National Technical University of Athens

School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

Software Development for the Shape Optimization of Airfoils using the Continuous Adjoint Method

Diploma Thesis
by

Dimitrios Dimos

Advisor:
Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2021

Acknowledgements

At this stage I would like to thank all the people that contributed and helped me to complete this dimploma Thesis.

First and foremost, I would like to pay my deep sense of graditude to my parents, Christos and Eugenia, for their continuous support all these years. Their contribution during my school and student years has been decisive. I thank them for all kind of sacrifices they made for me and for the fact that they were always there to help me.

I would also like to express my special thanks of gratitude to my professor K.C. Giannakoglou. I would like to thank him for being a great professor, as well as the most inspiring figure throughout all of my studies. I would also like to thank him for believing in me and for assigning me this very interesting project. I consider myself extremely fortunate to have been working under his supervision and I deeply acknowledge that none of this work would have been possible without his help, guidance and support.

Last but not least, I would also like to thank Dr. V. Asouti and Dr. X. Trompoukis of the research team of PCOpt/NTUA for always being eager to help when needed. I acknowledge that they spent time and effort to assist me and they where always happy to share some of their knowledge.

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν στην υλοποίηση της παρούσας διπλωματικής εργασίας.

Καταρχάς, θα ήθελα να εκφράσω την ευγνομωσύνη μου προς τους γονείς μου, Χρήστο και Ευγενία, για τη συνεχή στήριξη που μου δείξανε όλα αυτά τα χρόνια. Η συμβολή τους κατα τη διάρκεια των μαθητικών και φοιτητικών μου χρόνων υπήρξε μεγάλης σημασίας. Τους ευχαριστώ για όλες τις θυσίες που έκαναν για μένα και για το γεγονός ότι ήταν πάντα στο πλευρό μου.

Ακόμα θα ήθελα να πω ένα μεγάλο ευχαριστώ στον καθηγητή μου Κ.Χ. Γιαννάκογλου. Υπήρξε εξαιρετικός καθηγητής και πηγή έμπνευσης καθόλη τη διάρκεια των σπουδών μου. Τον ευχαριστώ επίσης για την εμπιστοσύνη που μου έδειξε και για την ανάθεση ενός τόσο ενδιαφέρον θέματος. Νιώθω πραγματικά πολύ τυχερός που δούλεψα υπό την καθοδήγησή του και αναγνωρίζω ότι η παρούσα εργασία δεν θα ήταν εφικτή χωρίς τη βοήθεια και στήριξή του.

Τέλος, ευχαριστώ βαθύτατα τη Δρ. Β. Ασούτη και τον Δρ. Ξ. Τρομπούκη της ερευνητικής ομάδας της ΜΠΥΡ&Β/ΕΜΠ, καθώς υπήρξαν πάντα πρόθυμοι να με βοηθήσουν όποτε τους χρειαζόμουν. Αναγνωρίζω ότι αφιέρωσαν χρόνο και κόπο και ήταν πάντα διατεθιμένοι να μοιραστούν τις γνώσεις τους μαζί μου.

National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Parallel CFD & Optimization Unit

Software Development for the Shape Optimization of Airfoils using the Continuous Adjoint Method

Diploma Thesis

Dimitrios Dimos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2021

Abstract

This dimploma thesis aims at the programming of a Gradient-Based shape optimization software for airfoils using the continuous adjoint method, in its Enchanced Surface Integral, E-SI, form, as already proposed by the Parallel CFD & Optimization Unit (PCOpt) of NTUA. The flow is simulated by solving the Euler equations on an unstructured grid, based on finite volume vertex-centered approach. After the engineer defines the objective function, the adjoint field can be computed similarly to the flow-field itself. Thus, the software is able to compute the sensitivity derivatives with respect to the design variables parameterizing the shape of the airfoil. Using these derivatives, the design variables are iteratively updated in order to improve the objective function's value by changing the airfoil shape.

Furthermore, a parametric analysis has been made with the goal of finding the most efficient and accurate parameter set for the optimization proccess. Different numerical schemes, such as Flux Vector Splitting and Roe's Approximate Riemann Solver in conservative and non-conservative form have been applied. Moreover, first and second order schemes have been implemented by also implementing the Van-Leer Van-Albada Limiter. Additionally, different grid densities have been used in order to investigate their effect. Last but not least, the continous adjoint (Severed) SI and Enchanced SI (E-SI) methods are compared with respect to the accuracy of the shape's sensitivity derivatives. This parametric analysis has been applied for three different airfoils, NACA 4412, RAE 2822 and FAUVEL for different free-stream conditions.

As a next step, the software is evaluated by comparing the adjoint sensitivity deriva-

tives with those resulting from the second-order Finite Difference scheme. Then, a proper parameter set is extracted that gives accurate and cost-efficient results. On the account of that, objective functions have been chosen, concerning the increase of lift coefficient for airfoil optimizations.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Ανάπτυξη λογισμικού για Βελτιστοποίηση Γεωμετρίας Αεροτομών με τη Συνεχή Συζυγή Μέθοδο

Διπλωματική εργασία

Δημήτριος Δήμος

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2021

Η διπλωματική αυτή εργασία αποσκοπεί στη δημιουργία ενός λογισμικού αιτιοκρατικής βελτιστοποίησης με τη συνεχή συζυγή μέθοδο διατυπωμένη σύμφωνα με την E-SI τεχνική που έχει προταθεί από την ερευνητική ομάδα της Μονάδας Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης του Εθνικού Μετσόβιου Πολυτεχνείου. Η πρόλεξη της ροής υλοποιείται με την αριθμητική επίλυση των εξισώσεων Euler σε μη δομημένο πλέγμα και τη κεντρο-κομβική διατύπωση πεπερασμένων όγκων. Αφού ο μηχανικός επιλέξει τη συνάρτηση-στόχο, πραγματοποιείται ο υπολογισμός του συζυγούς πεδίου με τρόπο παρόμοιο με αυτόν της πρόλεξης της ροής. Έτσι, το λογισμικό υπολογίζει τις παραγώγους ευαισθησίας ως προς τις μεταβλητές σχεδιασμού της αεροτομής. Με τη χρήση των παραγώγων αυτών, οι μεταβλητές σχεδιασμού μεταβάλλουν τις τιμές τους, ώστε να βελτιωθεί η τιμή της συνάρτησης-στόχου. Έτσι, αλλάζει και το σχήμα της αεροτομής.

Επίσης, πραγματοποιείται παραμετρική ανάλυση με σκοπό την εύρεση των κατάλληλων ρυθμίσεων για τη μείωση του κόστους και αύξηση της ποιότητας των αποτελεσμάτων κατά τη βελτιστοποίηση. Χρησιμοποιούνται διαφορετικά αριθμητικά σχήματα επίλυσης της φυσικής και της συζυγούς ροής, όπως το Flux Vector Splitting και ο επιλύτης Roe's Approximate Riemann σε συντηρητική και μη-συντηρητική μορφή. Ακόμα, συγκρίνεται η χρήση πρώτης και δεύτερης τάξης ακριβεια σχημάτων διακριτοποίησης, καθώς και ο περιοριστής Van-Leer Van-Albada. Επιπλέον, εξετάζεται η επίδραση της πυκνότητας του πλέγματος, ώστε να γίνει αντιληπτή η επίδραση της στα αποτελέσματα. Μια ακόμα παράμετρος που μελετάται είναι η χρήση διαφόρων μεθόδων της συζυγούς συνάρτησης. Συγκεκριμένα τα αποτελέσματα του E-SI συγκρίνονται με εκείνα το Severed SI. Η παραμετρική αυτή ανάλυση εφαρμόζεται σε τρεις αεροτομές, τη NACA 4412, τη RAE 2822 και τη FAUVEL, σε διαφορετικές συνθήκες αδιατάρακτης ροής, έτσι ώστε τα συμπεράσματα να είναι πιο γενικά.

Σε επόμενο βήμα, πιστοποιείται η ακρίβεια του λογισμικού, συγκρίνοντας τις παραγώγους ευαισθησίας με αυτές που προκύπτουν από τις δεύτερης τάξης ακρίβειας Πεπερασμένες Διαφορές. Καθορίζονται κατάλληλες ρυθμίσεις, ωστε ο κώδικας να παράγει αξιόπιστα και οικονομικά αποτελέσματα. Τέλος, επιλέγονται συναρτήσεις-στόχοι που σχετίζονται με την αύξηση του συντελεστή άνωσης σε τρεις αεροτομές.

Acronyms

CFD	Computational Fluid Dynamics
NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization unit
GBM	Gradient Based Method
CPs	Control Points
FVS	Flux Vector Splitting
SDs	Sensitivity Derivatives
FAE	Field Adjoint Equations
ABC	Adjoint Boundary Conditions
PDE	Partial Differential Equation
gdPDE	grid displacement Partial Differential Equation
RHS	Right Hand Side
E-SI	Enchanced SI
Sev-SI	Severed / Standard SI
(Non-)Cons	(Non-)Conservative
w.r.t.	with respect to

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΕΘΣ	Εργαστήριο Θερμικών Στροβιλομηχανών
ΜΠΥΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
ΜΔΕ	Μερική Διαφορική εξίσωση

Contents

Chapter 1

Introduction

The design process of the airfoils has changed significantly throughout the years . Except for the experiments, other methods have been developed to evaluate the flow around lifting bodies, mainly based on *Prandtl's lifting-line theory* [1]. However, methods like these show serious limitations and cannot be used on high Mach number and compressible flows. The improvement of the computational power and developement of numerical schemes made the arithmetic approximation of the *Navier-Stokes Equations* feasible via Computational Fluid Dynamics (CFD). Hence, the cost-expensive experiments have been reduced and the flow analysis can cover almost every case with higher levels of accuracy.

1.1 Shape Optimization

In the engineering sector, the goal is not limited to producing a working design, but to find an optimized one. Thus, optimization is of utmost importance in every scientific field. There are a lot of techniques that have been developed. Some of them rely on modeling, experience and human judgement. However, in complex cases with many and conflicting variables one cannot intuitively determine the optimal design. Because of this, computer-based optimization softwares have been developed and applied. Coded properly, it is possible for computers to evaluate a product, solving the primal problem (Model Analysis), and redesign it until the desirable result arises.

Computer-based optimization can be divided into two main categories: Stochastic [2] and deterministic methods [3, 4, 5]. The first ones are based on randomized search. They are able to converge on the global optimum even in multi-objective problems but they usually require a larger computational budget. The *Genetic* (GA) and *Evolutionary* (EA) algorithms [6] are the most common that both fall into this category and mimic the natural species evolution as formulated by Darwin

[7]. Their existence is only broached for completeness reasons, as this diploma thesis only focuses on the deterministic or Gradient-Based Optimization methods(GB).

In mechanical engineering, the main target is to find the best geometry that fits the problem. On account of that, shape optimization is considered of great importance, which raises the need for body parameterization.

1.1.1 Shape Parameterization

In order to parameterize a geometry, a set of features must be defined so as to control its shape. For example, for a Bezier Curve, [8] these can be the Control Points (CPs). Some of them may be fixed in order to preserve a set of the shape's characteristics, while the others be free to get displaced if needed as shown in Figure 1.1. The coordinates of the latter ones are referred to as the design variables b_n , $n = 1, \cdots, N$, where N is their number. In a similar way, design variables (b_n) result from each interpolation method.

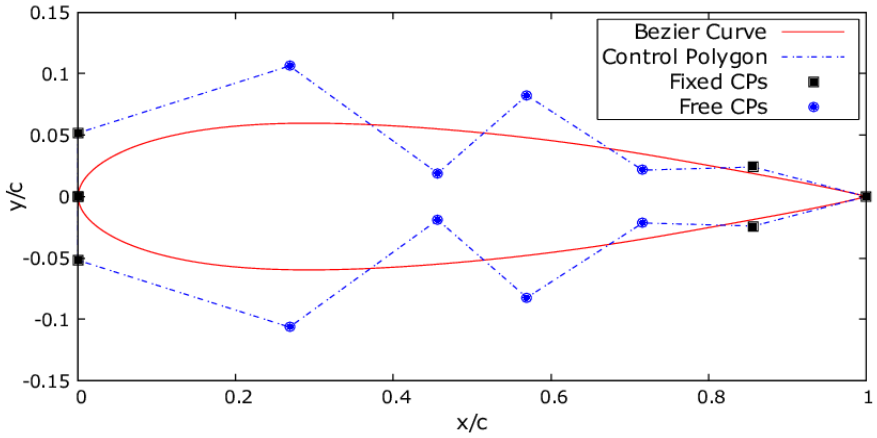


Figure 1.1: *Parameterization of an Airfoil using Bezier Curve.*

In order to optimize the shape of a geometry, an objective function (F) must be defined. The goal is to find the values of b_n variables which maximize or minimize

$$F = F(b_1, ..., b_n, ..., b_N) \tag{1.1}$$

1.1.2 Gradient - Based (GB) Methods

As its name indicates, GBO computes the derivatives of the objective function F w.r.t. the design variables $\delta F/\delta b_n$, also known as sensitivity derivatives (SDs). These can reveal how the function F changes if each design variable slightly increases or

decreases. This information is one of great importance as the algorithm can automatically change the design in order to come up with a new shape in each optimization cycle which minimizes or maximizes F. In contrast to stochastic methods, GBO may converge to a local optimum. However, they are less expensive.

There are several ways to compute the SDs. Each one of them affects significantly the efficiency of the GBO. The most common and simple is the *Finite Difference* method (FD) [4]. For a second-order FD scheme assuming that ϵ is an infinitesimally small quantity

$$\frac{\delta F}{\delta b_n} = \frac{F(b_1, ..., b_n + \epsilon, ..., b_N) - F(b_1, ..., b_n - \epsilon, ..., b_N)}{2\epsilon} \tag{1.2}$$

This method has two serious drawbacks. First of all, in order to compute each SD, F must be computed twice, meaning that the primal problem must be solved twice, too. Thus, the computational cost of FD can be estimated as $2N$ times the cost of primal problem, which is highly uneconomic for large scale optimization. Moreover, the SDs are affected by the value of ϵ. Although $\epsilon \to 0$, this is set, in numerical computations, as a small number. However, if it gets bigger than it should, arithmetic deviation occurs. On the other hand, setting it too small may lead to extremely small difference between the two terms in the numerator of eq. 1.2, practically zero due to round-off. This results to the degeneration of the computation. A fully-converged primal solution may correct the problem, but it has higher computational costs. The application of this method is likely to lead to the computation of derivatives with high uncertainty degree. There is an ϵ-space where each SD is independent from the value of ϵ and its estimation is considered acceptable, but in order to find it, the cost is extremely high, as the same derivative must be computed with a variety of ϵ numbers.

An alternative method for the computation of the sensitivity derivatives is the *Complex Variable* (CV) [5] method. According to this

$$\frac{\delta F}{\delta b_n} = \frac{Im[F(b_1, ..., b_n + i\epsilon, ..., b_N)]}{\epsilon} \tag{1.3}$$

where $i = \sqrt{-1}$, $\epsilon \to 0$ and Im is the imaginary part of the complex function F. Because the numerator of eq. 1.3 has only one term, the SDs are independent from the value of ϵ, solving the second drawback of the FD method. Nevertheless, in order to find all SDs, it requires N times the cost of the sollution of the primal problem. Half the cost of FD, but still a linear cost increase with the number N. Finally, modification on the existing software should be applied in order to read and calculate complex variables.

Another way to compute SDs is the *Direct Differentiation* (DD) [4]. Via this method, the primal equations are differentiated w.r.t. b_n and N linear systems arise. They are

solved and, then, the SDs' computation is straightforward, since all the necessary terms are given. The cost still scales with N. Moreover, the development of a new solver for the differentiated equations is needed.

The major disadvantage of the previous methods is the linearly increasing cost w.r.t. N. As a result, an optimization with a large amount of design variables would be extremely expensive.

1.1.2.1 The Adjoint Method

The adjoint method has a unique feature that makes it perfect for large-scale optimization problems [9, 10, 11]. It is independent from the amount of design variables N. In order not to compute the derivatives of the flow variables w.r.t. the design variables (DD method), a Lagrangian multiplier, also known as adjoint (Ψ), is introduced. Adding the volume integrals of the residuals of the primal equations multiplied by Ψ to the objective function F, an augmented objective function is defined (F_{AUG}). Since the residuals of the primal equations equal to zero, $F = F_{AUG}$. Finding the SDs of F_{AUG}, which equals to those of F, and setting Ψ in a way that the summation of the terms that contain the derivatives of the primal variables w.r.t. the design variables equal to zero, a new form of SDs arises. SDs that are only dependent on the known primal field, easily computed shape parameterization variables and the adjoint field. The latter has to be computed. Its equations are similar to the primal ones and approximately so is their cost. Thus, although a new software must be developed, the cost to compute SDs is N-independent. Each optimization cycle costs almost as much as two primal problem solutions.

There are two ways to extract the Field Adjoint Equations (FAE) and the Adjoint Boundary Conditions (ABC). Discrete adjoint, where the already discretized primal equations are multiplied with Ψ, extracting the discretized FAE and ABC directly and the continuous one, where the adjoint PDEs analytically derive from the differentiation of F_{AUG} and then discretized. On a general framework both of them can generate accurate SDs. Normally discrete adjoint has greater precission, but higher cost than the continuous. However, as the mesh size increases, the difference between the continuous and discrete gradient reduce [12].

In the literature there are two main formulations of the continuous adjoint. Both end up in the same FAE and ABC [13, 14]. Their difference lies on the final expression of SDs. The first published formulation [9] contains Field Integrals (FI) of grid SDs, i.e. derivatives of the grid nodes' coordinates w.r.t. the design variables. Using a second order FD scheme for their computation results in two grid displacement PDEs solutions for every derivative, which results the cost to scale linearly with the number of design variables. The second formulation [15] ends up in SDs with only Surface Integral (SI) terms. Thus, this method's cost is significantly lower than the FI's, especially in large-scale optimization problems. However, if the grid is not fine enough, SDs might not be accurate. In an attempt to preserve SI adjoint low cost

and FI adjoint precision, the Enhanced-SI (E-SI) adjoint has been developed [16]. FAE and ABC remain exactly the same and although the SDs consist only from surface integrals, their expression is different from the SI adjoint formulation.

In the following chapters the continuous SI and E-SI adjoint approaches will be thoroughly analyzed on an airfoil optimization problem and the main difference between the formulations will be further discussed.

1.2 The Necessity of a new Software

The research team of PCOpt/NTUA has already developed and uses two softwares for shape optimization. The first one is based on the OpenFOAM. However, its primal solver can only evaluate incompressible flows. For cases concerning compressible flows the second software, named PUMA, is used. PUMA is programmed on GPUs. Thus, it cannot be parallelized and has serious memory limitations.

For that reason, a new software programmed on CPUs is needed. More than a decade ago, a FORTRAN 77 code had been developed by PCOpt/NTUA for the computation of 2D, inviscid and compressible flow fields. Moreover, the SI adjoint method was implemented for the computation of SDs. However, the software was never tested and abandoned before E-SI adjoint was first proposed by PCOpt/NTUA [17]. Main purpose of this diploma thesis is to program E-SI adjoint on the existing code, correct its mistakes and test it on airfoil shape optimization cases.

1.3 Thesis Outline

The thesis consists of 6 chapters, including the introduction and conclusions. They are summarized below.

In Chapter 2, the 2D Euler equations are discretized and numerically solved. The Roe and FVS scheme is used on FV unstructured grid.

In Chapter 3, the airfoil optimization algorithm is formulated. At first it gets parameterized with a Bezier curve and the objective function is defined. Concerning the continuous E-SI adjoint method, there is a step-by-step analysis for its formulation, solution and discretization of FAE, ABC and SDs. Finally, the optimization flowchart is presented.

In Chapter 4, the parameters of the numerical schemes used to solve the primal and adjoint equations, are analysed and compared on three different airfoils. The main goal of this chapter is to determine the most efficient and accurate parameter set for the optimization algorithm.

In Chapter 5, the developed software, programed in FORTRAN 77, is used in order to increase the lift of three airfoils.

Chapter 2

CFD Analysis

In this chapter, the discretization and the numerical solution of the 2D Euler equations is presented.

2.1 Flow Equations

The PDEs in conservative form governing the flow of a 2D, inviscid and steady flow of a compressible fluid [18, 19] can be written as

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{f}_i}{\partial x_i} = 0 \tag{2.1}$$

where
$$\vec{U} = \begin{bmatrix} \rho \\ \rho\vec{u} \\ E_t \end{bmatrix}, \qquad \vec{f}_i = \begin{bmatrix} \rho u_i \\ \rho u_i \vec{u} + \vec{\delta}_i p \\ (E_t + p)u_i \end{bmatrix} \tag{2.2}$$

Eq. 2.1 is numerically solved with the use of pseudotime t. The variables that appear in eq. 2.2 are the density ρ, the velocity vector $\vec{u}$, the total energy per unit volume $E_t = \rho E = \dfrac{P}{\gamma - 1} + \dfrac{1}{2}\rho \left|\vec{u}\right|^2$ and the static pressure p. For $A_i = \dfrac{\partial \vec{f}_i}{\partial \vec{U}}$ eq. 2.1 is transformed in

$$\frac{\partial \vec{U}}{\partial t} + A_i \frac{\partial \vec{U}}{\partial x_i} = 0 \tag{2.3}$$

For the non-conservative formulation, the non-conservative variable vector $\vec{V}$ is defined

$$\vec{V} = \begin{bmatrix} \rho \\ u_i \\ p \end{bmatrix} \tag{2.4}$$

and for $M = \dfrac{\partial \vec{U}}{\partial \vec{V}}$ and $\overline{A_i} = M^{-1} A_i M$

$$\frac{\partial \vec{V}}{\partial t} + \overline{A_i} \frac{\partial \vec{V}}{\partial x_i} = 0 \tag{2.5}$$

In 2D cases ($i = 1, 2$)

$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \vec{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix},$$

$$\vec{f} = \begin{bmatrix} \vec{f_1} \\ \vec{f_2} \end{bmatrix} = \begin{bmatrix} \vec{f} \\ \vec{g} \end{bmatrix}, \qquad A_i = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix}$$

$$\vec{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{bmatrix}, \qquad \vec{g} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{bmatrix}$$

$$A = \frac{\partial \vec{f}}{\partial \vec{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2 + \frac{\gamma-1}{2}v^2 & (3-\gamma)u & -(\gamma-1)v & \gamma - 1 \\ -uv & v & u & 0 \\ -\gamma u \rho E + (\gamma-1)u(u^2+v^2) & \gamma\rho E - \frac{\gamma-1}{2}(v^2+3u^2) & -(\gamma-1)uv & \gamma u \end{bmatrix},$$

$$B = \frac{\partial \vec{g}}{\partial \vec{U}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma-3}{2}v^2 + \frac{\gamma-1}{2}u^2 & -(\gamma-1)u & (3-\gamma)v & \gamma - 1 \\ -\gamma v \rho E + (\gamma-1)v(u^2+v^2) & -(\gamma-1)uv & \gamma\rho E - \frac{\gamma-1}{2}(u^2+3v^2) & \gamma v \end{bmatrix} \tag{2.6}$$

Based on eqs. 2.1 to 2.5

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{f}}{\partial x} + \frac{\partial \vec{g}}{\partial y} = 0 \Longrightarrow$$

$$\frac{\partial \vec{U}}{\partial t} + A\frac{\partial \vec{U}}{\partial x} + B\frac{\partial \vec{U}}{\partial y} = 0 \tag{2.7}$$

or, in the conservative form,

$$\frac{\partial \vec{V}}{\partial t} + \overline{A}\frac{\partial \vec{V}}{\partial x} + \overline{B}\frac{\partial \vec{V}}{\partial y} = 0 \tag{2.8}$$

2.1.1 Eigenvalues and Eigenvectors of the flux Jacobian A and $\overline{A}$

Eigenvalues and eigenvectors reveal how the information is travelling through the computational field, i.e. its velocity and direction. Via the direct correlation of eq. 2.7 and 2.8 through $\overline{A_i} = M^{-1}A_iM$, both equations' eigenvalues are the same. For simplicity, they are computed by eq. 2.8.

$$det\left|\lambda_j I - \overline{\mathscr{A}}\right| = 0 \tag{2.9}$$

where $\overline{\mathscr{A}} = \overline{A_i}n_i$ and λ_j the case's eigenvalues. n_i are considered the components of the unitary vector ($\vec{n}$) right to surface of each control volume. For 2D fluxes $i = 1, 2$ and $j = 1, 2, 3, 4$. Therefore, $\vec{n} = (n_1, n_2) = (n_x, n_y)$. From the solution of eq. 2.9

$$\begin{aligned} \lambda_{1,2} &= \vec{u}\vec{n} \\ \lambda_{3,4} &= \vec{u}\vec{n} \pm \vec{c}|\vec{n}| \end{aligned} \tag{2.10}$$

where $c = \sqrt{\frac{\gamma(\gamma-1)(-u^2+2E)}{2}}$ the speed of sound. Finally, the matrices of the right P and the left P^{-1} eigenvectors of the $\mathscr{A} = A_in_i$ matrix are defined. In combination with

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix} \equiv \begin{bmatrix} \vec{u}\vec{n} & 0 & 0 & 0 \\ 0 & \vec{u}\vec{n} & 0 & 0 \\ 0 & 0 & \vec{u}\vec{n} + \vec{c}|\vec{n}| & 0 \\ 0 & 0 & 0 & \vec{u}\vec{n} - \vec{c}|\vec{n}| \end{bmatrix} \tag{2.11}$$

they diagonize $\mathscr{A}$

$$\mathscr{A} = P\Lambda P^{-1} \tag{2.12}$$

2.1.2 Boundary Conditions

For the analysis of the isolated 2D airfoil with the Euler equations two boundary conditions must be imposed. The no-penetration ($\vec{u} \cdot \vec{n} = 0$) on the airfoil's surface

(S_W) and the undisturbed flow at the farfield (S_{inf}). (Figure 2.1)

2.2 Discretization

For the 2D analysis, the grid that is used is unstructured, based on the FV vertex-centered approach [20] and consists of triagonal and quadrilateral elements. Around each node P, a control volume Ω_P is defined by connecting the middle points of each segment and the cell's barycenter with a closed polyline. Vertical to this line, the vector $\vec{n}$ is constructed. Finally, each node connected to P via a grid line is symbolized as Q. Figure 2.2 is a schematic representation of the above and Figure 2.1 shows the meshed computational field.

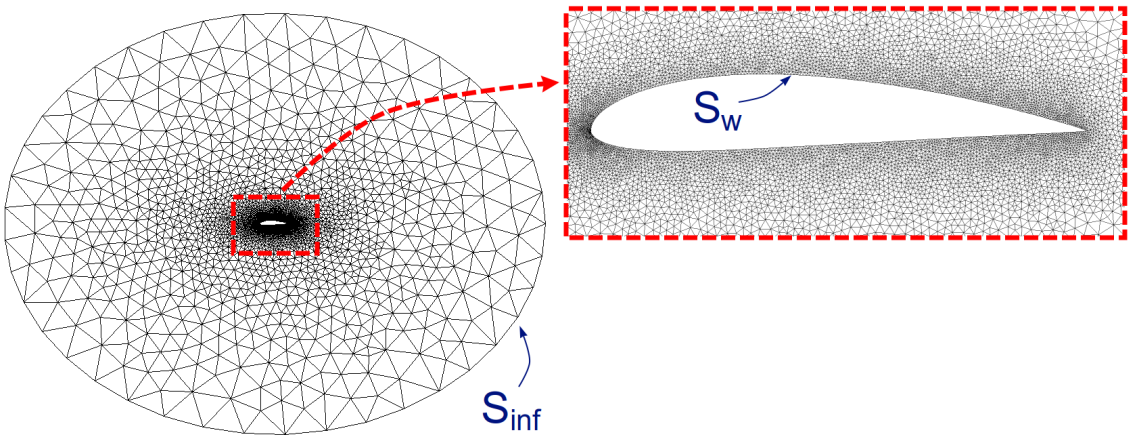


Figure 2.1: *Computational Grid around an isolated airfoil.*

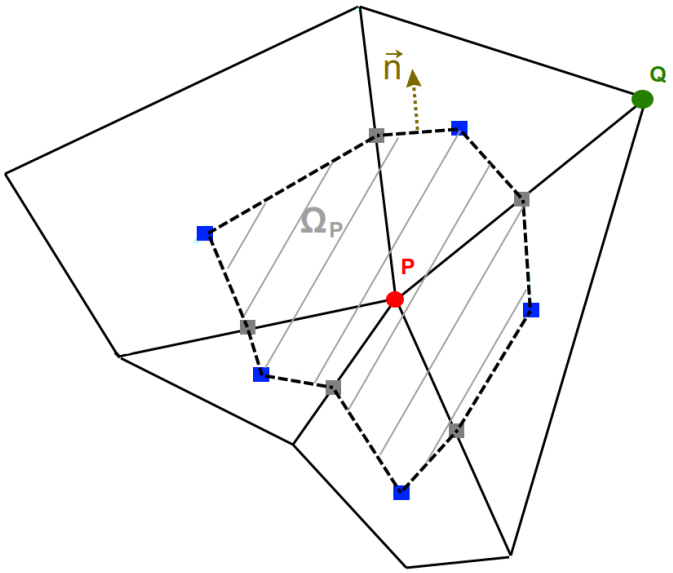


Figure 2.2: *Grid Elements (black continuous line), the Control Volume Ω_P around node P (grey hatch), its edges (black dashed line), the cell's barycenter (blue square), each edge's middle point (grey square), node P (red cycle), the neighbouring node Q (green cycle) and the vector right to the edge of the control volume Ω_P (yellow vector).*

For the numerical solution, the conservative form of equations is being chosen (eq. 2.7). After its volume integration on each finite volume Ω

$$\int_\Omega \frac{\partial \vec{U}}{\partial t} d\Omega + \int_\Omega \left(\frac{\partial \vec{f}}{\partial x} + \frac{\partial \vec{g}}{\partial y} \right) d\Omega = 0 \tag{2.13}$$

and the use of Green-Gauss theorem, it becomes

$$\int_\Omega \frac{\partial \vec{U}}{\partial t} d\Omega + \int_{\partial\Omega} \left(\vec{f} n_x + \vec{g} n_y \right) d\partial\Omega = 0 \tag{2.14}$$

Discretizing eq. 2.14

$$\frac{\Omega_P}{\Delta t_P} \Delta \vec{U}_P + \sum_Q \left(\vec{f} n_x + \vec{g} n_y \right) \Delta\partial\Omega = \frac{\Omega_P}{\Delta t_P} \Delta \vec{U}_P + \sum_Q \Phi_{PQ} \Delta\partial\Omega = 0 \tag{2.15}$$

The $\Phi_{PQ}\Delta\partial\Omega$ term describes the flux vector through the interface between the P and Q control volumes. The interface has a length equal to $\Delta\partial\Omega$. Because of this and in order for the flux to be computed, the flow variables must be computed on the inteface. Depending on the order of accuracy, the equations relating U_P to U_{PQ}^L and U_Q to U_{PQ}^R (Figure 2.3) are given as follows [21]

1^{st} order scheme :

$$\begin{aligned} U_{PQ}^{L} &= U_P \\ U_{PQ}^{R} &= U_Q \end{aligned} \tag{2.16}$$

2^{nd} order scheme :

$$\begin{aligned} U_{PQ}^{L} &= U_P + \frac{1}{2}(\vec{PQ})\nabla\vec{U}_P \\ U_{PQ}^{R} &= U_Q - \frac{1}{2}(\vec{PQ})\nabla\vec{U}_Q \end{aligned} \tag{2.17}$$

2^{nd} order scheme with Limiter:

$$\begin{aligned} U_{PQ}^{L} &= U_P + \frac{1}{2}LIM(\nabla\vec{U}_P, \nabla\vec{U}_Q) \\ U_{PQ}^{R} &= U_Q - \frac{1}{2}LIM(\nabla\vec{U}_Q, \nabla\vec{U}_P) \end{aligned} \tag{2.18}$$

where $LIM(a, b)$ a limiter's function.

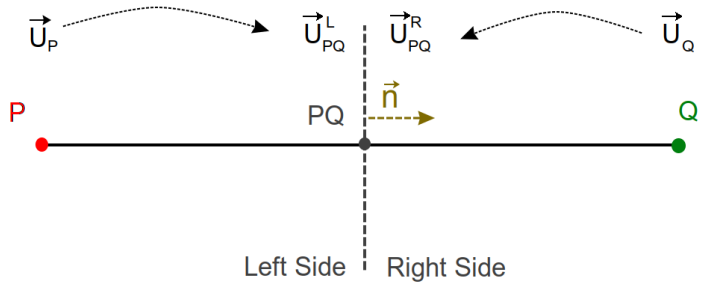


Figure 2.3: *The extrapolation of flow variables $\vec{U}$ from nodes P, Q to the middle of their inbetween grid edge PQ.*

According to the *Flux Vector Splitting* (FVS) scheme [22]

$$\vec{\Phi}_{PQ} = \mathscr{A}_{PQ}^{+}\vec{U}_{PQ}^{L} + \mathscr{A}_{PQ}^{-}\vec{U}_{PQ}^{R} \tag{2.19}$$

where $\mathscr{A}_{PQ}$ is the matrix computed by eq. 2.12 using the averaged primitive values

$\vec{V}_{PQ}$ of the PQ node,

$$\vec{V}_{PQ} = \frac{1}{2}\left[\vec{V}^L_{PQ} + \vec{V}^R_{PQ}\right] \tag{2.20}$$

Index "$+$" refers to the zeroing of the negative eigenvalues and "$-$" to the zeroing of the positive eigenvalues in eq. 2.12. Based on FVS, the *Roe approximate Riemann solver* [23] has been developed. According to it

$$\vec{\Phi}_{PQ} = \frac{1}{2}\left[\mathscr{A}^L_{PQ}\vec{U}^L_{PQ} + \mathscr{A}^R_{PQ}\vec{U}^R_{PQ}\right] - \frac{1}{2}\left|\tilde{\mathscr{A}}_{PQ}\right|(\vec{U}^R_{PQ} - \vec{U}^L_{PQ}) \tag{2.21}$$

where

$$\left|\tilde{\mathscr{A}}_{PQ}\right| = P\begin{bmatrix}|\lambda_1| & 0 & 0 & 0\\ 0 & |\lambda_2| & 0 & 0\\ 0 & 0 & |\lambda_3| & 0\\ 0 & 0 & 0 & |\lambda_4|\end{bmatrix}P^{-1} \tag{2.22}$$

The elements of the array above are computed on the PQ node using the Roe averaged values [23] of the variable $\vec{U}^L_{PQ}$ and $\vec{U}^R_{PQ}$, which are given by the equations

$$\begin{aligned}
\widetilde{\rho} &= \sqrt{\rho_L\rho_R}\\
\widetilde{u} &= \frac{u_L\sqrt{\rho_L} + u_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}\\
\widetilde{v} &= \frac{v_L\sqrt{\rho_L} + v_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}\\
\widetilde{H} &= \frac{H_L\sqrt{\rho_L} + H_R\sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}\\
\widetilde{c} &= \sqrt{(\gamma-1)(\widetilde{H} - \frac{\widetilde{u}^2 + \widetilde{v}^2}{2})}
\end{aligned} \tag{2.23}$$

where $H = \dfrac{\gamma}{\gamma-1}\dfrac{P}{\rho} + \dfrac{1}{2}(u^2+v^2)$ is the total enthalpy. Thus, eq. 2.21 transforms to

$$\vec{\Phi}_{PQ} = \frac{1}{2}\left[\mathscr{A}^L_{PQ}\vec{U}^L_{PQ} + \mathscr{A}^R_{PQ}\vec{U}^R_{PQ}\right] - \frac{1}{2}\left|\tilde{\mathscr{A}}_{PQ}\right|(\vec{U}^R_{PQ} - \vec{U}^L_{PQ}) \tag{2.24}$$

The last term that appears during the discretization of eq. 2.15, except from the geometrical ones (Ω_P and $\Delta\partial\Omega$) which are already known since the grid is available,

is the pseudo-time step Δt. It is estimated as

$$\Delta t = \frac{CFL}{T_i} \tag{2.25}$$

where CFL is the Courant − Friedrichs − Lewy number. The value CFL takes on should maximize the convergence rate without causing arithmetic stability issues. Moreover,

$$T_i = (|u_i| + c)\Omega_{P_i} \tag{2.26}$$

where Ω_{P_i} is the length of the P node's FV projection on the i direction. For the 2D analysis

$$T = (|u| + c)\Omega_{P_x} + (|v| + c)\Omega_{P_y} \tag{2.27}$$

2.2.1 Discretization of the Boundary Conditions

The nodes belonging to the boundaries of the computational field are associated with FVs as presented in Figure 2.4. In contrast to what was said thus far, flux Φ is coming in or going out from the FVs through the grid boundary segments. To compute this flux, the FVS scheme [22] is used.

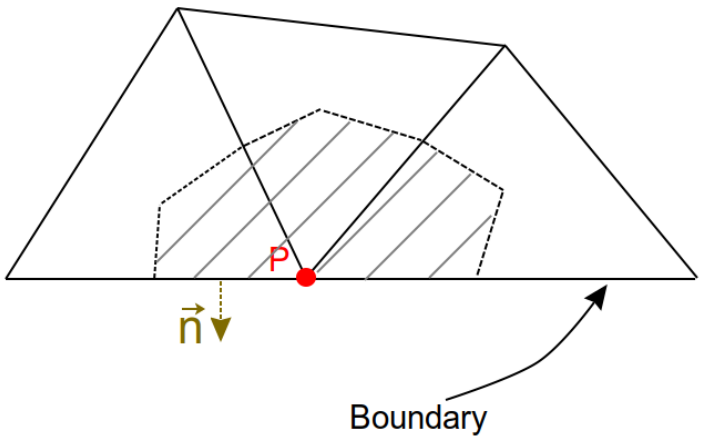


Figure 2.4: *Finite Volume around a boundary node P. The horizontal line corresponds to the boundary of the flow domain.*

Farfield

A farfield boundary is an interface between the undisturbed flow and the computational field. Thus, it is treated the same way as in Figure 2.3, where $\vec{U}_{PQ}^{L} = \vec{U}_P$ and $\vec{U}_{PQ}^{R} = \vec{U}_{inf}$. Likewise eq. 2.19

$$\vec{\Phi}_{inf} = \mathscr{A}_P^{+}\vec{U}_P + \mathscr{A}_P^{-}\vec{U}_{inf} \tag{2.28}$$

according to FVS. $\mathscr{A}_P^{+}$ is the matrix computed by eq. 2.12 if all the negative eigenvalues λ_j are zeroed and $\mathscr{A}_P^{-}$ the matrix if all positive eigenvalues λ_j are zeroed. Because of this, the farfield conditions affect the computational field only in places and to the extent the eigenvalues, determining the propagation of the information, allow it.

Solid Walls

According to eq. 2.15

$$\Phi_{PQ} = \vec{f}n_x + \vec{g}n_y = \begin{bmatrix} \rho u n_x \\ \rho u^2 n_x + p n_x \\ \rho u v n_x \\ u(\rho E + p) n_x \end{bmatrix} + \begin{bmatrix} \rho v n_y \\ \rho u v n_y \\ \rho v^2 n_y + p n_y \\ v(\rho E + p) n_y \end{bmatrix} = \begin{bmatrix} \rho(\vec{u}\cdot\vec{n}) \\ \rho u(\vec{u}\cdot\vec{n}) + p n_x \\ \rho v(\vec{u}\cdot\vec{n}) + p n_y \\ (\rho E + p)(\vec{u}\cdot\vec{n}) \end{bmatrix} \tag{2.29}$$

and applying the no-penetration condition $\vec{u}\cdot\vec{n} = 0$

$$\Phi_W = \begin{bmatrix} 0 \\ p n_x \\ p n_y \\ 0 \end{bmatrix} \tag{2.30}$$

2.3 Numerical Solution of flow equations

The variable ΔU^{n+1} is computed by

$$\Delta U^{n+1} = U^{n+1} - U^{n} \tag{2.31}$$

where n is the pseudo-time step. Knowing U^n from the previous iteration (or the field's initialization) and computing ΔU^{n+1}, the $n+1$ step variables emerge as

$$U^{n+1} = \Delta U^{n+1} + U^n \tag{2.32}$$

However, because of eq. 2.25 the $n+1$ step doesn't refer to the same pseudo-time for every node. For the unknown quantity to appear, eq. 2.15 is expressed for the step $n+1$ as

$$\frac{\Omega_P}{\Delta t_P}\Delta\vec{U}_P^{n+1} + \sum_Q \Phi_{PQ}^{n+1}\Delta\partial\Omega = 0 \tag{2.33}$$

In the **FVS scheme** [22], the 2^{nd} term of eq. 2.33 is combined with eq. 2.19 and eq. 2.32 resulting in

$$\begin{aligned}\sum_Q \Phi_{PQ}^{n+1}\Delta\partial\Omega = \sum_Q \left\{\mathscr{A}_{PQ}^{+}\vec{U}_{PQ}^{L^{n+1}} + \mathscr{A}_{PQ}^{-}\vec{U}_{PQ}^{R^{n+1}}\right\}\Delta\partial\Omega = \\ \sum_Q \Phi_{PQ}^{n}\Delta\partial\Omega + \sum_Q \left\{\mathscr{A}_{PQ}^{+}\Delta\vec{U}_{PQ}^{L^{n+1}} + \mathscr{A}_{PQ}^{-}\Delta\vec{U}_{PQ}^{R^{n+1}}\right\}\Delta\partial\Omega\end{aligned} \tag{2.34}$$

while for the **Roe's scheme** [23] instead of eq. 2.19, eq. 2.24 is being used. Thus,

$$\begin{aligned}\sum_Q \Phi_{PQ}^{n+1}\Delta\partial\Omega = \sum_Q \left\{\frac{1}{2}\left[\mathscr{A}_{PQ}^{L}\vec{U}_{PQ}^{L^{n+1}} + \mathscr{A}_{PQ}^{R}\vec{U}_{PQ}^{R^{n+1}}\right] - \frac{1}{2}\left|\tilde{\mathscr{A}}_{PQ}\right|(\vec{U}_{PQ}^{R^{n+1}} - \vec{U}_{PQ}^{L^{n+1}})\right\}\Delta\partial\Omega = \\ \sum_Q \Phi_{PQ}^{n}\Delta\partial\Omega + \sum_Q \left\{\frac{1}{2}\left[\mathscr{A}_{PQ}^{L} + \left|\tilde{\mathscr{A}}_{PQ}\right|\right]\Delta\vec{U}_{PQ}^{L^{n+1}} + \frac{1}{2}\left[\mathscr{A}_{PQ}^{R} - \left|\tilde{\mathscr{A}}_{PQ}\right|\right]\Delta\vec{U}_{PQ}^{R^{n+1}}\right\}\Delta\partial\Omega\end{aligned} \tag{2.35}$$

For simplicity reasons, first order scheme is considered (eq. 2.16). Assuming that

$$\sum_Q \Phi_{PQ}^{n+1}\Delta\partial\Omega = \sum_Q \Phi_{PQ}^{n}\Delta\partial\Omega + \sum_Q \left\{\mathscr{A}_{PQ}^{+}\Delta\vec{U}_P^{n+1} + \mathscr{A}_{PQ}^{-}\Delta\vec{U}_Q^{n+1}\right\}\Delta\partial\Omega \tag{2.36}$$

and

$$\sum_Q \Phi_{PQ}^{n+1}\Delta\partial\Omega = \sum_Q \Phi_{PQ}^{n}\Delta\partial\Omega + \sum_Q \left\{\frac{1}{2}\left[\mathscr{A}_{P} + \left|\tilde{\mathscr{A}}_{PQ}\right|\right]\Delta\vec{U}_P^{n+1} + \frac{1}{2}\left[\mathscr{A}_{Q} - \left|\tilde{\mathscr{A}}_{PQ}\right|\right]\Delta\vec{U}_Q^{n+1}\right\}\Delta\partial\Omega \tag{2.37}$$

emerge from eq. 2.34 and eq. 2.35 respectively. Finally, eq. 2.33 for the two schemes becomes

Flux Vector Splitting

$$\underbrace{\left\{\frac{\Omega_P}{\Delta t_P}+\sum_Q \mathscr{A}^+_{PQ}\right\}}_{I}\Delta\vec{U}^{n+1}_P+\underbrace{\left\{\sum_Q \mathscr{A}^-_{PQ}\right\}}_{II}\Delta\vec{U}^{n+1}_Q=-\sum_Q \Phi^n_{PQ}\Delta\partial\Omega \tag{2.38}$$

Roe's Approximate Riemann Solver

$$\underbrace{\left\{\frac{\Omega_P}{\Delta t_P}+\frac{1}{2}\sum_Q\left[\mathscr{A}_P+\left|\tilde{\mathscr{A}}_{PQ}\right|\right]\right\}}_{I}\Delta\vec{U}^{n+1}_P+\underbrace{\left\{\frac{1}{2}\sum_Q\left[\mathscr{A}_Q+\left|\tilde{\mathscr{A}}_{PQ}\right|\right]\right\}}_{II}\Delta\vec{U}^{n+1}_Q=-\sum_Q \Phi^n_{PQ}\Delta\partial\Omega \tag{2.39}$$

Considering that all the equations are extracted from m nodes, a square $m \times m$ matrix emerges. Every node's influence on itself (term I) and to its neighbours (term II) is added to that matrix. By definition, the I terms will be in the diagonal, while the II will be scattered in the remaining places. For example, based on eq. 2.39 and depicting the contribution of the node Q to P as II_{PQ}, the matrix given below is created for the seven-node grid in Figure 2.5,

$$\begin{bmatrix} I_1 & II_{12} & II_{13} & II_{14} & 0 & 0 & 0 \\ II_{21} & I_2 & II_{23} & II_{26} & 0 & 0 & 0 \\ II_{31} & II_{32} & I_3 & II_{34} & II_{35} & 0 & 0 \\ II_{41} & 0 & II_{43} & I_4 & II_{45} & 0 & II_{47} \\ 0 & 0 & II_{53} & II_{54} & I_5 & II_{56} & II_{57} \\ 0 & II_{62} & 0 & 0 & II_{65} & I_6 & II_{67} \\ 0 & 0 & 0 & II_{74} & II_{75} & II_{76} & I_7 \end{bmatrix} \begin{bmatrix} \Delta U_1^{n+1} \\ \Delta U_2^{n+1} \\ \Delta U_3^{n+1} \\ \Delta U_4^{n+1} \\ \Delta U_5^{n+1} \\ \Delta U_6^{n+1} \\ \Delta U_7^{n+1} \end{bmatrix} \tag{2.40}$$

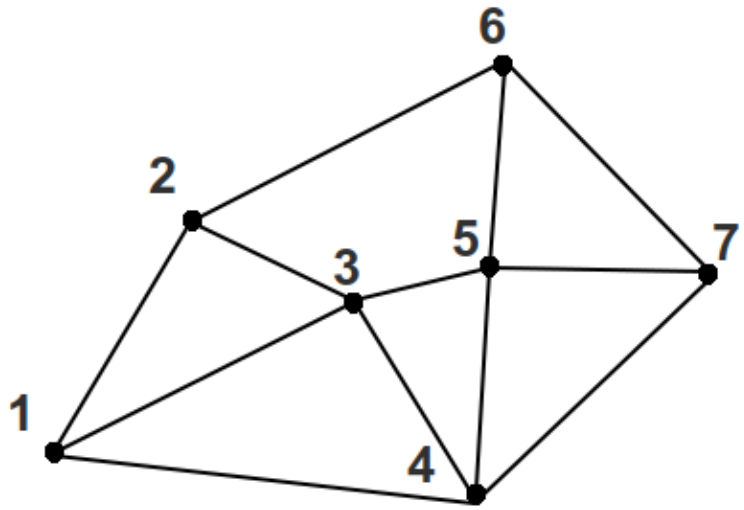


Figure 2.5: *Random 2D grid with 7 nodes. Based on it, the matrix of eq. 2.40 is assembled. Every node's finite volume is affected by its neighbours. Nodes that do not connect with grid line do not interact directly.*

Even for such a small grid, the presence of many zeroes in the above square matrix denotes that its storage in the computer memory is highly uneconomic. For that reason, the I terms are registered in a $m \times 1$ array called *Diag* and the II terms in a $2 \times l$ called *Zm*. l is the number of grid edges. The RHS of the equation originates from eq. 2.24.

2.3.1 Implementation of the Boundary Conditions

Farfield

From the flux between node P and the undisturbed flow, as described by eq. 2.28

$$\Phi_{inf}^{n+1} = \Phi_{inf}^{n} + \mathscr{A}_P^{+} \Delta \vec{U}_P^{n+1} \tag{2.41}$$

and, finally,

$$\left\{ \frac{\Omega_P}{\Delta t_P} + \underbrace{\mathscr{A}_P^{+}}_{III_{inf}} \right\} \Delta \vec{U}_P^{n+1} = \underbrace{-\Phi_{inf}^{n} \Delta \partial \Omega}_{IV_{inf}} \tag{2.42}$$

Solid Walls

Having defined Φ_W in eq. 2.30, with the use of

$$\Phi_W^{n+1} = \Phi_W^n + \frac{\partial \Phi_W}{\partial \vec{U}} \Delta \vec{U}_P^{n+1} \tag{2.43}$$

where
$$\frac{\partial P}{\partial \vec{U}} = (\gamma - 1) \begin{bmatrix} \frac{1}{2}(u^2 + v^2) \\ -u \\ -v \\ 1 \end{bmatrix} \tag{2.44}$$

and after applying the same precedure as before, the equation that arises is

$$\left\{ \frac{\Omega_P}{\Delta t_P} + \underbrace{\mathscr{A}_P^{Wall}}_{III_W} \right\} \Delta \vec{U}_P^{n+1} = \underbrace{-\Phi_W^n \Delta \partial \Omega}_{IV_W} \tag{2.45}$$

where
$$\mathscr{A}_P^{Wall} = (\gamma - 1) \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{u^2+v^2}{2} n_x & -u n_x & -v n_x & n_x \\ \frac{u^2+v^2}{2} n_y & -u n_y & -v n_y & n_y \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.46}$$

Terms III are added to I_P and IV to the RHS of the node P equation.

2.4 The Jacobi Solution Method

After the LHS (L) and RHS (R) matrices have been filled, the Jacobi method is applied. The system is described by the equation

$$L \; \Delta \vec{U} = \vec{R} \tag{2.47}$$

$$\text{where} \quad L = \begin{bmatrix} l_{11} & l_{12} & \cdots & l_{1n} \\ l_{21} & l_{22} & \cdots & l_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}, \quad \Delta\vec{U} = \begin{bmatrix} \Delta U_1 \\ \Delta U_2 \\ \vdots \\ \Delta U_n \end{bmatrix} \quad \text{and} \quad \vec{R} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \tag{2.48}$$

and n is the total number of the nodes. The L matrix can be written as

$$L = D + G_l + G_u \tag{2.49}$$

where D is the diagonal, G_l the lower and G_u the upper part of the matrix L. The system's solution is computed as

$$\Delta\vec{U}^{k+1} = D^{-1}\left[\vec{R} - (G_l + G_u)\Delta\vec{U}^{k}\right] \tag{2.50}$$

or as an element-based formula

$$\Delta U_i^{k+1} = \frac{1}{l_{ii}}\left(r_i - \sum_{j\neq 1} l_{ij}\Delta U_j^k\right), \quad i = 1, 2, \ldots, n \tag{2.51}$$

Jacobi is based on iterations (k) to find the solution of the system. The k+1 variables are computed iteratively until a convergence criterion is met between ΔU_i^{k+1} and ΔU_i^k.

Chapter 3

The Adjoint E-SI Method

In this chapter, the three different adjoint methods are described and the mathematical formulation of the E-SI adjoint is presented. Finally, the optimization routines based on adjoint is outlined.

3.1 Different Continuous Adjoint Formulations

For the computation of the $\frac{\delta U}{\delta b_n}$, the computational cost is large. In order to avoid this, an adjoint field Ψ_i is defined. Since $R_i = \frac{\partial f_i}{\partial x_i} = 0$ in the volume Ω, the objective function F can be transformed

$$F_{AUG} = F = F + \int_\Omega \Psi_i R_i d\Omega \tag{3.1}$$

where $i = 1, \ldots, E$ (E is the number of the state equations). Differentiating eq. 3.1 w.r.t. b_n yields

$$\frac{\delta F_{AUG}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n} \int_\Omega \Psi_i R_i d\Omega \tag{3.2}$$

To further expand the $\delta/\delta b_n$ derivative of the volume integral, two alternatives can be pursuited. According to the FI adjoint approach

$$\frac{\delta}{\delta b_n} \int_\Omega \Psi_i R_i d\Omega = \int_\Omega \Psi_i \frac{\delta R_i}{\delta b_n} d\Omega + \int_\Omega \Psi_i \cancelto{0}{R_i} \frac{\delta d\Omega}{\delta b_n} \tag{3.3}$$

and with the use of eq. B.1 and eq. B.4 the final form of the SDs emerge, which contains field integrals of $\frac{\delta x_k}{\delta b_n}$.

Their computation is costly and can be avoided with the use of SI adjoint approach. Based on it, for the second term of eq. 3.2 the Leibniz theorem (eq. B.2) is applied resulting in

$$\frac{\delta F_{AUG}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_\Omega \Psi_i \frac{\partial R_i}{\partial x_k} d\Omega + \underbrace{\int_S \Psi_i R_i n_k \frac{\delta x_k}{\delta b_n} dS}_{\text{LBterm}} \tag{3.4}$$

The LBterm (Leibniz term) is often ignored in the literature because of the assumption that $R_i = 0$ not only in the flow domain (Ω) but also along its boundaries (S). In fine grids, where the flow equations are satisfied very close to the boundaries, neglecting it won't affect the SDs' accuracy. However, depending on the case and especially the grid's coarseness it may lead to deviation in the SDs. That's the reason why the FI adjoint approach is more accurate, as it makes no assumptions at all. The straightforward computation of the LBterm is not the appropriate treatment. The computation of the flow equations residuals along the boundaries has numerical difficulties and uncertainties, especialy in on an unstructured grid.

The Enhanced SI (E-SI) adjoint proposed by PCOpt/NTUA aims to avoid the computation of the LBterm by introducing the new adjoint variables m_i^a, of the Laplace grid displacement Partial Differential Equations (gdPDEs) which hypothetically govern the grid dissplacement

$$R_i^m = \frac{\partial^2 m_i}{\partial x_j^2} = 0, \qquad \text{(i=1,2 for 2D flows)} \tag{3.5}$$

where m_i is the dissplacement of the grid's nodes in the cartesian space. A thorough analysis for the E-SI adjoint approach that is used in this diploma thesis is done in the following section. The E-SI adjoint manages to compute the SDs with the FI adjoint accuracy and SI adjoint cost.

3.2 Odjective Function (F)

For airfoil design, the goal is to maximize (or keep constant) the lift coefficient (C_L), minimize drag coefficient (C_D) or both. Thus, the function F for the optimization algorithm to minimize is

$$F = w_l \left(C_L - C_{L_{tar}}\right)^2 + w_d C_D^2 \tag{3.6}$$

where $C_{L_{tar}}$ is a preset value of the lift coefficient and w_d, w_l weight functions. The forces, lift (L) and drag (D), acted upon an airfoil are expressed right and parallel to the farfield velocity $\vec{u}_\infty$ respectively. These are computed from the pressure distribution on the airfoil surface, which is implemented right to the solid wall and its direction is opposite to the unitary vector $\vec{n} = (n_x, n_y)$. By definition, $\vec{n}$ points away from the solid wall. The two coefficients are expressed below w.r.t. the static pressure p to the flow equations and the farfield flow angle (a_∞) (Figure 3.1).

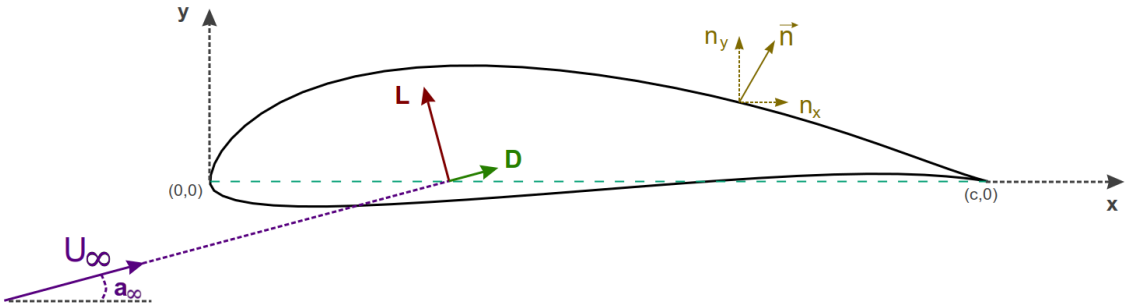


Figure 3.1: *The airfoil (black line), the farfield velocity and angle of attack (AoA) (purple vector), lift (red vector) and drag (green vector) forces.*

$$\begin{aligned} C_L &= \frac{L}{q} = \int_{S_W} \frac{p\,(n_y cos(a_\infty) - n_x sin(a_\infty))}{q} dS_W \\ C_D &= \frac{D}{q} = \int_{S_W} \frac{p\,(n_x cos(a_\infty) + n_y sin(a_\infty))}{q} dS_W \end{aligned} \tag{3.7}$$

where $q = \frac{1}{2}\rho_\infty U_\infty^2 c$. ρ_∞ is the farfield density and c is the airfoil's chord. Differentiating eq. 3.6 w.r.t. the design variable b_n (Appendix A) and replacing the coefficients with the use of eq. 3.7 yields

$$\frac{\delta F}{\delta b_n} = 2w_l \left(C_L - C_{L_{tar}}\right) \frac{\delta C_L}{\delta b_n} + 2w_d C_D \frac{\delta C_D}{\delta b_n} =$$

$$= \underbrace{\int_{S_W} \left[2w_l \left(C_L - C_{L_{tar}}\right) \frac{\left(n_y cos(a_\infty) - n_x sin(a_\infty)\right)}{q} + 2w_d C_D \frac{\left(n_x cos(a_\infty) + n_y sin(a_\infty)\right)}{q} \right] \frac{\delta p}{\delta b_n} dS_W}_{ABC^1} +$$

$$+ \underbrace{2w_l \left(C_L - C_{L_{tar}}\right) \int_{S_W} p \frac{\frac{\delta \overline{n_y}}{\delta b_n} cos(a_\infty) - \frac{\delta \overline{n_x}}{\delta b_n} sin(a_\infty)}{q}}_{SD^1} + \underbrace{2w_d C_D \int_{S_W} p \frac{\frac{\delta \overline{n_x}}{\delta b_n} cos(a_\infty) + \frac{\delta \overline{n_y}}{\delta b_n} sin(a_\infty)}{q}}_{SD^2} \tag{3.8}$$

where $\overline{n_i} = n_i dS_W$ (dimensional normal vector).

3.3 The E-SI Adjoint

Based on the above, for the E-SI adjoint, F_{AUG} (eq. 3.1) is further expanded to

$$F_{AUG} = F + \int_\Omega \Psi_i R_i d\Omega + \int_\Omega m_i^a R_i^m d\Omega \qquad \Longrightarrow \tag{3.9}$$

$$\frac{\delta F_{AUG}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \underbrace{\frac{\delta}{\delta b_n} \int_\Omega \Psi_i R_i d\Omega}_{Adj_\Psi} + \underbrace{\frac{\delta}{\delta b_n} \int_\Omega m_i^a R_i^m d\Omega}_{Adj_m} \tag{3.10}$$

On the RHS of eq.3.10, the first term is given by eq. 3.8. A more detailed derivation of the expressions for surface and even volume integrals of F can be found in [24]. The second term, with the use of eq. B.2 and eq. B.3 becomes

$$Adj_\Psi \equiv \int_\Omega \Psi_i \frac{\delta R_i}{\delta b_n} d\Omega = \int_\Omega \Psi_i \frac{\delta}{\delta b_n} \left[\frac{\partial f_k^i}{\partial x_k}\right] d\Omega =$$

$$= \int_\Omega \Psi_i \frac{\partial}{\partial x_k} \left[\frac{\delta f_k^i}{\delta b_n}\right] d\Omega - \int_\Omega \Psi_i \frac{\partial f_k^i}{\partial x_j} \frac{\partial}{\partial x_k} \left[\frac{\delta x_j}{\delta b_n}\right] d\Omega =$$

$$= \underbrace{\int_S \Psi_i A_k^{ij} \frac{\delta U_j}{\delta b_n} n_k \, dS}_{I} \underbrace{- \int_\Omega \frac{\delta U_j}{\delta b_n} A_k^{ij} \frac{\partial \Psi_i}{\partial x_k} d\Omega}_{FAE_\Psi^1} \underbrace{- \int_\Omega \Psi_i \frac{\partial f_k^i}{\partial x_j} \frac{\partial}{\partial x_k} \left[\frac{\delta x_j}{\delta b_n}\right] d\Omega}_{II} \tag{3.11}$$

where $f_k^i = A_k^{ij} U_j \Leftrightarrow \frac{\delta f_k^i}{\delta b_n} = A_k^{ij} \frac{\delta U_j}{\delta b_n}$. The eq's 3.11 therm I becomes

$$I = \int_S \Psi_i A_k^{ij} \frac{\delta U_j}{\delta b_n} n_k \, dS = \underbrace{\int_{S_{inf}} \Psi_i A_k^{ij} \frac{\delta U_j}{\delta b_n} n_k dS_{inf}}_{ABC^2} + \int_{S_W} \Psi_i A_k^{ij} \frac{\delta U_j}{\delta b_n} n_k dS_W \quad (3.12)$$

The second term of eq. 3.12 is further expanded

$$\begin{aligned} \int_{S_W} \Psi_i A_k^{ij} \frac{\delta U_j}{\delta b_n} n_k dS_W &= \int_{S_W} \Psi_i \frac{\delta(f_k^i \overline{n_k})}{\delta b_n} - \int_{S_W} \Psi_i f_k^i \frac{\delta(\overline{n_k})}{\delta b_n} = \\ &= \underbrace{\int_{S_W} \Psi_{k+1} n_k \frac{\delta p}{\delta b_n} dS_W}_{ABC^3} + \underbrace{\int_{S_W} \left[\Psi_{k+1} p - \Psi_i f_k^i\right] \frac{\delta(\overline{n_k})}{\delta b_n}}_{SD^3} \end{aligned} \quad (3.13)$$

because, on solid walls,

$$\left.\frac{\delta(f_k \overline{n_k})}{\delta b_n}\right|_{S_W} = \frac{\delta}{\delta b_n} \left[\vec{f}\, \overline{n_x} + \vec{g}\, \overline{n_y}\right]_{S_W} = \left[\begin{array}{c} 0 \\ \frac{\delta p}{\delta b_n}(\overline{n_x}) + p \frac{\delta(\overline{n_x})}{\delta b_n} \\ \frac{\delta p}{\delta b_n}(\overline{n_y}) + p \frac{\delta(\overline{n_y})}{\delta b_n} \\ 0 \end{array}\right]_{S_W} \quad (3.14)$$

$$u_n = u_i n_i = u n_x + v n_y = 0 \qquad \text{(slip - no-penetration condition)}$$

Term II in eq. 3.11 becomes

$$\begin{aligned} II = -\int_\Omega \Psi_i \frac{\partial f_k^i}{\partial x_j} \frac{\partial}{\partial x_k} \left[\frac{\delta x_j}{\delta b_n}\right] d\Omega &= \int_\Omega \frac{\partial}{\partial x_k} \left[\Psi_i \frac{\partial f_k^i}{\partial x_j}\right] \frac{\delta x_j}{\delta b_n} d\Omega - \int_S \Psi_i \frac{\partial f_k^i}{\partial x_j} \frac{\delta x_j}{\delta b_n} n_k dS = \\ &= \underbrace{\int_\Omega \frac{\partial}{\partial x_k} \left[\Psi_i \frac{\partial f_k^i}{\partial x_j}\right] \frac{\delta x_j}{\delta b_n} d\Omega}_{FAE_m^1} \underbrace{- \int_{S_W} \Psi_i \frac{\partial U_m}{\partial x_j} A_k^{im} \frac{\delta x_j}{\delta b_n} n_k dS_W}_{SD^4} \end{aligned} \quad (3.15)$$

$$\vec{U} = const.\Big|_{S=S_{inf}}$$

For the third term of eq. 3.10, eq. B.1, eq. B.2 and eq. B.7 are used

$$Adj_m \equiv \frac{\delta}{\delta b_n}\int_\Omega m_i^a R_i^m d\Omega = \int_\Omega m_i^a \frac{\partial R_i^m}{\partial b_n} d\Omega + \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS =$$

$$= \int_S m_i^a n_j \frac{\partial}{\partial x_j}\left[\frac{\partial m_i}{\partial b_n}\right] dS - \int_S \frac{\partial m_i^a}{\partial x_j} n_j \frac{\partial m_i}{\partial b_n} dS + \int_\Omega \frac{\partial^2 m_i^a}{\partial x_j^2}\frac{\partial m_i}{\partial b_n} d\Omega + \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS \quad (3.16)$$

and because $\dfrac{\delta x_i}{\delta b_n} = \dfrac{\partial m_i}{\partial b_n}$, as explained in [17]

$$Adj_m = \underbrace{\int_S m_i^a n_j \frac{\partial}{\partial x_j}\left[\frac{\delta x_i}{\delta b_n}\right] dS}_{ABC_m^1} \underbrace{- \int_S \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS}_{SD^5} \underbrace{+ \int_\Omega \frac{\partial^2 m_i^a}{\partial x_j^2}\frac{\delta x_i}{\delta b_n} d\Omega}_{FAE_m^2} \underbrace{+ \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS}_{SD^6} \quad (3.17)$$

The adjoint method is a technique that avoids the expensive computation of the derivatives of the flow variables w.r.t. b_n and the grid SDs in the flow domain Ω. To do so, the adjoint variables (Ψ_i, m_i^a) are properly defined and the Field Adjoint Equations (FAE) and Adjoint Boundary Conditions (ABC) are derived.

The field adjoint

In order to avoid the computation of $\dfrac{\delta U_j}{\delta b_n}$ in the volume Ω, the field adjoint equations (FAE_Ψ) are defined

$$-\int_\Omega \frac{\delta U_j}{\delta b_n} A_k^{ij} \frac{\partial \Psi_i}{\partial x_k} d\Omega = -A_k^{ij}\frac{\partial \Psi_i}{\partial x_k} = 0$$

Likewise the primal equations, in order to perform a stable numerical solution, a new term is added using pseudo-time.

$$\frac{\partial \Psi_i}{\partial t} - A_k^{ij}\frac{\partial \Psi_i}{\partial x_k} = 0 \quad \text{or} \quad \frac{\partial \vec{\Psi}}{\partial t} - A_i^T \frac{\partial \vec{\Psi}}{\partial x_i} = 0 \quad (3.18)$$

To avoid the computation of $\dfrac{\delta U_j}{\delta b_n}$ along the boundaries $S = S_{inf} \cup S_W$, the boundary conditions for the field adjoint equations (ABC_Ψ) are defined

$$\int_{S_{inf}} \frac{\delta U_j}{\delta b_n} A_k^{ij} n_k \Psi_i \, dS_{inf} = 0 \implies \frac{\delta U_j}{\delta b_n} A_k^{ij} n_k \Psi_i \Big|_{S=Sinf} = 0 \quad (3.19)$$

$$\int_{S_W} \Bigg[2w_l \left(C_L - C_{L_{tar}}\right) \frac{\left(n_y cos(a) - n_x sin(a)\right)}{q} \quad + \quad 2w_d C_D \frac{\left(n_x cos(a) + n_y sin(a)\right)}{q} \\ + \quad \Psi_2 n_x + \Psi_3 n_y \Bigg] \frac{\delta p}{\delta b_n} dS_W = 0 \qquad (3.20)$$

The adjoint grid displacement equations

In order to avoid the computation of $\frac{\delta x_j}{\delta b_n}$ in the volume Ω, the adjoint grid displacement equations (FAE_m) are defined

$$\int_\Omega \frac{\partial^2 m_j^a}{\partial x_i^2} \frac{\delta x_j}{\delta b_n} d\Omega + \int_\Omega \frac{\partial}{\partial x_k} \left[\Psi_i \frac{\partial f_k^i}{\partial x_j} \right] \frac{\delta x_j}{\delta b_n} d\Omega = 0 \quad \Longrightarrow \quad \frac{\partial^2 m_j^a}{\partial x_i^2} + \frac{\partial}{\partial x_k} \left[\Psi_i \frac{\partial f_k^i}{\partial x_j} \right] = 0 \quad \Longrightarrow$$

$$\frac{\partial^2 m_j^a}{\partial x_i^2} + \frac{\partial \Psi_i}{\partial x_k} \frac{\partial f_k^i}{\partial x_j} + \Psi_i \frac{\partial}{\partial x_k} \left[\frac{\partial f_k^i}{\partial x_j} \right] = \frac{\partial^2 m_j^a}{\partial x_i^2} + \cancel{\frac{\partial U_m}{\partial x_k} A_k^{im}} \cancel{\frac{\partial \Psi_i}{\partial x_k}} = 03.18 + \Psi_i \frac{\partial}{\partial x_k} \left[\frac{\partial f_k^i}{\partial x_j} \right] = 0 \Longrightarrow$$

$$\frac{\partial^2 m_j^a}{\partial x_i^2} + \Psi_i \frac{\partial}{\partial x_k} \left[\frac{\partial f_k^i}{\partial x_j} \right] = 0 \qquad (3.21)$$

In order to avoid the computation of $\frac{\partial}{\partial x_j} \left[\frac{\delta x_i}{\delta b_n} \right]$ along the boundaries $S = S_{inf} \cup S_W$, the boundary conditions for the adjoint grid displacement equations (ABC_m) are defined

$$\int_S m_i^a n_j \frac{\partial}{\partial x_j} \left[\frac{\delta x_i}{\delta b_n} \right] dS = 0 \quad \Longrightarrow \quad m_i^a|_S = 0 \qquad (3.22)$$

Sensitivity Derivatives

After the satisfaction of both FAE and ABC, the remaining terms constitute the SDs

$$\frac{\delta F}{\delta b_n} = \underbrace{2w_l \frac{C_L - C_{L_{tar}}}{q} \int_{S_W} p \left[\frac{\delta \overline{n_y}}{\delta b_n} cos(a) - \frac{\delta \overline{n_x}}{\delta b_n} sin(a) \right]}_{SD^1}$$

$$\underbrace{+2w_d \frac{C_D}{q} \int_{S_W} p \left[\frac{\delta \overline{n_x}}{\delta b_n} cos(a) + \frac{\delta \overline{n_y}}{\delta b_n} sin(a) \right]}_{SD^2} \underbrace{+ \int_{S_W} \left[\Psi_{k+1} p - \Psi_i f_k^i \right] \frac{\delta(\overline{n_k})}{\delta b_n}}_{SD^3}$$

$$\underbrace{- \int_{S_W} \Psi_i \frac{\partial U_m}{\partial x_j} A_k^{im} \frac{\delta x_j}{\delta b_n} n_k dS_W}_{SD^4} \underbrace{- \int_S \frac{\partial m_i^a}{\partial x_j} n_j \frac{\delta x_i}{\delta b_n} dS}_{SD^5} \underbrace{+ \int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n} dS}_{SD^6} = 03.22 \tag{3.23}$$

The SD^5 term distinguishes the $E-SI$ from the $Severed-SI$ (standard SI) adjoint [13]. The latter one considers $\frac{\partial}{\partial x_k}\left[\frac{\partial f_k}{\partial x_j}\right] = 0$ all over the field. Thus, there is no need to introduce the adjoint grid displacement fields m_i^a and SD^5 is not produced.

3.3.1 Numerical Solution of the adjoint equations

3.3.1.1 Solving FAE_Ψ

For the $\vec{\Psi}$ vector, eq. 3.18, 3.19 and 3.20 have emerged. Likewise eq 2.7,

$$\frac{\partial \Psi_i}{\partial t} - A^{ij} \frac{\partial \Psi_i}{\partial x} - B^{ij} \frac{\partial \Psi_i}{\partial y} = 0 \tag{3.24}$$

has an adjoint flux vector

$$\vec{\Phi}_{PQ} = -\mathscr{A}_{PQ}^{T,-} \vec{\Psi}_{PQ}^L - \mathscr{A}_{PQ}^{T,+} \vec{\Psi}_{PQ}^R \tag{3.25}$$

based on FVS [22] and

$$\vec{\Phi}_{PQ} = -\frac{1}{2}\left[\mathscr{A}_{PQ}^{T,L} - \left|\tilde{\mathscr{A}}_{PQ}^T\right|\right] \vec{\Psi}_{PQ}^L - \frac{1}{2}\left[\mathscr{A}_{PQ}^{T,R} + \left|\tilde{\mathscr{A}}_{PQ}^T\right|\right] \vec{\Psi}_{PQ}^R \tag{3.26}$$

based on Roe's scheme [23], where for the needs of the adjoint flow equations $\mathscr{A}^T = A_i^T n_i$. In contrast to the primal eq. 2.7, the adjoint eq. 3.24 cannot be written in a conservative form. Thus, based on Roe's scheme, a non-conservative adjoint flux

can be extracted [25]

$$\begin{aligned}\vec{\Phi}_{PQ} &= -\frac{1}{2}\mathscr{A}_{PQ}^{T,L}\left[\vec{\Psi}_{PQ}^{L}+\vec{\Psi}_{PQ}^{R}\right]-\frac{1}{2}\left|\tilde{\mathscr{A}}_{PQ}^{T}\right|\left[\vec{\Psi}_{PQ}^{R}-\vec{\Psi}_{PQ}^{L}\right]\\ \vec{\Phi}_{QP} &= \frac{1}{2}\mathscr{A}_{PQ}^{T,R}\left[\vec{\Psi}_{PQ}^{L}+\vec{\Psi}_{PQ}^{R}\right]+\frac{1}{2}\left|\tilde{\mathscr{A}}_{PQ}^{T}\right|\left[\vec{\Psi}_{PQ}^{R}-\vec{\Psi}_{PQ}^{L}\right]\end{aligned} \tag{3.27}$$

The adjoint boundary conditions on $\mathbf{S}_{inf}$ (eq. 3.19) emerge from equation

$$\left.\frac{\delta U_j}{\delta b_n}A_k^{ij}n_k\Psi_i\right|_{S=Sinf}=0\Longrightarrow\begin{bmatrix}\Psi_1\\ \Psi_2\\ \Psi_3\\ \Psi_4\end{bmatrix}_{S_{inf}}=\begin{bmatrix}0\\ 0\\ 0\\ 0\end{bmatrix} \tag{3.28}$$

and the adjoint flux is computed from

$$\vec{\Phi}_{inf}^{\Psi}=-\mathscr{A}_P^{T,-}\vec{\Psi}_P-\mathscr{A}_P^{T,+}\vec{\Psi}_{inf}=-\mathscr{A}_P^{-}\vec{\Psi}_P \tag{3.29}$$

The boundary conditions on $\mathbf{S}_W$ (eq. 3.20) become

$$2w_l\left(C_L-C_{L_{tar}}\right)\frac{n_y cos(a)-n_x sin(a)}{q}+2w_d C_D\frac{n_x cos(a)+n_y sin(a)}{q}+\Psi_2 n_x+\Psi_3 n_y=0 \tag{3.30}$$

and after taking into consideration eq. 3.30 and the no-penetration condition $\vec{u}\cdot\vec{n}=0$

$$\Phi_W^{\Psi}=\mathscr{A}_P^T\vec{\Psi}_P=\begin{bmatrix}\left[\frac{\gamma-1}{2}\vec{u}^2n_x-u(\vec{u}\cdot\vec{n})\right]\Psi_2+\left[\frac{\gamma-1}{2}\vec{u}^2n_y-v(\vec{u}\cdot\vec{n})\right]\Psi_3\\ \Psi_1 n_x+\left[(2-\gamma)un_x+\vec{u}\cdot\vec{n}\right]\Psi_2+\left[vn_x-(\gamma-1)un_y\right]\Psi_3+\left[\gamma E_t n_x-\frac{\gamma-1}{2}\vec{u}^2n_x-(\gamma-1)u(\vec{u}\cdot\vec{n})\right]\Psi_4\\ \Psi_1 n_y+\left[un_y-(\gamma-1)vn_x\right]\Psi_2+\left[(2-\gamma)vn_y+\vec{u}\cdot\vec{n}\right]\Psi_3+\left[\gamma E_t n_y-\frac{\gamma-1}{2}\vec{u}^2n_y-(\gamma-1)v(\vec{u}\cdot\vec{n})\right]\Psi_4\\ (\gamma-1)(\Psi_2 n_x+\Psi_3 n_y)+\gamma(\vec{u}\cdot\vec{n})\Psi_4\end{bmatrix}_P=$$

$$= \begin{bmatrix} -\frac{\gamma-1}{2}\vec{u}^2\left[2w_l\left(C_L - C_{L_{tar}}\right)\frac{n_y cos(a) - n_x sin(a)}{q} + 2w_d C_D \frac{n_x cos(a) + n_y sin(a)}{q}\right] \\ \Psi_1 n_x + (\Psi_2 u + \Psi_3 v)n_x - (1-\gamma)u\left[2w_l\left(C_L - C_{L_{tar}}\right)\frac{n_y cos(a) - n_x sin(a)}{q} + 2w_d C_D \frac{n_x cos(a) + n_y sin(a)}{q}\right] + \left[\gamma E_t - \frac{\gamma-1}{2}\vec{u}^2\right] n_x \Psi_4 \\ \Psi_1 n_y - (1-\gamma)v\left[2w_l\left(C_L - C_{L_{tar}}\right)\frac{n_y cos(a) - n_x sin(a)}{q} + 2w_d C_D \frac{n_x cos(a) + n_y sin(a)}{q}\right] + (\Psi_2 u + \Psi_3 v)n_y + \left[\gamma E_t - \frac{\gamma-1}{2}\vec{u}^2\right] n_y \Psi_4 \\ -(\gamma-1)\left[2w_l\left(C_L - C_{L_{tar}}\right)\frac{n_y cos(a) - n_x sin(a)}{q} + 2w_d C_D \frac{n_x cos(a) + n_y sin(a)}{q}\right] \end{bmatrix}_P \tag{3.31}$$

Each wall's edge contribution to the diagonal of the coefficient matrix is

$$\mathscr{A}_P^{T,Wall,\Psi} = \frac{\partial \Phi_W^{\Psi}}{\partial \vec{\Psi}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ n_x & un_x & vn_x & \gamma E_t n_x - \frac{\gamma-1}{2}\vec{u}^2 n_x \\ n_y & un_y & vn_y & \gamma E_t n_y - \frac{\gamma-1}{2}\vec{u}^2 n_y \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.32}$$

3.3.1.2 Solving FAE_m

Eq. 3.21 is integrated in the volume Ω

$$\int_\Omega \frac{\partial^2 m_j^a}{\partial x_i^2} d\Omega + \int_\Omega \Psi_i \frac{\partial}{\partial x_k}\left[\frac{\partial f_k^i}{\partial x_j}\right] d\Omega = 0 \tag{3.33}$$

and discretized, using eq. B.2 and Figure 2.2

$$\sum_Q \underbrace{\frac{\partial m_j^a}{\partial x_i}\overline{n_i}}_{R^n} + \Psi_i \frac{\partial}{\partial x_k}\left[\frac{\partial f_k^i}{\partial x_j}\right]_P \Omega_{cell} = 0 \tag{3.34}$$

where R is the equation's flux through each grid's segment and n is the time step, in the same manner with the CFD analysis. The second term is an already known quantity, as Ψ and f fields have been computed. Thus a Poisson's type equation

emerges. The space m_j^a derivative on PQ edge, as stated by [26], is

$$\frac{\partial m_{j,PQ}^a}{\partial x_i} = \frac{1}{2}\left[\frac{\partial m_{j,P}^a}{\partial x_i} + \frac{\partial m_{j,Q}^a}{\partial x_i}\right] - \left[\frac{1}{2}\left(\frac{\partial m_{j,P}^a}{\partial x_l} + \frac{\partial m_{j,Q}^a}{\partial x_l}\right) n_l - \frac{m_{j,Q}^a - m_{j,P}^a}{\left|\vec{PQ}\right|}\right] n_i \quad (3.35)$$

Using the Delta Formulation (eq. 2.32) and solving eq. 3.34 for the $n+1$ step

$$R^{n+1} = R^n + \frac{\partial R^n}{\partial m_j^a} \Delta m_j^{a\ n+1} \quad (3.36)$$

where for a 2D mesh with $\vec{P} = (X_P, Y_P)$

$$\begin{aligned} \frac{\partial R^n}{\partial m_j^a} \Delta m_j^{a\ n+1} = & - \left[\frac{X_Q - X_P}{\vec{PQ}^2}\overline{n_x} + \frac{Y_Q - Y_P}{\vec{PQ}^2}\overline{n_y}\right] \Delta m_{j,P}^{a\ n+1} \\ & + \left[\frac{X_Q - X_P}{\vec{PQ}^2}\overline{n_x} + \frac{Y_Q - Y_P}{\vec{PQ}^2}\overline{n_y}\right] \Delta m_{j,Q}^{a\ n+1} \end{aligned} \quad (3.37)$$

Hence, the second term of eq. 3.34 and R^n are the known quantities in every iteration. On the other hand, the square solution matrix is being filled with the terms of eq. 3.37. The first goes to the diagonal as it refers to the P node, around which the FV was defined.

3.3.1.3 Discretization of SDs

After their discretization, the SDs are computed. Because the numerical solutions are based on node-centered FV scheme, the U, Ψ and m^a fields are already known at each grid node. However, $\delta\overline{n_i}/\delta b_n$, n_i and dS are geometric quantities computed over the segments. Thus, an interpolation is needed. Separating the SD terms into two main categories, a more generic analysis can be made. The first one constitutes from SD^1 to SD^3 and is represented as

$$K_1 = \int_{S_W} \Phi \frac{\delta \overline{n}}{\delta b_n} \quad (3.38)$$

The second one refers to the remaining terms and has the form

$$K_2 = \int_{S_W} \Phi n dS \tag{3.39}$$

where Φ is any function of one or more flow or adjoint quantities known on each P node. For the description of finding K_1 and K_2 the symbols on Figure 3.2 are used.

For the K_1 integral, the U_M, Ψ_M and m_M^a quantities are linearly interpolated, between P and Q, on the middle of the segments and Φ_M is computed there. Hence, in a discrete form

$$K_1|_{node-P} = \frac{1}{2}\Phi_{M1} \left.\frac{\delta \overline{n}}{\delta b_n}\right|_{Segm.1} + \frac{1}{2}\Phi_{M2} \left.\frac{\delta \overline{n}}{\delta b_n}\right|_{Segm.2} \tag{3.40}$$

On the other hand, for the K_2 integral, the $\Phi n dS$ quantity is computed on nodes P and Q_i using the segment's PQ_i data. After that, an interpolation is made in the middle M_i. This process is repeated for both edges. A second linear interpolation between M_1 and M_2 computes the quantity at node P. In a discrete form,

$$K_2|_{node-P} = \frac{1}{2}\left[\Phi_{M1} + \Phi_{M2}\right] \tag{3.41}$$

where

$$\begin{aligned} \Phi_{Mi} &= \Phi_P \left[\frac{1}{2} n \frac{1}{2} dS\right]_{Segm.i} + \Phi_{Q_i} \left[\frac{1}{2} n \frac{1}{2} dS\right]_{Segm.i} = \\ &= \frac{1}{4}\left[\Phi_P (ndS)_{Segm.i} + \Phi_{Q_i} (ndS)_{Segm.i}\right] \end{aligned} \tag{3.42}$$

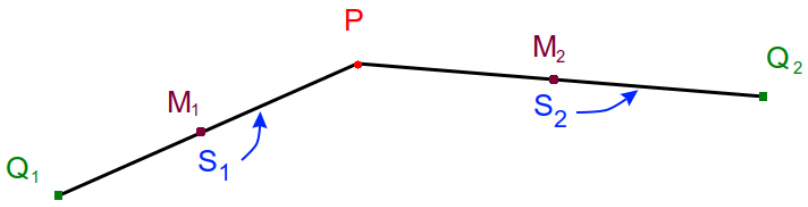


Figure 3.2: *Grid and Airfoil Edges (black line), node P (red dot), its 2 neighoubors $Q_{1,2}$ (green squares), the length of each segments $S_{1,2}$ and their middle points $M_{1,2}$ (purple dots).*

Instead of a straightforward computation of eq. 3.23, the chain rule can be used. Specificaly,

$$\begin{aligned} \frac{\delta F_{AUG}}{\delta b_n} &= \frac{\delta F_{AUG}}{\delta x}\frac{\delta x}{\delta b_n} \quad \text{or} \\ \frac{\delta F_{AUG}}{\delta b_n} &= \frac{\delta F_{AUG}}{\delta y}\frac{\delta y}{\delta b_n} \end{aligned} \tag{3.43}$$

where the last derivatives in eq. 3.43 are known via the parameterization (eq. A.5). The terms $\delta F_{AUG}/\delta x$ and $\delta F_{AUG}/\delta y$ must be numericaly computed at each node with the aforementioned techniques.

3.4 Optimization Flowchart

As indicated above, an optimization based on GBM needs to compute the SDs and update the design variables. Specifically, the continuous adjoint method's algorithm is presented in the Flowchart of Figure 3.3.

The progression of the design variables towards the optimal product depends on the SDs and the value of step η. η is usually an empirically defined number smaller than 1. It prevents dramatic change of b_n between two successive iterations. If η gets smaller than it should, then the algorithm will slow down and the cost of the optimization will rise. On the other hand, a high η value may lead to divergence. The odjective function F defined in eq. 3.6 must be minimized. Therefore, the $\eta\frac{\delta F}{\delta b_n}$ term gets substracted from b_n^{old}. On the contrary, if F is to be maximized, the same term is to be added.

Another important aspect is the termination criterion. A maximum number of iteration is preset. Hence, when the algorithm reaches that limit, the optimization ends. Moreover, although every new airfoil tends to be better than the previous one, the optimization turn-around time may increase a lot. For that reason, a minimun difference between F^{new} and F^{old} may be an additional termination criterion.

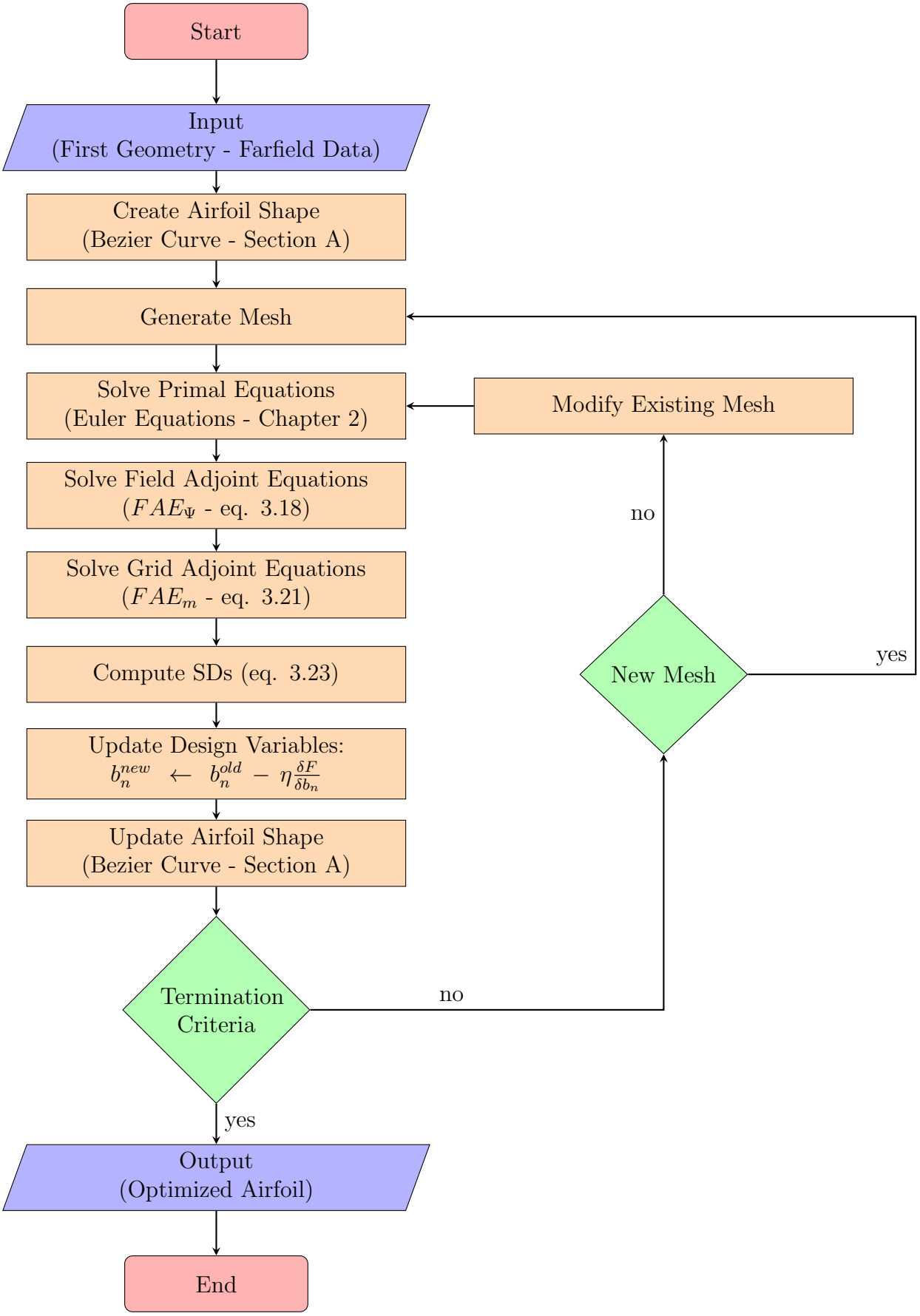


Figure 3.3: *Adjoint-based Optimization Algorithm.*

Chapter 4

Numerical Parametric Analysis

Based on Chapters 2 and 3, a software has been developed (in FORTRAN) for the evaluation of the flow around an airfoil and for the computation of its sensitivity derivatives using the standard (severed) SI and E-SI continuous adjoint method. Three airfoils are used as test-cases, namely the reparameterized NACA 4412, RAE 2822 and FAUVEL [27]. For simplicity reasons, during the following presentation each case is described by the reparameterized airfoil's name. Their shape, bezier points and farfield conditions are presented in Figures 4.1 - 4.3. For the computation of SDs, $w_l = 1,\ w_d = 0$ and $C_{L_{tar}} = 1.2$ (arbitrarily selected) are set. Hence eq. 3.6 becomes

$$F_{obj} = (C_L - 1.2)^2$$

Case 1: NACA 4412

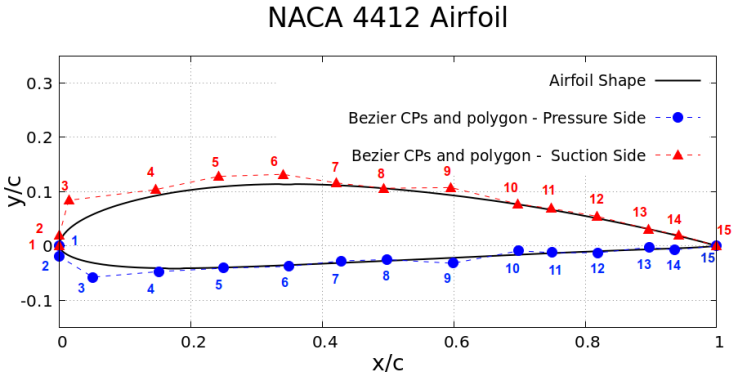
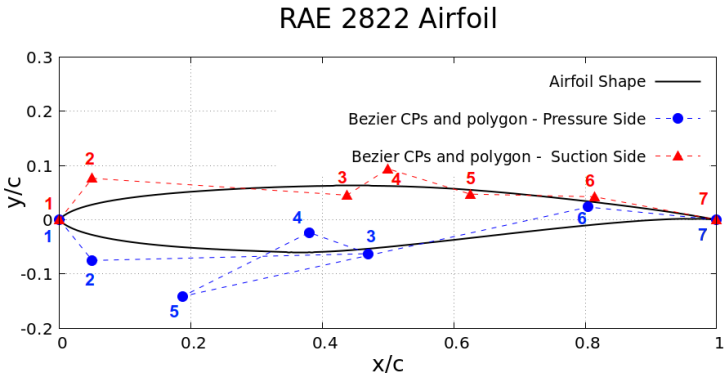


Figure 4.1: *Shape and Bezier Points & Polygon of NACA 4412 airfoil (not in scale).*

Farfield Conditions:

- Angle of Attack = 2^o
- Mach Number = 0.3

Case 2: RAE 2822

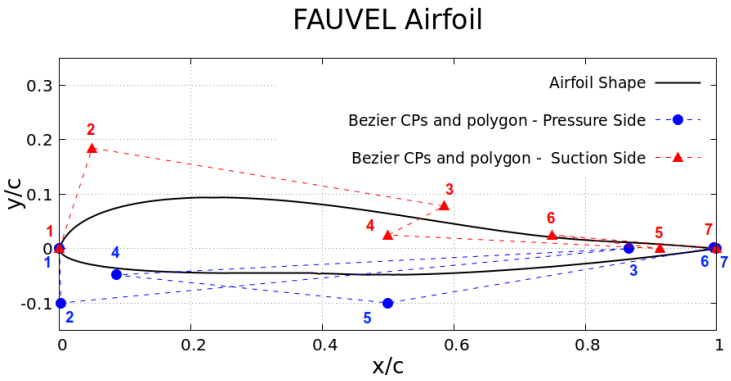


Farfield Conditions:

- Angle of Attack = 3^o
- Mach Number = 0.2

Figure 4.2: *Shape and Bezier Points & Polygon of RAE 2822 airfoil (not in scale),*

Case 3: FAUVEL



Farfield Conditions:

- Angle of Attack = 1^o
- Mach Number = 0.4

Figure 4.3: *Shape and Bezier Points & Polygon of FAUVEL airfoil (not in scale).*

The design variables of each case are the coordinates of the bezier CPs. Starting with pressure side, the first design variable is the y coordinate of the first CP, the second design variable is the x coordinate of the first CP, the third design variable is the y coordinate of the secomd CP and so on. The design variables of the suction side are numbered the same way. For instance, since NACA 4412 has 15 pressure side CPs, the design variable 31 is the y coordinate of the first suction side CP.

In the analysis that follows, the SDs for the coordinates of the leading and trailing edge CPs are not computed . During an optimization these are considered fixed. Thus, their SDs are not used.

4.1 The NACA 4412 Airfoil

4.1.1 Flow Prediction

At first, the flow equations around the airfoil are solved with two different numerical schemes, FVS and Roe's Approximate Riemann Solver. Moreover, first-order accuracy (eq. 2.16) is implemented, as well as second-order accuracy with (eq. 2.18) and without (eq. 2.17) the Van Leer - Van Albada limiter (Limiter). Their convergence paths are shown in Figure 4.4. Only the energy equation residuals are plotted, as they are absolutely representative and, usually, the last to meet the convergence criterion. Hence, it reveals the total number of iterations each case needs to run a flow simulation.

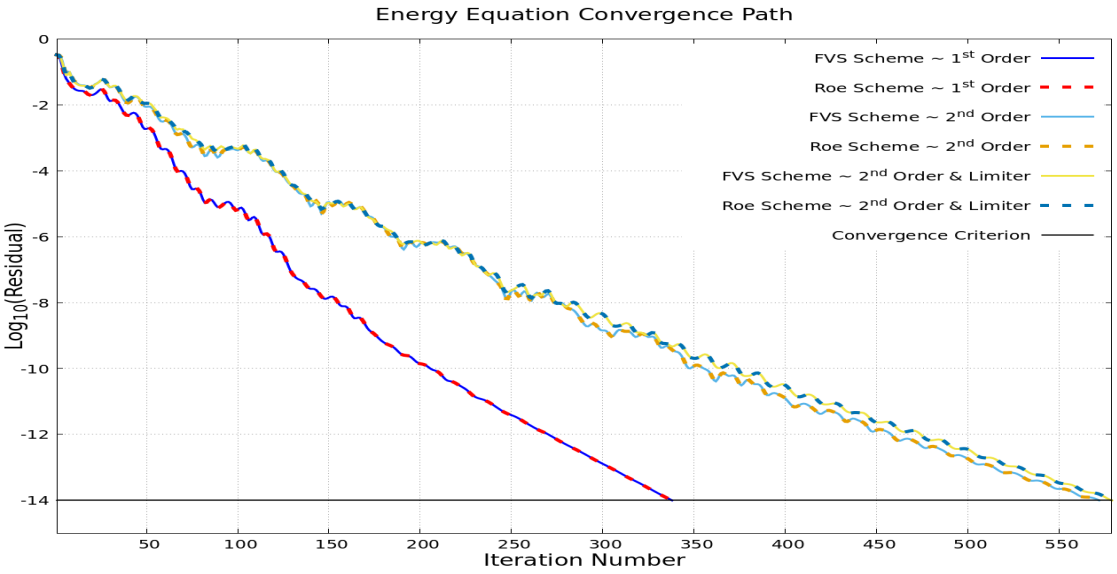
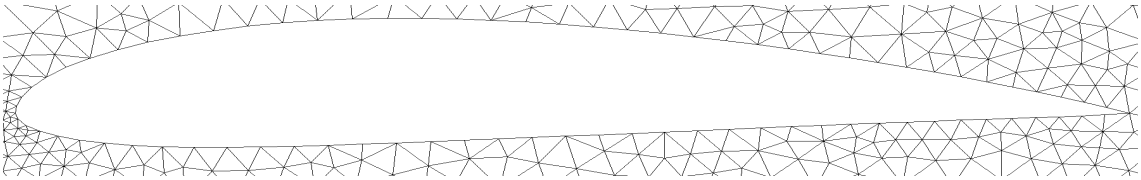


Figure 4.4: *NACA 4412 airfoil. Impact of numerical scheme and order of accuracy on the convergence path.*

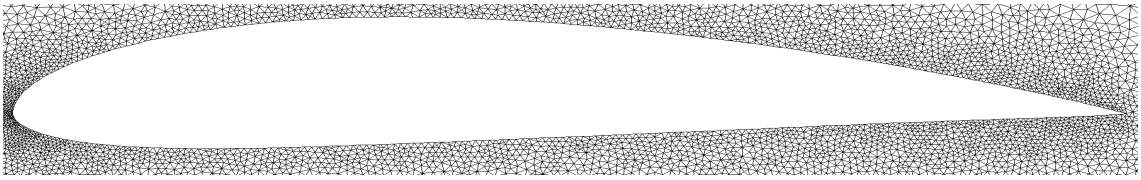
Both FVS and Roe scheme display the exact same convergence path. However, the order of accuracy affects it significantly. A first-order flow prediction is much more economical. It needs 338 iterations to converge. On the other hand, second-order with and without limiter needs 572 and 579 iterations, respectively. Moreover, each second-order scheme iteration is more cost-expensive, because ∇U must be calculated on every segment of the FV. To further investigate the impact of the order of accuracy, each case's results must be examined. To do so, four different grid densities have been used, as described in Table 4.1 and shown in Figure 4.5

Grid	Airfoil Nodes	Nodes	Triangular Cells
A	80	858	1625
B	400	8979	17517
C	540	23317	46013
D	700	39632	78463

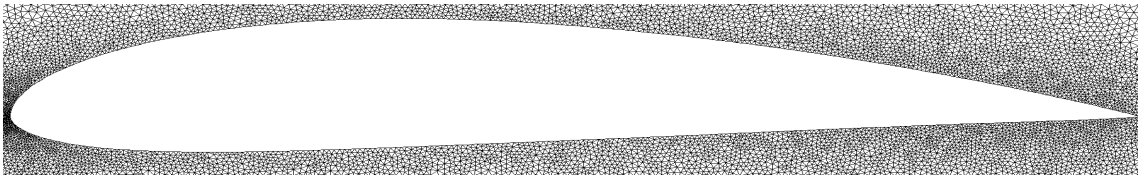
Table 4.1: *NACA 4412 airfoil. Grid characteristics. All unstructured grids comprise only triangular elements.*



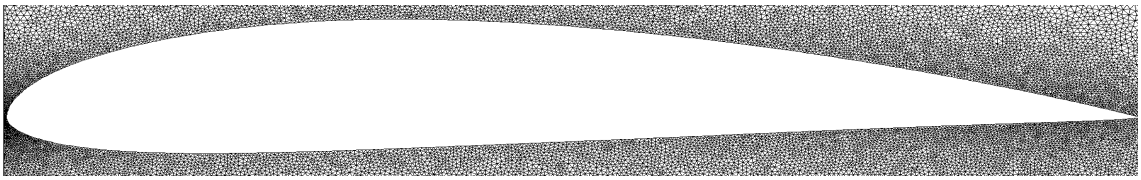
(a) *Grid A*



(b) *Grid B*



(c) *Grid C*



(d) *Grid D*

Figure 4.5: *NACA 4412 airfoil. Close-up views of the three grids used in the vicinity of the airfoil.*

To optimize the airfoil, the objective function (eq. 3.6) depends on both aerodynamic coefficients C_L and C_D (the latter is expected to be zero in a low speed i.e. in the absence of a shock wave, with the assumption of an inviscid fluid flow, excluding form drag). These are computed with the Roe scheme, on the four different grids, with two orders of accuracy, with and without limiter (FVS gives similar results).

Order of Accuracy	Grid	C_L	C_D	Iterations
1	A	$4.948\ 10^{-1}$	$123.5\ 10^{-3}$	101
	B	$6.188\ 10^{-1}$	$53.44\ 10^{-3}$	338
	C	$6.587\ 10^{-1}$	$34.02\ 10^{-3}$	573
	D	$6.849\ 10^{-1}$	$27.59\ 10^{-3}$	830
2	A	$7.231\ 10^{-1}$	$14.54\ 10^{-3}$	172
	B	$7.891\ 10^{-1}$	$3.788\ 10^{-3}$	572
	C	$7.803\ 10^{-1}$	$3.078\ 10^{-3}$	1103
	D	$7.851\ 10^{-1}$	$3.055\ 10^{-3}$	1515
2 & Limiter	A	$6.463\ 10^{-1}$	$31.74\ 10^{-3}$	180
	B	$7.871\ 10^{-1}$	$5.416\ 10^{-3}$	579
	C	$7.779\ 10^{-1}$	$3.433\ 10^{-3}$	1115
	D	$7.840\ 10^{-1}$	$3.283\ 10^{-3}$	1530

Table 4.2: *NACA 4412 airfoil. Lift and Drag coefficient computed on different grids and for two orders of accuracy, with and without limiter. Number of iterations to converge for the same threshold value are included in the last column.*

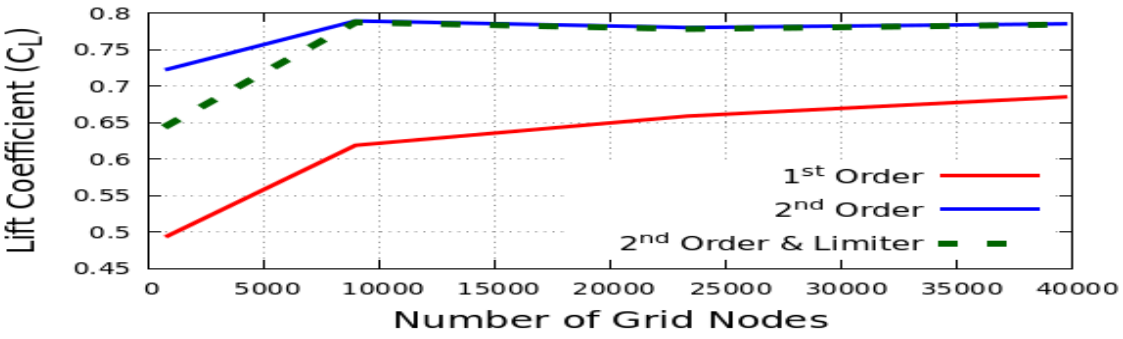


Figure 4.6: *NACA 4412 airfoil. Impact of grid density and order of accuracy on the C_L value.*

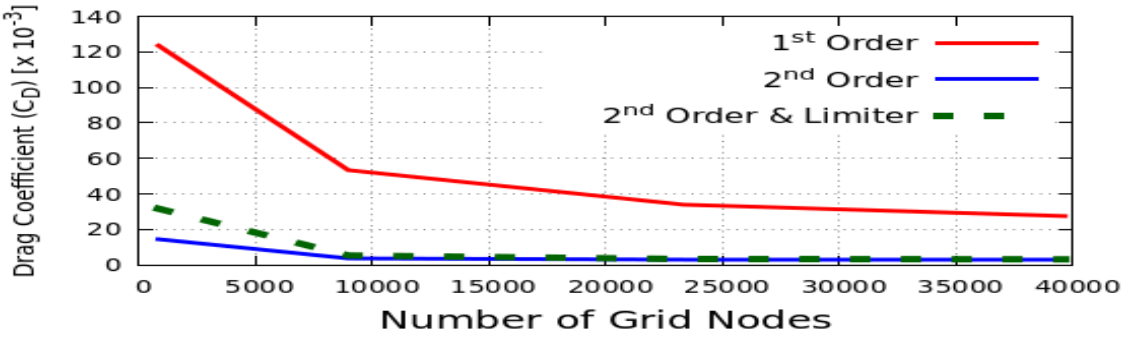


Figure 4.7: *NACA 4412 airfoil. Impact of grid density and order of accuracy on the C_D value.*

Second-order schemes need more iterations to converge and their results differ from those of the first-order (Table 4.2). Moreover, they are practically unaffected from grid density. On the other hand, the flow prediction using first-order schemes computes different flow variables, especially on coarse grids. Their results come closer to those of second-order as the grid density increases a lot (Figures 4.6 and 4.7). This can better be represented by computing the pressure coefficient (Figure 4.8). Both second-order schemes have similar results on a coarse grid.

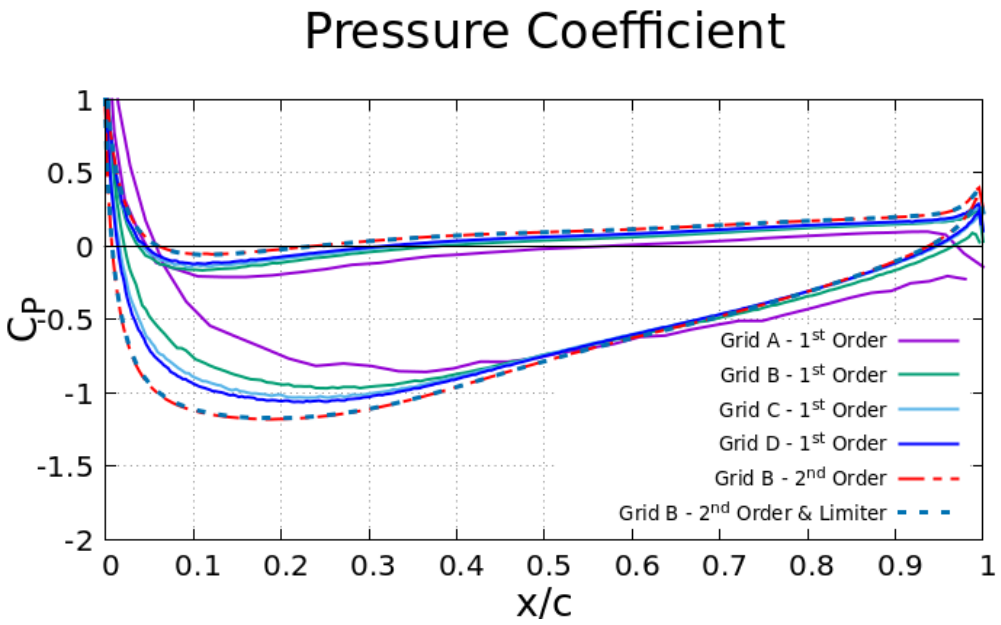


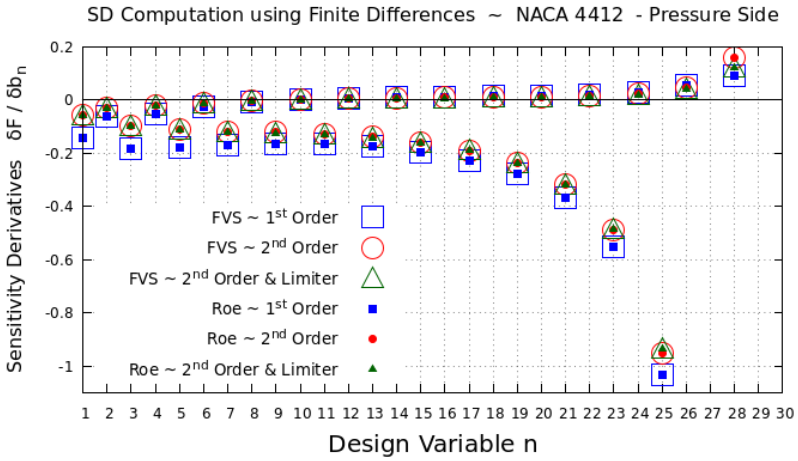
Figure 4.8: *NACA 4412 airfoil. Comparison of pressure distribution over the airfoil contour for different grid sizes and schemes of different accuracy.*

On grid A and first-order scheme, the C_P line is rough. This happens because the number of airfoil nodes is small.

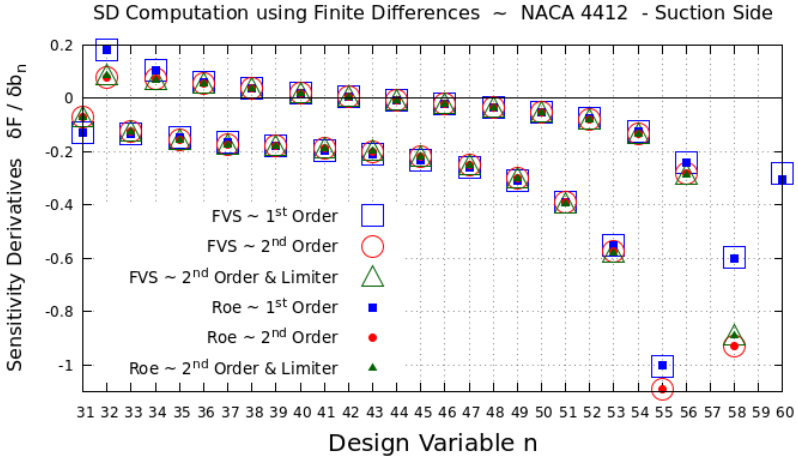
4.1.2 Comparison of SDs

The present diploma thesis focuses on the software development for the computation of each case's SDs using the continuous E-SI adjoint method. However, the same SDs are also computed with Finite Differences, which is assumed to compute reference derivatives' values. The second-order FD scheme (eq. 1.2) is being used with $\epsilon = 10^{-7}$.

SDs are compared for the aforementioned orders of accuracy and numerical schemes. Grid B is used in all these comparisons.



(a) *Pressure side*



(b) *Suction side*

Figure 4.9: *NACA 4412 airfoil. Comparison of the SDs computed with FDs using different orders of accuracy and numerical schemes.*

First- and second-order schemes do not compute the same SDs. However, because the Euler equations are used, only the maximization of lift coefficient is the target and the grid is dense enough, the difference between the SDs computed by first- and second-order schemes is small.

Consequently, the parameters of the adjoint schemes are analyzed and the method's accuracy is compared with FDs. The field adjoint equations (FAE_Ψ) are discretized and solved similar to the primal ones. However, in contrast to eq. 2.7, eq. 3.24 is non-conservative. Thus, the adjoint problem's Roe scheme possesses both a conservative (Cons Roe) and a non-conservative (NonCons Roe) form. The convergence paths of both schemes are shown in Figure 4.10. It must be noted that the adjoint's FVS convergence has no difference from the conservative Roe's. In the analysis that follows, adjoint and primal equations always have consistent schemes.

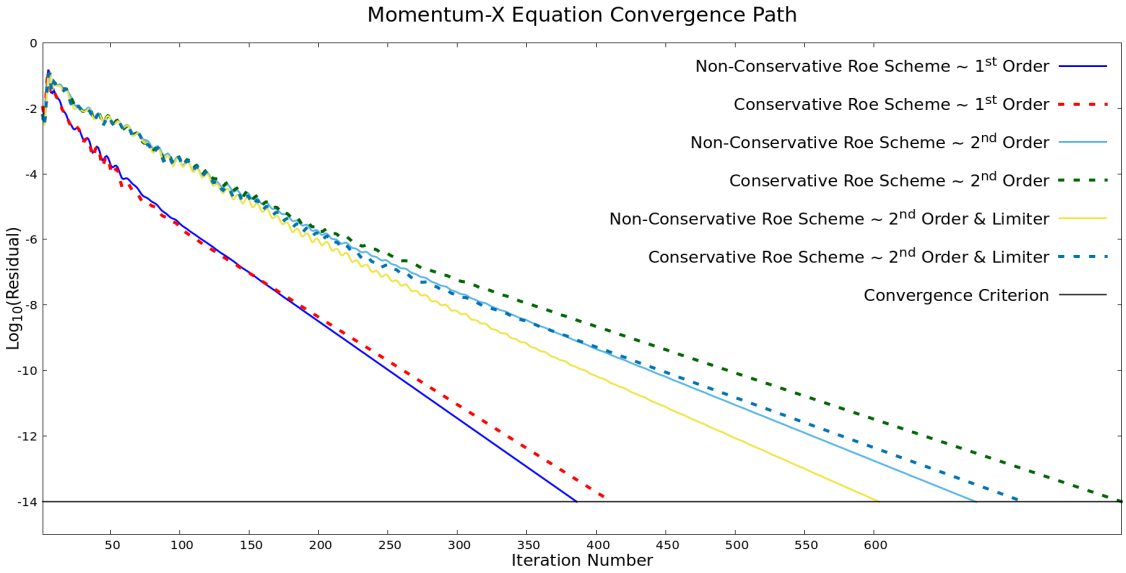


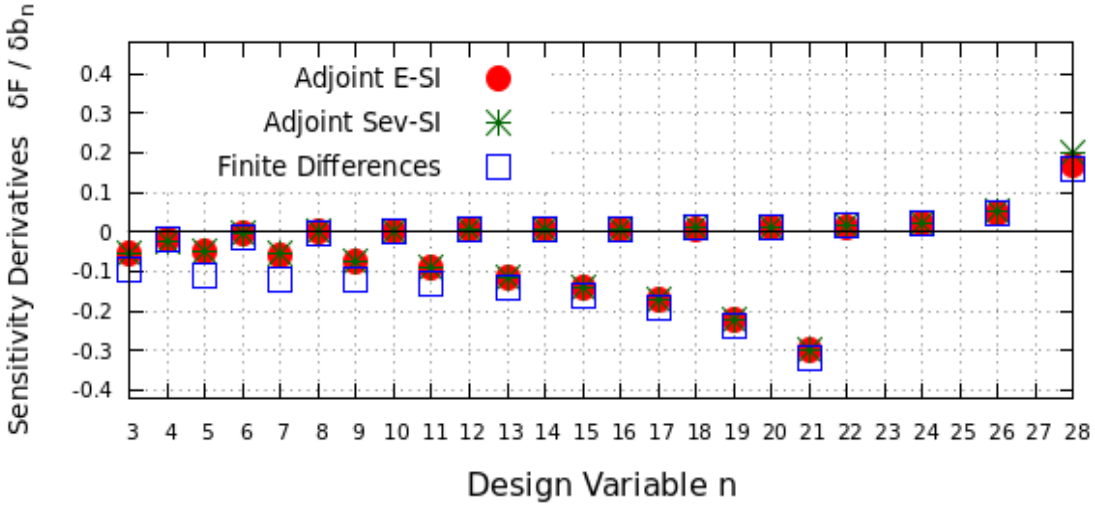
Figure 4.10: *NACA 4412 airfoil. Impact of adjoint numerical scheme and order of accuracy on convergence path.*

In all three cases, the Non-Conservative Roe scheme appears to be cheaper. The number of iterations needed are given in Table 4.3. Especially, with second-order of accuracy, non-conservative Roe is faster by more than 100 iterations.

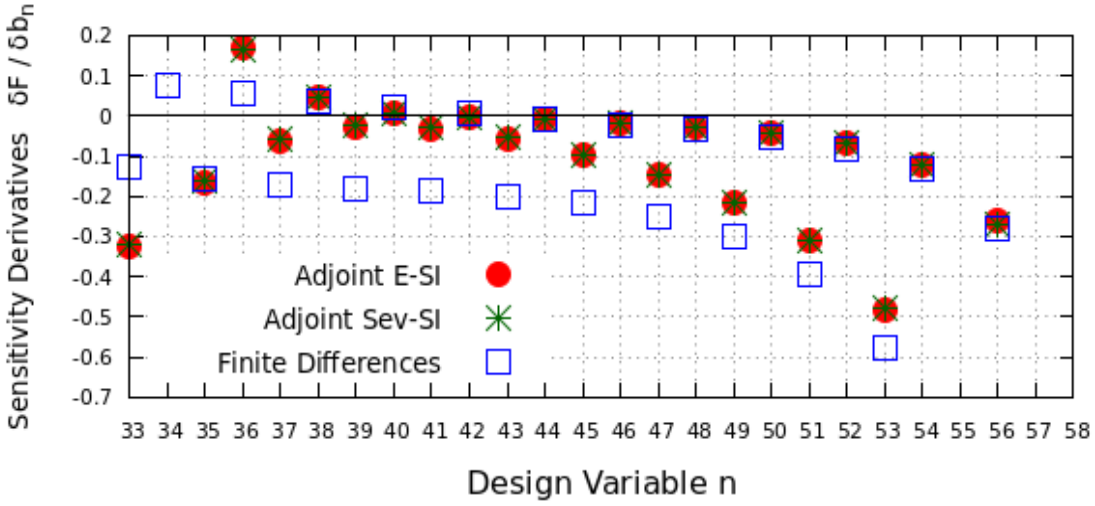
Order of Accuracy	Scheme	Iterations
1	Cons Roe	411
	Non-Cons Roe	386
2	Cons Roe	779
	Non-Cons Roe	674
2 & Lim	Cons Roe	708
	Non-Cons Roe	604

Table 4.3: *NACA 4412 airfoil. Number of iterations needed for the convergence of the adjoint problem.*

In order to compute the SDs with the adjoint method, both the primal and the adjoint equations must be solved. In what follows, adjoint and primal equations always use the same numerical scheme and order of accuracy. Moreover, adjoint SDs are compared with those computed by FDs using the same order of accuracy, too. Finally, zeroing the $SD5$ term in eq. 3.23, the Severed-SI's (SevSI) SDs emerge.

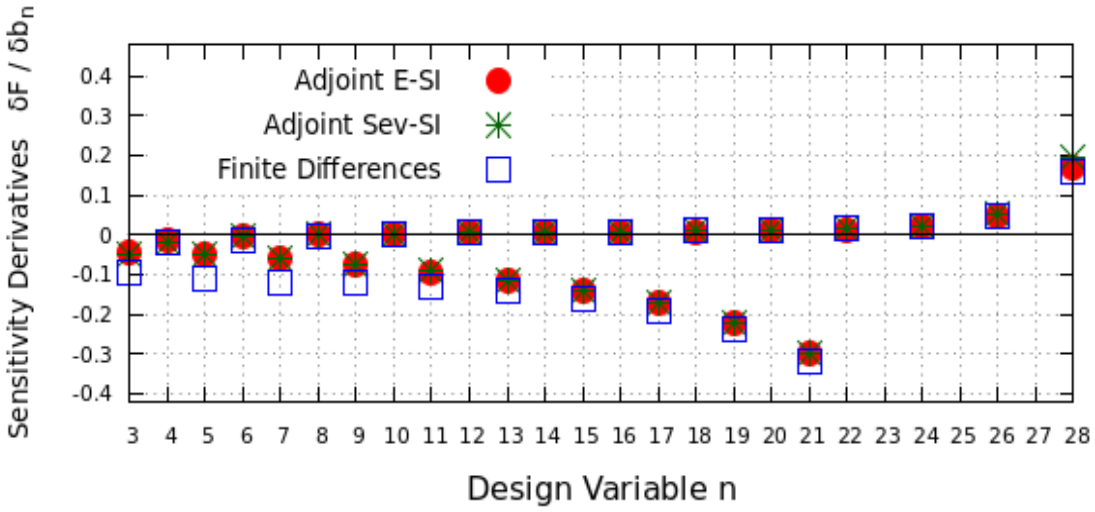


Pressure side SDs

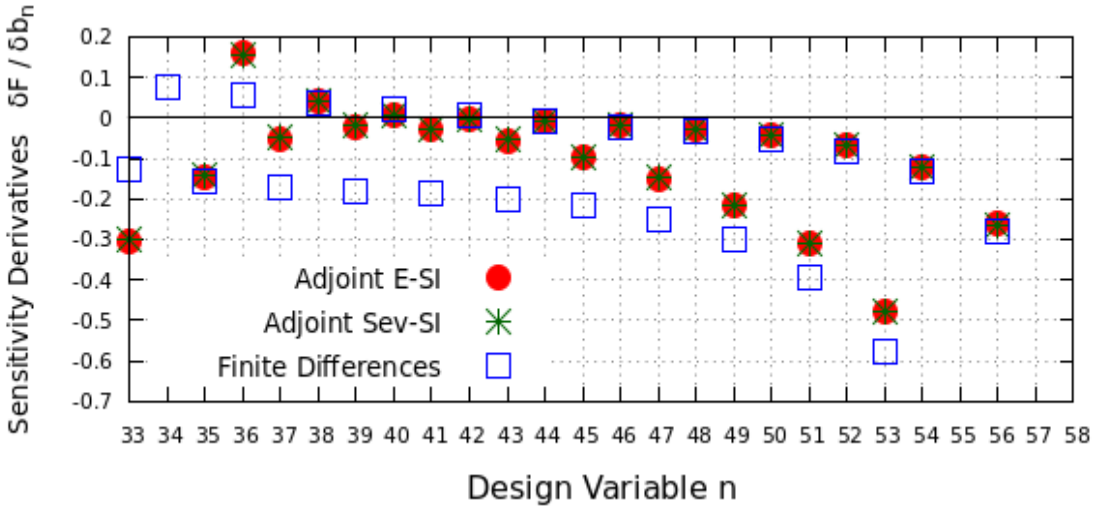


Suction side SDs

Figure 4.11: *NACA 4412 airfoil. Comparison of SDs computed by adjoint FVS scheme and* 2^{nd} *order scheme.*

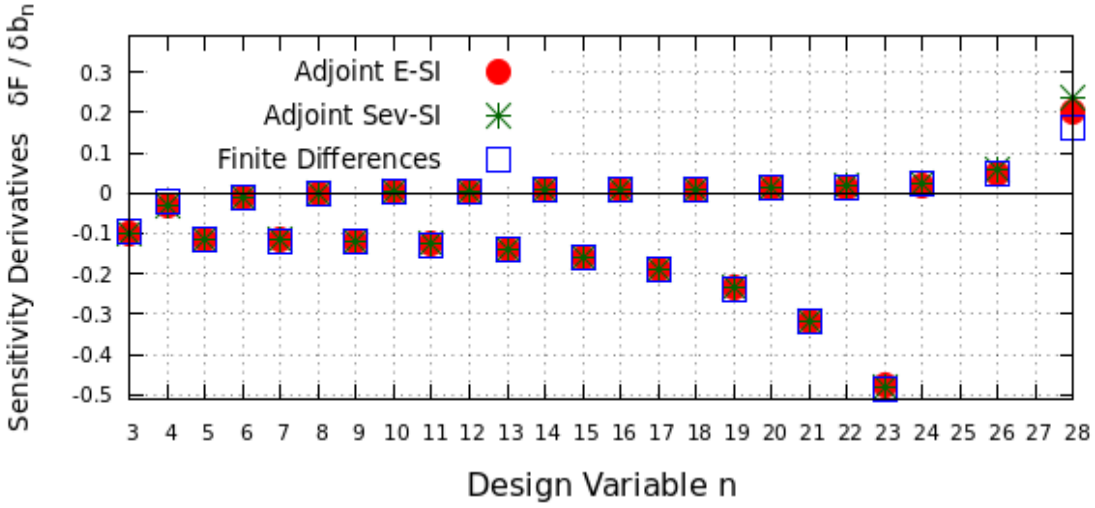


Pressure side SDs

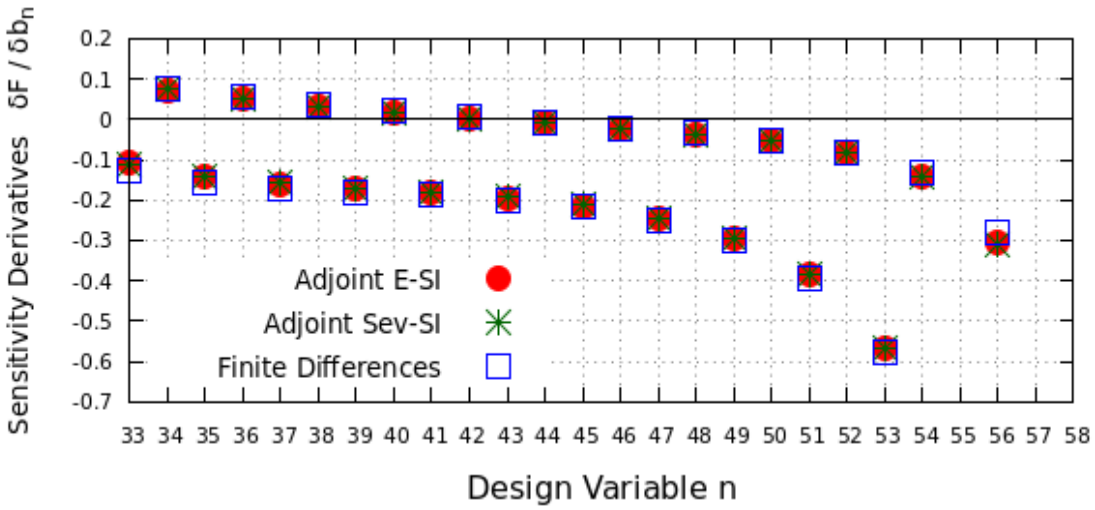


Suction side SDs

Figure 4.12: *NACA 4412 airfoil. Comparison of SDs computed by adjoint conservative Roe scheme and* 2^{nd} *order scheme.*



Pressure side SDs

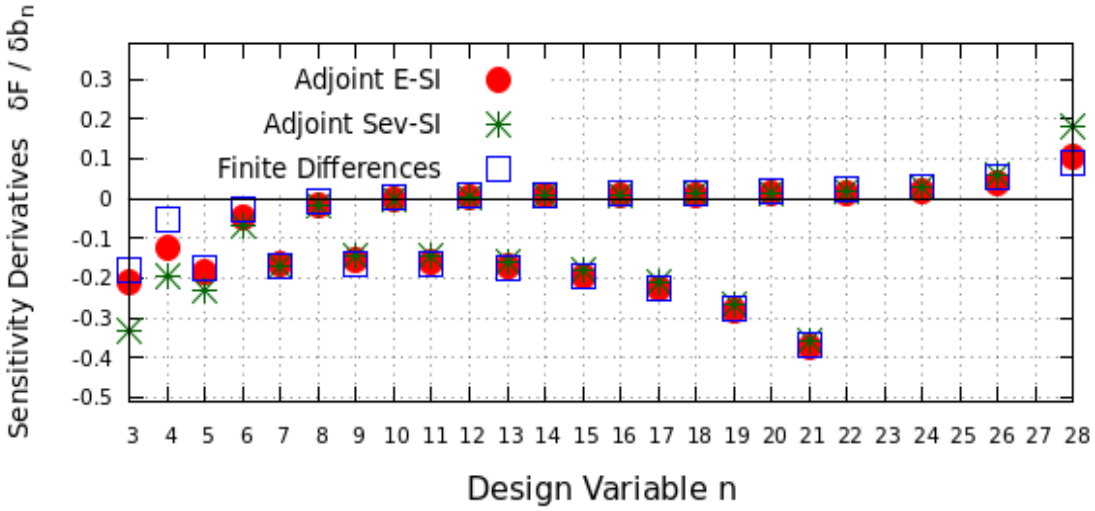


Suction side SDs

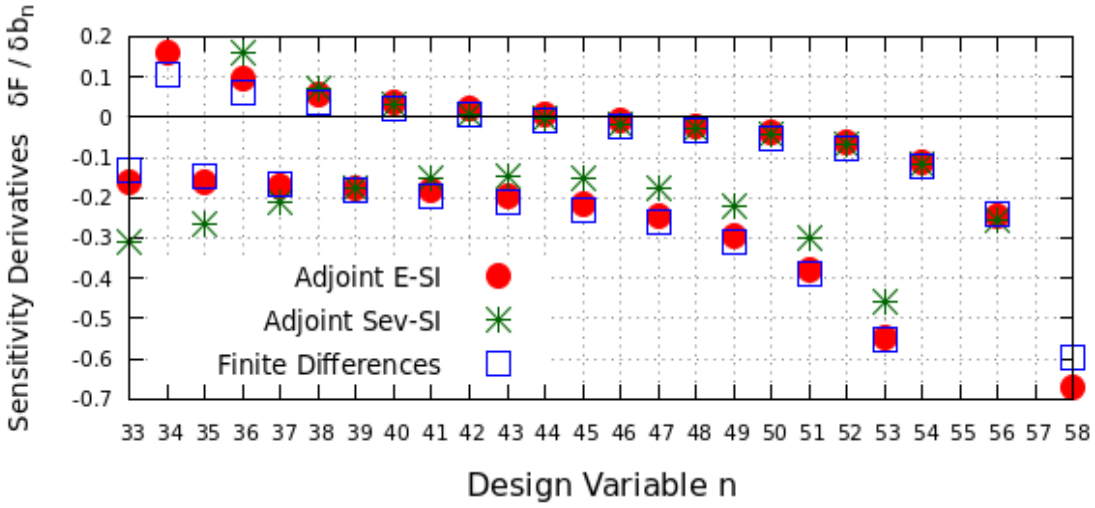
Figure 4.13: *NACA 4412 airfoil. Comparison of SDs computed by Adjoint non-sonservative Roe and 2^{nd} order scheme.*

All the SDs presented in Figures 4.11 to 4.13 are computed by second-order schemes without a limiter. The main goal is the adjoint SDs be close to FDs. In this way, a N-independent, i.e. less cost-expensive, optimization algorithm can be developed, as described in Chapter 1. The first two schemes, FVS and Conservative Roe, do not exhibit a good enough performance. For many design variables the difference between the adjoint and FD SDs is big, especially on the suction side. On the other hand, all the derivatives have the correct sign. Thus, an optimization is possible, but it might be less efficient. The main reason for this is that the adjoint equations

have a non-conservative form. Because of that, a mathematical error emerges during the discretization using conservative schemes, which for many SDs is not negligible. In support of this view, the Non-Conservative Roe scheme results to much more accurate SDs. In Figures 4.14 and 4.15, the SDs computed by the Non-Conservative Roe scheme with first- and second-order accuracy with limiter are presented.

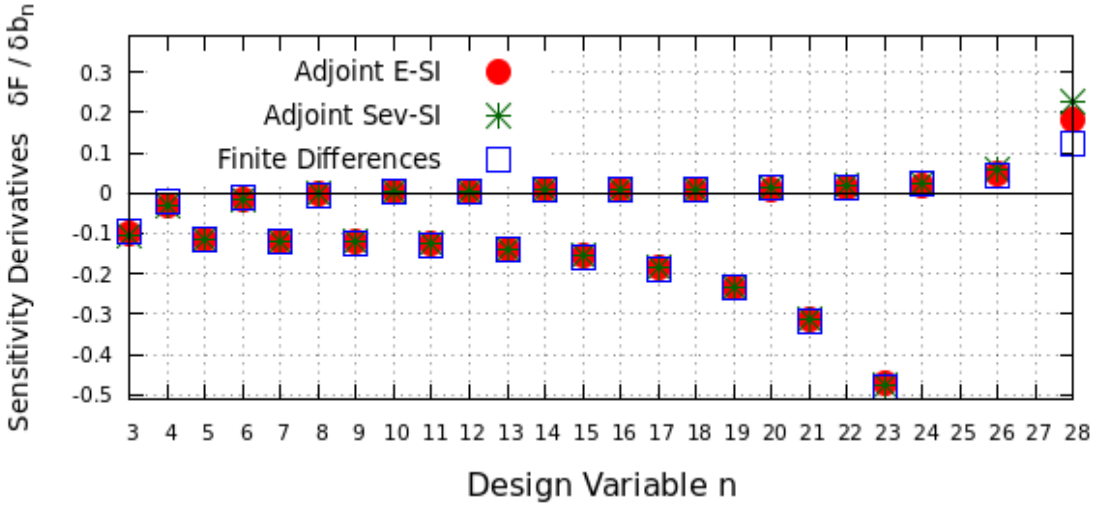


Pressure side SDs

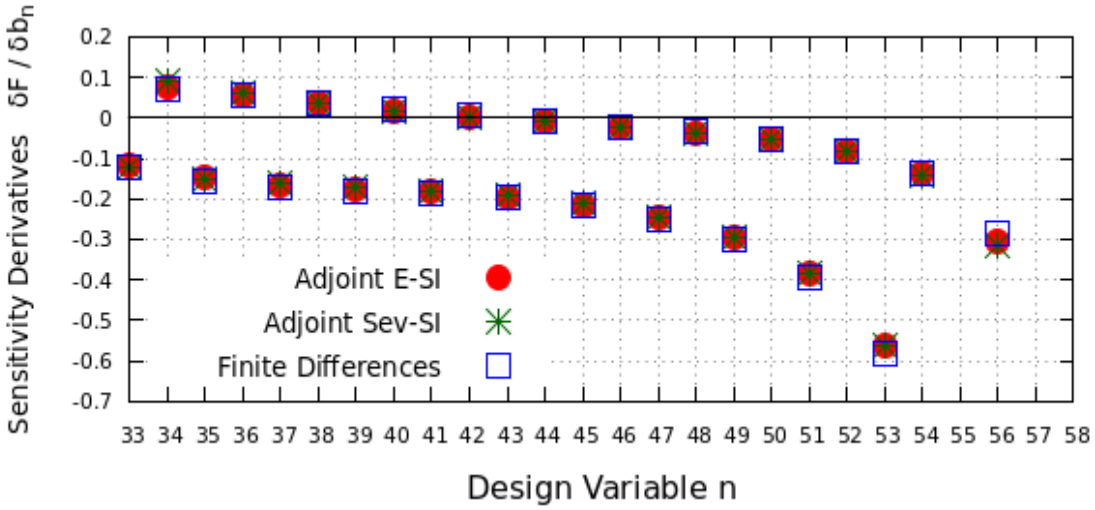


Suction side SDs

Figure 4.14: *NACA 4412 airfoil. Comparison of SDs computed by adjoint non-conservative Roe and* 1^{st} *order scheme.*



Pressure side SDs



Suction side SDs

Figure 4.15: *NACA 4412 airfoil. Comparison of SDs computed by adjoint non-conservative Roe and 2^{nd} order scheme & limiter.*

The first-order discretization doesn't always perform as well as the second one, especially close to the leading and trailing edge (the first and last design variables of each diagram). Concerning the two adjoint methods, Sev-SI and E-SI, no major differences can be noticed. Only in the computations of very few SDs, using first-order schemes, E-SI is more accurate. The Limiter in second-order of accuracy doesn't noticeably affect the results. It has great influence on transonic flows [28].

4.2 The RAE 2822 Airfoil

4.2.1 Flow Evaluation

As in the previous case, the flow around the airfoil is predicted using the same numerical schemes and discretization orders. Each primal solution's convergence history is presented in Figure 4.16.

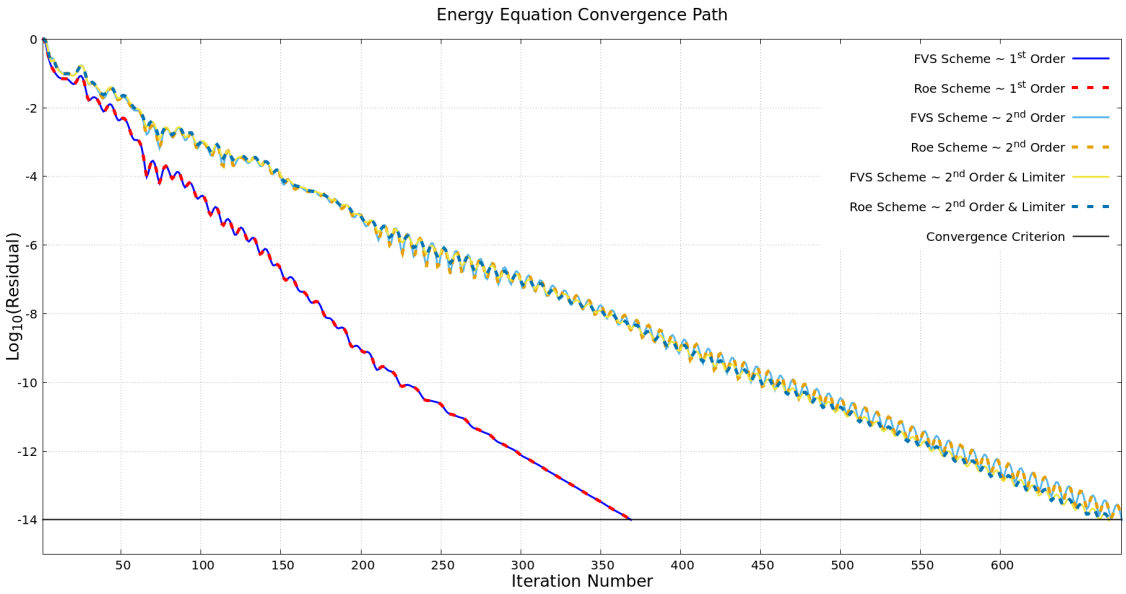


Figure 4.16: *RAE 2822 airfoil. Impact of numerical scheme and order of accuracy on convergence path.*

Once again, it can be seen that Roe and FVS follow similar convergence paths. The second-order schemes are more cost-expensive. Furthermore, the limiter doesn't have a great influence. To compare the results, four different grids, Table 4.4, are used. For each grid, only the Roe scheme and the first- and second-orders of accuracy without a limiter are used. The two force coefficients are shown in Table 4.5.

Grid	Airfoil Nodes	Nodes	Triangular Cells
A	80	864	1637
B	400	9345	18249
C	540	14674	28833
D	700	40448	80095

Table 4.4: *RAE 2822 airfoil. Grid Characteristics. All unstructured grids comprise only triangular elements.*

Order of Accuracy	Grid	C_L	C_D
1	A	$4.320\ 10^{-1}$	$106.2\ 10^{-3}$
	B	$4.911\ 10^{-1}$	$46.32\ 10^{-3}$
	C	$5.364\ 10^{-1}$	$31.15\ 10^{-3}$
	D	$5.500\ 10^{-1}$	$26.88\ 10^{-3}$
2	A	$5.496\ 10^{-1}$	$23.22\ 10^{-3}$
	B	$6.258\ 10^{-1}$	$5.187\ 10^{-3}$
	C	$6.282\ 10^{-1}$	$3.704\ 10^{-3}$
	D	$6.261\ 10^{-1}$	$3.618\ 10^{-3}$

Table 4.5: *RAE 2822 airfoil. Lift and Drag Coefficient computed on different grids and for two orders of accuracy.*

As expected first- and second-order have different results. For the following parametric analysis grid B is used.

4.2.2 Comparison of SDs

Different numerical schemes and different discretization orders are applied to compute the airfoil's SDs with second-order FD.

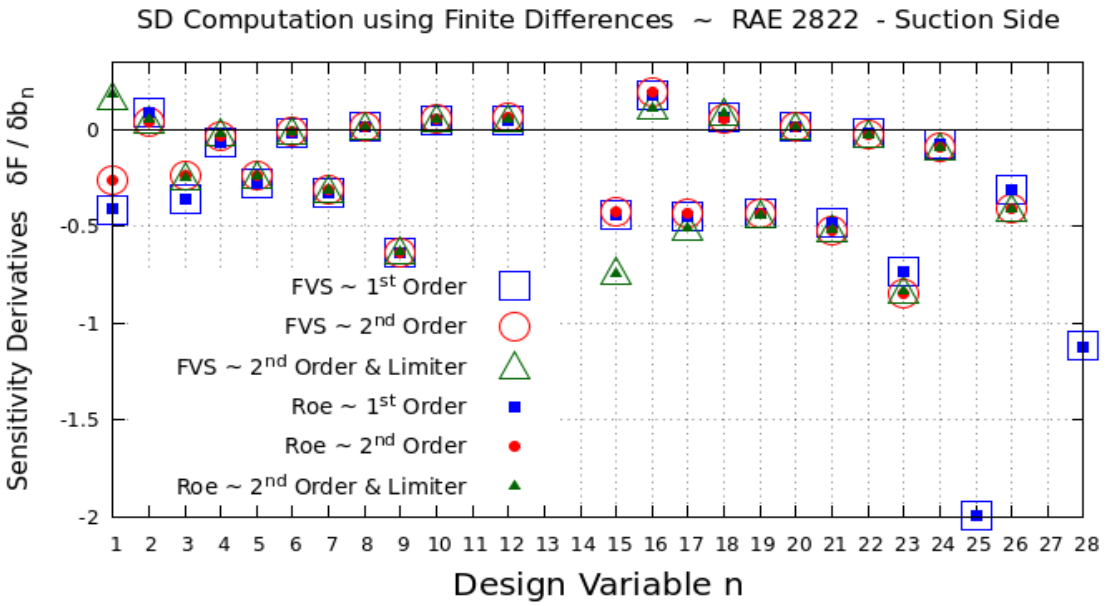


Figure 4.17: *RAE 2822 airfoil. Comparison of the SDs computed with FD using different orders of accuracy and numerical schemes.*

Based on Figure 4.17, FVS and Roe have no differences. The limiter has almost no effect. The implementation of a second-order scheme computes different SDs than those computed by the first-order scheme.

For the sake of completeness, the convergence path of the adjoint equations is presented in Figure 4.18. Conclusions are the same with those of the previous case. First-order discretization converges much faster. Moreover, Non-Conservative Roe scheme needs fewer iterations to meet the convergence criterion than Conservative Roe.

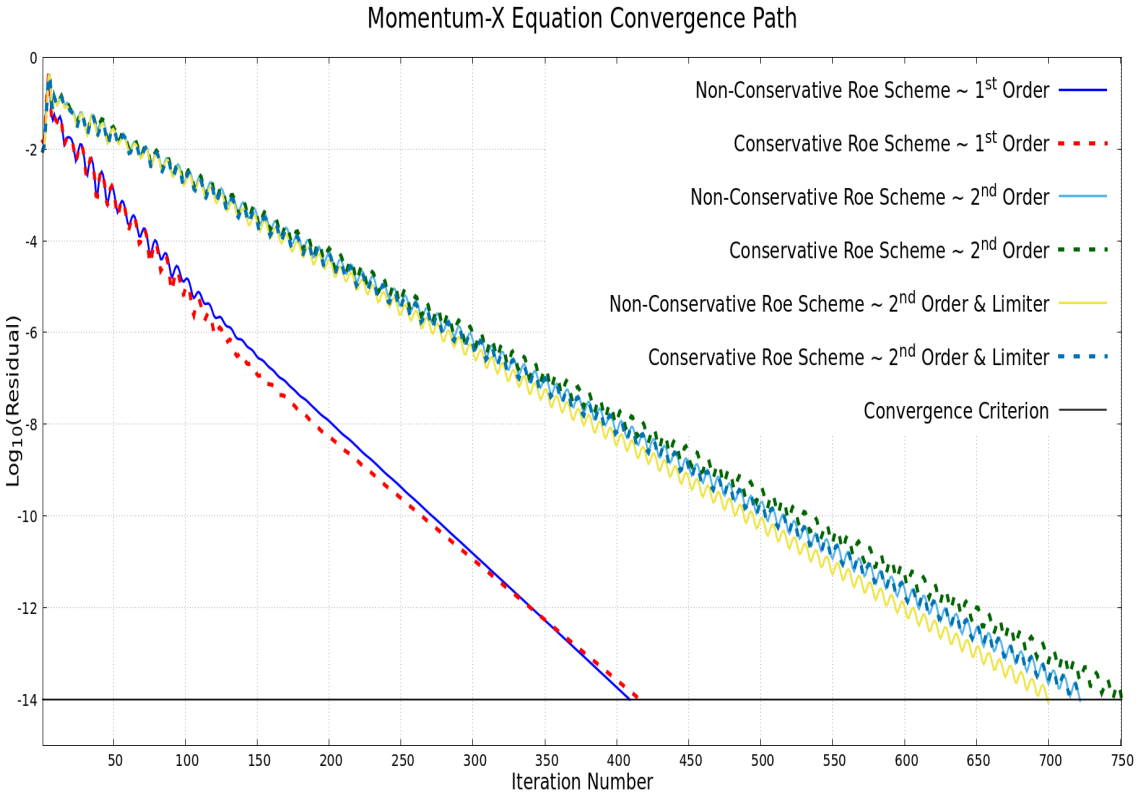
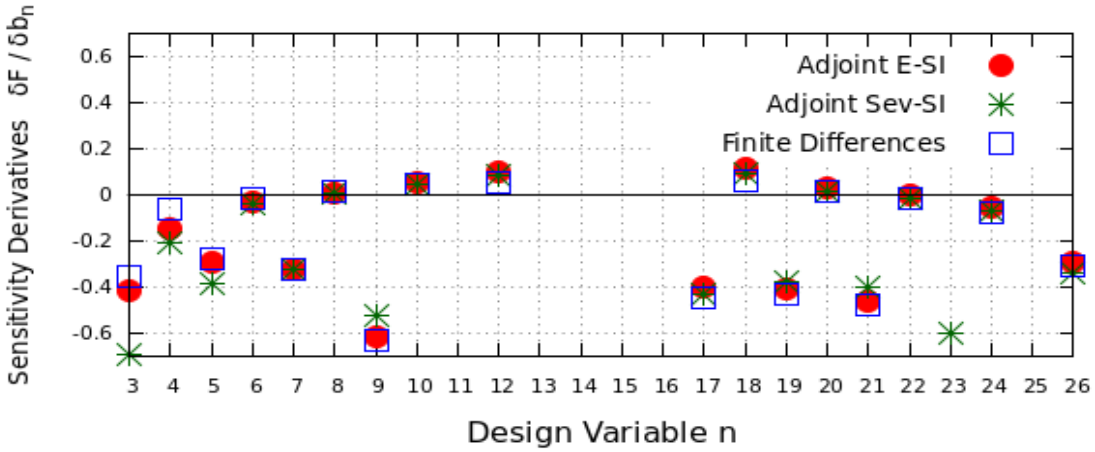
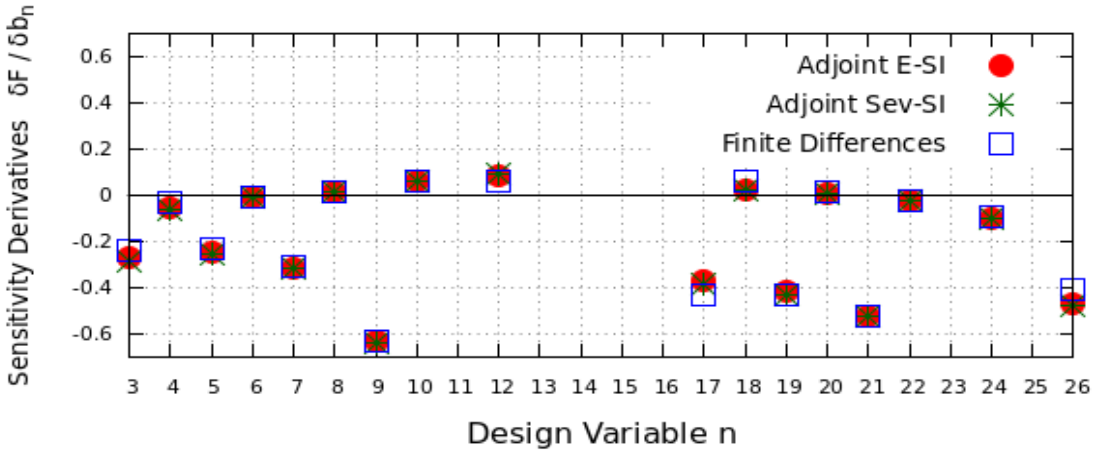


Figure 4.18: *RAE 2822 airfoil. Impact of adjoint numerical scheme and order of accuracy on convergence path.*

A comparison of the adjoint's SDs with those emerging from FDs takes place in case of the RAE 2822 airfoil, too. The parametric analysis confirms previous findings. As presented in Figures 4.19 and 4.20, conservative schemes don't perform well enough on computing SDs. On the other hand, non-conservative Roe computes much more accurate SDs. Finally, the use of the E-SI adjoint doesn't increase the accuracy of SDs, especially if second-order schemes are implemented.

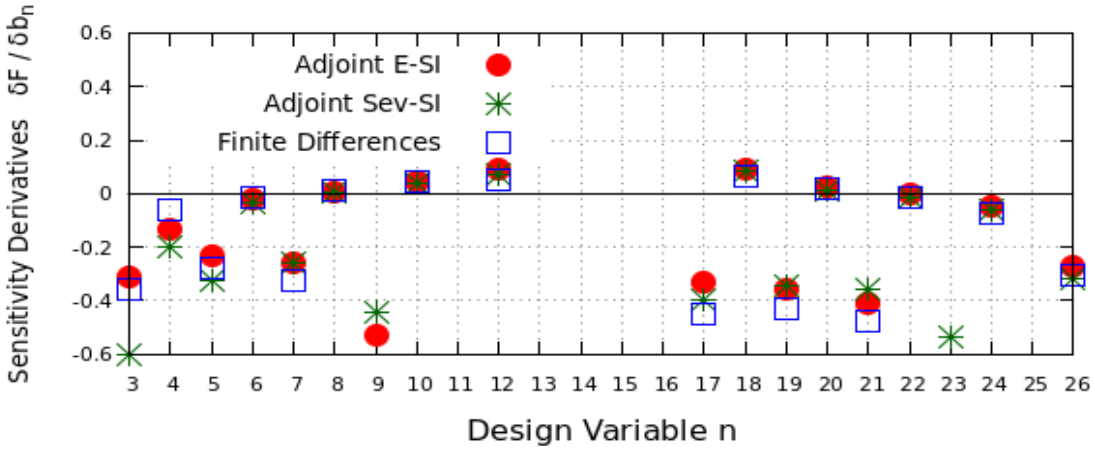


1^{st} order of accuracy SDs

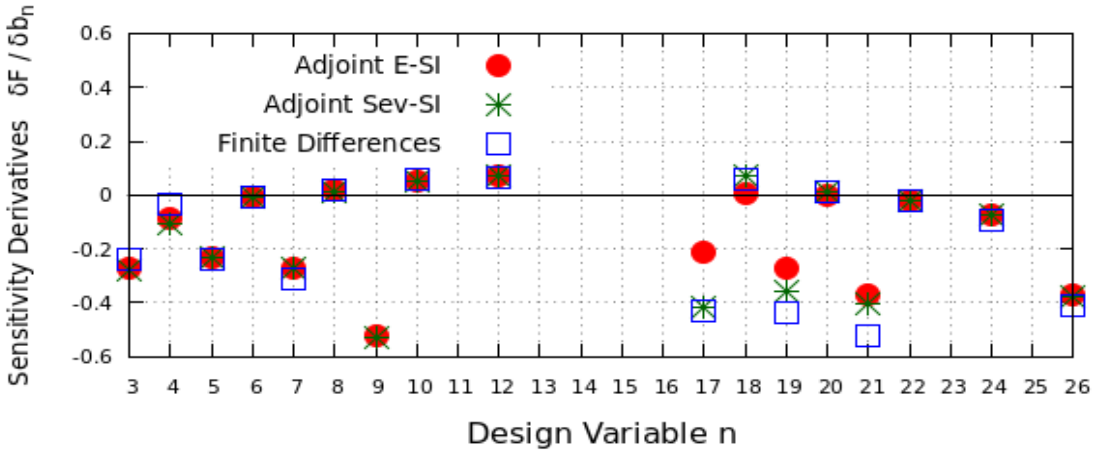


2^{nd} order of accuracy SDs

Figure 4.19: *RAE 2822 airfoil. Comparison of SDs computed by adjoint non-conservative Roe.*



1^{st} order of accuracy SDs



2^{nd} order of accuracy SDs

Figure 4.20: *RAE 2822 airfoil. Comparison of SDs computed by adjoint conservative Roe.*

4.3 The FAUVEL Airfoil

4.3.1 Flow Evaluation

As Figure 4.21 indicates, the implementation of first-order of accuracy leads to a less cost-expensive flow evaluation. However, comparing each order's pressure coefficient on the surface of the airfoil, (Figure 4.22) it is clear that first- and second-order schemes do not produce the same results.

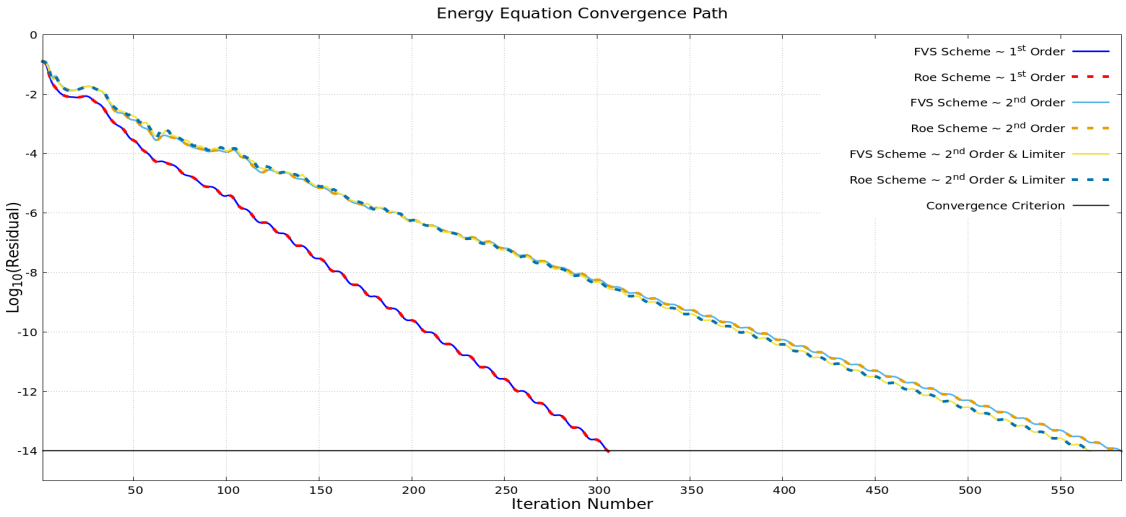


Figure 4.21: *FAUVEL airfoil. Impact of numerical scheme and order of accuracy on convergence path.*

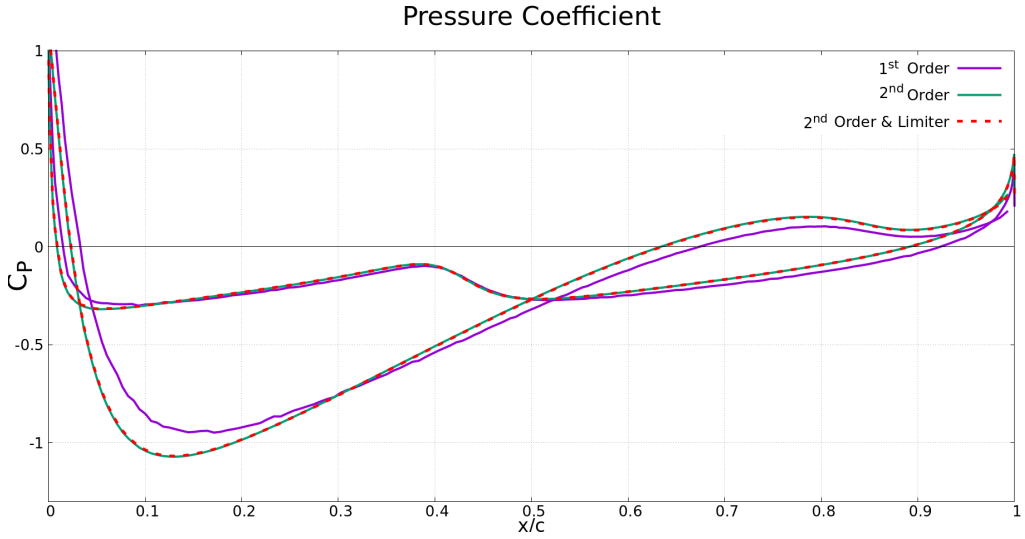


Figure 4.22: *FAUVEL airfoil. Comparison of pressure distribution over the airfoil contour for different order of accuracy.*

4.3.2 Comparison of SDs

Likewise the previous airfoils, the SDs are computed with both numerical schemes and all discretization orders. Based on Firgure 4.23, the SDs computed with FD and the implementation of first-order scheme are different from those computed with second-order scheme.

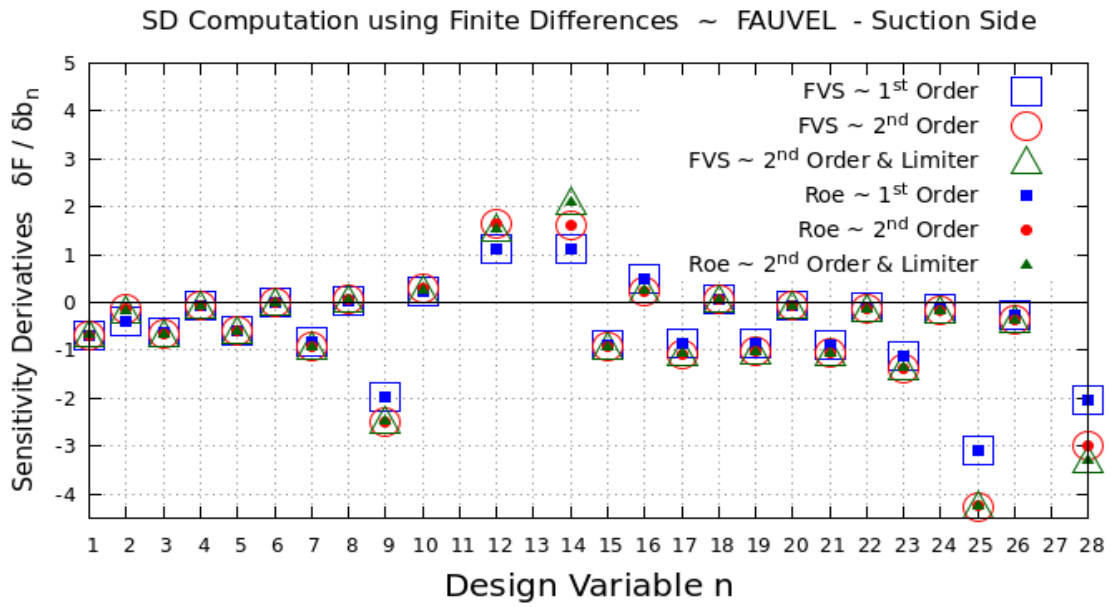


Figure 4.23: *FAUVEL airfoil. Comparison of the SDs computed with FD using different orders of accuracy and numerical schemes.*

Finally, in order to asses the developed software on a third case and analyse the impact of the adjoint methods and numerical schemes, the SDs are computed using eq. 3.23. The convergence paths of the two numerical schemes used for the adjoint problem are presented in Figure 4.24 (FVS has the same path as Conservative Roe).

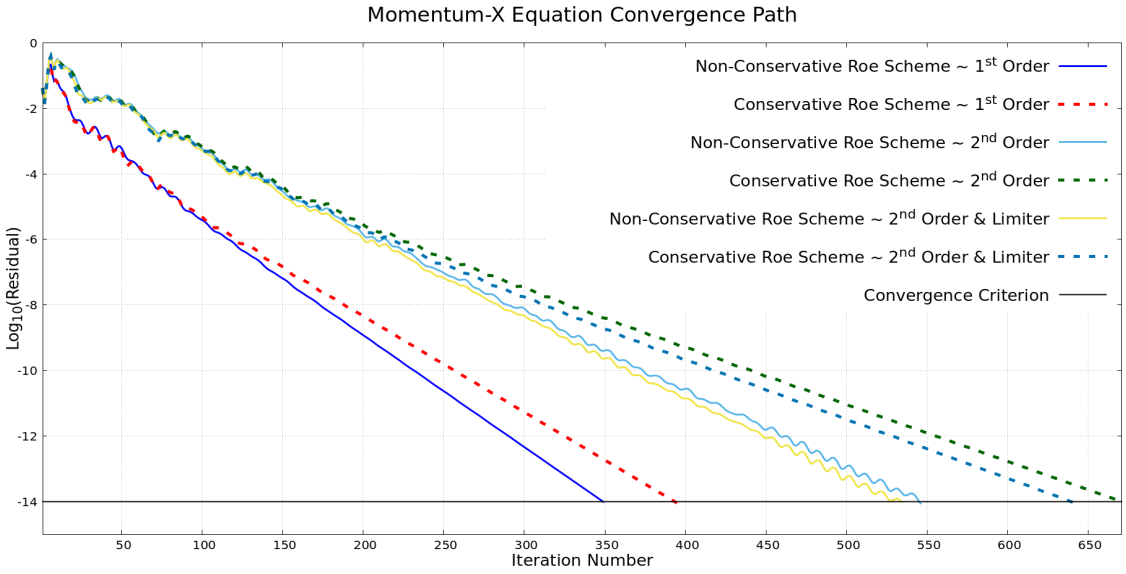
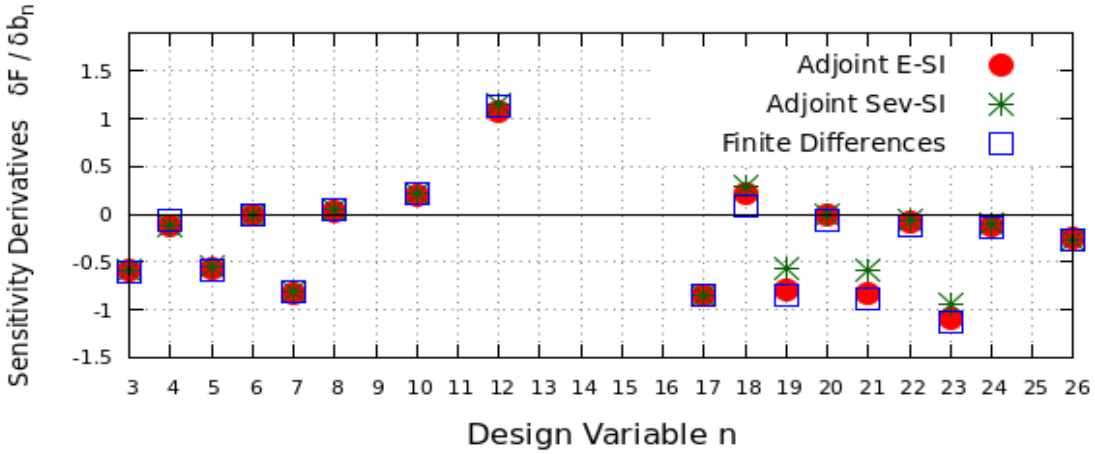
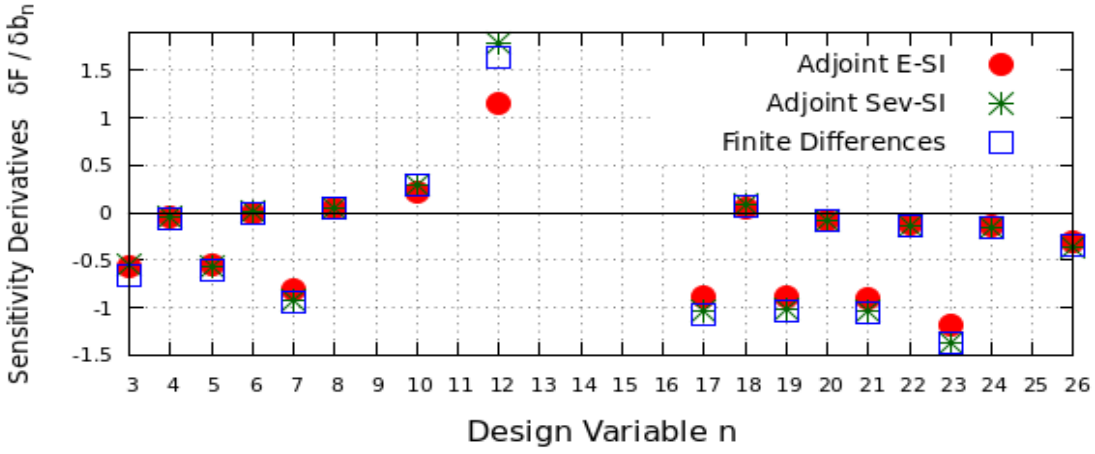


Figure 4.24: *FAUVEL airfoil. Impact of adjoint numerical scheme and order of accuracy on convergence path.*

As expected, the Non-Conservative Roe converges faster for every order of accuracy. The SDs computed can be seen in Figures 4.25 and 4.26. Non-Conservative Roe has indisputably better approach. Another important conclusion is that the E-SI and Sev-SI adjoint methods do not differ noticeably. The only exceptions are some design variables close to leading and trailing edge of the airfoil.

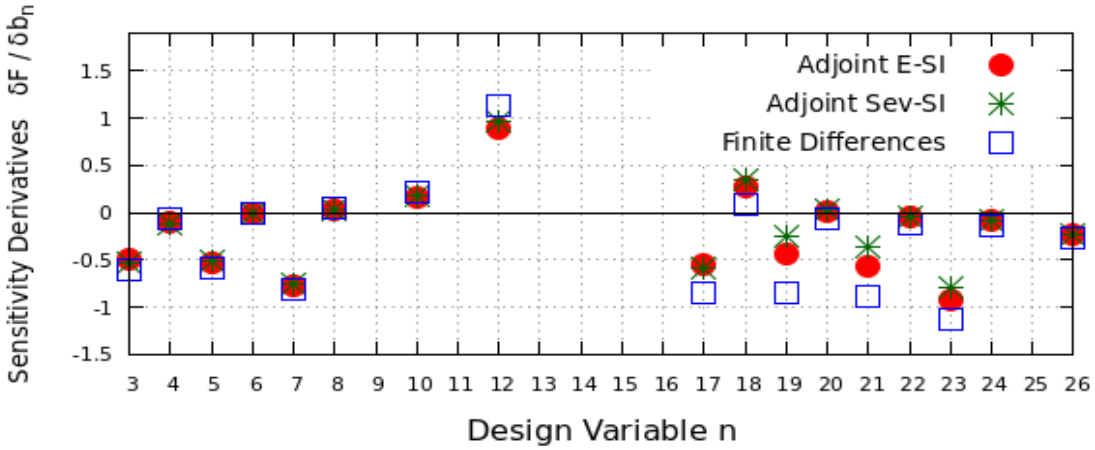


1^{st} order of accuracy SDs

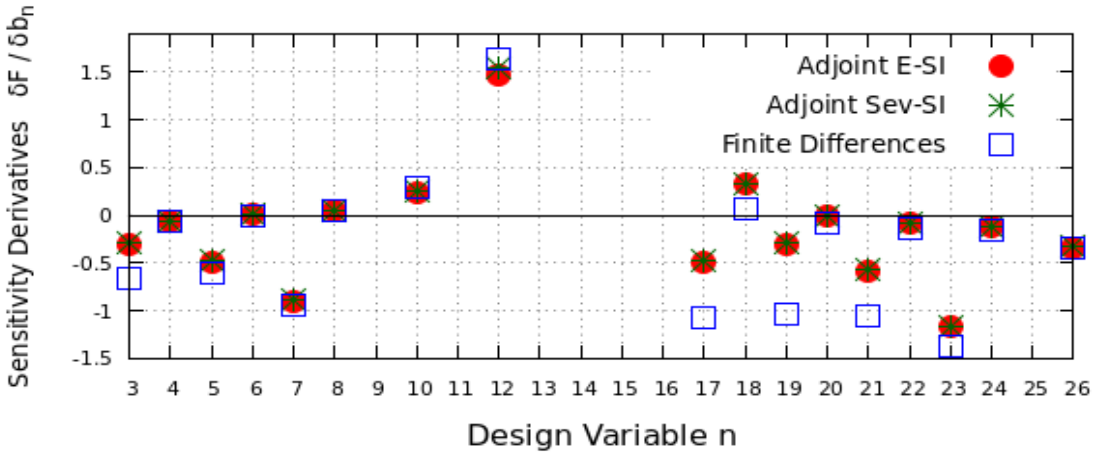


2^{nd} order of accuracy SDs

Figure 4.25: *FAUVEL airfoil. Comparison of SDs computed by adjoint non-conservative Roe.*



1^{st} order of accuracy SDs



2^{nd} order of accuracy SDs

Figure 4.26: *FAUVEL airfoil.Comparison of SDs computed by adjoint conservative Roe.*

Chapter 5

E-SI Adjoint-Based Shape Optimization

In this chapter, the three aforementioned reparameterized airfoils are optimized. Their farfield conditions are the ones mentioned in Chapter 4.

5.1 Adjoint Optimization Parameters

Because the cost of computing all SDs is constant and doesn't depend on N if the adjoint method is used, it makes it suitable for large scale optimization problems. However, there are other parameters that need to be selected in order to achieve an efficient optimization.

To begin with, the numerical schemes need to be determined. Concerning the adjoint equations, non-conservative Roe results to a better approximation of SDs and needs less iterations to converge, in comparison with FVS and conservative Roe. This makes it the best choice. Although the primal schemes do not have any noticeable differences, Roe is implemented. The reason for that is for adjoint and primal equations to have consistent schemes. Furthermore, second-order of accuracy is selected without limiter.

Moreover, the E-SI adjoint method is applied, although the SDs do not differ much from those emerging from the adjoint Sev-SI method. In more complicated cases, where the flow is viscous and the objective functions doesn't consist only from lift coefficient, E-SI adjoint method is noticeably more accurate than Sev-SI [17], [13], [14].

5.2 The Optimization Process

In this section, the three airfoils presented in Chapter 4 are optimized. The goal is to increase their lift by either 20% or 40%.

5.2.1 20% Lift increase

Based on eq. 3.6, that has been programmed in the developed software, the objective functions are presented on Table 5.1

Reparameterized Airfoil	Starting C_L	Target C_L	Objective Function
NACA 4412	0.789	0.94	$F^1_{obj} = (C_L - 0.94)^2$
RAE 2822	0.625	0.75	$F^2_{obj} = (C_L - 0.75)^2$
FAUVEL	0.161	0.2	$F^3_{obj} = (C_L - 0.2)^2$

Table 5.1: *The objective function, based on eq. 3.6, used to optimize NACA 4412, RAE 2822 and FAUVEL airfoils. The target is to increase C_L by 20%.*

The optimization process terminates if

$$|F_i - F_{i-1}| < 10^{-4}$$

where F is the objective function and i the index of the optimization cycle. All three cases optimization histories are presented in Figure 5.1. $\eta = 10^{-3}$ for all three optimizations.

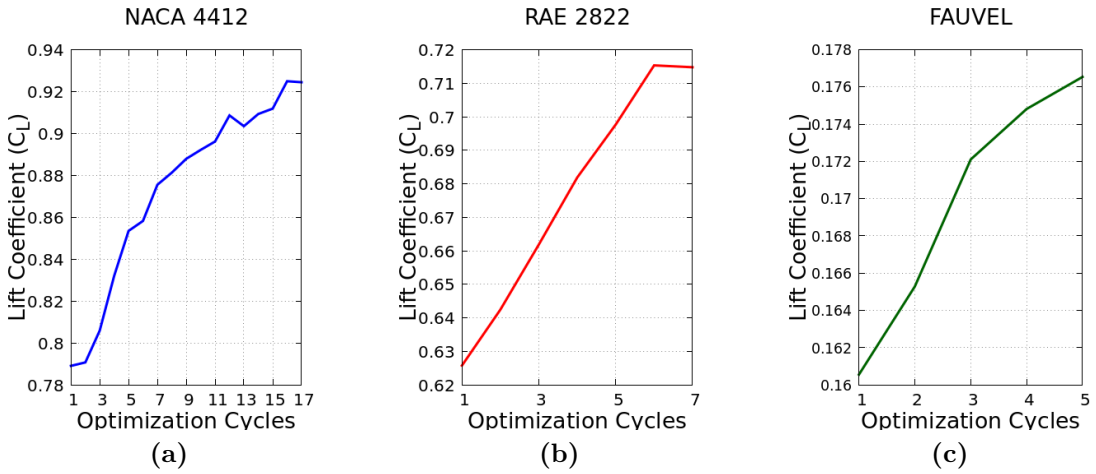
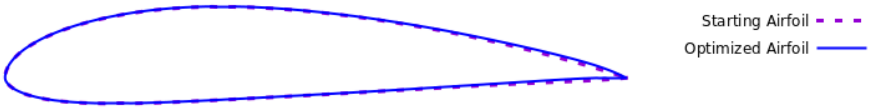
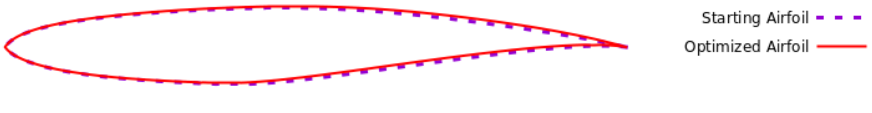


Figure 5.1: *The three optimization paths. The target is to increase C_L by 20%.*

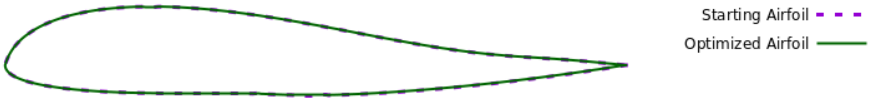
In Figure 5.2 the optimized airfoils are presented.



(a) *NACA 4412*



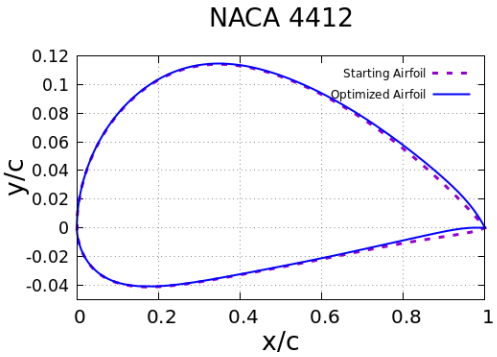
(b) *RAE 2822*



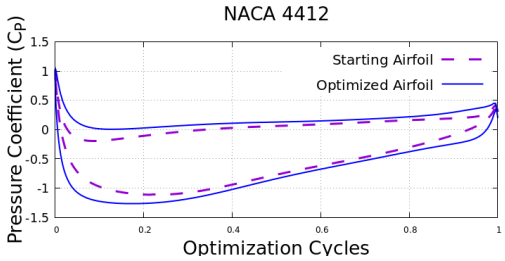
(c) *FAUVEL*

Figure 5.2: *Optimized airfoils (on scale). The target is to increase* C_L *by* 20%.

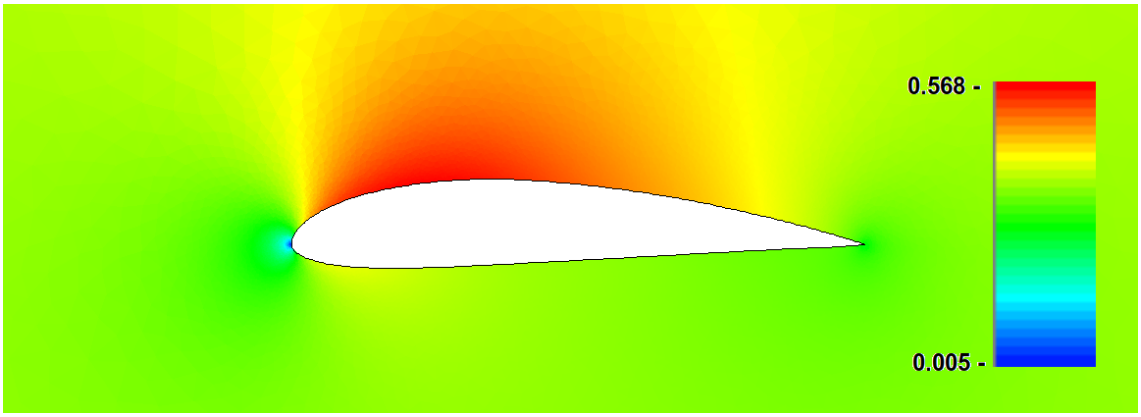
However, because the shape changes are not clear in-scale, in Figures 5.3 to 5.5 the airfoils are plotted out of scale. Moreover, the different Mach number fields and pressure distribution between the starting and optimized airfoils are presented.



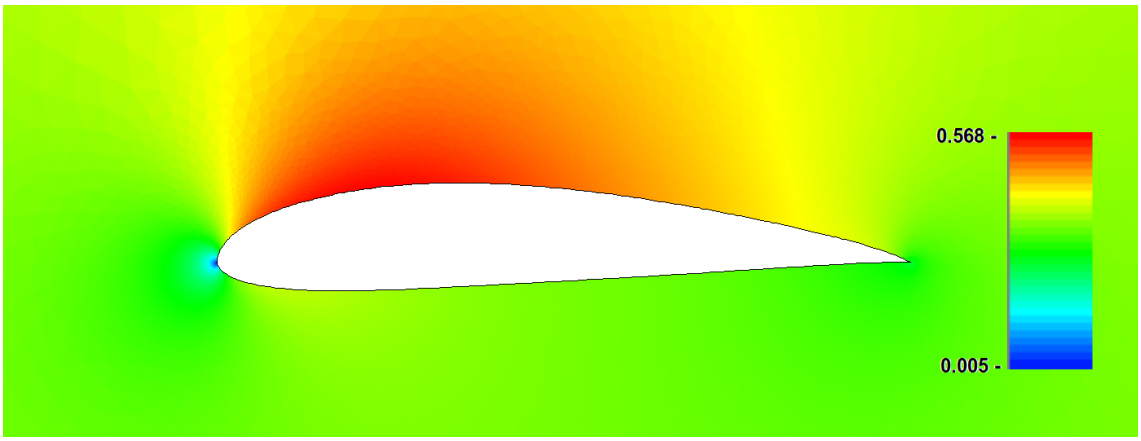
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*

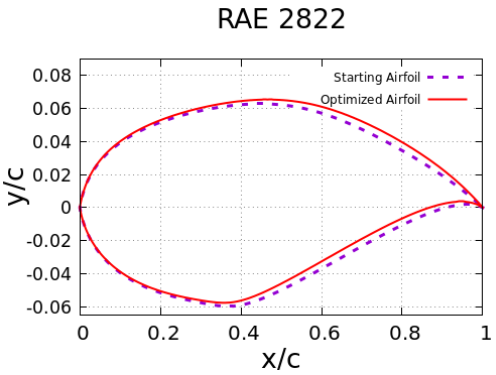


(c) *Starting Mach field*

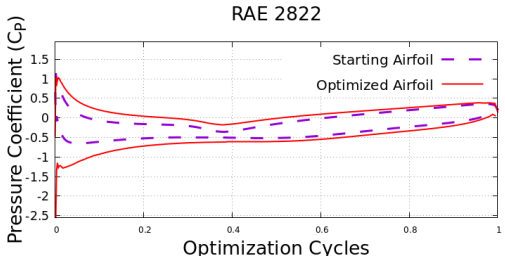


(d) *Optimized Mach field*

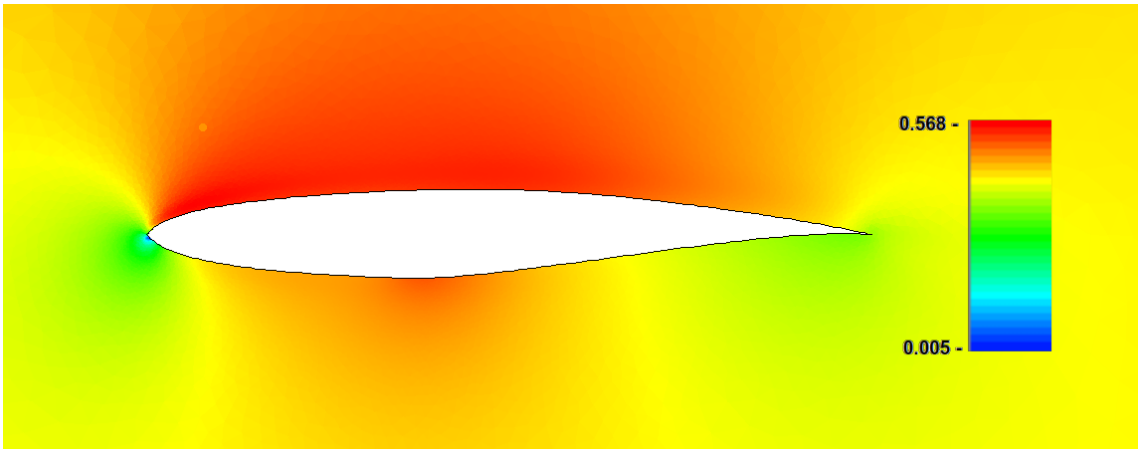
Figure 5.3: *NACA 4412 airfoil. Optimization results. The target is to increase C_L by* 20%.



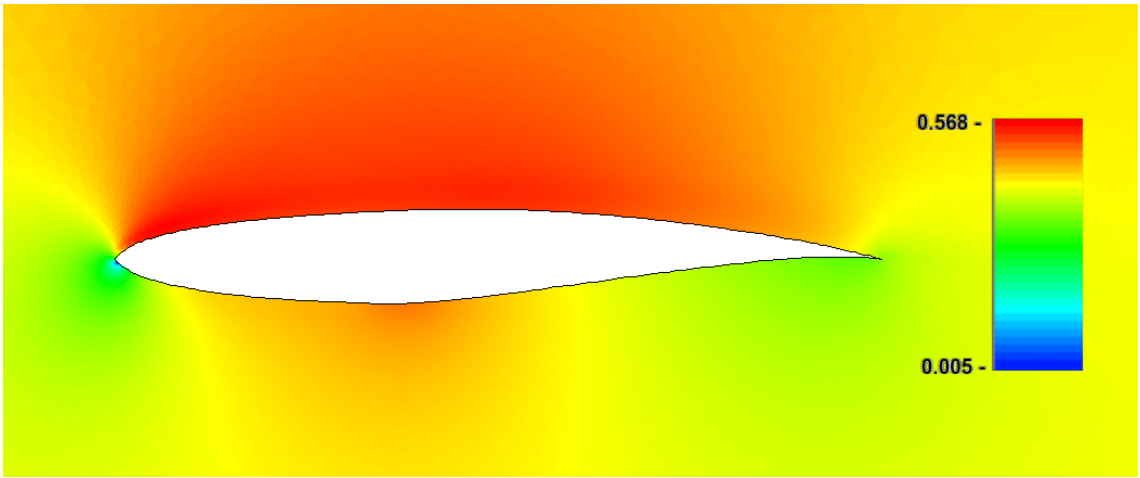
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*

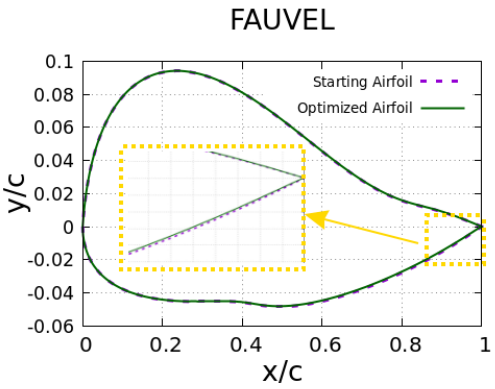


(c) *Starting Mach field*

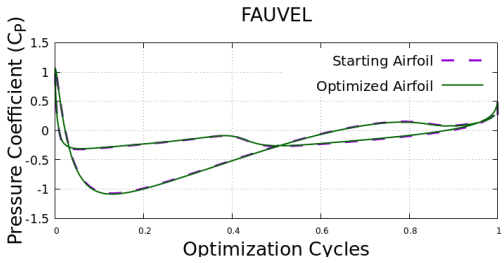


(d) *Optimized Mach field*

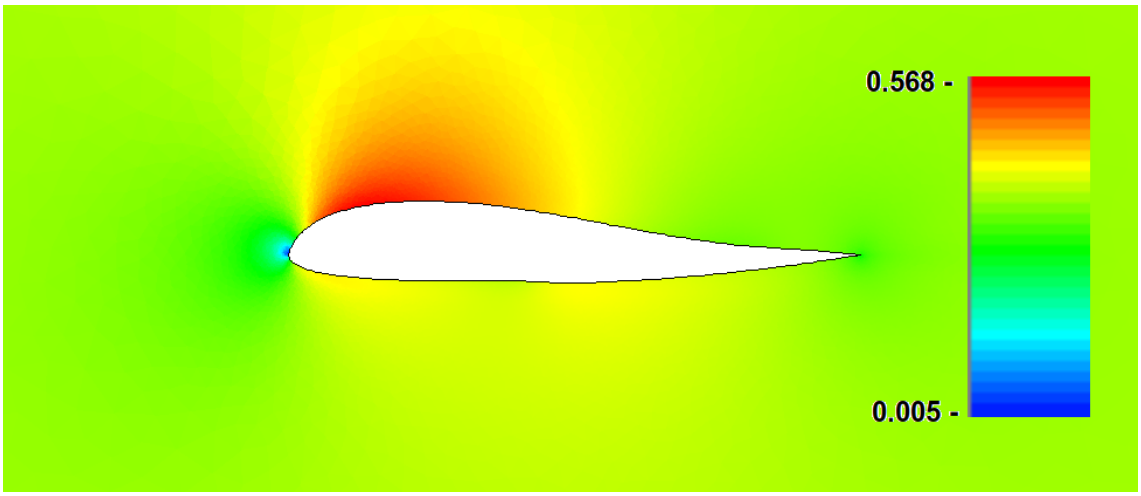
Figure 5.4: *RAE 2822 airfoil. Optimization results. The target is to increase C_L by* 20%.



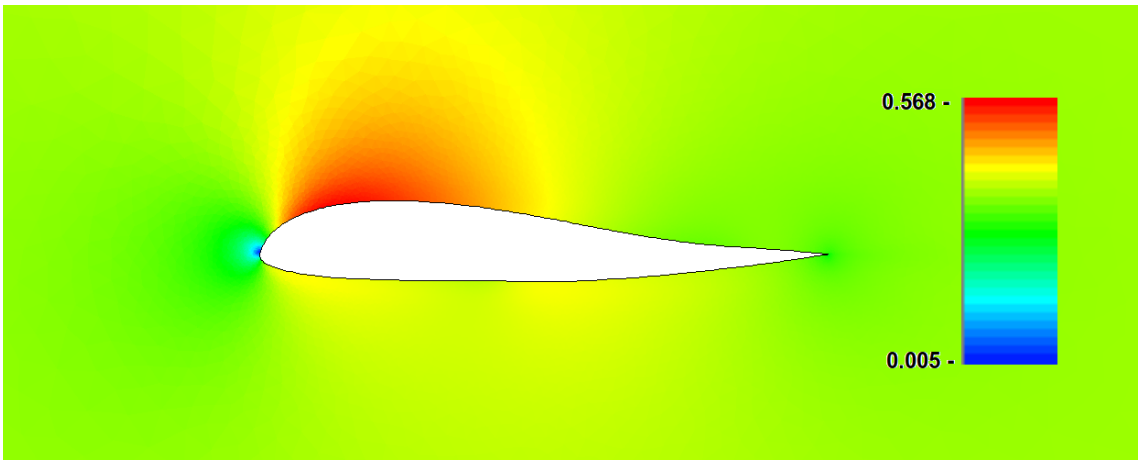
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*



(c) *Starting Mach field*



(d) *Optimized Mach field*

Figure 5.5: *FAUVEL airfoil. Optimization results. The target is to increase C_L by* 20%.

The main changes in order to increase the airfoil's lift happens near the trailing edge. The camber is increased. Concerning the FAUVEL airfoil, the target lift coefficient is not very different from the starting one. Thus, the optimization cycles are fewer than those of the other two cases and the airfoil shape doesn't change a lot. Furthermore, the Mach number field, although it alters it is not very clear in the Figures above. For that reason another optimization is performed for all three airfoils targeting to increase their lift by 40%.

5.2.2 40% Lift increase

In Table 5.2 the new objective functions are defined. Moreover, a new finish criteria is set

$$|F_i - F_{i-1}| < 10^{-6}$$

Reparameterized Airfoil	Starting C_L	Target C_L	Objective Function
NACA 4412	0.789	1.10	$F^1_{obj} = (C_L - 1.1)^2$
RAE 2822	0.625	0.87	$F^2_{obj} = (C_L - 0.87)^2$
FAUVEL	0.161	0.22	$F^3_{obj} = (C_L - 0.22)^2$

Table 5.2: *The objective function, based on eq. 3.6, used to optimize NACA 4412, RAE 2822 and FAUVEL airfoils. The target is to increase C_L by 40%.*

The three new optimization paths are presented in Figure 5.6 and the optimized geometries, in real life scale, in Figure 5.7

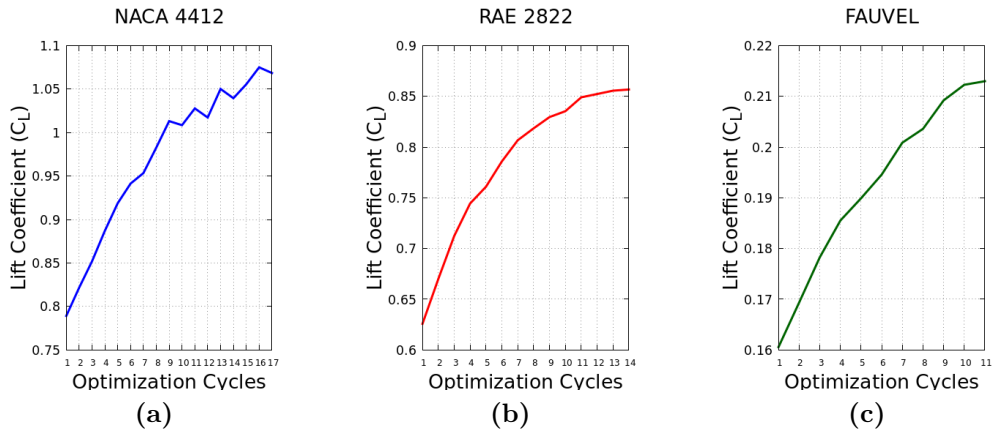
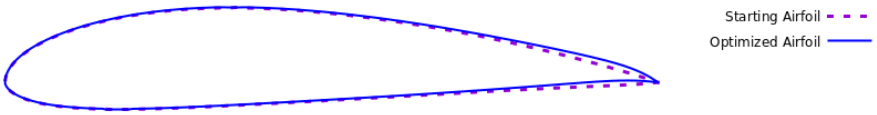
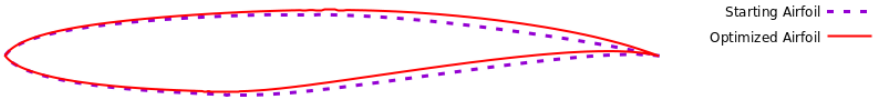


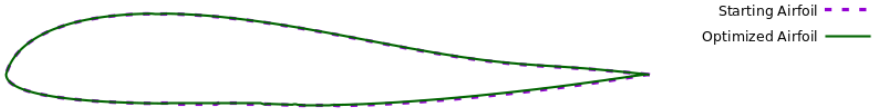
Figure 5.6: *The three optimization paths. The target is to increase C_L by 40%.*



(a) *NACA 4412*



(b) *RAE 2822*

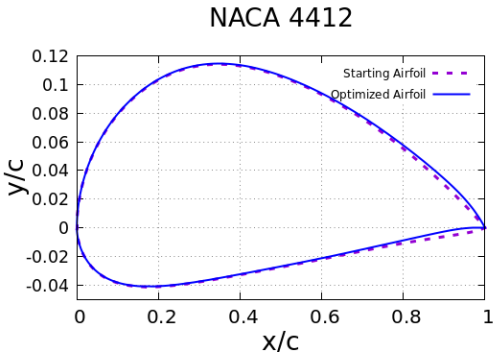


(c) *FAUVEL*

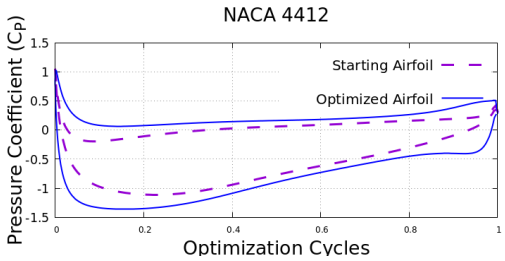
Figure 5.7: *Optimized airfoils (on scale). The target is to increase* C_L *by* 40%.

Similar to previous optimization process, the differences between the starting and the optimized geometry are better presented if the airfoils are not plotted in-scale. The new Mach number fields and pressure coefficient are also presented (Figures 5.8 to 5.10).

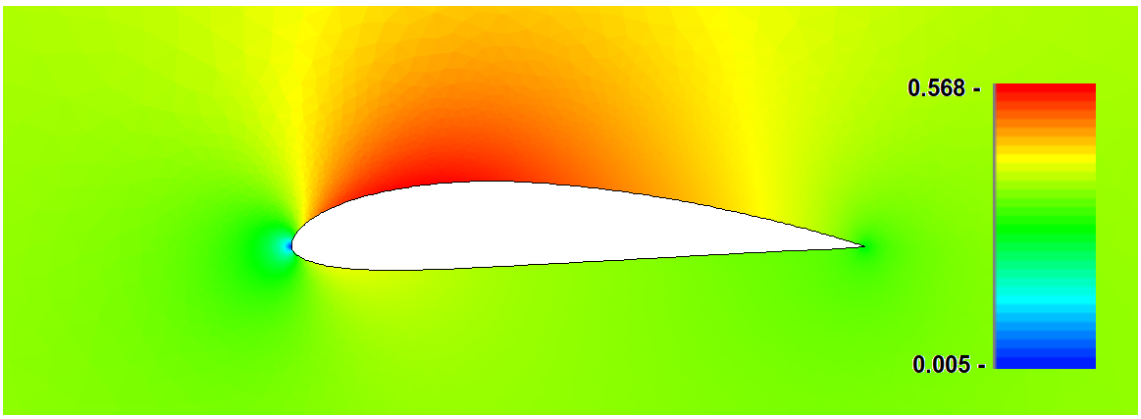
Concerning NACA 4412 and FAUVEL airfoils, the Mach number fields are slightly different between the starting and optimized geometry. On the other hand, the flow prediction around the optimized RAE 2822 results to a much different shape and Mach number field. A better representation of the increased lift is the pressure coefficient. As pressure gets bigger on the airfoil pressure side and lower on the suction side, the lift is increased. Moreover, one should also pay attention to the relation between Mach number and pressure. In places where the Mach number gets increased, the pressure drops and vice-versa.



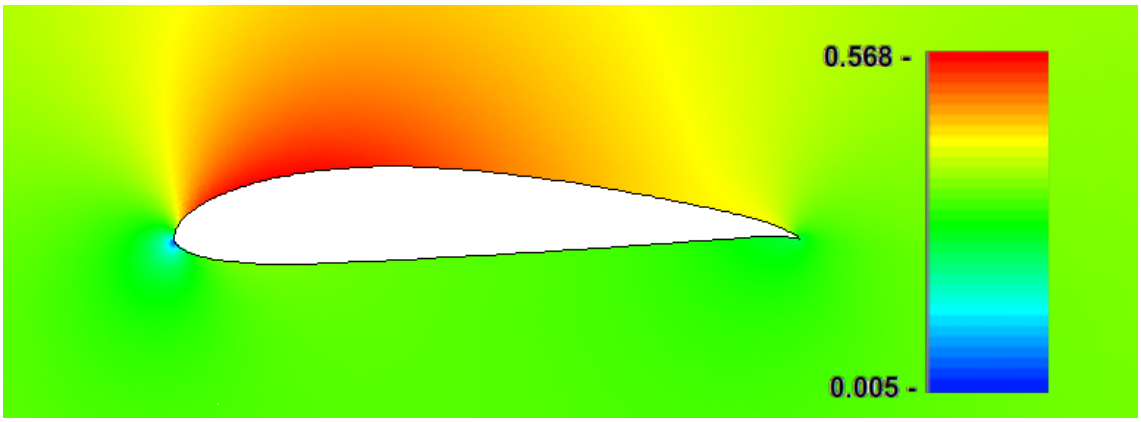
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*

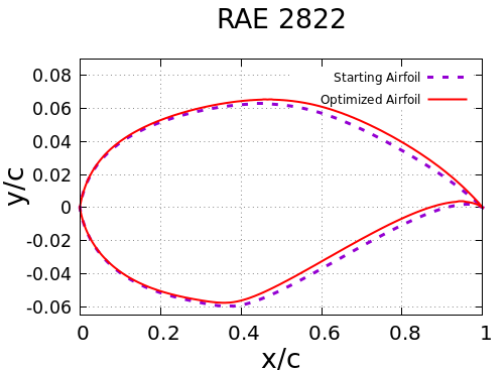


(c) *Starting Mach field*

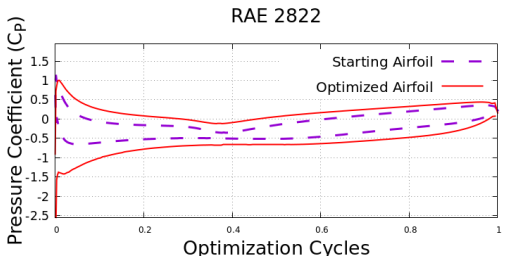


(d) *Optimized Mach field*

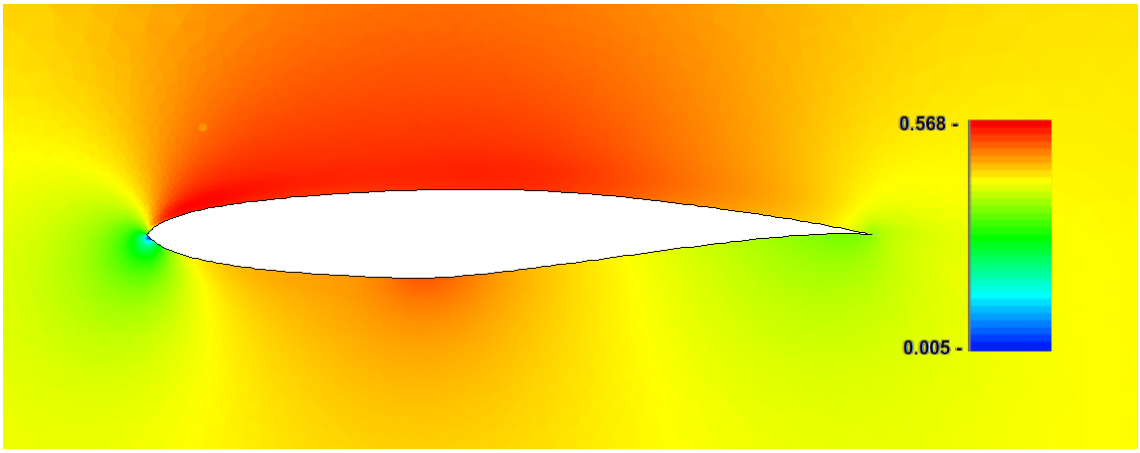
Figure 5.8: *NACA 4412 airfoil. Optimization results. The target is to increase C_L by* 40%.



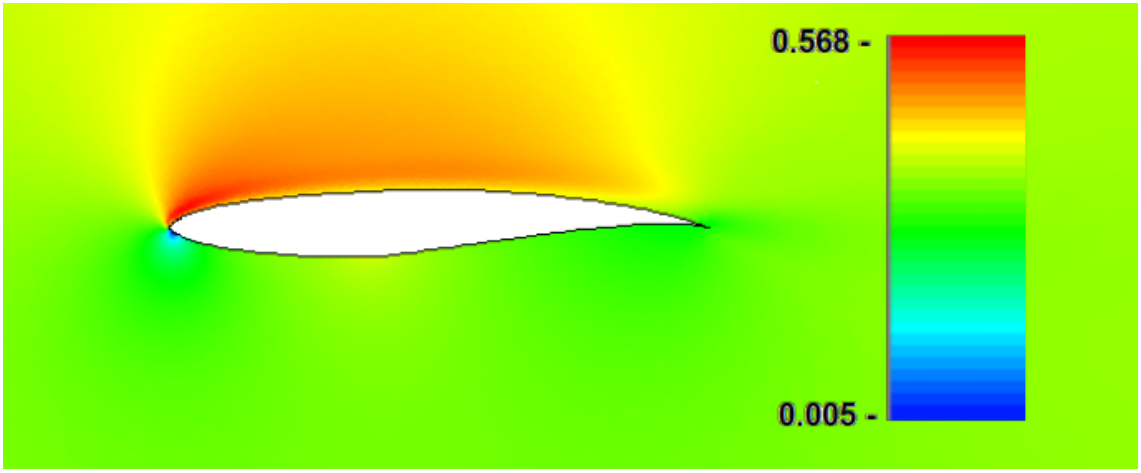
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*

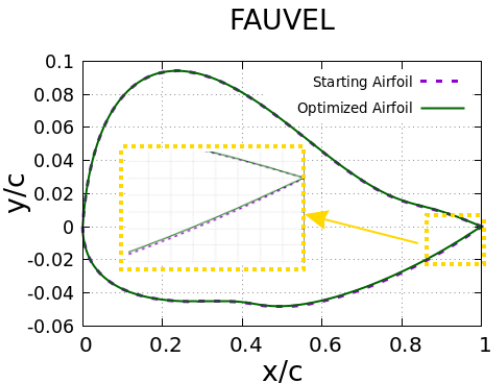


(c) *Starting Mach field*

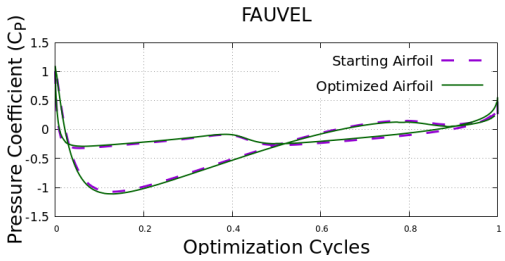


(d) *Optimized Mach field*

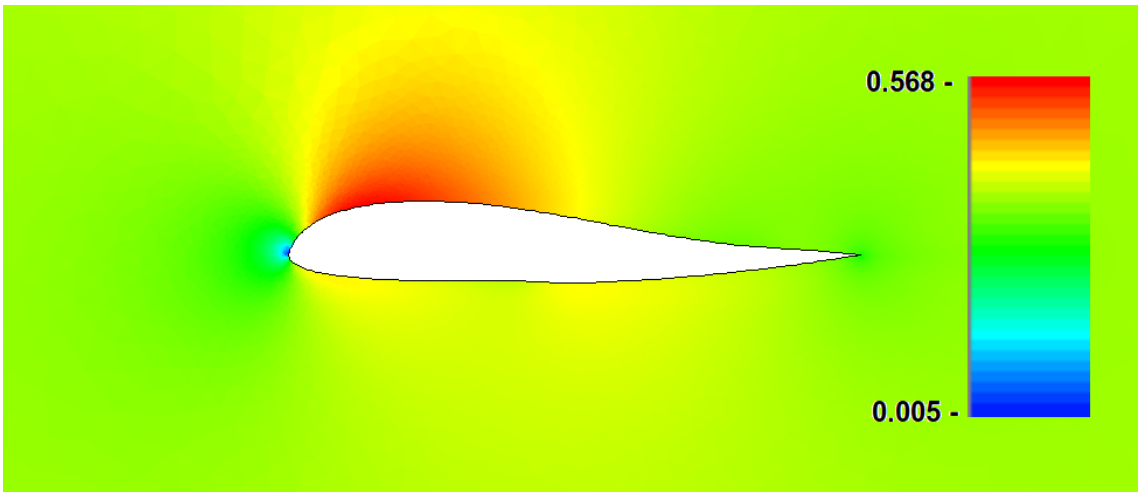
Figure 5.9: *RAE 2822 airfoil. Optimization results. The target is to increase C_L by* 40%.



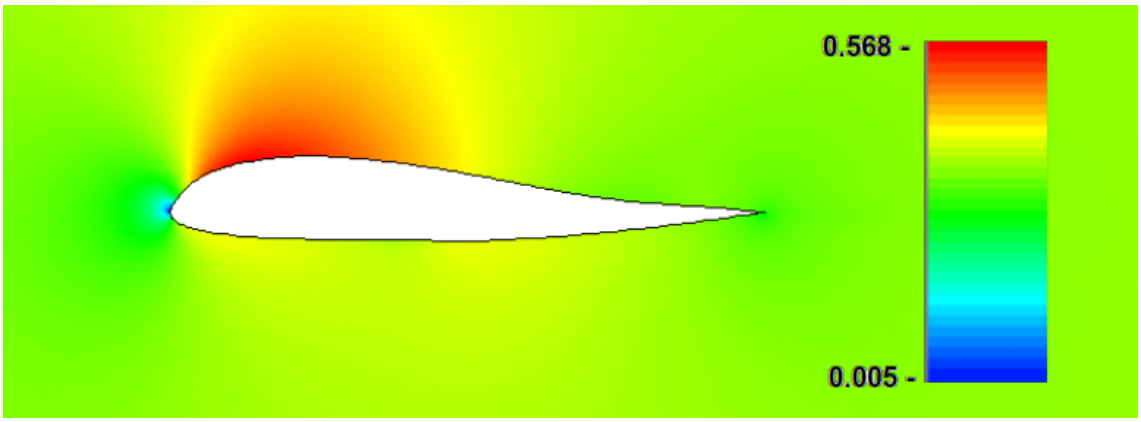
(a) *Shape changes (not in-scale)*



(b) *Pressure coefficient C_P*



(c) *Starting Mach field*



(d) *Optimized Mach field*

Figure 5.10: *FAUVEL airfoil. Optimization results. The target is to increase C_L by* 40%.

Chapter 6

Summary - Conclusion

In this diploma thesis, a practically abandoned FORTRAN code for almost a decade or so, developed by the PCOpt/NTUA, was updated in order to perform shape optimization using the E-SI adjoint method, instead of the Sev-SI adjoint, which was already programmed. E-SI adjoint has a low computational cost as Sev-SI adjoint, but without compromising the accuracy of the computed sensitivities. The main target is to make this code "equivalent" to the well-developed PUMA software. PUMA is programmed by the PCOpt/NTUA in C++ and CUDA to run on GPUs and is the main software that the PCOpt/NTUA uses for optimization on compressible flows (though PUMA also simulates incompressible flows).

Concerning the FORTRAN code, at the moment, it predicts the flow around an airfoil using the Euler equations in a 2D and unstructured FV grid, using the node-centered approach. For the computation of the SDs, the continuous adjoint method is used. The main optimization goal is the increase of lift and the minimization of drag coefficient. The latter is expected to be zero. However, it is programmed in order for this to be ready when its Navier-Stokes counterpart will be in place.

The flow of primal problem (flow prediction) can be solved with the use of either the FVS or the Roe scheme. Moreover, different orders of accuracy and a limiter can be implemented. Another important aspect is the density of the grid. Likewise, for the adjoint problem, similar numerical schemes and orders of accuracy are developed. The only difference is the use of another variant of Roe scheme, namely the non-conservative one.

The software's accuracy in computing each case's SDs is tested with a second-order Finite Differences scheme.

Regarding the primal equations' solver, the two numerical schemes are practically the same. No major differences are noticed. The order of accuracy highly affects the results. First-order leads to a fast convergence. Its results differ from those emerging from second-order schemes. However, as the density of the grid increases, first-order results tend to become similar to those of second-order. The use of the Van Leer Van Albada limiter doesn't have any worth mentioning differences.

The adjoint equations are non-conservative. Thus, the conservative schemes (FVS and Roe) are not mathematically correct and have low accuracy. That's why the Non-Conservative Roe is implemented. The orders of accuracy have the aforementioned influence on the adjoint field, too. In the conducted analysis, which involves the Euler equations and the increase of lift, E-SI doesn't differ from Sev-SI.

The adjoint method is a very economical one. With only a few equivalent flow solutions the three airfoils, presented in Chapter 5, are optimized and reached the target lift coefficient. They tend to increase their camber in order to increase their lift. The main changes happen near their trailing edge. Moreover, on each airfoil's pressure side, the Mach number decreases and pressure increases. On the suction side, the opposite phenomenon takes place.

Concluding this thesis, the developed software has achieved to predict 2D, compressible inviscid flows around an airfoil and increase their lift using the adjoint E-SI method.

Before closing, let us make clear that the purpose of this work was to make a first step in bringing the Fortran CFD code (primal and adjoint for shape optimization) closer to the well-developed GPU-enabled PUMA code of the same group. Since, during the last decade, development is exclusively based on the GPU-enabled PUMA code, there are good reasons to upgrade the Fortran code too, given the CPU cluster of the PCOpt/NTUA. Upgrading should, sooner or later, pass from 2D to 3D, from inviscid to viscous/turbulent flows and should finally include parallelization using MPI.

Appendix A

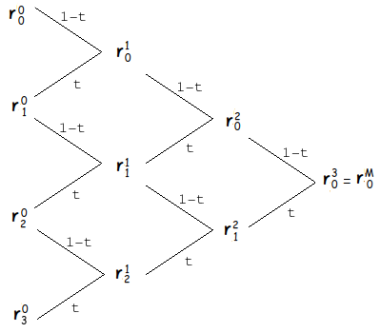
Bezier-Curve Shapes

On this dimploma thesis the airfoil's parameterization is based on the Bezier theory [8]. According to it, the geometry emerges from $M+1$ control points (CPs) P_j, with $\vec{r}_j{}^0, \quad j = 0, \cdots, M$ position vectors.

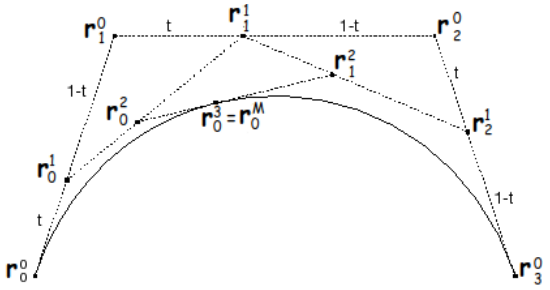
To begin with, a parameter $t \in [0, 1]$ is defined. Using the de Casteljau algorithm

$$\vec{r}_i{}^a(t) = (1-t)\vec{r}_i{}^{a-1}(t) + t\vec{r}_{i+1}{}^{a-1}(t) \tag{A.1}$$

where $a = 1, \cdots, M$ and $i = 0, \cdots, M-a$. This process eventually ends up in the $\vec{r}_0{}^M(t)$ quantity, which is the Bezier-Curve. A visualization of the algorithm and its physical meaning is presented in Figure A.1. Each column creates the next one, except from the first which is given.



(a) *Steps in diagramm*



(b) *Geometrical meaning of each term (The continuous line is the bezier curve)*

Figure A.1: *The de Casteljau algorithm*

Despite the method's simlicity a more direct and analytic expression of the curve is

needed. Thus, the Bernstein, $B_i^N(t)$, polyonyms are used

$$B_i^M(t) = \binom{M}{i} t^i (1-t)^{M-i} \tag{A.2}$$

where

$$\binom{M}{i} = \frac{M!}{i!(M-i)!}$$

The curve function is now stated as

$$\vec{r}_0{}^M(t) = \sum_{i=0}^{M} \vec{r}_i{}^0 B_i^M(t) \tag{A.3}$$

and called Bezier-Bernstein. Likewise $\vec{r}_0{}^M(t) = (x(t), y(t))$, the coordinate of each curve point can be expressed as

$$\begin{aligned} x(t) &= \sum_{i=0}^{M} x_i B_i^M(t) \\ y(t) &= \sum_{i=0}^{M} y_i B_i^M(t) \end{aligned} \tag{A.4}$$

where (x_i, y_i) are the cartesian coordinates of the P_i CP. Only under special condition can a $y = y(x)$ expression be extracted. In general, they are connected through the t parameter.

Differentiating the bezier curve w.r.t. the CPs' coordinates

$$\begin{aligned} \frac{\delta x(t)}{\delta x_i} &= \frac{\delta y(t)}{\delta y_i} = B_i^M(t) \\ \frac{\delta x(t)}{\delta y_i} &= \frac{\delta y(t)}{\delta x_i} = 0 \end{aligned} \tag{A.5}$$

However, (x_i, y_i) are the design variables for a shape optimization case. Thus, eq A.5 gradients can be also stated as $\delta x(t)/\delta b_n,\ \delta y(t)/\delta b_n$ (bezier curve's SDs).

Appendix B

Useful Mathematical Equations

Below are listed some equations on which the analysis of the adjoint method is based.[24]

1. Leibniz Integral Theorem

$$\frac{\delta}{\delta b_n}\int_{\Omega(b_n)} F(\vec{\mathbf{U}}, b_n)\, d\Omega = \int_{\Omega(b_n)} \frac{\partial}{\partial b_n} F(\vec{\mathbf{U}}, b_n)\, d\Omega + \int_{\partial\Omega(b_n)} F(\vec{\mathbf{U}}, b_n)\frac{\delta x_k}{\delta b_n}\vec{n_k}\partial\Omega \quad \text{(B.1)}$$

It is a formula for differentiation of a definite integral whose limits are functions of the differential variable.

2. Green-Gauss Theorem

$$\int_{\Omega} (\nabla F)\, d\Omega = \int_{S} (F\vec{n})\; dS \quad \text{(B.2)}$$

It relates the volume to surface integrals. More specificall the surface integral of a vector field over a closed surface (flux through the surface) is equal to the volume integral of the divergence over the region inside the surface. Intuitively, it states that the sum of all sources of the field in a region (with sinks regarded as negative sources) gives the net flux out of the region.

3. Material Derivative of Φ

$$\frac{\delta\Phi}{\delta b_n} = \frac{\partial\Phi}{\partial b_n} + \frac{\partial\Phi}{\partial x_k}\frac{\delta x_k}{\delta b_n} \quad \text{(B.3)}$$

The material derivative shows that the $\delta/\delta b_n$ and $\partial/\partial b_n$ derivation symbols refer to different physical meanings. Their distinction is of great importance. Φ is considered as a flow quantity calculated on node P of the grid. A variation of b_n, as a geometric variable, results in the change of the flow field. This change is described by the $\partial\Phi/\partial b_n$ term, computed on P. However, b_n also causes the deformation of grid.

This creates the second term on the RHS of the eq. B.3, which refers to the influence of the dissplacement of P in the unchanged flow field. Both these terms describe the total alteration of Φ, expressed as $\delta/\delta b_n$. The influence of b_n on Φ can be graphically presented in Figure B.1.

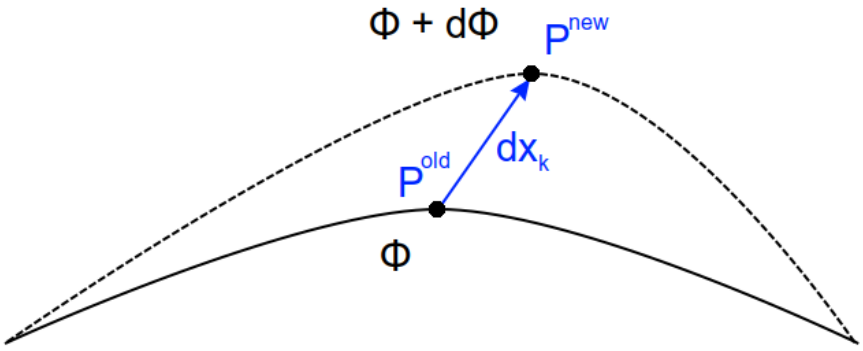


Figure B.1: *Physical representation of the total change of the quantity Φ. The term $\frac{\partial \Phi}{\partial b_n}$ refers to the flow alteration on the same location P^{old} and the $\frac{\partial \Phi}{\partial x_k}\frac{\delta x_k}{\delta b_n}$ to the change of the location where Φ is computed for constant flow field.*

4. Material Derivative of $\dfrac{\partial \Phi}{\partial x_i}$

$$\frac{\delta}{\delta b_n}\left[\frac{\partial \Phi}{\partial x_i}\right] = \frac{\partial}{\partial x_i}\left[\frac{\delta \Phi}{\delta b_n}\right] - \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_i}\left[\frac{\delta x_k}{\delta b_n}\right] \tag{B.4}$$

The proof of the equations is briefly presented below. Replacing Φ with $\partial\Phi/\partial x_i$ in eq. B.3

$$\frac{\delta}{\delta b_n}\left(\frac{\partial \Phi}{\partial x_i}\right) = \frac{\partial}{\partial b_n}\left(\frac{\partial \Phi}{\partial x_i}\right) + \frac{\partial}{\partial x_k}\left(\frac{\partial \Phi}{\partial x_i}\right)\frac{\delta x_k}{\delta b_n} \tag{B.5}$$

Differantiating eq. B.3 with $\partial/\partial x_i$

$$\frac{\partial}{\partial x_i}\left(\frac{\delta \Phi}{\delta b_n}\right) = \frac{\partial}{\partial x_i}\left(\frac{\partial \Phi}{\partial b_n}\right) + \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\frac{\delta x_k}{\delta b_n} + \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_i}\left(\frac{\delta x_k}{\delta b_n}\right) \tag{B.6}$$

Substracting eq. B.5 from eq. B.6, the eq. B.4 arrises.

5. Material Derivative of $\dfrac{\partial^2 \Phi}{\partial x_i \partial x_j}$

$$\frac{\delta}{\delta b_n}\left[\frac{\partial^2 \Phi}{\partial x_i \partial x_j}\right] = \frac{\partial^2}{\partial x_i \partial x_j}\left[\frac{\delta \Phi}{\delta b_n}\right] - \frac{\partial^2}{\partial x_i \partial x_k}\frac{\partial}{\partial x_j}\left[\frac{\delta x_k}{\delta b_n}\right] - \frac{\partial^2}{\partial x_j \partial x_k}\frac{\partial}{\partial x_i}\left[\frac{\delta x_k}{\delta b_n}\right] - \frac{\partial \Phi}{\partial x_k}\frac{\partial}{\partial x_i \partial x_j}\frac{\delta x_k}{\delta b_n} \tag{B.7}$$

Εκτενής Περίληψη Διπλωματικής Εργασίας

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Ανάπτυξη λογισμικού για Βελτιστοποίηση Γεωμετρίας Αεροτομών με τη Συνεχή Συζυγή Μέθοδο

Δημήτριος Δήμος

Επιβλέπων:
Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2021

Η Πρόλεξη της Ροής

Η ροή θεωρείται μόνιμη, συμπιεστή και μη-συνεκτική. Έτσι, χρησιμοποιούνται οι εξισώσεις Euler

$$\frac{\partial \vec{U}}{\partial t} + \frac{\partial \vec{f_i}}{\partial x_i} = 0 \tag{1}$$

με
$$\vec{U} = \begin{bmatrix} \rho \\ \rho\vec{u} \\ E_t \end{bmatrix}, \qquad \vec{f_i} = \begin{bmatrix} \rho u_i \\ \rho u_i \vec{u} + \vec{\delta_i} P \\ (E_t + P)u_i \end{bmatrix} \tag{2}$$

Οι εξισώσεις της χρονικά μόνιμης ροής λύνονται με τη χρήση του ψευδοχρόνου t. Οι μεταβλητές που εμφανίζονται είναι ρ η πυκνότητα, $\vec{u}$ το διάνυσμα της ταχύτητα, $E_t = \rho E = \dfrac{P}{\gamma - 1} + \dfrac{1}{2}\rho \left|\vec{u}\right|^2$ η συνολική ενέργεια ανά μονάδα όγκου και P η στατική πίεση. Για $A_i = \dfrac{\partial \vec{f_i}}{\partial \vec{U}}$ προκύπτει η συντηρητική μορφή των εξισώσεων της ροής

$$\frac{\partial \vec{U}}{\partial t} + A_i \frac{\partial \vec{U}}{\partial x_i} = 0 \tag{3}$$

Ως οριακές συνθήκες ορίζονται αυτές της αδιατάρακτης ροής στο επ΄ άπειρον πεδίο και η συνθήκη μη-εισχώρησης στα στερεά τοιχώματα. Η επίλυση υλοποιείται σε μη δομημένο πλέγμα πεπερασμένων όγκων με τη χρήση της κεντρο-κομβικής προσέγγισης. Για τη διακριτοποίηση των εξισώσεων χρησιμοποιούνται τα σχήματα FVS (εξ. 2.38) και Roe (εξ. 2.39). Η τάξη ακρίβειας ορίζεται από τις εξισώσεις 2.16 - 2.18. Αναλυτικά η διαδικασία εύρεσης του ροϊκού πεδίου περιγράφεται στο κεφάλαιο 2 του αγγλικού κειμένου.

Η Συνεχής Συζυγής Μέθοδος

Η συνεχής συζυγής μέθοδος μπορεί να αναπτυχθεί ακολουθώντας τρεις μεθοδολογίες, οι οποίες καταλήγουν στο ίδιο σύστημα συζυγών εξισώσεων και συζυγών οριακών συνθηκών, διαφορετικές όμως εκφράσεις παραγώγων ευαισθησίας, [14]. Η SI διατύπωση οδηγεί σε μια έκφραση για τις SDs η οποία περιλαμβάνει μόνο επιφανειακά ολοκληρώματα, έχει μικρό υπολογιστικό κόστος αλλά μπορεί κατά περίπτωση να υστερεί σε ακρίβεια. Η FI διατύπωση οδηγεί σε μια έκφραση για τις SDs η οποία περιλαμβάνει τόσο επιφανειακά όσο και χωρικά ολοκληρώματα, χαρακτηρίζεται από υψηλή ακρίβεια, αλλά και από υψηλό κόστος λόγω της ανάγκης υπολογισμού των παραγώγων ευαισθησίας του πλέγματος $\delta x_k / \delta b_n$ στον όγκο Ω του υπολογιστικού χωρίου. Η προσέγγιση Enhanced SI (E-SI), που αναπτύχθηκε από τη ΜΠΥΡ&Β/ΕΜΠ, εξαλείφει την ανάγκη

υπολογισμού του $\delta x_k/\delta b_n$ στο Ω, οδηγώντας σε κόστος αντίστοιχο της SI και ακρίβεια αντίστοιχης της FI προσέγγισης.

Η συνάρτηση-στόχος είναι η

$$F = w_l \left(C_L - C_{L_{tar}}\right)^2 + w_d C_D^2 \tag{4}$$

όπου w_l και w_d συναρτήσεις βάρους και $C_{L_{tar}}$ η επιθυμητή τιμής του συντελεστή άνωσης C_L. Τέλος, C_D είναι ο συντελεστής αντίστασης. Η συνάρτηση αυτή αποτελεί επιφανειακό ολοκλήρωμα της μορφής

$$F = \int_S F_{S_i} n_i dS \tag{5}$$

Εισάγοντας τη συζυγή συναρτήση της ροής Ψ_i και του πλέγματος m_i^a, θεωρώντας ότι το τελευταίο διέπεται από τις εξισώσεις Laplace, και παραγωγίζοντας την παραπάνω σχέση

$$\frac{\delta F_{AUG}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \frac{\delta}{\delta b_n}\int_\Omega \Psi_i R_i d\Omega + \frac{\delta}{\delta b_n}\int_\Omega m_i^a R_i^m d\Omega \tag{6}$$

όπου Ω το υπολογιστικό χωρίο, R_i τα υπόλοιπα των ροϊκών εξισώσεων και R_i^m των αντίστοιχων Laplace. Συγκεντρώνοντας όλους του όρους που αποτελούνται από παραγώγους ως προς τις μεταβλητές σχεδιασμού των πλεγματικών εξισώσεων στον όγκο και των ροϊκών μεγεθών στον όγκο και τις επιφάνειες του υπολογιστικού χωρίου και θέτοντας μηδέν τους συντελεστές τους, προκύπτουν οι συζυγείς εξισώσεις, οριακές συνθήκες και οι παράγωγοι ευαισθησίας (SD), εξ. 3.18 - 3.23. Για λόγους συντομίας παρουσιάζονται μόνο οι SD

$$\begin{aligned}
\frac{\delta F_{AUG}}{\delta b_n} &= \underbrace{2w_l \frac{C_L - C_{L_{tar}}}{q}\int_{S_W} p\left[\frac{\delta \overline{n_y}}{\delta b_n}cos(a) - \frac{\delta \overline{n_x}}{\delta b_n}sin(a)\right]}_{SD^1} \\
&\underbrace{+2w_d \frac{C_D}{q}\int_{S_W} p\left[\frac{\delta \overline{n_x}}{\delta b_n}cos(a) + \frac{\delta \overline{n_y}}{\delta b_n}sin(a)\right]}_{SD^2} \underbrace{+\int_{S_W}\left[\Psi_{k+1}p - \Psi_i f_k^i\right]\frac{\delta(\overline{n_k})}{\delta b_n}}_{SD^3} \\
&\underbrace{-\int_{S_W}\Psi_i \frac{\partial U_m}{\partial x_j}A_k^{im}\frac{\delta x_j}{\delta b_n}n_k dS_W}_{SD^4} \underbrace{-\int_S \frac{\partial m_i^a}{\partial x_j}n_j\frac{\delta x_i}{\delta b_n}dS}_{SD^5} \underbrace{+\int_S m_i^a R_i^m n_k \frac{\delta x_k}{\delta b_n}dS}_{SD^6} = 03.22
\end{aligned} \tag{7}$$

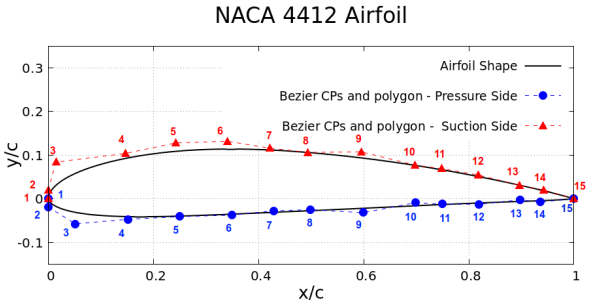
Ο όρος $SD5$ διαχωρίζει την SI προσέγγιση από την E-SI. Κατά τη πρώτη η συζυγής εξίσωση του πλέγματος δεν υφίσταται και ο όρος δεν δημιουργείται. Σημαντική παρατήρηση αποτελεί ακόμα ότι σε αντίθεση με τις εξισώσεις του πλέγματος, οι α-

ντίστοιχες συζυγείς έχουν μη-συντηρητική μορφή. Έτσι, κρίνεται επιτακτική η ανάγκη εισαγωγής νέου μη-συντηρητικού σχήματος διακριτοποίησης. Στην παρούσα εργασία χρησιμοποιείται το μη-συντηρητικό Roe.

Παραμετρική Ανάλυση Αριθμητικής Επίλυσης

Για τις ροϊκές εξισώσεις

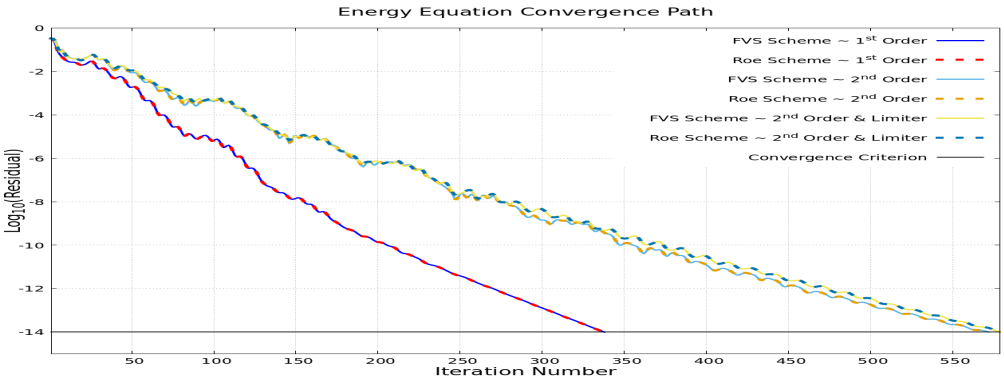
Καταρχάς για την επίλυση εξισώσεων της ροής το λογισμικό που δημιουργήθηκε διαθέτει δυο αριθμητικά σχήματα, δυο διαφορετικές τάξεις ακρίβειας, πρώτη και δεύτερη, και την εφαρμογή περιοριστή. Η επιρροή καθενός στοιχείου εξετάζεται σε τρεις αεροτομές. Στη συγκεκριμένη περίληψη αναφέρεται μόνο η αναπαραμετροποιημένη NACA 4412, η γεωμετρία και οι συνθήκες αδιατάρακτης ροής της οποίας παρουσιάζονται στο παρακάτω σχήμα.



Επ΄ Άπειρον πεδίο:

- Γωνία πρόσπτωσης $= 2^o$
- Αριθμός Mach $= 0.3$

Σχήμα 1: *Η Μορφή και τα σημεία - πολύγωνα Bezier της αεροτομής NACA 4412*



Σχήμα 2: *Αεροτομή NACA 4412. Σύγκριση αριθμητικών σχημάτων και τάξεων ακρίβειας αναφορικά με την πορεία σύγκλισης.*

Παρατηρώντας την πορεία σύγκλισης των εξισώσεων ενέργειας στο Σχήμα 2 παρατηρείται πως τα δυο σχήματα είναι όμοια μεταξύ τους. Οι διαφορές τους δεν είναι παρατηρήσιμες στο διαγράμματα. Από την άλλη, οι τάξεις ακρίβειας διαδραματίζουν

σημαντικό ρόλο στη διαδικασία επίλυσης των εξισώσεων. Η χρήση πρώτης τάξης ακρίβειας κάνει τον αλγόριθμο πολύ πιο γρήγορο και οικονομικό. Ο περιοριστής έχει μικρή επίπτωση στην πορεία σύγκλισης. Για τη σύγκριση της ποιότητας των αποτελεσμάτων χρησιμοποιούνται τέσσερα πλέγματα διαφορετικής πυκνότητας (Σχήμα 4.5 και Πίνακας 4.1). Τα αποτελέσματα παρουσιάζονται στον παρακάτω πίνακα.

Τάκη Ακρίβειας	Πλέγμα	C_L	C_D	Επαναλήψεις
1	Α	$4.948\ 10^{-1}$	$123.5\ 10^{-3}$	101
	Β	$6.188\ 10^{-1}$	$53.44\ 10^{-3}$	338
	Γ	$6.587\ 10^{-1}$	$34.02\ 10^{-3}$	573
	Δ	$6.849\ 10^{-1}$	$27.59\ 10^{-3}$	830
2	Α	$7.231\ 10^{-1}$	$14.54\ 10^{-3}$	172
	Β	$7.891\ 10^{-1}$	$3.788\ 10^{-3}$	572
	Γ	$7.803\ 10^{-1}$	$3.078\ 10^{-3}$	1103
	Δ	$7.851\ 10^{-1}$	$3.055\ 10^{-3}$	1515
2 & Περιοριστής	Α	$6.463\ 10^{-1}$	$31.74\ 10^{-3}$	180
	Β	$7.871\ 10^{-1}$	$5.416\ 10^{-3}$	579
	Γ	$7.779\ 10^{-1}$	$3.433\ 10^{-3}$	1115
	Δ	$7.840\ 10^{-1}$	$3.283\ 10^{-3}$	1530

Πίνακας 1: *Αεροτομή NACA 4412. Συντελεστές άνωσης και αντίστασης για διαφορετικές πυκνότητες πλέγματος και τάξεις ακρίβειας και περιοριστή.*

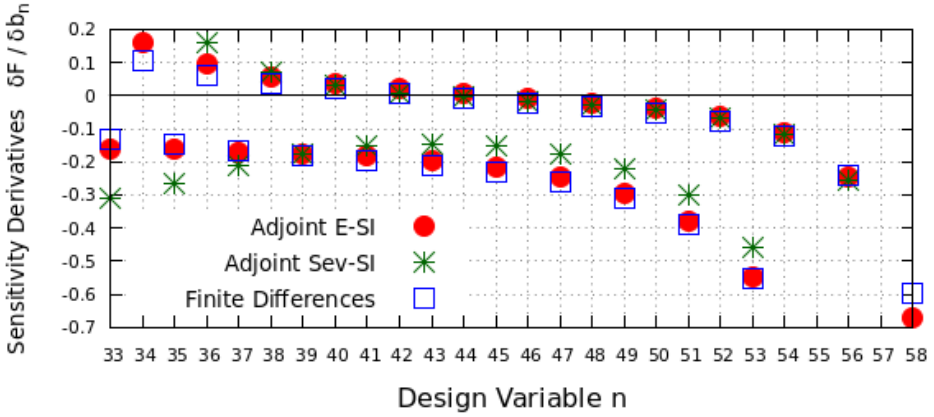
Όπως φαίνεται τα αποτελέσματα της δεύτερης τάξης ακρίβειας είναι με καλή προσέγγιση ανεξάρτητα από την πυκνότητα του πλέγματος. Σε αντίθεση αυτά της πρώτης τάξης μεταβάλλονται καθώς το πλέγμα γίνεται πιο πυκνό (Σχήμα 4.6 και 4.7). Θεωρητικά σε πλέγμα άπειρης πυκνότητας, η πρώτη τάξη ακρίβειας έχει ίδια αποτελέσματα με τη δεύτερη. Αυξάνοντας όμως τους κόμβους του υπολογιστικού χωρίου, αυξάνεται και το κόστος ανά επανάληψη. Παρόμοια συμπεράσματα εξάγονται και από τη σύγκριση του συντελεστή πίεσης πάνω στην επιφάνεια της αεροτομής (Σχήμα 4.8). Καθόλη τη μετέπειτα ανάλυση χρησιμοποιείται το πλέγμα Β.

Για τον υπολογισμό των SDs

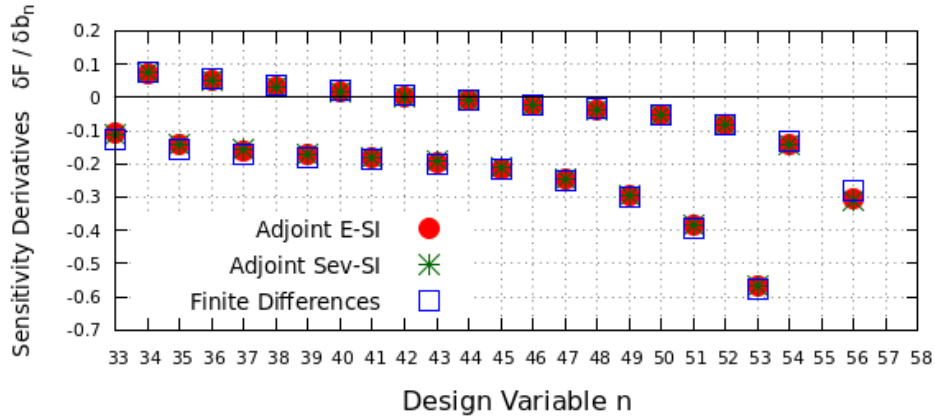
Όπως αναφέρθηκε νωρίτερα τα δύο συντηρητικά σχήματα δεν παρουσιάζουν διαφορές μεταξύ τους. Επίσης μικρή επίπτωση έχει και ο περιοριστής. Έτσι, για λόγους συντομίας παρουσιάζονται μόνο τα αποτελέσματα που εξάγονται από τις δύο παραλλαγές του σχήματος Roe για πρώτης και δεύτερης τάξης ακρίβεια στην πλευρά υποπίεσης της αεροτομής (Σχήμα 3). Τα υπόλοιπα αποτελέσματα μπορούν να βρεθούν στο αγγλικό κείμενο.

Οι SDs υπολογίζονται με τη συζυγή μέθοδο και συγκρίνονται με τις αντίστοιχες που προκύπτουν από πεπερασμένες διαφορές δεύτερης τάξης, χρησιμοποιώντας κάθε φορά το ίδιο σχήμα και τάξη ακρίβειας κατά τη διακριτοποίηση των εξισώσεων. Οι συζυγείς

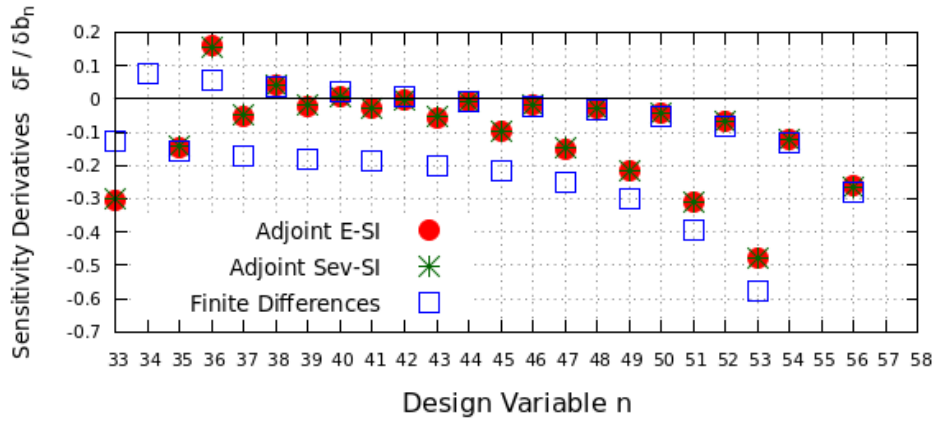
εξισώσεις επηρεάζονται από αυτά με τον ίδιο τρόπο που επηρεάζονται και οι ροϊκές εξισώσεις, όπως φαίνεται στο Σχήμα 2.



$1^{ης}$ Τάξης Ακρίβειας - Μη-Συντηριτικό σχήμα Roe - Πλευρά υποπίεσης



$2^{ης}$ Τάξης Ακρίβειας - Μη-Συντηριτικό σχήμα Roe - Πλευρά υποπίεσης



$2^{ης}$ Τάξης Ακρίβειας -Συντηριτικό σχήμα Roe - Πλευρά υποπίεσης

Σχήμα 3: *Αεροτομή NACA 4412. Σύγκριση σχημάτων, τάξης ακρίβειας και προσέγγισης κατά των υπολογισμό των SDs της*

Όσον αφορά το συντηρητικό σχήμα Roe, αυτό αποκλίνει αρκετά από τις πραγματικές τιμές. Το στοιχείο αυτό είναι αναμενόμενο, καθώς δημιουργούνται σφάλματα κατά τη διακριτοποίηση μη-συντηρητικών εξισώσεων με συντηρητικά σχήματα. Από την άλλη, το μη-συντηρητικό σχήμα Roe έχει ικανοποιητική ακρίβεια. Η μέθοδοι Sev-SI και E-SI δεν έχουν μεγάλες διαφορές, με εξαίρεση ορισμένες μεταβλητές σχεδιασμού όταν χρησιμοποιείται πρώτης τάξης ακρίβειας.

Παρόμοια αποτελέσματα παραμετρικής ανάλυσης παρουσιάζουν και οι άλλες δυο αεροτομές που μελετώνται στο κεφάλαιο 4.

Επιλογή βέλτιστων παραμέτρων βελτιστοποίησης

Η συζυγής μέθοδος υπολογίζει τις παραγώγους ευαισθησίας κάθε γεωμετρίας απαιτώντας σταθερό κόστος. Δηλαδή δεν αυξάνεται συναρτήσει του πλήθους των μεταβλητών σχεδιασμού N. Καθίσταται λοιπόν κατάλληλη για πολυπαραμετρική βελτιστοποίηση.

Όσον αφορά τα αριθμητικά σχήματα, για τις συζυγείς εξισώσεις το πιο κατάλληλο, από αυτά που διαθέτει ο κώδικας που δημιουργήθηκε, είναι το Roe και συγκεκριμένα η μη-συντηρητική μορφή του. Οι παράγωγοι που υπολογίζονται με τη χρήση του είναι πιο ακριβής και απαιτεί λιγότερες υπολογιστικές επαναλήψεις για να συγκλίνει. Παρόλο που για τις εξισώσεις της φυσικής ροής δεν παρατηρείται διαφορά στα δύο υπάρχοντα σχήματα, επιλέγεται το Roe για λόγους συνάφειας με τις συζυγείς εξισώσεις. Χρησιμοποιείται δεύτερης τάξης ακρίβεια.

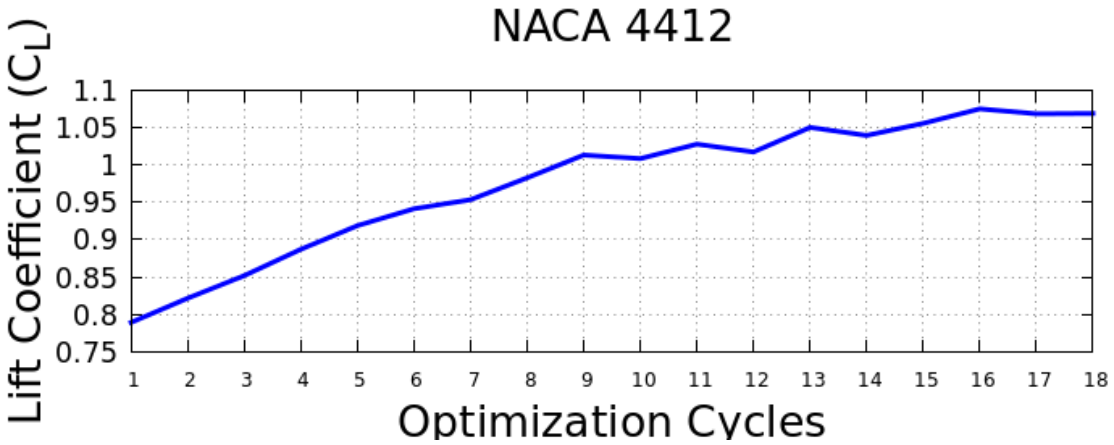
Τέλος, η συζυγής μέθοδος που επιλέγεται είναι η E-SI, παρόλο που οι διαφορές της με τη μέθοδο Sev-SI τυχαίνει να μην είναι μεγάλες στα προβλήματα που εξετάζονται. Το γεγονός αυτό οφείλεται στις εξισώσεις Euler και στο ότι η συνάρτηση-στόχος αποτελείται μόνο από το συντελεστή άνωσης.

Βελτιστοποίηση με τη Συζυγή Μέθοδο E-SI

Μετά την ανάπτυξη, και αξιολόγηση του λογισμικού, καθώς και την παραμετρική μελέτη των ρυθμίσεών του, πραγματοποιείται βελτιστοποίηση της NACA 4412 στις ίδιες συνθήκες επ΄ άπειρον ροής με σκοπό την αύξηση του συντελεστή άνωσης κατά 40%. Η συνάρτηση-στόχος που χρησιμοποιείται είναι η

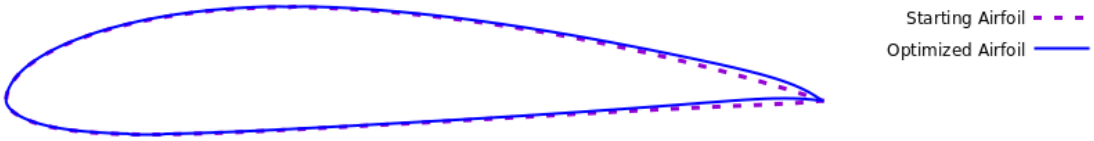
$$F_{obj}^{1} = (C_L - 1.1)^2$$

Η βελτιστοποίηση τερματίζεται εάν η διαφορά δύο διαδοχικών F_{obj} είναι μικρότερη από 10^{-6}. Το βήμα η που χρησιμοποιείται κατά την ανανέωση των μεταβλητών σχεδιασμού τίθεται ίσο με 10^{-3}. Η πορεία της βελτιστοποίησης φαίνεται στο Σχήμα 4.

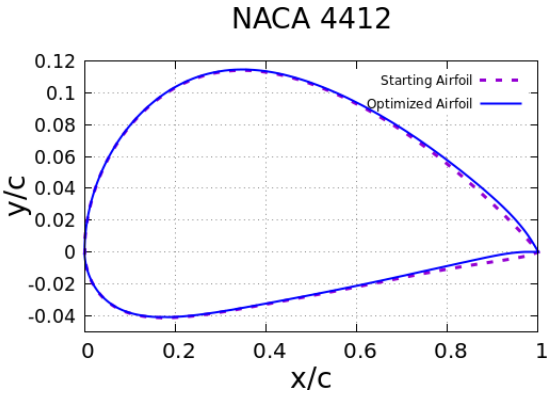


Σχήμα 4: *Η αετοτομή NACA 4412. Πορεία βελτιστοποίησης με στόχο* $C_L = 1.1$.

Στο Σχήμα 5 παρουσιάζονται οι αλλαγές στη γεωμετρία της αεροτομής σε φυσικές και μη-φυσικές διαστάσεις. Όπως γίνεται αντιληπτό οι κύριες διαφορές μεταξύ αρχικής και βελτιστοποιημένης γεωμετρίας βρίσκονται κοντά στην ακμή εκφυγής. Η αεροτομή τείνει να αυξήσει την καμπυλότητά της.



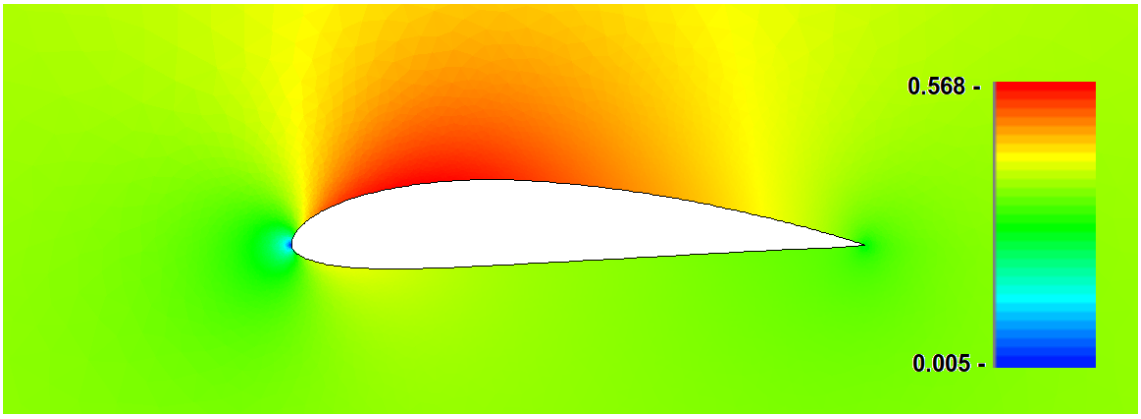
(α′) *Φυσικές διαστάσεις*



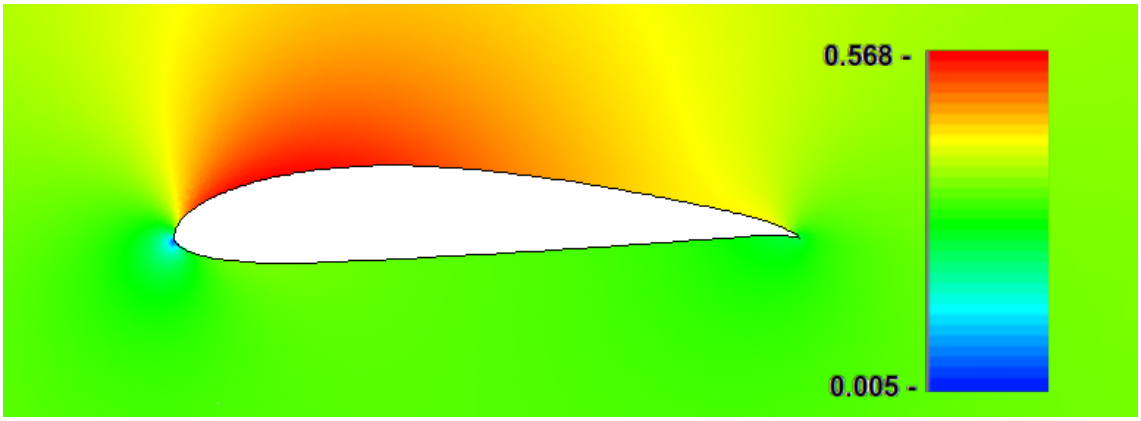
(β′) *Μη-φυσικές διαστάσεις*

Σχήμα 5: *Η αετοτομή NACA 4412. Αλλαγή γεωμετρίας κατά τη βελτιστοποίηση με στόχο* $C_L = 1.1$.

Τέλος, παρουσιάζεται και το πεδίο του αριθμού Mach της ροής για την αρχική και τελική γεωμετρία (Σχήμα 6).



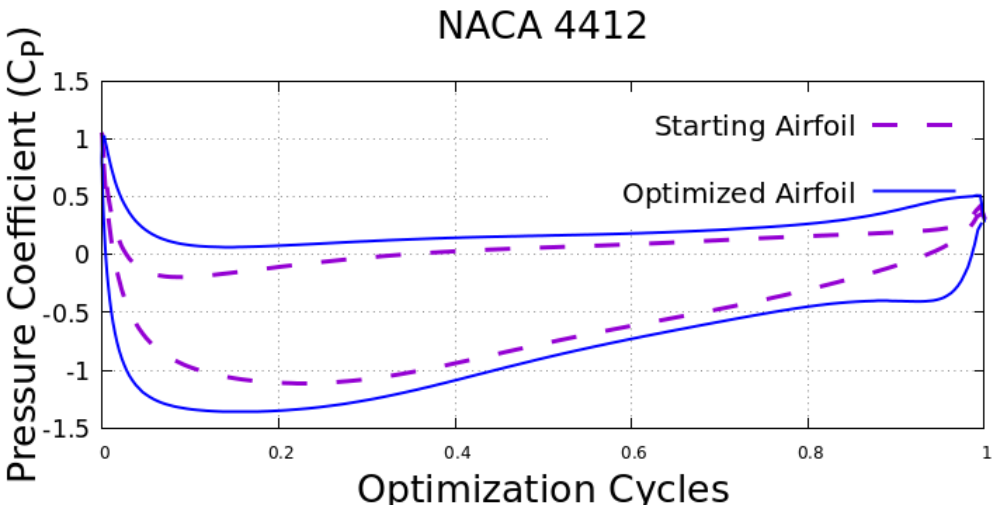
(α′) *Αρχικό πεδίο Mach*



(β′) *Τελικό πεδίο Mach*

Σχήμα 6: *Η αεροτομή NACA 4412. Μεταβολή του πεδίου του αριθμού Mach κατά τη βελτιστοποίηση με στόχο* $C_L = 1.1$.

Όπως στη γεωμετρία, έτσι και στο πεδίο του αριθμού Mach οι αλλαγές δεν είναι μεγάλες. Περιορίζονται μόνο στην περιοχή κάτω από την πλευρά πίεσης. Τα αποτελέσματα της βελτιστοποίησης γίνονται πιο αισθητά από την κατανομή του συντελεστή πίεσης. Όσο η πίεση αυξάνεται στην πλευρά πίεσης και μειώνεται στην πλευρά υποπίεσης, τόσο μεγαλύτερη γίνεται η άνωση (Σχήμα 7).



Σχήμα 7: *Η αετοτομή NACA 4412. Μεταβολή του συντελεστή πίεσης C_P κατά τη βελτιστοποίηση με στόχο $C_L = 1.1$.*

Bibliography

[1] Tchieu A. A. and L., Anthony: *A discrete-vortex model for the arbitrary motion of a thin airfoil with fluidic control*. Journal of Fluids and Structures, 27(5):680–693, 2011, ISSN 0889-9746. https://www.sciencedirect.com/science/article/pii/S0889974611000260, IUTAM Symposium on Bluff Body Wakes and Vortex-Induced Vibrations (BBVIV-6).

[2] Spall, J. C.: *Stochastic Optimization*, pages 173–201. Handbook of Computational Statistics. Springer Handbooks of Computational Statistics. Springer, Berlin, Heidelberg, 2012.

[3] B., Mohammadi and O., Pironneau: *Applied Shape Optimization for Fluids*, pages 81–115. Oxford University Press, 2010.

[4] B., Sengupta, K.J., Friston, and W.D., Penny: *Efficient gradient computation for dynamical models*. NeuroImage, 98:521–527, 2014, ISSN 1053-8119. https://www.sciencedirect.com/science/article/pii/S1053811914003097.

[5] Sorber L., van Barel M. and Lathauwer L. de: *Unconstrained optimization of real functions in complex variables*. SIAM Journal on Optimization, 22:879–898, 2012. https://epubs.siam.org/doi/10.1137/110832124.

[6] Giannakoglou, K.C.: *Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence*. Progress in Aerospace Sciences, 38(1):43–76, 2002, ISSN 0376-0421. https://www.sciencedirect.com/science/article/pii/S0376042101000197.

[7] Darwin, C.: *The origin of species by means of natural selection, or, the preservation of favoured races in the struggle for life*. John murray, 1876.

[8] Γιαννάκογλου Κ. Χ., Αναγνωστόπουλος Ι. και Μπεργελές Γ.: *Αριθμητική Ανάλυση για Μηχανικούς*. Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2003.

[9] A., Jameson: *Aerodynamic design via control theory*. Journal of Scientific Computing, 3, 1988. https://doi.org/10.1007/BF01061285.

[10] Giannakoglou, K. C. and Papadimitriou, D. I.: *Adjoint Methods for Shape Optimization*, pages 79–108. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, ISBN 978-3-540-72153-6. https://doi.org/10.1007/978-3-540-72153-6_4.

[11] Papoutsis-Kiachagias, E. M., Asouti, V. G., Giannakoglou, K. C., Gkagkas, K., Shimokawa, S., and Itakura, E.: *Multi-point aerodynamic shape optimization of cars based on continuous adjoint.* Structural and Multidisciplinary Optimization, 59:675–694, 2019, ISSN 1615-1488. https://doi.org/10.1007/s00158-018-2091-3.

[12] S., Nadarajah and A., Jameson: *A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization.* AIAA Paper 2000-0667, AIAA 38th. Aerospace Sciences Meeting and Exhibit, Reno, NV, 2000.

[13] Kavvadias, I.S., Papoutsis-Kiachagias, E.M., and Giannakoglou, K.C.: *On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization.* Journal of Computational Physics, 301:1–18, 2015, ISSN 0021-9991. https://www.sciencedirect.com/science/article/pii/S0021999115005318.

[14] Giannakoglou, KC, Papoutsis-Kiachagias, EM, Kavvadias, IS, and Gkaragkounis, K Th: *Continuous adjoint in shape & topology optimization-recent developments & applications.* In *Seminar on Adjoint CFD Methods in Industry and Research*, pages 43–52, 2016.

[15] Anderson, W.K. and Venkatakrishnan, V.: *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation.* Computers & Fluids, 28(4):443–480, 1999, ISSN 0045-7930. https://www.sciencedirect.com/science/article/pii/S0045793098000413.

[16] Gkaragkounis, K.T., Papoutsis-Kiachagias, E.M., and Giannakoglou, K.C.: *The continuous adjoint method for shape optimization in conjugate heat transfer problems with turbulent incompressible flows.* Applied Thermal Engineering, 140:351–362, 2018, ISSN 1359-4311. https://www.sciencedirect.com/science/article/pii/S1359431118311748.

[17] Kavvadias, I.S.: *Continuous Adjoint Methods for Steady and Unsteady Turbulent flows with Emphasis on the Accuracy of Sensitivity Derivatives.* PhD thesis, Lab. of Thermal Turbomachines, N.T.U.A., Athens, 2015.

[18] Ασούτη, Β.: *Μέθοδοι αεροδυναμικής ανάλυσης και σχεδιασμού για ροές υψηλών και χαμηλών ταχυτήτων, σε πολυεπεξεργαστικό περιβάλλον.* Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2009.

[19] Γιαννάκογλου, Κ. Χ.: *Μέθοδοι Βελτιστοποίησης στην Αεροδυναμική.* Πανεπιστημιακές Εκδόσεις Ε.Μ.Π., Αθήνα, 2006.

[20] Kazolea, M., Delis, A.I., Nikolos, I.K., and Synolakis, C.E.: *An unstructured finite volume numerical scheme for extended 2d boussinesq-type equations.* Coastal Engineering, 69:42–66, 2012, ISSN 0378-3839. https://www.sciencedirect.com/science/article/pii/S0378383912001032.

[21] Versteeg, H.K. and Malalasekera, W.: *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, chapter 5. Pearson Education Limited, 2007, ISBN 9780131274983. https://books.google.gr/books?id=RvBZ-UMpGzIC.

[22] B., van Leer: *Flux-Vector Splitting for the Euler Equation*, pages 80–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, ISBN 978-3-642-60543-7. https://doi.org/10.1007/978-3-642-60543-7_5.

[23] Roe, P.L: *Approximate riemann solvers, parameter vectors, and difference schemes*. Journal of Computational Physics, 43(2):357–372, 1981, ISSN 0021-9991. https://www.sciencedirect.com/science/article/pii/0021999181901285.

[24] Papoutsis-Kiachagias, E.M.: *Adjoint methods for turbulent flows, applied to shape and topology optimization and robust design*. PhD thesis, Lab. of Thermal Turbomachines, N.T.U.A., Athens, 2013.

[25] Trompoukis, X. S., Tsiakas, K. T., Ghavami N., M, Asouti, V. G, and Giannakoglou, K. C: *The continuous adjoint method on graphics processing units for compressible flows*. In *OPT-i, International Conference on Engineering and Applied Sciences Optimization*, pages 4–6, 2014.

[26] Weiss, J. M., Maruszewski, J. P., and Smith, W. A.: *Implicit solution of preconditioned navier-stokes equations using algebraic multigrid*. AIAA Journal, 37(1):29–36, 1999. https://doi.org/10.2514/2.689.

[27] G.S., Avinash and Lal, S Anil: *Inverse design of airfoil using vortex element method*. International Journal of Fluid Machinery and Systems, 11:163–170, June 2018.

[28] Τσιάκας, Κ.: *Ανάπτυξη μεθόδων βελτιστοποίησης με χρήση επεξεργαστών καρτών γραφικών και εφαρμογή στις στροβιλομηχανές*. Διδακτορική διατριβή, Εργαστήριο Θερμικών Στροβιλομηχανών, Ε.Μ.Π., Αθήνα, 2019.