this study are illustrated in Figure 3-29. 36 control points define the surface of the wing. The control points at the leading and trailing edges are kept fixed, and the remaining 24 points are taken as the design variables.

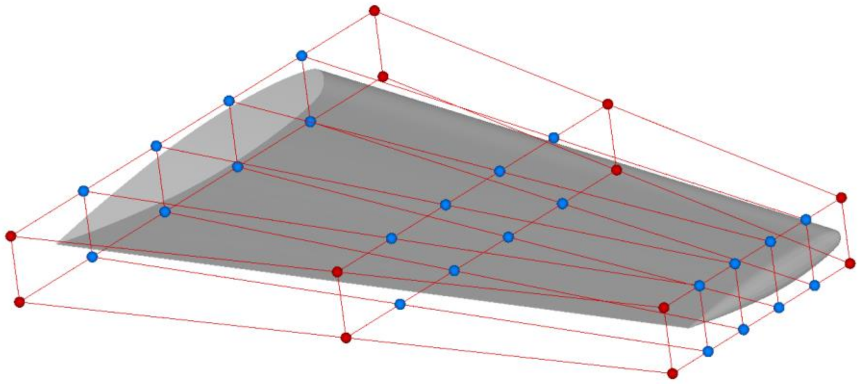


Figure 3-29 FFD box and control points (blue spheres depict active control points)

The objective function I is taken as the drag coefficient constrained by the fixed lift coefficient. The constraint is similarly imposed by a penalty function.

$$I = C_D + 2(C_L - C_L^{initial})^2 \qquad \text{Eq 3-4}$$

The sensitivity derivatives of the objective function with respect to the design variables located at the root, mid-span, and tip chord computed by adjoint solver against the derivatives computed by the finite central difference approximation is depicted respectively in Figure 3-30,

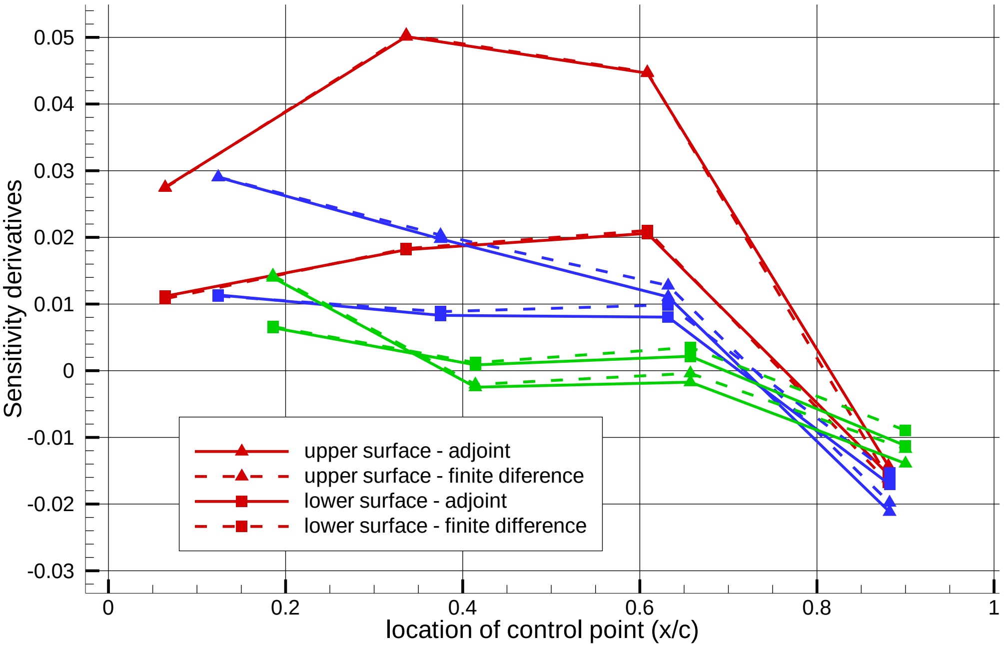


Figure 3-30 Comparison of sensitivity derivatives computed by the finite difference and the adjoint methods at the root chord (red), at the mid-span chord (blue), at the tip chord (green)

Figure 3-30 shows that the drag coefficient is much sensitive to the design variables placed upstream of the aft shock location. That is because the upstream flow of the shock is supersonic, and it is not affected by the downstream. The trends and the values of the sensitivity derivatives of the adjoint solver agree well with the results of the finite difference approximation. The maximum deviation is only 2.4% of the

$L_2\,norm$ of the sensitivity derivatives computed by the finite difference approximation.

3.2.4 Endurance Parameter Sensitivity: Low Aspect Ratio Wing

In the subchapter, the validation of the adjoint solver in the presence of a laminar to turbulent transition for a three-dimensional flow is performed. To this end, flow around MDO 505 wing is regarded. The flow is at M 0.15, an angle of attack 2.1°, and Re 3.4×10^6 per meter, which is the same flow condition regarded in case 3 defined in the subchapter 3.1.5.

Akin to the parametrization of the ONERA M6 wing, to manage the geometry, an FFD box with 36 control points is utilized (Figure 3-31). The perturbation of control points is limited in the vertical direction.

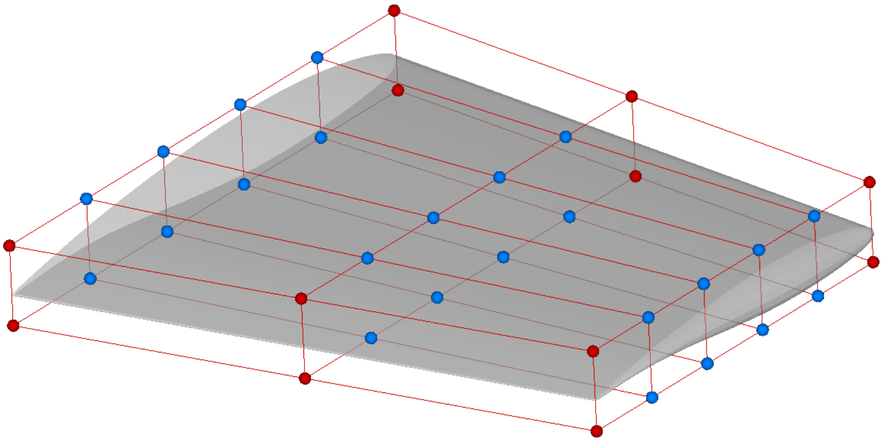


Figure 3-31 FFD box and control points (blue spheres depict active control points)

In this case, considering MALE/HALE type UAVs, the endurance parameter stated in Eq 3-5 is considered as the objective function.

$$I = {C_L}^{3/2}/C_D \qquad \text{Eq 3-5}$$

The results of the validation study are illustrated in Figure 3-32.

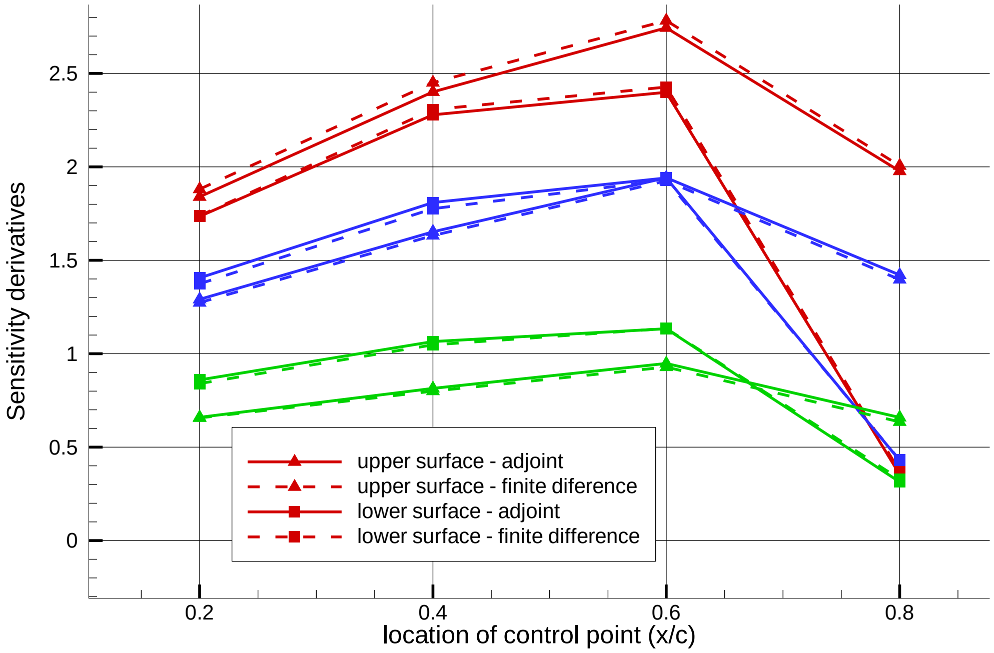


Figure 3-32 Comparison of sensitivity derivatives computed by the finite difference and the adjoint method at the root chord (red), at the mid-span chord (blue), at the tip chord (green)

According to the results, the agreement of the adjoint derivatives with the derivatives computed by the finite difference approximation is well. The maximum deviation in derivative values is less than 1% of the the $L_2\ norm$ of the derivatives computed by the finite difference approximation. Hence, the accuracy of the sensitivity derivatives computed by the adjoint solver for three-dimensional natural laminar flow is encouraging to employ the adjoint solver in an aerodynamic shape optimization considering laminar to turbulent transition.

3.2.5 Endurance Parameter Sensitivity: High Aspect Ratio Wing

The accuracy of the adjoint solver developed is next validated against a three-dimensional natural laminar flow case. The natural laminar flow over the high aspect ratio wing, which is studied in the flow solver validation study (subchapter 3.1.6) is considered. The sensitivity derivatives for the optimization of the endurance parameter are computed and compared against the finite difference based predictions.

The three-dimensional wing is parametrized by a three-dimensional FFD box with 36 control points (Figure 3-33), 12 of which are fixed in space. The remaining 24 control points are allowed to move only in the vertical direction.

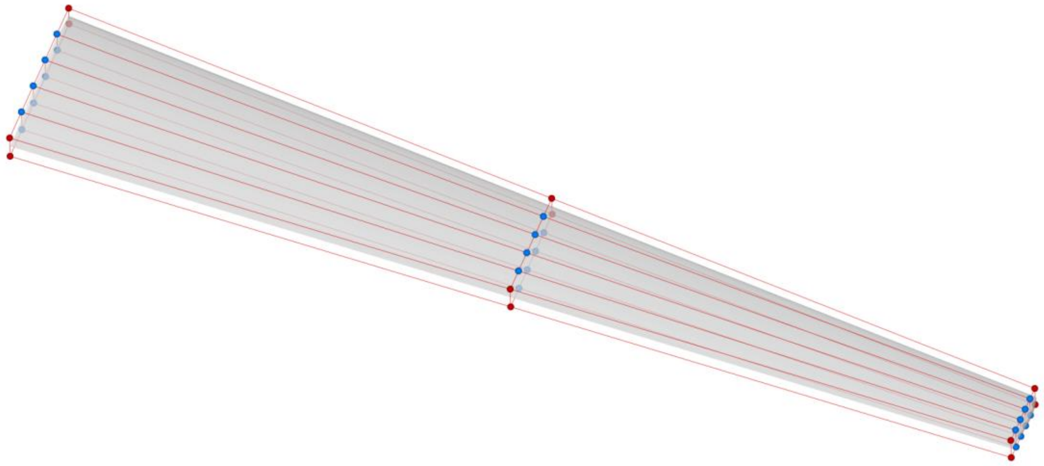


Figure 3-33 FFD box and control points (blue spheres depict active control points)

In this case, since the wing considered is a high-endurance UAV wing, the endurance parameter stated in Eq 3-6 is defined as the objective function to optimize.

$$I = {C_L}^{3/2}/C_D \qquad \text{Eq 3-6}$$

The adjoint and the finite difference based predictions are compared in Figure 3-34.

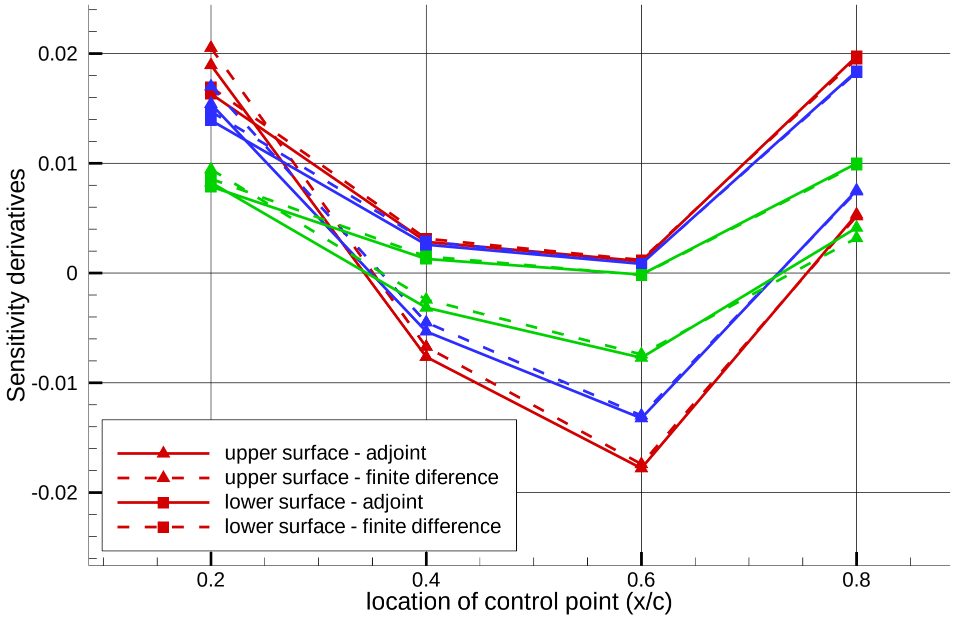


Figure 3-34 Comparison of sensitivity derivatives computed by the finite difference and the adjoint derivatives at the root chord (red), at the mid-span chord (blue), at the tip chord (green)

Similar to the previous cases, the predictions of the adjoint method are consistent with the finite difference approximation. The maximum deviation in sensitivity derivative values is below 3% of the the $L_2\ norm$ of the gradient computed by finite central difference technique. Thus the comparison reaffirms the accuracy of the sensitivity derivatives computed by the adjoint solver for three-dimensional natural laminar flows.

3.3 Optimization Studies

In order to demonstrate the functionality of the sensitivity derivatives computed by the adjoint solver, sample optimization studies considering the cases in the validation studies regarding the adjoint solver are performed through the optimization framework detailed in the subchapter 2.3. Since the studies are performed through the framework, the optimization algorithm utilized in the optimization studies to follow is the quasi-Newton method. In the studies, the parametrization and the objective functions are the same as in the corresponding validation studies.

3.3.1 Drag Minimization at a Fixed Lift: RAE 2822 Airfoil

The first optimization study is to minimize the drag coefficient of RAE 2822 airfoil at the flow condition of 2.8° angle of attack, M 0.73, and Re 6.5×10^6 for a fixed lift. The constraint of the lift coefficient is imposed by a penalty function. Hence the optimization problem is an unconstrained single-point optimization that may be written as is stated below,

$$\min_{\boldsymbol{\alpha}} \; I(\boldsymbol{\alpha}) = c_d(\boldsymbol{\alpha}) + 2\left(c_l(\boldsymbol{\alpha}) - c_l^{initial}(\boldsymbol{\alpha})\right)^2 \qquad \text{Eq 3-7}$$

where $\boldsymbol{\alpha}$ refers to the design variables. The parametrization is the same as illustrated in Figure 3-25.

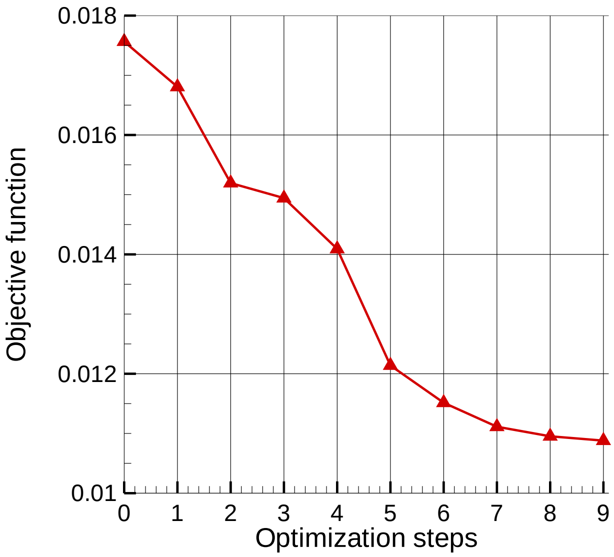


Figure 3-35 The evolution of the objective function value along the optimization steps

Figure 3-35 shows the history of the value of the objective function along optimization steps. The algorithm converges to the final design in 9 steps, excluding steps regarding line-search.

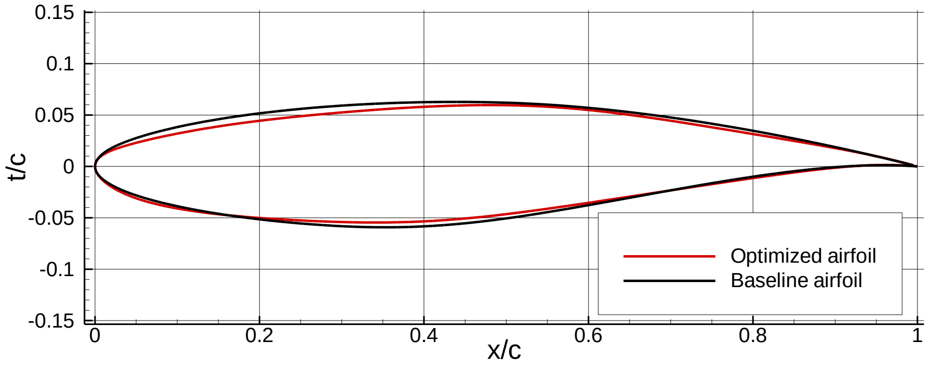


Figure 3-36 The baseline and optimized profile

The final design is illustrated below. At the final design, the maximum thickness of the airfoil is reduced, and the maximum thickness location is shifted to a bit downstream.

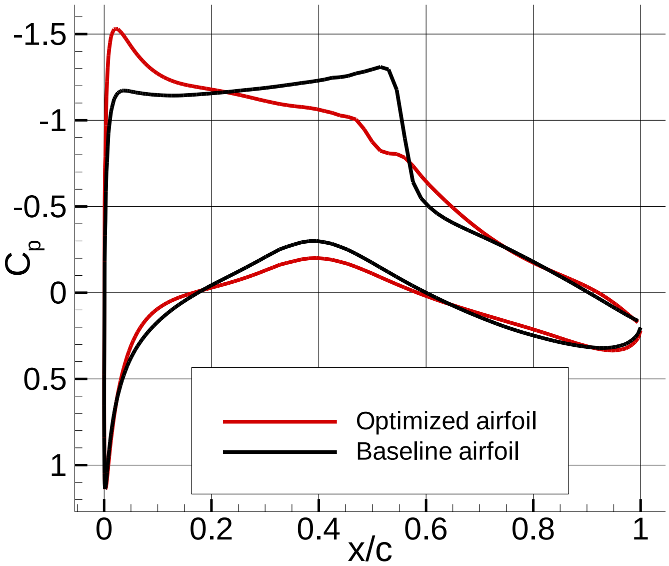


Figure 3-37 The pressure coefficient distributions of the baseline and optimized profile

Figure 3-37 illustrating the pressure coefficient distribution of both the baseline and optimized airfoils shows that the strong shock that appeared on the upper surface of the baseline airfoil is considerably weakened by the optimization algorithm and replaced with wiggles indicating two weak shocks appearing in between 0.45 and 0.55 chord. The resulting weaker shock is also observed in the Mach distribution given in Figure 3-38.

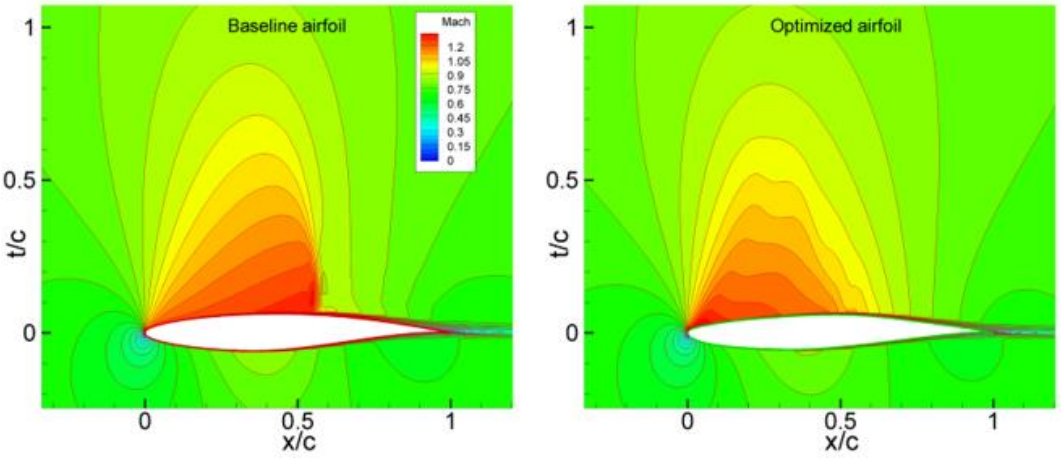


Figure 3-38 Mach distributions

To weaken the shock leads to reducing the drag coefficient of the airfoil 66 drag counts. The drag coefficient of the airfoil is now reduced down to 0.0109 from 0.0175. It should also be noted that by means of the penalty function, the constraint on the lift is satisfied, the lift coefficient stays almost constant as it drops from 0.821 to 0.820.

3.3.2 Maximization Range Parameter Maximization: NLF(1)-0.416

In the second optimization study, the maximization of the c_l/c_d ratio (i.e., range parameter) of NLF(1)-0416 airfoil at the flow condition of M 0.1, and Re 2×10^6 for a fixed angle of attack that corresponds to c_l 1.0 is considered. The defined unconstrained single-point optimization problem may be written as,

$$\max_{\boldsymbol{\alpha}} \; I(\boldsymbol{\alpha}) = c_l(\boldsymbol{\alpha})/c_d\,(\boldsymbol{\alpha}) \tag{Eq 3-8}$$

where $\boldsymbol{\alpha}$ refers to the design variables. The airfoil is parameterized with the same FFD box used in the validation study, as illustrated in Figure 3-27. Similar to the previous case, the control points, excluding the points located at the leading and trailing edges, are perturbed in the optimization study.

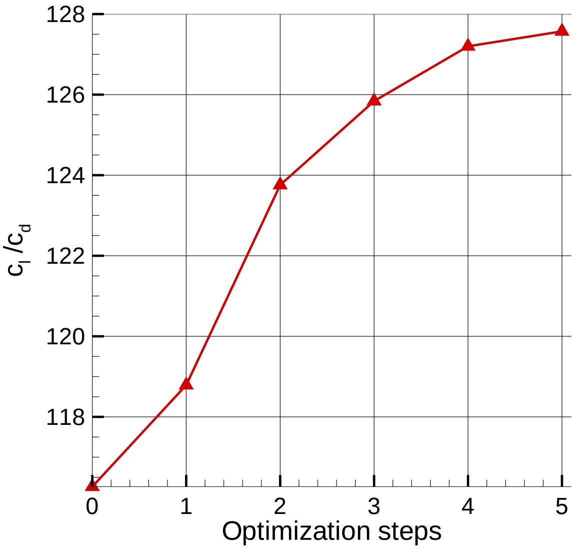


Figure 3-39 Evolution of c_l/c_d along the optimization steps

Figure 3-39 shows the evolution of the objective function along the optimization steps. The local maximum design point has been achieved by five steps, excluding steps regarding line-search.

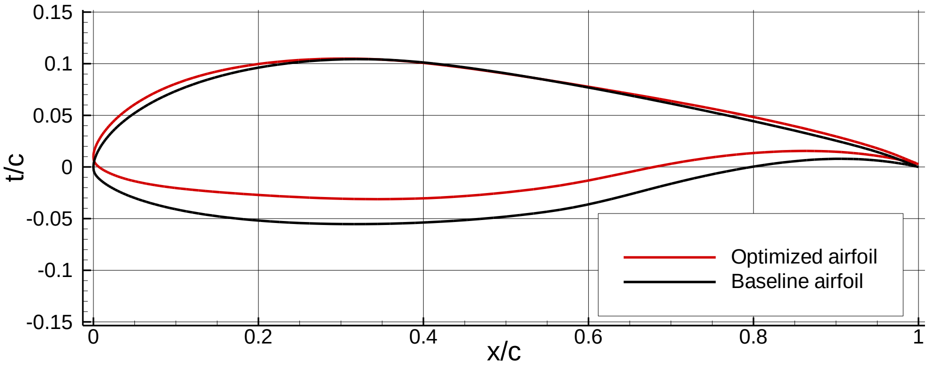


Figure 3-40 The baseline and optimized airfoils

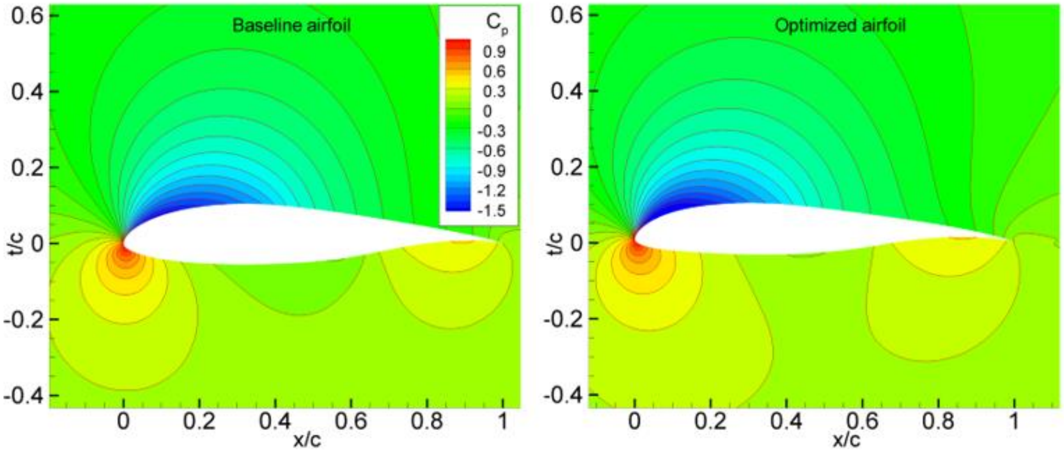


Figure 3-41 Pressure coefficient distributions on the baseline and optimized airfoils

Figure 3-40 shows the baseline and the optimized profiles. The corresponding pressure and skin friction coefficients are given in Figure 3-41 and Figure 3-42. It is observed that the minimum pressure is decreased at the suction side, whereas the pressure at the lower side is increased. Hence the lift coefficient is increased. That is achieved by enlarging the maximum camber of the airfoil. Meanwhile, the maximum thickness is decreased to reduce the drag coefficient by the optimization algorithm.

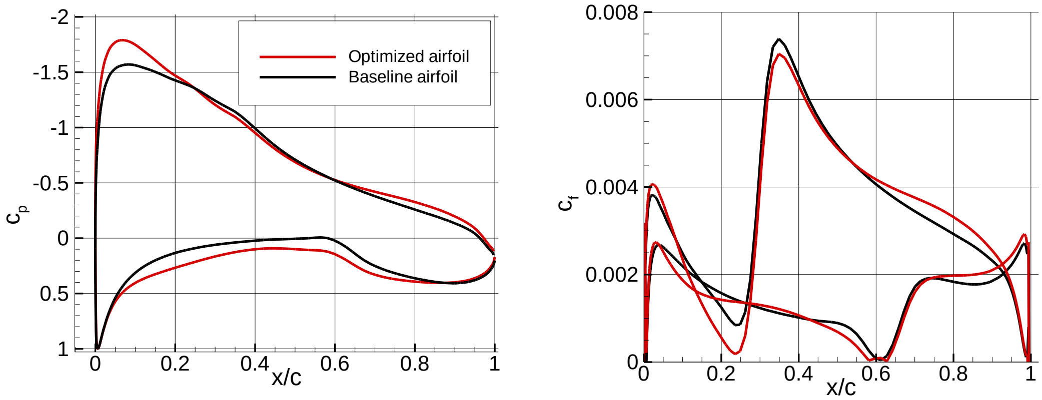


Figure 3-42 The pressure coefficient distributions of the baseline and optimized profile

Figure 3-42 illustrates the comparison of the pressure coefficient and skin friction coefficient distributions of the baseline and the optimized airfoil. When the figure is evaluated, it is evident that the algorithm pushes not to amplify the adverse pressure gradient on the upper surface, because an increase in the adverse pressure gradient results in a thicker boundary layer, hence higher pressure drag. It also causes an earlier transition onset. Therefore the algorithm does not make a significant change on the upper surface to keep the transition onset on the upper surface similar. An increase in lift is obtained by increasing the camber.

As a result, the initial lift coefficient value of 1.0 has been increased to 1.12. The drag coefficient value has only increased by 1 drag count from 0.0086 to 0.0087, and the range parameter increases from 116.3 to 127.6.

3.3.3 Drag Minimization at a Fixed Lift: ONERA M6 Wing

As the first three-dimensional optimization study using sensitivity derivatives computed by the adjoint solver, the minimization of the drag coefficient of the ONERA M6 wing at a fixed lift coefficient is considered. Similar to the study performed on RAE 2822 airfoil, the constraint is imposed by a penalty function to convert the problem into an unconstrained optimization problem. Accordingly, the defined optimization problem may be written as is stated in Eq 3-9,

$$\min_{\boldsymbol{\alpha}} \; I(\boldsymbol{\alpha}) = C_D(\boldsymbol{\alpha}) + 2\left(C_L(\boldsymbol{\alpha}) - C_L^{initial}(\boldsymbol{\alpha})\right)^2 \qquad \text{Eq 3-9}$$

where $\boldsymbol{\alpha}$ refers to the design variables. The parametrization of the surface of the wing is carried out using the FFD box defined in Figure 3-29.

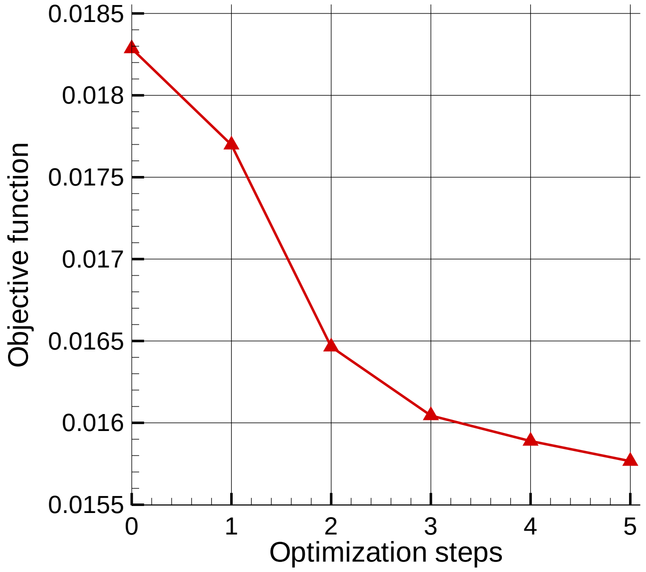


Figure 3-43 Evolution of objective function value along optimization steps

Figure 3-43 illustrates the history of the defined objective function along the optimization cycles. The convergence of the optimization study is achieved within 5 optimization steps, excluding the iterations performed regarding line-search.

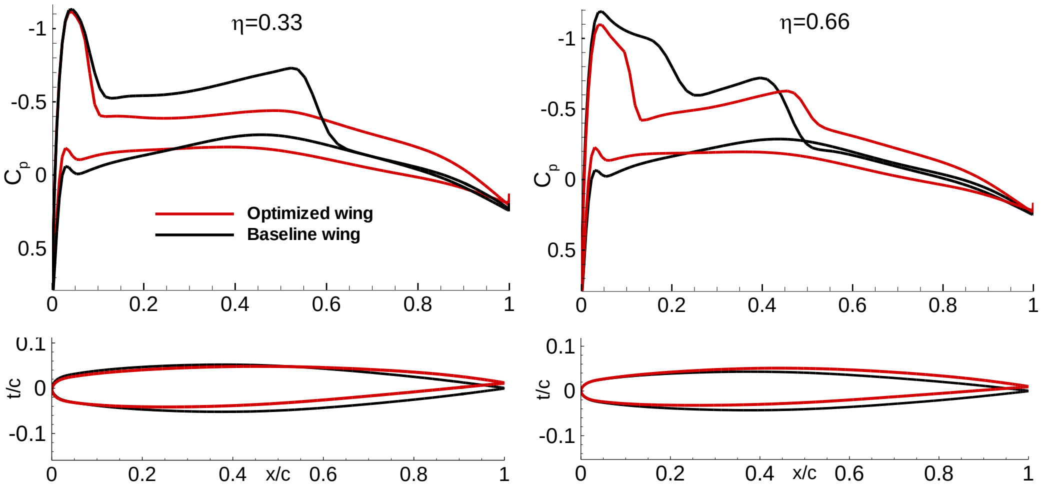


Figure 3-44 The wing sections of the baseline wing and the optimized wing (at η=0.33 and 0.66)

In order to minimize the drag coefficient, the algorithm decreases the maximum thickness of the wing sections. Moreover, the algorithm reduces the incidence angle such that the flow angle and the incidence angle of the wing are closer to each other.

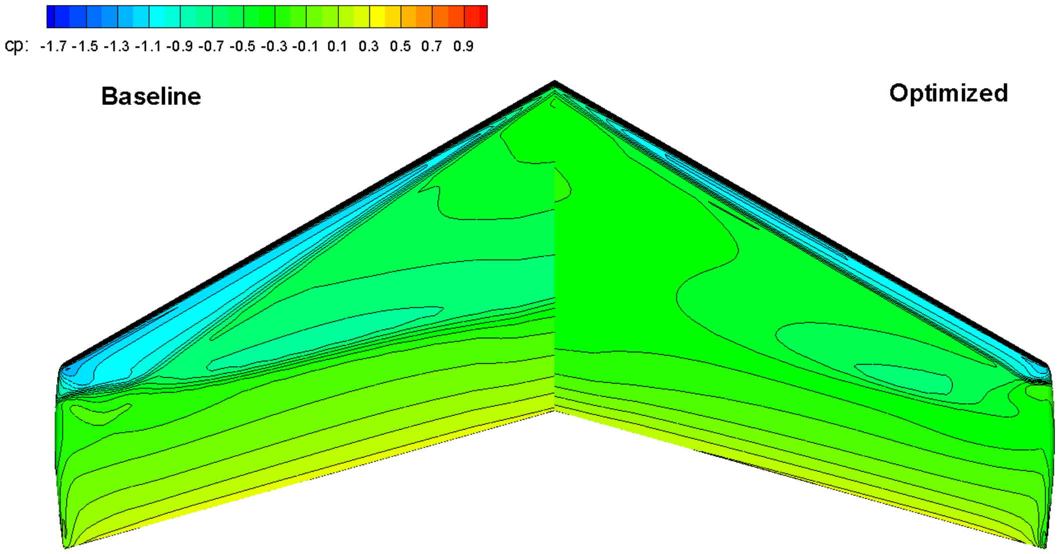


Figure 3-45 Pressure coefficients for baseline geometry and optimized geometry

Reducing the maximum thickness and incidence angles of the wing sections resulted in that the aft shock is either removed or weakened along the spanwise (Figure 3-44, Figure 3-45). Furthermore, the lift decrease due to the reduction of the incidence angle has been compensated by alleviating the lift loss occurring due to shock.

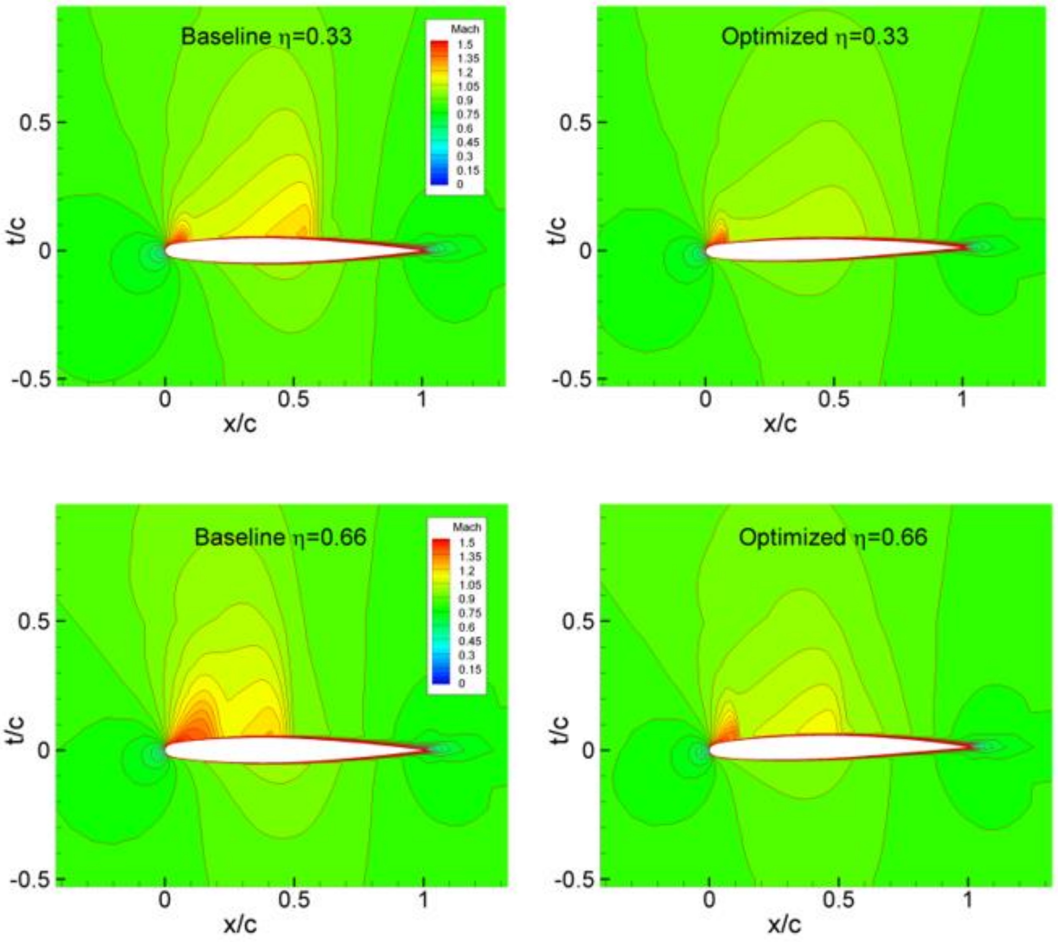


Figure 3-46 Mach contours at different wingspan sections

Consequently, although there are still shocks on the upper surface, as seen in Figure 3-46, the optimization algorithm executes its duty properly, and the drag coefficient decreases from 0.0182 to 0.0157 by keeping the lift coefficient constant.

3.3.4 Endurance Parameter Maximization: Low Aspect Ratio Wing

In this case, the endurance parameter of the low aspect ratio MDO 505 wing considered in validation studies is optimized at a fixed angle of attack 2.1°. The optimization problem may be stated as in Eq 3-10,

$$\max_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = {C_L}^{3/2}(\boldsymbol{\alpha})/C_D(\boldsymbol{\alpha}) \qquad \text{Eq 3-10}$$

where $\boldsymbol{\alpha}$ denotes the design variables of the problem. In the study, the gradients computed by the adjoint solver regards the physics of the transitional flow modeled by the B-C transition model, hence the optimization algorithm also does. The flow over the wing is at M 0.15, and Re 3.4×10^6 for 1 m chord.

The parametrization of the wing is employed by the same FFD box utilized earlier in validation studies (Figure 3-31). The FFD box has 36 control points 24 of which are taken as the design variables. The movement of them is limited in the vertical direction.

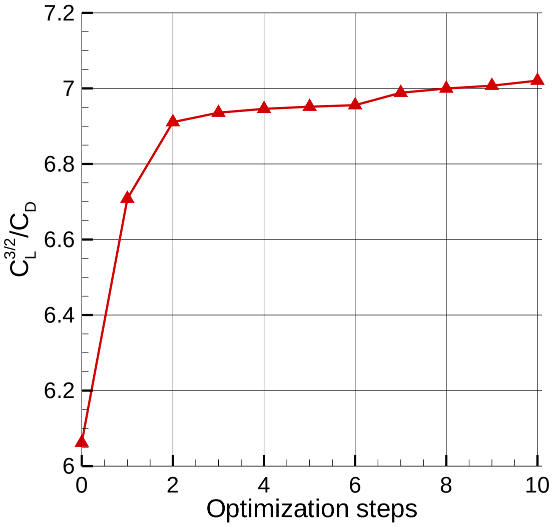


Figure 3-47 Evolution of endurance parameter value throughout optimization steps

The optimization process is similarly driven by the quasi-Newton algorithm available in Dakota. Figure 3-47 illustrates the evolution of the defined objective function throughout the optimization steps. The convergence of the optimization study is achieved within 10 optimization steps.

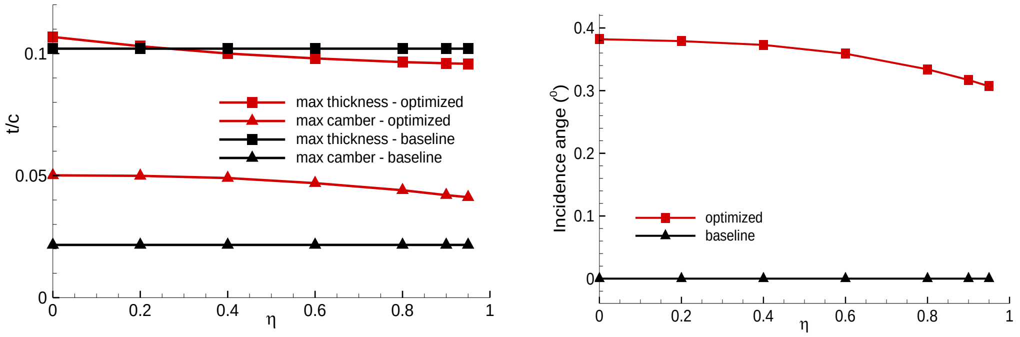


Figure 3-48 Maximum thickness and maximum camber to chord ratio distribution on the baseline and optimized wing along the span

Figure 3-48 shows the variations of the thickness, the camber, and incidence angle along the wing span. The change in the maximum thickness is observed to be rather small, but the camber, in general, is increased all along the span. The effect of increased camber and incidence angle may be observed in the pressure coefficient contours given in Figure 3-49. It should also be noted that the relative decrease in the camber and the incidence angle towards the tip tends to create an elliptic lift distribution, which helps to lower the induced drag.

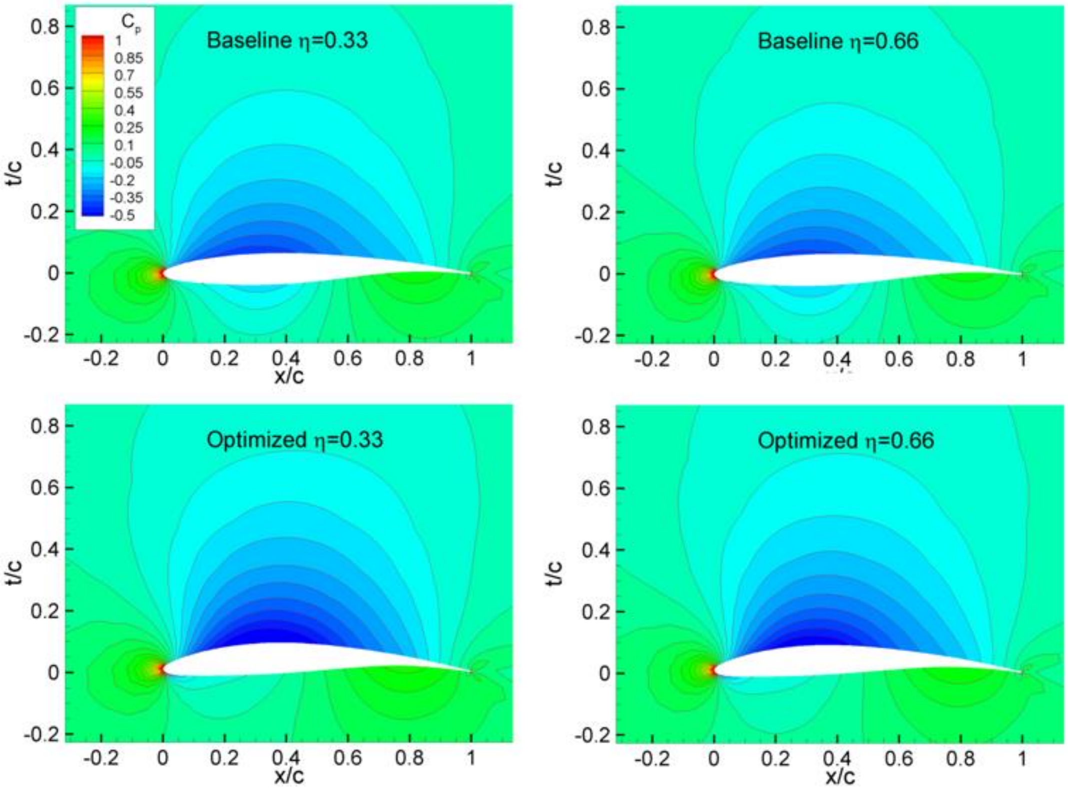


Figure 3-49 Pressure coefficient contours for the baseline and optimized wing

In Figure 3-50, the change in the distribution of pressure and the skin friction coefficients are given in two spanwise stations. The optimized pressure distributions, which cause the lift to increase, have higher suction pressures due to increased camber. Whereas, the skin friction distribution at $\eta = 0.33$ indicates an early transition onset closer to the root, which causes the skin friction drag to increase by 3 counts. Yet, there is no flow separation observed on the optimized wing. Furthermore, it should be noted that the sensitivity derivatives computed by the adjoint solver account for the transitional flow modeled by the B-C transition model.

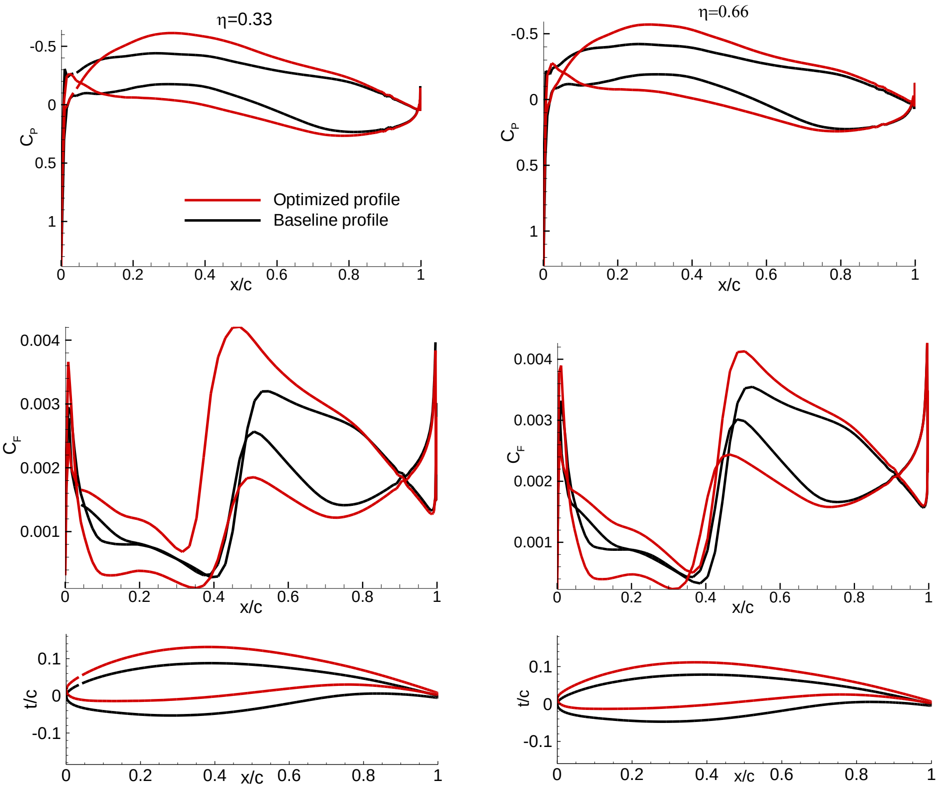


Figure 3-50 The wing sections of the baseline wing and the optimized wing

To give a better insight, the skin friction contours on the upper surfaces of the baseline and optimized wing are given in Figure 3-51.

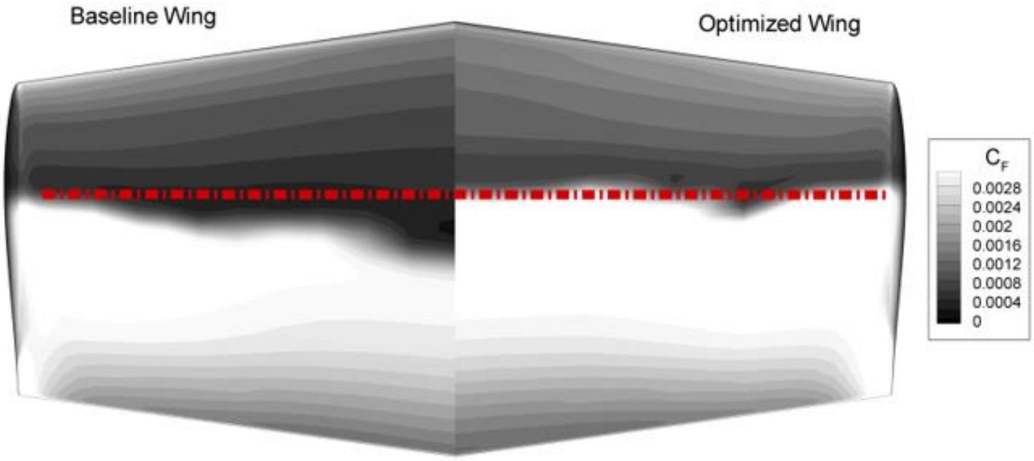


Figure 3-51 Skin friction distribution on the upper surfaces of the baseline and optimized wing

As a result of the optimization study, the endurance parameter of the wing is increased by around 16%. Thus, the framework demonstrates its capability in three-dimensional aerodynamic shape optimization for natural laminar flows, as well.

3.3.5 Endurance Parameter Maximization: High Aspect Ratio Wing

In this optimization study, the endurance parameter of the experimental MALE UAV wing considered in validation studies is optimized at a fixed angle of attack. The optimization problem may be stated as is in Eq 3-11,

$$\max_{\boldsymbol{\alpha}} I(\boldsymbol{\alpha}) = {C_L}^{3/2}(\boldsymbol{\alpha})/C_D(\boldsymbol{\alpha})\,, \qquad \text{Eq 3-11}$$

where $\boldsymbol{\alpha}$ denotes the design variables of the problem. In the study, as is in the previous optimization study, the gradients computed by the adjoint solver regards the physics of the transitional flow modeled by the B-C transition model, hence the optimization algorithm also does. The flow over the wing is at M 0.2, and Re 600×10^3 at the mean aerodynamic chord, and an angle of attack 6°.

Similar to previous cases, the parametrization technique is the FFD technique. The FFD box created for parametrization contains 36 control points. The FFD box and the control points are as demonstrated in Figure 3-31.

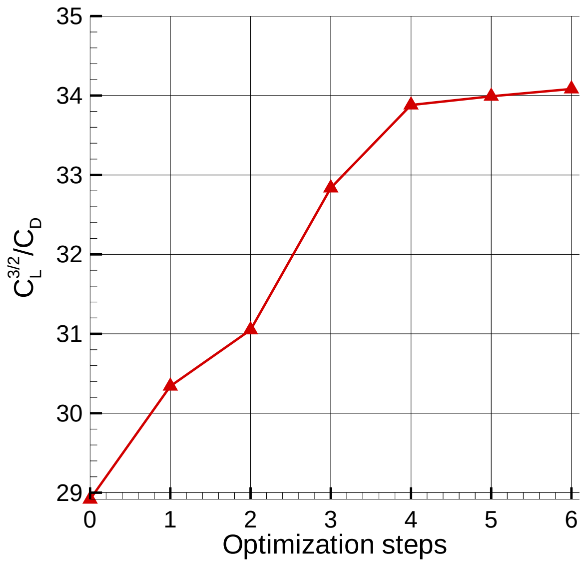


Figure 3-52 Evolution of endurance parameter value along optimization steps

The optimization algorithm is the quasi-Newton algorithm. Figure 3-52 illustrates the history of the defined objective function along the optimization steps. In the study, the convergence is achieved within 11 optimization steps.

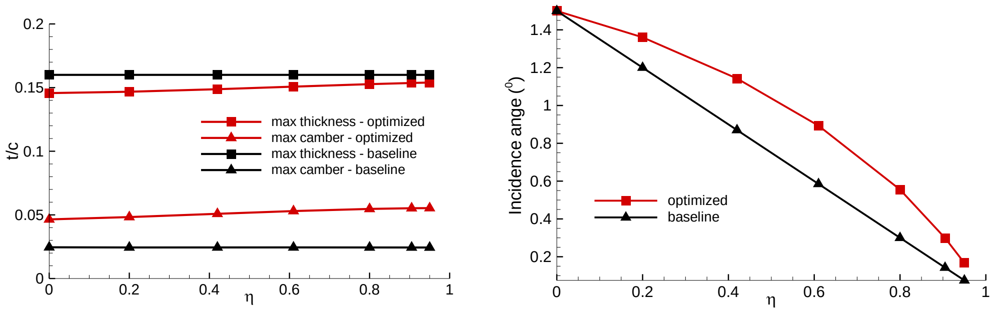


Figure 3-53 Maximum thickness and maximum camber to chord ratio distribution on the baseline and optimized wing along the span

Figure 3-53 shows that the optimization algorithm increases the camber and the incidence angle along the wingspan, whereas the thickness is reduced to decrease the drag coefficient. The effect of the change of the camber, thickness, and incidence angle along the span in aerodynamic efficiency may be evaluated through the lift distribution.

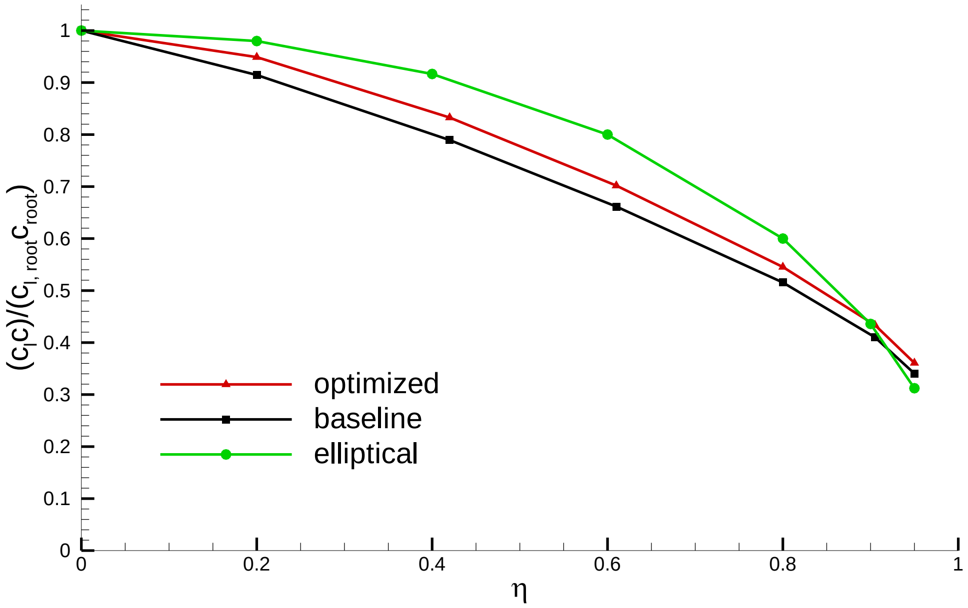


Figure 3-54 The lift distribution along the span

Figure 3-54 depicts the normalized lift distributions of the baseline and the optimized wing, as well as the elliptic lift distribution. According to the results, the change in wing sections done by the optimization algorithm results in a lift distribution closer to an elliptic lift distribution, hence a lower induced drag at the corresponding lift.

Figure 3-55 Skin friction coefficient distribution on the upper surfaces of the baseline and the optimized wing

In the case, the algorithm focused on increasing the lift and reduce the induced drag. The drag occurring due to the viscosity remains in the background. Figure 3-55 depicts skin friction coefficient distributions on the upper surface. According to the results, the transition onset moves forward towards the leading edge, which results in 25 drag counts increase in the skin friction coefficient. Nevertheless, the effect of increasing the lift coefficient, and having a lift distribution closer to that of an elliptic, the endurance parameter is increased by around 18%.

CHAPTER 4

CONCLUSION

A discrete adjoint-based aerodynamic shape optimization framework for natural laminar flows is successfully developed. An in-house RANS solver is heavily modified in *Fortran 95* and is equipped with the B-C transition model in order to predict the transition onset for natural laminar flows. Validation studies for both turbulent and natural laminar flows over airfoils and wings are successfully performed. It is also shown that the B-C transition model predicts the transition onset accurately and agrees well with the numerical and experimental studies.

Following the further development and the validation of the in-house RANS solver, an accompanying discrete adjoint solver is next developed for natural laminar flows. In the development of the adjoint solver, an AD tool, *TAPENADE*, is utilized to take the discrete derivatives of the flow solver, namely flux Jacobians. In the study, the usage of the AD tool is straightforward and less laboring. Since the computed Jacobian matrices are the exact Jacobians of the discrete flow solver, there is no simplification, such as frozen turbulence. The implementation of the B-C transition model does not increase the rank of the Jacobian matrices and does not degrade the computational efficiency and memory requirements. The solution of the adjoint variables is obtained by a multifrontal LU factorization provided by *MUMPS* library under *PETSc* environment. Although the memory requirement for the direct solution of the system of equations by the LU decomposition is large, the direct solution is free from numerical instabilities often faced in iterative solutions.

In the aerodynamic shape optimization study, aerodynamic surfaces are parametrized by the FFD method. Once the sensitivity derivatives of the objective function with respect to grid coordinates are evaluated, their projection on the FFD coordinates is obtained by the use of the *SU2_DEF* module provided by the *SU2* software suite.

The development of the adjoint solver is followed by the validation studies against the finite difference based predictions. It is shown that the sensitivity gradients computed by the adjoint method are in very good agreement with the finite difference predictions for both turbulent and natural laminar flows. The present results also show that the usage of an AD provides an accurate method for the development of a discrete adjoint solver, as well as providing a straightforward and robust method.

An open-source optimization tool, *DAKOTA*, is used to drive the optimization steps based on the sensitivity derivatives/gradients computed by the adjoint solver. The quasi-newton method available in *DAKOTA* is chosen as the optimization method.

The framework parameterizes the shape by using the FFD technique. The framework, then, provides a new set of design variables, deforms the grid, and solves the flow over the modified shape. Moreover, the flow solver also computes the partial derivatives appearing in the adjoint system. Subsequently, the adjoint solver calculates the grid sensitivities of the objective function. The grid sensitivities are projected to design parameters by employing *SU2_DOT_AD*. Hence, the sensitivity of the objective function to design parameters are computed. The objective function and the sensitivity gradients are then delivered to the optimization algorithm. This cycle lasts until the convergence criterion is achieved.

Finally, in order to demonstrate the functionality of the developed discrete adjoint method, the optimization of the aerodynamic characteristics of several airfoils and wings are performed via the optimization framework. In the optimization studies, the method succeeds in optimizing aerodynamic characteristics of the airfoils and wings for both turbulent and natural laminar flows. In the first optimization study, the drag coefficient of RAE 2822 in transonic flow is decreased from 175 drag counts to 109 drag counts at a fixed lift. The lift coefficient of NLF(1)-0416 airfoil is then increased from 1.0 to 1.12, which only resulted in 1 drag count increase. The drag coefficient of the ONERA M6 wing is decreased by 25 drag counts at a fixed lift. Finally, the

MDO 505 wing and TAI’s experimental high aspect ratio wing is optimized for endurance, and the endurance parameters of them are increased by 16% and 18%, respectively. Thus, the implementation has shown the functionality of adjoint methods in high-fidelity aerodynamic shape optimization for natural laminar flows as well as turbulent flows.

4.1 Further Studies

Numerical results show that the incorporation of gradients into a surrogate model, can be used to break the *curse of dimensionality* (Dalbey, 2013). However, that is only attractive when the computational cost to compute the gradients is comparable to that of the objective function. Fortunately, that is possible by adjoint methods. Hence, as an extension of the study, we are currently looking into combining the adjoint method with a kriging surrogate-based model to achieve the global optimum. In this way, it is aimed to take advantage of the best features of both methods.

The kriging surrogate-based optimization algorithms are successful in achieving the global optimum; however, they face the *curse of dimensionality* problem when dealing with a large number of design variables. On the other hand, the adjoint method is successful and computationally efficient to compute the gradients; however, when combined with a gradient-based optimization algorithm, it achieves the local optimum.

Although, by incorporating the gradients into a surrogate model, the *curse of dimensionality* problem is overcome, there is still an open issue. The correlation matrices appearing in a gradient-enhanced kriging method are ill-conditioned, which is a problem to be solved to get a robust and efficient method. Therefore, in the extension study, a recently developed method (Özkaya & Gauger, 2020), that enables surrogate models to incorporate gradient information without causing robustness problems, will be implemented to overcome the problem.

REFERENCES

Abu-Ghannam, B. J., & Shaw, R. (1980). Natural Transition of Boundary Layers-The Effects of Turbulence, Pressure Gradient, and Flow History. *Journal of Mechanical Engineering Science, 22*(5), 213-228. doi:10.1243/JMES_JOUR_1980_022_043_02

Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Ebeida, M. S., Eddy, J. P., Eldred, M. S., . . . Winokur, J. G. (2014, updated 2019). *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.11 User's Manual.* Sandia Technical Report SAND2014-4633.

Ajwani, D., & Meyerhenke, H. (2010). Chapter 5. Realistic Computer Models. *Lecture Notes in Computer Science, 5971:194-*. doi:10.1007/978-3-642-14866-8_5

Albring, T., Beckett, Z. Y., & Gauger, N. R. (2019). Challenges in Sensitivity Computations for (D)DES and URANS. *AIAA Scitech 2019 Forum.* doi:10.2514/6.2019-0169

Albring, T., Sagebaum, M., & Gauger, N. R. (2016). Efficient Aerodynamic Design using the Discrete Adjoint Method in SU2. *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* Washington, D.C. doi:10.2514/6.2016-3518

Althaus, D. (1972). *Stuttgarter Profil Katalog 1.* Sttutgart University.

Amestoy, P. R., Buttari, A., L'excellent, J.-Y., & Mary, T. (2019). Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures. *ACM Transactions on Mathematical Software, 45*(1), 2:1-2:26. doi:10.1145/3242094

Amestoy, P. R., Duff, I. S., L'excellent, J., & Koster, J. (2001). A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling.

SIAM Journal on Matrix Analysis and Applications,, 23(1), 15-41. doi:10.1137/s0895479899358194

Angrand, F. (1983). Optimum Design for Potential Flows. *International Journal of Numerical Methods in Fluids*, 265-282. doi:10.1002/fld.1650030306

Balan, A., Park, M., & Anderson, K. (2019). Adjoint-based Anisotropic Mesh Adaptation for a Stabilized Finite-Element Flow Solver. *AIAA Aviation 2019 Forum.* doi:10.2514/6.2019-2949

Balay, S., Abhyankar, S., Adams, M. F., Brown, j., Brune, P., Buschelman, K., . . . Munson, T. (2019). Retrieved from PETSc Web Page: https://www.mcs.anl.gov/petsc.

Balay, S., Abhyankar, S., Adams, M. F., Brown, j., Brune, P., Buschelman, K., . . . Zhang, H. (2020). *PETSc Users Manual.* ANL-95/11 - Revision 3.13, Argonne National Laboratory.

Balay, S., Gropp, W. D., McInnes, L. C., & Smith, B. F. (1997). Efficient Management of Parallelism in Object Oriented Numerical Software Libraries. In E. Arge, A. M. Bruaset, & H. p. Langtangen (Eds.), *Modern Software Tools in Scientific Computing* (pp. 163-202). Boston, MA: Birkhauser Press.

Barth, T., & Jespersen, D. (1989). The design and application of upwind schemes on unstructured meshes. *AIAA 27th Aerospace Sciences Meeting.* Reno, Nevada. doi:10.2514/6.1989-366

Basha, W., & Ghaly, W. (2007). Drag Prediction in Transitional Flow Over Two-Dimensional Airfoils. *Journal of Aircraft, 44*, 824-832. doi:10.2514/1.22939

Bischof, C. H., Roh, L., & Mauer, A. (1997). ADIC: An Extensible Automatic Differentiation Tool for ANSI-C. *Software: Practice and Experience, 27*(12), 1427-1456. doi:10.1002/(SICI)1097-024X(199712)27:12%3C1427::AID-SPE138%3E3.0.CO;2-Q

Bischof, C., Khademi, P., Mauer, A., & Carle, A. (1996). Adifor 2.0: Automatic Differentiation of Fortran 77 Programs. *IEEE Computational Science & Engineering, 3*(3), 18-32. doi:10.1109/99.537089

Blazek, J. (2001). *Computational Fluid Dynamics: Principles and Applications.* Elsevier Science.

Boussinesq, J. (1877). La Theroie des Eaux Courantes. *Memoires presentes par divers savants a l'Academie des Sciences de l'Institut National de France, 23*(1).

Boussinesq, J. (1896). Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes à grande section. *Comptes Rendus de l' Academia Des Sciences, 122*.

Broyden, C. G. (1969). A new double-rank minimization algorithm. *AMS Notices, 16*, 670.

Cakmakcioglu, S. C., Bas, O., & Kaynak, U. (2017). A correlation-based algebraic transition model. *Proceedings of the Institution of Mechanical Engenieers, Part C: Journal of Mechanical Engineering Science*. doi:10.1177/0954406217743537

Campell, R. L., Campell, M. L., & Streit, T. (2011). Progress Toward Efficient Laminar Flow Analysis and Design. *29th AIAA Applied Aerodynamics Conference.* Honolulu, Hawaii. doi:10.2514/6.2011-3527

Carnarius, A., Thiele, F., Özkaya, O. E., & Gauger, N. R. (2010). Adjoint Approaches For Optimal Flow Control. *5th Flow Control Conference.* doi:10.2514/6.2010-5088

Cheylan, I., Fritz, G., Ricott, D., & Sagaut, P. (2019). Shape Optimization Using the Adjoint Lattice Boltzmann Method for Aerodynamic Applications. *AIAA journal, 57*(7). doi:10.2514/1.J057955

Choi, Y.-H., & Merkle, C. L. (1993). The Application of Preconditioning in Viscous Flows. *Journal of Computational Physics, 105*(2), 207-223. doi:10.1006/jcph.1993.1069

Coder, J. G., & Maughmer, M. D. (2014). Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation. *AIAA journal, 52*(11), 2506-2512. doi:10.2514/1.J052905

Cook, P. H., McDonald, M. A., & Firmin, M. C. (1979). Aerofoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements. *Experimental Data Base for Computer Program Assessment.* AGARD Report AR-138.

Coquillart, S. (1990). Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *17th annual conference on Computer graphics and interactive techniques, SIGGRAPH 1990, 24(4),* pp. 187-196. doi:doi.org/10.1145/97879.97900

Crouch, J. (2015). Boundary-Layer Transition Prediction for Laminar Flow Control (Invited). *45th AIAA Fluid Dynamics Conference.* doi:10.2514/6.2015-2472

Cummings, R. M., & Garcia, J. A. (1993). *Numerical prediction of transition of the F-16 wing at supersonic speeds.* NASA-CR-194407.

Dalbey, K. R. (2013). *Efficient and Robust Gradient Enhanced Kriging Emulators.* SAND2013-7022, Sandia National Laboratories, Optimization and Uncertainty Quantification Department, Albuquerque, NM.

Deka, M., Brahmachary, S., Thirumalaisamy, R., & Dalal, A. (2018). A new Green-Gauss reconstruction on unstructured meshes. Part I: Gradient reconstruction. *Journal of Computational Physics*. doi:10.1016/j.jcp.2018.10.023

Diakakis, K., Papadakis, G., & Voutsinas, S. G. (2019). Assessment of transition modeling for high Reynolds flows. *Aerospace Science and Technology, 85*, 416-428. doi:10.1016/j.ast.2018.12.031

Dilgen, C. B., Dilgen, S. B., Fuhrman, D. R., Sigmund, O., & Lazarov, B. S. (2018). Topology optimization of turbulent flows. *Journal of Computer Methods in Applied Mechanics and Engineering, 331*, 363-393. doi:10.1016/j.cma.2017.11.029

Diskin, B., & Thomas, J. (2008). *Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio.* NIA Report No. 2008-12, National Institute of Aerospace.

Diskin, B., & Thomas, J. L. (2011). Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes. *AIAA journal, 49*(4), 836-854. doi:10.2514/1.J050897

Djeddi, R., & Ekici, K. (2019). FDOT: A Fast, memory-efficient and automated approach for Discrete adjoint sensitivity analysis using the Operator overloading Technique. *Journal of Aerospace Sicence and Technology*, 159-174. doi:10.1016/j.ast.2019.05.004

Djeddi, R., & Ekici, K. (2020). Memory Efficient Adjoint Sensitivity Analysis for Aerodynamic Shape Optimization. *AIAA Scitech 2020 Forum.* doi:10.2514/6.2020-0885

Duck, P. W., & Hall, P. (Eds.). (1995). *IUTAM Symposium on Nonlinear Instability and Transition in Three-Dimensional Boundary Layers.* doi:10.1007/978-94-009-1700-2

Dwight, R. P. (2006). Robust Mesh Deformation using the Linear Elasticity. *Proceedings of the Fourth International Conference on Computational Fluid Dynamics,* (pp. 401-406). Berlin.

Eastman, J. N., Kenneth, W. E., & Pinkerton, R. M. (1933). *The Characteristics of 78 Related Airfoil Sections from Tests in the Variable Density Wind Tunnel.* NACA-TR-586.

Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., & Alonso, J. J. (2016). SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal, 54*(3), 828-846. doi:10.2514/1.j053813

Elliot, J., & Peraire, J. (1996). Aerodynamic design using unstructured meshes. *27th Fluid Dynamics Conference.* New Orleans. doi:10.2514/6.1996-1941

Evgrafov, A., Gregersen, M., & Sorensen, M. P. (2011). Convergence of Cell Based Finite Volume Discretizations for Problems of Control in the Conduction Coefficients. *ESAIM Mathematical Modelling and Numerical Analysis, 45*(6), 1059-1080. doi:10.1051/m2an/2011012

Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal, 13*(3), 317-322. doi:10.1093/comjnl/13.3.317

Fujino, M. (2005). Design and Development of the Honda Jet. *Journal of Aircraft, 42*(3), 755-764. doi:10.2514/1.12268

Fujino, M., Yoshizaki, Y., & Kawamura, Y. (2003). Natural-Laminar-Flow Airfoil Development for a Lightweight Business Jet. *Journal of Aircraft, 40*(4), 609-615. doi:10.2514/2.3145

Giering, R., & Kaminski, T. (1998). Recipes for Adjoint Code Construction. *ACM Transactions on Mathematical Software, 24*(4), 437-474. doi:10.1145/293686.293695

Glowinski, R., & Pironneau, O. (1975). On the numerical computation of the minimum-drag profile in laminar flow. *Journal of Fluid Mechanics, 72,* 385-399. doi:10.1017/s0022112075003436

Gökhan, A. (2014). *Micro-Siting of Wind Turbines using Navier-Stokes Solutions Coupled with a Numeriacal Weather Predictions Model.* Doctoral dissertation, Middle East Technical Universtiy, Ankara.

Goldfarb, D. (1970). A Family of Variable-Metric Methods Derived by Variational Means. *Mathematics of Computation, 24*(109), 23-26. doi:10.2307/2004873

Green, B. E., Whitesides, J. L., Campell, R. L., & Mineck, R. E. (1997). Method for the Constrained Design of Natural Laminar Flow Airfoils. *Journal of Aircraft, 34*(6), 706-712. doi:10.2514/2.2248

Griewank, A., & Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* (2nd ed.). Philadelphia: SIAM.

Griewank, A., Juedes, D., & Utke, J. (1996). Algorithm 755: ADOL-C: A Package for the Automatic differentiation of Algorithms Written in C/C++. *ACM Transactions on Mathematical Software, 22*(2), 131-167. doi:10.1145/229473.229474 ·

Haider, F., Croisille, J.-P., & Courbet, B. (2009). Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids. *Numerische Mathematik, 113*(4), 555-600. doi:10.1007/s00211-009-0242-6

Harris, J. E., Iyer, V., & Radwan, S. (1987). Numerical solutions of the compressible 3-D boundary-layer equations for aerospace configurations with emphasis on LFC. *Research in Natural Laminar Flow and Laminar-Flow Control*, (pp. 517-545).

Harten, A., Lax, P. D., & van Leer, B. (1983). On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Reviews, 25*(1), 35-61.

Hascoet, L., & Pascual, V. (2013). The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification. *ACM Transactions on Mathematical Software, 39*(3), 1-43. doi:10.1145/2450153.2450158

Haselbacher, A., & Blazek, J. (2000). Accurate and Efficient Discretization of Navier-Stokes Equations on Mixed Grids. *AIAA journal*, *38*(11). doi:10.2514/2.871

He, P., Mader, C. A., Martins, J. R., & Maki, K. J. (2020). DAFoam: An Open-Source Adjoint Framework for Multidisciplinary Design Optimization with OpenFOAM. *AIAA Journal, 58*(3), 1304-1319. doi:10.2514/1.J058853

Iyer, V., Spall, R. E., & Dagenhart, J. R. (1994). Computational study of transition front on a swept wing leading-edge model. *Journal of aircraft, 31*(1), 72-78. doi:10.2514/3.46457

Jameson, A. (1988). Aerodynamic design via control theory. *Journal of Scientific Computing*, 233-260. doi:10.1007/BF01061285

Jameson, A. (1995). Optimum Aerodynamic Design Using CFD and Control Theory. *12th Computational Fluid Dynamics Conference.* doi:10.2514/6.1995-1729

Jameson, A. (1999). Re-Engineering the Design Process Through Computation. *Journal of Aircraft*, 36-50. doi:10.2514/2.2412

Jameson, A., & Reuther, J. (1994). Control Theory Based Airfoil Design using the Euler Equations. *5th Symposium on Multidisciplinary Analysis and Optimization.* doi:10.2514/6.1994-4272

Jameson, A., Pierce, N., & Martinelli, L. (1997). Optimum aerodynamic design using the Navier-Stokes equations. *35th Aerospace Sciences Meeting & Exhibit.* Reno. doi:10.2514/6.1997-101

Kammegne, M. J., Botez, R. M., Manou, M., & Mebarki, Y. (2016). Experimental Wind Tunnel Testing of a New Multidisciplinary Morphing Wing Model. *Conference: Proceedings of the 18th International Conference on Mathematical Methods, Computational Techniques and Intelligent System (MAMETICS '16).* Venice, Italy.

Kaynak, Ü., Bas, O., Cakmakcioglu, S. C., & Tuncer, İ. H. (2019). Transition Modeling for Low to High Speed Boundary Layer Flows with CFD Applications. doi:10.5772/intechopen.83520

Kenway, G. K., Mader, C. A., He, P., & Martins, J. R. (2019). Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences, 110*. doi:10.1016/j.paerosci.2019.05.002

Khayatzadeh, P., & Nadarajah, S. (2011). Aerodynamic Shape Optimization via Discrete Viscous Adjoint Equations for the k−ωSST Turbulence and γ−Reθ Transition Models. *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.* Orlando, Florida. doi:10.2514/6.2011-1247

Khayatzadeh, P., & Nadarajah, S. (2012). Aerodynamic Shape Optimization of Natural Laminar Flow (NLF) Airfoils. *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.* doi:10.2514/6.2012-61

Kim, H., & Rho, O.-H. (1998). Aerodynamic Design of Transonic Wings Using the Target Pressure Optimization Approach. *Journal of Aircraft, 35*(5), 671-677. doi:10.2514/2.2374

Koreanschi, A., Gabor, O. S., Acotto, J., Brianchon, G., Portier, G., Botez, R. M., . . . Mebarki, Y. (2017). Optimization and design of an aircraft's morphing wing-tip demonstrator for drag reduction at low speed, Part I – Aerodynamic optimization using genetic, bee colony and gradient descent algorithms. *Chinese Journal of Aeronautics, 30*(1), 149-163. doi:10.1016/j.cja.2016.12.013

Lamousin, H. J., & Waggenspack, N. N. (1994). NURBS-based free-form deformations. *IEEE Computer Graphics and Applications, 14*(6), 59-65. doi:10.1109/38.329096

Lozano, C. (2018). Singular and Discontinuous Solutions of the Adjoint Euler Equations. *AIAA journal, 56*(11). doi:10.2514/1.J056523

Lynde, M. N., & Campell, R. L. (2016). Expanding the Natural Laminar Flow Boundary for Supersonic Transports. *34th AIAA Applied Aerodynamics Conference.* doi:10.2514/6.2016-4327

Lyu, Z., Kenway, G. K., Paige, C., & Martins, J. R. (2013). Automatic Differentiation Adjoint of the Reynolds-Averaged Navier–Stokes Equations with a Turbulence Model. *21st AIAA Computational Fluid Dynamics Conference.* doi:10.2514/6.2013-2581

Lyu, Z., Xu, Z., & Martins, J. R. (2014). Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization. *The Eighth International Conference on Computational Fluid Dynamics (ICCFD8).* Chengdu, Sichuan, China. doi:10.1.1.664.9175

Mader, C. A., & Martins, J. R. (2014). Computing Stability Derivatives and Their Gradients for Aerodynamic Shape Optimization. *AIAA journal, 52*(11), 2533-2545. doi:10.2514/1.J052922

Mader, C. A., Martins, J. R., & Maki, K. J. (2020). DAFoam: An Open-Source Adjoint Framework for Multidisciplinary Design Optimization with OpenFOAM. *AIAA journal, 58*(3). doi:10.2514/1.J058853

Mader, C. A., Martins, J. R., Alonso, J. J., & van der Weide, E. (2008). ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers. *AIAA journal, 46*(4), 863-878. doi:10.2514/1.29123

Martins, J. R., & Hwang, J. T. (2013). Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models. *AIAA journal, 51*(11), 2582-2599. doi:10.2514/1.J052184

Masatsuka, k. (2020). Retrieved from I do like CFD. Useful books on Computational Fluid Dynamics: http://ossanworld.com/cfdbooks/cfdcodes.html

Mavriplis, D. (2003). Revisiting the Least-squares Procedure for Gradient Reconstruction on Unstructured Meshes. *16th AIAA Computational Fluid Dynamics Conference.* Orlando, Florida. doi:10.2514/6.2003-3986

Mayle, R. E. (1991). The Role of Laminar-Turbulent Transition in Gas Turbine Engines. *Proceedings of the ASME 1991 International Gas Turbine and Aeroengine Congress and Exposition, Volume 5: Manufacturing Materials and Metallurgy; Ceramics; Structures and Dynamics; Controls, Diagnostics and Instrumentation; Education; IGTI Scholar Award; General.* Orlando, Florida. doi:10.1115/91-GT-261

McMasters, J. (1974). Low-Speed Airfoil Bibliography. *Technical Soaring an International Journal, 3*.

Mengze, W., Qi, W., & Tamer, Z. A. (2019). Discrete adjoint of fractional-step incompressible Navier-Stokes solver in curvilinear coordinates and application to data assimilation. *Journal of Computational Physics*. doi:10.1016/j.jcp.2019.06.065

Menter, F. R., Langtry, R. B., Likki, S. R., Suzen, Y. B., Huang, P. G., & Volker, S. (2004). A Correlation-Based Transition Model Using Local Variables: Part I — Model Formulation. *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, (pp. 57-67). Vienna, Austria. doi:10.1115/GT2004-53452

Menter, F. R., Langtry, R. B., Likki, S. R., Suzen, Y. B., Huang, P. G., & Völker, S. (2006). A Correlation-Based Transition Model Using Local Variables—Part I: Model Formulation. *Journal of Turbomachinery, 128*(3), 413-422. doi:10.1115/1.2184352

Meza, J. C., Olivia, R. A., Hough, P. D., & Williams, P. J. (2007). OPT++: An object-oriented toolkit for nonlinear optimization. *ACM Transactions on Mathematical Software, 33*(2), Article 12. doi:10.1145/1236463.1236467

Nadarajah, S., & Jameson, A. (2000). A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *38th Aerospace Sciences Meeting and Exhibit.* Reno, Nevada. doi:10.2514/6.2000-667

Nadarajah, S., & Jameson, A. (2001). Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *15th AIAA Computational Fluid Dynamics Conference.* Anaheim, CA. doi:10.2514/6.2001-2530

Naumann, U. (2011). *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Dİfferentiation.* Philadelphia: SIAM.

Nemili, A., Özkaya, E., Gauger, G. R., Kramer, F., & Thiele, F. (2017). Accurate Discrete Adjoint Approach for Optimal Active Separation Control. *AIAA journal, 55*(9). doi:10.2514/1.J055009

Nielsen, E., & Anderson, K. (2000). Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations. *AIAA journal, 37*. doi:10.2514/2.640

Nishikawa, H. (2019). An Implicit Gradient Method for Cell-Centered Finite-Volume Solver on Unstructured Grids. *AIAA Scitech 2019 Forum.* San Diego, California. doi:10.2514/6.2019-1155

Nishikawa, H., & Kitamura, K. (2008). Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers. *Journal of Computational Physics, 227*(4), 2560-2581. doi:10.1016/j.jcp.2007.11.003

OpenMP Architecture Review Board. (2018). OpenMP Application Programming Interface, Version 5.0. http://www.openmp.org.

Özkaya, E., & Gauger, N. R. (2020). Global Aerodynamic Design Optimization via Primal-Dual Aggregation Method. (A. Dilmann, G. Heller, E. Kramer, C. Wagner, C. Tropea, & S. Jakirlic, Eds.) *New Results in Numerical and Experimental Fluid Mechanics XII. DGLR 2018. Notes on Numerical Fluid*

Mechanics and Multidisciplinary Design, 142. doi:10.1007/978-3-030-25253-3_5

Pironneau, O. (1973). On optimum profiles in Stokes flow. *Journal of Fluid Mechanics, 59*, 117-128. doi:10.1017/S002211207300145X

Pironneau, O. (1974). On optimum design in fluid mechanics. *Journal of Fluid Mechanics, 64*, 97-110. doi:10.1017/S0022112074002023

Radespiel, R., Rossow, C., & Swanson, R. C. (1990). Efficient cell-vertex multigrid scheme for the three-dimensional Navier-Stokes equations. *AIAA journal, 28*(8), 1464-1472. doi:10.2514/3.25239

Rashad, R., & Zingg, D. (2016). Aerodynamic Shape Optimization for Natural Laminar Flow Using a Discrete-Adjoint Approach. *AIAA journal, 54*(11). doi:10.2514/1.J054940

Reuther, J., Jameson, A., Alonso, J., Rimlinger, M., & Sauders, D. (1997). Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers. *35th Aerospace Sciences Meeting & Exhibit.* Reno. doi:10.2514/6.1997-103

Reuther, J., Jameson, A., Farmer, J., Martinelli, L., & Saunders, D. (1996). Aerodynamic shape optimization of complex aircraft configurations via an adjoint formulation. *Aerospace Sciences Meeting and Exhibit.* Reno. doi:10.2514/6.1996-94

Robinson, D. F., & Hassan, H. A. (1997). Modeling turbulence without damping functions using k-zeta model. *15th Applied Aerodynamics Conference,* (pp. 751-760). doi:10.2514/6.1997-2312

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 357-372. doi:10.1016/0021-9991(81)90128-5

Rudnik, R. (1997). *Untersuchung der Leistungsfähigkeit von Zweigleichungs-Turbulenzmodellen bei Profilumströmungen.* Dissertation. DLR-Forschungsbericht 97-49.

Saad, Y., & Schultz, M. H. (1986). GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing, 7*(3), 856–869. doi:10.1137/0907058

Sagebaum, M., Albring, T., & Gauger, N. R. (2019). High-Performance Derivative Computations using CoDiPack. *ACM Transactions on Mathematical Software, 45*(4), 1-26. doi:10.1145/3356900

Samareh, J. (2004). Aerodynamic Shape Optimization Based on Free-Form Deformation. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* Albany, New York. doi:10.2514/6.2004-4630

Savill, A. M. (1993). Some recent progress in the turbulence modelling of by-pass transition. In R. M. SO, C. G. Speziale, & B. Launder (Eds.), *Near-Wall Turbulent Flows* (p. 829). Amsterdam, The Netherlands: Elsevier Science.

Schmitt, V., & Charpin, F. (1979). Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers. *Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04.* AGARD AR-138.

Schroeder, W. J. (Ed.). (2010). *The VTK's User's Guide* (11th ed.). Kitware, Inc.

Schroeder, W. J., Martin, B., & Lorensen, B. (2006). *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics* (4th ed.). Kitware, Inc.

Schubauer, G. B., & Klebanoff, P. S. (1956). *Contributions on the mechanics of boundary-layer transition.* NACA-TR-1289, NACA, Washington, DC.

Sederberg, T. W., & Parry, S. R. (1986). Free-form deformation of solid geometric models. *13st Annual Conference on Computer Graphics and Interactive*

Techniques, SIGGRAPH 1986, 20(4), pp. 151-160. doi:10.1145/15886.15903

Seitz, A., Hübner, A., & Risse, K. (2019). The DLR TuLam project: design of a short and medium range transport aircraft with forward swept NLF wing. *CEAS Aeronautical Journal*. doi:10.1007/s13272-019-00421-1

Seitz, A., Kruse, M., Wunderlich, T. F., & Bold, J. (2011). The DLR Project LamAiR: Design of an NLF Forward Swept Wing for Short and Medium Range Transport Application. *29th AIAA Applied Aerodynamics Conference.* Honolulu, Hawaii. doi:10.2514/6.2011-3526

Shanno, D. F. (1980). On Variable-Metric Methods for Sparse Hessians. *Mathematics of Computation, 34*(150), 499-514. doi:10.2307/2006098

Shei, L., & Wang, Z. J. (2016). Adjoint-based Mesh Adaptation for the 3D Navier-Stokes Equations with the High-order CPR Method. *54th AIAA Aerospace Sciences Meeting.* San Diego, California. doi:10.2514/6.2016-1829

Shima, E., Kitamura, K., & Haga, T. (2013). Green–Gauss/Weighted-Least-Squares Hybrid Gradient Reconstruction for Arbitrary Polyhedra Unstructured Grids. *AIAA journal, 51*(11). doi:10.2514/1.J052095

Shiriaev, D., & Griewank, A. (1996). ADOL-F Automatic Differentiation of Fortran Codes. In *Computational Differentiation: Techniques, Applications, and Tools* (pp. 375-384). SIAM.

Slater, J. W., Dudek, J. C., & Kenneth, E. T. (2000). *The NPARC Alliance Verification and Validation Archive.* NASA/TM-2000-209946, NASA.

Smith, A. M., & Gamberoni, N. (1956). *Transition, Pressure Gradient, and Stability Theory.* ES-26388, Douglas Aircraft Company, Engineering Department.

Somers, D. M. (1981). *Design and Experimental Results for a Natural-Laminar-Flow Airfoil for General Aviation Applications.* NASA-TP-1861, NASA Langley Research Center, Hampton, Virginia.

Sozer, E., Brehm, C., & Kiris, C. C. (2014). Gradient Calculation Methods on Arbitrary Polyhedral Unstructured Meshes for Cell-Centered CFD Solvers. *AIAA 52nd Aerospace Sciences Meeting.* National Harbor, Maryland. doi:10.2514/6.2014-1440

Spalart, P., & Allmaras, S. (1992). A One-Equation Turbulence Model for Aerodynamic Flows. *30th Aerospace Sciences Meeting and Exhibit.* Reno, Nevada. doi:10.2514/6.1992-439

Straka, C. W. (2005). ADF95: Tool for automatic differentiation of a Fortran code designed for large numbers of independent variables. *Computer Physics Communications, 168*(2), 123-139. doi:10.1016/j.cpc.2005.01.011

Suzen, Y. B., & Huang, P. G. (2000). Modeling of Flow Transition Using an Intermittency Transport Equation. *ASME. Journal of Fluids Engineering, 122*(2), 273-284. doi:10.1115/1.483255

Syrakos, A., Varchanis, S., Dimakopoulos, Y., Goulas, A., & Tsamopoulos, J. (2017). *On the order of accuracy of the divergence theorem (Green-Gauss) method for calculating the gradient in finite volume methods.* hal-01532882v2. doi:10.1063/1.4997682

Thomas, J. P., Hall, K. C., & Dowell, E. H. (2005). Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities. *AIAA journal, 43*(9), 1931-1936. doi:10.2514/1.731

Toro, E. F. (2009). *Riemann Solvers and Numerical Methods for Fluid Dynamics, A practical Introduction* (3rd ed.). Springer. doi:10.1007/978-3-540-49834-6

Utke, J. (2004). *OpenAD: Algorithm implementation user guide.* Technical Memorandum 274, Argonne National Laboratory, Argonne, IL.

Utke, J., Naumann, U., Fagan, M., Tallent, N., Strout, M., Heimbach, P., . . . Wunsch, C. (Eds.). (2008). OpenAD/F: A Modular Open-Source Tool for Automatic

Differentiation of Fortran Codes. *ACM Transactions on Mathematical Software, 34*(4), 1-36. doi:10.1145/1377596.1377598

van Ingen, J. L. (1956). *A Suggested Semi-empirical Method for the Calculation of the Boundary Layer Transition Region.* Report V.T.H.-74, Delft, Netherland.

Venkatakrishnan, V. (1993). On the accuracy of limiters and convergence to steady state solutions. *31st AIAA, Aerospace Sciences Meeting and Exhibit.* Reno, Nevada. doi:10.2514/6.1993-880

Venkatakrishnan, V. (1995). Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters. *Journal of Computational Physics, 118*(1), 120-130. doi:10.1006/jcph.1995.1084

Vijayan, P., & Kallinderis, Y. (1994). A 3D Finite-Volume Scheme for the Euler Equations on Adaptive Tetrahedral Grids. *Journal of Computational Physics, 113*(2), 249-267. doi:10.1006/jcph.1994.1133

Viken, J. K., Watson-Viken, S. A., Pfenninger, W., Morgan, H. L., & Campell, R. L. (1987). Design of the low-speed NLF(1)-0414F and the high-speed HSNLF(1)-0213 airfoils with high-lift systems. *Research in Natural Laminar Flow and Laminar-Flow Control.*

Walters, D. K., & Cokljat, D. (2008). A Three-Equation Eddy-Viscosity Model for Reynolds-Averaged Navier–Stokes Simulations of Transitional Flow. *Journal of Fluids Engineering, 130*(12). doi:10.1115/1.2979230

Walters, D. K., & Leylek, J. H. (2004). A New Model for Boundary-Layer Transition Using a Single-Point RANS Approach. *Journal of Turbomachinery, 126*(1), 193-202. doi:10.1115/1.1622709

Warren, E. S., & Hassan, H. A. (1998). Transition Closure Model for Predicting Transition Onset. *Journal of Aircraft, 35*(5), 769-775. doi:10.2514/2.2368

Warren, E. W., & Hassan, H. A. (1998). Alternative to the e^n Method for Determining Onset of Transition. *AIAA journal*, *36*(1), 111-113. doi:10.2514/2.361

Wauters, J., Degroote, J., & Vierendeels, J. (2019). Comparative Study of Transition Models for High-Angle-of-Attack Behavior. *AIAA journal*, 1-16. doi:10.2514/1.J057249

Yang, Z., & Mavriplis, D. J. (2019). Discrete Adjoint Formulation for Turbulent Flow Problems with Transition Modelling on Unstructured Meshes. *AIAA Scitech 2019 Forum.* doi:10.2514/6.2019-0294

Yu, Y., Lyu, Z., Xu, Z., & Martins, J. R. (2018). On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Journal of Aerospace Science and Technology*, 183-199. doi:10.1016/j.ast.2018.01.016

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Kaya, Halil

EDUCATION

Degree	**Institution**
MS	METU Aerospace Engineering
BS	METU Aerospace Engineering
High School	Turgutlu Halil Kale Fen Lisesi, Manisa

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2008-Present	Turkish Aerospace Industries	Aerodynamics Engineer

FOREIGN LANGUAGES

Advanced English, Elementary German

PUBLICATIONS

1. Haser, S., A., Kaya H. ve Uzol O. (2014). Küt Firar Kenarlı Kanat Kesitlerinin Girdap Parçacık Yöntemi ile Aerodinamik Analizi. *V. Ulusal Havacılık ve Uzay Konferansı,* Kayseri.

2. Alayli, H., Kaya H. and Haser, S. A. (2017). Wing Design Optimization of a MALE Unmanned Aerial Vehicle. *9th Ankara International Aerospace Conference*, Ankara.

3. Kaya, H., Tiftikçi, H. ve Tuncer, İ. H. (2018). Otomatik Türev Araçları ile Ayrık Adjoint Çözücü Geliştirilmesi. *VII. Ulusal Havacılık ve Uzay Konferansı,* Samsun

4. Karakoc, A. ve Kaya H. (2018). A Multi-Objective Multi-Disciplinary Optimization Approach for NATO AVT 251 UCAV-MULDICON, *2018 Applied Aerodynamics Conference,* Atlanta, Georgia. doi: 10.2514/6.2018-3001

5. Kaya, H., Tiftikci H., Kutluay Ü. and Sakarya E. (2018). A Multi-Fidelity Aerodynamic Modeling Approach for NATO AVT 251UCAV-MULDICON, *2018 Applied Aerodynamics Conference,* Atlanta, Georgia. doi: 10.2514/6.2018-3000

6. Kaya H., Tuncer, İ. H. ve Tiftikci, H. (2019). Development of a 2D Discrete Adjoint Solver using Automatic Differentiation. *10th Ankara International Aerospace Conference,* Ankara.

7. Kaya, H., Tiftikci H., Kutluay Ü. and Sakarya E. (2019). Generation of Surrogate-based Aerodynamic Model of a UCAV Configuration using an Adaptive Co-Kriging Method. *Journal of Aerospace Science and Technology,* 95 doi: 10.1016/j.ast.2019.105511