



**Technical University of Crete**  
School of Production Engineering and Management

---

Συμβολή στο Βέλτιστο Σχεδιασμό Αεροτομών  
On the Optimal Design of Airfoils

---

by

**Stavros N. Leloudas**

**Diploma Thesis**

Supervisor: Dr. Ioannis K. Nikolos, Associate Professor

**Chania, June 2015**

**“Intentionally Blank”**



**Technical University of Crete**  
School of Production Engineering and Management

---

Συμβολή στο Βέλτιστο Σχεδιασμό Αεροτομών  
On the Optimal Design of Airfoils

---

by

**Stavros N. Leloudas**

Approved by:

Dr. Ioannis K. Nikolos  
Associate Professor  
Technical University of Crete  
School of Production Engineering  
& Management

---

Dr. Anargyros I. Delis  
Associate Professor  
Technical University of Crete  
School of Production Engineering  
& Management

---

Dr. Michalis I. Konsolakis  
Assistant Professor  
Technical University of Crete  
School of Production Engineering  
& Management

---

**Chania, June 2015**

Copyright © Stavros N. Leloudas, 2015  
All rights reserved.

*“It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong.”*

Richard P. Feynman (1918-1988)

**“Intentionally Blank”**

## Abstract

During recent years, computer graphics techniques such as Free-Form Deformation (FFD), have become extremely useful and widely employed in the field of Aerodynamic Shape Optimization and particular throughout the design of airfoil sections. Although FFD is a powerful parameterization and deformation technique of any given arbitrary two- or three-dimensional shape, there is no guarantee that provides the preservation of the shape's enclosed area or volume respectively, after its application. Given the importance of the structural integrity required by aerodynamic shapes, such as aircraft wings and wind turbine blades, the necessity of including a cross-sectional area preservation constraint (among several other geometrical and aerodynamic ones) arises during the optimization process of the airfoil sections forming the aforementioned applications. Even though previous works exist, where a cross-sectional area constraint is utilized, the implementation is done by either non-linear time consuming expressions or by penalty function approaches, which are not always sufficient and do not guarantee the exact satisfaction of a strict equality constraint throughout the design process. In this work an airfoil optimization scheme is presented, based on Area-Preserving Free-Form Deformation technique, which serves as an alternative approach for the handling and satisfaction of a strict cross-sectional area equality constraint, while a parallel Differential Evolutionary (DE) algorithm is utilized for the optimization procedure. The DE algorithm is combined with two Artificial Neural Networks (ANNs), a multilayer perceptron (MLP) feed-forward ANN and a Radial Basis Functions (RBF) network, which serve as surrogate models, to decrease the computational cost of the optimization procedure. In each iteration of the DE algorithm, before the evaluation of the fitness function for each candidate solution, an area preservation step is applied to that solution in order to meet the cross-sectional area constraint. The area preservation step is achieved by solving an area correction sub problem, which consists of computing and applying the minimum possible offset to each free-to-move control point of the FFD lattice, subject to the area conservation. Due to the linearity of the area constraint in each axis, the extraction of an inexpensive closed-form solution to the sub problem is possible by using Lagrange Multipliers method. The proposed technique overcomes the disability of Evolutionary Algorithms (EAs) to effectively treat strict equality constraints such as exact area preservation one. Throughout the optimization process both structural and aerodynamic requirements can

be taken into account, as constraints while the objective function is focused on the improvement of aerodynamic efficiency. Additionally, the use of multiple surrogate models, in conjunction with the inexpensive solution to the area correction sub problem, render the optimization process time saving. This thesis demonstrates the applicability and effectiveness of the proposed methodology.

## Περίληψη

Κατά τη διάρκεια των τελευταίων ετών, τεχνικές προερχόμενες από τον τομέα των υπολογιστικών γραφικών, όπως η μέθοδος Ελεύθερης Παραμόρφωσης, έχουν γίνει εξαιρετικά χρήσιμες και ευρέως εφαρμόσιμες κατά τη διαδικασία βελτιστοποίησης εν γένει αεροδυναμικών σχημάτων και συγκεκριμένα αεροτομών. Αν και η Ελεύθερη Παραμόρφωση είναι μια πανίσχυρη τεχνική παραμετροποίησης και παραμόρφωσης οποιουδήποτε σχήματος αυθαίρετης γεωμετρίας, ύστερα από την παραμόρφωση του εκάστοτε αντικειμένου η ιδιότητα της διατήρησης του εμβαδού ή του όγκου, που περικλείεται από το σχήμα για διδιάστατα ή τριδιάστατα σχήματα αντίστοιχα, δεν παρέχεται. Δεδομένης της απαιτούμενης δομικής ακεραιότητας και αντοχής από αεροδυναμικές διατάξεις όπως πτερύγια, προκύπτει η αναγκαιότητα της συμπερίληψης ενός περιορισμού, με σκοπό την διάτρηση του εμβαδού, κατά τη διαδικασία βελτιστοποίησης π.χ. αεροτομών. Παρά το γεγονός της ύπαρξης εργασιών όπου ο περιορισμός της διατήρησης του εμβαδού χρησιμοποιείται κατά τη διαδικασία βελτιστοποίησης, η εφαρμογή του πραγματοποιείται είτε με μη-γραμμικές σχέσεις, οι οποίες αυξάνουν τον υπολογιστικό χρόνο, είτε με συναρτήσεις τιμωρίας, οι οποίες δεν είναι πάντα αποτελεσματικές και δεν εγγυόνται ότι ο αυστηρός περιορισμός ισότητας θα είναι ενεργός κατά τη διάρκεια της διαδικασίας. Στην παρούσα διπλωματική εργασία παρουσιάζεται μια νέα μεθοδολογία βελτιστοποίησης, βασισμένη στην τεχνική της Ελεύθερης Παραμόρφωσης με Διατήρηση του Εμβαδού Διατομής, η οποία λειτουργεί ως ένα εναλλακτικό μέσο χειρισμού και ικανοποίησης του παραπάνω περιορισμού ισότητας, ενώ ένας παράλληλος Διαφορικός Εξελικτικός (ΔΕ) αλγόριθμος χρησιμοποιείται για τη διαδικασία βελτιστοποίησης. Ο ΔΕ αλγόριθμος συνδυάζεται με δύο Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ), ένα Πολυεπίπεδο Perceptron (Multi-Layer perceptron - MLP) και ένα ΤΝΔ Ακτινικών Συναρτήσεων Βάσης (Radial Basis Function – RBF ANN), τα οποία λειτουργούν ως υποκατάστατα μοντέλα (surrogate models) με σκοπό τη μείωση του υπολογιστικού κόστους της διαδικασίας βελτιστοποίησης. Σε κάθε επανάληψη του ΔΕ αλγορίθμου, προτού τον υπολογισμό της αντικειμενικής συνάρτησης για κάθε υποψήφια λύση, ένα βήμα διόρθωσης του εμβαδού διατομής εφαρμόζεται στην υπό εξέταση γεωμετρία με σκοπό την ικανοποίηση του αντίστοιχου περιορισμού. Η υλοποίηση του παραπάνω βήματος επιτυγχάνεται μέσω της επίλυσης ενός υπο-προβλήματος διόρθωσης του εμβαδού διατομής, της υπο εξέταση γεωμετρίας, το οποίο αποτελείται από τον υπολογισμό και

την εφαρμογή των ελάχιστων δυνατών μετατοπίσεων στα σημεία ελέγχου του ήδη παραμορφωμένου πλέγματος της Ελεύθερης Παραμόρφωσης, ώστε το εμβαδόν της παραγόμενης γεωμετρίας ύστερα από την εφαρμογή των προαναφερθέντων μετατοπίσεων να ισούται με το εμβαδόν αναφοράς. Λόγω της γραμμικότητας του περιορισμού ισότητας του εμβαδού σε κάθε έναν από τους κύριους άξονες, είναι δυνατή η εξαγωγή λύσης κλειστής μορφής στο υπο-πρόβλημα, χρησιμοποιώντας την τεχνική των πολλαπλασιαστών Lagrange. Η προτεινόμενη μεθοδολογία αντιπαρέρχεται την αδυναμία των Εξελικτικών Αλγορίθμων σχετικά με την αποτελεσματική διαχείριση των αυστηρών περιορισμών ισότητας. Κατά τη διάρκεια της βελτιστοποίησης τόσο τα αεροδυναμικά αλλά και τα δομικά απαιτούμενα χαρακτηριστικά από μία αεροτομή λαμβάνονται υπόψιν. Επιπλέον, η χρησιμοποίηση πολλαπλών υποκατάστατων μοντέλων, σε συνδυασμό με την υπολογιστικά πολύ αποδοτική λύση στο υπο-πρόβλημα διόρθωσης του εμβαδού, καθιστούν την διαδικασία βελτιστοποίησης χρονικά αποδοτική. Στην παρούσα διπλωματική εργασία, η χρήση της προτεινόμενης μεθοδολογίας αποδεικνύει την εφαρμοσιμότητα και την αποτελεσματικότητά της.

## Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Associate Professor Ioannis K. Nikolos for the continuous support, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my undergraduate studies.

Also, I would like to express my heartfelt gratitude to my labmate and PhD candidate Giorgos A. Strofylas for his constant motivation and support during the period of our collaboration. I am thankful for his contribution to this thesis, for the stimulating discussions, and for all the fun we have had during the last year.

Finally but most importantly, none of this would have been possible without the love and patience of my beloved family and closest friends, which have been a constant source of love, concern, support and strength all these years. This thesis is dedicated to them.

*Stavros N. Leloudas*

*Chania, June 2015*

**“Intentionally Blank”**

# Table of Contents

<b>Abstract</b> .....	<b>V</b>
<b>Abstract (In Greek)</b> .....	<b>VII</b>
<b>Acknowledgements</b> .....	<b>IX</b>
<b>Table of Contents</b> .....	<b>XI</b>
<b>List of Figures</b> .....	<b>XIII</b>
<b>List of Tables</b> .....	<b>XV</b>
<b>List of Symbols</b> .....	<b>XV</b>
<b>Chapter 1 - Introduction</b> .....	<b>1</b>
1.1 Airfoil Shape Optimization .....	1
1.2 Geometrical Constraints .....	3
1.2.1 The Cross-Sectional Area Constraint in Airfoil Optimization .....	4
1.3 Evolutionary Algorithms and Equality Constraints .....	5
1.4 Literature Survey on Parameterization Techniques .....	6
1.4.1 Geometrical Parameterization Approaches .....	6
1.4.2 Parameterization Techniques in Aerodynamic Shape Optimization.....	11
1.4.3 Free-Form Deformation in Aerodynamic Shape Optimization.....	14
1.4.4 Free-Form Deformation and Area Preservation .....	16
1.5 The proposed approach .....	16
1.6 The Structure of the Diploma Thesis .....	17
<b>Chapter 2 - Free-Form Deformation</b> .....	<b>19</b>
2.1 Introduction to FFD Technique.....	19
2.1.1 Basic FFD Variations .....	19
2.1.2 FFD Methodology Steps.....	22
2.2 Two-Dimensional B-Spline FFD .....	23
2.2.1 The Implementation Procedure .....	23
2.2.2 Cross-Sectional Area Inequality .....	26

<b>Chapter 3 – Area-Preserving Free-Form Deformation .....</b>	<b>27</b>
3.1 Introduction to Area-Preserving FFD .....	27
3.2 Mathematical Analysis .....	27
3.2.1 The Area Preservation Problem.....	28
3.2.2 Cross-Sectional Area Calculation.....	28
3.2.3 The Area Constraint .....	30
3.2.4 The Area Correction Sub Problems.....	31
3.2.5 Closed-form Solution .....	34
<b>Chapter 4 – Differential Evolution Algorithm and Optimization Scheme.....</b>	<b>37</b>
4.1 Introduction to Evolutionary Algorithms .....	37
4.2 Differential Evolution .....	38
4.3 Surrogate Models .....	40
4.4 A surrogate-assisted DE algorithm .....	41
4.4.1 Artificial Neural Networks .....	41
4.4.2 Multi-Layer Perceptron ANN.....	42
4.4.3 Radial Basis Function ANN .....	43
4.4.4 The use of surrogates for accelerating DE algorithm .....	45
4.5 The Optimization Procedure .....	46
4.6 The Evaluation Software - XFOIL .....	48
<b>Chapter 5 – Validation .....</b>	<b>49</b>
5.1 High Lift, High Reynolds Number Airfoil Optimization.....	49
5.2 Computational Results .....	52
<b>Chapter 6 – Conclusions.....</b>	<b>59</b>
<b>References.....</b>	<b>60</b>

# List of Figures

Figure 1.1: Direct Airfoil Optimization .....	1
Figure 1.2: Wind turbine blade and the respective airfoil sections.....	2
Figure 1.3: Geometry of an airfoil section.....	3
Figure 1.4a: Initial model [26] .....	7
Figure 1.4b: Deformed model [26].....	7
Figure 1.5: Airfoil defined by a set of boundary points.....	8
Figure 1.6: Airfoil represented by a B-Spline curve.....	9
Figure 1.7: Airfoil section formed by a B-Spline with 15 control points [35] .....	12
Figure 1.8: Bezier curve representation of NACA4412 airfoil [36].....	12
Figure 1.9: Bezier approach used by Grasso [5].....	13
Figure 1.10: The 11 PARSEC design variables [37].....	13
Figure 1.11: Upper and lower airfoil’s curve parameterized by NURBS [41].....	14
Figure 1.12: 3D NFFD parametric lattice [42] .....	14
Figure 1.13a: Initial Lattice (white dots) [45].....	15
Figure 1.13b: Deformed Lattice (black dots) [45].....	15
Figure 1.14a: Initial FFD lattice and airfoil [46] .....	16
Figure 1.14b: Deformed airfoil (New) [46].....	16
Figure 2.1: Classic FFD application [32].....	19
Figure 2.2a: Lattice Structures [51] .....	20
Figure 2.2b Catmull-Clark volumes [51].....	20
Figure 2.3: Global deformation using a t-FFD [53].....	21
Figure 2.4a: Classic FFD Lattice [54].....	21
Figure 2.4b: w-TFFD lattice [54] .....	21
Figure 2.5: DU-06-W-200 airfoil in a 2D FFD lattice .....	25
Figure 2.6: The movement of the CPs causes the deformation of the lattice. ....	25
Figure 2.7: Deformed airfoil section.....	26
Figure 3.1: The positions of the control points and the lattice formation after the application of AP FFD .....	29

Figure 3.2: The resulting airfoil (green) after the application of the AP FFD .....	30
Figure 3.3: NACA 0012 airfoil inside an initial FFD lattice .....	33
Figure 3.4: The resulting airfoil after the application of a classic FFD .....	33
Figure 3.5: The positions of the control points and the lattice formation after the application of AP FFD, only in y direction.....	33
Figure 3.6: The resulting airfoil (green) after the application of the AP FFD, only in y direction .....	34
Figure 4.1: Principle flowchart of an Evolutionary process [59].....	38
Figure 4.2: A multilayer perceptron (MLP) artificial neural network with two hidden layers and a single output.....	42
Figure 4.3: A RBF ANN.....	44
Figure 4.4: Flowchart of the modified DE algorithm [58].....	45
Figure 4.5: Flowchart with the major steps of the optimization process including an AP FFD .....	47
Figure 4.6: A sample XFOIL output file.....	48
Figure 5.1: FFD Control Points Enumeration.....	50
Figure 5.2: Comparison between the lift coefficient of the reference airfoil DU-06-W-200 and the optimal one as a function of a.o.a .....	53
Figure 5.3: Initial and optimal airfoil geometries .....	54
Figure 5.4: The convergence history of the best and worst solution of each generation .....	54
Figure 5.5: The convergence history of the best solution of each generation by using an AP FFD and a penalty function approach for the satisfaction of the cross sectional area constraint.....	55
Figure 5.6: Comparison between the lift coefficient of the optimal airfoil produced by an optimization scheme that employs AP FFD and the optimal one produced by a classical optimization scheme utilizing a penalty function approach.....	55
Figure 5.7: Comparison between the drag coefficient of the reference airfoil DU-06-W-200 and the optimal as a function of a.o.a .....	56
Figure 5.8: Comparison between the lift-to-drag ratios of the reference DU-06-W-200	

airfoil and the optimal one. ....	57
Figure 5.9: Comparison between the pitching moment coefficient of the reference airfoil DU-06-W-200 and the optimal as a function of a.o.a. ....	57

## List of Tables

Table 1.1: Frequent geometrical characteristics under restriction .....	3
Table 2.1: Cross-Sectional Area Comparison .....	26
Table 5.1: Design Variables .....	50
Table 5.2: Wall-clock computation time, number of exact and total evaluations .....	56
Table 5.3: Output values for the constraints .....	58

## List of Symbols

$R_{ref}$	Initial (reference) airfoil geometry
$A_{ref}$	Reference cross-sectional area
$R_{def}$	Deformed airfoil by a classic FFD
$A_{def}$	Airfoil cross-sectional area after the FFD
$\bar{A}$	Cross-sectional area after the AP FFD
$\alpha$	Angle of attack, a.o.a
$C_l$	Lift coefficient
$C_d$	Drag coefficient
$C_m$	Pitching moment coefficient
$Re$	Reynolds number

**“Intentionally Blank”**

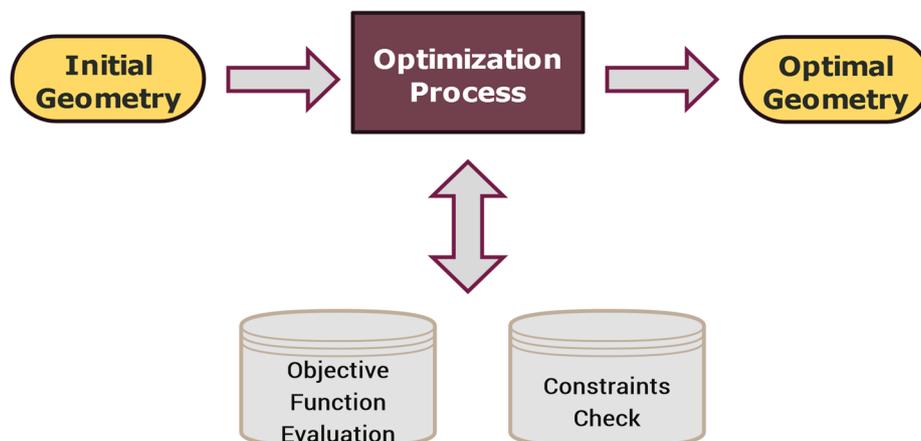
# Chapter 1

## Introduction

### 1.1 Airfoil Shape Optimization

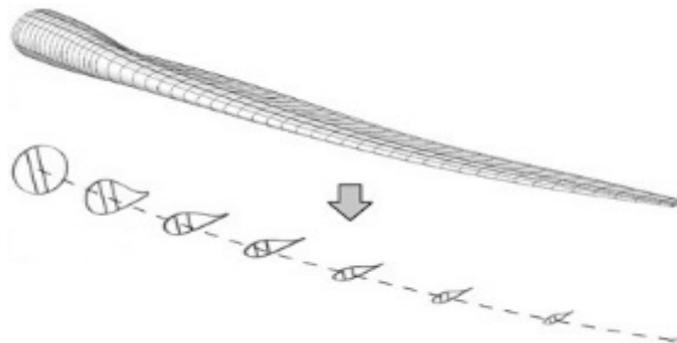
Since the early 80s, the systematic exploitation of wind turbines systems, as an alternative source of renewable energy, combined with the ever increasing demand of aviation industries for efficient aerodynamic shapes, extracted by automated and well-organized design cycles, has pushed researchers and engineers to the limits in order to develop powerful numerical optimization schemes for the design of high performance airfoil sections, employed to the aforementioned applications.

The direct airfoil optimization approach (Figure 1.1) refers to the design of high performance airfoils with respect to specific criteria, usually generated by the modification of a reference airfoil section having good performance to the selected criteria of the final design. Although the objectives of each problem typically diverse, depending on the application of interest, the most frequently utilized criteria concern the maximization or minimization of aerodynamic properties, such as the lift coefficient (maximization), the drag coefficient (minimization), the lift-to-drag ratio (maximization) as well as the smooth reaction after stall effects, subject to a variety of aerodynamic and geometrical constraints [1–6].



*Fig. 1.1: Direct Airfoil Optimization*

All the way through the design of an airfoil, aerodynamic efficiency and structural integrity are two strongly competitive properties acting in conflict. An optimization procedure concerned with high aerodynamic performance, leads the design in thin geometries with inadequate structural characteristics, while on the other hand, the opposite case leads in relatively thick geometries with no practical utilization in applications in which augmented aerodynamic efficiency is required. Hence, during the optimization process of an airfoil section, a fine compromise between the two aforementioned features has to be made, depending on the necessities of the application under study. In particular, structural requirements have higher priority than the aerodynamic ones, concerning airfoil sections which form the inner parts of a blade or wing, near the base, due to the enormous torsional stresses they go through, whereas increased aerodynamic properties are essential for airfoil sections located in the outer part, near the tip (Figure 1.2). The usual approach dictates the establishment of a proficient and adequate set of non-conflicting geometrical constraints, throughout the optimization process, to ensure the preservation of the necessary structural properties and guarantee the manufacturing feasibility of the final design, given a direct numerical optimization scheme focused on the improvement of aerodynamic performance [4, 7–11].



**Fig. 1.2:** Wind turbine blade and the respective airfoil sections

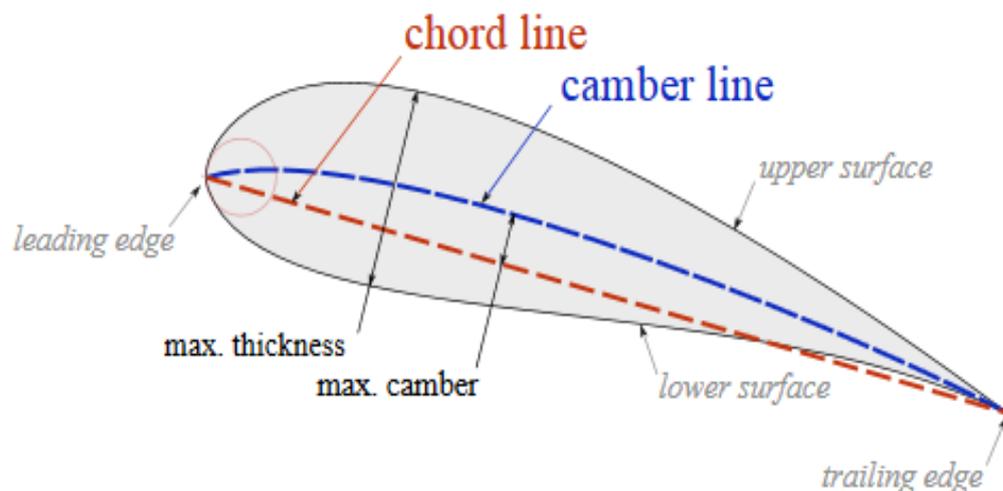
Under this prism, a strict cross-sectional area preservation constraint is commonly necessary, combined with further geometrical constraints to insure the required rigidity as well as the ability of the new blade/wing to sustain the expected loads without performing a detailed elastic analysis of each candidate geometry [8], having a structurally integral, as the initial one. Consequently, the development of efficient methods for the satisfaction of the mentioned constraints is rendered vital for the success of the design procedure.

## 1.2 Geometrical Constraints

The establishment of proficient geometrical constraints, in order to ensure the required structural properties, concerning a specific aerodynamic application, is a quite challenging task with significant impact to the effectiveness of the final design. Although finding geometric quantities related to structural integrity requires a simple literature overview, it is the proper importation of the mentioned parameters into applicable constraints along with the fine non-conflicting combination of the different constraints into an adequate set that requires a delicate handling by the designer. In Table 1.1 the most frequently placed under restriction geometrical characteristics, are presented [8], while in Figure 1.3 the geometry of an airfoil section is illustrated .

Geometrical Characteristics Under Restriction
Cross-Sectional Area
Minimum Thickness
Maximum Thickness
Thickness at Specific Location
Leading Edge Radius

**Tab. 1.1:** Frequent geometrical characteristics under restriction



**Fig. 1.3:** Geometry of an airfoil section

**Leading edge:** is the point at the front of the airfoil that has maximum curvature.

**Trailing edge:** is defined similarly as the point of maximum curvature at the rear of the airfoil.

**Chord line:** A straight line from leading edge to trailing edge.

**Mean camber line (MCL):** The line midway between upper and lower surfaces. The MCL for a cambered airfoil necessarily rises above the chord line. The MCL for a symmetric airfoil is coincidental (same as) the chord line itself.

**Maximum thickness:** Maximum thickness of the airfoil is the maximum distance from the bottom edge to the top edge.

**Maximum thickness location:** The maximum thickness location is the point along the chord line where maximum thickness occurs.

**Maximum camber:** The maximum separation of the MCL from the chord line. Maximum camber is normally expressed as a percentage or fraction of the chord.

### 1.2.1 The Cross-Sectional Area Constraint in Airfoil Optimization

In the works of Leifsson and Koziel, Lee and Eyi [3,12], a cross-sectional area inequality constraint is utilized in conjunction with lift and pitching moment ones, during the optimization of transonic airfoils in order to ensure that the cross-sectional area of the final design will not be smaller than a limit value while Jeong et al. [13] proposed a drag minimization scheme for transonic airfoils subject to a cross-sectional area equality constraint coupled with PARSEC parameterization technique. Zingg et al. [14] imposed a combination of several thickness constraints to prevent crossover between the upper and the lower curves of the airfoil, along with a cross-sectional area inequality constraint, to provide suitable internal volume for fuel storage as well as adequate structural properties, while the control points correspond to leading and trailing edges were set fixed so the chord length not to change. Furthermore, Ahn et al. [10] proposed an optimization scheme that utilizes either a cross-sectional area constraint or a maximum thickness one, combined with a constraint that concerns the minimum acceptable value of lift coefficient throughout a drag minimization problem. On the other hand Dennis et al. [8] established a mix of geometrical constraints, including an equality cross-sectional area one, during the redesign of an existing two-dimensional cascade of supersonic exit turbine airfoils while Nikolos [4] suggested the combination of a cross-sectional area equality constraint, with an inequality one

regarding the coordinate of the center of gravity of the airfoil in the chord direction, as long as concerns the geometric point of view, in a study for high-lift airfoil design. Finally, Driver and Zingg [15] imposed an area preservation constraint on a lift-to-drag ratio maximization problem by using a NACA-0012 as the baseline airfoil.

Although the cross-sectional area of the final design approximated the cross-sectional area of initial model in the most of the aforementioned cases, however a strict area preservation was not achieved. It is therefore obvious that the handling and the satisfaction of the required constraints is a truly challenging endeavor in numerical optimization, arising from the strong nonlinearity and complexity usually found in real-world problems and involves the cooperation of both the optimization algorithm and the geometrical parameterization technique for its overcoming.

### **1.3 Evolutionary Algorithms and Equality Constraints**

During the last years, Evolutionary Algorithms (EAs) have been extensively applied in constrained engineering optimization problems [4,16–19] as a quite versatile, effective and robust optimization technique, capable of global searching and dealing with complicated and very demanding real-world problems [20]. However, despite the unquestionable advantages of EAs, the handling of strict nonlinear equality constraints, such an area preservation one, has been proved as a major drawback related to the efficiency of evolutionary methods in optimization. In order to overcome this barrier, several approaches have been developed, which can be grouped as [21]: penalty functions, special representations and operators, repair algorithms, separation of objectives and constraints, and hybrid methods. Among them, penalty function is the most commonly utilized approach to handle so inequality as equality constraints.

A penalty method replaces a constrained optimization problem by a series of unconstrained problems whose solutions ideally converge to the solution of the original constrained problem. The unconstrained problems are formed by adding a term, called a penalty function, to the objective function that consists of a penalty factor multiplied by a measure of violation of the constraints. The measure of violation is nonzero when the constraints are violated and is zero in the region where constraints are not violated. Though, the utilization of penalty functions requires the proper establishment of specific penalty factors, depending on the optimization problem under consideration,

which constitutes a major disadvantage concerning the robustness of the whole optimization method [22,23].

Consequently the utilization of penalty functions is not always sufficient and does not guarantee the exact satisfaction of a strict equality constraint. Nevertheless, penalty function approaches have been widely employed in the field of airfoil optimization to ensure the met of cross-sectional area constraints, either in an equality or an inequality form, where a minimum acceptable value is set [3,4,8,10,13–15,24].

## 1.4 Literature Survey on Parameterization Techniques

In this section, at first a comprehensive literature overview is conducted, on the most frequently utilized geometric parameterization techniques for multidisciplinary optimization and subsequently, a brief introduction to the basic ideas behind the Free-Form Deformation (FFD) methodology, including its use as a parameterization technique in Aerodynamic Shape Optimization, is presented. Finally the shortcoming FFD methodology to preserve the area or volume enclosed by the initial shape after its application, is highlighted.

### 1.4.1 Geometrical Parameterization Approaches

At the very starting point of an optimization procedure, the designer has to deal with a vital question, which will affect the entire process. The quest concerns the selection of the parameterization technique in order to describe the shape of interest with a small but sufficient number of parameters. According to Andreoli et al. [25] the selected parameterization technique should be:

- **versatile:** possibility to describe quite a broad spectrum of potentially complex 2D or 3D shapes.
  
- **concise:** it should use as few parameters as possible, because the number of design parameters heavily affect the CPU cost of the optimization process.

Herein, an introduction to nine major parameterization methodologies is implemented, as presented in the work of Samareh [26].

### Basis Vector Approach

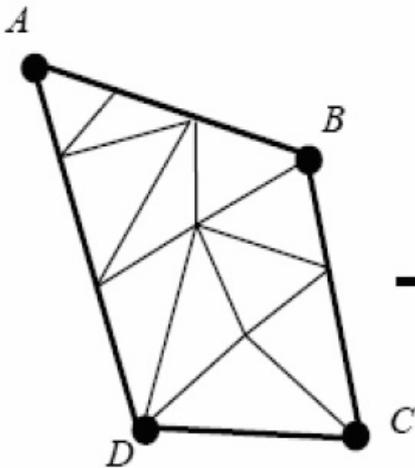
Picket et al. [27] proposed a technique that combines three partial derivative terms into a set of basis vectors. The geometry of the shape changes according to the following equation (Eq. (1.1)):

$$R = r + \sum_n u_n U_n \quad (1.1)$$

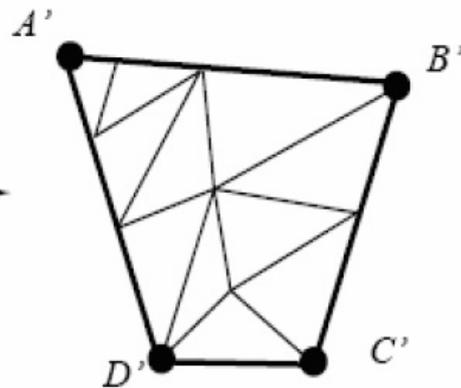
where  $R$ ,  $r$  are the resulting and the initial shapes respectively,  $u_n$  are the design variables and  $U_n$  the design perturbations based on several proposed shapes. Assuming that the basis vectors remain constant throughout the optimization cycle, this technique is a good approach. However, it is difficult to generate a set of consistent basis vectors for multiple disciplines. As a result, this method can be applied only to problems involving a simple discipline with simple geometry changes.

### Domain Element Approach

The technique is based on linking a set of grid points to a macro element, called domain element that controls the shape of the model. In Figure 1.4a a domain element formed by 4 nodes ( $A, B, C, D$ ) is presented, while in Figure 1.4b the movement of the grid points belonging to the element is illustrated, caused by the displacement of the 4 nodes ( $A', B', C', D'$ ).



**Fig. 1.4a:** Initial model [26]



**Fig. 1.4b:** Deformed model [26]

The connection between the grid points and the domain element is implemented through an inverse mapping. The parametric coordinates of each grid point with respect

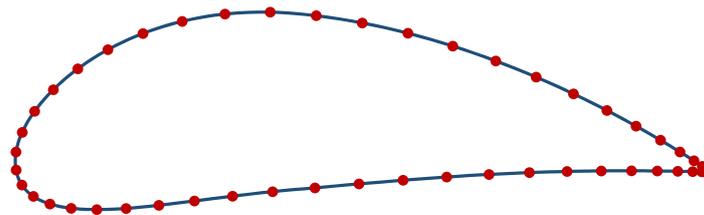
to domain element remain constant during the alteration of the element's shape. The domain element approach, as well as the aforementioned Basis Vector Approach, are capable to handle only simple geometry deformations, so their utilization is limited.

### Partial Differential Equation Approach

Bloor and Wilson [28] presented a compact and effective technique for the parameterization of the surfaces of an airplane. The method transforms the surface generation into a boundary value problem and produces surfaces as solutions to elliptic Partial Differential Equations (PDE).

### Discrete Approach

The discrete approach dictates the utilization of the coordinates of the boundary points of the shape as design variables (Figure 1.5).

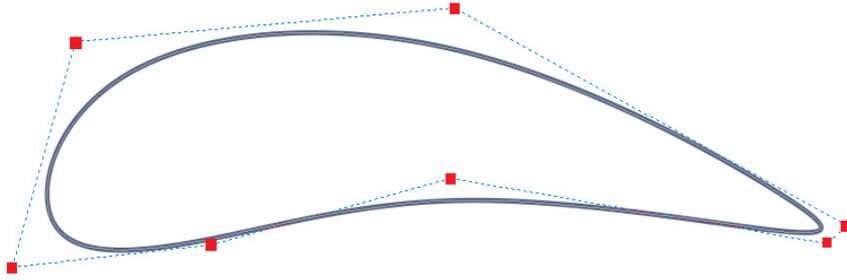


*Fig. 1.5: Airfoil defined by a set of boundary points*

This approach is easy to implement, and the geometry changes are limited only by the number of design variables. However, it is difficult to maintain a smooth geometry and the optimum solution may be impractical to manufacture. Additionally, for a model with a large number of boundary points, the number of design variables becomes very large, which leads to high cost and a difficult optimization problem to solve.

### Polynomial and Spline Approach

In this approach, the geometry of the interested shape is represented by a compact polynomial curve, which could be defined by a relative small number of control points. Therefore, by the utilization of the coordinates of the control points as design variables during the optimization procedure, a significant reduction of the design variables is achieved. In Figure 1.6 an airfoil section formed by a 3<sup>rd</sup> degree B-Spline curve, consisted by 7 control points, is illustrated.



**Fig. 1.6:** Airfoil represented by a B-Spline curve

Thus, the geometry of the airfoil section presented in Figure 1.6 could be totally described by the maximum number of 14 design variables, which are the  $x$  and  $y$  coordinates of the 7 control points. Additionally, further reduction of the design variables is possible by enforcing movement constraints to particular control points. As long as airfoil parameterization by polynomial curves is concerned, the control points corresponding to leading and trailing edges are usually fixed in order to maintain the chord length unchanged during the optimization procedure. Also, a common approach is to utilize only the  $y$  coordinates of the control points as design variables.

A  $p^{th}$  degree B-Spline curve is defined by

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, \quad a \leq u \leq b \quad (1.2)$$

where  $\mathbf{P}_i = (x_i, y_i)$  are the control points and  $N_{i,p}(u)$  are the  $p^{th}$  degree B-Spline Basis Functions defined over the nonperiodic knot vector [29]:

$$\mathbf{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (1.3)$$

Instead of B-Spline curves, Non-Uniform Rational B-Splines (NURBS) and Bézier curves could be used as well for the parameterization of two-dimensional shapes. The Bézier form is an effective and accurate representation for shape optimization of simple curves, however, complex curves require a high degree Bézier curve. Given that a Bézier curve consisted by  $p$  control points is a polynomial of  $p - 1$  degree, the representation of complex shapes, with many control points, render the computational procedure of the curve inefficient.

A  $n^{\text{th}}$  degree Bézier curve [29] is defined by:

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i, \quad 0 \leq u \leq 1 \quad (1.4)$$

The basis function  $B_{i,n}(u)$  are the classical  $n^{\text{th}}$  degree Bernstein polynomial given by:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (1.5)$$

Although B-Spline curves offer great properties such as partition of unity, flexibility and convenient geometrical deformation via control points, they are incapable of representing any conic section (hyperbola, parabola, ellipse, etc.) accurately. In order to overcome this particular shortcoming, an extension of B-Spline theory is necessary. NURBS curves are a special form of B-Splines, which can represent virtually any desired shape, from points, straight lines, and polylines to conic sections and free-form curves with arbitrary shapes [26]. A  $p^{\text{th}}$  degree NURBS curve is defined by:

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) w_i}, \quad a \leq u \leq b \quad (1.6)$$

where  $\mathbf{P}_i = (x_i, y_i)$  are the control points,  $w_i$  is the weight of the  $i^{\text{th}}$  control point and  $N_{i,p}(u)$  are the  $p^{\text{th}}$  degree B-Spline Basis Functions defined over the nonperiodic knot vector  $\mathbf{U}$ , defined in Eq. (1.3). Other types of knot vectors can be used as well.

### CAD-Based Approach

Most solid modeling CAD systems use either a boundary representation (B-Rep) or a constructive solid geometry method to represent a physical, solid object [30]. The parameterization of a given model constitutes so far a difficult task for CAD software given that any inaccuracy on the parametric model causes significant problems concerning the automated computational grid generation procedure.

### Analytical Approach

Hicks and Henne [31] introduced a compact parameterization of airfoil sections. The formulation was based on adding shape functions (analytical functions) linearly to the baseline shape. The contribution of each parameter is determined by the value of

the participating coefficients (design variables) associated with that function. All participating coefficients are initially set to zero, so the first computation gives the baseline geometry. The shape functions are smooth functions based on a set of previous airfoil designs. This method is very effective for wing parameterization, but it is difficult to generalize it for a complex geometry.

### **Free-Form Deformation Approach**

Free-Form Deformation is a powerful deformation and parameterization technique introduced by Sederberg and Parry [32], established on the work of Barr [33], for the deformation of solid geometric models. The original version was based on the indirect manipulation of the object by enclosing it into a parametric 3D space, formed by trivariate Bernstein polynomials. Then, by deforming the parametric lattice, a deformation of the embedded model is achieved. Since then, numerous different versions of FFD have been proposed, with major applications in a great variety of technological fields. Although FFD was first presented in order to handle 3D entities in the field of computer graphics, during the last decades there is an intense utilization of FFD versions concerning aerodynamic shape parameterization.

### **Multidisciplinary Aero/Struc Shape Optimization Using Deformation**

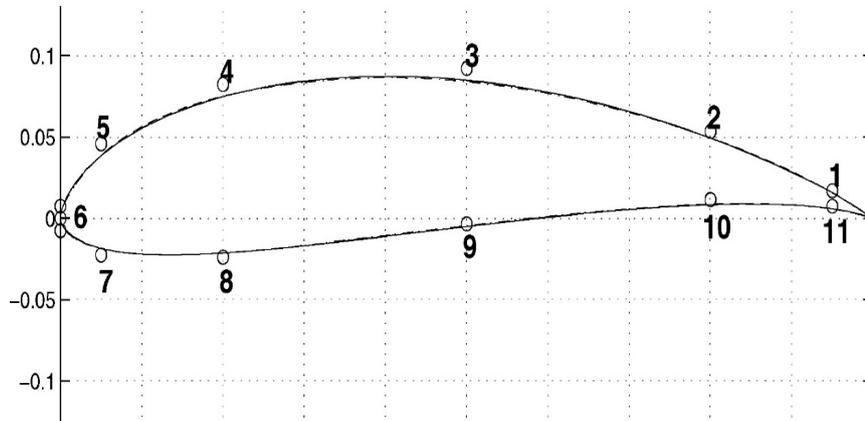
#### **(MASSOUD) Approach**

MASSOUD is a novel parameterization approach for complex shapes suitable for a multidisciplinary design optimization application. The approach is an extension of FFD in which the design variables acquire physical significance. The proposed methodology is implemented into three main steps; initially the shape's perturbation is parameterized rather than the geometry itself, then Soft Object Animation (SOA) algorithms are used and finally the deformation is related to the aerodynamic shape's physical design variables.

#### **1.4.2 Parameterization Techniques in Aerodynamic Shape Optimization**

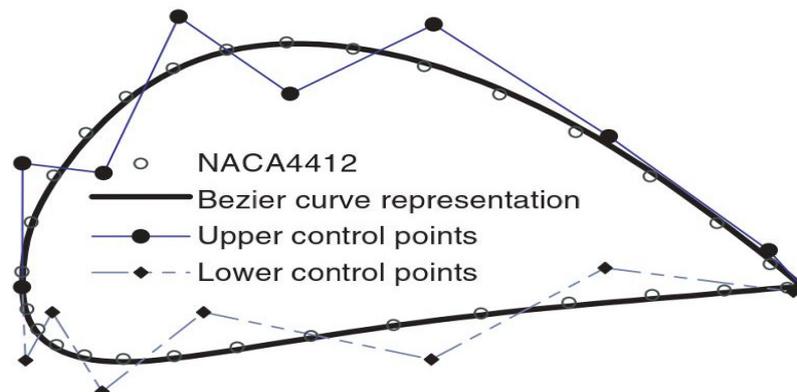
Regarding the field of Aerodynamic Shape Optimization, Oyama and Fujii [34] utilized B-Spline curves to parameterize the geometry of the airfoil, setting the control points on the leading and trailing edges to be fixed and using the  $x$  and  $z$  coordinates of the control points as design variables.

Secanell and Suleman [35] chose a uniform cubic B-Spline with 15 control points to represent the airfoil section, while from the 15 control points, the y coordinates of control points numbered 1–5 and 7–11 in Figure 1.7, are used as design variables.



**Fig. 1.7:** Airfoil section formed by a B-Spline with 15 control points [35]

In [36] the airfoil shape is parameterized using two Bézier curves, one for the upper and one for lower surface with the y coordinates of the free-to-move control points in the role of design variables (Figure 1.8), while the control points representing the leading and trailing edge were fixed.



**Fig. 1.8:** Bezier curves representation of NACA4412 airfoil [36]

Grasso [5] used a composite parameterization based on Bézier curves; the airfoil is separated into four different segments, each one represented by a cubic Bézier curve (Figure 1.9), with continuity constraints in order to achieve local control and easy handling.

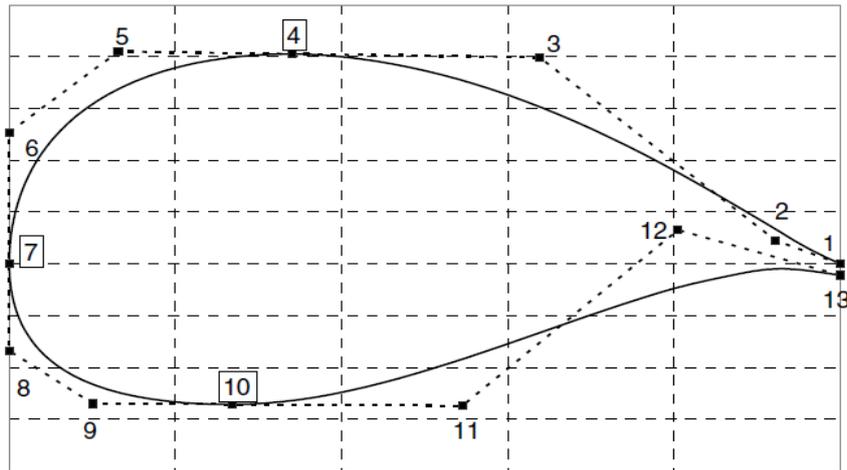


Fig. 1.9: Bezier approach used by Grasso [5]

In the recent works of Della Vecchia et al. [37] a different approach is presented by the utilization of PARSEC parameterization technique introduced by Sobieczky [38,39], where 11 design variables with physical meaning concerning the geometry of the airfoil are used (Figure 1.10). While, earlier, Kharal and Saleem [40] combined the Bezier-PARSEC method with a Genetic Algorithm and three different neural networks to extract the parameters which describe the airfoil with respect to a given pressure distribution.

Liang et al. [41] took advantage of the beneficial properties of Non-Uniform Rational B-Splines (NURBS) to separately parameterize the upper and the lower surface of the airfoil (Figure 1.11). The design variables were formed by the  $x, y$  coordinates and the weights of the 14 free-to-move control points.

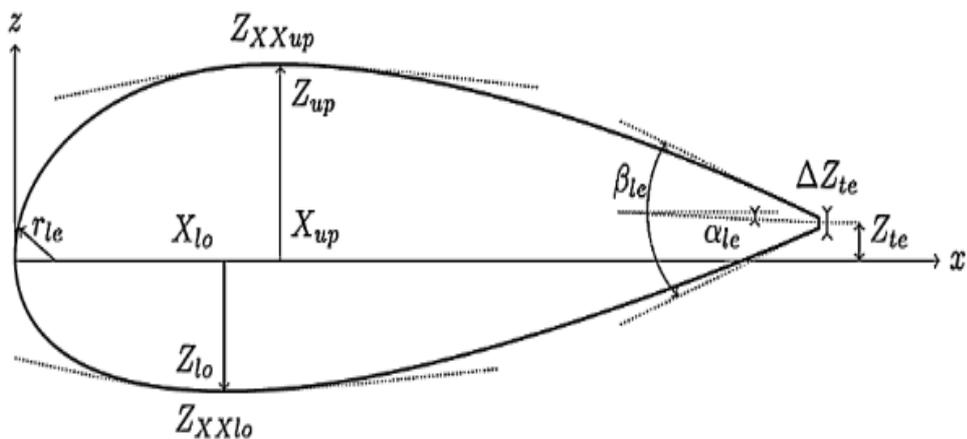


Fig. 1.10: The 11 PARSEC design variables [37]

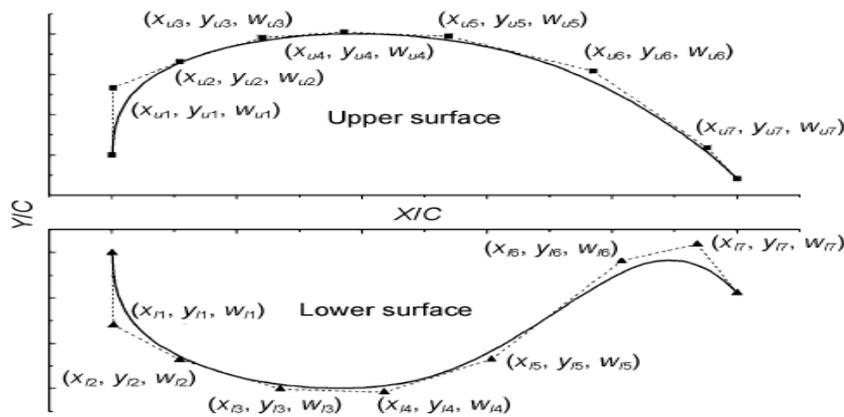


Fig. 1.11: Upper and lower airfoil's curve parameterized by NURBS [41]

### 1.4.3 Free-Form Deformation in Aerodynamic Shape Optimization

FFD was first introduced to model simple deformations of rigid objects in the field of computer graphics. Since then, the utilization of the FFD methodology as parameterization and deformation technique has spread in a variety of scientific fields due to its great versatility and applicability. Although the first proposed version of FFD by Sederberg and Parry [32] referred to the manipulation of 3D shapes by using Bernstein polynomials to form the control grid (lattice), during the last years many different versions have been proposed regarding the developments of 2D versions along with the alteration of the parametric curves forming the FFD lattice.

Samareh [42] used a NURBS-based FFD (NFFD), which was first introduced by Lamousin and Waggenspack [43], in order to parameterize aerodynamic CSM (Computational Structural Mechanics) models. The FFD lattice (Figure 1.12) is formed by trivariate B-Spline basis functions defined over non-uniform knot vectors.

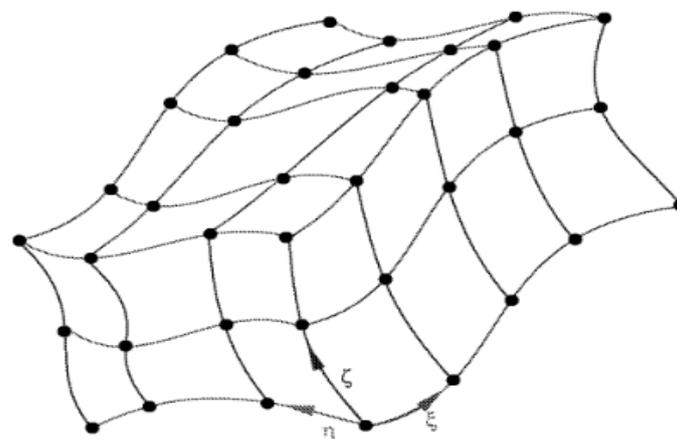
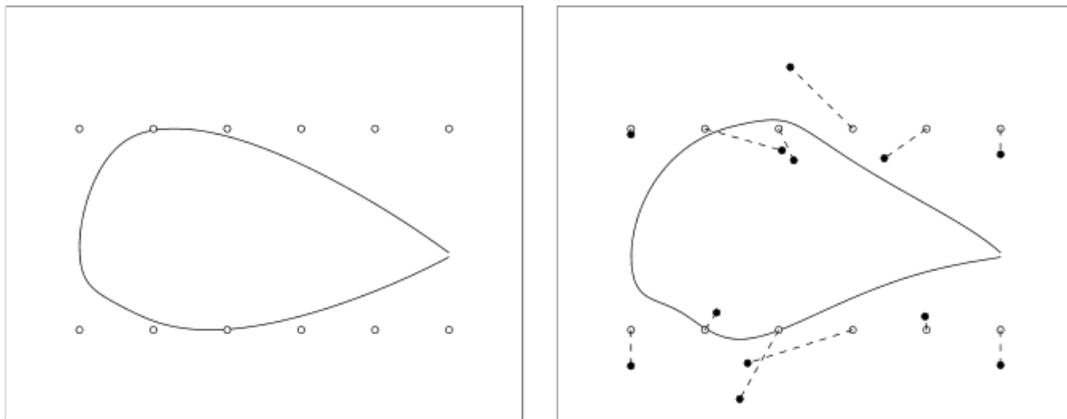


Fig. 1.12: 3D NFFD parametric lattice [42]

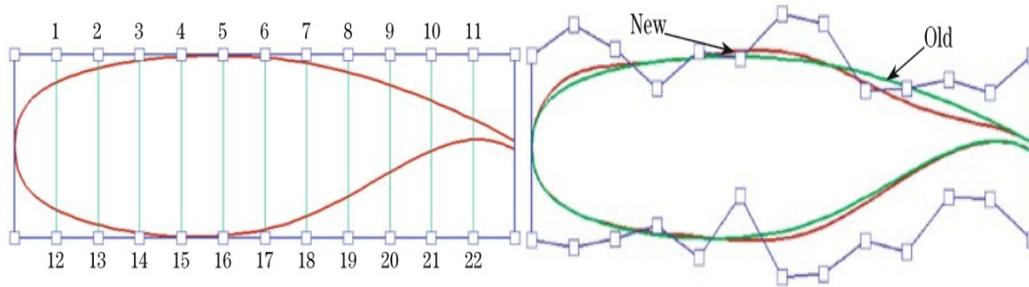
Later on, Amoiralis and Nikolos [44] performed a comparative study between FFD and B-Spline representation. The tests were been carried out during inverse airfoil design procedures, showing that FFD approach outperformed the classic B-Splines one concerning the achieved accuracy in the approximation of the reference pressure distributions.

Ghisu et al. [45] used a FFD formed by a tensor product trivariate Bernstein polynomial in order to handle airfoil sections during an optimization process with respect to ice accretion. The FFD lattice consisted by 12 control points (6x2x1), while both displacements in the  $x$  and  $y$  axis were permitted for the control points that were not at the extreme left and right positions (leading and trailing edges) in order to maintain the chord length during the design process. For the control points corresponding to the extreme (right and left) positions, a movement only along the  $y$  axis was permitted. In Figure 1.13a, the airfoil embedded to the initial FFD lattice is illustrated, while in Figure 1.13b the deformation of the airfoil caused by the movement of the control points is presented.



**Fig. 1.13a:** Initial Lattice (white dots)[45]      **Fig. 1.13b:** Deformed Lattice (black dots)[45]

In [46], a NFFD approach was utilized as parameterization technique during the design process of natural laminar flow supercritical airfoils, while the non-dominated sorting genetic algorithm II (NSGA-II) served as the optimizer. In the particular application, the FFD lattice formed by 26 (2x13) control points (Figure 1.14a) and the design variables set as the  $y$  coordinates of the control points, excluding those which were located at the leading and trailing edges of the airfoil, as shown in Figure 1.14b.



**Fig. 1.14a:** Initial FFD lattice and airfoil [46] **Fig. 1.14b:** Deformed airfoil (New) [46]

Amoignon et al. [47] at first, examined the influence of the basis function's degree on the performance of a smooth airfoil, parameterized with a NFFD, showing that the highest possible degree improves the performance of the airfoil through a specific optimization scheme, keeping constant the number of design parameters. Subsequently, a combination of the FFD technique with Radial Basis Functions (RBF) was implemented, in order to handle more effectively the deformations of shapes with complex geometries and finally an adaptive FFD lattice method was proposed to achieve a greater reduction to the cost function.

#### 1.4.4 Free-Form Deformation and Area Preservation

Even though FFD is a versatile deformation tool, the area (2D version) or volume (3D version) preservation property after its application is not provided. This is a drawback throughout the optimization process of airfoil sections (or other objects), where the baseline cross-sectional area should be preserved mainly for reasons concerning structural integrity, as described above.

### 1.5 The proposed approach

In this thesis an alternative technique for the exact preservation of the cross-sectional area enclosed by an airfoil or by any given 2D shape, called Area-Preserving Free-Form Deformation (AP FFD), is developed by coupling a classic 2D B-Spline-based FFD with an area correction step, based on the work of Hahmann et al. about a Volume-Preserving FFD [48]. The proposed technique takes action after the application of a typical 2D FFD, which usually produces deformed geometries with no respect on area preservation. The AP FFD utilizes the Lagrange Multipliers optimization method to compute the minimum required displacements, which have to be applied at the control points of the deformed lattice, in order to preserve the enclosed area of the initial

(non-deformed) model whereas the adjustment to the FFD lattice should be minimal in order to respect the deformation prescribed by the user or the optimizer.

## **1.6 The Structure of the Diploma Thesis**

The rest of this diploma thesis is organized as follows:

- In **Chapter 2**, at first a brief introduction to the major FFD variations is conducted and then an extensive presentation of a 2D B-Spline-based FFD is included.
- In **Chapter 3** a detailed demonstration and formalization of the utilized AP FFD methodology is implemented.
- In **Chapter 4** the surrogate-assisted Differential Evolution algorithm as well as the whole optimization scheme are outlined.
- In **Chapter 5** the AP FFD methodology is integrated within a numerical optimization scheme and is tested, to prove its efficiency and effectiveness.
- In **Chapter 6** the conclusions resulting from the application of the proposed methodology are presented and discussed.

**“Intentionally Blank”**

# Chapter 2

## Free-Form Deformation

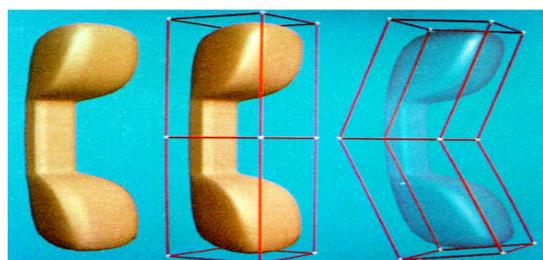
### 2.1 Introduction to FFD Technique

Free-form Deformation is considered as one of the most powerful, versatile and applicable parameterization and deformation techniques of two- as well as three-dimensional shapes with major applications in a wide range of scientific and technological fields such as aerodynamic design, computer graphics and medicine, related to cancer detection procedures.

The fundamental idea of FFD is the indirect handling of the interested object by enclosing it into a parametric space (2D or 3D). Then, by deforming the parametric lattice, a deformation of the embedded model is achieved. Based on this idea, a large number of different FFD versions was developed during the last years.

#### 2.1.1 Basic FFD Variations

Barr [33], at first, studied and developed new hierarchical solid modeling operations in order to simulate transformations of geometric objects. A bit later, Sederberg and Parry [32], based on [33], introduced the first version of FFD (classic FFD) for the deformation of solid geometric objects in a free-form manner. In their approach, the lattice is defined as trivariate tensor product of Bernstein polynomials and the control parameters are actually the coefficients of the polynomials. By altering these control parameters, the object embedded in the lattice is deformed, as shown in Figure 2.1.



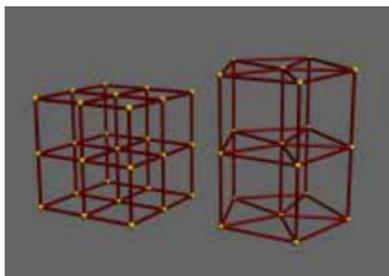
*Fig. 2.1: Classic FFD application [32]*

Griessmair and Purgathofer [49] modified the classic FFD approach proposed by Sederberg and Parry by using B-Spline instead Bernstein polynomials for the construction of the parametric space.

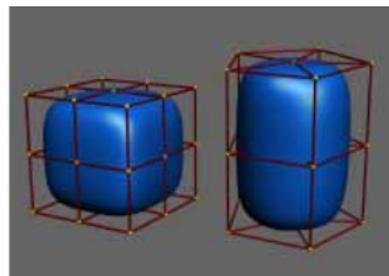
Coquillart [50] proposed the Extended Free-Form Deformation (EFFD) technique to model cloth-like surfaces with natural characteristics. In order to achieve the aforementioned goal, Coquillart was the first to propose the utilization of arbitrary topology lattices for the fittest combination between the object under deformation and the FFD lattice.

Lamousin and Waggenpack [43] introduced the NURBS-based FFD (NFFD) technique, in which the construction of the lattice was implemented by B-Spline basis functions defined over non-uniform knot vectors. The combination of rational B-Spline basis functions with the non-uniform knot vectors provides additional functionality for controlling the deformations of the enclosed object.

In the work of MacCracken and Joy [51] a new free-form deformation technique is presented that generalizes previous methods by allowing 3-dimensional deformation lattices of arbitrary topology, based on the work of Coquillart [50]. The technique uses an extension of the Catmull-Clark subdivision methodology to successively refine a 3-dimensional lattice into a sequence of lattices that converge uniformly to a region of 3-dimensional space. In Figure 2.2a the initial lattice structures are illustrated, while in Figure 2.2b are presented the defined Catmull-Clark volumes by the respective lattices.



**Fig. 2.2a:** Lattice Structures [51]

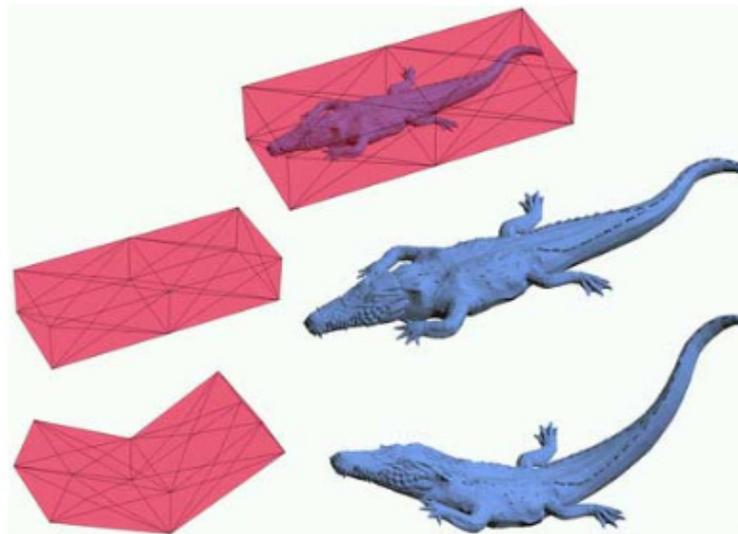


**Fig. 2.2b** Catmull-Clark volumes [51]

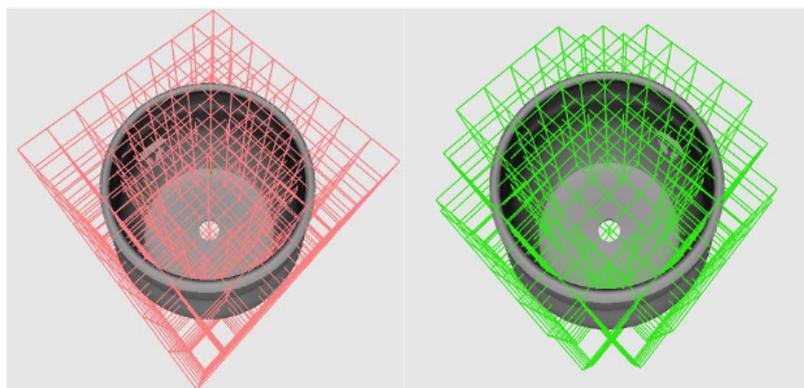
In the work of Ilic and Fua [52], the utilization of the powerful DFFD (Dirichlet FFD) extension instead of the conventional FFD shape deformation approach to fit deformable surface models to noisy 3D image data, is proposed. DFFDs provide to the user the ability to place control points at arbitrary locations rather than on a regular lattice, and thus leads to a much greater flexibility.

Kobayashi and Ootsubo [53] proposed a new version of FFD, called t-FFD. In this approach, an initial shape of large-scale polygonal mesh or point cloud is deformed by using a control mesh, which is composed of a set of triangles with arbitrary topology and geometry as the control lattice (Figure 2.3).

Later on, Song and Yang [54], based on the work of Sederberg et al. [55] concerning T-Splines theory, introduced a novel FFD version, called weighted-TFFD (w-TFFD). In this particular approach, the FFD lattice is formed by trivariate weighted T-Spline volumes which permit T-junctions. Weighted T-spline volumes are a natural generalization of NURBS volumes but permit more flexible control lattices. Thus, w-TFFD holds many virtues of traditional FFDs and is more adaptive to objects with arbitrary topology or complex shape. In Figures 2.4a and 2.4b the difference between a complete control lattice (red) and a w-TFFD lattice (green), is illustrated.



**Fig. 2.3:** Global deformation using a t-FFD [53]



**Fig. 2.4a:** Classic FFD Lattice [54]    **Fig. 2.4b:** w-TFFD lattice [54]

### 2.1.2 FFD Methodology Steps

Despite the numerous different FFD versions, either for 2D or 3D geometric models, according to Lamousin and Waggenspack [43] there are four main steps to implement any FFD technique.

#### Step 1: Construction of the Parametric Lattice

The shape of interest, is embedded into a two- or three- dimensional parametric space (lattice) consisted by an ordered or arbitrary mesh of control points, depending on the particular FFD version and application. The topology of the parametric lattice should be such that it wraps the embedded shape under study, while the nature of the basis function that form the FFD lattice has a significant impact on the handling of the embedded model.

#### Step 2: Embedding the Object within the Lattice

This stage consist of the assignment of a unique set of parametric coordinates ( $u, v, w$ ) to each point ( $x, y, z$ ) of the enclosed shape, where  $u, v, w$  are the parametric variables that define the parametric coordinates system. Herein, it is useful to note that the parametric coordinates of each point of the model do not change during the deformation stage, respect to the specific parametric coordinates system. Due to a lack of analytical methodologies for the aforementioned assignment problem, approximate methodologies have developed such as Quadtree and Octree for 2D and 3D problems respectively.

#### Step 3: Deforming the Parametric Space

The deformation of the parametric space is implemented by the movement of the points of the control grid. Especially as long as concern the weighted FFD versions, a deformation could be also achieved only by the modification of the weights of the control points.

#### Step 4: Evaluating the Effects of the Deformation

The parametric coordinates of the points (*Step 2*) are used with the deformed control lattice (*Step 3*) to evaluate the new locations of the embedded point set. The topology of the original model is then used to reconstruct the deformed object.

## 2.2 Two-Dimensional B-Spline FFD

In this diploma thesis, due to the nature of Airfoil Optimization Problem, a 2D B-Spline-based FFD version is utilized, which is defined as a mapping from a subspace  $V \subset R^2 \rightarrow \bar{V} \subset R^2$ . The main idea behind FFD is not to directly deform the shape of interest, but to achieve an indirect manipulation by embedding the object into a parametric control grid (lattice)  $V$ ; then by transforming the geometry of the particular lattice, every object enclosed to it undergoes the same deformation  $\bar{V}$ .

### 2.2.1 The Implementation Procedure

#### Step 1: Construction of the Parametric Lattice

In this application of FFD, a 2D lattice, formed by a two-variate B-Spline, is chosen to take advantage of the benefits B-Splines offer, such as partition of unity, flexibility and convenient geometrical deformation via control points. Additionally, by using a B-Spline lattice the alteration of a control point does not modulate the entire geometry of the enclosed object, so a focused deformation can be achieved. A planar B-Spline surface is obtained by taking a bidirectional net of control points, two knot vectors, and the products of the univariate B-Spline functions [29]

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij}, \quad \mathbf{P}_{ij} = (x_{ij}, y_{ij}) \quad (2.1)$$

where  $\mathbf{P}_{ij}$  are the control points of the FFD lattice,  $n + 1, m + 1$  are the number of control points on each parametric direction and  $u, v$  are the parametric variables that define the parametric coordinates system. Let  $N_{i,p}(u)$  be a B-Spline Basis function of  $p$  degree in  $u$  parametric direction and  $N_{j,q}(v)$  be a B-Spline Basis function of  $q$  degree in  $v$  parametric direction defined over the open and uniform knot vectors  $\mathbf{U}, \mathbf{V}$  respectively

$$\mathbf{U} = \{u_0, u_1, \dots, u_{n+p+1}\} \quad (2.2)$$

$$\mathbf{V} = \{v_0, v_1, \dots, v_{m+q+1}\} \quad (2.3)$$

Concerning the degrees  $p, q$  of the Basis functions, they must satisfy the following inequalities

$$1 \leq p \leq n, \quad 1 \leq q \leq m \quad (2.4), (2.5)$$

The value of each knot of the  $\mathbf{U}$  knot vector is calculated by the following formula:

$$u_i = \begin{cases} 0, & 0 \leq i \leq p + 1 \\ i - p, & p + 1 \leq i \leq n + 1 \\ n - p + 1, & n + 1 \leq i \leq n + p + 1 \end{cases} \quad (2.6)$$

while the calculation of the values of the  $V$  knot vector is implemented respectively. The  $i$ -th B-spline basis function of degree  $p$ , written as  $N_{i,p}(u)$ , is defined by the utilization of the *Cox-de Boor* recursion formula, as follows:

$$N_{i,p}(u) = \frac{u - u_i}{u_{i-p} - u_i} N_{i,p-1} + \frac{u_{i+1,p-1} - u}{u_{i+p+1} - u_{i+1}} \quad (2.7)$$

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Given that the parametric coordinates  $(u_t, v_t)$  of a point inside the parametric space are known, then the vector of the respective Cartesian coordinates  $(x_t, y_t)$  is calculated by the following equation:

$$R(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v)} \quad (2.8)$$

### Step 2: Embedding the Object within the Lattice

After the construction of the FFD lattice, a quadtree algorithm has to be implemented, so a unique parametric pair of coordinates  $(u_t, v_t)$  to be assigned in every single point  $(x_t, y_t)$  of the shape to be deformed (the airfoil in our case). For each point of the object, the following algorithm is repeatedly applied [44].

- a. The parametric area is divided into four equal subareas.
- b. The Cartesian coordinates of each subarea vertex are calculated using Eq. (2.8)
- c. These coordinates are compared to the Cartesian coordinates of the object's point under consideration, in order to identify the subarea in which the corresponding point lies.
- d. The latter subarea is divided into four new equal subareas and steps b-d are repeated for a prescribed number of subdivisions, or until a desirable accuracy is achieved. The desired parametric coordinates of the searched

point are defined as the parametric coordinates of the center of the subarea, in which the point resides, resulting from the last subdivision [56].

In Figure 2.5, a reference airfoil (DU-06-W-200) is embedded into an initial 2D FFD lattice formed by a two-variate B-Spline function. The parametric space is defined by the parametric coordinates  $(u, v)$  while a unique pair of parametric coordinates  $(u_t, v_t)$  has been assigned to each one of the  $k$  boundary points of the airfoil.

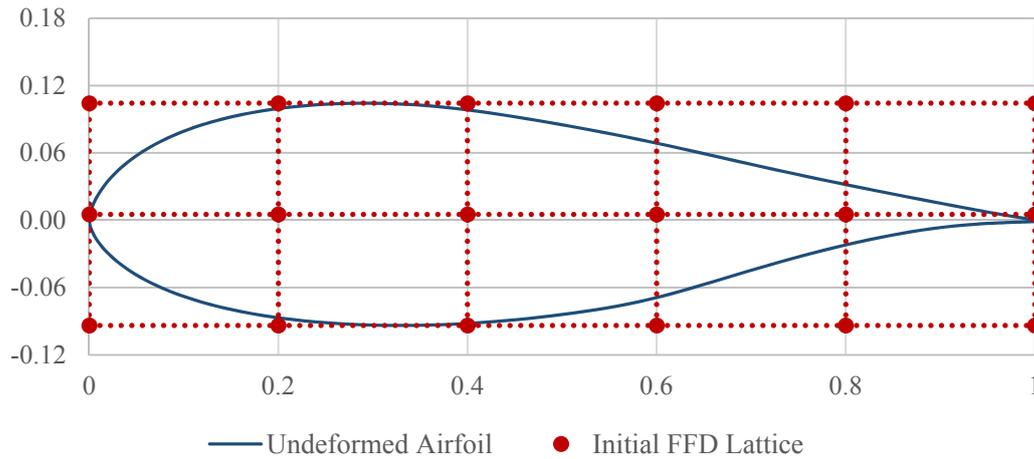


Fig. 2.5: DU-06-W-200 airfoil in a 2D FFD lattice

Step 3: Deforming the Parametric Space

In Figure 2.6 the deformation of the initial FFD lattice caused by the movement of the B-Spline control points is presented.

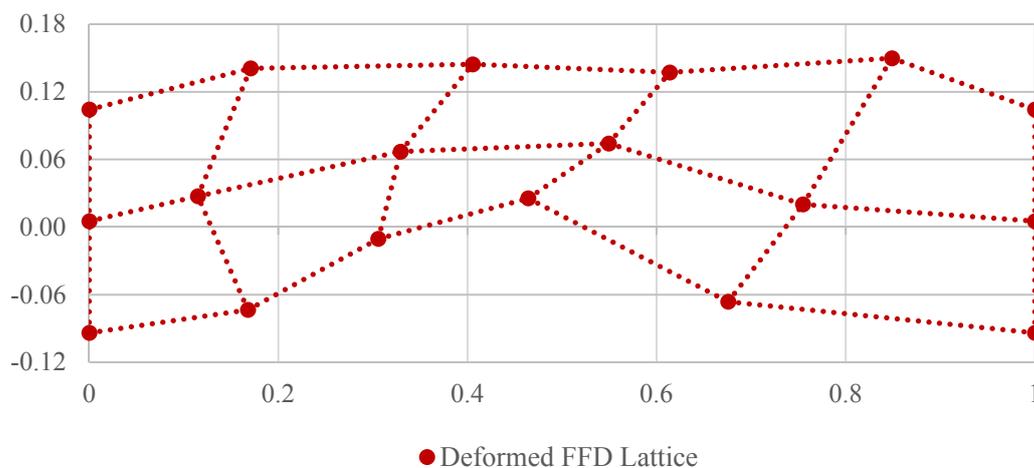
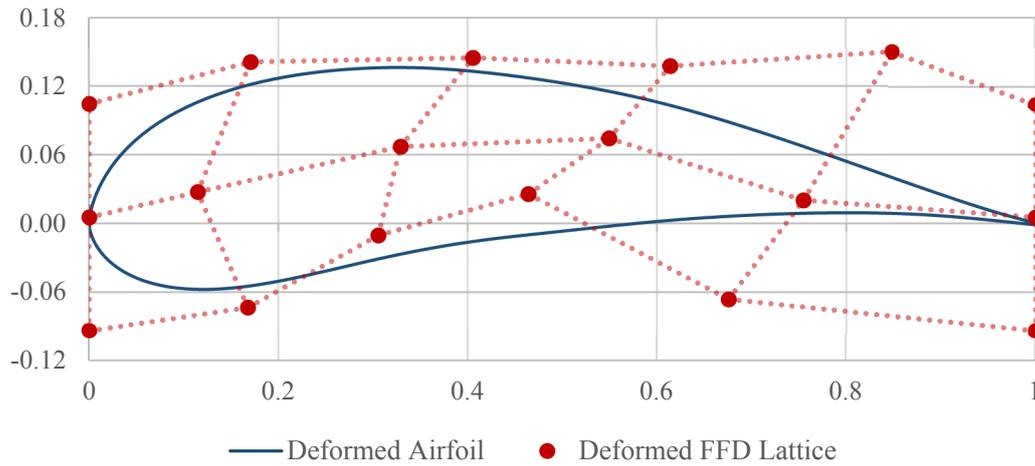


Fig. 2.6: The movement of the control points causes the deformation of the lattice

Step 4: Evaluating the Effects of the Deformation

The recovery of the deformed shape, i.e. the calculation of the Cartesian coordinates of the deformed airfoil, is achieved by importing the new Cartesian coordinates of the control points from *Step 3* into Eq. (2.8) and implementing Eq. (2.8) for each one of the airfoil's boundary points, whose parametric coordinates are known from *Step 2*.



**Fig. 2.7:** Deformed airfoil section

### 2.2.2 Cross-Sectional Area Inequality

As highlighted in Chapter 1, after the deformation of a two-dimensional shape by utilizing the FFD technique, the cross-sectional area of the deformed model is not equivalent to the cross-sectional area of the initial (reference) model. In order to prove our case, in Table 2.1, the cross-sectional area of the reference DU-06-W-200 airfoil along with the one of the deformed shape (Figure 2.6) after the application of the FFD, are compared.

Model	Cross-Sectional Area
Initial Airfoil (DU-06-W-200)	0.1223
Deformed Airfoil (Fig. 2.7)	0.1050

**Tab. 2.1:** Cross-sectional area comparison

As noted, a decrease by 14.14% occurred to the cross-sectional area of the baseline airfoil after the implementation of the FFD.

# Chapter 3

## Area-Preserving Free-Form Deformation

### 3.1 Introduction to Area-Preserving FFD

So far, the significance of preserving cross-sectional area throughout the production of the candidate geometries during an airfoil optimization procedure, along with the inability of the FFD technique to provide the particular feature, have been clearly understood. The AP FFD supplements a classic FFD, described in Chapter 2, in order to handle and satisfy a strict area preservation constraint throughout an airfoil optimization procedure. The methodology is based on the work of Hahmann et al. [48], concerning the application of a Volume-Preserving FFD after the deformation of 3D models.

The aim of the proposed AP FFD methodology is to conserve the area  $A_{ref}$  enclosed inside a given planar 2D shape  $R_{ref}$ , after the model is deformed by a classic FFD prescribed in Section 2.2. The AP FFD technique is consisted by two main parts. At first, a classical 2D FFD is applied to the model, which results to a deformed shape denoted as  $R_{def}$ . After that, the area  $A_{def}$  embedded into  $R_{def}$  usually differs, compared to the cross-sectional area  $A_{ref}$  of the reference shape  $R_{ref}$ . Subsequently an area correction step is implemented; the control points of the FFD lattice are adjusted in order to recover the baseline cross-sectional area. The adjustment to the FFD grid should be minimal in order to respect the deformation prescribed by the optimizer, given that the AP FFD is joined within a shape optimization scheme.

### 3.2 Mathematical Analysis

The objective of the following analysis is the computation of the minimum required offsets  $\theta_{ij} = (\theta_{ij}^x, \theta_{ij}^y)$ , whose application to the control points  $\mathbf{P}_{ij}$  of the deformed FFD lattice will result into a new deformed geometry (airfoil), having a value of cross-sectional area  $\bar{A}$  equal to the reference one,  $A_{ref}$ .

### 3.2.1 The Area Preservation Problem

The Area Preservation Problem (APP) that has to be solved is formed as follows:

$$\min \sum_{i=0}^n \sum_{j=0}^m \|\boldsymbol{\theta}_{ij}\|^2 \quad (3.1)$$

$$\text{subject to } \bar{A} - A_{ref} = 0 \quad (3.2)$$

The APP consists of the minimization of the squared distance  $\|\boldsymbol{\theta}_{ij}\|^2$  between the FFD control points, provided that the cross-sectional area  $\bar{A}$ , after the application of the particular offsets will be equal to the reference one. Instead of that, a minimization of the squared distance between the shape's boundary points, before and after the application of the area correction step could be also possible, however that would involve the solution of a linear system of equations. During an airfoil optimization process, an AP FFD has to be applied to each candidate geometry (airfoil), so the area preservation constraint to be meet. Therefore, the APP needs to be solved numerous thousand times, which render the extraction of a time-efficient solution to the aforementioned problem vital for the efficiency of the whole design procedure. Herein, the developed method is based on the minimization of the distance between the FFD control points, which enables the possibility of extracting a time-saving closed-form solution as an alternative to a linear system.

### 3.2.2 Cross-Sectional Area Calculation

Let  $R_{ref}$  be the initial (reference) airfoil, consisted of  $k$  points, where the  $1^{st}$  and the  $k^{th}$  points are coincidental. The area inside a planar free form closed polygon is given as:

$$A_{ref} = \frac{1}{2} \sum_{t=1}^{k-1} (y_t x_{t+1} - y_{t+1} x_t) \quad (3.3)$$

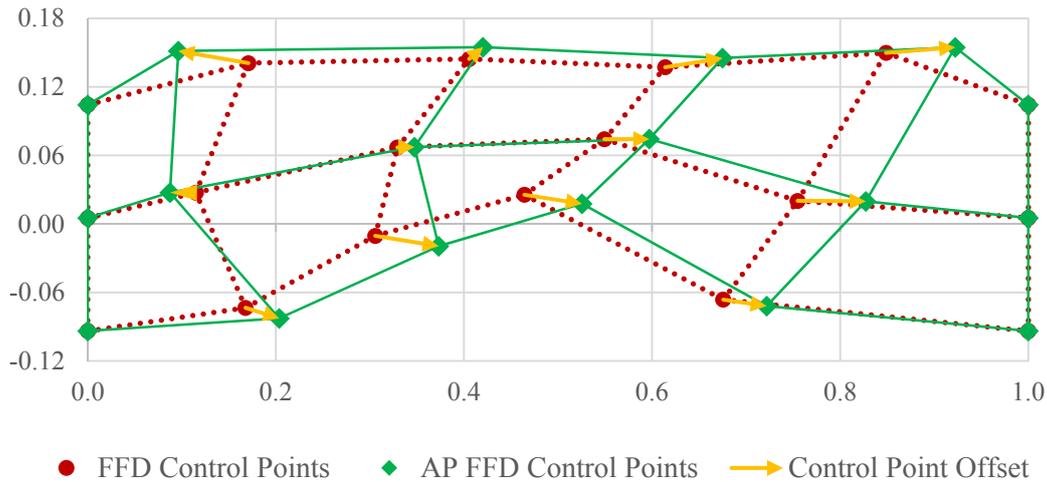
where  $(x_t, y_t)$  are the Cartesian coordinates of the  $R_{ref}$  (undeformed model). Denoting as  $(\tilde{x}_t, \tilde{y}_t)$  the Cartesian coordinates of the deformed airfoil  $R_{def}$ , the area  $A_{def}$  of the deformed object is computed by:

$$A_{def} = \frac{1}{2} \sum_{t=1}^{k-1} (\tilde{y}_t \tilde{x}_{t+1} - \tilde{y}_{t+1} \tilde{x}_t) \quad (3.4)$$

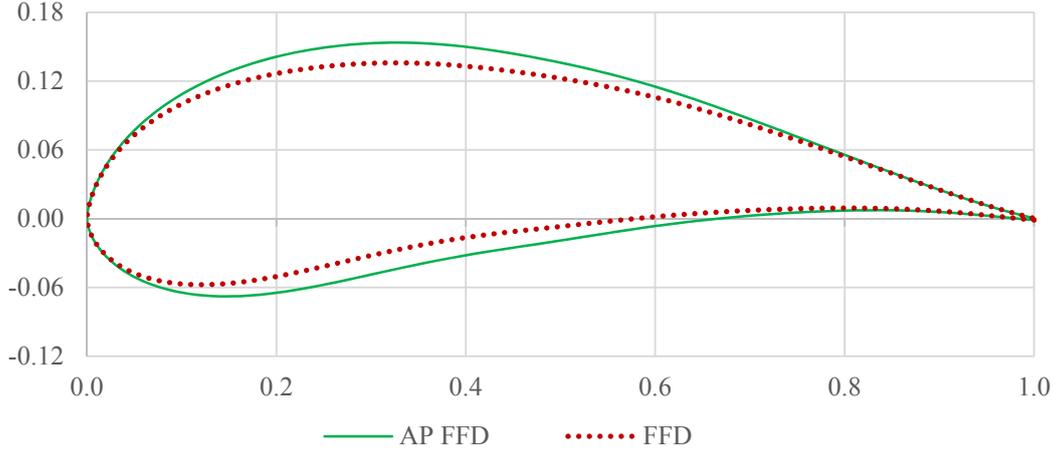
In general, after the application of a classic FFD,  $A_{ref} \neq A_{def}$  occurs as shown in Table 2.1 for the example of Section 2.2. Let  $\bar{A}$  be the area of the airfoil after the implementation of the AP FFD, after the application of the, unknown yet, offsets to the control points of the deformed lattice.

$$\bar{A} = \frac{1}{2} \sum_{t=1}^{k-1} \begin{vmatrix} (\tilde{y}_t + \theta_t^y) & (\tilde{x}_t + \theta_t^x) \\ (\tilde{y}_{t+1} + \theta_{t+1}^y) & (\tilde{x}_{t+1} + \theta_{t+1}^x) \end{vmatrix} \quad (3.5)$$

where  $\theta_t = (\theta_t^x, \theta_t^y)$  are the resulting displacements of the airfoil boundary points due to the offsets  $\theta_{ij} = (\theta_{ij}^x, \theta_{ij}^y)$  applied to the control points of the deformed lattice in order to correct the area of  $R_{def}$ . In Figure 3.1, the application of the displacements  $\theta_{ij} = (\theta_{ij}^x, \theta_{ij}^y)$  to every control point  $(i, j)$  of the deformed lattice along with the new AP FFD lattice formation are presented, while in Figure 3.2, a comparison between the deformed by a classic FFD airfoil (red dashed) and the new airfoil geometry (green line), after the application of the offsets, with cross-sectional area equal to the reference airfoil, is illustrated. As observed, no displacement has been applied to the control points located at the extreme left and extreme right (leading and trailing edges) positions. This is to maintain the chord length unchanged. The capability of the methodology to provide the particular feature will be demonstrated later through the analysis.



**Fig. 3.1:** The positions of the control points and the lattice formation after the application of AP FFD



**Fig. 3.2:** The resulting airfoil (green) after the application of the AP FFD

The equations connecting the offsets applied to the control points  $\theta_{ij} = (\theta_{ij}^x, \theta_{ij}^y)$  and the displacements of the model's points  $(\theta_t^x, \theta_t^y)$  due to  $\theta_{ij}$ , are extracted based on the B-Spline theory:

$$\theta_t^x = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u_t) N_{j,q}(v_t) \theta_{ij}^x \quad (3.6)$$

$$\theta_t^y = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u_t) N_{j,q}(v_t) \theta_{ij}^y \quad (3.7)$$

where  $(u_t, v_t)$  are the parametric coordinates of the  $t^{\text{th}}$  boundary point of the airfoil.

### 3.2.3 The Area Constraint

By inserting Eq. (3.6) and Eq. (3.7) into Eq. (3.5) and by substituting Eq. (3.5) into Eq. (3.2) the area constraint  $\bar{A} - A_{ref} = 0$  is formulated as

$$A_{def} + \sum_{i=0}^n \sum_{j=0}^m \theta_{ij}^x \alpha_{ij} + \sum_{i=0}^n \sum_{j=0}^m \theta_{ij}^y b_{ij} - A_{ref} = 0 \quad (3.8)$$

where:

$$\alpha_{ij} = \frac{1}{2} \sum_{t=1}^{k-1} [\tilde{y}_t N_{i,p}(u_{t+1}) N_{j,q}(v_{t+1}) - \tilde{y}_{t+1} N_{i,p}(u_t) N_{j,q}(v_t)] \quad (3.9)$$

$$b_{ij} = \frac{1}{2} \sum_{t=1}^{k-1} [\tilde{x}_{t+1} N_{i,p}(u_t) N_{j,q}(v_t) - \tilde{x}_t N_{i,p}(u_{t+1}) N_{j,q}(v_{t+1})] \quad (3.10)$$

Hence, the area constraint of the APP is a bilinear function of the offsets  $\theta_{ij}^x, \theta_{ij}^y$ . Consequently in order to extract a solution to APP, a nonlinear minimization problem has to be solved, composed by a quadratic objective function (Eq. (3.1)) and a linear area constraint (Eq. (3.2) or Eq. (3.8)). In this work, instead of trying to solve the particular nonlinear problem, which could be proved a quite time-consuming process, a different approach is utilized. As long as Eq. (3.8) is a bilinear function of the unknowns  $\theta_{ij}^x, \theta_{ij}^y$ , a separation to  $x$  and  $y$  axis is proposed; this separation leads to the generation of two Area Correction Sub Problems, to be solved successively. Each one of them consists of the computation of the minimum displacements according to one axis at a time, subjected to a linear local area constraint, by setting the displacement in the other axis equal to zero,  $\theta_{ij} = (\theta_{ij}^x, 0)$  for the computation of the  $\theta_{ij}^x$  and  $\theta_{ij} = (0, \theta_{ij}^y)$  for the computation of the  $\theta_{ij}^y$ . Thus, separating the area correcting deformation according to the axes makes it linear in each axis, which permit the extraction of a closed-form solution by using Lagrange multipliers method. The combination of the solutions extracted by the two sub problems is totally equal to the solution of the APP.

### 3.2.4 The Area Correction Sub Problems

Letting  $w_x = w_y = 0.5(A_{ref} - A_{def})$ , the two Area Correction sub problems are formed as follows.

*Area Correction Sub Problem in x-axis (SPx):*

$$\min \sum_{i=0}^n \sum_{j=0}^m (\theta_{ij}^x)^2 \quad (3.11)$$

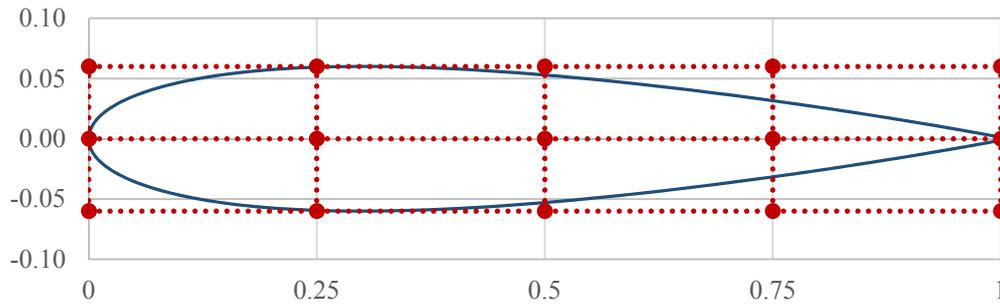
$$\text{subject to } \sum_{i=0}^n \sum_{j=0}^m \theta_{ij}^x \alpha_{ij} = w_x \quad (3.12)$$

*Area Correction Sub Problem in y-axis (SPy):*

$$\min \sum_{i=0}^n \sum_{j=0}^m (\theta_{ij}^y)^2 \quad (3.13)$$

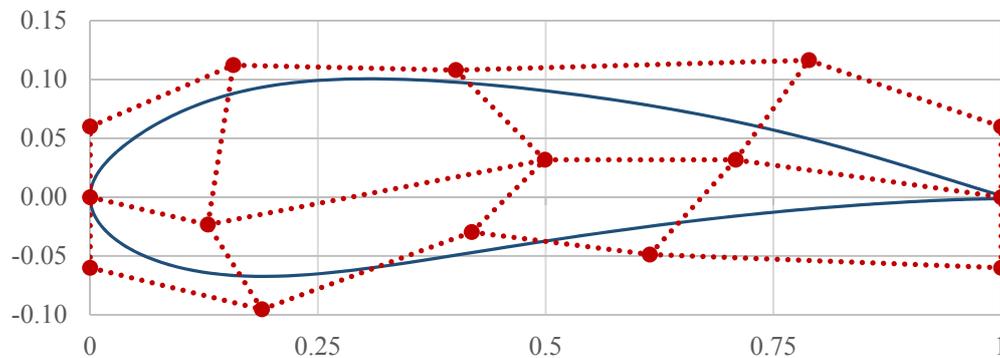
$$\text{subject to } \sum_{i=0}^n \sum_{j=0}^m \theta_{ij}^y \alpha_{ij} = w_y \quad (3.14)$$

Note that the combination of the two local area constraints (3.12), (3.14) is equal with the utilization of the area constraint (3.8) of the initial APP. Additionally, the separation of the main area preservation problem to the two axes provides extra advantages; by modulating the coefficients  $w_x$  and  $w_y$ , the user is able to define the percentage of lost or gained area to be corrected as well as the percentage of the contribution of  $\theta_{ij}^x$  and  $\theta_{ij}^y$  to the magnitude of the displacement vector  $\theta_{ij}$ . For example, by setting  $w_x = 0.3(A_{ref} - A_{def})$  and  $w_y = 0.7(A_{ref} - A_{def})$ , the total area of the initial model is preserved ( $0.3 + 0.7 = 1$ ), while the displacement of each control point in  $x$  direction will be a 30% fraction of the total displacement. Furthermore, in many applications concerning the handle and deformation of airfoil sections, a movement of the control points only in the normal to chord ( $y$ ) direction is desirable. In cases like that the AP FFD can be customized by setting  $w_x = 0$  and  $w_y = (A_{ref} - A_{def})$  and solving only the SPy problem, so that the extracted solution to be formed as  $\theta_{ij} = (0, \theta_{ij}^y)$ . Additionally, there is an option of setting the displacements of specific control points equal to zero. This feature finds an application in airfoil optimization as it applies to the control points that affect the positions of leading and trailing edges. Due to the necessity of keeping those positions fixed, in order to keep the chord length unchanged, the control points that affect the pre-mentioned positions have to be fixed, at least at the chord-wise direction. The implementation of the AP FFD with fixed control points is achieved by solving the same sub problems, while setting  $\theta_{rs}^x, \theta_{rs}^y, \alpha_{rs}, b_{rs}$  equal to zero for the control points and direction(s) of interest. In Figures 3.3 – 3.6, the application of an AP FFD to a NACA 0012, as reference airfoil, only in  $y$  direction is presented, while the control points located in positions  $x = 0$  and  $x = 1$  are set fixed. In Figure 3.3, a NACA 0012 airfoil inside a 2D B-Spline FFD lattice consisted of 5 control points in the chordwise direction ( $x$  direction) and 3 control points in the normal to chord direction ( $y$  direction), is illustrated. The degree of the basis function in the chordwise direction ( $u$  parametric direction) is 4 while the degree of the basis function in the normal to chord direction ( $v$  parametric direction) is 2.



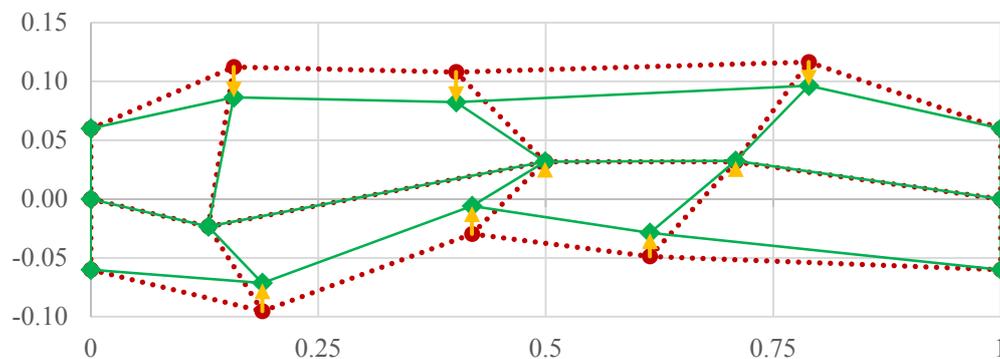
**Fig. 3.3:** NACA 0012 airfoil inside an initial FFD lattice

In Figure 3.4, the deformation of the parametric space (lattice) caused by the movement of the control points, dictates an indirect deformation to the initial airfoil (FFD). The cross-sectional area of the resulting airfoil is 26.8% greater than the reference one.

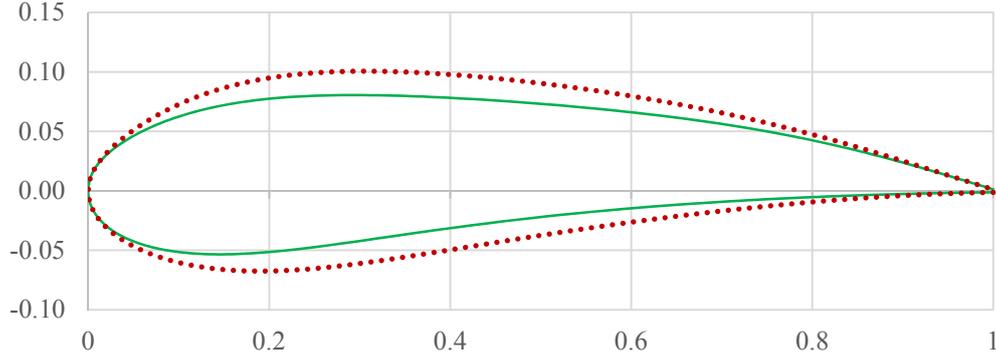


**Fig. 3.4:** The resulting airfoil after the application of a classic FFD

In Figure 3.5 the application of the AP FFD only in y direction is illustrated. The displacement of the control points is permitted only in the normal to the chord direction. Finally, in Figure 3.6 the airfoil geometry (green) after the application of the AP FFD is compared with the one of Figure 3.4, which emerged by the implementation of a classic FFD.



**Fig. 3.5:** The positions of the control points and the lattice formation after the application of AP FFD, only in y direction



**Fig. 3.6:** The resulting airfoil (green) after the application of the AP FFD, only in  $y$  direction

### 3.2.5 Closed-form Solution

By separating the main APP into SPx and SPy a Lagrange Multipliers method can be applied to each one of them to extract a closed-form solution. The Lagrangian function for the SPx is formed as:

$$\Lambda(\theta_{ij}^x, \lambda_x) = \sum_{i=0}^n \sum_{j=0}^m (\theta_{ij}^x)^2 + \lambda_x \left( \sum_{i=0}^n \sum_{j=0}^m \theta_{ij}^x \alpha_{ij} - w_x \right) \quad (3.15)$$

The partial derivatives of the Lagrangian function are formulated below:

$$\frac{\partial \Lambda}{\partial \lambda_x} = 0 \Leftrightarrow \sum_{r=0}^n \sum_{s=0}^m \theta_{rs}^x \alpha_{rs} = w_x \quad (3.16)$$

$$\frac{\partial \Lambda}{\partial \theta_{ij}^x} = 0 \Leftrightarrow 2\theta_{ij}^x + \lambda_x \alpha_{ij} = 0 \Leftrightarrow \theta_{ij}^x = -\frac{\lambda_x}{2} \alpha_{ij} \quad (3.17)$$

Substituting (3.17) into (3.16) results in:

$$\lambda_x = -\frac{2w_x}{\sum_{r=0}^n \sum_{s=0}^m (\alpha_{rs})^2} \quad (3.18)$$

Therefore, by combining (3.17) and (3.18), the solution of the SPx problem results as:

$$\theta_{ij}^x = \frac{\alpha_{ij} w_x}{\sum_{r=0}^n \sum_{s=0}^m (\alpha_{rs})^2} \quad (3.19)$$

By applying the same methodology, a closed-form solution results for the SPy as well. Note that the two sub problems have to be solved and applied successively, but not in a

particular order. In this case, the computation of  $b_{ij}$  has to be performed after the application of displacements  $\theta_{ij}^x$  to the control points, due to the dependency of  $b_{ij}$  to the Cartesian coordinates of the deformed model. Thereafter the solution of the SPy is given as:

$$\theta_{ij}^y = \frac{b_{ij}w_y}{\sum_{r=0}^n \sum_{s=0}^m (b_{rs})^2} \quad (3.20)$$

**“Intentionally Blank”**

# **Chapter 4**

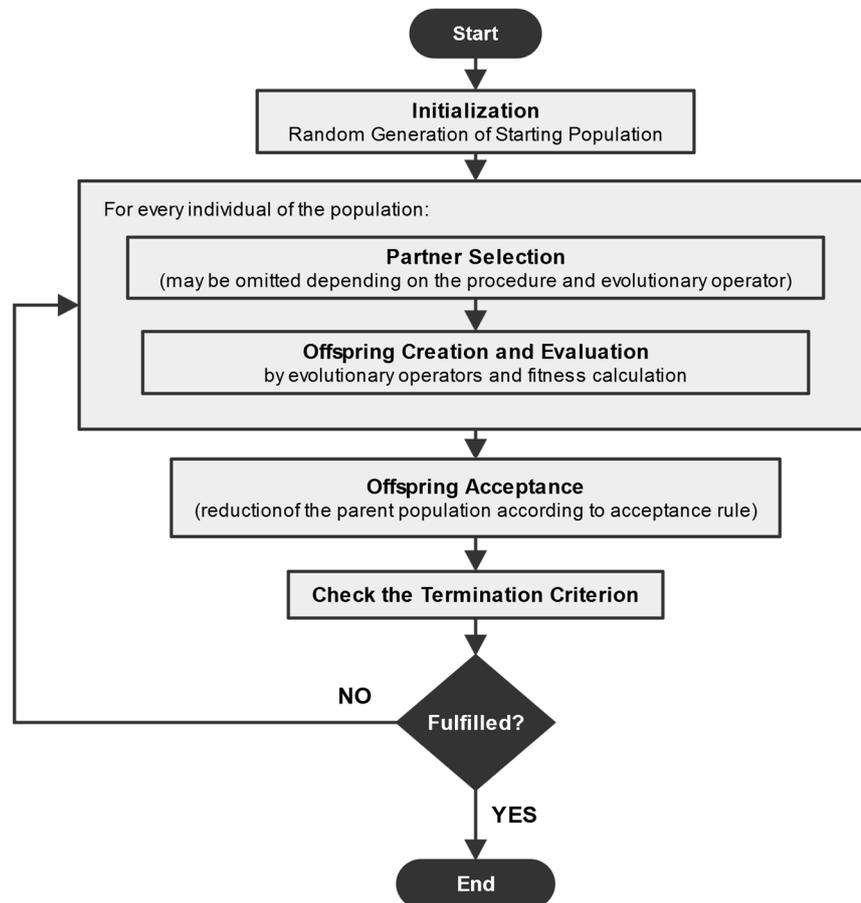
## **Differential Evolution Algorithm and Optimization Scheme**

### **4.1 Introduction to Evolutionary Algorithms**

Evolutionary Algorithms (EAs) is a general term used to indicate any metaheuristic optimization algorithm, based on search methods and populations. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection. Candidate solutions to the optimization problem play the role of individuals in a population, and the fitness function determines the quality of the solutions. Contrary to many other optimization methods, EAs work with a set of solutions at the same time, the so-called population. Accordingly, they frequently supply not only one solution when treating multi-modal problems, but also several different solutions, the quality of which is partly comparable with that of the best. Evolution of the population then takes place after the repeated application of the above operators [57].

EAs are a class of search methods with remarkable balance between exploitation of the best solutions and exploration of the search space. They combine elements of directed and stochastic search and, therefore, are more robust than directed search methods. Additionally, they may be tailored easily to the specific problem under consideration, taking into account its special characteristics [58]. The natural selection process is simulated in EAs, using a population of individuals to evolve through certain procedures. Each individual in the population is represented through its chromosome - a string of numbers (bit strings, integers or floating point numbers), in a similar way to chromosomes in nature; it contains the encoding of the design variables of the optimization problem. Each individual's quality is represented by a fitness function, which depends on the specific optimization problem under consideration. During the past decade, evolutionary algorithms have been established in the field of engineering design as a very versatile, effective and robust optimization technique, capable of

dealing with complicated and very demanding real-world single/multi-disciplinary, single/multi-objective, discontinuous, non-differentiable, multi-modal design optimization problems [58]. In Figure 4.1 the principle flowchart of Evolutionary Algorithms is illustrated.



*Fig. 4.1: Principle flowchart of an Evolutionary process [59]*

## 4.2 Differential Evolution

Differential evolution (DE), introduced in the work of Storn and Price [60, 61], is a reliable, versatile and fast EA, which has demonstrated better convergence performance than other EAs. The DE algorithm is basically a type of Evolutionary Strategy, with special characteristics, so that it can effectively deal with continuous optimization problems, which are common in engineering design. The standard DE algorithm uses a fixed size population of  $N_p$  chromosomes, which is randomly initialized; an iterative process then is established and at each generation  $G$ , a new population is produced. At each generation, each element of the population can be replaced with a new generated one. The new element (the trial vector) is a linear

combination between a randomly selected element and a scaled difference between two other randomly selected elements. In the selection stage, the trial vector competes against the population vector of the same index, in a one-by-one competition, and the survivors of the  $N_p$  competitions become the parents for the next generation in the evolutionary cycle [61].

Herein, a description of the basic elements constitute the DE algorithm follows as introduced in [60, 61] and presented in [58]. Given a cost function

$$f(\mathbf{X}): \mathbb{R}^n \rightarrow \mathbb{R} \quad (4.1)$$

where  $\mathbf{X}$  is a vector of  $n$ -dimensional space, containing the  $n$  design variables of the problem under consideration. The objective is the minimization of the cost function  $f$  by modulating the values of the design variables that compose  $\mathbf{X}$ .

$$\mathbf{X} = (x_1, x_2, \dots, x_n) , x_i \in \mathbb{R} \quad (4.2)$$

Throughout the optimization procedure, each design variable is bounded between a lower  $x_i^L$  and an upper  $x_i^U$  value, which are selected by the designer and determine the range of each design variable.

$$x_i^L \leq x_i \leq x_i^U , i = 1, 2, \dots, n \quad (4.3)$$

The  $N_p$  members of the initial population are randomly generated in the space defined by the lower and upper limits of the design variables

$$x_{k,i}^0 = r (x_i^U - x_i^L) + x_i^L , i = 1, 2, \dots, n , k = 1, 2, \dots, N_p \quad (4.4)$$

where  $r$  is a uniformly distributed random value within  $[0, 1]$ .

The mutation operator of the DE (differential mutation) is based on a triplet of randomly selected different members of the current population. A new individual is generated by adding the weighted difference vector between the two members of the triplet to the third one (the "donor"). This perturbed individual and the initial population member is then subjected to a crossover operation, which produces the final candidate solution.

$$x'_{k,i}{}^{G+1} = \begin{cases} x_{C_{k,i}}^G + F(x_{A_{k,i}}^G - x_{B_{k,i}}^G) & \text{if } (r \leq Cr \text{ or } i = i^*) \\ x_{k,i}^G & \text{otherwise} \end{cases} \quad (4.5)$$

In the previous equation  $x_{C_k,i}^G$  are the elements of the "donor" vector,  $G$  is the current generation and

$$\begin{aligned}
k &= 1, 2, \dots, N_p, \quad i = 1, 2, \dots, n \\
A_k &\in [1, \dots, N_p], \quad B_k \in [1, \dots, N_p], \quad C_k \in [1, \dots, N_p] \\
A_k &\neq B_k \neq C_k \neq k \\
Cr &\in [0, 1], \quad F \in [0, 1], \quad r \in [0, 1]
\end{aligned} \tag{4.6}$$

while  $i^*$  is a random integer within  $[1, n]$ , chosen once for all members of the population. The random number  $r$  is seeded for every gene of each chromosome.  $F$  and  $Cr$  are DE control parameters for mutation and crossover operations, which (for the standard DE algorithm) remain constant during the search process and affect the convergence behavior and robustness of the algorithm. Their values also depend on the objective function, the characteristics of the problem and the population size.  $Cr$  controls the fraction of design variables that are copied from the mutant. In addition, the trial design variable with randomly chosen index,  $i^*$ , is taken from the mutant in order to ensure that the trial vector does not duplicate the initial one.

The population for the next generation is selected between the current population and the final candidates by using a one-by-one comparison between all members of the current population and their candidate replacements. If each candidate vector is better fitted than the corresponding current one, the new vector replaces the vector with which it was compared. The DE selection scheme is described as follows [61] (for a minimization problem):

$$\mathbf{X}_k^{G+1} = \begin{cases} \mathbf{X}_k^{G+1} & \text{if } f(\mathbf{X}_k^{G+1}) \leq f(\mathbf{X}_k^G) \\ \mathbf{X}_k^G & \text{otherwise} \end{cases} \tag{4.7}$$

### 4.3 Surrogate models

Most engineering design problems require experiments and/or simulations to evaluate design objective and constraint functions as function of design variables. For example, in order to find the optimal airfoil shape for an aircraft wing, an engineer simulates the air flow around the wing for different shape variables. For many real world problems, however, a single simulation can take many minutes, hours, or even days to complete. As a result, routine tasks such as design optimization, design space

exploration, sensitivity analysis and what-if analysis become impossible since they require thousands or even millions of simulation evaluations. One way of alleviating this burden is by constructing approximation models, known as surrogate models that mimic the behavior of the simulation model as closely as possible while being computationally cheap(er) to evaluate. Surrogate models are constructed using a data-driven, bottom-up approach. The exact, inner working of the simulation code is not assumed to be known (or even understood), solely the input-output behavior is important. A model is constructed based on modeling the response of the simulator to a limited number of intelligently chosen data points [62–64].

#### **4.4 A surrogate-assisted DE algorithm**

In this thesis, a DE algorithm, presented in Section 4.2, is combined with two ANNs (Artificial Neural Networks), which serve as surrogate models, in order to reduce the number of the time-consuming exact evaluations required per generation during a direct airfoil optimization procedure. Although EAs have been combined with a various types of surrogate models [62,65–68], based on the results of [4] a Multilayer Perceptron (MLP) and a Radial-Basis Function ANNs are chosen as surrogates (Figure 4.2 and 4.3). The multiplicity of different surrogates is a crucial fact, concerning that the effectiveness of each surrogate varies, depending on the nature of the optimization model. Therefore in each generation the surrogate with the best performance among the others takes action.

##### **4.4.1 Artificial Neural Networks**

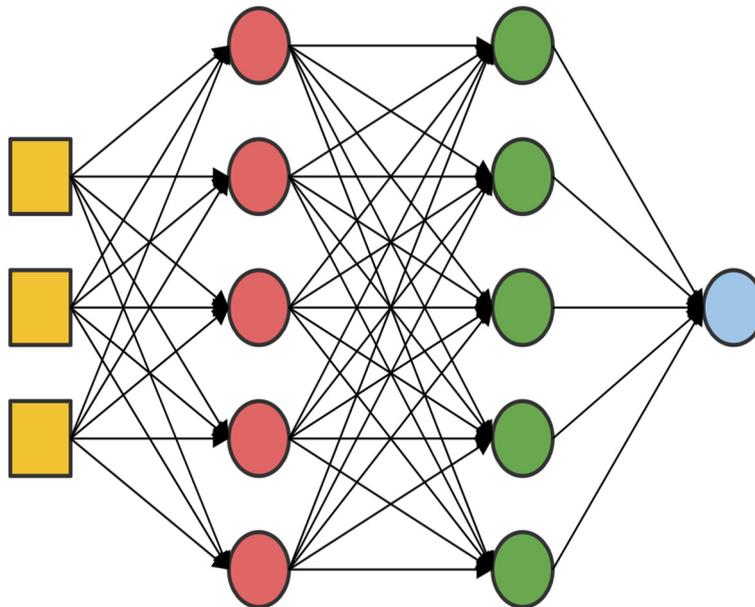
Artificial neural networks are a family of statistical learning models inspired by biological neural networks and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. An ANN consists of a pool of simple processing units, which communicate by sending signals to each other over a large number of weighted connections [69, 70].

Each unit performs a relative simple job: receive input from neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherently parallel in the sense that many units can carry out their computations at the same time Artificial neural networks can be most adequately characterized as "computational models" with particular properties such as the ability to adapt or learn,

to generalize, or to cluster or organize data, and which operation is based on parallel processing [69].

#### 4.4.2 Multi-Layer Perceptron ANN

A feed-forward network has a layered structure and the data flow from input to output units and no feedback connections are present. Each layer consists of units, which receive their input from units a layer directly below and send their output to units in a layer directly above the unit. There are no connections within a layer. With  $i, h$  and  $o$ , input, hidden and output units are denoted respectively;  $x_{kp}$  is the  $k^{th}$  element of the  $p^{th}$  input pattern vector;  $y_{kp}$  is the activation value of the network when input pattern vector  $p$  was fed into the network;  $d_{kp}$  is the  $k^{th}$  element of the desired output of the network when input pattern  $p$  was fed into the network;  $w_{ij}$  are the weights of the connection from unit  $j$  to unit  $k$ . The  $N_i$  inputs are fed into the first layer of  $N_{h,1}$  hidden units. These neural networks are commonly refer to as *multilayer perceptrons*. No processing takes place inside the input units. The activation of a hidden unit is a function  $F_k$  of the weighted inputs, plus a bias  $\theta_k$ . The output of the hidden units is distributed over the next layer of  $N_{h,2}$  hidden units, until the last layer of hidden units, of where the outputs are fed into a layer of  $N_o$  output units [69]. The architectural layout of a multilayer perceptron (MLP) is illustrated in Figure 4.2.



**Fig. 4.2:** A multilayer perceptron (MLP) artificial neural network with two hidden layers and a single output

The network is a fully connected one, which means that a node in any layer of the network is connected to all the nodes in the previous layer. Each node in the network includes a nonlinear activation function  $F(s_{kp})$  of the total input  $s_{kp}$ , which in this work is the logistic function:

$$y_k^p = F(s_{kp}) = \frac{1}{1 + \exp(-s_k^p)} \quad (4.8)$$

The total input to unit  $k$  is simply the weighted sum of the separate outputs from each of the connected units plus the bias:

$$s_k^p = \sum_j w_{jk}^p y_j^p + \theta_k^p \quad (4.9)$$

The synaptic weights are determined in the supervised training procedure, through successive weight adaptations, using the back-propagation algorithm or other more efficient algorithms. In order to train the network, a set of input vectors and the associated output vectors (training examples) are needed (learning samples). These are “presented” to the network in successive epochs and in randomized order from epoch to epoch. In our case the training examples consist of chromosome vectors for the input layer and the corresponding objective functions for the single node of the output layer. Test samples (other than the learning ones) are used in order to test the validity of the approximation and the generalization ability of the network [58].

#### 4.4.3 Radial Basis Function ANN

A RBF network (Figure 4.3) is a three layer, fully connected feed-forward ANN performing a non-linear mapping from the input space to the hidden space, followed by a linear mapping from the hidden to the output space ( $L$  being the number of input nodes,  $M$  being the number of hidden nodes, while the output layer has a single node). For an input vector  $\mathbf{xx} = [xx_1, xx_2, \dots, xx_L]$  the corresponding output vector  $yy(\mathbf{xx})$  is given as

$$yy(\mathbf{xx}) = \sum_{i=1}^M w_i \varphi_i(\mathbf{xx}) \quad (4.10)$$

$\varphi_i(\mathbf{xx})$  being the output of the  $i^{th}$  hidden unit:

$$\varphi_i(\mathbf{xx}) = F(\|\mathbf{xx} - cc_i\|), \quad i = 1, \dots, M \quad (4.11)$$

The connections (weights) to the output unit ( $w_i, i = 1, \dots, M$ ) are the only adjustable parameters. The non-linear activation function  $F$  in our case is the Gaussian RBF. The RBF centers in the hidden units  $cc_i, i = 1, \dots, M$  are selected in a way to maximize the generalization properties of the network. The standard process is to select the input vectors in the training set as RBF centers, which may lead to bad generalization of the network. The proposed solution [70] is the selection of  $M < NR$  in order to provide a better generalizing capability. Direct learning is used in this work, based on a matrix formulation of the governing equations of RBF network. The presentation of the network with the  $NR$  input patterns allows the formulation of a  $(NR \times M)$  matrix  $H$ . The output unit values result in the form of the matrix product:

$$H(NR \times M)w(M \times 1) = yy(NR \times 1) \tag{4.12}$$

where  $yy$  is the desired output vector provided by the training data set,  $w$  being the synaptic weights vector consisting of  $M$  unknowns.  $H$  is inverted through the Gram-Schmidt technique. It should be mentioned that the computational cost of the neural networks is the cost of their training, while the use of an already trained network to evaluate a new candidate adds negligible computation cost to the optimization procedure.

In this work the number of hidden units in each one of the two hidden layers of the MLP network was twice the number of the design variables. For the RBF networks the number of centers ( $M$ ) was set equal to 2/3 of the number of training data ( $NR$ ).

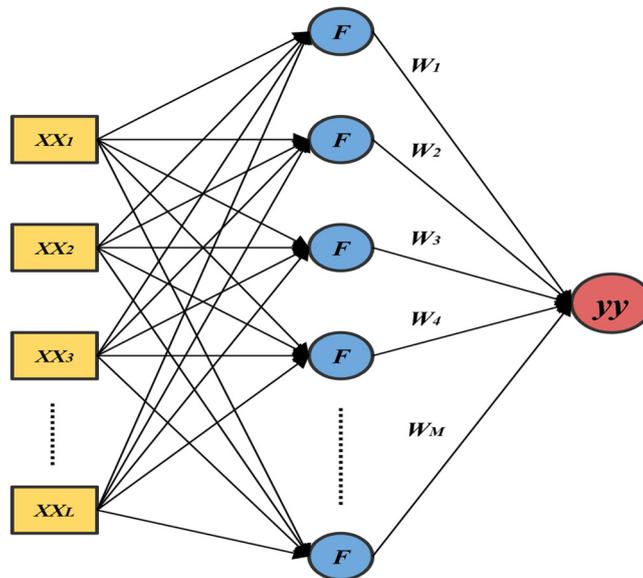


Fig. 4.3: A RBF ANN

**4.4.4 The use of surrogates for accelerating DE algorithm**

Through the evaluation stage, each new trial vector is pre-evaluated by the utilization of the surrogate models. If the trial vector is pre-evaluated as lower-fitted (i.e., having a higher cost function in minimization problems) than the corresponding vector of the current population, then no further exact evaluation is performed; the trial vector is abandoned and the current vector is transferred to the next generation. Otherwise an exact re-evaluation is performed, followed by a new comparison between the two vectors. If the trial vector is still better-fitted then it passes to the next generation. Otherwise the trial vector is abandoned and the current vector will pass to the next generation [58].

In the first two generations all individuals are exactly evaluated, in order to create a starting database for the construction of the initial surrogate models (training and testing procedures). Furthermore, an additional small percentage of the candidate solutions are selected in each generation with uniform probability to be exactly evaluated, in order to enhance the robustness of the procedure. The prescribed procedure results in the fact that only exactly evaluated trial vectors have the opportunity to pass to the new generation. Consequently, in every generation the current population always includes exactly evaluated individuals. A flowchart of the described procedure is illustrated in Figure 4.4.

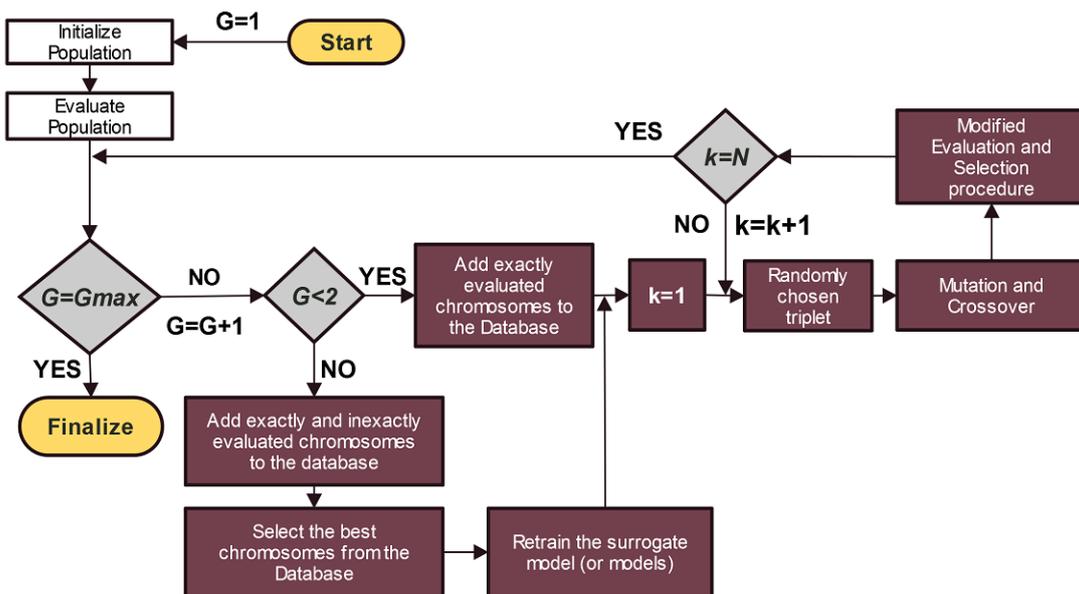


Fig. 4.4: Flowchart of the modified DE algorithm [58]

The different surrogate models can be used separately or as an ensemble. In the first case a single surrogate is used throughout the optimization procedure, while in the

second case in each generation both surrogates are retrained and tested (on the same training and testing data sets) and then one is used for the pre-evaluation of the members of the population. The selected model to be used as surrogate is the one attaining the minimum value concerning the testing error. Compared to the cost of the exact evaluations, the training of more than one surrogate models in each generation adds negligible cost to the computational procedure. During the evolution of the population the best surrogate for the specific region of the landscape is automatically selected, based on their predictive capabilities and the adopted selection criterion [4].

#### **4.5 The Optimization Procedure**

The optimization process begins with the selection of a reference airfoil achieving satisfying performance concerning some characteristics of interest, such as high lift coefficient, high lift-to-drag ratio, low absolute value of pitching moment coefficient and smooth attribute to stall regions. Then, a 2D parametric FFD lattice is constructed, that encloses the initial airfoil. The user has the ability of choosing the number of control points on each parametric direction, which has an impact on the number of design variables, as well as the degree of the B-Splines Basis functions that form the FFD lattice. Subsequently, the determination of the design variables as well as their lower and upper bounds are defined by the user.

At the first generation ( $G=1$ ) of the DE algorithm, every chromosome of the fixed size population  $N_p$  is randomly initialized, based on upper and lower bounds of the design variables (genes). Subsequently, at each generation a new population is produced by replacing (after competition) each member of the population with a new one, called trial vector. The trial vector is formed as a linear combination of a randomly chosen chromosome of the current population and a scaled difference between two other also randomly chosen chromosomes. Then, a one by one competition between the trial vector and its corresponding chromosome of the current generation is implemented; this is repeated for all the members of the current population in order to extract the  $N_p$  chromosomes that will form the next generation [61]. The assistance of the surrogate models lies on time-saving, by not exact evaluating each one of the produced trial vectors, but using a trained neural network instead [71,72]. A detailed description of the utilized DE algorithm as well as the combination with the surrogate models in the airfoil optimization is presented in [4].

Given that a chromosome has been chosen to be exact evaluated, an area correction step has to be implemented. After the implementation of the Area-Preserving FFD on the specific chromosome, presented in Chapter 3, an accepted airfoil geometry is produced, which has the exact same cross-sectional area as the reference airfoil. Then, the pre-mentioned airfoil is imported to XFOIL (or other evaluation software) to obtain the desired data for the calculation of the fitness function. The aforementioned function is formed based on the required characteristics of the optimal airfoil, combined with penalty functions to satisfy the required constraints. The previously described procedure is illustrated as a flow chart in Figure 4.5.

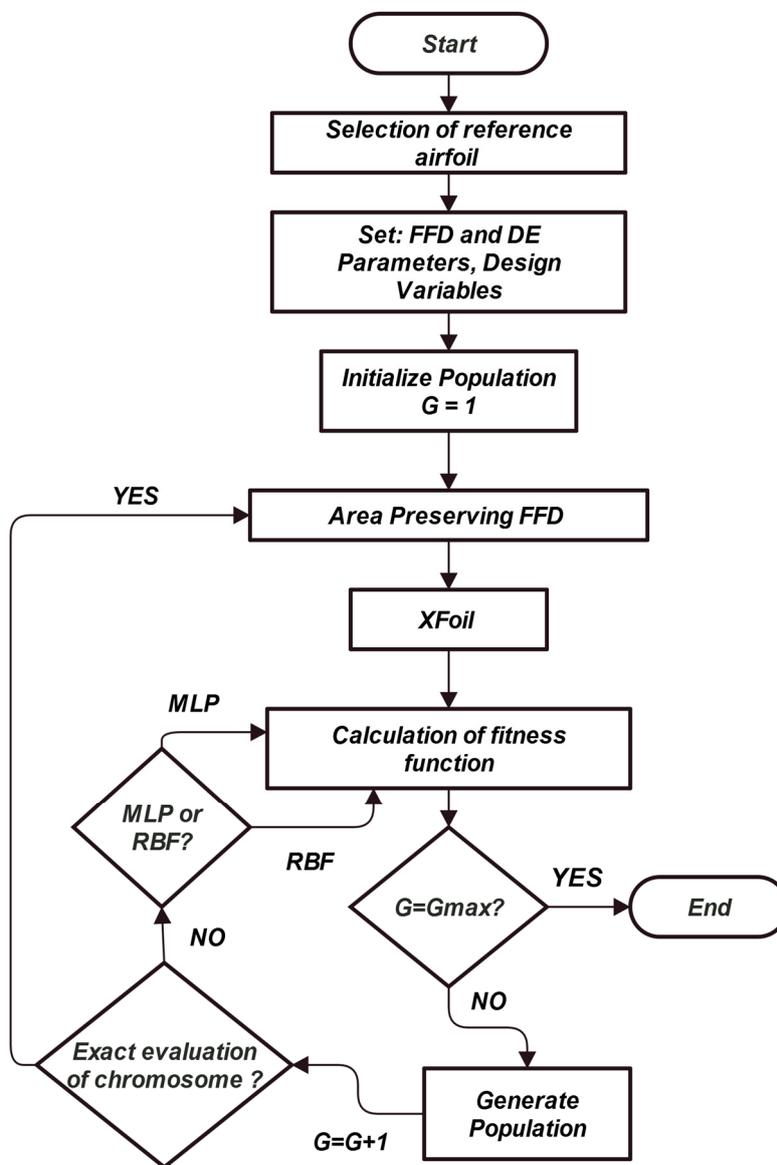


Fig. 4.5: Flowchart with the major steps of the optimization process including AP FFD

## 4.6 The Evaluation Software – XFOIL

XFOIL is an interactive program for the design and analysis of subsonic isolated airfoils. Given the coordinates specifying the shape of a 2D airfoil, Reynolds and Mach numbers, XFOIL can calculate the pressure distribution on the airfoil and hence lift and drag characteristics. The XFOIL software, developed by Drela in the ‘80s [73], is used to calculate the lift, drag and pitching moment coefficients ( $C_l, C_d, C_m$ ) of each exact-evaluated airfoil in a variety of angles of attack ( $\alpha$ ). In this thesis the Version 6.99 of XFOIL, released in 2013, was utilized. In Figure 4.6 a sample XFOIL output file is presented, where “alpha” is the angle of attack, “CL” is the lift coefficient ( $C_l$ ), “CD” is the drag coefficient ( $C_d$ ) and “CM” is the pitching moment coefficient ( $C_m$ ).

```

XFOIL          Version 6.99

calculated polar for: airfoil

1 1 Reynolds number fixed      Mach number fixed

xtrf = 1.000 (top)      1.000 (bottom)
Mach = 0.000      Re = 7.000 e 6      Ncrit = 9.000

  alpha    CL      CD      CDp      CM      Top_Xtr  Bot_Xtr
-----
0.000    0.0855   0.00581  0.00126 -0.0200  0.1863  0.5507
1.000    0.2057   0.00595  0.00138 -0.0218  0.1481  0.6000
2.000    0.3262   0.00621  0.00157 -0.0235  0.1284  0.6234
3.000    0.4461   0.00657  0.00183 -0.0250  0.1128  0.6311
4.000    0.5653   0.00692  0.00212 -0.0265  0.1020  0.6413
5.000    0.6833   0.00736  0.00248 -0.0278  0.0916  0.6468
6.000    0.7998   0.00787  0.00290 -0.0289  0.0818  0.6509
7.000    0.9154   0.00836  0.00336 -0.0298  0.0772  0.6587
8.000    1.0292   0.00893  0.00389 -0.0305  0.0715  0.6628
9.000    1.1401   0.00962  0.00451 -0.0308  0.0662  0.6648
10.000   1.2494   0.01029  0.00518 -0.0309  0.0627  0.6702
11.000   1.3531   0.01120  0.00605 -0.0302  0.0565  0.6770
12.000   1.4548   0.01204  0.00690 -0.0293  0.0548  0.6806
13.000   1.5418   0.01320  0.00803 -0.0261  0.0509  0.6821
14.000   1.6231   0.01462  0.00949 -0.0227  0.0485  0.6864
15.000   1.7004   0.01646  0.01138 -0.0197  0.0467  0.6926
16.000   1.7681   0.01908  0.01404 -0.0164  0.0429  0.6968
17.000   1.8356   0.02188  0.01690 -0.0138  0.0418  0.6990
18.000   1.8877   0.02579  0.02085 -0.0106  0.0384  0.7009
19.000   1.9374   0.02996  0.02513 -0.0078  0.0366  0.7085
20.000   1.9749   0.03527  0.03054 -0.0051  0.0344  0.7139
21.000   2.0026   0.04181  0.03719 -0.0030  0.0315  0.7170
22.000   2.0099   0.05073  0.04623 -0.0018  0.0276  0.7222
23.000   2.0003   0.06214  0.05782 -0.0022  0.0247  0.7298
24.000   1.9663   0.07718  0.07309 -0.0051  0.0218  0.7342
25.000   1.8877   0.09910  0.09535 -0.0126  0.0197  0.7384
26.000   1.7218   0.13641  0.13326 -0.0325  0.0195  0.7421

```

*Fig. 4.6: A sample XFOIL output file*

# Chapter 5

## Validation and Results

### 5.1 High Lift, High Reynolds Number Airfoil Optimization

Herein, the proposed optimization procedure is put into the test in order to examine its robustness and effectiveness when the Area-Preserving FFD comes into play. The optimization process is utilized for the design of a high lift airfoil, operating at Reynolds number  $Re = 7 * 10^6$ , based on the DU-06-W-200 (DU series) as the reference one, which was developed in Delft University of Technology with major applications in the design of wind turbine's blades sections.

The parameterization technique, used for the manipulation of the geometry, is a B-Spline-based FFD presented in Section 2.2. The initial 2D FFD lattice was built around the DU-06-W-200 by setting 6 control points along the  $x$  (chordwise) and 3 control points along to the  $y$  direction, as shown in Figure 2.5, whereas the degrees of the B-Spline Basis functions were set as the highest possible, e.g. 5 and 2 respectively in order to achieve a smooth representation of the airfoil geometry.

Consequently, the number of the design variables is partially determined. As already mentioned, the control points that correspond to leading and trailing edges ( $x = 0$  and  $x = 1$ ) are fixed, therefore the number of the design variables is 24, corresponding to the  $x$  and  $y$  coordinates of the 12 free to move control points of the FFD lattice (Figure 2.7). The permitted range of the design variables varies between 6% and 20% of the chord, in the direction normal to the chord ( $y$  direction), whereas in the chordwise direction ( $x$  direction) was set as 10% of the chord length. The extraction of the presented ranges was based on a trial and error basis, in order to achieve upper and lower bounds that do not restrict or magnify inefficiently the search space. In Table 5.1 the employed design variables during this particular optimization process are presented along with their respective upper and lower bounds. The enumeration of the control points of the FFD lattice is realized as follows;  $CP_{ij}$  refers to the control point

located in the  $i^{th}$  column and  $j^{th}$  line of the control grid, where  $i = 0,1,\dots,n$  and  $j = 0,1,\dots,m$ , as illustrated in Figure 5.1.

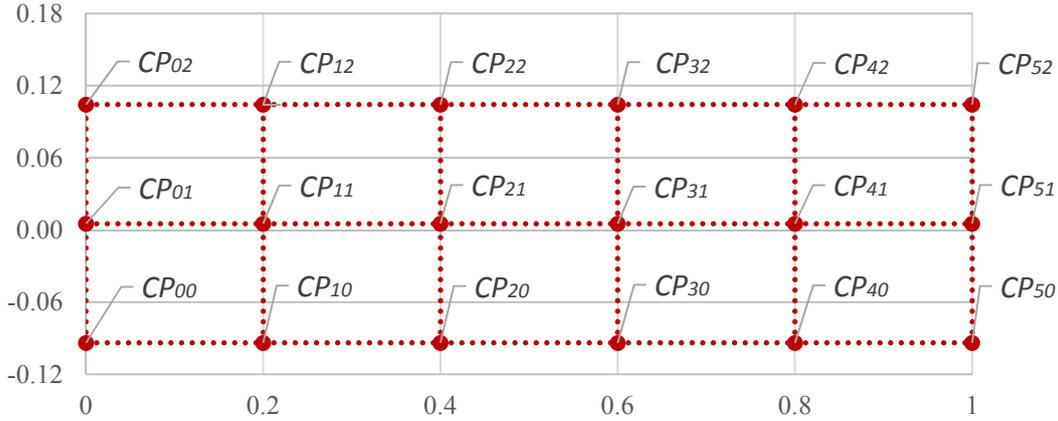


Fig. 5.1: FFD Control Points enumeration

No.	Design Variable	Lower Bnd	Upper Bnd	No.	Design Variable	Lower Bnd	Upper Bnd
1	$CP_{10} - x$ Coord	0.15	0.25	13	$CP_{10} - y$ Coord	-0.2	0
2	$CP_{11} - x$ Coord	0.15	0.25	14	$CP_{11} - y$ Coord	-0.03	0.03
3	$CP_{12} - x$ Coord	0.15	0.25	15	$CP_{12} - y$ Coord	0.1	0.2
4	$CP_{20} - x$ Coord	0.35	0.45	16	$CP_{20} - y$ Coord	-0.2	0
5	$CP_{21} - x$ Coord	0.35	0.45	17	$CP_{21} - y$ Coord	-0.03	0.03
6	$CP_{22} - x$ Coord	0.35	0.45	18	$CP_{22} - y$ Coord	0.1	0.2
7	$CP_{30} - x$ Coord	0.45	0.65	19	$CP_{30} - y$ Coord	-0.1	0.1
8	$CP_{31} - x$ Coord	0.45	0.65	20	$CP_{31} - y$ Coord	0	0.1
9	$CP_{32} - x$ Coord	0.45	0.65	21	$CP_{32} - y$ Coord	0.1	0.2
10	$CP_{40} - x$ Coord	0.65	0.85	22	$CP_{40} - y$ Coord	-0.1	0.1
11	$CP_{41} - x$ Coord	0.65	0.85	23	$CP_{41} - y$ Coord	0	0.1
12	$CP_{42} - x$ Coord	0.65	0.85	24	$CP_{42} - y$ Coord	0.1	0.2

Tab. 5.1: Design Variables

The optimization problem examined herein is defined as the design of a new airfoil with maximum mean lift coefficient in a range of angles of attack between  $0^\circ$  and  $26^\circ$ , with respect to one geometric and one aerodynamic inequality constraint that concern the position of airfoil's centroid in  $x$  axis and the maximum allowed absolute value of pitching moment coefficient respectively, based on the geometry of DU-06-

W-200. Moreover, during the optimization procedure, a strict area preservation constraint is utilized by the direct implementation of the AP FFD methodology. The formation of the aforementioned constraints is presented below.

*Centroid constraint:*

$$x_{c,ref} - 0.05 \leq x_c \leq x_{c,ref} + 0.05 \quad (5.1)$$

*Moment coefficient constraint:*

$$|\overline{C_m}| \leq |C_{m,ref}| \quad (5.2)$$

$$|\overline{C_m}| = \frac{1}{27} \sum_{i=0}^{26} |c_{m_i}| \quad (5.3)$$

*Cross-sectional area constraint:*

$$\bar{A} = A_{ref} \quad (5.4)$$

where  $x_c$ ,  $|\overline{C_m}|$  and  $\bar{A}$  are the  $x$  coordinate of the centroid, the mean absolute value of pitching moment coefficient and the cross-sectional area (after the implementation of the AP FFD) of each candidate solution respectively. For the DU-06-W-200 airfoil, examined herein, the reference values are  $A_{ref} = 0.1223$ ,  $x_{c,ref} = 0.39$  and  $|C_{m,ref}| = 0.0315$  computed based on the XFOIL results in a range between  $0^\circ$  and  $26^\circ$  a.o.a with a step of  $1^\circ$  and fixed Reynolds number equal to  $7 \cdot 10^6$ . The constraints in Eq. (5.1) and Eq. (5.2) are satisfied utilizing two penalty functions. In that way an indirect manipulation of the constraints is achieved as the algorithm tries to lead the design in geometries that produce lower (better) values of the fitness function. By the utilization of the penalty functions included in the fitness, in conjunction with the employment of the AP FFD for the satisfaction of the required constraints, the problem is transposed into an unconstrained minimization one. The formalization of the penalty functions is as follows:

*Centroid penalty function:*

$$f_x = \begin{cases} 1, & \text{if } x_{c,ref} - 0.05 \leq x_c \leq x_{c,ref} + 0.05 \\ 1 - 2|x_{c,ref} - x_c|, & \text{otherwise} \end{cases} \quad (5.5)$$

Pitching moment penalty function:

$$f_m = 1 - \sum_{i=0}^{26} f_{m_i} \quad (5.6)$$

$$f_{m_i} = \begin{cases} 0, & \text{if } |c_{m_i}| \leq |c_{m,ref}| \\ |c_{m_i}| - |c_{m,ref}|, & \text{otherwise} \end{cases} \quad (5.7)$$

Therefore, the fitness function of the under minimization unconstrained problem, is formed as

$$f = 2 - \bar{C}_l f_x f_m \quad (5.8)$$

The mean lift coefficient is calculated by computing the area below the  $C_l$  curve and dividing this area by 27, which is the number of different angles of attack in which the airfoil is evaluated. This approach benefits airfoils that converge in more angles of attack, against others that do not perform in a smooth way with respect to the whole spectrum of the examined angles.

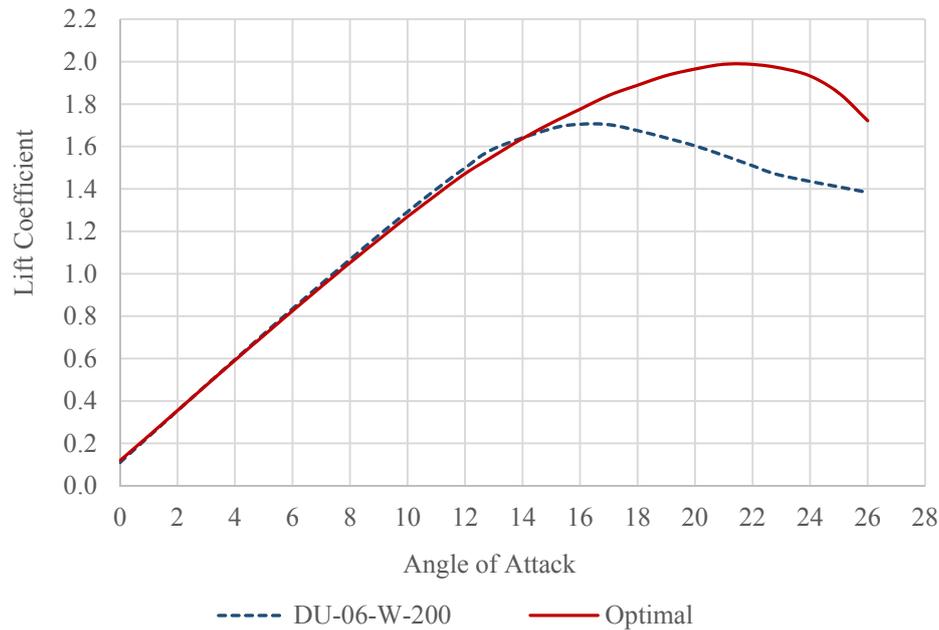
Regarding the use of the surrogate models, in the current test case a combination of MLP and RBF ANNs was utilized. The first two generations of the DE perform exact evaluations for every chromosome, in order to create an adequate training sample for the ANNs. Subsequently, in each generation the ANN with the smallest testing error is chosen to serve as a surrogate model.

Further acceleration was achieved by utilizing a parallel version of the aforementioned DE on a DELL™ R815 PowerEdge™ server with four AMD Opteron™ 6380 sixteen-core processors at 2.50 GHz (64 cores in total). The population size  $N_p$  of the DE was set equal to 50, while the algorithm was executed for 1000 generations.

## 5.2 Computational Results

In Figure 5.2 a comparison of the lift coefficient for the initial and the optimized airfoil is presented. It is becoming quite evident that the produced airfoil has an improved performance (relating to the adopted design criteria) at the region between  $16^\circ$  to  $26^\circ$  a.o.a. by achieving a maximum lift coefficient at  $21^\circ$  instead of  $16^\circ$ . Additionally, a maximum lift coefficient increase of 16.57% was achieved while the mean lift coefficient  $\bar{C}_l$ , at the range of  $(0^\circ - 26^\circ)$  a.o.a was increased by 28.69 %.

Consequently, after the optimization process an expansion of the high lift region is attained as well as the capacity of high lift production in greater a.o.a.

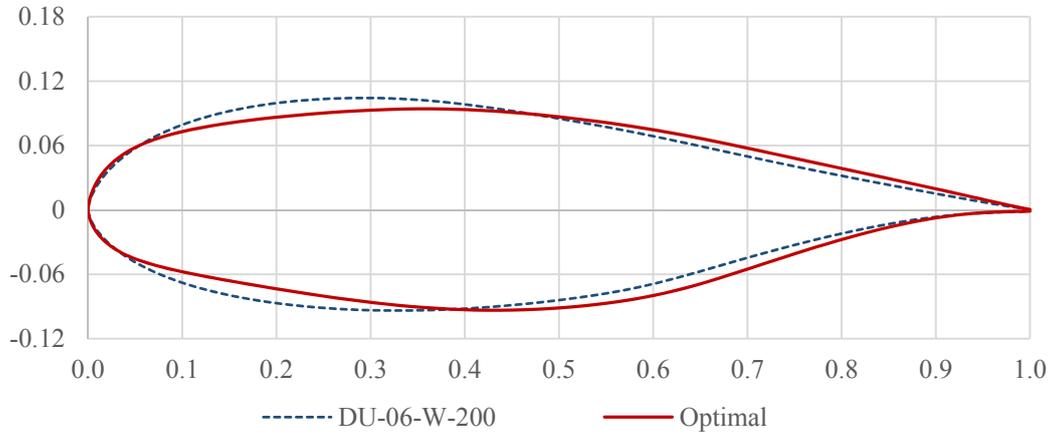


**Fig. 5.2:** Comparison between the lift coefficient of the reference airfoil DU-06-W-200 and the optimal one as a function of a.o.a.

In Figure 5.3 the optimal and the initial airfoils are presented. It should be noticed that a relatively small modification to the initial geometry produces quite different results, highlighting the strong non-linearity of the search space. The cross-sectional area of the produced airfoil is exactly the same to the initial one, proving that AP FFD is a versatile methodology that can be easily adopted in optimization schemes in order to replace other techniques of satisfying a strict area constraint without any loss of the convergence capability and efficiency that DE offers.

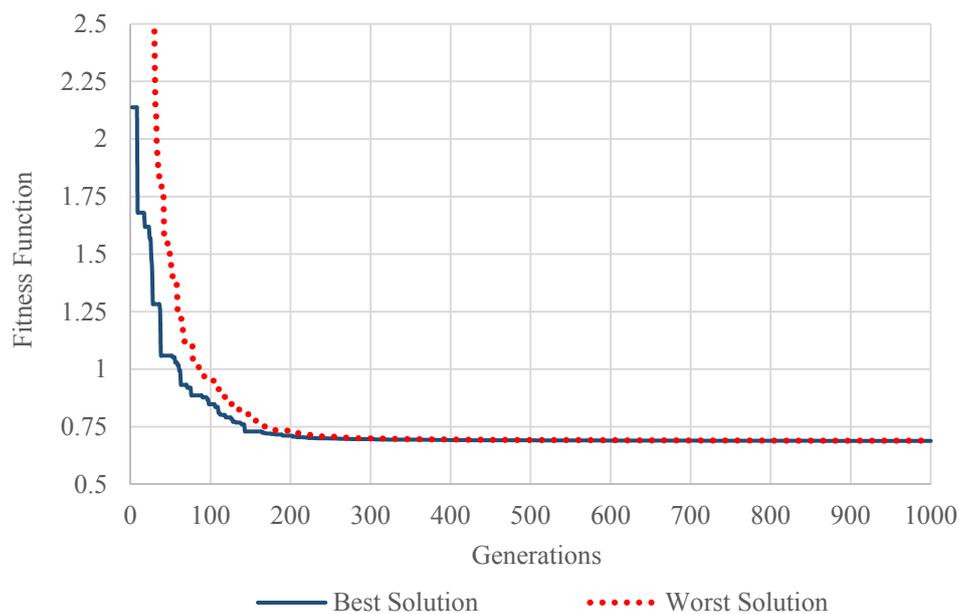
In Figure 5.4 the convergence history of the fitness function for the best and worst chromosomes of each generation is presented. Additionally, in order to prove that the proposed AP FFD technique could easily and effectively replace a penalty function approach, regarding the satisfaction of an exact area preservation constraint, the same test case was executed for a second time, by utilizing a classic penalty function method instead an AP FFD one, while all the other parameters maintained identical. In Figure 5.5 a comparison between the convergence histories of the best solution using a AP FFD and a penalty function approach is illustrated. As shown, the employment of the proposed AP FFD instead of a penalty function does not cause any loss of the convergence capability and efficiency that DE provides. On the contrary, it leads to a better value of the fitness function and produces an airfoil geometry with higher lift

coefficient than the one produced by the optimization process with the penalty function approach, as shown in Figure 5.6, regarding the range of high angles of attack ( $18^\circ - 26^\circ$ ), where high lift is more necessary.

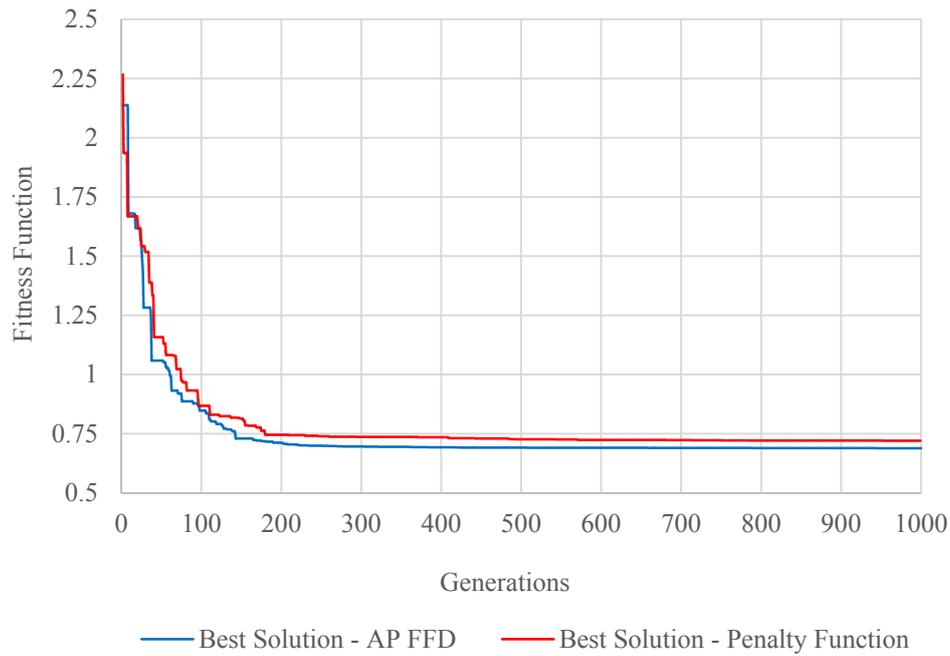


**Fig. 5.3:** Initial and optimal airfoil geometries.

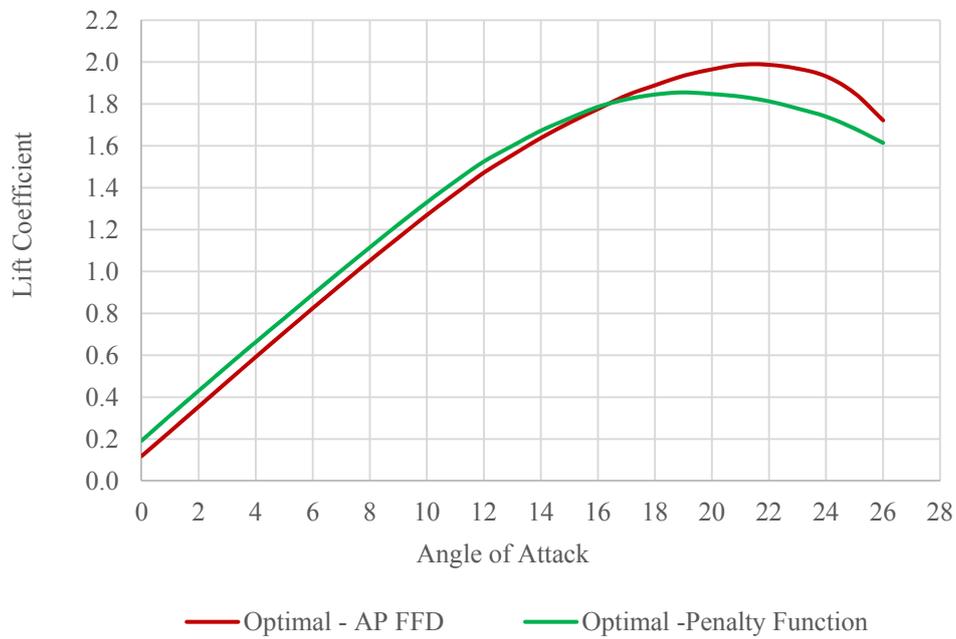
Besides the high performance of the optimal airfoil concerning the mean lift coefficient, a noticeable drag coefficient reduction is achieved within the same region between  $16^\circ$  and  $26^\circ$  a.o.a., which consequently leads to higher lift-to-drag ratios. Figures 5.7, 5.8 present a comparison between the initial and the optimal airfoil on drag coefficient and lift-to-drag ratio respectively, in order to highlight the improved aerodynamic properties resulted by the optimization procedure without any significant loss of the structural characteristics.



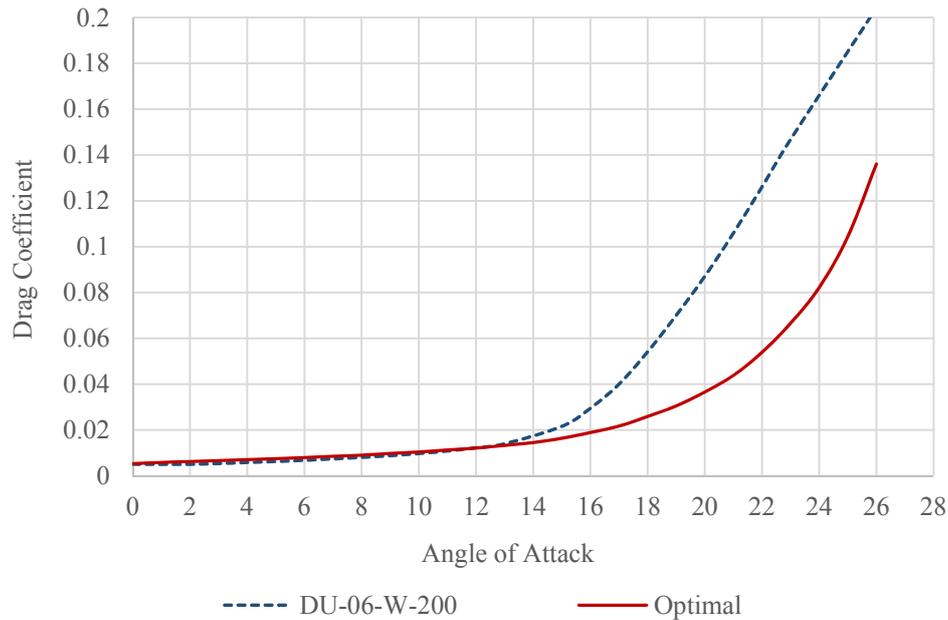
**Fig. 5.4:** The convergence history of the best and worst solution of each generation.



**Fig. 5.5:** The convergence history of the best solution of each generation by using an AP FFD and a penalty function approach for the satisfaction of the cross sectional area constraint.



**Fig. 5.6:** Comparison between the lift coefficient of the optimal airfoil produced by an optimization scheme that employs AP FFD and the optimal one produced by a classical optimization scheme utilizing a penalty function approach.



**Fig. 5.7:** Comparison between the drag coefficient of the reference airfoil DU-06-W-200 and the optimal as a function of a.o.a.

In Table 5.2 the total elapsed computation time is presented as well as the number of exact and total evaluations. Note that using surrogate models results in a reduction of the total number of exact evaluations by a factor of 0.48. Although in this case the time required for each exact evaluation is about 1-2 seconds, in cases where costly Computational Fluid Dynamics (CFD) models are utilized as flow solvers, each exact evaluation may be last a few hours, so a decrease of computational time at almost in half is crucial for the efficiency of the design process.

Test Case	Wall-Clock Time (s)	Exact Evaluations	Total Evaluations
DU-06-W-200	8820	25860	50000

**Tab. 5.2:** Wall-clock computation time, number of exact and total evaluations.

Figure 5.9 contains a comparison between the pitching moment coefficient of the reference airfoil DU-06-W-200 and the produced one as a function of a.o.a. The value of moment coefficient is quite important given that airfoil sections that form a specific blade should have similar values of moment coefficients in order to avoid the development of high torsional stresses on the blade. Table 5.3 illustrates the resulting values of the constraint quantities, compared to the reference ones, as well as the upper

and lower bounds of each constraint. At the end of the optimization procedure both centroid and pitching moment inequality constraints are satisfied. Concerning the cross-sectional area preservation, the equality constraint was exactly satisfied by the application of AP FFD to each candidate solution.

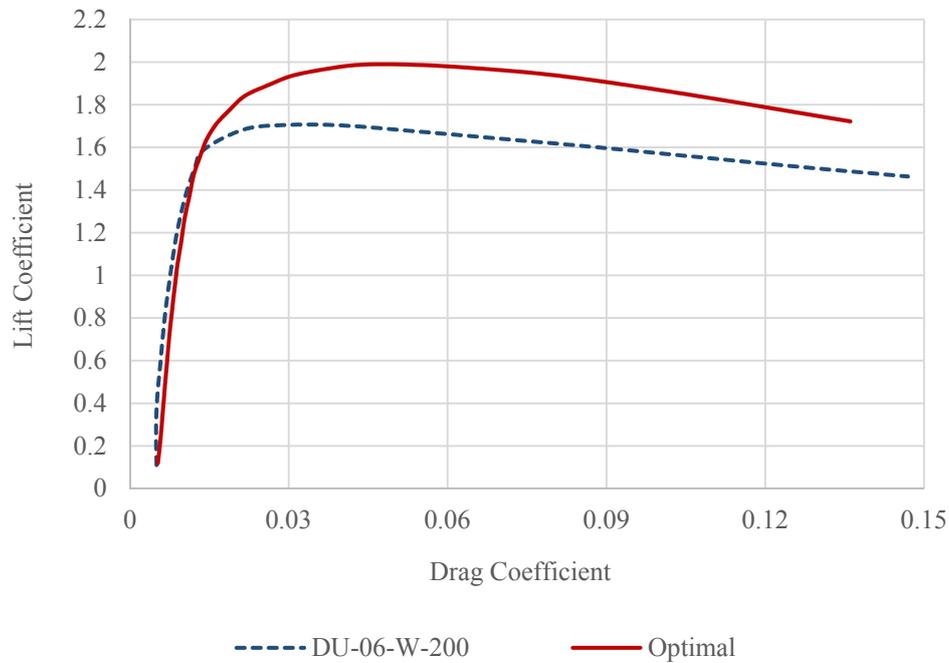


Fig. 5.8: Comparison between the lift-to-drag ratios of the reference DU-06-W-200 airfoil and the optimal one.

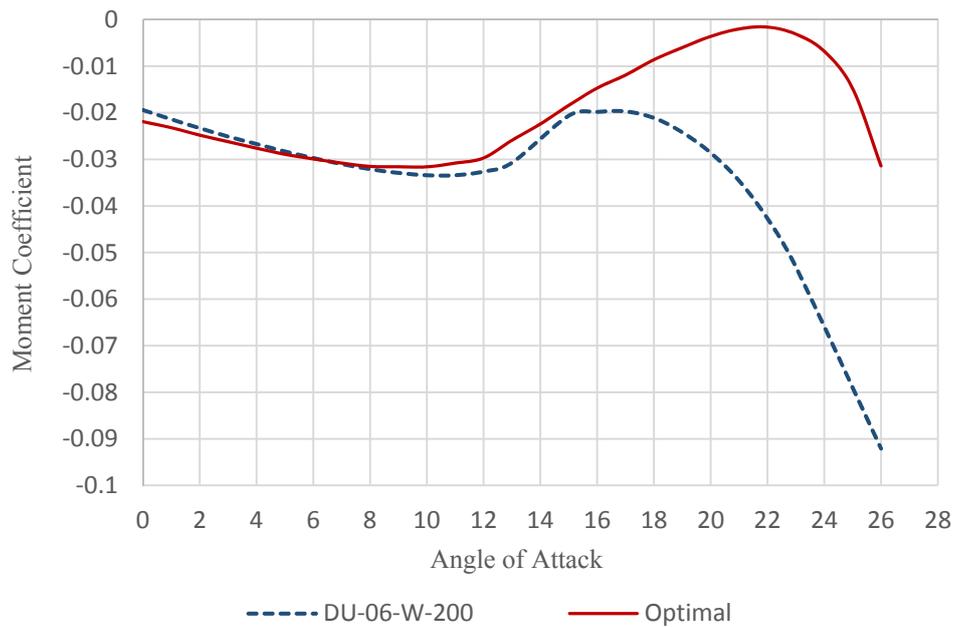


Fig. 5.9: Comparison between the pitching moment coefficient of the reference airfoil DU-06-W-200 and the optimal one as a function of a.o.a.

Constraint	Satisfaction Approach	Optimal Airfoil Value	Reference value	Lower Bound	Upper Bound
$x_c$	Penalty Function	0.4165	0.393	0.3430	0.4430
$ \overline{C_m} $	Penalty Function	0.0200	0.0315	0	0.0315
$\overline{A}$	AP FFD	0.1223	0.1223	0.1223	0.1223

**Tab. 5.3:** Output values for the constraints

# Chapter 6

## Conclusions

In this diploma thesis, a novel direct numerical airfoil optimization scheme is proposed, which employs the Area-Preserving FFD as an alternative technique for the exact satisfaction of a strict cross-sectional area equality constraint, instead of the classic penalty function approach; it is combined with a 2D B-Spline-based FFD as geometrical parameterization technique. The exact area preservation is achieved by solving an area correction sub-problem, which consists of computing and applying the minimum possible offset to each free-to-move control point of the deformed FFD lattice, after the application of a typical FFD, subject to the area preservation constraint. Due to the linearity of the area constraint in each axis, a closed-form solution to the sub-problem is able to be extracted, which, in conjunction with a surrogate-assisted Differential Evolution algorithm, renders the optimization procedure time efficient and effective.

The proposed optimization scheme was tested through a lift coefficient maximization problem, based on a DU-06-W-200 airfoil as the reference one. The validation results indicate a fine collaboration between the DE algorithm and the proposed AP FFD methodology, even though the aforementioned approach alters the produced by the DE algorithm chromosomes prior to their evaluation. Additionally, the results show the promising capabilities of the introduced procedure to achieve an improvement of the aerodynamic airfoil characteristics, without violating the area preservation constraint.

Finally the AP FFD technique was compared to the classic penalty function approach, concerning the cross-sectional area constraint, showing that the proposed methodology is able to attain better values of the objective function without any loss of the convergence capabilities of the DEVA

## References

- [1] Eyi, S., Lee, K. D., Rogers, S. E., Kwak, D., 1996, “High-lift design optimization using Navier-Stokes equations”, *AIAA J. Aircr.*, **33**(3), pp. 499–504.
- [2] Duvigneau, R., Visonneau, M., 2006, “Simulation and optimization of stall control for an airfoil with a synthetic jet” *Aerosp. Sci. Technol.*, **10**(4), pp. 279–287.
- [3] Leifsson, L., Koziel, S., 2010, “Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction”, *J. Comput. Sci.*, **1**(2), pp. 98–106.
- [4] Nikolos, I.K., 2013, “On the use of multiple surrogates within a differential evolution procedure for high-lift airfoil design”, *Int. J. Adv. Intell. Paradigms*, **5**(4), pp. 319.
- [5] Grasso, F., 2013, “Development of thick airfoils for wind turbines”, *AIAA J. Aircr.*, **50**(3), pp. 975–981.
- [6] Poole, D. J., Allen, C. B., Rendall, T., 2014, “Aerofoil Inviscid Drag Minimization by Constrained Global Optimization”, 5<sup>th</sup> European Conference on Computational Mechanics (ECCM V), July 2025, Barcelona, Spain
- [7] Green, B. E., Whitesides, J. L., Campbell, R. L., Mineck, R. E., 1997, “Method for the Constrained Design of Natural Laminar Flow Airfoils”, *AIAA J. Aircr.*, **34**(6), pp. 706–712.
- [8] Dennis, B. H., Dulikravich, G. S., Han, Z.-X., “Constrained shape optimization of airfoil cascades using a Navier-Stokes solver and a genetic/SQP algorithm”, ASME International Gas Turbine and Aeroengine Congress and Exhibition, Indianapolis, IN, USA, June 7-10, 1999.
- [9] Brian H. Dennis, George S. Dulikravich, and Zhen-Xue Han. "Optimization of Turbomachinery Airfoils with a Genetic/ Sequential-Quadratic-Programming Algorithm", *Journal of Propulsion and Power*, Vol. 17, No. 5 (2001), pp. 1123-1128.
- [10] Ahn, J., Kim, H. J., Lee, D. H., Rho, O. H., 2001, “Response Surface Method for Airfoil Design in Transonic Flow”, *AIAA J. Aircr.*, **38**(2), pp. 231–238.
- [11] Ribeiro, A. F. P., Awruch, A. M., Gomes, H. M., 2012, “An airfoil optimization technique for wind turbines”, *Appl. Math. Model.*, **36**(10), pp. 4898–4907.
- [12] Lee, K. D., Eyi, S., 1993, “Transonic airfoil design by constrained optimization”, *AIAA J. Aircr.*, **30**(6), pp. 805–806.
- [13] Jeong, S., Yamamoto, K., Obayashi, S., “Kriging-based Probabilistic Method for Constrained Multi-Objective Optimization Problem”, *AIAA 1st Intelligent Systems Technical Conference*, , Chicago, Illinois, USA, 20 – 22 September 2004

- [14] Zingg, D. W., Diosady, L., Billing, L., “Adaptive Airfoils for Drag Reduction at Transonic Speeds”, 24<sup>th</sup> AIAA Applied Aerodynamics Conference, 5 – 8 June 2006, San Francisco, California, USA
- [15] Driver, J., Zingg, D. W., 2007, “Numerical Aerodynamic Optimization Incorporating Laminar-Turbulent Transition Prediction”, AIAA J., **45**(8), pp. 1810–1818.
- [16] Oyama, A., Liou, M. S., Obayashi, S., 2004, “Transonic Axial-Flow Blade Optimization: Evolutionary Algorithms/Three-Dimensional Navier-Stokes Solver”, J. Propuls. Power, **20**(4), pp. 612–619.
- [17] Giannakoglou, K. C., Papadimitriou, D. I., Karpolis, I. C., 2006, “Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels”, Comput. Methods Appl. Mech. Eng., **195**(44–47), pp. 6312–6329.
- [18] Lee, D. S., Gonzalez, L. F., Srinivas, K., Periaux, J., 2008, “Robust evolutionary algorithms for UAV/UCAV aerodynamic and RCS design optimisation”, Comput. Fluids, **37**(5), pp. 547–564.
- [19] Montoya, F. G., Manzano-Agugliaro, F., López-Márquez, S., Hernández-Escobedo, Q., Gil, C., 2014, “Wind turbine selection for wind farm layout using multi-objective evolutionary algorithms”, Expert Syst. Appl., **41**(15), pp. 6585–6595.
- [20] Back, T., Hammel, U., Schwefel, H.P., 1997, “Evolutionary computation: comments on the history and current state”, IEEE Trans. Evol. Comput., **1**(1), pp. 3–17.
- [21] Coello, C. A. C., 2000, “Constraint-Handling using an Evolutionary Multiobjective Optimization Technique”, Civ. Eng. Environ. Syst., **17**(4), pp. 319–346.
- [22] Richardson, J. T., Palmer, M. R., Liepins, G. E., Hilliard, M., 1989, “Some Guidelines for Genetic Algorithms with Penalty Functions,” Proceedings of the Third International Conference on Genetic Algorithms, San Francisco, California, USA, pp. 191–197.
- [23] Deb, K., 2000, “An efficient constraint handling method for genetic algorithms”, Comput. Methods Appl. Mech. Eng., **186**(2–4), pp. 311–338.
- [24] Kim, H. J., Rho, O. H., 1997, “Dual-point design of transonic airfoils using the hybrid inverse optimization method”, AIAA J. Aircr., **34**(5), pp. 612–618.
- [25] Andreoli, M., Ales, J., Desideri, J-A, “Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics”, RR-5019, 2003.
- [26] Samareh, J. A., 1999, “A survey of shape parameterization techniques”, NASA Conference Publication, Citeseer, pp. 333–344.
- [27] Pickett, R. M., Rubinstein, M. F., and Nelson, R. B., 1973, “Automated Structural Synthesis Using a Reduced Number of Design Coordinates”, AIAA J., **11**(4), pp. 489–494.

- [28] Bloor, M. I. G., and Wilson, M. J., 1995, “Efficient parameterization of generic aircraft geometry”, *AIAA J. Aircr.*, **32**(6), pp. 1269–1275.
- [29] Piegl, L., and Tiller, W., 1997, “The NURBS Book”, Springer-Verlag Berlin Heidelberg.
- [30] Lacourse, D. E., “Handbook of Solid Modeling”, 1995, McGraw-Hill, New York.
- [31] Hicks, R. M., Henne, P. A., 1978, “Wing Design by Numerical Optimization”, *AIAA J. Aircr.*, **15**(7), pp. 407–412.
- [32] Sederberg, T. W., Parry, S. R., 1986, “Free-form Deformation of Solid Geometric Models”, *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, pp. 151–160.
- [33] Barr, A. H., 1984, “Global and Local Deformations of Solid Primitives”, *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, pp. 21–30.
- [34] Oyama, A., Fujii, K., 2004, “Airfoil Design Optimization for Airplane for Mars Exploration”, *The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*.
- [35] Secanell, M., Suleman, A., 2005, “Numerical evaluation of optimization algorithms for low-Reynolds-number aerodynamic shape optimization”, *AIAA J.*, **43**(10), pp. 2262–2267.
- [36] Hacıoğlu, A., Özkol, I., 2005, “Inverse airfoil design by using an accelerated genetic algorithm via distribution strategies”, *Inverse Probl. Sci. Eng.*, **13**(6), pp. 563–579.
- [37] Della Vecchia, P., Daniele, E., D’Amato, E., 2014, “An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm”, *Aerosp. Sci. Technol.*, **32**(1), pp. 103–110.
- [38] Sobieczky, H., 1997, “Geometry Generator for CFD and Applied Aerodynamics”, *New Design Concepts for High Speed Air Transport*, Springer Vienna, pp. 137–157.
- [39] Sobieczky, H., 1999, “Parametric Airfoils and Wings”, *Recent Development of Aerodynamic Design Methodologies*, Vieweg+Teubner Verlag, pp. 71–87.
- [40] Kharal, A., Saleem, A., 2012, “Neural networks based airfoil generation for a given using Bezier–PARSEC parameterization”, *Aerosp. Sci. Technol.*, **23**(1), pp. 330–344.
- [41] Liang, Y., Cheng, X., Li, Z., Xiang, J., 2010, “Multi-objective robust airfoil optimization based on non-uniform rational B-spline (NURBS) representation”, *Sci. China Technol. Sci.*, **53**(10), pp. 2708–2717.
- [42] Samareh, J. A., 1999, “A Novel Shape Parameterization Approach”, Report No. NASA-TM-1999-209116.
- [43] Lamousin, H. J., Waggenspack, J., W.N., 1994, “NURBS-based free-form deformations”, *IEEE Comput. Graph. Appl.*, **14**(6), pp. 59–65.

- [44] Amoiralis, E. I., Nikolos, I. K., 2008, “Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design”, *J. Comput. Inf. Sci. Eng.*, **8**(2).
- [45] Ghisu, T., Jarrett, J. P., Parks, G. T., 2011, “Robust design optimization of airfoils with respect to ice accretion”, *AIAA J. Aircr.*, **48**(1), pp. 287–304.
- [46] Zhao K., Gao, Z.-H., Huang J.-T., 2014, “Robust design of natural laminar flow supercritical airfoil by multi-objective evolution method”, *Appl. Math. Mech.*, **35**(2), pp. 191–202.
- [47] Amoignon, O., Hradil, J., Navratil, J., 2014, “A numerical study of adaptive FFD in aerodynamic shape optimization”, *Proc. 52nd Aerospace Sciences Meeting, AIAA*, National Harbor, Maryland.
- [48] Hahmann, S., Bonneau, G.P., Barbier, S., Elber, G., Hagen, H., 2011, “Volume-preserving FFD for programmable graphics hardware”, *Vis. Comput.*, **28**(3), pp. 231–245.
- [49] Griessmair, J., Purgathofer, W., “Deformation of solids with trivariate B-splines”, *Proceedings of Eurographics*, North Holland, Hamburg, September 1989, pp. 137–148.
- [50] Coquillart, S., 1990, “Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling,” *Comput. Graph.*, **24**(4), pp. 187–193.
- [51] MacCracken, R., Joy, K. I., 1996, “Free-form deformations with lattices of arbitrary topology”, *SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA pp. 181–188.
- [52] Ilic, S., Fua, P., 2002, “Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data”, *Computer Vision*, Springer Berlin Heidelberg, pp. 704–717.
- [53] Kobayashi, K. G., and Ootsubo, K., 2003, “t-FFD: Free-Form Deformation by Using Triangular Mesh”, *Proceedings of the Eighth ACM symposium on Solid Modeling and Applications*, ACM Press, pp. 226–234.
- [54] Song, W., and Yang, X., 2005, “Free-Form Deformation With Weighted T-Spline”, *The Visual Computer: International Journal of Computer Graphics*, **21**(3), pp. 139–151.
- [55] Sederberg, T. W., Zheng, J., Bakenov, A., Nasri, A., 2003, “T-splines and T-NURCCs”, *SIGGRAPH '03 Proceedings*, ACM, New York, NY, USA, pp. 477–484.
- [56] Patrikalakis, N. M., Maekawa, T., 2010, “Shape Interrogation for Computer Aided Design and Manufacturing”, *Springer Berlin Heidelberg*.
- [57] Yu, X., and Gen, M., 2010, *Introduction to Evolutionary Algorithms*, Springer London, London.

- [58] Nikolos, I.K., 2011, “Surrogate Modeling in Evolutionary Based Engineering Design Optimization”, Computational Science, Engineering & Technology Series, Saxe-Coburg Publications, Stirlingshire, UK, pp. 173–203.
- [59] Coello, C. A. C., Lamont, G. B., Veldhuizen, D. A. V., 2006, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [60] Storn, R., and Price, K., 1995, “DE—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Space”, ICSI, Technical Report No. TR-95-012.
- [61] Price, K.V., Storn, R.M. Lampinen, J.A., 2005, “Differential Evolution, a Practical Approach to Global Optimization”, Springer-Verlag, Berlin, Heidelberg.
- [62] Giannakoglou, K. C., 2002, “Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence”, *Prog. Aerosp. Sci.*, **38**(1), pp. 43–76.
- [63] Torczon, V., Trosset, M. W., 1998, “Using approximations to accelerate engineering design optimization”, *Proceedings Of The 7th AIAA/USAF/NASA/ISSMO Multidisciplinary Analysis & Optimization Symposium (Held At Saint Louis, Missouri)*, Paper 98-4800
- [64] Forrester, A. I. J., Keane, A. J., 2009, “Recent advances in surrogate-based optimization”, *Prog. Aerosp. Sci.*, **45**(1–3), pp. 50–79.
- [65] Yaochu Jin, Olhofer, M., Sendhof, B., 2000, “On Evolutionary Optimization with Approximate Fitness Functions”, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '00)*, Las Vegas, Nevada, USA, July 8-12, 2000, pp. 786–793.
- [66] Ong, Y. S., Nair, P. B., Keane, A. J., 2003, “Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling”, *AIAA J.*, **41**(4), pp. 687–696.
- [67] Regis, R. G., Shoemaker, C. A., 2004, “Local function approximation in evolutionary algorithms for the optimization of costly functions,” *IEEE Trans. Evol. Comput.*, **8**(5), pp. 490–505.
- [68] Zhang, J., Sanderson, A. C., 2009, “Adaptive Differential Evolution”, Springer, Berlin, Heidelberg
- [69] Kröse, B., Krose, B., Smagt, P. van der, Smagt, P., 1993, “An introduction to Neural Networks”, University of Amsterdam
- [70] Haykin, S., 1998, “Neural Networks: A Comprehensive Foundation”, Prentice Hall PTR, Upper Saddle River, NJ, USA.
- [71] Nikolos, I.K., 2004, “Inverse design of aerodynamic shapes using differential evolution coupled with artificial neural network”, *Proc. Conf. ERCOFTAC in Design Optimization: Methods and Applications*, Athens.

[72] Nikolos, I.K., Papadopoulou, M.P. Karatzas, G.P., 2010, “Artificial neural network and differential evolution methodologies used in single- and multi-objective formulations for the solution of subsurface water management problems”, *Int. J. Adv. Intell. Paradigms*, **2**(4), pp.365–377.

[73] Drela, M., 1989, “XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils”, *Proc. Conf. on Low Reynolds Number Airfoil Aerodynamics*, University of Notre Dame, Notre Dame, Indiana