VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
LETECKÝ ÚSTAV

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AEROSPACE ENGINEERING

ADAPTIVE PARAMETERIZATION FOR AERODYNAMIC SHAPE OPTIMIZATION IN AERONAUTICAL APPLICATIONS

ADAPTIVNÍ TVAROVÁ PARAMETRIZACE PRO AERODYNAMICKÉ OPTIMALIZACE V LETECKÝCH APLIKACÍCH

DIZERTAČNÍ PRÁCE
DOCTORAL THESIS

AUTOR PRÁCE
AUTHOR

Ing. JIŘÍ HRADIL

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Ing. ANTONÍN PÍŠTĚK, CSc.

BRNO 2015

ABSTRACT

The goal of this doctoral thesis is to analyze and develop parameterization algorithms for 2D and 3D shape optimization in the context of industrial aircraft aerodynamic design based on simulations with CFD.

Aerodynamic shape optimization is an efficient tool that aims at reducing the cost of the process of aircraft design. A tool that is based on automatization of the search for the optimum shape. Key part of successful aerodynamic shape optimization is the use of appropriate parameterization method, a method that should guarantee the possibility of reaching optimum shape.

The parameterization methods used in aerodynamic shape optimizations are still not ready for complex industrial applications, which are present on modern passenger aircrafts with swept cranked wings with winglets and engine pylons, fuselage-wing interactions etc. So there is a need for general parameterization method that applies on wide variety of different geometries.The Free-Form Deformation (FFD[1]) parameterization can, thanks to its geometry handling qualities, be the answer to this need.

Adaptive parameterization should automatically modify parameterization grid (lattice) to get appropriate lattice in regions of interest. Such that will allow sufficient control of deformations of the object with respect to reaching optimum shape and fulfilling optimization constraints. First application is in the surface deformation. The other proposed goal is development of the FFD parameterization that can do both surface deformations and CFD mesh deformations, while enabling large object deformations and preserving the level of mesh quality during the process.

KEYWORDS

FFD, adaptive parameterization, mesh deformation, adjoint, CFD, unstructured meshes

ABSTRAKT

Cílem mé disertační práce je analyzovat a vyvinout parametrizační metodu pro 2D a 3D tvarové optimalizace v kontextu průmyslového aerodynamického návrhu letounu založeném na CFD simulacích.

Aerodynamická tvarová optimalizace je efektivní nástroj, který si klade za cíl snížení nákladů na návrh letounů. Nástroj založený na automatickém hledání optimálního tvaru. Klíčovou částí úspěšného optimalizačního procesu je použití vhodné parametrizační metody, metody schopné garantovat možnost dosažení optimálního tvaru. Parametrizační metody obecně používané v oblasti aerodynamické tvarové optimalizace momentálně nejsou připravený na komplikované průmyslové aplikace vyskytující se u moderních dopravních letounů, které mají šípová zalomená křídla s winglety a motorovými gondolami, přechodové prvky spojující např. trup s křídlem atd.. Existuje tedy potřeba nalezení obecné parametrizační metody, která bude aplikovatelná na širokou škálu různých geometrických tvarů. Free-Form Deformation (FFD[1]) parametrizace může, vzhledem ke svým schopnostem při zacházení s geometrií, být odpovědí na tuto potřebu.

Adaptivní parametrizace by se měla být schopna automaticky přizpůsobit danému tvaru tak, aby byly její kontrolní body vhodně rozmístěny. Což umožní dostatečnou kontrolu deformací objektu, která zaručí možnost vytvoření optimálního tvaru objektu a splnění geometrických omezení.

Primární aplikací takové parametrizační metody je deformace tvaru objektu. Dalším navrhovaným cílem je modifikace FFD parametrizační metody pro současné deformace tvaru objektu a CFD výpočetní sítě, umožňující velké deformace objektu při zachování kvality výpočetní sítě.

KLÍČOVÁ SLOVA

FFD, adaptivní parametrizace, deformace sítě, adjoint, CFD, nestrukturované sítě

HRADIL, Jiří *Adaptive parameterization for aerodynamic shape optimization in aeronautical applications*: doctoral thesis. BRNO: Brno University of Technology, Faculty of Mechanical Engineering, Institute of aerospace engineering, 2015. 136 p. Supervised by prof. Ing. Antonín Píštěk, CSc.

DECLARATION

I declare that I have written my doctoral thesis on the theme of "Adaptive parameterization for aerodynamic shape optimization in aeronautical applications" independently, under the guidance of the doctoral thesis supervisor and using the technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the doctoral thesis I furthermore declare that, as regards the creation of this doctoral thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal and/or ownership rights and I am fully aware of the consequences in the case of breaking Regulation § 11 and the following of the Copyright Act No 121/2000 Sb., and of the rights related to intellectual property right and changes in some Acts (Intellectual Property Act) and formulated in later regulations, inclusive of the possible consequences resulting from the provisions of Criminal Act No 40/2009 Sb., Section 2, Head VI, Part 4.

BRNO ....

(author's signature)

ACKNOWLEDGEMENT

I would like to express my gratitude to my special supervisor Olivier Amoignon, Ph.D. for his advises and for the fruitful discussions we have had together during my stay at the Swedish Defense Research Institute FOI. I would also like to thank to my special supervisor Ing. Robert Popela, Ph.D. for his guidance during my doctoral studies. My gratitude also belongs to Ing. Jan Navrátil for his assistance in solving numerous issues of optimizations. Finally, I would like to thank my wife Gabriela for her support and patience.

CONTENTS

1 Introduction **10**

1.1 Overview 10

1.2 Motivation and Goals 10

1.3 Thesis organization 11

2 Current state-of-the-art **12**

2.1 Introduction 12

2.2 Parameterization 13

2.3 Volume Mesh deformation techniques 17

3 Free-Form Deformation (FFD) **19**

3.1 Introduction 19

3.2 Theoretical background 19

3.3 FFD procedure: 24

3.3.1 Construction of parametric volume (Lattice of control points): 24

3.3.2 Embedding the object within the volume 26

3.3.3 Deformation of the parametric volume 27

3.3.4 Evaluating the effect of the deformation on the embedded object 27

3.4 FFD gradients 27

3.5 FFD geometry handling 29

3.5.1 FFD basic properties: 29

3.6 Impact of the NURBS degree 36

3.7 FFD in aerodynamic shape optimization - 2D test case 39

3.7.1 NACA 0012 airfoil optimization 39

4 Adaptive FFD parameterization with respect to geometry **47**

4.1 Introduction 47

4.2 Coordinates transformation using RBF 48

4.2.1 FFD-RBF parameterization procedure 48

4.2.2 Test case: Wing trailing edge fixation 51

4.3 FFD-RBF in aerodynamic shape optimization - 3D test cases 53

4.3.1 CRM wing 53

4.3.2 Passenger aircraft 58

4.3.3 Complex geometrical constraints handling: EV-55 Outback landing gear nacelle aerodynamic shape optimization 63

5 Adaptive FFD parameterization with respect to optimization 70
5.1 Introduction 70
5.2 Enrichment 70
5.2.1 Enrichment procedure 70
5.3 FFD Multi-grid 74
5.3.1 FFD Multi-grid procedure: 74

6 FFD for CFD mesh deformation 77
6.1 Introduction 77
6.2 Procedure: 77
6.3 Numerical experiments: FFD vs. Standard methods 80
6.3.1 Quality measure: 80
6.3.2 Quality evaluation plan: 81
6.3.3 2D meshes: 82
6.3.4 3D meshes: 85
6.3.5 Efficiency: CPU time and memory demands: 89
6.4 3D Aerodynamic shape optimization using FFD for CFD mesh deformation 93
6.4.1 Basic FFD 94
6.4.2 FFD with RBF coordinate transformation 97

7 Outcomes of the doctoral thesis 102
7.1 Free-Form Deformation (FFD) parameterization 102
7.2 Adaptive FFD parameterization with respect to geometry 102
7.3 Adaptive FFD parameterization with respect to optimization 103
7.4 FFD for CFD mesh deformations 103

8 Conclusions 105

Bibliography 106

List of symbols, physical constants and abbreviations 118

Publications of the author 121

List of appendices 122

A Appendix A 123
A.1 Influence of number of RBF centers in x, y and z directions on the fixation error 123

B Appendix B **125**

B.1 Deformed 2D CFD Euler meshes of NACA 0012 airfoil 125

B.2 Deformed 2D CFD RANS meshes of RAE 2822 airfoil 127

B.3 Deformed 3D CFD Euler meshes of highly swept wing 128

B.4 Deformed 3D CFD RANS CRM wing meshes 131

Curriculum Vitae **133**

1 INTRODUCTION

1.1 Overview

The doctoral thesis is focused on development of Free-Form Deformation[1] parameterization method for deformation of shapes and CFD grids, used in the environment of shape optimization as an advanced tool for aircraft aerodynamic design.

It is hard to imagine aerodynamic design of modern aircrafts without the use of CFD simulation methods. Their benefits are known for quite long time and they are widely used to supplement or even replace wind tunnel testing in aircraft design. As the progress of computer hardware power rapidly increases, it practically enables more and more detailed simulations to be performed. Availability of powerful computers is also one of the reasons for growing popularity of the aerodynamic shape optimization techniques which results in significant cost savings in design cycle. However, because of the complexity of aerodynamic design problems, numerical shape optimizations still remain expensive tasks[2]. Therefore advanced optimization strategies complemented with appropriately capable parameterization methods are needed.

Parameterization methods[3] work either with description of the geometry or with description of the deformations of the geometry. The important aspect being how do they perform on complex shape configurations while using high-fidelity analysis tools like CFD. A suitable parameterization should be effective, easy to implement and provide analytical sensitivity derivatives of the of the model with respect to optimization variables.

1.2 Motivation and Goals

The parameterization defines possible object shapes and shape changes which are used as design variables during the optimization process. The number of parameters has major influence on the computational time cost, Andreoli[4] emphasized that presence of a large number of design variables can result in problems with convergence for most existing optimization algorithms. A parameterization method intended for use in wide variety of aeronautical applications needs to be flexible enough to produce optimal geometry without requirement of too many design variables. Since the designer may have a limited a priori knowledge of the design space an adaptive parameterization method would help the optimization performance in general.

The FFD parameterization developed for computer graphics by Sederberg and Parry[1] is a method able to deform any object in any form of description and topology (a contrast to CAD model representation of an object). There are many variations of FFD parameterization developed to further enhance its abilities[5, 6].

The goal of the thesis is to accurately resolve problems of shape optimization including geometric constraints with the focus on adaptivity with respect to the geometric features because it is a difficulty for FFD[7], including the approach[8] that is being applied here.

The fact that the FFD parameterization can smoothly deform anything that is embedded within a lattice of control points[9] is a foundation for the second goal of the thesis. The development of FFD parameterization for both surface and CFD mesh deformations that brings simplification to the optimization process by using parameters of surface mesh description as optimization variables.

Two of the design problems[10, 11] proposed by the AIAA discussion group provide the basis for an evaluation of used algorithms of FFD parameterization. The tests cover airfoil and wing design, and involve several challenges for parameterization such as geometric constraints and the possibility to test the convergence of design spaces.

Goals

- The primary goal is to develop and verify FFD[1] parameterization method in the context of aircraft design. A method that could automatically adapt the parameterization and that would be able to handle complex geometry deformations and demands on complicated geometrical constraints.
- The secondary goal is to test the ability of FFD parameterization to deform CFD computational meshes.

1.3 Thesis organization

- Chapter 2 Current state-of-the-art
- Chapter 3 Free-Form Deformation (FFD) parameterization
- Chapter 4 Adaptive FFD parameterization with respect to geometry
- Chapter 5 Adaptive FFD parameterization with respect to optimization
- Chapter 6 FFD for CFD mesh deformation
- Chapter 7 Outcomes of the doctoral thesis
- The conclusions and are given in 8

2 CURRENT STATE-OF-THE-ART

2.1 Introduction

Current global situation highlights the role of shape optimization in the industrial aircraft design. Market competition forces the manufacturers to bring new and better designs in shortest time possible. It will be very hard if not impossible to fulfill the rising demands on development and operational cost reductions without the help of modern optimization tools.

It is a use of optimization methods that should make the aircraft aerodynamic design process more effective and to explore potential of novel aircrafts concepts. Aerodynamic shape optimization is able to expose the areas for improvement which may not be revealed nor by intuition neither by experience[12]. Optimization tools that are used for aerodynamic shape optimization in aeronautical applications can be divided into several groups. They differ in used optimization methods, parameterization methods and flow solution techniques. Each one is more or less time consuming to use and is suitable for different phase of the aircraft design. Among all of them genetic and evolutionary algorithms are now most widely used in preliminary design phase, followed by gradient-based methods and response surface methods. Thanks to recent implementation of adjoint sensitivity solution into some CFD solvers, the gradient-based optimizations are gaining popularity in technical praxis[13].

Flow solution methods are divided into several groups, which have their origin in historical development. From simple to complex, the solvers are: linear potential, nonlinear potential (adds nonlinearity), Eulerian (adds rotation) and Navier-Stokes solvers (adds viscous effects). As the complexity of the solver rises a more complex physics is taken into account. Consequences are rising computational cost and unfortunately also decreasing credibility of the results. Progressive Navier-Stokes solvers, such as DES, LES and ultimately DNS, are improving this credibility issues, but they calculation cost is, for the purpose of aerodynamic shape optimization, still prohibitive.

Optimization loop is typically formed by:

- *Shape definition* geometrical representation of the object or initial shape plus its change
- *Mesh generation* discretization of the geometry to computational mesh or modification of initial mesh
- *Flow solution* aerodynamic simulation of the flow field around the object (panel method, CFD)
- *Cost function evaluation* use of coefficients from flow solution to calculate the cost function value
- *Optimization* algorithm that is minimizing the cost function
- *Convergence?* formulation of convergence of the solution that decides if the optimization algorithm should make another loop or if the optimum solution has been found
- *Optimum* goal of the optimization process

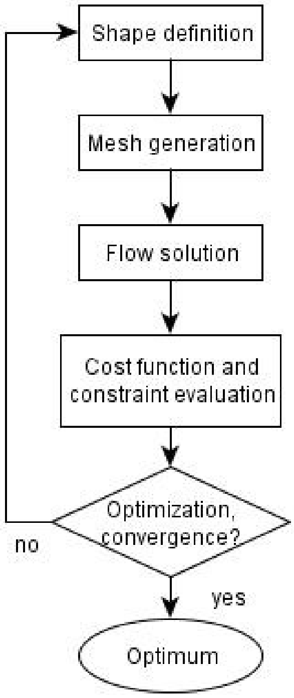

Shape definition
Mesh generation
Flow solution
Cost function and constraint evaluation
Optimization, convergence?
no
yes
Optimum

Fig. 2.1: Key processes of aerodynamic shape optimization

An example of the global optimization in conceptual design could be the combination of genetic optimization algorithm with potential flow solver. As was used by Ali and Behdinan[14]. Such approach can analyze many very different kinds of parameters and find the best solution relatively fast.

On the other side of design cycle, for fine-tuning of the details in local optimization we can use gradient based (adjoint) optimization method with FFD parameterization and Navier-Stokes flow solution. As used by e.g. Samareh[15]

2.2 Parameterization

Very important part of optimization process is the parametric description of the object geometry. Parameterization influences computational cost of the optimization as well as the quality of its product. Parameterization defines possible object shapes and shape changes by a set of parameters which are used as design variables during the optimization process. It is essential to use appropriate parameterization for each particular optimization task.

According to Samareh[3] the successful parameterization process must:

1. be automated
2. provide consistent geometry changes across all disciplines
3. provide sensitivity derivatives (preferably analytical)
4. fit into the product development cycle times
5. have a direct connection to the CAD system used for design
6. produce a compact and effective set of design variables for the solution time to be feasible.

Different parameterization methods use different amount of parameters for description of the object shape. The number of optimization parameters has major influence on the computational time cost. This stands for genetic and evolutionary methods as well as for RSM and gradient based optimizations. Exception is the adjoint approach for calculating the sensitivity gradients for the gradient-based optimization, where the computational time is not limited by the amount of parameters and can compute gradients of all parameters in a single adjoint calculation. Needless to say that not every kind of parameterization can provide analytical sensitivity derivatives and only those methods that can guarantee constant topology of the geometry (surface mesh) can use finite difference approach to calculate the sensitivity derivatives[16]. The direct parameterization method that uses as many design parameters as there are nodes in the surface mesh of the object is prone to problems with smoothness of the surface, caused by the surface gradients. A piecewise polynomial interpolations, such as B-splines, may be cause wiggles in the deformed shapes when using larger number of design parameters[17].

There are two basic parameterization tactics, parameterization of object shape and parameterization of object deformations. Shape parameterization will give us required (optimal) shape from scratch (e.g. wing planform). Parameterization of sharp edges, creases and other un-smooth profiles could cause difficulties. Parameterization of deformations will give us required (optimal) shape from some starting shape (reference wing). Parameterization of deformation over sharp edges, creases and other un-smooth profiles remains smooth.

The list of parameterization methods suitable for AERODYNAMIC SHAPE OPTIMIZATION is quite long. Samareh[3] presents detailed overview of parameterization techniques. Some methods were specially developed for parameterization of airfoils (e.g. PARSEC), they are often used even for wing parameterizations. Individual cross-sections of the wing are parameterized and interpolation of the geometry between them is used. This rather limits possible local shape modifications.

For complex 3D objects more sophisticated parameterization methods could be used.

Common types of parameterization methods that are currently used for aerodynamic shape optimization. They can be generally divided into two groups: Methods that represent the shape of the object and methods that deform existing shapes

Methods that represent the shape of the object

Closed formulation approach

Uses compact formulation for parameterization of shape (airfoil). Well known airfoils were created using NACA 4-series function, PARSEC method (see Fig. 2.2) developed for transonic airfoils by Sobieczky, describes the airfoil with smallest number of parameters possible.

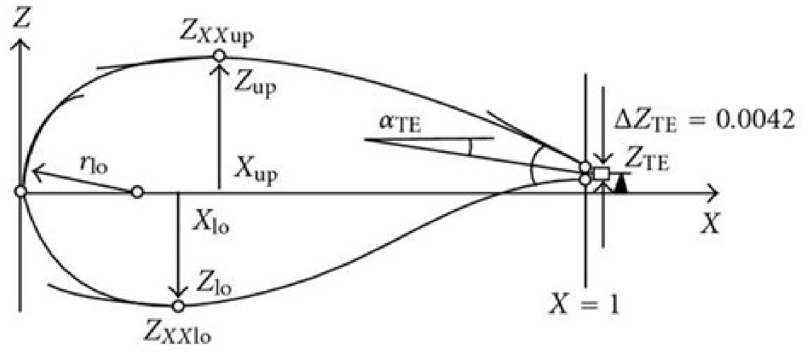


Fig. 2.2: Example of PARSEC parameterization

Polynomial and spline Using polynomial and spline parameterization of shape reduces number of parameters, often Bezier-Bernstein[18] and B-spline curves are used. Most universal is nonuniform rational B-spline (NURBS)[15, 19]. It was successfully used in 3D simple models, however complex 3D models need many curves and surfaces parameterization that results in large number of control points and can cause irregular or wavy geometry. Polynomial and spline parameterization is very often integrated into CAD description of the geometry, while the CAD software uses several methods of interpolation and does things like calculating intersection and etcetera. So the description of the geometry is complicated and therefor rather un-practical to have CAD software included in shape optimization loop.

Methods that deform existing shapes

Discrete approach
This very simple parameterization method uses boundary (surface) nodes as parameters. Starting from given mesh (geometry) it is useful for deformation parameterization. This approach was compared to the PARSEC and Hicks-Henne parameterization by Wu et al.[20] in the airfoil shape optimization task. Disadvantages of this method are huge number of parameters and incapability to maintain smooth (manufacturable) surface shape. So some smoothing algorithm must be implemented.

Analytical approach adds some shape function to baseline shape. For example Hicks-Henne used in[21, 22] and Chebyshev used by Carpentieri[23] basic functions can be used. These functions are smooth, so they cannot create sharp edges. This method is well suited for airfoil and wing parameterization.

Radial basic functions (RBF) Method modifies discretized surface in volumetric spaces with radial basic functions. It was used for wing parameterization[24] and is described by Amoignon[25]. Parameterized surface is smooth.

Free-Form Deformation (FFD) The FFD[1] embeds the object into parametric space built by lattice of control points and by modification of this lattice a deformation is passed on the object. The parametric space is usually represented by Bernstein, Bézier, B-Spline or NURBS.

The FFD parameterization, as an essential part of this thesis is fully described in dedicated chapter 3

2.3 Volume Mesh deformation techniques

Mesh deformation is used to adjust existing computational mesh to changes in geometry[26, 27]. Thanks to this procedure it is not necessary to create new mesh every time the geometry is changed and therefore significantly speed up the optimization process itself. The idea is to generate the computational mesh only once at the beginning.

Quality of the mesh after morphing has to be checked and has to remain in acceptable tolerance[16]. Especially in the case of large shape deformation some morphing methods may not be able to maintain good quality mesh and completely new mesh may need to be generated every time the tolerance is exceeded.

Mesh deformation techniques are mostly based on: spring analogy, Laplace equation methods or elliptic differential equation approach. Methods based on spring analogy are frequently used for their simplicity of implementation, on the other hand their lack of robustness makes them often un-practical. The techniques based on Laplace equation are most popular, thought they are effective only for small deformation. Mesh deformation using RBF is described by Jakobsson and Amoignon[28].

Major part of present mesh morphing methods first deforms the surface boundary mesh and after that they try to repair the interior volume mesh. This can be done by moving, adding, reconnecting or deleting mesh nodes. All of these activities except moving the nodes result in mesh topology changes. This precludes direct use of previous flow field calculation during the optimization. Instead some kind of results interpolation, has to be used. This of course slows down the optimization process. Most of the existing techniques particularly for unstructured mesh deformations are computationally expensive or mathematically complicated for practical use in optimization.

Nevertheless the elimination of mesh generation in every iteration is very compelling. For this reason, morphing techniques have been implemented in a number of commercial software codes. (ANSA Sculptor[13]).

Volume Mesh deformation characteristic case An example of mesh deformation technique is the work of Hsu, Chang and Samareh[29], in which they presented method based on linear elastic finite element analysis that they implemented into NASTRAN commercial FEM software. Proposed approach needs two steps in the finite element analysis. Firstly the mesh is deformed using homogenous material properties and after that re-deformed again, this time using inhomogeneous material properties.

Surface movements such as translation, rotation and cambering have been investigated. Two test cases were examined, 2D airfoil mesh and 3D aircraft mesh (see

Fig. 2.3). The deformed meshes show good quality verified visually and in the 2D case also by performed Navier-Stokes calculation. Great feature of this method is that connectivity of the elements during deformation stays intact and the deformed mesh has the same topology as the initial one. Deformation of 3D aircraft mesh of 855 727 tetrahedral elements took about 170 minutes on Intel Xeon 3.06 GHz, 3GB RAM computer.

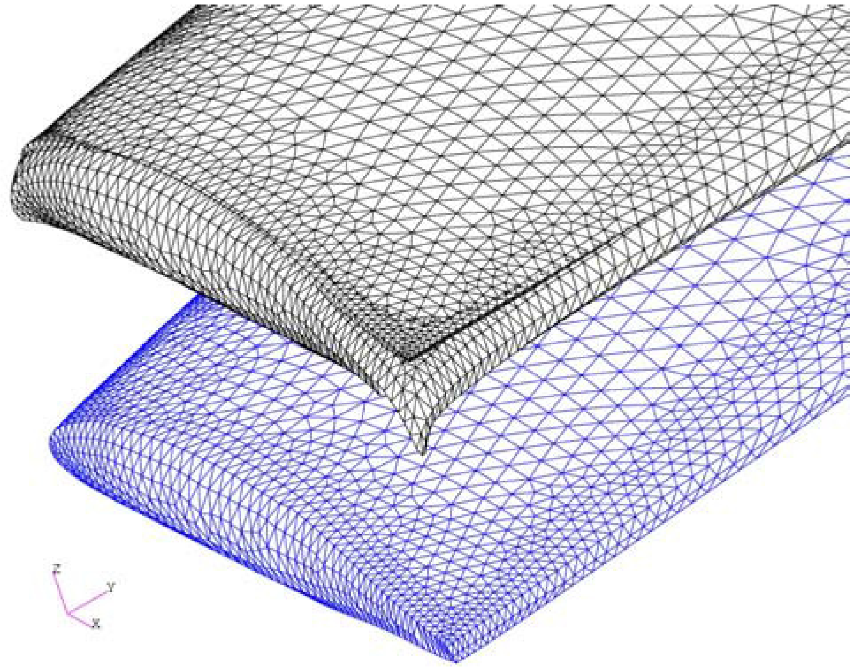


Fig. 2.3: Example of 3D deformed mesh (Source:[29])

3 FREE-FORM DEFORMATION (FFD)

3.1 Introduction

FFD parameterization method, an essential part of this work, is described here in detail. The FFD parameterization is rather complicated but also very powerful method. It was developed for computer graphics for morphing images (e.g. Boubekeur et al.[30]) and deforming models, first published by Sederberg and Parry[1]. It is usually linked with polynomial and spline parameterization techniques [15, 8, 31, 32, 31, 4, 33, 34]. It is ideal for parameterization of objects of high geometry complexity. FFD makes it possible to deform only part of the domain of interest while the rest of the geometry remains intact and the transition between deformed and undeformed parts is smooth. It belongs among the parameterization methods that deform existing shapes.

3.2 Theoretical background

The FFD algorithm embeds the model or models into parallelepiped lattice of control points and by modification of this lattice a deformation is passed on the model. The FFD treats the model as it is made of clear rubber that can be stretched, compressed, twisted, tapered or bent and yet preserves its topology. The FFD parameterization method can deform almost any type of geometrical model because its formulation is independent of the object's grid topology. It allows to deform truly arbitrary shapes with minimal set of variables. It can control surface continuity as well as volume preservation. The analytic sensitivities derivatives can be easily calculated for use in gradient-based optimization. The FFD can be used hierarchically to reach both local and global deformations.

One of the most important aspects that defines the FFD is the representation of parametric volume. Initially Bernstein[1, 35, 36] and Bézier[4, 32, 37, 38, 39, 40] polynomials, later B-Spline[41, 42, 36, 43, 44, 45, 34, 5, 46, 47, 48] and NURBS[7, 33, 15, 8, 49] were used. The NURBS offers the best capabilities of handling complex geometry, for which it has also become the backbone of CAD.

Because of all these advantages, the FFD is largely used in the field of geometric modeling[47, 50, 48], computer graphics[1, 35, 41, 43, 42, 45, 44, 51, 52, 53, 54, 55, 56, 57], and more recently in medicine[7, 58, 59, 50, 46] for image registration.

More importantly, the FFD has been used for aerodynamic shape optimizations of 2D[60] and 3D[61, 62] rotor blades, wings [33, 34, 4, 32, 37, 38, 39, 40, 15, 63, 64, 65, 62], concept[49], Blended-Wing-Body[12] and supersonic[33, 62, 66] aircrafts, elbow tube[67], sail[9], train[68] and car[5]. The capability of volume deformations makes the FFD suitable also for computational fluid dynamics grids deformations[49, 63, 9]. Further more the FFD can be conveniently used in aero-structural applications [33, 63, 9].

Use of the FFD parameterization method in either commercial software packages (ANSYS FLUENT, ANSA) or in open-source code SU^2[62] underlines its potential.

The main drawback of the FFD is the necessity of use of parallelepiped lattice of control points [35, 36, 33, 6]. The parallelepiped lattice makes it difficult to control some geometrical constraints [6] that are useful in optimization (fixed edges, angles of attack).

Various authors used such parametric volume representation (Bézier[38], B-spline[60]), that allowed them to have trapezoidal lattice instead of parallelepiped. The biggest obstacle of using arbitrary lattice of control points in combination with NURBS volume representation is the embedding of an object into such lattice, because NURBS based FFD is defined in parallelepiped lattices only [7]. In general it is not possible to say that the solution of embedding of an object into non-parallelepiped FFD NURBS lattice always exists.

The limitation caused by parallelepiped lattice was approached by Coquillart[35] with Extended FFD which introduced lattices of arbitrary shapes and their combinations. The difficulty with embedding step is solved by using Bézier representation of the parametric volume. Hsu, Hughes and Kaufmann[41] developed method called Direct Free-Form Deformation (DFFD), where the user directly manipulates the object points and the modification of the lattice of control points automatically computed by the modeling system.

MacCraken and Joy[42] published another variant of FFD in which they use lattice of control points of arbitrary topology in order to enable the desired deformations. Their technique uses an extension of the Catmull-Clark subdivision methodology to refine the 3D lattice. Ono et al.[52] introduced FFD parameterization method in which an automatic process hierarchically refines the initial bounding lattice to approximate the shape of the object. To achieve greater flexibility, Ilic and Fua[53] proposed Dirichlet FFD method, that place the control points into an arbitrary locations rather than on regular lattice. Kobayashi and Ootsubo[55] developed a variation of FFD called t-FFD to handle large-scale objects in more efficient way. t-FFD embeds the object into control mesh, which is constituted of a set of triangles with arbitrary topology and geometry.

Samareh[15] presented FFD method suitable for 3D aerodynamic shape optimization, which uses bivariate surface representation to reduce the number of design variables and to provide better control of surface shape changes. Song and Yang[56] published FFD with weighted T-spline, method that uses T-splines and T-junctions to adapt the lattice to objects with arbitrary topology or complex shape. McDonnell and Qin[57] developed interactive points-based FFD for polygonal meshes. They use ellipsoidal radial basis functions as parametric volume representation, which does not require explicit construction of the FFD lattice. Duvigneau[38] introduced approach that adapts the FFD parameterization to a particular aerodynamic shape optimization. The adaption principle stands on modification of the mapping (embedding) to minimize the ineffectiveness of the current parameterization. Sacharov, Surmann and Biermann[48] proposed another adaptive FFD method. In their approach the FFD lattice is automatically refined to decrease the approximation error during reverse engineering of the CAD/CAM data.

As suggested by Sederbeg and Parry[1], Lamousin[7] and later used by Kenway et al.[63], several adjacent FFD lattices can be constructed around the complex object of interest. The only problem of this approach is only C^0 continuity preservation on the boundaries between FFD lattices which limits its application.

Different parametric volume representations in FFD:

An important factor that influences the FFD abilities is the kind of parametric volume representation used.

Bézier curves: Bézier parameterization was developed for automotive components drawings in 1960s and is currently widely used for representation of the shape in computer graphics and geometric modeling. The curve is geometrically defined, so the parameters have geometric meaning.

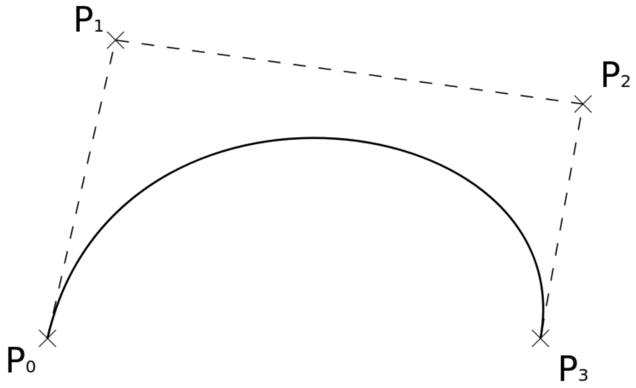


Fig. 3.1: Example of Bézier curve

advantages:

- Bézier parameterization is efficient and accurate representation for shape optimization of simple curves[3]

- Has degree elevation property, that means that it can increase the degree of the curve without changing the shape of the curve. This enables the use of hierarchy of embedded parameterizations
- Embedding is simple, only needs to solve linear equations

disadvantages:

- Bézier curves/patches can describe only smooth objects[4], for non-smooth objects they need to use very high order curves/patches (with danger of oscillation - higher degree increases round-off error - it is inefficient to compute high degree Bézier curve) or several curves/patches joined by some continuity condition C^0, C^1, use of several glued patches destroys the degree elevation property.

B-splines: B-splines are known since 19^{th} century, B-spline is an abbreviation of basis spline. It is a piecewise polynomial function.

advantages:

- Allow high degree of locality and flexibility - low degree B-spline can accurately and efficiently represent complex shapes[3].
- Guaranteed continuity when any of its control points are moved, in contrast to, for example, Bézier splines[41]

disadvantages:

- Cannot accurately represent implicit conic section shapes[3].

Nonuniform rational B-splines (NURBS): NURBS are special form of B-spline can accurately represent very complex shapes

advantages:

- In comparison with B-Splines[50]: NURBS can allow nonuniform distribution of control points and of the knot vector. Between moving control points and adjusting their weights, NURBS provide a much more flexible tool than uniform B-spline.
- Complex geometry handling: include weights as extra degree of freedom (virtually changes the stiffness of the rubber)
- Local nature of the deformations: changes in control point positions or weights affects only part of the object within the FFD lattice based on NURBS degree used.
- Smoothness[33]: a NURBS curve of the order p, having no multiple interior knots, is p-2 differentiable. As a result, the NURBS representation was able to handle a complex deformation and still maintain smooth surface curvature.

- NURBS formulation is the most general free-form surface representation[15].
- It is invariant under linear transformation. A NURBS curve of the order p, having no multiple interior knots, is p-2 differentiable. The approximation is local in nature.
- A NURBS curve is contained in the convex hull of its control points. The NURBS approximation is variation diminishing.
- Fundamental advantage of NURBS-based FFD[49] is that a given geometry can be parameterized to machine accuracy provided the inverse mapping search is tightly converged. This remains true whether a geometry is available in discrete form,such as in a surface triangulation, or in analytical form, such as with NURBS patches. Technically an entire wing can be twisted using only 8 control points, a difficult task for a B-spline surface parameterization.

For the purpose of aerodynamic shape optimization of practical aeronautical tasks we need parameterization that gives the optimization strong control over possible shape deformations. It seems that the best way to do that is to use FFD based on NURBS[7] and **develop a method that would resolve the biggest drawback of FFD parameterization and enable use of non-parallelepiped lattices adaptable to the shape of the object.** That is described in section 4.2 where a parameterization method is proposed in which the FFD is supplemented with RBF[28].

3.3 FFD procedure:

All the FFDs have the same basic procedure consisting of four main steps (Amoiralis[8]):

1. Construction of parametric volume (Lattice of control points)
2. Embedding the object within the volume
3. Deformation of the parametric volume
4. Evaluating the effect of the deformation on the embedded object

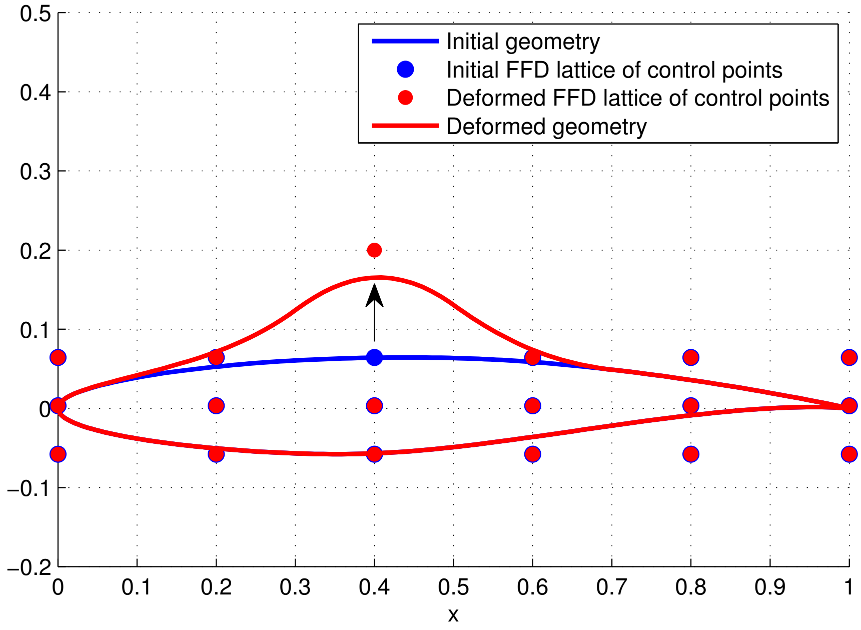


Fig. 3.2: Basic principle of the use of FFD parameterization for deformation

3.3.1 Construction of parametric volume (Lattice of control points):

A 1D, 2D or 3D lattice is constructed around/in the object that should be deformed. This defines parametric coordinate system.

NURBS definition

Nodes of the lattice are used as control points to define NURBS volume (plane) that contains the object to be deformed. NURBS polynomials are defined in each lattice direction u, v, w. Constraints of polynomial degrees:

$$1 \leq p \leq a, 1 \leq m \leq b, 1 \leq n \leq c \tag{3.1}$$

where *p,m,n* define degree of the basic polynomial function in corresponding direction, a+1, b+1, c+1 are numbers of the control points in each direction. NURBS uses knot vectors, where

$$\mathbf{U} = (u_0, u_1, ..., u_q), q = a + p + 1 \tag{3.2}$$

$$\mathbf{V} = (u_0, u_1, ..., u_r), r = b + m + 1 \tag{3.3}$$

$$\mathbf{W} = (u_0, u_1, ..., u_s), s = c + n + 1 \tag{3.4}$$

The equations are given just for x directions for now on, since the equations in other directions (dimensions) are formulated analogically. Values of U knot vector are calculated as

$$u_i = \begin{pmatrix} 0 & 0 \leq i \leq p \\ \\ i - p & p < i \leq (q - p - 1) \\ \\ q - 2p & (q - p - 1) < i \leq q \end{pmatrix} \tag{3.5}$$

and unified with range of x coordinates of parametric u coordinate. This knot vector has p multiple identical members at the beginning and at the end.

NURBS basic functions N are defined for every direction (u,v,w) of the parametric volume. N for u direction is calculated with standard recursive formula.

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3.6}$$

$$N_{i,0}(u) = \begin{pmatrix} 1 & u_i \leq u < u_{i+1} \\ \\ 0 & otherwise \end{pmatrix} \tag{3.7}$$

u is vector of Cartesian coordinates of geometry (points) that are to be embedded, i is position in knot vector and $u_{i..}$ are coordinates in knot vector.
The Cartesian coordinates of a geometry points within the 3D volume with parametric coordinates *u,v,w* are calculated using

$$R(u) = \frac{\Sigma_{i=0}^{a} \Sigma_{j=0}^{b} \Sigma_{k=0}^{c} G_{ijk}^{x} P_{ijk}^{x} N_{i,p}(u) N_{j,m}(u) N_{k,n}(u)}{\Sigma_{i=0}^{a} \Sigma_{j=0}^{b} \Sigma_{k=0}^{c} G_{ijk}^{x} N_{i,p}(u) N_{j,m}(u) N_{k,n}(u)} \tag{3.8}$$

for x direction, In general **R** are Cartesian coordinates of a point in a parametric space (u,v,w), $\mathbf{P}_{ijk}$ is a matrix of control points Cartesian coordinates (x,y,z) and $\mathbf{G}_{ijk}$ is matrix of its weights.
For 2D:

$$R(u) = \frac{\Sigma_{i=0}^{a}\Sigma_{j=0}^{b}G_{ij}^{x}P_{ij}^{x}N_{i,p}(u)N_{j,m}(u)}{\Sigma_{i=0}^{a}\Sigma_{j=0}^{b}G_{ij}^{x}N_{i,p}(u)N_{j,m}(u)} \tag{3.9}$$

For 1D:

$$R(u) = \frac{\Sigma_{i=0}^{a}G_{i}^{x}P_{i}^{x}N_{i,p}(u)}{\Sigma_{i=0}^{a}G_{i}^{x}N_{i,p}(u)} \tag{3.10}$$

Example: 1D vertical control point movement results in vertical geometry point movement, new y_f point coordinate is calculated:

$$y_f(v) = y_0(v) + \frac{\Sigma_{i=0}^{a}G_{i}^{y}P_{i}^{y}N_{i,p}(v)}{\Sigma_{i=0}^{a}G_{i}^{y}N_{i,p}(v)} \tag{3.11}$$

where y_0 is initial geometry y coordinate value and P_i^y is y coordinate of each control point.

3.3.2 Embedding the object within the volume

This step consist of identifying parametric coordinates that represents the object coordinates to be deformed. So an inverse problem needs to be solved in this step. That means to find such parametric coordinates u,v,w that their product $R(u,v,w)$ would be equal to $object(x,y,z)$ The form of R(u,v,w) of course depends on the parametric volume representation used.

- While using Bézier the problem can be simplified to the solution of three linear equations.
- B-spline representation generally requires numerical search technique such as Newton-Raphson method, but if the parametric and object coordinates are aligned, then thanks to the B-spline linear precision property the embedding operation vanishes [47, 48].
- In the NURBS parametric volume representation, due to the multiplicity of outer knots, the parametric coordinates have to be found by numerical search. The Octree algorithm[8], Golden section[7], Secant method or Newton-Raphson methods are often used. Numerical search can be very costly if the object's description is large (big matrix of coordinates).

Fortunately the embedding needs to be done only once at the beginning of the optimization.

3.3.3 Deformation of the parametric volume

In this step the lattice of control points is changed or/and the weights are modified, if not the weights have values of 1.

3.3.4 Evaluating the effect of the deformation on the embedded object

The deformed coordinates R are calculated using corresponding equation, for 3D 3.8.

3.4 FFD gradients

For the use of gradient-based optimization algorithms is necessary to derive the gradients of the FFD lattice control points that corresponds to adjoint sensitivities (gradients) on the object coordinates.

2D

for loop over every q^{th} of r object points:
change in FFD lattice control points $\mathbf{P}$ x coordinates results in change in x object coordinates

$$\Delta \mathbf{P}^x_q - > \Delta x^x_q \tag{3.12}$$

$$\delta R(u_q) = \frac{\Sigma^a_{i=0}\Sigma^b_{j=0} G^x_{ij} \delta P^x_{ij} N_{i,p}(u) N_{j,m}(u)}{\Sigma^a_{i=0}\Sigma^b_{j=0} G^x_{ij} N_{i,p}(u) N_{j,m}(u)} \tag{3.13}$$

for the adjoint sensitivities on the c_L
$\nabla c_L/_{\mathbf{P}} <=>$ for all $\delta \mathbf{P}$

$$\delta c_L = \nabla c^T_L/_{\mathbf{P}} \delta \mathbf{P} \tag{3.14}$$

$$\delta c_L = \nabla c^T_L/_{\mathbf{R}} \delta \mathbf{R} \tag{3.15}$$

for the adjoint sensitivities in x direction:

$$\delta c_L = \Sigma^r_{q=1} \frac{\delta c_L}{\delta x_q} \frac{\Sigma^a_{i=0}\Sigma^b_{j=0} G^x_{ij} \delta P^x_{ij} N_{i,p}(u) N_{j,m}(u)}{\Sigma^a_{i=0}\Sigma^b_{j=0} G^x_{ij} N_{i,p}(u) N_{j,m}(u)} \tag{3.16}$$

for one FFD lattice control point coordinate

$$\frac{\delta c_L}{\delta P^x_{ij}} = G^x_{ij} \Sigma^r_{q=1} \left(\frac{\Sigma^a_{i=0}\Sigma^b_{j=0} N_{i,p}(u) N_{j,m}(u)}{\Sigma^a_{i=0}\Sigma^b_{j=0} G^x_{ij} N_{i,p}(u) N_{j,m}(u)} \right) \frac{\delta c_L}{\delta x_q} \tag{3.17}$$

similarly for the adjoint sensitivities in y direction:

$$\frac{\delta c_L}{\delta P^y_{ij}} = G^y_{ij}\Sigma^r_{q=1}(\frac{\Sigma^a_{i=0}\Sigma^b_{j=0}N_{i,p}(v)N_{j,m}(v)}{\Sigma^a_{i=0}\Sigma^b_{j=0}G^y_{ij}N_{i,p}(v)N_{j,m}(v)})\frac{\delta c_L}{\delta y_q} \tag{3.18}$$

3D

The equations for 3D are derived analogically to 2D.

$$\frac{\delta c_L}{\delta P^x_{ijk}} = G^x_{ijk}\Sigma^r_{q=1}(\frac{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}N_{i,p}(u)N_{j,m}(u)N_{k,n}(u)}{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}G^x_{ijk}N_{i,p}(u)N_{j,m}(u)N_{k,n}(u)})\frac{\delta c_L}{\delta x_q} \tag{3.19}$$

$$\frac{\delta c_L}{\delta P^y_{ijk}} = G^y_{ijk}\Sigma^r_{q=1}(\frac{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}N_{i,p}(v)N_{j,m}(v)N_{k,n}(v)}{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}G^y_{ijk}N_{i,p}(v)N_{j,m}(v)N_{k,n}(v)})\frac{\delta c_L}{\delta y_q} \tag{3.20}$$

$$\frac{\delta c_L}{\delta P^z_{ijk}} = G^z_{ijk}\Sigma^r_{q=1}(\frac{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}N_{i,p}(w)N_{j,m}(w)N_{k,n}(w)}{\Sigma^a_{i=0}\Sigma^b_{j=0}\Sigma^c_{k=0}G^z_{ijk}N_{i,p}(w)N_{j,m}(w)N_{k,n}(w)})\frac{\delta c_L}{\delta z_q} \tag{3.21}$$

3.5 FFD geometry handling

There are two ways to deform the geometry with NURBS-based FFD parameterization:

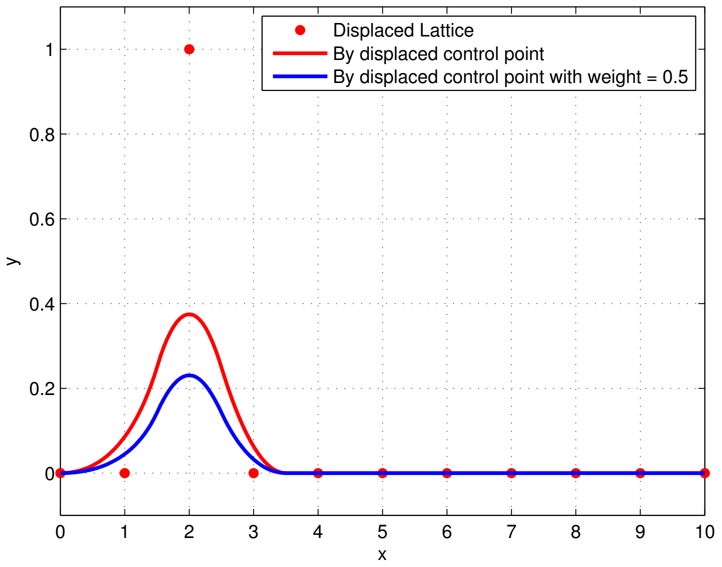


Fig. 3.3: 1D FFD two ways of deformation of a line

- Displacement of the control points is used solely in majority of cases. It means just to modify P_i member in corresponding equation 3.10.
- Modification of the weight of the control points. Can be used in combination with control point displacement to achieve even higher control of the deformation. It means to modify P_i and G_i members in corresponding equation 3.10.

3.5.1 FFD basic properties:

The FFD parameterization, as described in section 3.2 has certain qualities:

- Local control
- Global control
- Smoothness of the deformations
- Complex geometry handling
- Hierarchy of multiple FFDs

Local control

Using FFD, it is also possible to deform only part of the geometry that is embedded within the FFD lattice while the rest of the geometry remains intact and the transition between deformed part in the lattice and undeformed parts outside the lattice is smooth if the outer shell (layer) of the FFD remains fixed see Fig. 3.4.Lattice in smaller area of airfoil is created to enable modification of geometry. This shows the possibility to deform only part of interest and the rest of complete geometry remains intact. Changes in control point positions or weights affects only part of the object in the region:$(u_i, u_{i+p+1}), (v_j, v_{j+m+1}), (w_k, w_{k+n+1})$based on equation 3.2 inside the FFD lattice. So the size of the effected area is dependent on the discretization of the FFD lattice and NURBS degree see Fig. 3.8.

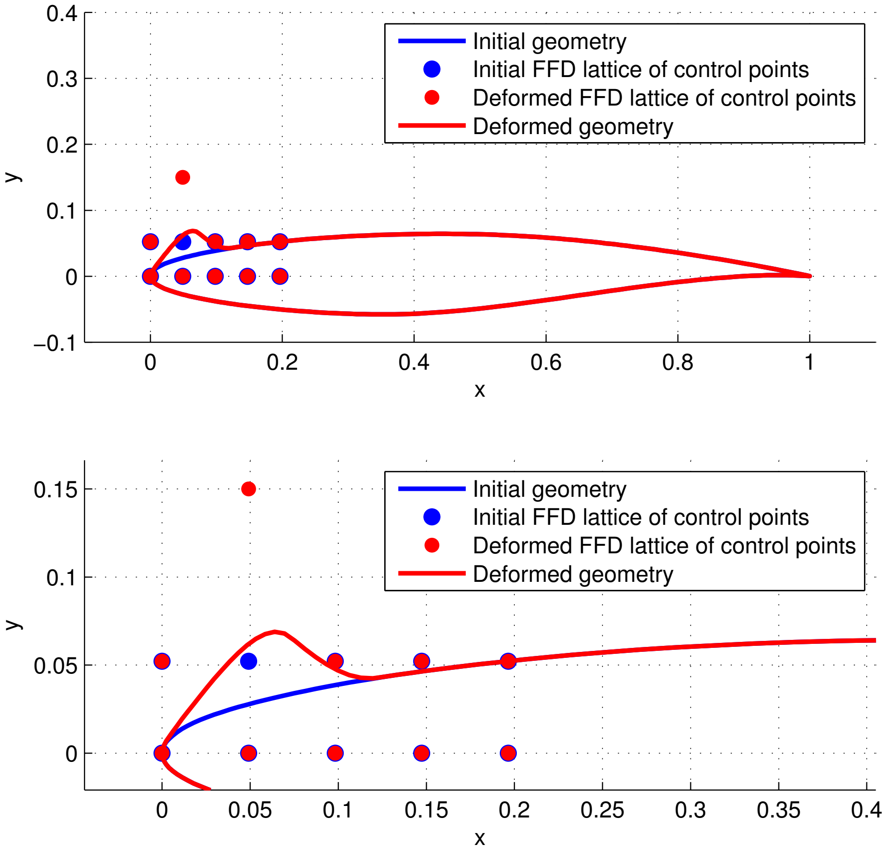


Fig. 3.4: Local deformation of airfoil using 2D FFD (Global and close-up view)

The Fig. 3.5 shows a FFD lattice built over a wing tip area and the deformation caused by displacement of some control points. That illustrates case for aerodynamic shape optimization of wing tip for which the flow solution of the whole wing is needed.

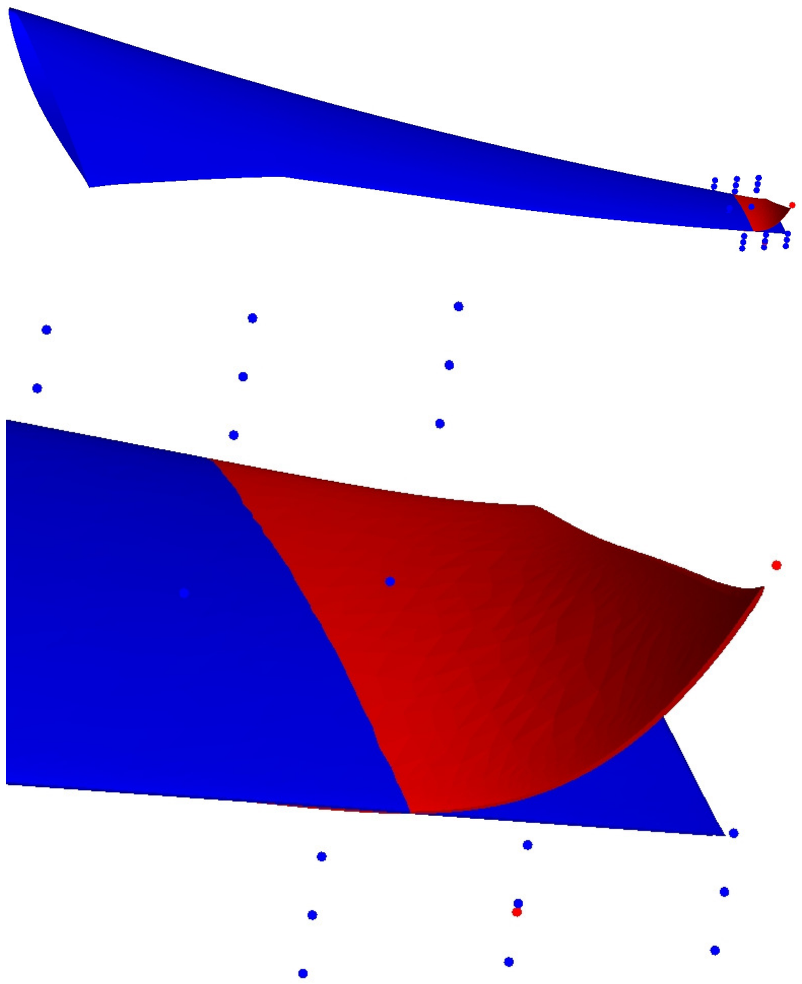


Fig. 3.5: Local deformation of wing tip using 3D FFD (Global and close-up view)

Global control

Everything that is embedded into the FFD lattice can be deformed. FFD allows to deform whole objects of arbitrary shapes with minimal set of variables see Fig. 3.6 and 3.7. Even one variable (lattice control point displacement in one direction) can control deformation of the whole object in one direction, see explanation in section 3.5.1. The minimal possible size of the FFD lattice (2 control points for each direction, that means 2 control points for 1D, 4 for 2D and 8 for 3D) can be often un-practical because it constraints possible NURBS degrees (see equation 3.1), which effects deformation smoothness. So in order to have 2^{nd} NURBS degree there is a need to have at least 3 control points in each direction.

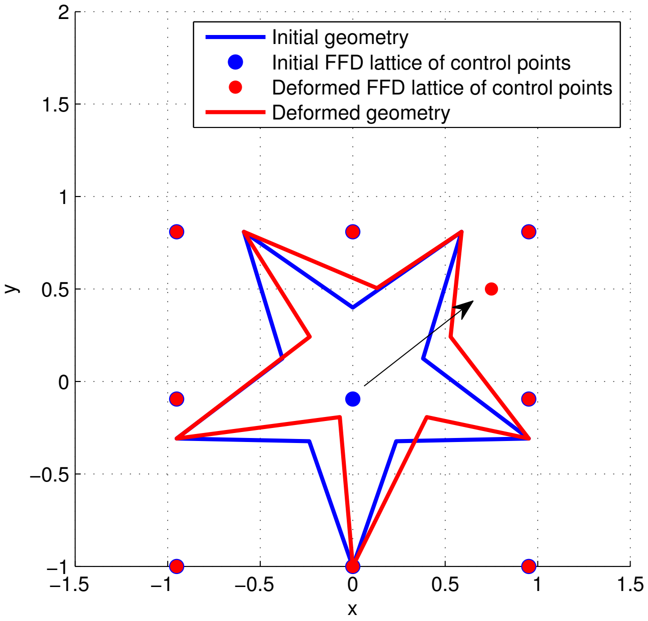
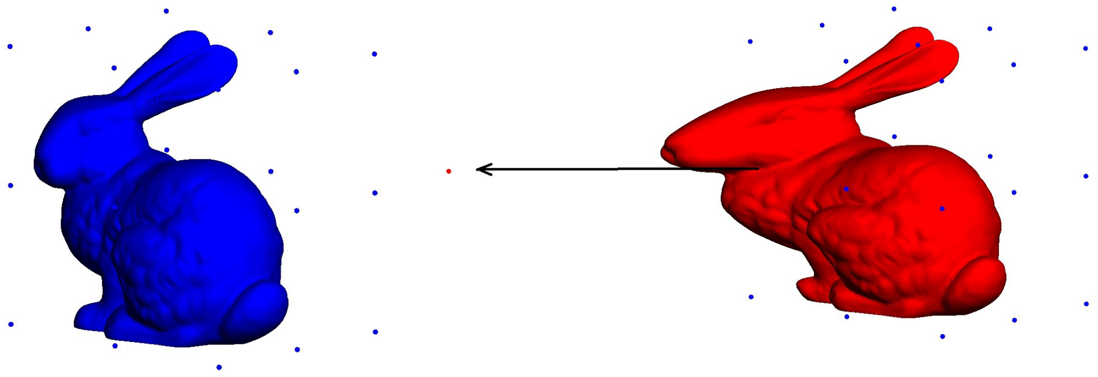


Fig. 3.6: Deformation of complicated 2D object with displacement of one FFD lattice control point



(a) Initial geometry

(b) Deformed geometry

Fig. 3.7: Deformation of complicated 3D object with displacement of one FFD lattice control point

Smoothness of deformations

The NURBS based FFD can ensure smoothness of deformations[33] by setting appropriate NURBS degree.

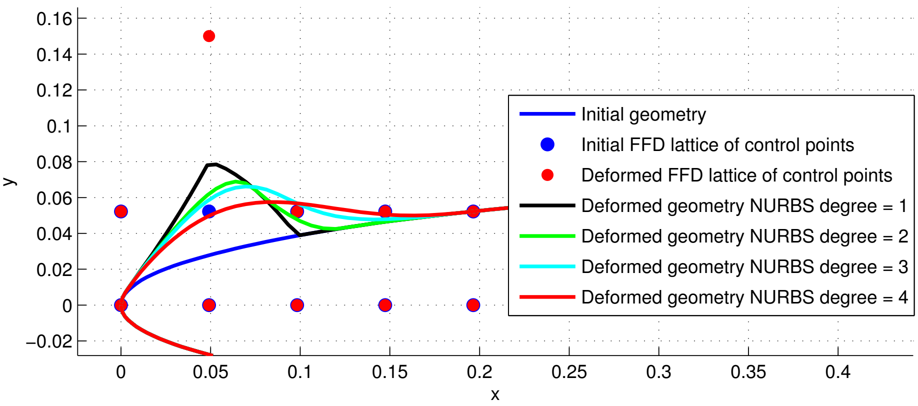


Fig. 3.8: NURBS degree influence on the deformed area

2^{nd} and higher degrees NURBS produce smooth deformation, obviously 1^{st} NURBS degree does not. Degree of NURBS defines degree of "shape", that is important for CFD analysis, because higher degree shapes have smooth derivatives that can contribute to better flow solution precision.

Complex geometry handling

This attribute gave the FFD name, where Free-Form deformation really means that the method is able to deform any object in any form of description and topology. It is possible to deform very complicated object with the FFD, but the level of control over the deformations is related to the number of control points used.

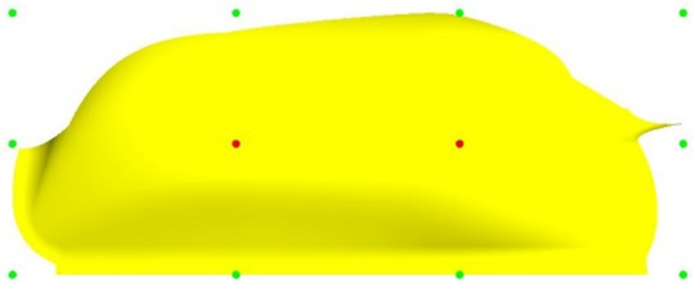


Fig. 3.9: FFD parameterization of complex 3D object

Hierarchy of multiple FFDs

A number of independent or dependent FFDs can be used hierarchically to deal with specific parameterization tasks, to reach both local and global deformations. The basic examples are:

- Several independent FFDs for parameterization of distinct areas. The deformation process is than completely independent.
- Several independent FFDs for parameterization of areas with some common parts of the geometry, so called overlapping FFD lattices (see Fig. 3.10). The deformation process is than usually driven by defined hierarchy. A simplified aircraft geometry is deformed using two FFD parameterizations. One for the wing and the other for fuselage, while their lattices are overlapped in the wing-fuselage region.

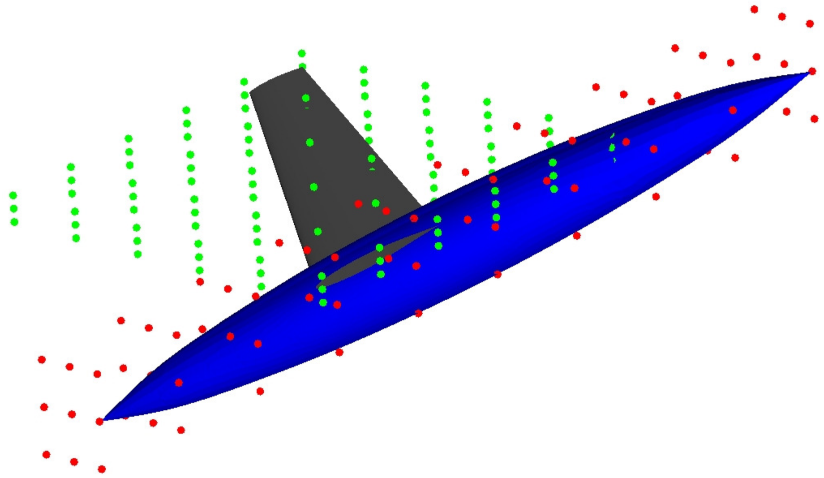


Fig. 3.10: Overlapping FFD lattices in 3D

- Several FFDs for parameterization of areas with some common parts of the geometry, where some FFD control points are defined as common for more FFDs and those control point displacements are the same (dependent FFDs).

- Hierarchy of independent or dependent FFDs in which one contains the other/s. Usually one FFD contains all of the geometry and drives the global deformations and the smaller FFDs takes care of local deformations in the areas of interest.
- Adjacent FFD lattices [1, 7, 63], where the outer faces of adjacent FFDs are shared. Two adjacent FFD lattices are constructed so that they share one outer line or column of control points in 2D or one outer plane of control points in 3D. The deformations caused by displacement of common control point will result in deformation in geometries embedded in FFD lattices. This approach can guarantee only C0 continuity[63]. As can be seen on Fig. 3.11 the airfoil is not deformed smoothly on the boundary between the two lattices, nevertheless the continuity is preserved.

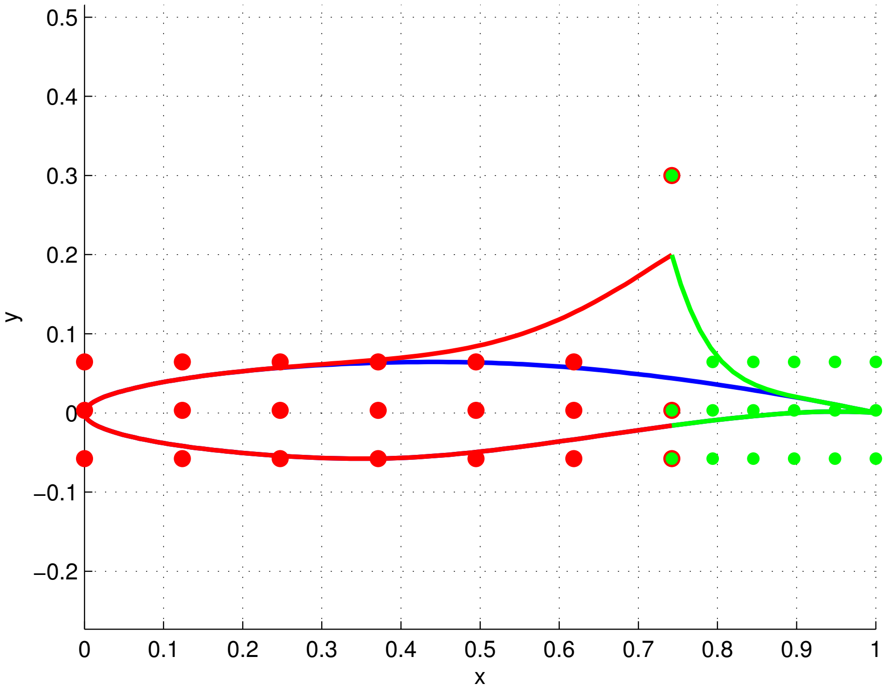


Fig. 3.11: Deformation of RAE 2822 airfoil with two adjacent FFD lattices

Most of this require special care to guaranty smooth transitions, if the outer shells of the FFD lattices are not fixed.

3.6 Impact of the NURBS degree

The influence of the NURBS degree on the regularity of shapes produced by optimization and its impact on the convergence speed of the optimization is studied in the following parts.

Behavior of the optimization with respect to the size of the FFD lattice was studied. Big parameterization can cause regularity issues such as wiggles. Wiggles created on the shape during the optimization can degrade mesh quality and prevent the flow solver to find a solution, slowing down gradient-based algorithms or eventually causing the optimization to stop. Wiggles can occur when increasing the number of design parameters even if parameterizations mostly produce smooth geometric changes. The reason is that the values of the parameters are driven by an optimization process trying to minimize a cost function[17], not to preserve the quality of a representation of the geometry.

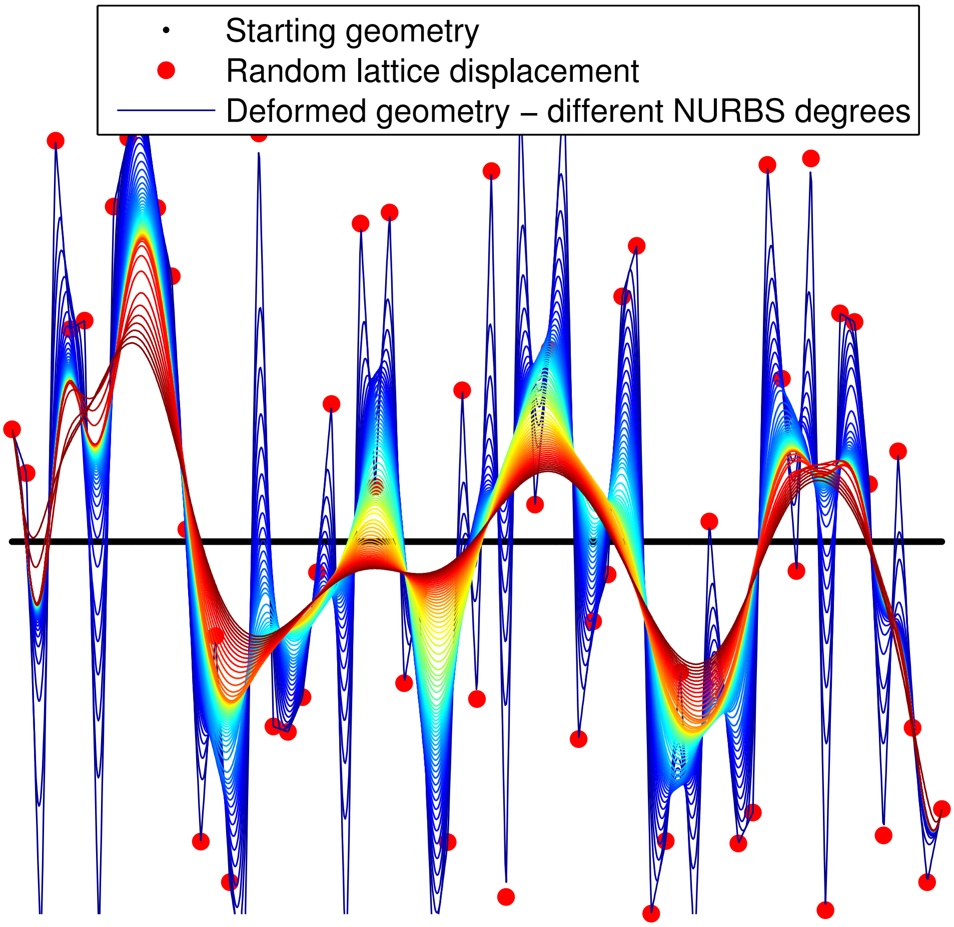


Fig. 3.12: Oscillation influenced by NURBS degree

In order to illustrate the influence of the NURBS degree on oscillations, the control points of a 2D FFD lattice deforming a straight line are randomly displaced. Fig. 3.12 shows the horizontal line deformed by FFD control points displacements using NURBS degree varying between 1 (blue) and 64 (red).

Oscillations are visibly damped by increasing the NURBS degree. Similar results would be observed in 3D latices because 2D or 3D FFD use products of (1D) NURBS defined in each direction (see Equ. 3.8).

The impact of the NURBS degree on the optimization is now illustrated by the resolution of two inverse geometric problems of design (see Fig. 3.13). Each problem is solved using several algorithms (Steepest descent, Conjugate gradient and Conjugate gradient with restart) in order to exhibit the major trends.

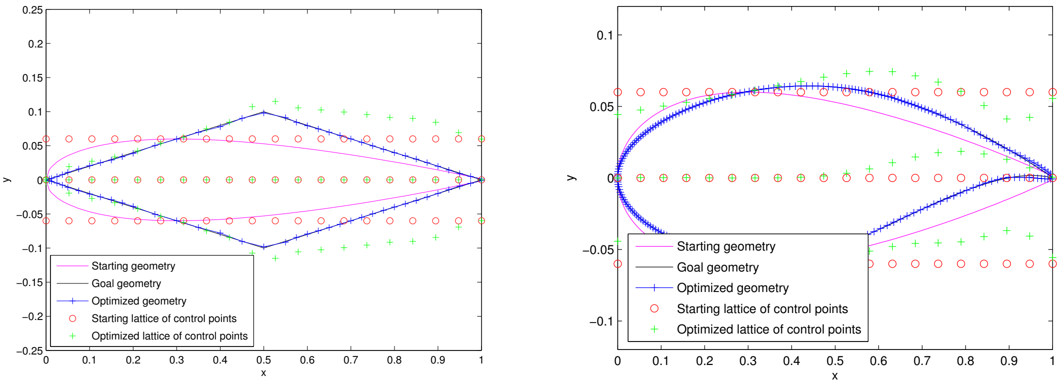


Fig. 3.13: Two examples of inverse design: towards a diamond-shaped airfoil (left) and towards a smooth airfoil (right).

In one case the objective is to obtain a diamond like airfoil from the NACA 0012 geometry and in the other case the targeted airfoil is the RAE 2822. In both cases the square of the difference of coordinates between the airfoil being deformed and the targeted shape is minimized.

Fig. 3.14 shows that the cost of the optimization rises when increasing the NURBS degree in the case where the target is the diamond shape. On the contrary, the cost of optimization is not affected by the NURBS degree when the initial and the target geometry, e.g. RAE2822, can be obtained by smooth shape transformations. This behavior of FFD may suggest that increasing the NURBS degree, which practically damps oscillations (see Fig. 3.12), will not affect the performance of the optimization, if we seek smooth shape solutions, which is the case in aeronautical applications.

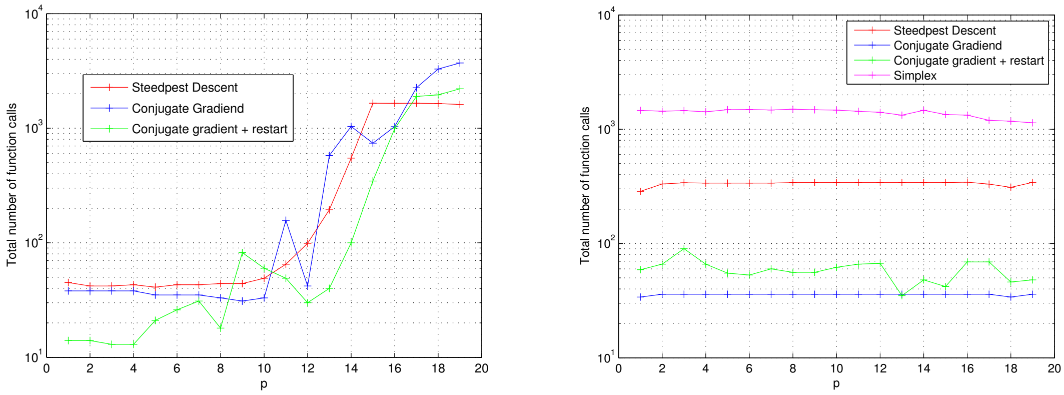


Fig. 3.14: Cost of optimization dependency on the NURBS degree - towards a diamond-shaped airfoil (left) and towards a smooth airfoil (right).

The influence of NURBS degree on aerodynamic shape optimization results is studied in section 3.7.1 for 2D and in section 4.3.3 for 3D.

3.7 FFD in aerodynamic shape optimization - 2D test case

An airfoil design case[69] proposed by the AIAA Discussion Group on Aerodynamic Design Optimization was proposed as an aerodynamic shape optimization benchmark case. As such it gives a practical testing platform for application of the FFD parameterization method for aerodynamic shape optimization.

3.7.1 NACA 0012 airfoil optimization

The case description was communicated by the AIAA Aerodynamic Design Optimization Discussion Group [10]. It consists in minimizing the drag of the symmetric NACA 0012 airfoil in inviscid flow at M=0.85 with geometric constraints.

$$\begin{array}{rrcll} & \min c_D & & & \\ \text{subject to:} & y\left(x\right) & \geq & y_{\text{NACA0012}}\left(x\right) & x \in [0,1] \end{array} \tag{3.22}$$

The optimizations are carried out by gradient-based algorithm, namely the Sequential Quadratic Programming (SQP) from NLOPT[70] software package.

Tab. 3.1: NACA 0012 grid dependency study

Nodes	c_L	c_D
2826	0.0000	0.0395
42556*	0.00096	0.0475
168464	0.0016	0.0484
670336	0.0018	0.0485

* grid used for optimization

A set of unstructured meshes was generated with IcemCFD. The results compared in Tab. 3.1 suggest that a grid independent solution requires more than 200,000 nodes. The results of optimization are later cross-checked using the same set of grids.

Geometry: Zero thickness trailing edge NACA 0012 is used here. Defined as:

$$y = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4) \tag{3.23}$$

where, $x \in [0,1]$. The zero thickness trailing edge is achieved through a modification of the x^4 coefficient.

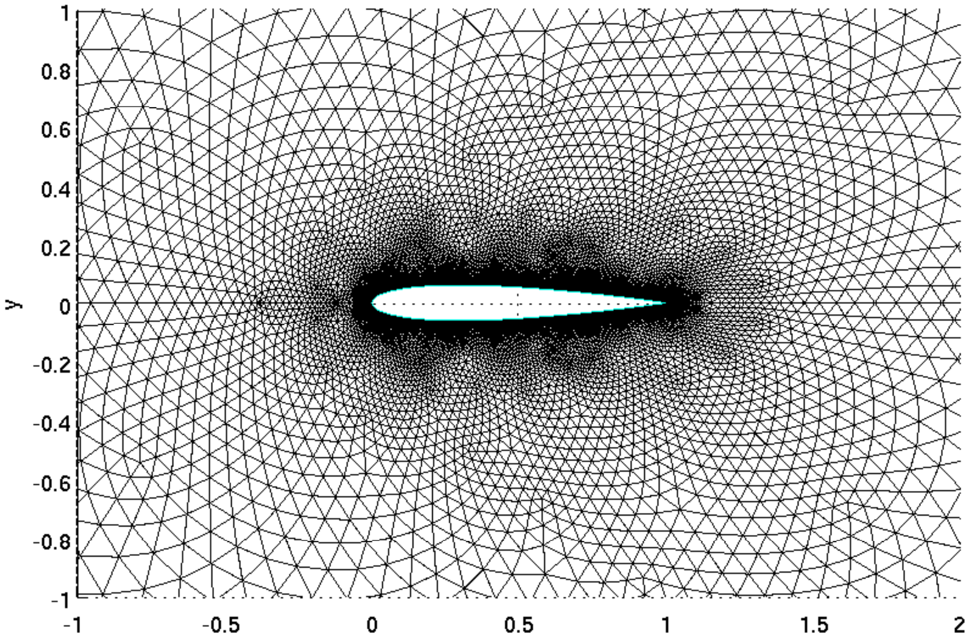

y
x: -1, -0.5, 0, 0.5, 1, 1.5, 2
y: -1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1

Fig. 3.15: NACA 0012 mesh with 42556 nodes

Mesh: Unstructured meshes were generated and their results compared.

Parameterization: 2D FFD lattice was constructed around the NACA 0012 airfoil geometry. For the purpose of optimization the movement of middle layer of FFD lattice control points was fixed, the upper layer control points displacements were used as optimization variables and the bottom layer displacements were mirroring the upper layer see Fig. 3.16

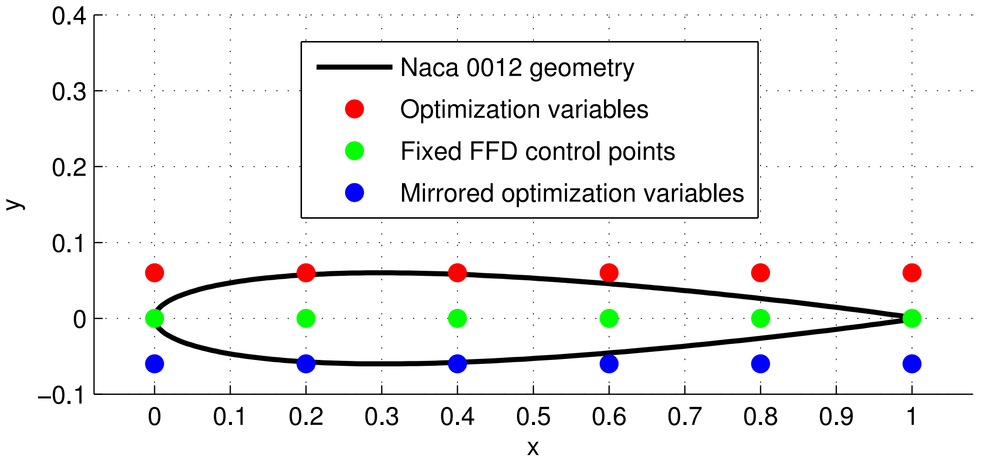


Fig. 3.16: Example of FFD parameterization setup for the case with 6 variables

Optimization results, effect of dimensionality: The study of Vassberg et al.[69] showed that this problem would be an excellent benchmark for parameterizations (in 2D) and optimization strategies because the non-trivial optimal shape seems to be unique at Mach number 0.85. The tests carried out with FFD show similar trends as shown in Tab. 3.2

Tab. 3.2: Results of NACA0012 optimization for different (FFD^b) lattices.

No.	$c_{D_{opt}}$	$c^a_{L_{opt}}$	cost^c
Baseline	0.04750	0.00096	1
3	0.03144	0.00125	23
6	0.02132	0.00690	32
11	0.01300	-0.02718	197
21	0.01187	0.00059	239
41	0.01138	0.00036	280

[a] CFD grid size is 42556 nodes.

[b] NURBS degree $p =$ No. $- 1$.

[c] Flow and adjoint solutions

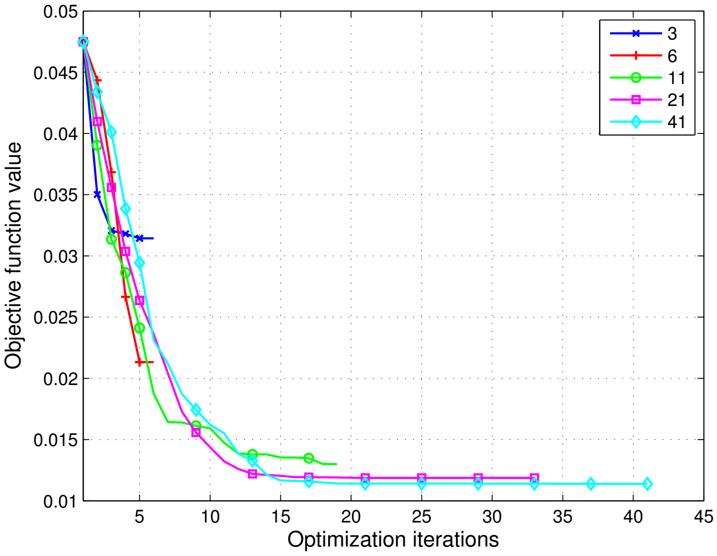


Fig. 3.17: NACA: History of optimization for various number of design variables (maximum NURBS degree is used in all cases). Only the feasible steps are shown.

As can be observed in Tab. 3.2 as much as 41 parameters are needed to get close to final converged solution (the difference between 21 and 41 parameters is only 4%), which correspond to the claim that the case requires close to 40 design parameters to be solved[69].

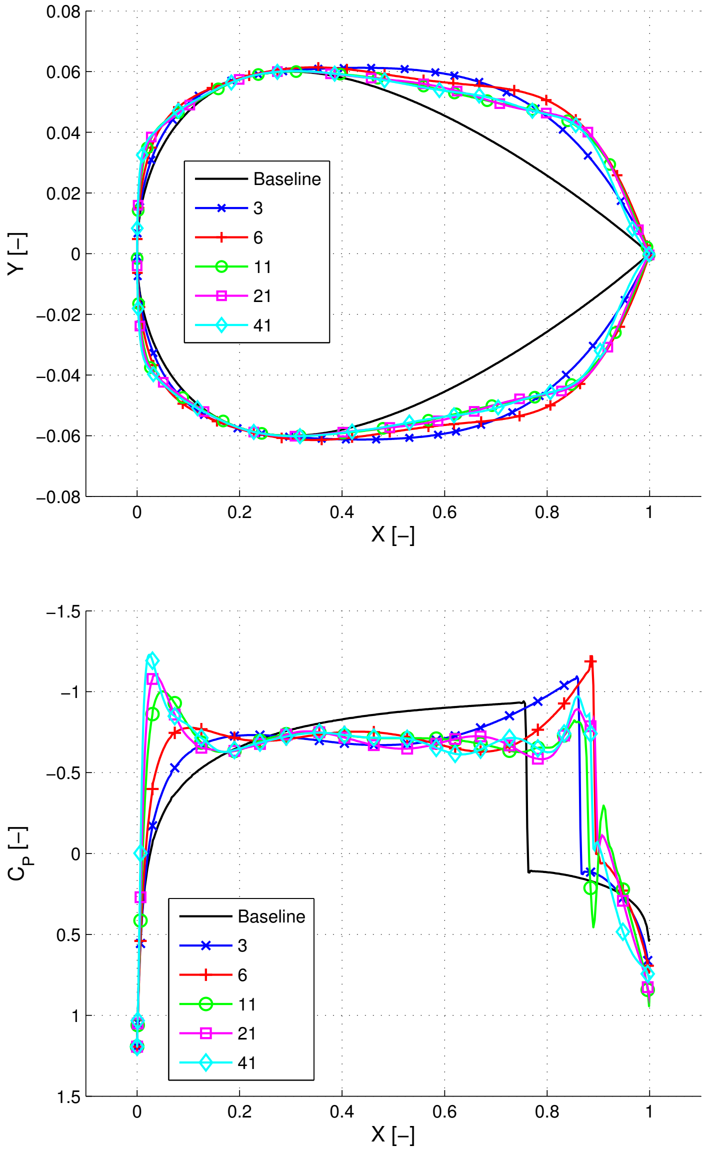


Fig. 3.18: NACA 0012: Comparison of optimal shapes for various number of design variables and distributions of pressure coefficients.

Mesh dependence analysis: Meshes from NACA 0012 grid dependency study (Tab. 3.1) are deformed based on the design obtained with FFD and 41 design parameters (see Tab. 3.4).

Tab. 3.3: Optimized NACA0012 grid dependency study (* grid used for optimization

Nodes	c_L	c_D
2826	-2.892e-4	0.0346
42556*	3.601e-4	0.0114
168464	1.008e-3	0.0119
670336	-4.721e-4	0.0119

* grid used for optimization

Tab. 3.4: Comparison of $c_{D_{opt}}$ for different initial design variables in NACA0012 optimization.

	$\mathbf{X}_0$	$\mathbf{X}_1$	$\mathbf{X}_2$
21	0.01187	0.01124	0.01408
41	0.01138	0.01049	0.01027

[a] CFD grid size is 42556 nodes.

[b] NURBS degree $p = \mathrm{N} - 1$.

[c] $\mathbf{X}_0 = \mathbf{0}$, $\mathbf{X}_1 = 0.25\mathbf{X}_{opt}$, $\mathbf{X}_2 = 0.75\mathbf{X}_{opt}$.

Initial design parameters effect: Further tests included different initial guess for the design parameters located between $\mathbf{X}_0 = \mathbf{0}$ and the solutions $\mathbf{X}_{opt}$ obtained with 21 or 41 parameters, see Tab. 3.4. It shows that the results are dependent on the starting point, a result that can depend on the optimizer NLOPT and needs to be investigated.

NURBS degree effect

This NACA 0012 test case gives practical application to illustrate the influence of the NURBS degree (discussed earlier in 3.6) using an FFD with 6 lattice points and increasing the NURBS degree from 2 to 5, the maximum for this lattice. For the mesh used here (42556 nodes) the baseline airfoil shows a drag of 475 drag counts.

Tab. 3.5: Influence of NURBS degree on the NACA 0012 minimum drag obtained with 6 FFD parameters

Degree	$c_{D_{opt}}$	cost CFD+adjoint
2	0.0247	44
3	0.0243	43
4	0.0223	34
5	0.0213	32

Tab. 3.5 indicates that for this particular case increasing the NURBS degree not only improved the cost function but also accelerated convergence (Fig. 3.19). The sole increase of the NURBS degree with 6 parameters of design gives here a gain of 10% compared to the maximum drag reduction (372 drag counts) that was obtained with a lattice of 41 points[71].

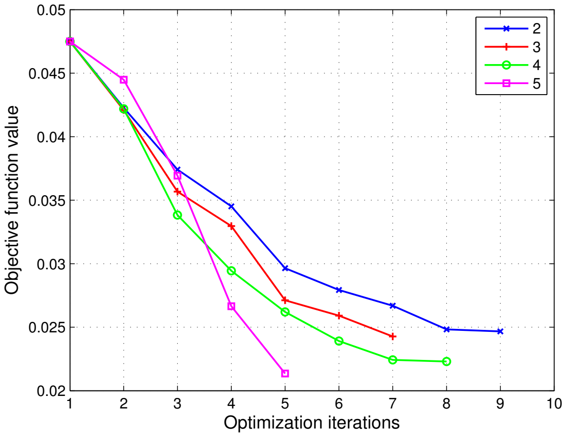


Fig. 3.19: NURBS degree influence on the optimization of the NACA0012 airfoil with 6 parameters.

FFD NURBS weights

As mentioned in section 3.5, the weights in NURBS-based FFD can be used as additional optimization variables. An optimization of NACA 0012 case with 3 control points displacements and their weights was performed.

Tab. 3.6: NACA0012 optimization with FFD using weights

No.	$c^{a}_{D_{opt}}$	cost[b]
Baseline	0.04760	1
3	0.03134	19
3 + 3 weights	0.03049	14
6	0.01711	42

[a] solution on 42556 nodes grid

[b] cost given as the sum of flow and adjoint computations.

Tab. 3.6 shows that adding the 3 weights parameters led to almost 3 % decrease in drag in comparison to just 3 control point displacements parameters. Adding 3 weights to 3 displacements gave 6 optimization parameters combined and if we compare the result to 6 control point displacements we see that using weights led to 78 % smaller decrease of drag. That means weights does not have the same power over control of deformations as displacements and that their use for general aerodynamic shape optimization cannot be recommended.

On the other hand the use of weights does not require any modification to the parameterization and thus can be a possible way how to further improve the optimizations in the cases where the FFD lattice cannot be altered.

Deformation in two directions

Displacements of control points in x direction were added to previously used y direction optimization variables to test influence of multi-direction deformations.

The results in Tab. 3.7 shows that the the additional x displacements parameters gave worse results compared to pure y displacements with the same number of parameters (3y + 3x vs. 6y etc.). The addition of x displacements parameters gave worse results even than only y displacements despite of using twice the number of optimization parameters, with the exception of 3y + 3x case which gave better results than 3y, which is probably caused by the fact that using only 3y displacements is not enaugh and that the additional 3x displacements are beneficial even if not very effective. Clearly the use of only y direction displacements is more effective in 2D airfoil aerodynamic shape optimization cases.

Tab. 3.7: NACA0012 optimization with FFD using displacements in x and y directions

$No.$	$c^a_{D_{opt}}$	cost[b]
Baseline	0.04760	1
3y	0.03134	19
3y + 3x	0.02366	16
6y	0.01711	42
6y + 6x	0.01749	13
11y	0.00641	144
11y + 11x	0.01412	15
21y	0.00471	95
21y + 21x	0.02103	12

[a] solution on 42556 nodes grid

[b] cost given as the sum of flow and adjoint computations.

As in the case of weights the use of x direction displacements additional to the y direction displacements does not require any modification to the parameterization and thus can be a possible way how to further improve the optimizations in the cases where the FFD lattice cannot be altered.

4 ADAPTIVE FFD PARAMETERIZATION WITH RESPECT TO GEOMETRY

4.1 Introduction

The purpose of this work is to develop a parameterization based on Free-Form Deformation[1] in the context of aircraft design. One of the goals is adaptivity with respect to the geometric features because it is a difficulty for FFD[7], including the NURBS-based approach[8] that is being applied here.

Practical aerodynamic shape optimizations often involves challenge in the form of complicated geometric constraints. One way of solving them is to add some penalty definition into the formulation of optimization cost function. That of course further stiffens the optimization process and can even lead to its failure. The other way is to have a parameterization that will be able to take care of some of the geometrical constraints, such as requirements of fixation of some part of the geometry (points, edges, sections). An example is to keep constant the trailing edge of a wing undergoing an optimization[11].

In aerodynamic shape optimization methods based on FFD have been applied to the design of rotor blades[60, 61], wings[33, 34, 4, 63, 64, 65], Blended-Wing-Body[12] and supersonic[33, 62, 66] aircraft design.

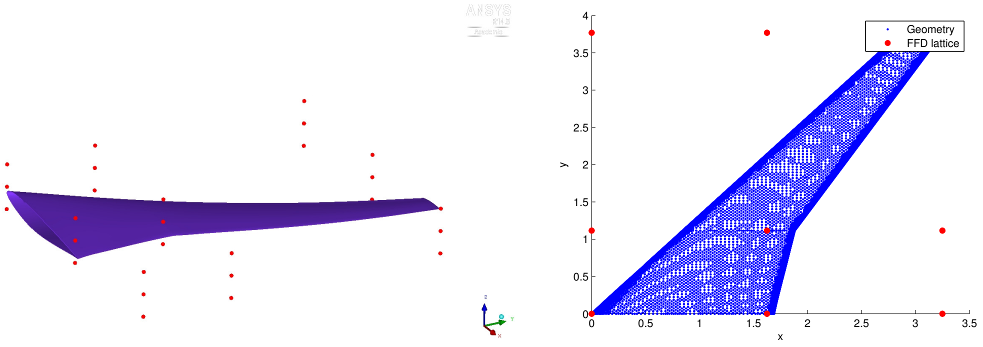


Fig. 4.1: A coarse FFD lattice.

The standard parallelepiped lattice of control points is not well suited for more complicated geometry handling. The solution of this disadvantage is given in following section as well as study of various improvements. The goal is to map the geometry into the standard parallelepiped lattice of control points in such way that the mapped geometry fills the lattice as much as possible. So the control points positions are close to the surface of the geometry, thus enable its better control.

4.2 Coordinates transformation using RBF

The FFD used here requires a parallelepiped lattice of control points[35, 36, 6]. Control of non-planar curves and other geometric constraints can thus become a difficult task[6]. This is the reason for using a Radial Basis Function (RBF) parameterization for coordinates transformation of the object, for example a wing or a highly cambered airfoil, that is parameterized by FFD-RBF: this transformation deforms the object that now "fills" the FFD lattice where embedding, an operation described below, and deformations are taking place.

4.2.1 FFD-RBF parameterization procedure

The FFD-RBF procedure consists of eight main steps:

1. Construction of FFD parametric volume (FFD lattice of control points)
2. Construction of RBF centers adapted to the object
3. Construction of artificial FFD lattice
4. Mapping of the object into the artificial FFD lattice
5. Embedding the mapped (transformed) object within the FFD parametric volume
6. Deformation of the parametric volume
7. Evaluating the effect of the deformation on the embedded object
8. Mapping the deformed object back into the real coordinates

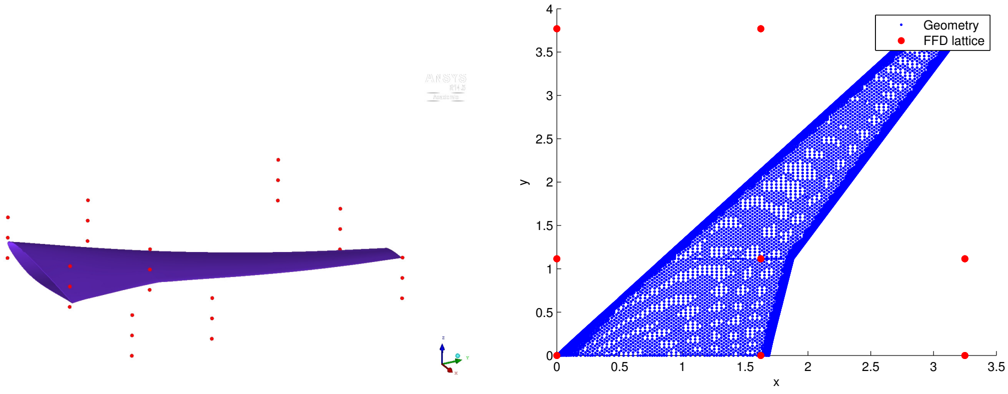
Construction of FFD parametric volume (FFD lattice of control points) This step is the same as in basic FFD parameterization (section 3.3.1).

Construction of RBF centers adapted to the object The location of the RBF centers (Fig. 4.3a), the usual term that designates the vertices of the RBF equivalent to the FFD lattice, need not to be the same as the FFD lattice as Fig. 4.2a. The RBF lattice is adapted to the objects (wing) geometry.

Construction of artificial FFD lattice Artificial FFD lattice is created just for the purpose of RBF coordinate transformation, this artificial FFD lattice has the same outer dimensions as the FFD lattice (the same box), and it has the same discretization (number of control points in all directions) as the RBF lattice (see Fig 4.4a).

Mapping of the object into the artificial FFD lattice RBF parameterization algorithm[28] is used to map the geometry into the standard parallelepiped lattice of control points. The mapping (transformation) matrix is based on difference between the locations of adapted RBF lattice centers 4.4b and the locations of articial FFD lattice 4.4a control points. This transformation matrix is then used to map the object into the FFD lattice.

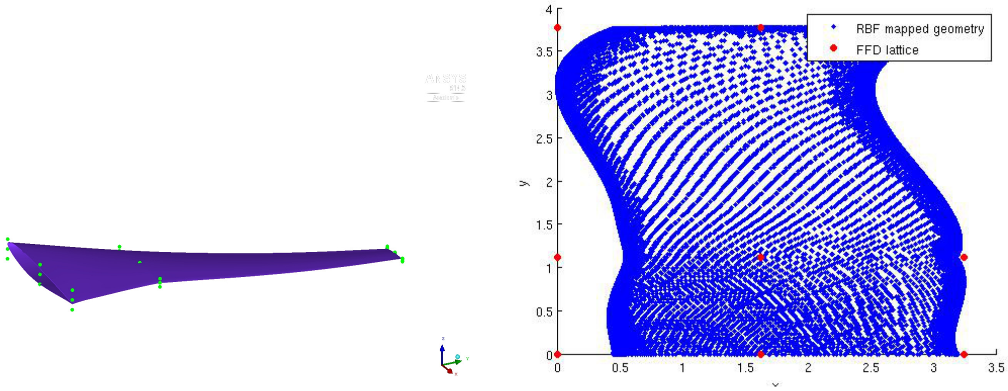
The wavy nature of RBF parameterization can cause problems in the sense that they map some points of the geometry outside the FFD lattice volume (see Fig. 4.3b), which is of-course unacceptable for the embedding step of the FFD. To solve this an iterative procedure was used, where the RBF control points nearest to the outside geometrical points are identified and their matrix of displacement modified until the complete geometry is well mapped inside the FFD lattice. This mapped geometry is then used by the FFD parameterization in the usual way.



(a) Iso view

(b) Top view

Fig. 4.2: FFD lattice constructed around wing

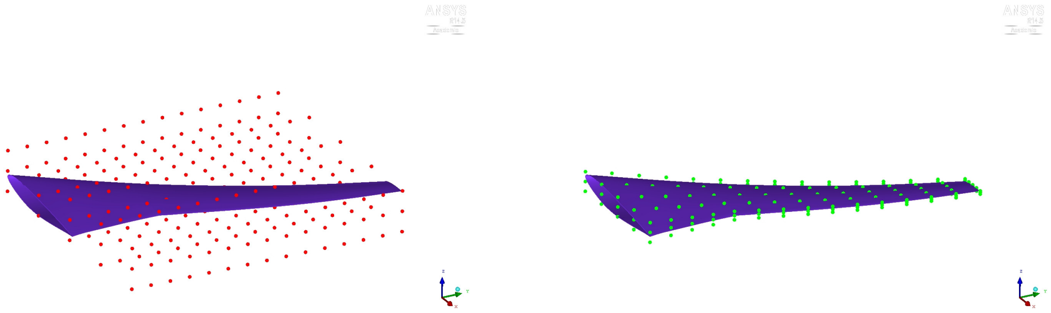


(a) RBF lattice constructed around wing

(b) Wing geometry mapped by RBF into the FFD lattice -top view

Fig. 4.3: RBF coordinate transformation example

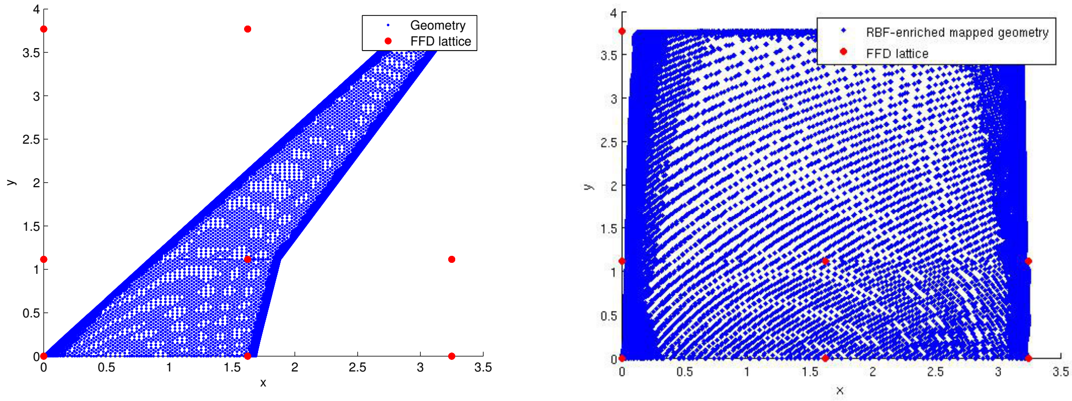
The RBF lattice discretization does not need to be the same as the FFD lattice discretization. So denser lattice can be used with the prospect of qualitative improvement of the mapping. Analysis of influence number of RBF centers is given in section 4.2.2.



(a) Artificial FFD lattice

(b) Dense RBF lattice

Fig. 4.4: Dense RBF coordinate transformation lattices



(a) Basic FFD parameterization of wing

(b) Wing geometry mapped by dense RBF into the FFD lattice

Fig. 4.5: Comparison of basic FFD parameterization and FFD with dense RBF coordinate transformation

Embedding the mapped (transformed) object within the FFD parametric volume This step is in the principle the same as in basic FFD parameterization (section 3.3.2).

Deformation of the parametric volume Again the same step as in basic FFD (section 3.3). In this step the lattice of control points is changed or/and the weights are modified.

Evaluating the effect of the deformation on the embedded object Also this step is the same as in basic FFD (section 3.3). The deformed coordinates R are calculated using corresponding equation, for 3D 3.8.

Mapping the deformed object back into the real coordinates The transformation matrix used to map the object into the FFD lattice is now utilized to map the deformed geometry back into the real coordinates.

4.2.2 Test case: Wing trailing edge fixation

This case comes from CRM wing optimization 4.3.1, which is described in following section. In this test case the trailing edge of the wing has to be fixed. That is especially challenging, because the wing is swept, cranked and twisted. Fulfillment of this geometrical constraint is quantified by fixation error, and tested up to unrealistically severe conditions (deformations). The tests summarized in Tab. 4.1 indicate deviations on the trailing edge when the FFD control points in the vertical plane adjacent to the trailing edge are fixed and all other control points are displaced vertically using random distributions. In principle this would forbid embedded points to move in that plane, but the coordinate transformation does not exactly place the trailing edge of the wing in this plane of the FFD lattice, causing errors.

Random deformations ran_{def} are defined as:

$$ran_{def} = \frac{random}{\|random\|_{\infty} ch} \tag{4.1}$$

where:

$$random \in [-1 : 1] \tag{4.2}$$

and

$$ch = 7.5\% \text{ of kink chord} \tag{4.3}$$

The fixation error is defined in % as max norm deformation on trailing edge divided by maximum FFD deformation:

$$Error = \frac{max(|\Delta z_1|, |\Delta z_2|, ..., |\Delta z_n|)}{max(|\Delta P_{ij1}|, |\Delta P_{ij2}|, ..., |\Delta P_{ijk}|)} * 100 \tag{4.4}$$

Where Δz_1 is deformation of observed curve point and ΔP_{ij1} is deformation of FFD lattice control point.

The results, shown in the Tab. 4.1, are the averaged values of 20 random deformations (normalized to have the same maximum amplitude) for RBF mapping of increasing sizes. Note that the number of RBF centers effects the RBF mapping abilities, a general conclusion can be stated that with increasing number of RBF centers the fixation error decreases. Dense RBF adaption lattice (see Fig. 4.4b) is constructed around the CRM wing using sections in span-wise direction, in the direction of minimal changes of the geometry. The denser the discretization of the RBF lattice is in this direction the smaller the fixation error. See section 4.3.3 that is giving results of deeper investigation of this phenomena.

Tab. 4.1: Geometric constraint test: averaged errors on the fixation of the trailing edge under FFD[1] deformation.

No. of RBF centers	RBF lattice dimensions	Error [%]
0[2]	-	39.92
27	3x3x3	19.36
64	4x4x4	15.40
225	5x15x3	2.60
384	6x16x4	1.87

[1] a = b = c = p = m = n = 2, N = 18

[2] no RBF coordinate transformation

4.3 FFD-RBF in aerodynamic shape optimization - 3D test cases

It is essential to verify the FFD in 3D aeronautical applications, to evaluate potential of FFD parameterizations with RBF coordinate transformation and identify its possible limitations. To investigate its ability to handle complex geometry deformations and demands on complicated geometrical constraints. Three major test cases were selected for this demonstration. Aerodynamic shape optimization of CRM wing which is a testing platform for evaluation of CFD software in drag prediction workshops, transonic passenger aircraft wing optimization and aerodynamic shape optimization of commuter aircrafts landing gear nacelle which was suggested by Evektor company.

4.3.1 CRM wing

The case [11], suggested by the AIAA Aerodynamic Design Optimization Discussion Group, concerns the optimization of a transonic wing in viscous flow is an excellent testbed for testing FFDs many properties.

The test case was designed to be as close to real wing for the passenger aircraft as possible and is quite restrictive. The use of FFD parameterization with RBF coordinate transformation in this test case was published in the AIAA SciTech 2014 by Amoignon, Hradil and Navratil[71], which also contains relevant mesh dependency study.

Geometry: The geometry specification is given by the AIAA Aerodynamic Design Optimization Discussion Group. It is based on NASAs Common Research Model (CRM) wing that was and in some modifications still is used in AIAA CFD Drag Prediction Workshops.

Parameterization: Developed FFD-RBF parameterization is compared to basic FFD parameterization, both use the same FFD lattice. The basic FFD parameterization of wing is presented in Fig. 4.7a and the FFD-RBF in Fig. 4.7b The FFD lattice has 9, 9, 3 control points in x, y, z directions. In total 243, from which 1 is fixed in order to eliminate possible shift of the whole wing geometry. Maximal possible NURBS degree is used in all three directions.

Mesh and CFD setup: Unstructured meshes consisting of tetrahedral elements were generated in ANSYS IcemCfd meshing software. Relatively coarse mesh (854184 nodes was used). The Edge[72] CFD solver was used for simulation of inviscid

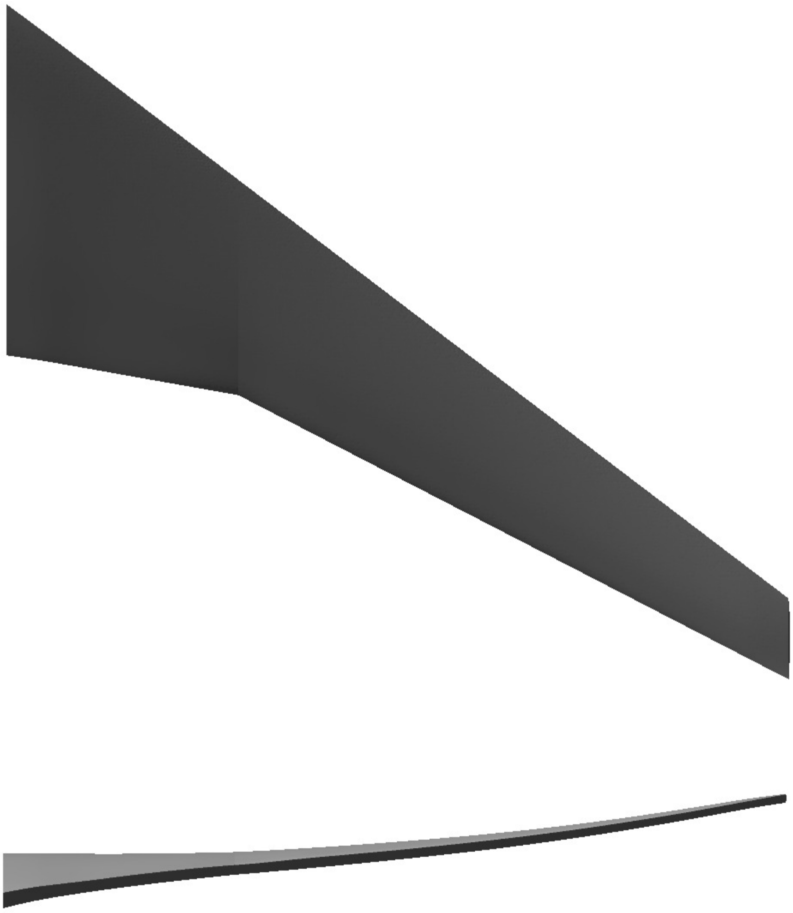


Fig. 4.6: Geometry of the CRM wing, top and back view

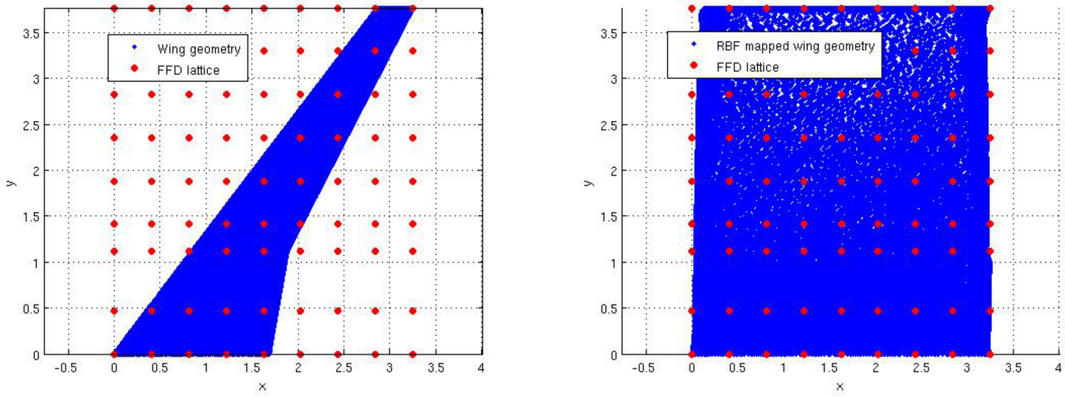
M=0.88 flow. The calculations were done on 2 Intel Xeon E5-2690 processors having 16 cores in total.

Optimization Some of the constraints from the original case [11] were relaxed in order to untie the optimization algorithm to obtain bigger improvement in the cost function value. That would give clearer view of influence of different aspects of the parameterization. Moment and volume constraints were removed as well as fixation of trailing edge, and the equality lift constrained was changed to inequality.

Optimization setup:

$$\begin{aligned} &\min c_D \\ s.t.:\quad & c_L \geq 0.5 \\ & t\left(y\right) \geq 0.25\, t_{\mathrm{CRM}}\left(y\right), \quad \text{for all span-wise positions}\, y \end{aligned} \tag{4.5}$$

The optimizations are carried out by gradient-based algorithm, namely the Sequential Quadratic Programming (SQP) from NLPQLP[73] software package, the convergence (stopping) criteria are:



(a) Basic FFD

(b) FFD-RBF

Fig. 4.7: Comparison of parameterizations of CRM wing geometry

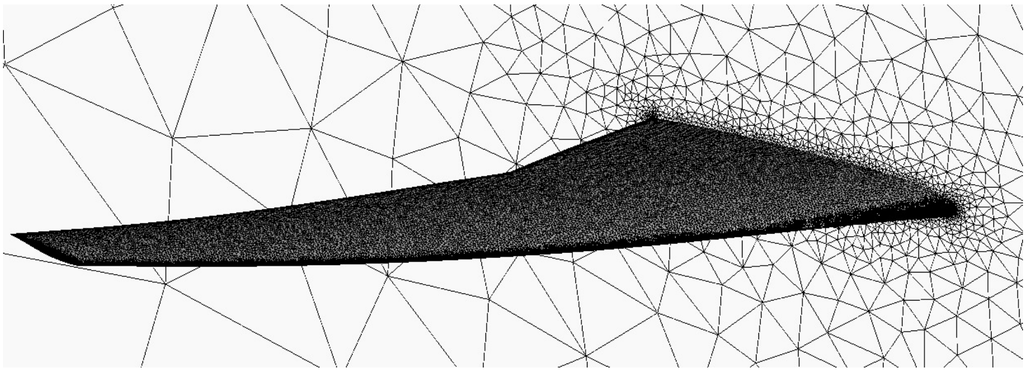


Fig. 4.8: Mesh of the CRM wing

- desired final accuracy (relative difference between last two iterations) = 1e-5
- maximum number of iterations (number of gradient calls) = 50
- maximum number of function calls during the line search = 10

The gradients were obtained from adjoint solution calculated in Edge program.

The CFD mesh deformations are done by standard Laplace method also in the program Edge, which adjusts the CFD grid to the deformed surface grid. In the case that the CFD mesh deformation fails and thus no CFD solution is obtained resulting in no CF value. The NLPQLP optimizer then halves the step size in the line-search part until a valid mesh is obtained from the meshdeform (CF is obtained) or stopping criteria is reached.

Results The Tab. 4.2 shows results of two optimization cases. The first uses basic FFD parameterization, the second uses RBF coordinate transformation to map the wing geometry into the FFD lattice. The RBF mapping procedure gave

Tab. 4.2: Comparison of CRM wing optimizations with different FFD parameterizations

	FFD	FFD-RBF
c_D baseline	0.017973	0.017973
c_D optimal	0.015079	0.012874
c_D reduction	16.1 %	28.4 %
Cost in CFD+adjoint iterations	27	43
Cost in CPU time	21 783	37 579
Cost in real time	3h 47min	6h 31min

approximately 12.3% better reduction in drag. That is caused by better control of the parameterization method over the shape deformations, since more control points are closer to the surface.

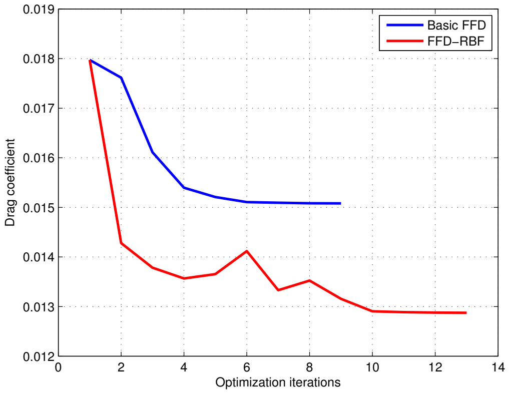
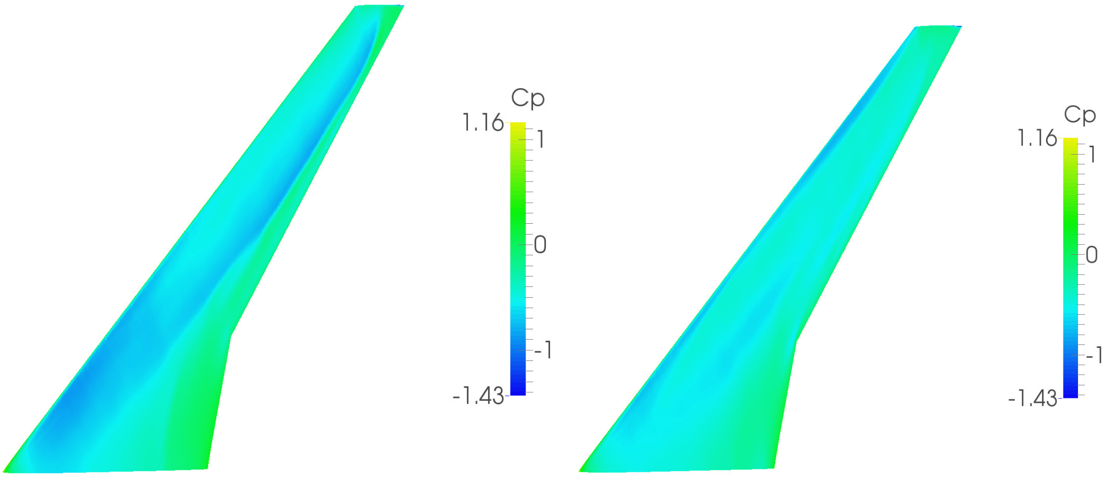


Fig. 4.9: Comparison of optimization history, using basic FFD and FFD-RBF parameterizations

The improved control over the deformations is probably behind the steep fall of the drag coefficient in Fig. 4.9.

Fig. 4.10 shows comparison of resulting pressure coefficient distributions of basic FFD and FFD-RBF optimizations. Note that the basic FFD was not able to suppress shock waves as good as the FFD-RBF.

Fig. 4.11 shows comparison of wing section shapes of initial wing geometry and wing optimized with basic FFD and FFD-RBF optimizations.



(a) Basic FFD

(b) FFD-RBF

Fig. 4.10: Comparison of pressure coefficient distribution on CRM wing

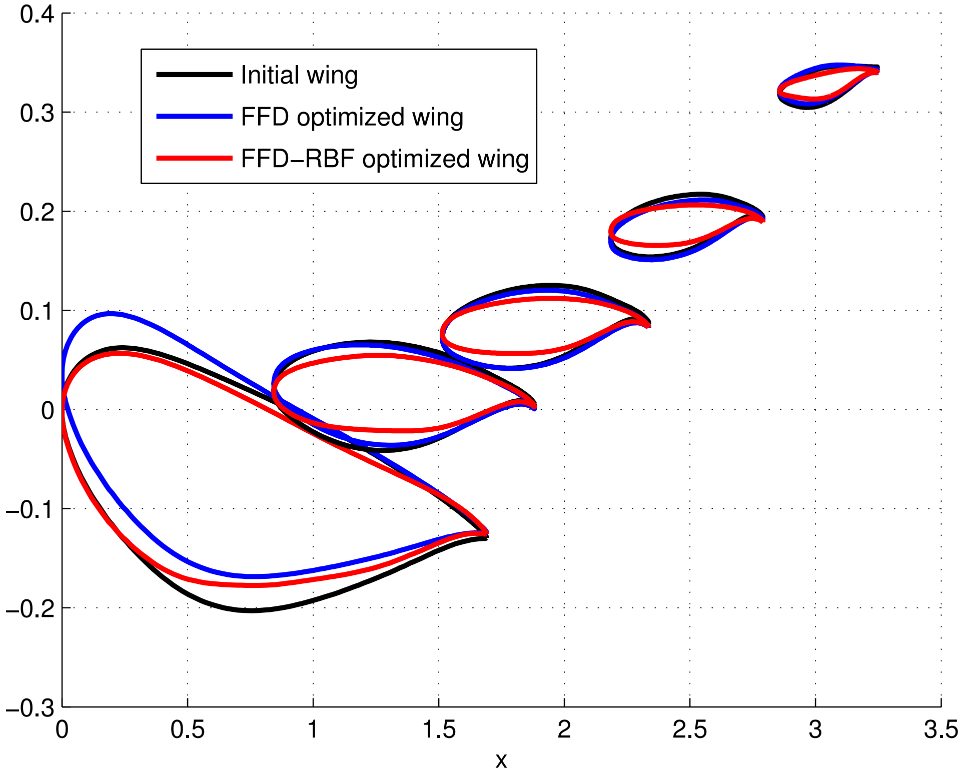


Fig. 4.11: CRM wing section shapes (axis not in scale)

Both optimizations ended by reaching maximum number of function calls during the line search and that was caused be ever failing CFD mesh deformation procedure. This issue is later addressed in chapter 6 by using FFD also for CFD mesh deformation.

4.3.2 Passenger aircraft

The issues with CRM wing CFD mesh deformation were a reason to continue on testing the FFD parameterization behavior on another test case. So for analysis of dimensionality of the optimization and NURBS degree influence on the optimization results a series of test were performed on transonic passenger aircraft. The aircrafts geometry comes from SAAB within a Clean-Sky project called OPTLAM, so all the results and plots are consider as confidential and therefore are presented in relative values and without scales. The goal is to optimize wing shape in order to decrease drag coefficient of the whole aircraft.

Optimization Original optimization goal and constraints are:

$$\begin{array}{rlcl} & \min c_D & & \\ s.t.: & c_L & = & c_{L0} \\ & Vo & \geq & Vo_{\text{Initial}} \\ & & \text{Fixed} & \text{planform shape (only vertical movement allowed)} \end{array} \tag{4.6}$$

The optimization variables are the FFD lattice control point displacements. The optimizations are carried out by gradient-based algorithm, namely the Sequential Quadratic Programming (SQP) from NLPQLP[73] software package. The gradients were obtained from adjoint solution calculated in Edge program. The CFD volume mesh was deformed using Laplace mesh deformation tool in Edge program.

Geometry: The geometry consists of fuselage and wing. The area for optimization is a major part of wing geometry, from the first "root" planar section till the wing tip.

Parameterization: The FFD-RBF parameterization was used in all the tests. Similar setup of parameterization as in CRM wing case was used in first test which dealt with three different meshes. The root section of the FFD lattice was fixed in order to guarantee smooth transition between deformed and undeformed part of the wing. The RBF lattice adapted to the wing is depicted in Fig. 4.12. The second group of test was investigating the effect of dimensionality on the optimization results and the third the effect of NURBS degree.

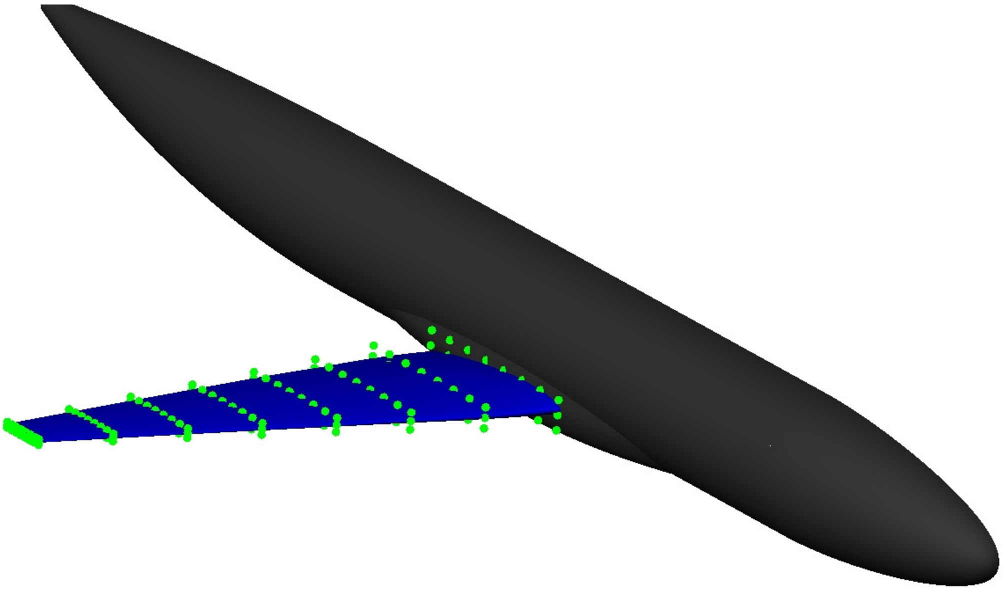


Fig. 4.12: RBF adapted lattice on the Passenger aircraft

Mesh and CFD setup: Unstructured meshes consisting of tetrahedral elements were generated in ANSYS IcemCfd meshing software. Three meshes were generated, medium size mesh with 763 874 nodes was used in most of the tests. The Edge[72] CFD solver was used in Euler flow setup.

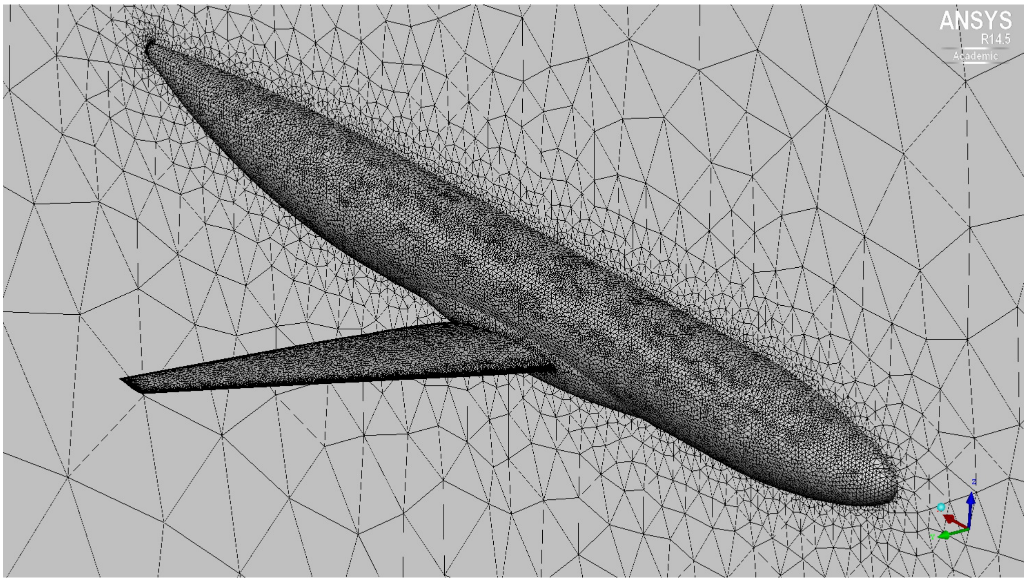


Fig. 4.13: Mesh of the Passenger aircraft

Effect of CFD mesh size on the optimization

216 parameters were used in this study of effect of CFD mesh size on the optimization results. Tab. 4.3 summarize the results. The baseline coefficients of different meshes are within 1% difference. The finer the mesh used the better are the results of the optimizations.

Tab. 4.3: Results of passenger aircraft case optimization with 216 design variables on three meshes

Mesh	Case	$c_{D_{opt}}[\%]$	$c_{L_{opt}}[\%]$	$c_{m_{opt}}[\%]$	$Vo_{opt}[\%]$	cost[a]
Coarse	Baseline	99.50	100.01	99.84	100.00	1
Coarse	Optimized	34.81	99.85	96.76	99.99	83
Medium [b]	Baseline	100	100	100	100	1
Medium	Optimized	33.59	99.88	96.82	99.99	89
Fine	Baseline	100.67	100.01	99.98	100.00	1
Fine	Optimized	32.95	99.89	96.74	99.98	90

[a] cost = CFD + adjoint (drag, lift)

[b] reference case

Effect of dimensionality

The test were carried out with different FFD lattice sizes in x and y directions see Fig. 4.4.

One of the conclusions is that increase in the number of parameters let to better results of the optimization as can be seen in Fig. 4.14. The other observation is that the number of parameters in span-wise direction has more pronounced effect than in stream-wise direction, at least for quite fine FFD lattices, see results for 408 and 432 (the different number is caused by the fixed root section of the FFD lattice).

Tab. 4.4: Results of passenger aircraft wing optimization with varying number of variables on medium mesh.

No.	a	b	c	$c_{D_{opt}}[\%]$	$c_{L_{opt}}[\%]$	$c_{m_{opt}}[\%]$	$Vo_{opt}[\%]$	cost[a]
Baseline	-	-	-	100	100	100	100	1
30	4	2	2	48.97	99.98	97.71	100.00	113
108	8	2	2	38.82	99.81	96.87	100.01	68
216	8	8	2	33.59	99.88	96.82	99.99	89
408	16	8	2	31.54	99.76	96.95	100.01	114
432	8	16	2	30.44	100.11	96.81	99.99	131

[a] cost = CFD + adjoint (drag, lift)

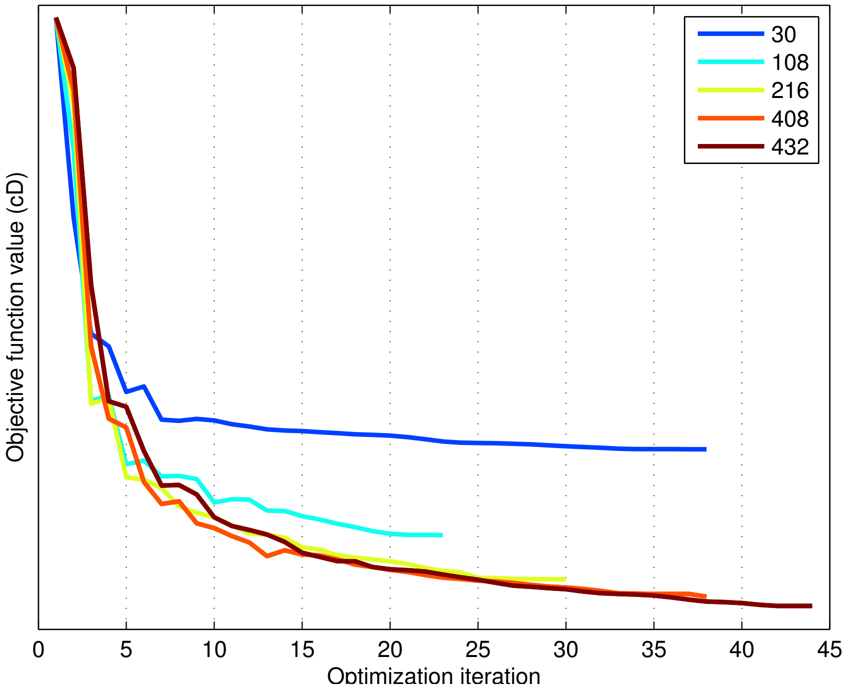


Fig. 4.14: Comparison of optimization history, using different number of parameters

NURBS degree influence on the optimization results

Different NURBS degree in stream-wise direction 108 variables used here. Maximum NURBS degree was used for the other directions. As can be seen the effect of NURBS degree in stream-wise direction on the optimization results is rather inconclusive.. The NURBS degree in stream-wise direction does not affect the reached minimum (the differences are considered in tolerance of the precision of the CFD), see Tab. 4.5. What it affects is the number of iteration needed to reach the optimum, the higher the degree the faster the optimization convergence is (with the exception of NURBS degree = 6.

Tab. 4.5: Results of passenger aircraft wing optimization with varying number of NURBS degree in stream-wise direction

NURBS degree p	$c_{D_{opt}}$[%]	$c_{L_{opt}}$[%]	$c_{m_{opt}}$[%]	Vo_{opt}[%]	cost^a
Baseline	100	100	100	100	1
2	37.81	99.96	97.04	99.93	95
4	38.44	100.14	97.20	99.94	77
6	37.76	99.89	96.94	99.98	83
8	38.82	99.81	96.87	100.01	68

[a] cost = CFD + adjoint (drag, lift)

Different NURBS degree in stream-wise and span-wise direction 216 variables used here. This time NURBS degree varies both in stream-wise and span-wise direction.

Tab. 4.6: Results of passenger aircraft wing optimization with varying number of NURBS degree in both stream-wise and span-wise directions

NURBS degree $p = m$	$c_{D_{opt}}$[%]	$c_{L_{opt}}$[%]	$c_{m_{opt}}$[%]	Vo_{opt}[%]	cost^a
Baseline	100	100	100	100	1
2	31.74	100.11	96.76	99.98	131
4	31.37	100.09	96.80	99.99	119
5	31.44	100.03	96.69	99.98	113
6	31.97	100.11	96.88	100.00	108
8	33.59	99.88	96.82	99.99	89

[a] cost = CFD + adjoint (drag, lift)

The table Tab. 4.6 gives similar conclusion as came out of the previous test. The NURBS degree does not affect the reached minimum, and with increase in the degree a slight acceleration of optimization convergence is obtained.

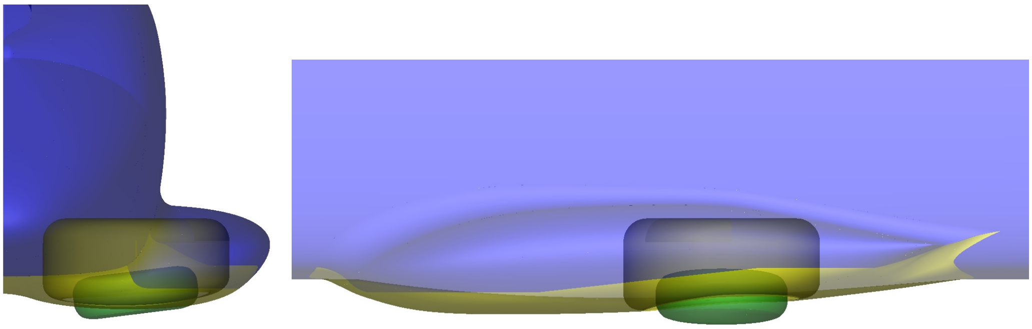
4.3.3 Complex geometrical constraints handling: EV-55 Outback landing gear nacelle aerodynamic shape optimization

The commuter aircraft landing gear nacelle optimization is an excellent test case for demonstrating the FFD-RBF parameterization. The ultimate goal was to decrease drag of the aircraft with "open" landing gear nacelle that would allow the use of smaller landing gear doors. A multi-point optimization in cruise and climb conditions subjected to geometrical constraints such as inner structure of landing gear nacelle and landing gear itself. Navier-Stokes calculations are needed in order to decrease viscous and pressure drag as the aircrafts cruise speed is well bellow transonic speeds where no wave drag exists and thus Euler flow solution will not be sufficient. The aircrafts geometry comes from Evektor, spol. s.r.o., so all the results and plots are consider as confidential and as such are presented in relative values and without scales.

Geometry: Only part of the landing gear nacelle surface area was allowed for modification as the aircraft is already in late design phase. In the original state of the geometry the wheel was hidden inside closed landing gear nacelle. The Evektor company requested to minimize the negative effect of open landing gear nacelle that does not cover the wheel.

Fig. 4.15: EV-55 Outback. Image downloaded from `http://www.evektor.cz/outback/fotogalerie.aspx`

A hole was cut into the nacelle and cavity inside the landing gear nacelle was created to simulate the inner flow around the wheel, see Fig. 4.16. Only the yellow area, which consists of part of the surface of landing gear nacelle and the cavity, was subject of deformations. The surface of the aircraft (blue) as well as the wheel (green) were not to be deformed.



(a) front view

(b) side view

Fig. 4.16: Transparent view of the landing gear nacelle

Mesh and CFD setup Unstructured hybrid mesh consisting of tetrahedral elements and prismatic layers was generated in ANSYS IcemCfd meshing software. Mesh with 2 328 907 nodes. The Edge[72] CFD solver was used in RANS setup with S-A turbulence model.

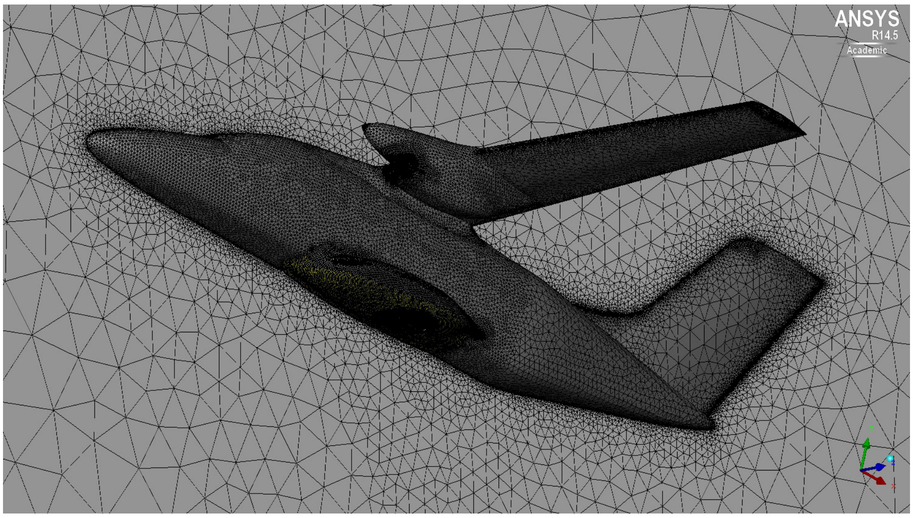


Fig. 4.17: EV-55 Outback CFD mesh

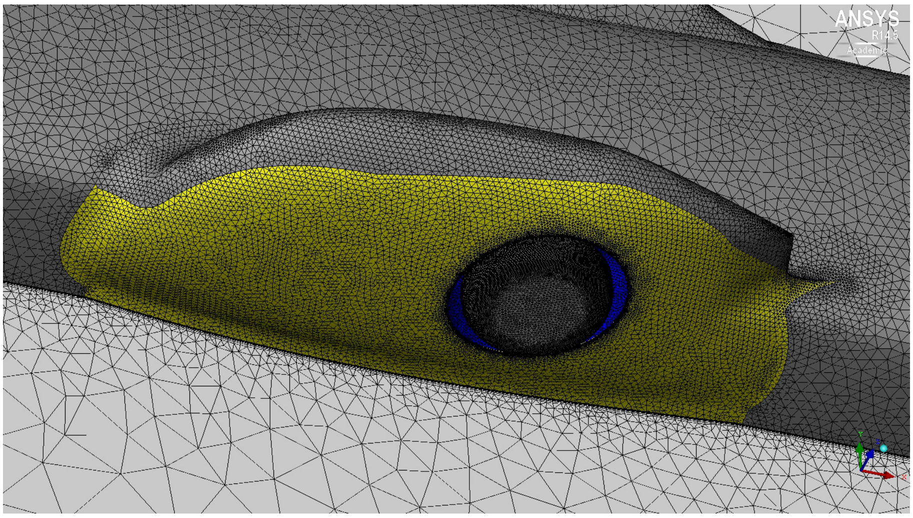


Fig. 4.18: EV-55 Outback detail of mesh on the landing gear nacelle

Parameterization The FFD-RBF parameterization was used on part of the geometry designed for deformation. A set of optimization cases were performed ranging from 1 to 9 parameters. These parameters were as usual FFD control points displacements and they were located on the bottom side of the nacelle, see red point in Fig. 4.19. All the (green) points were frozen in order to fix the boundary curve that separates the deformable and undeformable areas.

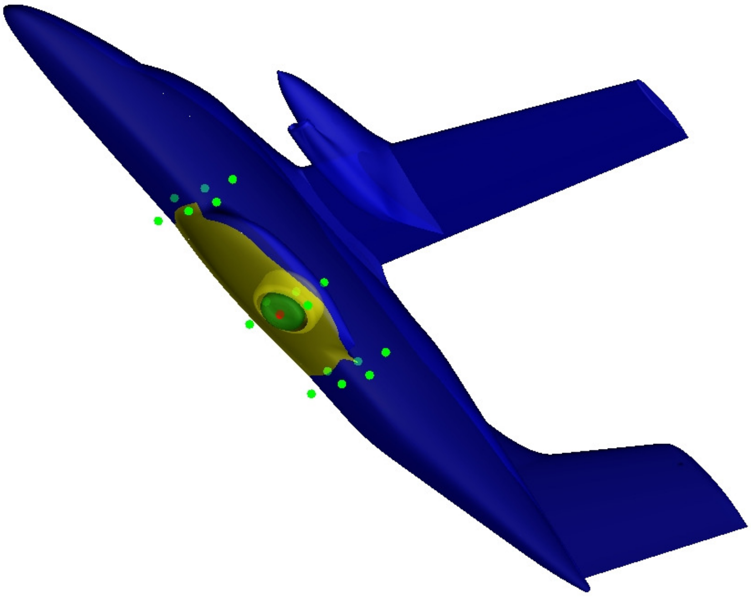
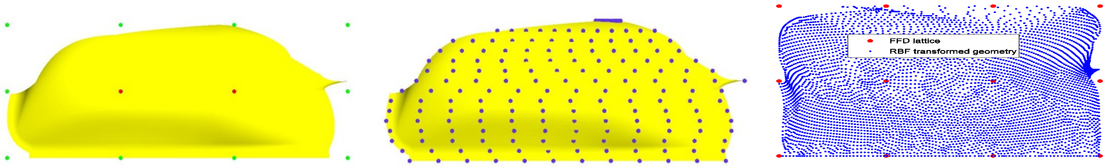


Fig. 4.19: EV-55 Outback Parameterization example



(a) FFD lattice (b) RBF lattice (c) mapped geometry

Fig. 4.20: FFD parameterization with RBF coordinate transformation processes

Geometrical constrains fixation Only the yellow area Fig. 4.19 can be deformed, the boundary curve that surrounds it has to remain fixed. Fixation error was defined in order to quantitatively evaluate how precisely was the boundary between deformable and fixed geometry maintained.

The FFD lattice was constructed around the deformable geometry (see Fig. 4.20a), RBF coordinate transformation (see Fig. 4.20b) was used to map the deformable surface into the FFD lattice (see Fig. 4.20c). The influence of number of RBF centers in x, y and z directions on the fixation error (equation 4.4) is given in Appendix A Fig. A.1 to A.3.

Comparison of fixation error with basic FFD parameterization and with optimal FFD-RBF parameterization is given in Tab. 4.7, the use of RBF coordinate transformation gave 47,1 % improvement over the standard FFD parameterization. The fixation error is determined for quantitative evaluation of constraint handling, the quality of fixation of the non-deformable geometry is demanded by necessity of volume mesh deformation between each optimization iteration. In other words, if the parameterization fails to keep the geometry fixed in some tolerance the volume mesh deformation process will crash and the optimization would be stopped, with the use of FFD-RBF this had never happened. The number of RBF centers for coordinate transformation have influence on the value of fixation error and they should be adjusted for every new type of object to be optimized.

Tab. 4.7: Boundary curve fixation error

	Error [%]
FFD	53.4
FFD-RBF	6.3

Aerodynamic shape optimization The gradient based optimization approach adopted in previous test was no longer an option, since adjoint of Navie-Stokes equations was not part of used version of Edge flow solver and calculation of gradients with finite differences was due to size of the mesh and slow character of CFD solution too expensive. Because of that a simplex optimization approach in NLOPT[70] optimization software package was used. Propagation of wall deformations to volume mesh was done by FOIs in-house software meshdeform. A set of 6 optimizations with different number of optimization variables was performed.

$$CF = 0.7c_{D1} + 0.3c_{D2} \tag{4.7}$$

where c_{D1} stands for drag coefficient in cruise conditions and c_{D2} for climb flight conditions.

The original geometry with closed nacelle was also analyzed and compared to baseline (initial) geometry that should be optimized, where the baseline stands for the geometry with open landing gear nacelle. As expected the whole in the smooth surface and a cavity caused additional drag, the drag of the whole aircraft increased roughly by 6 %. That is a serious increase considering that the area of the closed nacelle generated only 1.4% of total drag in the original case (see Tab. 4.8).

Tab. 4.8: EV-55 Outback closed nacelle CF according to area

	CF value portion [%]
Wing	57.0
Engine nacelle	11.6
Fuselage	29.9
Landing gear nacelle	1.4

The shapes and pressure coefficient distributions for the baseline and with 6 parameters optimized case are presented in Fig. 4.21 for cruise conditions and in Fig. 4.22 for climb conditions. Note the nacelle shape change (inflation) result of the optimization process.

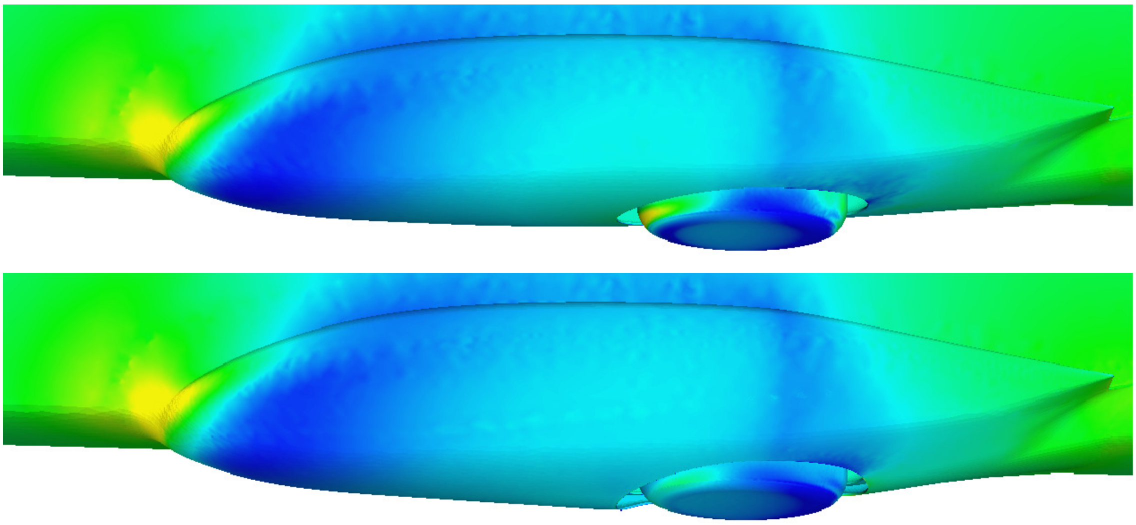


Fig. 4.21: Comparison of initial (top) and optimal (bottom) shapes and pressure coefficient distribution in cruise conditions

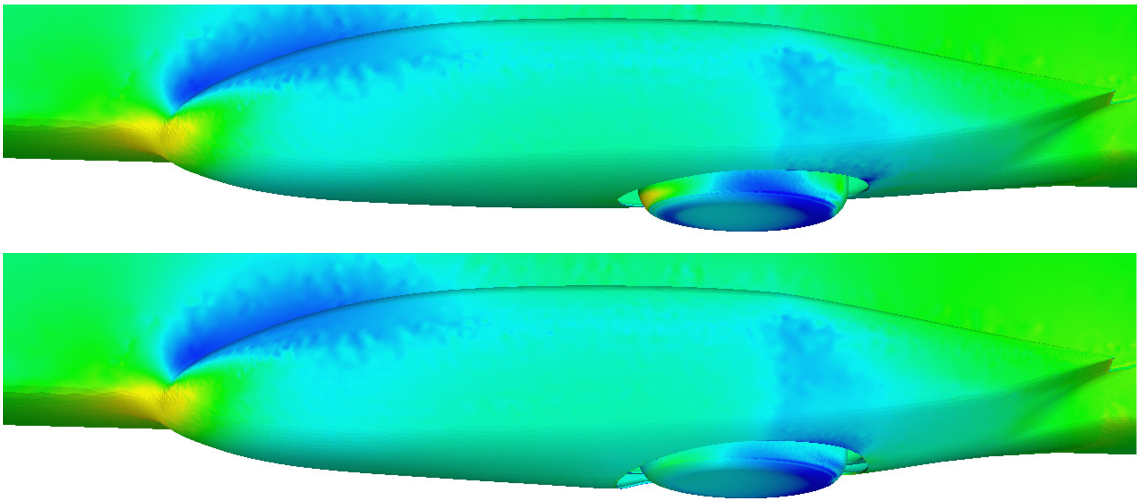


Fig. 4.22: Comparison of initial (top) and optimal (bottom) shapes and pressure coefficient distribution in climb conditions

The results of optimization of the open case are summarized in Tab. 4.9. Approximately 2 % decrease can be observed throughout all optimizations.

Tab. 4.9: EV-55 Outback optimization results

Parameters	Improvement [%]
baseline	-
1	2.012
2	2.015
3	2.122
4	2.051
6	1.980
9	2.129

5 ADAPTIVE FFD PARAMETERIZATION WITH RESPECT TO OPTIMIZATION

5.1 Introduction

Another kind of adaptivity of the parameterization is the adaptivity with respect to the optimization. That means that the parameterization is adapted (changed) during the optimization process, usually after some criteria is reached. The motivation of such operation is to accelerate the optimization run, in other words to make it cost less in computational time. Acceleration of the optimization procedure in real time would be also valuable while shortening the design cycle time.

Two different optimization acceleration methods were investigated on the tasks of aerodynamic shape optimization, Enrichment and Multi-grid.

5.2 Enrichment

Adaptive optimization approach called Enrichment is a method based on increase of the number of optimization parameters and their smart insertion into the FFD lattice. The enrichment procedure is tested on NACA 0012 2D optimization case analyzed in section 3.7.1. A conclusion is that the case needs of large number of optimization parameters (see Tab. 3.2).

5.2.1 Enrichment procedure

The enrichment optimization procedure starts with coarse FFD lattice and runs in a loop. After the insertion criteria is met the FFD lattice is enriched in chosen section, the last geometry is embedded into the new FFD lattice and the loop repeats until the stopping criteria is met.

Insertion criteria: Minimal decrease of relative function value between last two optimization iterations

Choosing section for enrichment: The FFD lattice is locally refined at a location that depends on the size of the area under the shape gradient (obtained by the adjoint flow solver).

Calculation of area:

$$A_{section} = \sum_{i=1}^{n-1} (x_{p_{i+1}} - x_{p_i}) \frac{abs(g_{r_{i+1}}) + abs(g_{r_i})}{2} \tag{5.1}$$

where x_p is surface mesh point coordinate in x direction and g_r is gradient in that surface point. So in the case of just one section enrichment a column of control points is added into the FFD lattice (see Fig. 5.1) where the area is largest (where the shape changes have biggest potential).

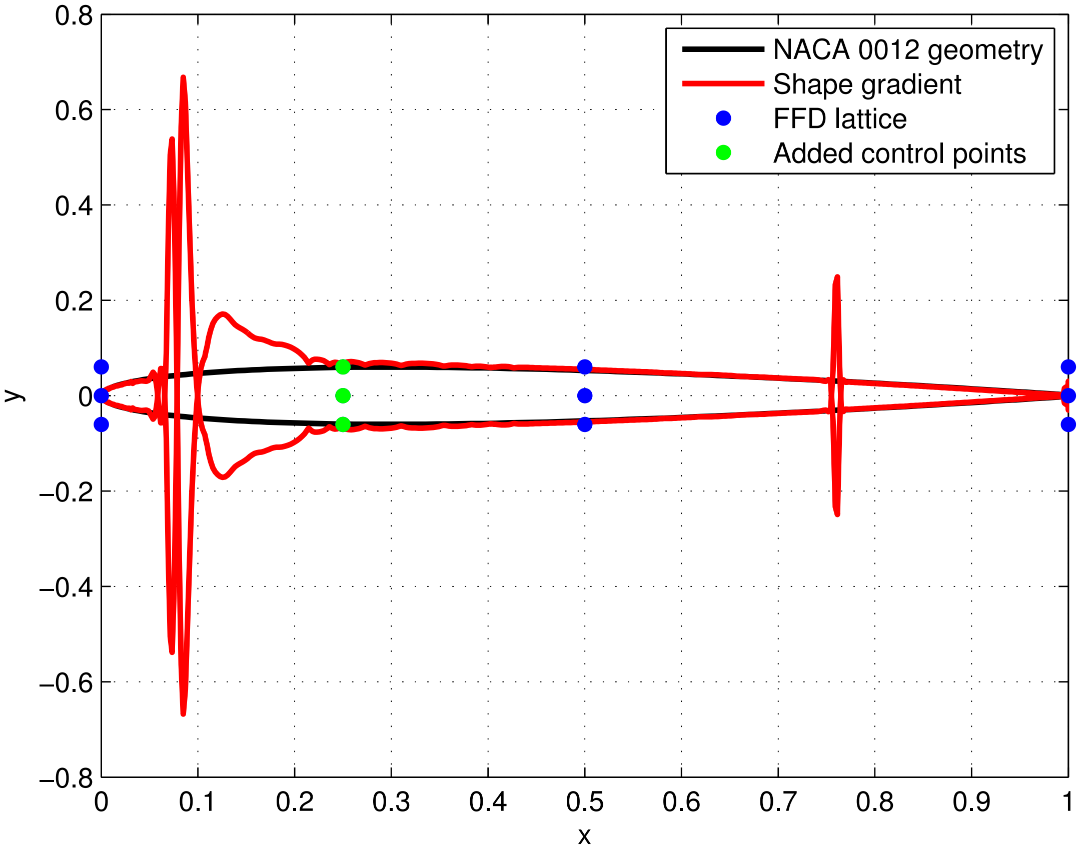


Fig. 5.1: One step of enrichment of FFD lattice based on Shape gradient

Stopping criteria:

- Maximum number of total optimization iterations
- Minimal difference of optimization parameters between last two optimization iterations

One step enrichment test case

A number of optimizations with regular initial FFD lattices are compared with the same FFD lattices enriched by one column of control points and with regular lattices of the same sizes. Maximal NURBS degree was used (in x direction) based on previous results in section 3.7.1. The insertion criteria was set to 0.02. The optimization finished after one of the optimization stopping criteria was met, the stopping criteria were: 30 optimization iterations or the difference of optimization parameters between last two optimization iterations was smaller than 1e-8.

Tab. 5.1: Comparison of Enrichment (+1) and NACA0012^a optimization for a hierarchy of FFD lattices with max. NURBS degree^b

Regular			Enriched			Regular		
No.	$c_{D_{opt}}$	cost^c	No.	$c_{D_{opt}}$	cost^c	No.	$c_{D_{opt}}$	cost^c
Baseline	0.04750	1	-	-	-	-	-	-
3	0.03144	23	3 + 1	0.03053	39	4	0.02964	16
6	0.02132	32	6 + 1	0.01952	41	7	0.01526	37
11	0.01300	197	11 + 1	0.01832	224	12	0.01246	64
21	0.01187	239	21 + 1	0.01433	225	22	0.01123	185

[a] CFD grid size is 42556 nodes.

[b] NURBS degree $p = \mathrm{N} - 1$.

[c] Number of flow and adjoint flow solutions

The result in Tab. 5.1 shows that the enrichment process did not fulfilled the expectations. In cases of small number of parameters (3 and 6) the addition of one more brought some improvement of the drag coefficient, but the comparison with regular FFD parameterization of the same number of elements (4 to 22 parameters) is not favorable at all in all analyzed cases. The regular FFD of the same size as the enriched was able to find better optimum and converged faster (see Fig. 5.2)

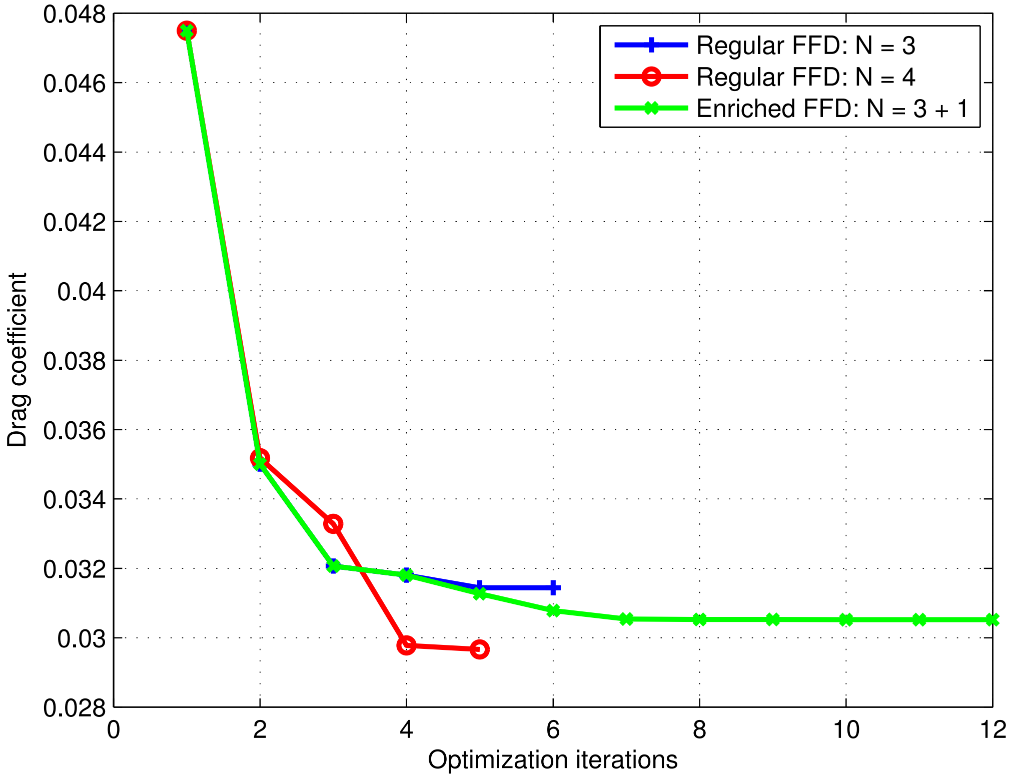


Fig. 5.2: Comparison of history of regular and enriched NACA 0012 optimization using 3, 4 resp. 3 + 1 parameters

After this failure of the enrichment approach, where even the one step did not lead to acceleration of the optimization process, no further tests were performed. A note must be made that the enrichment procedure is influenced by the insertion criteria which further complicates finding of one general beneficial setup for wide variety of cases.

5.3 FFD Multi-grid

Multi-grid (Multilevel) methods were developed for solving differential equations using a hierarchy of discretization and are currently part of most CFD solvers

Multi-grid approach for solving the optimization problem is based on hierarchy of CFD grids sharing the same parameterization. Using few levels (from coarse to fine) of CFD grids could lead to accelerated convergence of the optimization toward the optimal solution and thus increase the computational efficiency. This assumption is based on the fact, that the CFD and adjoint solution is achieved at much lower computational cost on coarse grid than on fine one. Similarly to multigrid method in CFD a results (of the optimization) on the coarse grid is used to accelerate optimization convergence of fine grid.

5.3.1 FFD Multi-grid procedure:

The Multi-grid (MG) principle was studied on the CRM wing case used previously for other optimization analysis in section 4.3.1. Here the same setup of the parameterization and optimization is used as in FFD-RBF case.

Two meshes were selected, coarse mesh with 169 381 nodes and medium with 854184 nodes. A significant differences can be observed in Tab. 5.2 between the results of coarse and medium mesh. The coarse mesh gave smaller c_D of baseline shape and also the result of optimization is better than with medium mesh. The explanation is that the CFD solution on the coarse mesh is not grid independent and as shown in our publication in AIAA conference proceedings[71] a larger grid is necessary for reliable results. Nevertheless the reliability issue is not of concern, since the optimization on coarse mesh is performed just to give the optimization on the medium mesh a better initial design.

Tab. 5.2: Results of CRM wing optimization on M = 0.88 with 242 design variables using NLPQLP using Laplace-Spring on two meshes.

Mesh	Case	$c_{D_{opt}}$	$c_{L_{opt}}$	cost[a]	
Coarse	Baseline	0.01676	0.49969	-0.23313	1
Coarse	Optimized	0.01131	0.50209	-0.23876	52
Medium	Baseline	0.01797	0.49948	-0.23638	1
Medium	Optimized	0.01287	0.50194	-0.21874	43

[a] cost = CFD + adjoint (drag, lift)

Several multilevel cases were investigated in respect to different number of main iterations done on coarse mesh. The investigation revealed a problems with the transition between the two meshes. The optimal displacements of the coarse mesh were used as the initial guess for the medium mesh, but the initial deformation of the medium mesh failed in almost all the cases. Needless to say that the mesh deformations during the optimization on the coarse mesh were alright.

One working case, which used 10 main optimization iterations on coarse mesh in the first step, is here presented.. Results of the this Multi-grid FFD optimization are given in Tab. 5.3.

Tab. 5.3: Comparison of medium mesh optimization and multilevel optimization results

	Multi-grid	Medium
c_D baseline	0.017973	0.017973
c_D optimal	0.013026	0.012874
c_D reduction	27.5 %	28.4 %
Cost in CFD + adjoint iterations	35 Coarse + 34 Medium	43
Cost in CPU time	36 158	37 579
Cost in real time	7h 4min	6h 31min

Comparison of the MG optimization with the medium mesh optimization shows that the MG gave 0.9 % worse c_D and was 3.4 % faster in CPU time measurement and 8.4 % slower in real time. That disagreement in CPU and real time was caused by usage of 8 Intel Xeon E5-2690 cores for coarse mesh and 16 cores for medium mesh. The number of CPU cores is usually correlated to the size of the CFD mesh and is always trade-off between efficiency of the computational cluster usage and real time results availability.

The Tab. 5.3 also shows that the cost in CFD + adjoint iterations is about the same for coarse and medium mesh in MG case and that the 34 iterations on medium mesh in MG case is lower than 43 in medium case, so some effect acceleration of the optimization on medium case is observed. Both MG medium mesh and medium optimizations ended by reaching maximum number of function calls during the line search and that was caused be ever failing CFD mesh deformation procedure.

Fig. 5.3 shows the comparison of medium mesh optimization and MG optimization convergence history. A deeper fall of the c_D can be observed in the beginning of the MG case which then progresses towards the optimum, which was just above 0.011 but was not reached since the MG algorithm stopped it after 10 main iterations. Right after that a first c_D value on the medium mesh was calculated (the

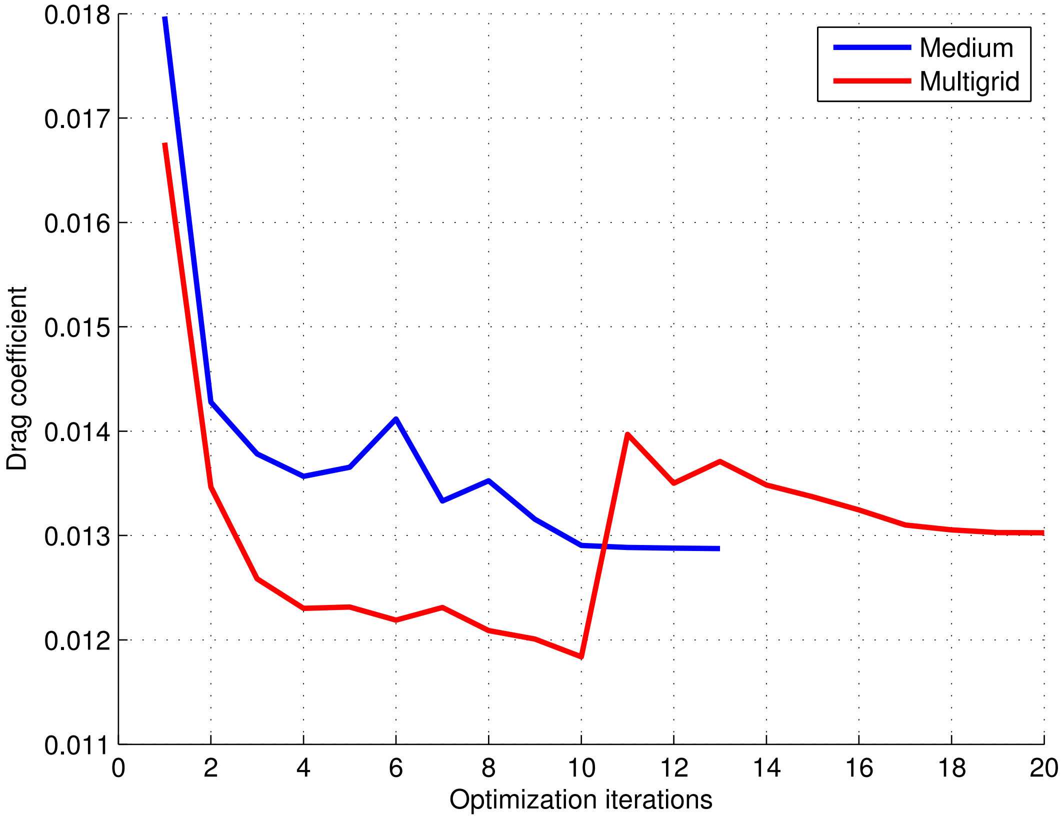


Fig. 5.3: Comparison of medium mesh optimization and MG optimization convergence history

peak in the red line at 11^{th} iteration), note that the value of c_D was much lower than the initial value of c_D on the blue curve.

6 FFD FOR CFD MESH DEFORMATION

6.1 Introduction

The second proposed objective of the thesis is development of FFD parameterization for both surface deformations and CFD mesh deformations, while enabling large object deformations and preserving the level of mesh quality during the process. This approach will bring simplification to the optimization process by using parameters of surface mesh description as optimization variables, so there will be need neither for new mesh generation, nor for using another mesh morphing program.

Mesh deformation is standard way of adjusting the computational mesh to changes in object shape during the optimization procedure, so there is no need to generate the CFD mesh again after every iteration as in the past. Laplace smoothing in which large system of equations has to be solved is very common as well as spring analogy [74] method in which is each element edge represented by a spring with corresponding stiffness (also system of equations). Another approach to CFD mesh deformation is RBF[28] which is independent of the mesh connectivities unlike the above mentioned.

The capability of smooth volume deformations makes FFD a suitable candidate for CFD mesh deformation[49, 63, 9] The FFD is independent of the mesh topology, so structured or unstructured meshes are deformed by the same algorithm as well as hybrid meshes.

Motivation of using FFD parameterization for mesh deformation (other than problems with failing standard methods in previous cases) is in simplification of the optimization process. The object's shape (subject to the optimization cost function) will be deformed together with the volume mesh that surrounds it. Thanks to that the use of another mesh morphing program can be avoided.

Tests in 2D and 3D, in comparison to standard methods, namely Laplace and Spring analogy were performed. Both Euler and RANS meshes were used.

The CFD mesh deformation methods are evaluated with respect to:

- Mesh quality after morphing focused on aspect ratio and skewness.
- Cost of the deformation process with focus on the CPU time and storage demands of all parts of the process.

6.2 Procedure:

The general procedure for CFD mesh deformation with FFD is very similar to basic FFD procedure in section 3.3, the biggest difference is in the construction of the FFD lattice.

1. Usually a initial lattice of control points is constructed around the object (surface mesh) that is to be deformed. Then one or more layers of control points are added on that lattice. These additional layers defines how big part of the CFD mesh will be deformed (see Fig. 6.1).

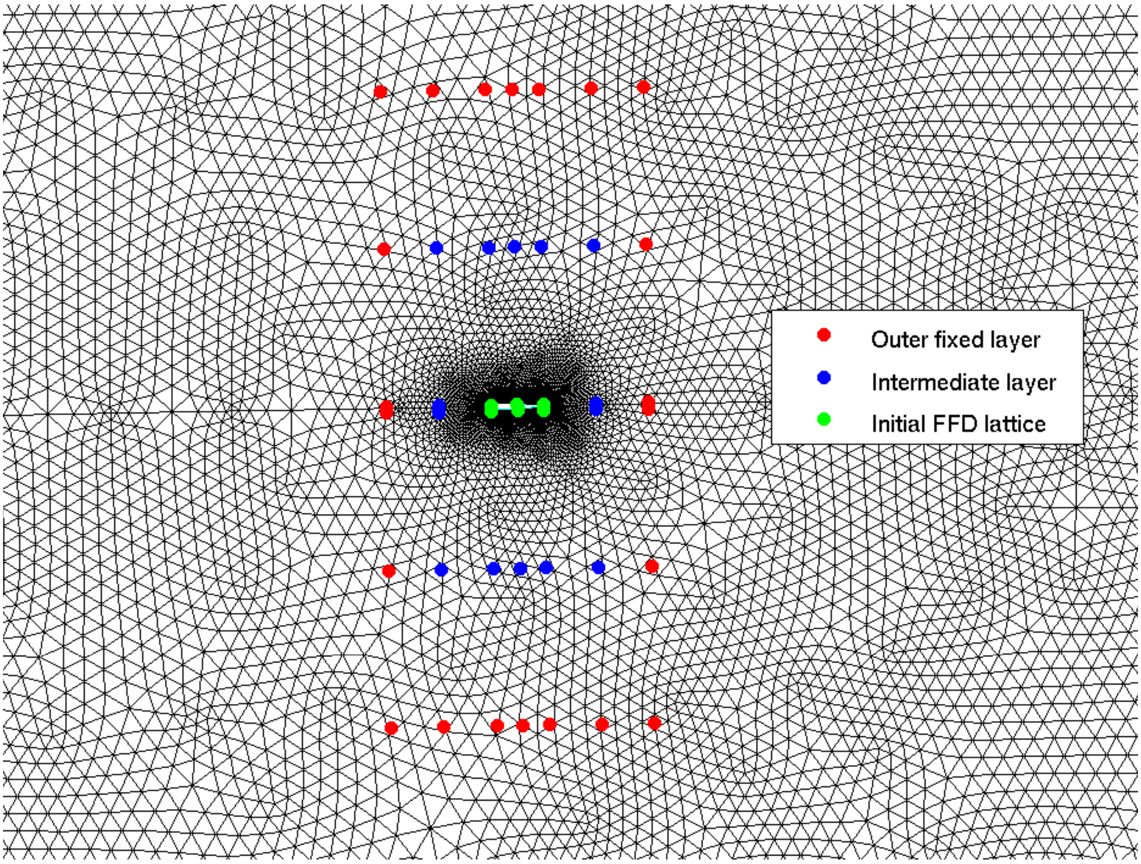


Fig. 6.1: FFD lattice for deformation of CFD mesh in NACA 0012 case

The lattice construction has a practical limitations (boundary surfaces, multisegment high lift devices) which limits the applicability of the method.
An example of such limit is depicted on Fig. 6.2, typical task would be to find new position of the flap and its angle of deflection, the deformation of CFD mesh with such FFD lattice would result in deformation of the airfoil as well, which is of course prohibited. Note also the typical narrow gap between the flap itself and the airfoil, a problem of most CFD mesh deformation tools that would be hardly overcome by the FFD.

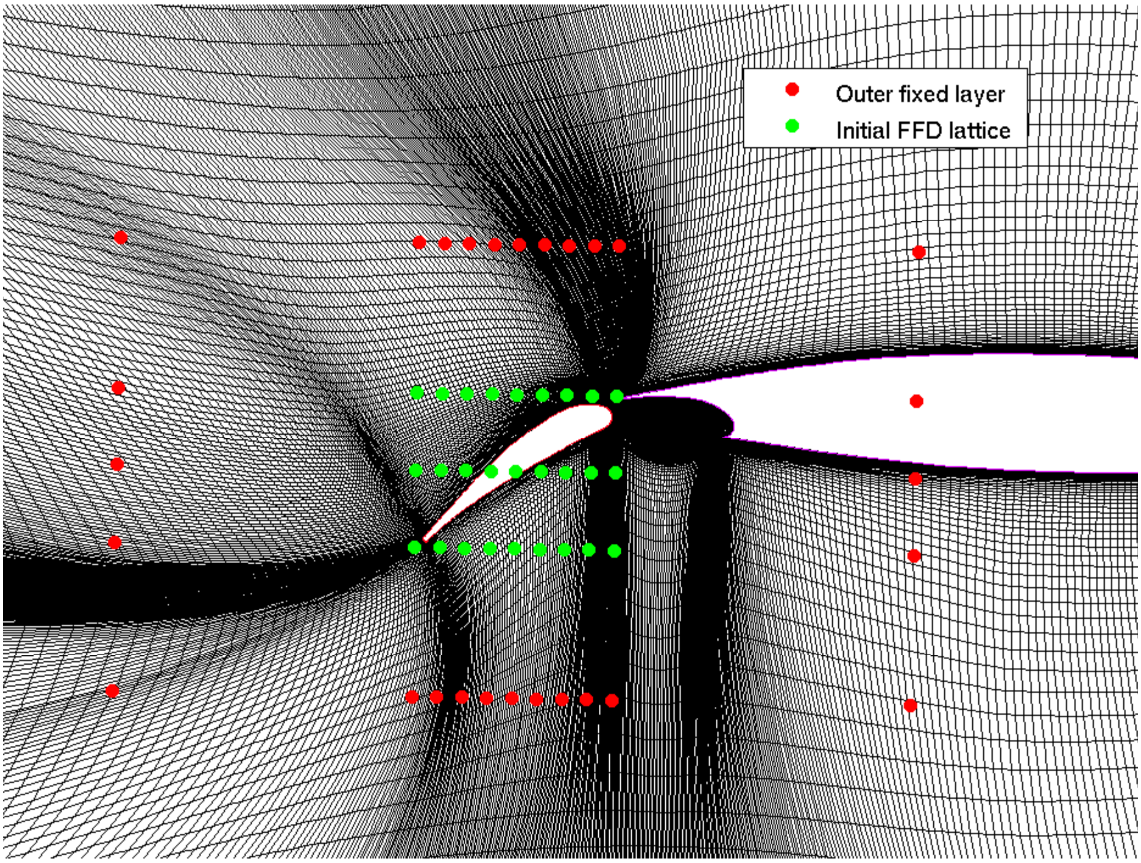


Fig. 6.2: Example of practical limitations of FFD for CFD mesh deformations, in the task of flap position optimization

2. The part of the CFD mesh that is located inside the FFD lattice is embedded within the parametric volume.
3. The lattice is deformed. Preferably the control points of the initial lattice are displaced (as optimization variables), the additional layers of control points can be displaced to shift the majority of volume cell deformations further from the objects surface. The outer most layer of the FFD lattice has to be fixed in order to keep the transition between the deformed and undeformed volume mesh smooth.
4. The deformed coordinates of the CFD mesh are calculated using corresponding equation, for 3D 3.8.

6.3 Numerical experiments: FFD vs. Standard methods

The CFD mesh deformation capabilities of FFD parameterization is analyzed and compared to Laplace and Spring analogy standard methods in terms of quality of the deformed mesh and in terms of computational efficiency of the deformation process.

6.3.1 Quality measure:

Aspect ratio (Ar):

It is the ratio between lengths of the longest and the shortest edge of a cell. The aspect ratio should be ideally equal to 1 to ensure accurate results.

Here a measure that also takes into account the surface for tri elements and volume for tetra elements is taken, this measure is defined so that aspect ratio equal to 1 is the best ant equal to 0 worst. To many elements with too low aspect ratio can result in high interpolation error of the CF solution.

Tri elements:

$$Ar = \frac{(\frac{\text{area}}{l_{max}})_{actual}}{(\frac{\text{area}}{l_{max}})_{optimal}} \tag{6.1}$$

Tetra elements:

$$Ar = \frac{(\frac{\text{volume}}{r_c^3})_{actual}}{(\frac{\text{volume}}{r_c^3})_{optimal}} \tag{6.2}$$

where the optimal (equilateral) triangle is a triangle inside the same circumcircle as the actual tri element and optimal tetra is a tetra circumscribed into the same circle as the actual tetra element, the l_{max} is the longest edge of the triangle and r_c is circumscribed radius.

Quad elements: the aspect ratio is the minimum of 8 ratios from 4 parallelograms constructed from two vectors adjacent to each node.

$$Ar = min(\frac{A_1}{a}, \frac{A_1}{b}, ..., \frac{A_4}{a}) \tag{6.3}$$

where the a, b are adjacent vector and the area A is calculated as

$$A = a * b * sind(\alpha) \tag{6.4}$$

where α is angle between the adjacent vectors.

Hexa elements:

$$Ar = \frac{l_{min}}{l_{max}} \tag{6.5}$$

where l_{max} is the longest edge and l_{min} is the shortest edge of the hexa element.

Skewness (Sk):

The skewness of a CFD mesh is a measure that indicates mesh quality and suitability. Large skewness compromises the accuracy of the interpolated regions.

Tri and tetra elements:

$$Sk = \frac{\text{optimal triangle area} - \text{triangle area}}{\text{optimal triangle area}} \tag{6.6}$$

where the optimal (equilateral) triangle is a triangle inside the same circumcircle

Quad and Hexa elements:

$$Sk = max\left[\ \tfrac{\alpha_{max}-90}{90},\quad \tfrac{90-\alpha_{min}}{90}\ \right] \tag{6.7}$$

where α_{max} is an angle in a face or cell.

6.3.2 Quality evaluation plan:

1. Visual inspection, that is to identify apparent errors in the mesh like inverted cells (see Fig. 6.3) or violated surface boundaries.

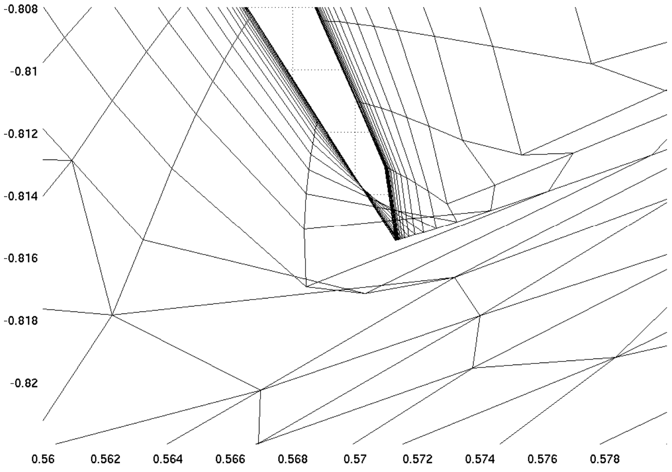


Fig. 6.3: Example of inverted cells in the area of airfoil trailing edge

2. Quality measurement calculation of skewness and aspect ratio of all mesh elements.
3. Dual - software build in Edge that also checks correctness and quality of the mesh. The outcome of the program is not usable for CFD simulation in the case of finding bad elements. That gives definitive stop to any further simulations with that particular mesh, that is a very practical condition monitored during the optimization procedure.
4. Influence of morphed mesh on CFD convergence and results is compared to convergence and results on meshes deformed with standard methods. The convergence of the solution is also monitored during the optimization procedure to reveal possible mesh deformation related issues.

6.3.3 2D meshes:

A comparison of Laplace, Spring analogy and FFD methods for CFD mesh deformations is here demonstrated by a search for maximal rotation angle of airfoils. That is equivalent to increase of the angle of attack imposed to the far-field boundary condition. Results of CFD simulation of rotated airfoils and increased angle of attack serves as ultimate quality evaluation. The rotation case was selected because it put demands both on aspect ratio and skewness of the deformed mesh elements. The meshdeform program in Edge was used to test the Laplace and Spring analogy performance.

Test description

The test is designed to keep increasing angle of attack until the dual program reports error or the meshdeform Edge program fails. That is done for Laplace, Spring analogy and FFD methods and for Euler and RANS meshes. The NACA 0012 Euler mesh is a mesh from section 3.7.1, the RAE 2822 RANS mesh comes from other part of publication by Amoignon, Hradil and Navratil[71] .

Initial FFD lattice of control points with the dimensions 3x3 (see Fig. 6.4 green points) was generated in the vicinity of the airfoil. The rotation of the control points of the initial FFD lattice around the origin was used to deform the airfoils geometry with the standard FFD procedure. Laplace and Spring analogy deformations were performed in meshdeform program that requires initial CFD (undeformed) mesh and deformed boundary nodes (airfoil) to produce deformed CFD mesh.

In the case of FFD method a one layer of control points was added on the initial lattice. The added outer layers was fixed, see red points in Fig. 6.4.

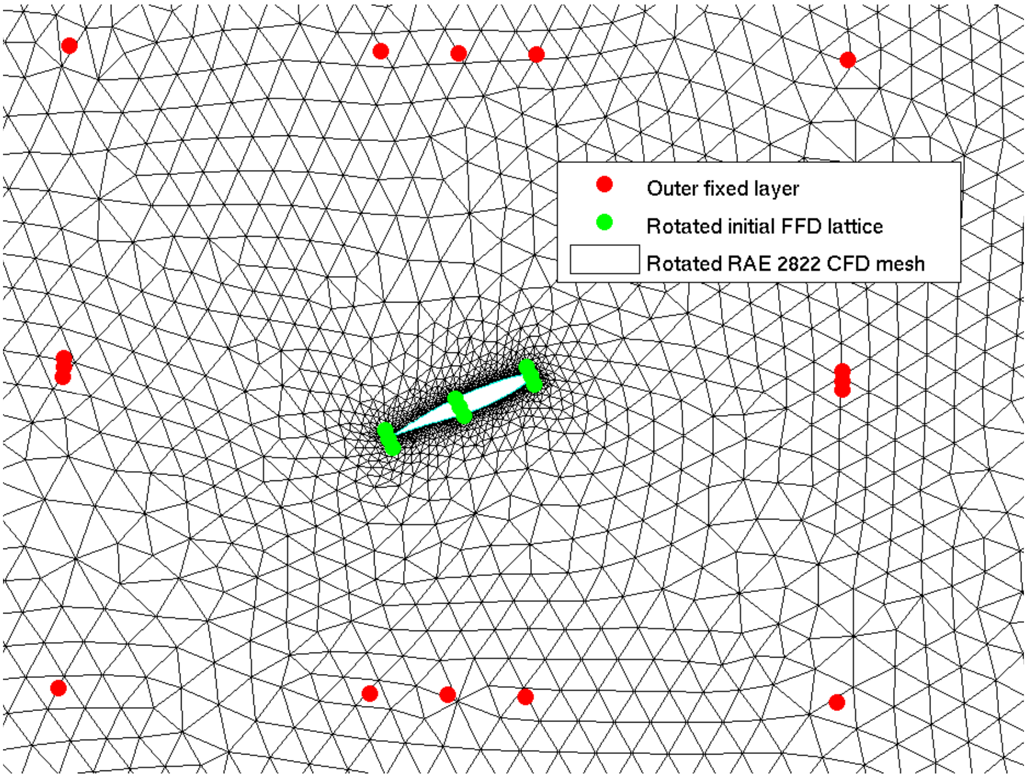


Fig. 6.4: Example of rotation of RAE 2822 CFD mesh with FFD

Results

The maximum achieved angle of mesh deformations by rotation are summarized in Tab. 6.1, note that the FFD method achieved much higher angles than the standard methods on both meshes. Visual inspection revealed nothing suspicious in the cases of Euler mesh (see Fig. B.1 to B.6 in the Appendix B). That cannot be said about Fig. B.9 which exhibits elements that are violation surface of the airfoil close to trailing edge, nevertheless the dual program evaluation went throw and the Edge flow solver converged even with such mesh. Note that the spring analogy method failed completely to deform the RANS mesh.

Tab. 6.1: Maximal angle of rotation

Mesh	Laplace	Spring analogy	FFD
NACA 0012 Euler	16°	34°	58°
RAE 2822 RANS	25°	failed	56°

Tab. 6.2 shows worst calculated element aspect ratios and skewness for NACA 0012 and their comparison to the baseline undeformed mesh. Note that both skew-

ness and aspect ratio of worst element of the deformed CFD grid is better using FFD than the standard methods in all comparable angles.

Tab. 6.2: Maximal angle of rotation test, NACA 0012 Euler mesh

Angle [°]	Method	worst Sk	worst Ar
0	-	0.3355	0.5802
16	Laplace	0.8868	0.1020
16	Spring A.	0.6226	0.3661
16	FFD	0.6120	0.3738
34	Spring A.	0.9999	0.0015
34	FFD	0.8836	0.1561
58	FFD	0.9997	0.0029

Tab. 6.3 shows worst calculated element aspect ratios and skewness for RAE 2822 RANS mesh and their comparison to the baseline undeformed mesh, of which is very hard to tell any conclusion. Note that the results are very different from the ones obtained in the case of Euler grid, the difference is due to the extremely narrow first layer of the prismatic elements.

Tab. 6.3: Maximal angle of rotation test, RAE 2822 RANS mesh

Angle [°]	Method	worst Sk	worst Ar
0	-	0.9999	3.65e-4
25	Laplace	0.9997	5.08e-4
25	FFD	0.9993	6.17e-4
56	FFD	0.9998	3.70e-4

Tab. 6.4 shows results of the CFD solution in program Edge on NACA 0012 airfoil and their comparison to the baseline undeformed mesh under equivalent angle of attack (AoA). The same for RAE 2822 airfoil is presented in Tab. 6.5.

The results are almost identical for all deformation methods, but they slightly differ from the undeformed mesh results under equivalent AoA. That is probably caused by not sufficient mesh quality, in other words the flow solution is still mesh dependent. Nevertheless the FFD exhibits good compliance with the results of standard methods which is what matters most.

Tab. 6.4: CFD results on deformed NACA 0012 airfoil meshes

Angle [°]	Method	c_L	c_D	c_m
16	AoA	1.71	0.51269	-0.83558
16	Laplace	1.7161	0.56456	-0.96446
16	Spring A.	1.7161	0.56456	-0.96446
16	FFD	1.7124	0.56361	-0.96121
34	AoA	0.92132	0.68432	-0.48773
34	Spring A.	0.85098	0.59729	-0.57917
34	FFD	0.84126	0.59522	-0.57695
58	AoA	0.65013	1.00400	-0.54626
58	FFD	0.66359	1.02280	-0.81761

Tab. 6.5: CFD results on deformed RAE 2822 airfoil meshes

Angle [°]	Method	c_L	c_D	c_m
25	AoA	1.0002	0.62982	-0.20859
25	Laplace	0.9857	0.56017	-0.2836
25	FFD	0.98319	0.56009	-0.22637
56	AoA	0.72429	1.31500	-0.32782
56	FFD	0.94493	1.23870	-0.07099

6.3.4 3D meshes:

Similarly to 2D tests, here a comparison of Laplace, Spring analogy and FFD methods for 3D CFD mesh deformations is demonstrated, this time by a search for maximal elevation of wing tip. That is defined to imitate bending of wing by aerodynamic forces, however quite unrealistically extreme for the purpose of testing of the CFD mesh deformation methods. The elevation case was selected because it put demands both on aspect ratio and skewness of the deformed 3D mesh elements. Again meshdeform program in Edge was used to test the Laplace and Spring analogy performance.

Test description

The test is designed to keep increasing elevation of control points in the wing tip area until the dual program reports error or the meshdeform Edge program fails. That is done for Laplace, Spring analogy and FFD methods and for Euler and RANS meshes. A highly swept wing geometry was used to create Euler mesh for testing, the RANS mesh comes from CRM wing optimization done as a part of publication (Amoignon, Hradil, Navratil[71]).

Initial FFD lattice of control points 3x3x3 (see Fig. 6.5 green points) was generated in the vicinity of the wing. The elevation of tip section control points (red points) of the initial FFD lattice was used to deform the wing geometry with the standard FFD procedure. Laplace and Spring analogy deformations were performed in meshdeform program that requires initial CFD (undeformed) mesh and deformed boundary nodes (airfoil) and produces deformed CFD mesh.

In the case of FFD method a one layer of control points was added on the initial lattice in 5 of its sides, no layer was added in the wing root section direction to preserve symmetry plane. The added outer layers were fixed, see blue points in Fig. 6.10b.

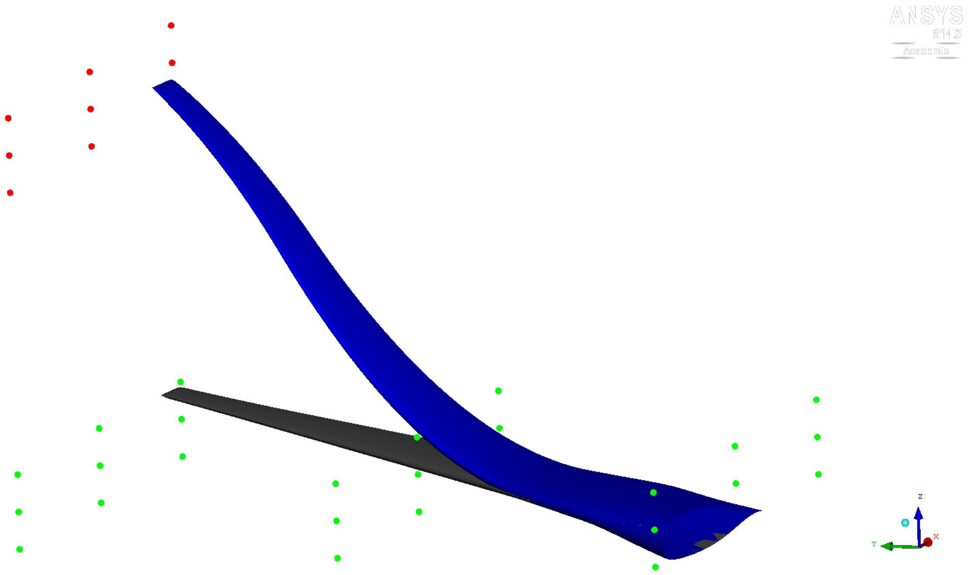


Fig. 6.5: Example of wing tip control points elevation and surface deformation for CFD mesh deformation tests

Results

The maximum achieved mesh deformations by elevation are summed in Tab. 6.6, note that on the Euler mesh the FFD method achieved one order higher elevation in comparison with Laplace method and 37% in comparison with Spring analogy method. The spring analogy method completely failed to deform the RANS mesh as happened previous in 2D tests. Visual inspection is very limited due to the complexity of 3D mesh, nevertheless some cuts of meshes are presented in the Appendix

B (see Fig. B.10 to B.14). Note that it is almost impossible to distinguish between meshes deformed by different methods Fig. B.10, B.11, B.13.

Tab. 6.6: Maximal wing tip section control points elevation

Mesh	Laplace	Spring analogy	FFD
Euler wing (half span = 35.73m)	2 m	12 m	19 m
RANS CRM wing (half span = 3.77m)	0.3 m	failed	4.2 m

The extreme CFD mesh deformation managed by FFD method in the Euler case was achieved by elevation of the tip section control points by more than 50% of the half-span, it is important to say that however distorted the elements may be (see Fig. B.12) all the checking procedures were completed successfully and Edge was able to converge flow solution. That is also the case of RANS mesh, here were the tip section control points elevated by more than 110% and still the deformed mesh went through checks and CFD. Note the high change of the thickness of the prismatic layers between Fig. B.14 and Fig. B.13, which would compromise precision of the CFD solution.

Tab. 6.7 shows number of elements that have worse aspect ration and skewness than 0.05 in Euler deformed meshes with different elevation of the tip section control points.

Note that number of worst skewed elements has not increased. The spring analogy method deformed mesh had only 11 elements with lower aspect ration than the initial mesh in the case of elevation by 2m, which was slightly better than 31 with FFD and 63 with Laplace. 12m elevation gave us comparison of spring analogy and FFD, in which the spring analogy added 220 tapered elements, almost 6 times less than added the FFD method. In the case of 19m elevation with the FFD method the deformed mesh contained 6045 elements with aspect ratio lower than 0.05.

Tab. 6.7: Number of elements that have worse aspect ration and skewness than 0.05 for deformed Euler meshes

Elevation	method	Number of elements	Number of elements
$[m]$	name	with low Sk	with low Ar
0	-	5	696
2	Laplace	5	759
2	Spring A.	5	707
2	FFD	5	727
12	Spring A.	5	916
12	FFD	5	1 978
19	FFD	5	6 045

Tab. 6.8 shows number of elements that have worse aspect ration and skewness than 0.05 in RANS deformed meshes with different elevation of the tip section control points. Here a number of worst skewed elements in the FFD deformed mesh has significantly increased only in the extreme elevation 4.2m case. The number of lowest aspect ration elements obtained in the meshes deformed with Laplace and FFD is very similar to the initial undeformed mesh, where the FFD method is slightly better. In the case of extreme 4.2m elevation with the FFD method the deformed mesh contained a huge number of 120 932 elements with aspect ratio lower than 0.05.

Tab. 6.8: Maximal angle of rotation test, RAE 2822 RANS mesh

Elevation	method	Number of elements	Number of elements
$[m]$	name	with low Sk	with low Ar
0	-	18	4 628
0.3	Laplace	18	4 780
0.3	FFD	18	4 706
4.2	FFD	1251	120 932

The extremely deformed meshes, especially RANS mesh, contain large number of poor quality elements, the maximal elevations were the last valid meshes reported by dual program in the search we must take them as such and judge the for the purpose of testing of the methods not for generating precise CFD results.

Tab. 6.9: CFD results on deformed highly swept wing Euler meshes

Elevation [m]	method	c_L	c_D	c_m
2	Laplace	0.40635	0.02400	-0.31746
2	Spring A.	0.40582	0.02393	-0.31704
2	FFD	0.40518	0.02385	-0.31643
12	Laplace	0.42849	0.03040	-0.33632
12	FFD	0.42394	0.02975	-0.33176
19	FFD	0.42942	0.03428	-0.33458

Tab. 6.9 shows results of the CFD solution in program Edge on highly swept wing Euler mesh case. The same for CRM wing RANS mesh is presented in Tab. 6.10.

Tab. 6.10: CFD results on deformed CRM wing RANS meshes

Elevation [m]	method	c_L	c_D	c_m
0.3	Laplace	0.30288	0.01450	-0.08603
0.3	FFD	0.30218	0.01449	-0.08549
4.2	FFD	0.32159	0.02156	-0.06401

The results of CFD calculations, obtained for the different CFD mesh deformation methods compared on the same elevation, deviates at most by 2% in Euler mesh cases and at most by 0.6% in RANS mesh cases.

6.3.5 Efficiency: CPU time and memory demands:

The computational efficiency of CFD mesh deformation method is an important factor that is also affecting the whole aerodynamic shape optimization process cost. In the case of CRM wing optimization (Tab. 4.2 in section 4.3.1) just the CFD mesh deformation needs approximately 32 % of the whole real time.

CPU time:

The CPU time is measured on deformations of 3D Euler and RANS meshes by using single CPU core. Each the deformation is repeated 5 times to suppress random effects that can occur as the procedures are conducted in Matlab environment. The standard deformation methods are again executed in meshdeform Edge package subprogram which is written in Fortran code, the FFD procedure is written in C code.

The FFD specifics is that the mesh needs to be first embedded into the FFD lattice which is a very expensive process, however that is needed only once. Tab. 6.11 shows results of measurements of CPU time. In the case of Euler mesh a rough agreement between time of Laplace and FFD methods can be observed, the spring analogy being several times slower. The RANS mesh deformation compares only Laplace and FFD methods, since the spring analogy failed in the process, the Laplace method being slightly faster even if we do not include the time needed for embedding.

Tab. 6.11: CPU time demands of different mesh deformation techniques

Mesh	Laplace	Spring analogy	FFD	FFD embedding
Euler (77k nodes)	7,1 s	56,9 s	4,3 s	9,7 s
RANS (965k nodes)	42,6 s	failed	51,0 s	166,5 s

Fig. 6.6 shows dependency of CPU time demands on increasing elevation of the tip section control points that deforms the Euler mesh. Note that the embedding procedure time was included only at the first run, the following calculations of the deformation is not dependent on the amplitude of the elevation (as expected since the FFD does not depend on the mesh topology). The standard methods seems to be little sensitive to the elevation amplitude as well, that is probably caused by this specific deformation test and does not fully reflect the usual behavior.

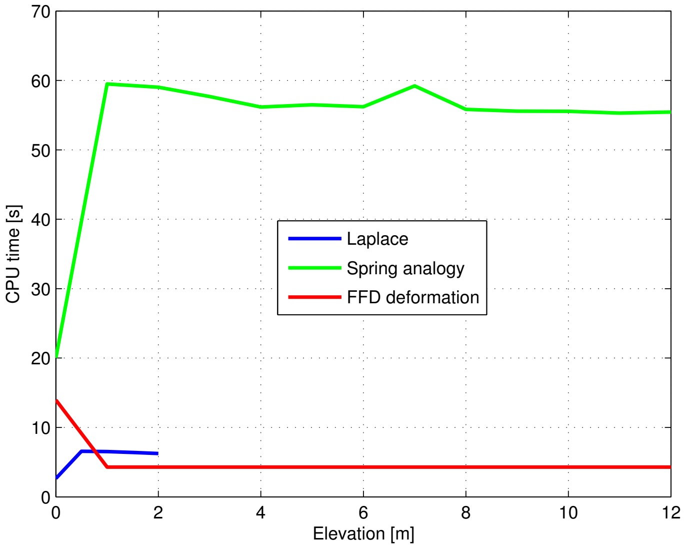


Fig. 6.6: CPU time demands for deformation of Euler mesh

Fig. 6.7 shows the same dependency on the RANS mesh. Again the FFD embedding time is added only once and again the following deformations takes about the same time. The Laplace procedure is slightly faster in most of the tested deformations.

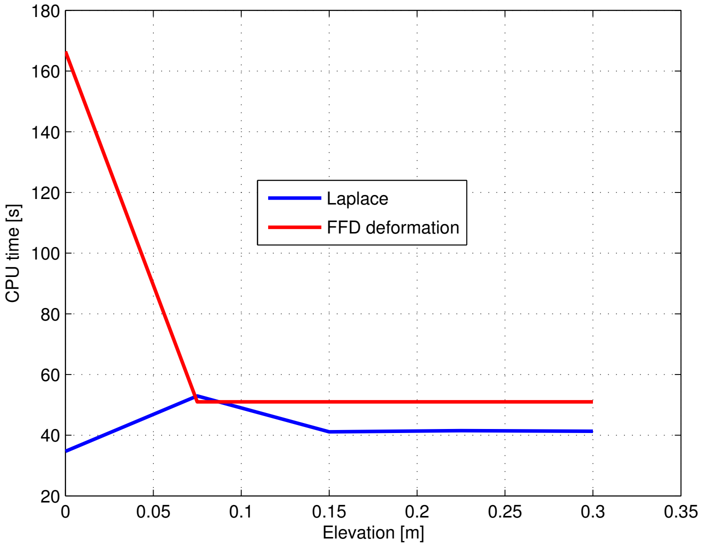


Fig. 6.7: CPU time demands for deformation of RANS mesh

Parallelization: As stated above, all the test were calculated using single CPU core, that was done just for the comparison purposes. The FFD method is topology independent by definition, in other words it treats every point individually. That means that the parallelization of the deformation task is theoretically limited only by number of mesh nodes and it is very easy to implement the parallelization from the FFD point of view. That makes it appropriate for use on multi-core processors or on computer clusters. Translation of the code into e.g. CUDA code would enable further massive parallelization for the use on GPUs.

Dependency of CPU mesh size on CPU time was analyzed for single CPU and 4 CPU core processor in Matlab environment and is presented in Fig. 6.8.

The parallel calculations using 4 CPUs were 3 times in faster average than the single CPU calculations, while the bigger the grid the more effective the use of multiple cores were (see Fig. 6.9).

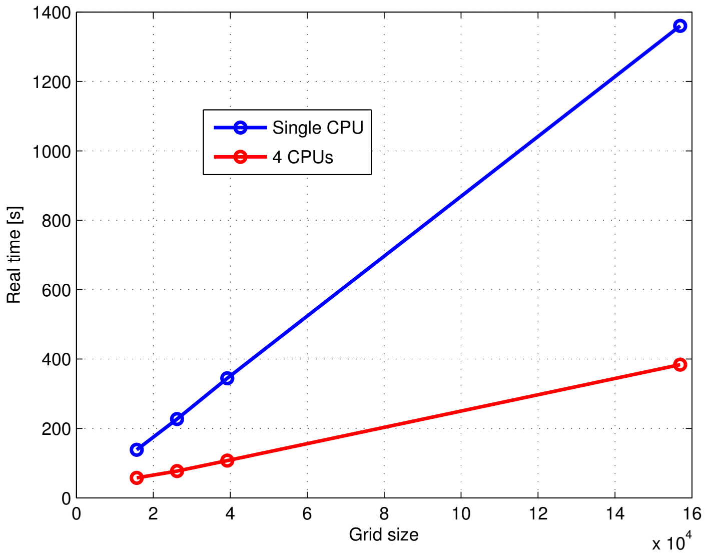


Fig. 6.8: Comparison of Single and parallel deformation time dependency on CFD grid size

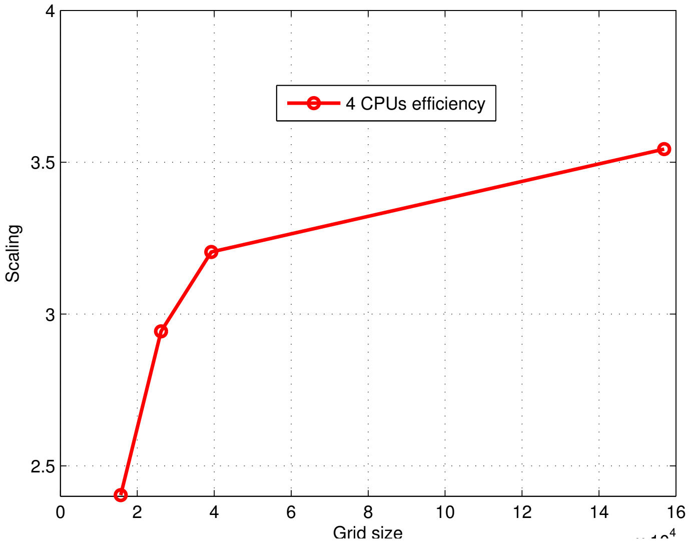


Fig. 6.9: Efficiency of use of 4 CPUs with respect to CFD grid size

Memory demands:

Procedure that is using meshdeform program from edge is operating with file (.bdis file) that contains information about surface nodes displacements, size of this file is used here for comparison of the demands of the FFD method.

The only important file worth saving in the FFD method for CFD mesh deformation is the NURBS matrix that contains coordinates of the object in the parametric space, a result of the embedding of the object into the FFD lattice. The size of the matrix depends on the number of control points in x,y and z direction and on number of mesh nodes.

Tab. 6.12: Memory demands of different mesh deformation techniques

Mesh	Nodes	FFD lattice	NURBS file	.bdis file
Euler	77k	7x7x7	27.8 MB	272 kB
RANS	965k	11x10x5	352 MB	909 kB

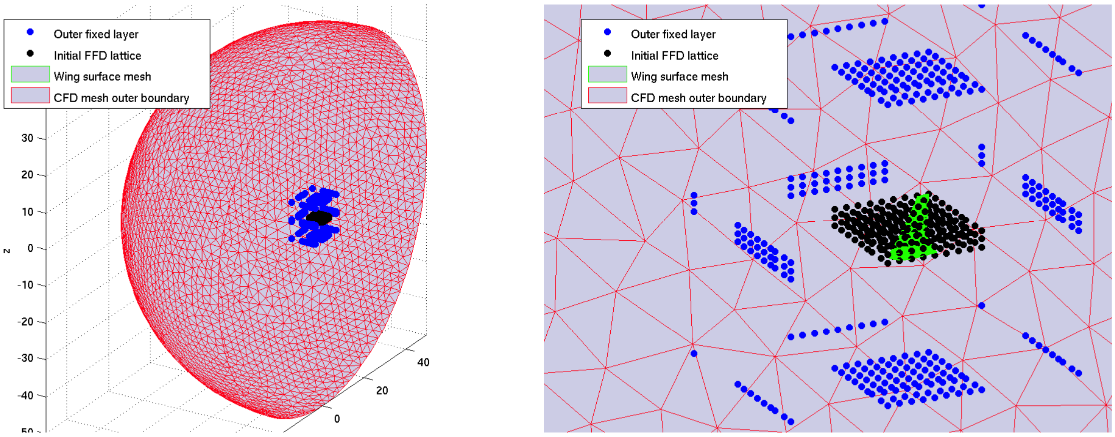
As can be seen in Tab. 6.12 the memory needed for NURBS matrix is huge in comparison to the .bdis file size. The NURBS file is 104.6 times bigger in the Euler mesh case and 396.5 times bigger in the case of RANS mesh. If we extrapolate the size of the RANS mesh to 50 million the NURBS file would probably need 18.2 GB of memory.

6.4 3D Aerodynamic shape optimization using FFD for CFD mesh deformation

The use of FFD parameterization for mesh deformation approach described above is used in CRM wing optimization test case 4.3.1, in which one of the conclusions was that the optimization stopped due to inability of the CFD mesh deformation tool to modify the CFD mesh around demanded surface shape. The case description allow shape deformations of the root section of the wing that is located in the symmetry plane of the wing. Since the optimization variables are vertical displacements of the control points no special care needs to be taken in the symmetry plane. The only modification to the CFD mesh deformation procedure is that no additional layer above the initial FFD that is constructed around the wing is created in the symmetry plane, that allows the CFD mesh nodes in that plane to slide freely without compromising the mesh quality.

6.4.1 Basic FFD

Here a comparison of optimization cases with Laplace and FFD CFD mesh deformation approach on basic FFD parameterization without the RBF coordinate transformation is given. As can be seen on Fig. 6.10a only part of the whole CFD mesh that is located inside the FFD lattice (blue outer layer) is selected to be deformed, that can be seen on Fig. 6.10b which shows detail of the layout of the FFD lattice for CFD mesh deformation. The gap between the initial and outer layer defines how steep will the deformation of the mesh propagate from the displaced control points. The outer fixed layer of control points should be placed in the area where the elements are big with respect to the ones at the surface of the object. The gap is specific to each volume mesh, here a distance of one half span of the wing is used as most of the small elements are located close to the surface of the wing. That is true just for unstructured meshes, in the case of structured meshes tuning of the gap size can be advised. Only the black control points of the initial FFD lattice (constructed around the wing) are allowed do move.



(a) Overview

(b) Detail

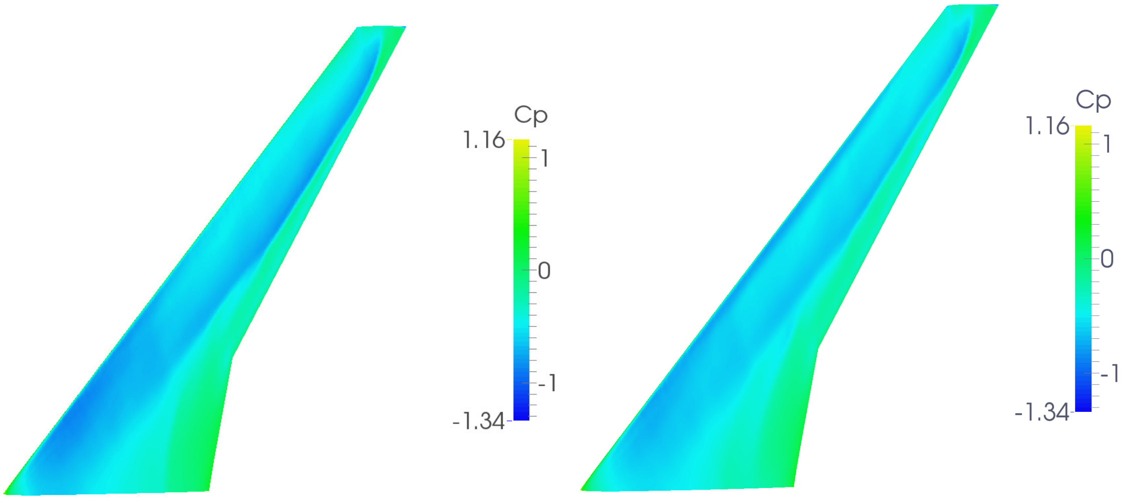
Fig. 6.10: FFD lattice layout for the CFD mesh deformation

The use of FFD both for surface and CFD mesh deformation resulted in 7.4 % better results compared to FFD for surface and Laplace for CFD mesh, see Tab. 6.14. With the use of FFD the optimization stopped after reaching convergence criteria.

Fig. 6.11 shows that the case with Laplace CFD mesh deformation was not able to get rid of shock waves as good as the case with FFD CFD mesh deformation.

Tab. 6.13: Comparison of CRM wing optimization results with different mesh deformation techniques

	Laplace	FFD
c_D baseline	0.017973	0.017973
c_D optimal	0.015079	0.013748
c_D reduction	16.1 %	23.5 %
Cost in CFD+adjoint iterations	27	24



(a) Laplace

(b) FFD

Fig. 6.11: Comparison of pressure coefficient distributions on CRM wing - top view

A huge difference can be observed in the resulting shape of the wing, see Fig. 6.12. The FFD allowed the optimizer to do much bigger deformations which led to odd wave created on the leading edge of the wing. That shape is probably caused by the use of basic FFD parameterization without the RBF mapping, that means that the optimizer had insufficient control over the deformations of the wing sections..

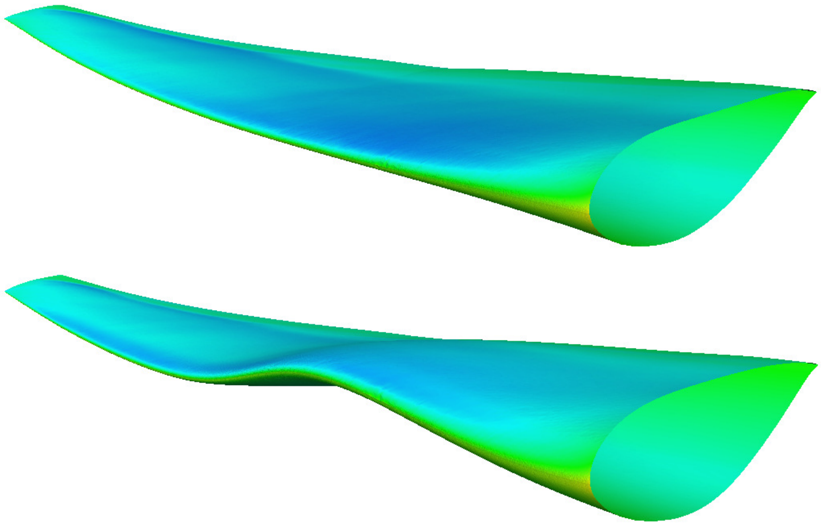


Fig. 6.12: Comparison of CRM wing shapes initial with the use of Laplace (top) and FFD (bottom) CFD mesh deformation techniques

Fig. 6.13 shows comparison of wing section shapes of initial wing geometry and wing optimized with basic FFD using Laplace and FFD CFD mesh deformation methods. Again the odd (wave-like)shape can be observed in some section shapes.

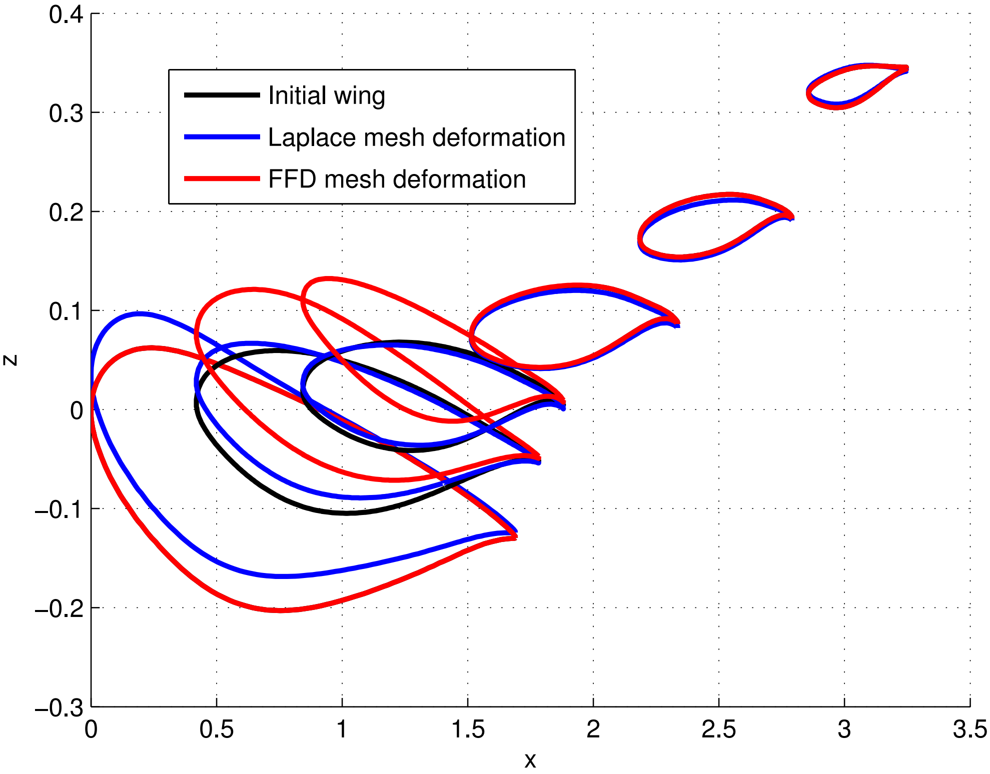


Fig. 6.13: CRM wing section shapes (axis not in scale)

6.4.2 FFD with RBF coordinate transformation

As illustrated above the optimizer is dependent on used parameterization, the RBF coordinate transformation described in section 4.2 was incorporated into the CFD mesh deformation procedure in order to allow the use of FFD-RBF for both surface and CFD mesh deformations. Fig. 6.14 shows only part of the whole initial CFD mesh (green points) that is located inside the FFD lattice (blue outer layer), Fig. 6.15 depicts the FFD-RBF parameterization layout for the CFD mesh deformation and mesh nodes after the RBF coordinate transformation. Only the red control points of the initial parallelepiped FFD lattice are allowed do move.

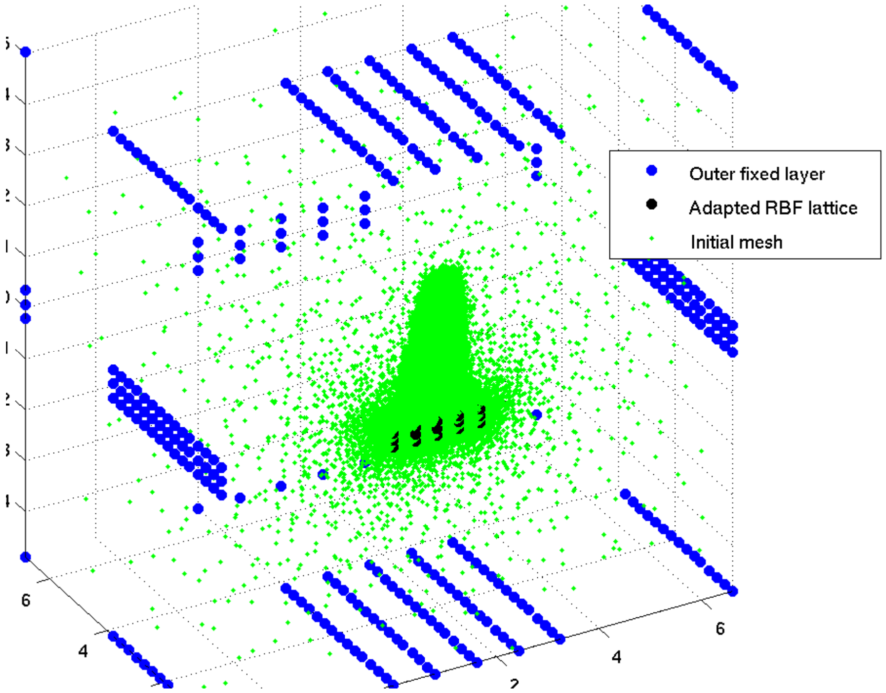


Fig. 6.14: FFD-RBF lattice layout for the CFD mesh deformation, before RBF coordinate transformation

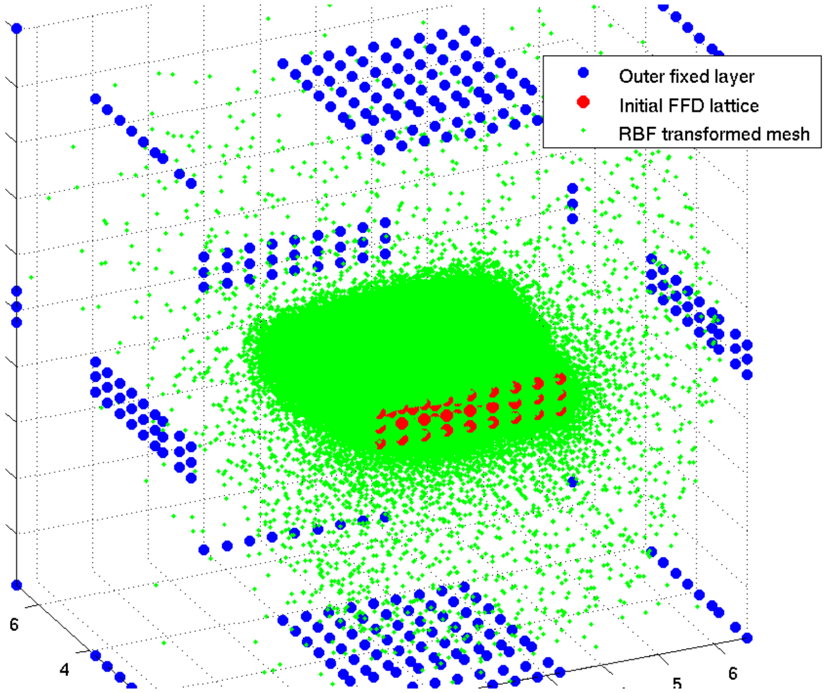


Fig. 6.15: FFD-RBF lattice layout for the CFD mesh deformation, after RBF coordinate transformation

With the use of FFD the optimization stopped after reaching prescribed maximum number of optimization iterations. The Fig. 6.16 shows that the optimizer was not able to fulfill the lift constraint and that it did most of the optimization iterations trying to reach it.

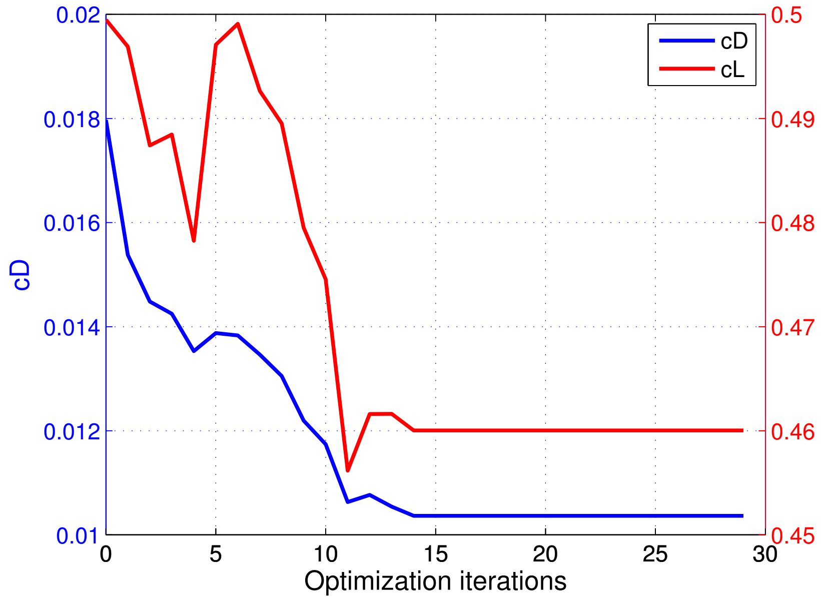


Fig. 6.16: History of c_D and c_L during the optimization

Tab. 6.14 shows that the optimum given as a result by the NLPQLP software was 5.4 % worse compared to the optimum in the case with Laplace CFD mesh deformation tool. What is more interesting is that the results in the column named FFD-RBF violated c_L that show the values for minimal c_D, there the FFD-RBF case outperformed the Laplace case by 10.4 % producing better glide ratio (39 vs. 43.1).

Tab. 6.14: Comparison of CRM wing optimization results with different mesh deformation techniques

	Laplace	FFD-RBF	FFD-RBF violated c_L
c_D baseline	0.017973	0.017973	0.017973
c_D optimal	0.012874	0.013832	0.010993
c_D reduction	28.4 %	23 %	38.8 %
c_L baseline	0.502438	0.499089	0.473289
Glide ratio	39.0	36.1	43.1
Cost in CFD+adjoint iterations	43	24	51

Regardless of the behavior of the optimization algorithm which is sensitive to various phenomena, the FFD-RBF showed its potential in its role of CFD mesh deformation tool. That is also illustrated in Fig. 6.17, note the distinctive difference in shape of the wing tips, which tells us that the FFD-RBF is capable of much bigger deformations.

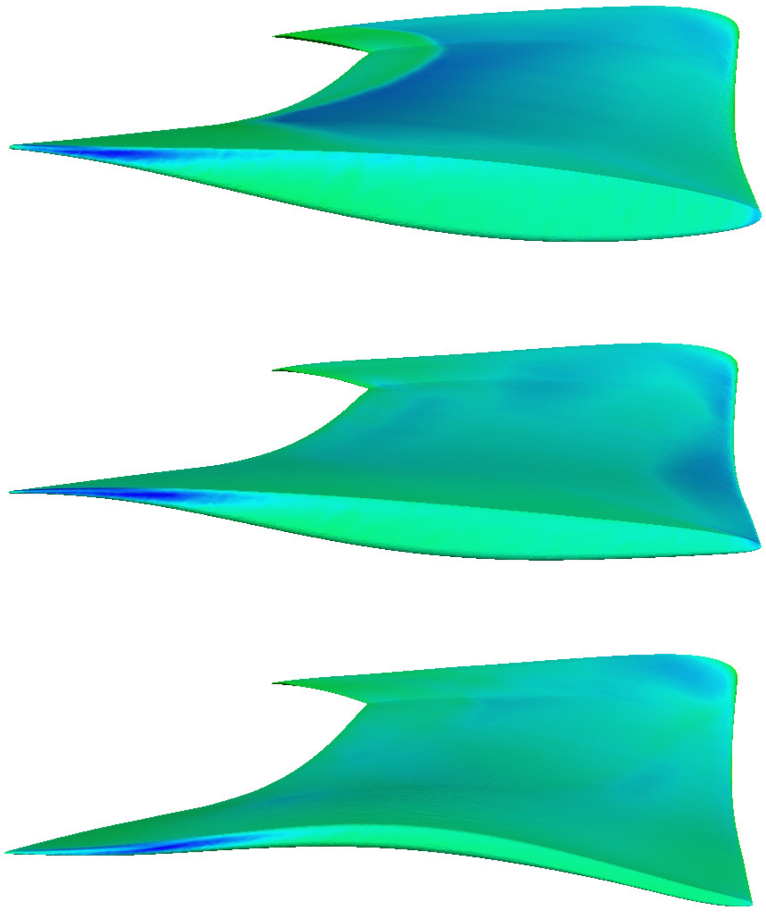


Fig. 6.17: Comparison of CRM wing: initial (top) optimized with the use of Laplace (middle) and FFD-RBF violated c_L (bottom) CFD mesh deformation techniques

Fig. 6.18 shows comparison of wing section shapes of initial wing geometry and wing optimized with FFD-RBF using Laplace and FFD-RBF (violated) CFD mesh deformation methods. That again illustrates haw capable the FFD-RBF method is in achieving big CFD mesh deformations

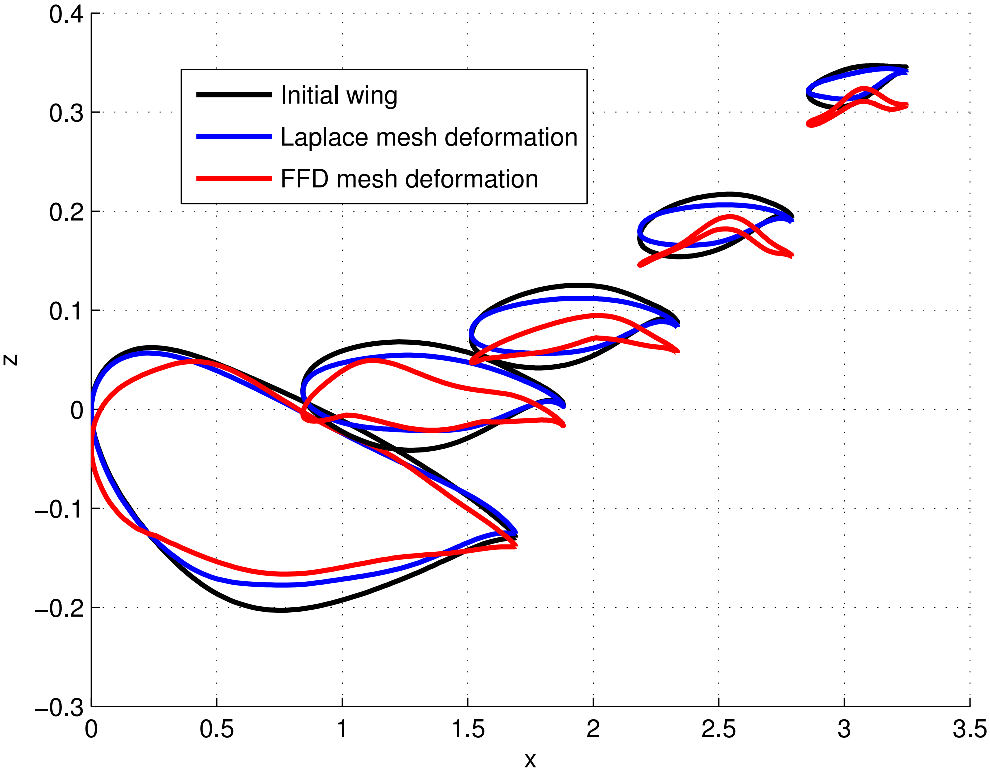


Fig. 6.18: CRM wing section shapes (axis not in scale)

7 OUTCOMES OF THE DOCTORAL THESIS

This chapter describes accomplished goals of the thesis and notable facts about developed parameterization method.

7.1 Free-Form Deformation (FFD) parameterization

NURBS based FFD parameterization properties were identified and tested. The most important outcomes are:

- Impact of NURBS degree: The NURBS degree affects FFDs geometry handling characteristics. It was illustrated on straight line deformations and on inverse geometry optimization, where higher NURBS degree practically damps oscillations. The results of NACA 0012 airfoil optimization shows that the in increasing the NURBS degree not only improved the cost function but also accelerated convergence. The acceleration of convergence was also observed in 3D on passenger aircrafts wing optimization.
- Dimensionality: A parametric study both on NACA 0012 airfoil optimization and on passenger aircrafts wing optimization concluded that the bigger the number of parameters the better the results. Investigation of added weights and multi-directional displacements on NACA 0012 airfoil showed that their use is not effective, but can be beneficial in the cases where the FFD lattice cannot be altered.

7.2 Adaptive FFD parameterization with respect to geometry

Adaptivity of the FFD parameterization to the geometry was achieved by using RBF coordinate transformation. The motivation was to enable better control of the deformations and thus further improve the optimum. The other motivation was to handle complex geometrical constraints imposed on the optimization problem.

- The FFD-RBF greatly improves FFD's geometry handling capabilities, which was proven in to cases with complex geometrical constraints. In the case of CRM wing trailing edge fixation, and in the case of EV-55 Outback commuter plane nacelle boundary curve fixation.
- The benefit of RBF mapping is also in the aerodynamic shape optimization behavior that profits from the improved geometry handling as was observed

in CRM wing optimization. as well as in the optimization of EV-55 Outback commuter aircrafts landing gear nacelle.

7.3 Adaptive FFD parameterization with respect to optimization

Adaptivity of developed parameterization method during the optimization process was investigated using Enrichment and Multi-grid methods. These test were aiming on acceleration of the aerodynamic shape optimization process.

- The Enrichment procedure was inspected on NACA 0012 test case, with the conclusion that the FFD parameterization is not sensitive on location of control points as on its quantity.
- The benefits of using Multi-grid approach to the optimization of CRM wing is rather inconclusive. Some savings of CPU time were observed, but the effort to expose the acceleration properties of the Multi-grid was corrupted by failing CFD mesh deformation tool.

7.4 FFD for CFD mesh deformations

Free-Form Deformation parameterization method was used in the CFD mesh deformation application. Unlike the standard methods which adjusts the volume mesh to the shape changes of the surface, the FFD was applied both to surface and volume mesh deformations simultaneously which is suitable for aerodynamic shape optimization process. The smooth volume deformations capabilities of NURBS-based FFD method as well as its independency of the mesh topology makes it appropriate for CFD mesh deformations.

- The FFD method was successfully tested on deformations by rotation in 2D and by bending in 3D on both Euler and RANS meshes. The FFD method surpassed the capabilities of Laplace and Spring analogy standard methods in all test in terms of maximal achievable deformation.
- The qualitative comparison showed no deficiencies in visual evaluation, in calculated aspect ratios and skewness. Obtained converged CFD flow solutions show almost identical results between the methods.
- The tests of efficiency of the FFD method in terms of CPU time needed for the mesh deformation results are comparable to the results of Laplace method and faster than Spring analogy. The expensive embedding procedure is needed only once, the file that contains NURBS matrix (a result of the embedding) needs significant disk space (350MB for 1M node mesh).

- The test of use of the FFD method for CFD mesh deformation during the aerodynamic shape optimization process gave promising results. The FFD was capable of bigger deformations and found better optimum. The FFD enhanced by the RBF coordinate transformation enabled the optimizer to make bigger deformations that the standard methods and is therefore perspective for further development.
- The time expensive embedding part of the FFD procedure can be parallelized.

Possible future applications of FFD: The developed algorithms could be used in the field of aero-elasticity, for coupling CFD with FEM and during time dependent deformations.

8 CONCLUSIONS

The doctoral thesis *Adaptive parameterization for aerodynamic shape optimization in aeronautical applications* is focused on practical problems of parameterization and its use in aerodynamic shape optimization in particular.

As the primary goal of the thesis an adaptive FFD parameterization method for applications in the field of aircraft design was developed and verified. A method that could automatically adapt the original parameterization, and that would be able to handle complex geometry deformations and demands on complicated geometrical constraints.

Developed Free-Form Deformation parameterization is capable of accurate embedding of complex geometry in orthogonal lattices (with the use of RBF coordinate transformation), which was verified on 2D and 3D aerodynamic shape optimization cases. It is also competent of handling complicated constraints that are often necessary in industrial applications.

As the secondary goal, a technique based on FFD for deformations of CFD computational meshes was developed. The FFD is capable of required CFD mesh deformations, quality and effectivity of such use of the FFD was tested. The adaptivity to the geometry of the FFD-RBF was also incorporated into the CFD mesh deformation procedure and its benefits were proven on working aerodynamic shape optimization of wing.

Developed FFD-RBF parameterization method was incorporated into automatized environment for aerodynamic shape optimizations and is ready for use in aeronautical applications ranging from simple 2D airfoils to complex constrained 3D surfaces.

BIBLIOGRAPHY

[1] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," *Computer Graphics*, vol. 20(4), pp. 151–160, 1986.

[2] X. Han and D. Zingg, "An adaptive geometry parameterization for aerodynamic shape optimization," *Optimization and Engineering*, vol. 15, pp. 69–91, 2014.

[3] J. A. Samareh, "Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization," *AIAA Journal*, vol. 39, pp. 877–884, 2001.

[4] M. Andreoli, A. Janka, and J.-A. Désidéri, "Free-form-deformation parameterization for multilevel 3d shape optimization in aerodynamics," *INRIA Rapport de recherche no 5019*, 2003.

[5] D. Siegr, S. Menzel, and M. Botsch, "A comprehensive comparison of shape deformation methods in evolutionary design optimization," *EngOpt 2012 : International Conference on Engineering Optimization*, 2012.

[6] R. Raffin, "Free form deformation or deformations non-constrained by geometries or topologies," *Lecture notes in Computational Vision in Biomechanics 7*, 2013.

[7] H. J. Lamousin and W. N. Waggenspack, "Nurbs-based free-form deformations," *IEEE Computer Graphics and Applications*, vol. 14, pp. 59–65, 1994.

[8] E. I. Amoiralis and I. K. Nikolos, "Freeform deformation versus b-spline representation in inverse airfoil design," *Journal of Computing and Information Science in Engineering*, 2008.

[9] M. Lombardi, A. Parolini, N. Quarteroni, and G. Rozza, "Numerical simulation of sailing boats: dynamics, fsi, and shape optimization," *MOX-Report*, vol. 13, 2011.

[10] S. Nadarajeh, "Aerodynamic design optimization: Drag minimization of the naca 0012 in transonic inviscid flow," *AIAA Aerodynamic Design Optimization Discussion Group*, 2013.

[11] L. Osusky and D. Zingg, "Lift-constrained drag minimization of a wing allowing section and twist variation with flow governed by the reynolds-averaged navier-stokes equations," 2013. AIAA Aerodynamic Design Optimization Discussion Group.

[12] Z. Lyu and J. R. R. A. Martins, “Aerodynamic shape optimization of a blended-wing-body aircraft,” in *Proceedings of the 51st AIAA Aerospace Sciences Meeting*, (Grapevine, TX), AIAA, AIAA, 01/2013 2013. AIAA 2013-0283.

[13] P.-K. E. M. Othmer, C. and K. Haliskos, “Cfd optimization via sensitivitiy-based shape morphing.,”

[14] N. Ali and K. Behdinan, “Conceptual aircraft design - a genetic search and optimization approach,” *ICAS 2002 Congress*, 2002.

[15] J. A. Samareh, “Aerodynamic shape optimization based on free-form deformation,” *AIAA Paper No. 2004-4630*, 2004.

[16] J. A. Samareh, “Geometry and grid/mesh generation issues for cfd and csm shape optimization,” *Optimization and Engineering*, vol. 6.

[17] P. Frank and G. Shubin, “A comparison of optimization-based approaches for a model computational aerodynamics design problem,” *Journal of Computational Physics*, vol. 98, pp. 74–89, 1992.

[18] L. M. A. Wong, E. S. and N. Qin, “Parallel adjoint-based optimization of a blended wing body aircraft with shock control bumps,” *The Aeronautical Journal*, 2006.

[19] J. E. Hicken and D. W. Zingg, “Integrated parametrization and grid movement using B-spline meshes,” in *The 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, no. AIAA–2008–6079, (Victoria, British Columbia, Canada), Sept. 2008.

[20] Y. S. Wu, H-S. and F. Liu, “Comparison of three geometric representation of airfoils for aerodynamic optimization,” *AIAA Paper 4095*, 2003.

[21] Y. J. W. Lee, B. J. and S.-H. Yoon, “Aerodynamic shape optimization using overset mesh technique for multiple body aircraft geometries..”

[22] C.-h. Sung and K. J. H., “Efficient aerodynamic design method using a tightly coupled algorithm,” *AIAA Journal*, vol. 39, no. 9, 2002.

[23] G. Carpentieri, *An adjoint-based shape-optimization method for aerodynamic design*. PhD thesis, 2009.

[24] A. O., “Towards high fidelity multidisciplinary design of air-craft components.,”

[25] A. O., “Efficient aerodynamic design method using a tightly coupled algorithm,” *AIAA Journal*, vol. 39, no. 9, 2002.

[26] X. M. C. T. Guo, M. and X. An, “A mesh deformation method for aeroelastic simulation,”

[27] H. Jasak and Z. TukoviÄ‡, “Automatic mesh motion for the unstructured finite volume method,” 2004.

[28] S. Jakobsson and O. Amoignon, “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization,” *Computers & Fluids*, vol. 36, pp. 1119–1136, 2007.

[29] C. C.-L. Hsu, S-Y. and J. A. Samareh, *A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software.* BiblioBazaar, 2013.

[30] S. O. Boubekeur, T. and C. Schlick, “Simod: making freeform deformation size-intensive,”

[31] D. C. Berkenstock and M. J. Aftosmis, “Structure-preserving parametric deformation of legacy geometry,” *AIAA Paper 2008-6026*, 2008.

[32] J.-A. Désidéri and A. Janka, “Multilevel shape parameterization for aerodynamic optimization application to drag and noise reduction of transonic/supersonic business jet,” *ECCOMAS 2004*, 2004.

[33] J. A. Samareh, “Multidisciplinary aerodynamic-structural shape optimization using deformation (massoud),” *AIAA Paper No. 2000-4911*, 2000.

[34] O. Ronzheimer, *Shape Parameterization Using Freeform Deformation*, vol. 89, pp. 211–222. Springer Berlin Heidelberg, 2005.

[35] S. Coquillart, “Extended free-form deformation: A sculpturing tool for 3d geometric modeling,” *Computer Graphics*, vol. 24(4), pp. 187–193, 1990.

[36] E. W. Perry and R. Balling, “A new morphing method for shape optimization,” *AIAA paper*, pp. 1510–1519, 1998.

[37] B. Abou el Majd, R. Duvigneau, and J.-A. Désidéri, “Aerodynamic shape optimization using a full and adaptive multilevel algorithm,” *Project Team Opale*, 2004.

[38] R. Duvigneau, “Adaptive-parameterization-using-free-form deformation for aerodynamic shape optimization,” *INRIA Rapport de recherche no 5949*, 2006.

[39] R. Duvigneau, B. Chaigne, and J.-A. Désidéri, “Multi-level parameterization for shape optimization in aerodynamics and electromagnetics using a particle swarm optimization algorithm,” *INRIA Rapport de recherche no 6003*, 2006.

[40] R. Duvigneau, B. Abou el Majd, and J.-A. Désidéri, "Towards a self-adaptive parameterization for aerodynamic shape optimization," *ESAIM: PROCEEDINGS*, vol. 22, pp. 169–174, 2007.

[41] W. M. Hsu, H. J. F., and K. H., "Direct manipulation of free-form deformations," *Computer Graphics*, vol. 26(2), pp. 177–184, 1992.

[42] R. MacCracken and K. I. Joy, "Free-form deformations with lattices of arbitrary topology," *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 1881–188, 1996.

[43] J. Feng, L. Ma, and G. Peng, "A new free-form deformation through the control of parametric surfaces," *Computers and Graphics*, vol. 20, pp. 531–539, 1996.

[44] J. Feng, P.-A. Heng, and T.-T. Wong, "Accurate b-spline free-form deformation of polygonal objects," *Journal of Graphics Tools*, vol. 3, pp. 11–27, 1998.

[45] C. Chua and U. Neumann, "Hardware-accelerated free-form deformation," 2000.

[46] M. et al., "Fast free-form deformation using graphics processing units," *Computer Methods and Programs in Biomedicine*, vol. 98 (3), pp. 278–284, 2010.

[47] J. E. Gain and N. A. Dodgson, "Adaptive refinement and decimation under free-form deformation," *Eurographics UK 17th Annual Conference*, 1999.

[48] A. Sacharov, T. Surmannm, and D. Biermann, "Adaptive free-form deformation for the modification of cad/cam data," *ADVCOMP 2011 : The Fifth International Conference on Advanced Engineering Computing and Applications in Sciences*, 2011.

[49] H. Gagnon and D. Zingg, "Two-level free-form deformation for high-fidelity aerodynamic shape optimization," *AIAA Paper 2012-5447*, 2012.

[50] J. Wang and T. Jiang, "Nonrigid registration of brain mri using nurbs," *Pattern Recogn. Lett.*, vol. 28, pp. 214–223, Jan. 2007.

[51] G. Hirota, R. Maheshwari, and M. C. Lin, "Fast volume-preserving free form deformation using multi-level optimization," in *Proceedings of the fifth ACM symposium on Solid modeling and applications*, SMA '99, (New York, NY, USA), pp. 234–245, ACM, 1999.

[52] Y. Ono, B. yu Chen, T. Nishita, and J. Feng, "Free-form deformation with automatically generated multiresolution lattices," in *Proceedings of the IEEE Cyberworlds 2002 Conference*, pp. 472–479, 2002.

[53] S. Ilic and P. Fua, "Using dirichlet free form deformation to fit deformable models to noisy 3-d data," in *In European Conference on Computer Vision*, pp. 704–717, 2002.

[54] J. Hua and H. Qin, "Free-form deformations via sketching and manipulating scalar fields," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*, SM '03, (New York, NY, USA), pp. 328–333, ACM, 2003.

[55] K. G. Kobayashi and K. Ootsubo, "t-ffd: free-form deformation by using triangular mesh," in *Proceedings of the eighth ACM symposium on Solid modeling and applications*, SM '03, (New York, NY, USA), pp. 226–234, ACM, 2003.

[56] W. Song and X. Yang, "Free-form deformation with weighted t-spline," *The Visual Computer*, vol. 21, 2005.

[57] K. T. McDonnell and H. Qin, "Pb-ffd: A point-based technique for free-form deformation," *Journal of Graphics Tools*, vol. v12 i3, pp. 25–41, 2006.

[58] X. Battle, Y. Bizais, C. Le Rest, and A. Turzo, "Tomographic reconstruction using free-form deformation models," *SPIE Conference on Image Processing*, vol. 3661, 1999.

[59] D. Mattes, D. R. Haynor, H. Vesselle, L. T. K., and W. Eubank, "Pet-ct image registration in the chest using free-form deformations," *IEEE Transactions on Medical Imaging*, vol. 22, 2003.

[60] S. Menzel, M. Olhofer, and B. Sendhoff, "Application of free form deformation techniques in evolutionary design optimisation," 2005.

[61] T. Economon, F. Palacios, and J. Alonso, "Optimal shape design for open rotor blades," *AIAA Paper 2012-3018*, 2012.

[62] F. Palacios, J. Alonso, M. Colonno, A. Aranake, A. Campos, S. Copeland, T. Economon, A. Lonkar, T. Lukaczyk, and T. Taylor, "Stanford university unstructured (su2): An open source integrated computational environment for multiphysics simulation and design," *AIAA Paper 2013-0287*, 2013.

[63] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins, "A cad-free approach to high-fidelity aerostructural optimization," in *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, (Fort Worth, TX), September 2010. AIAA 2010-9231.

[64] D. Pinelli and G. Sereno, "Constrained multipoint aerodynamic optimization of a transonic bussines jet wing," in *XVII Congresso Nationale AIDAA*, 2005.

[65] K. Hatanaka, J. Brezillon, and A. Ronzheimer, “A fully automated cad-based framework for shape optimization,” in *ICAS 2012 28th international congress of the aeronautical sciences*, 2012.

[66] F. Palacios, J. Alonso, M. Colonno, J. Hicken, and T. Lukaczyk, “Adjoint-based method for supersonic aircraft design using equivalent area distribution,” *AIAA Paper 2012-0269*, 2012.

[67] R. Cimrman and E. Rohan, “On shape optimization of conduits with incompressible flow,” in *Applied and Computational Mechanics I*, pp. 393–400, 2007.

[68] Y. G. YAO ShuanBao, GUO DiLong, “Three-dimensional aerodynamic optimization design of high-speed train nose based on ga-grnn,” *SCIENCE CHINA Technological Sciences*, vol. 55, no. 11, p. 3118, 2012.

[69] J. Vassberg, N. Harrison, D. Roman, and A. Jameson, “A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem,” in *29th AIAA Applied Aerodynamics Conference, Honolulu, Hawaii*, no. AIAA 2011-3176, 2011.

[70] S. G. Johnsonm, *The NLopt nonlinear-optimization package.*

[71] O. Amoignon, J. Navrátil, and J. Hradil, “Study of parameterizations in the project cedesa,” *AIAA Paper No. 2014-0570*, 2014.

[72] P. Eliasson, *Edge, A Navier-Stokes Solver, for Unstructured Grids*, November 2001.

[73] K. Schittkowski, *NLPQLP: A Fortran Implementation of a Sequantial Programming Algorithm with Distributed and Non-Monotone Line Search - User’s Guide, Version 4.2*, 2014.

[74] J. I. Gargoloff and P. G. A. Cizmas, “Mesh generation and deformation algorithm for aeroelasticity simulations,” *Journal of Aircraft*, vol. 45, no. 3, 2008.

LIST OF FIGURES

2.1 Key processes of aerodynamic shape optimization 13

2.2 Example of PARSEC parameterization 15

2.3 Example of 3D deformed mesh (Source:[29]) 18

3.1 Example of Bézier curve 21

3.2 Basic principle of the use of FFD parameterization for deformation . 24

3.3 1D FFD two ways of deformation of a line 29

3.4 Local deformation of airfoil using 2D FFD (Global and close-up view) 30

3.5 Local deformation of wing tip using 3D FFD (Global and close-up view) 31

3.6 Deformation of complicated 2D object with displacement of one FFD lattice control point 32

3.7 Deformation of complicated 3D object with displacement of one FFD lattice control point 32

3.8 NURBS degree influence on the deformed area 33

3.9 FFD parameterization of complex 3D object 34

3.10 Overlapping FFD lattices in 3D 34

3.11 Deformation of RAE 2822 airfoil with two adjacent FFD lattices . . . 35

3.12 Oscillation influenced by NURBS degree 36

3.13 Two examples of inverse design: towards a diamond-shaped airfoil (left) and towards a smooth airfoil (right). 37

3.14 Cost of optimization dependency on the NURBS degree - towards a diamond-shaped airfoil (left) and towards a smooth airfoil (right). . . 38

3.15 NACA 0012 mesh with 42556 nodes 40

3.16 Example of FFD parameterization setup for the case with 6 variables 40

3.17 NACA: History of optimization for various number of design variables (maximum NURBS degree is used in all cases). Only the feasible steps are shown. 41

3.18 NACA 0012: Comparison of optimal shapes for various number of design variables and distributions of pressure coefficients. 42

3.19 NURBS degree influence on the optimization of the NACA0012 airfoil with 6 parameters. 44

4.1 A coarse FFD lattice. 47

4.2 FFD lattice constructed around wing 49

4.3 RBF coordinate transformation example 49

4.4 Dense RBF coordinate transformation lattices 50

4.5 Comparison of basic FFD parameterization and FFD with dense RBF coordinate transformation 50

4.6 Geometry of the CRM wing, top and back view 54
4.7 Comparison of parameterizations of CRM wing geometry 55
4.8 Mesh of the CRM wing 55
4.9 Comparison of optimization history, using basic FFD and FFD-RBF parameterizations 56
4.10 Comparison of pressure coefficient distribution on CRM wing 57
4.11 CRM wing section shapes (axis not in scale) 57
4.12 RBF adapted lattice on the Passenger aircraft 59
4.13 Mesh of the Passenger aircraft 59
4.14 Comparison of optimization history, using different number of parameters 61
4.15 EV-55 Outback. Image downloaded from `http://www.evektor.cz/outback/fotogalerie.aspx` 63
4.16 Transparent view of the landing gear nacelle 64
4.17 EV-55 Outback CFD mesh 64
4.18 EV-55 Outback detail of mesh on the landing gear nacelle 65
4.19 EV-55 Outback Parameterization example 65
4.20 FFD parameterization with RBF coordinate transformation processes 66
4.21 Comparison of initial (top) and optimal (bottom) shapes and pressure coefficient distribution in cruise conditions 68
4.22 Comparison of initial (top) and optimal (bottom) shapes and pressure coefficient distribution in climb conditions 68
5.1 One step of enrichment of FFD lattice based on Shape gradient . . . 71
5.2 Comparison of history of regular and enriched NACA 0012 optimization using 3, 4 resp. 3 + 1 parameters 73
5.3 Comparison of medium mesh optimization and MG optimization convergence history 76
6.1 FFD lattice for deformation of CFD mesh in NACA 0012 case 78
6.2 Example of practical limitations of FFD for CFD mesh deformations, in the task of flap position optimization 79
6.3 Example of inverted cells in the area of airfoil trailing edge 81
6.4 Example of rotation of RAE 2822 CFD mesh with FFD 83
6.5 Example of wing tip control points elevation and surface deformation for CFD mesh deformation tests 86
6.6 CPU time demands for deformation of Euler mesh 90
6.7 CPU time demands for deformation of RANS mesh 91
6.8 Comparison of Single and parallel deformation time dependency on CFD grid size 92
6.9 Efficiency of use of 4 CPUs with respect to CFD grid size 92

6.10 FFD lattice layout for the CFD mesh deformation 94
6.11 Comparison of pressure coefficient distributions on CRM wing - top view 95
6.12 Comparison of CRM wing shapes initial with the use of Laplace (top) and FFD (bottom) CFD mesh deformation techniques 96
6.13 CRM wing section shapes (axis not in scale) 97
6.14 FFD-RBF lattice layout for the CFD mesh deformation, before RBF coordinate transformation 98
6.15 FFD-RBF lattice layout for the CFD mesh deformation, after RBF coordinate transformation 98
6.16 History of c_D and c_L during the optimization 99
6.17 Comparison of CRM wing: initial (top) optimized with the use of Laplace (middle) and FFD-RBF violated c_L (bottom) CFD mesh deformation techniques 100
6.18 CRM wing section shapes (axis not in scale) 101
A.1 Influence of number of RBF parameterization on fixation error in x dirrection 123
A.2 Influence of number of RBF parameterization on fixation error in y dirrection 124
A.3 Influence of number of RBF parameterization on fixation error in z dirrection 124
B.1 16 degree mesh rotation with Laplace method 125
B.2 16 degree mesh rotation with Spring analogy method 125
B.3 16 degree mesh rotation with FFD method 125
B.4 34 degree mesh rotation with Spring analogy method 126
B.5 34 degree mesh rotation with FFD method 126
B.6 58 degree mesh rotation with FFD method 126
B.7 25 degree mesh rotation with Laplace method 127
B.8 25 degree mesh rotation with FFD method 127
B.9 56 degree mesh rotation with FFD method 127
B.10 Comparison of deformed meshes in the case of 2m elevation of wing tip control points displacement with Laplace (top), Spring analogy (middle) and FFD (bottom) methods in back view of wing tip area . 128
B.11 Comparison of deformed meshes in the case of 12m elevation of wing tip control points displacement with Spring analogy (middle) and FFD (bottom) methods in back view of wing tip area 129
B.12 Deformed mesh in the case of 19m elevation of wing tip control points displacement with FFD method in back view of wing tip area 130

B.13 Comparison of deformed meshes in the case of 0.3m elevation of wing tip control points displacement with Laplace (top) and FFD (bottom) methods in back view of wing tip area 131

B.14 Deformed mesh in the case of 4.2m elevation of wing tip control points displacement with FFD method in back view of wing tip area 132

LIST OF TABLES

3.1 NACA 0012 grid dependency study 39

3.2 Results of NACA0012 optimization for different (FFD^b) lattices. . . . 41

3.3 Optimized NACA0012 grid dependency study (* grid used for optimization 43

3.4 Comparison of $c_{D_{opt}}$ for different initial design variables in NACA0012 optimization. 43

3.5 Influence of NURBS degree on the NACA 0012 minimum drag obtained with 6 FFD parameters 44

3.6 NACA0012 optimization with FFD using weights 45

3.7 NACA0012 optimization with FFD using displacements in x and y directions 46

4.1 Geometric constraint test: averaged errors on the fixation of the trailing edge under FFD^1 deformation. 52

4.2 Comparison of CRM wing optimizations with different FFD parameterizations 56

4.3 Results of passenger aircraft case optimization with 216 design variables on three meshes 60

4.4 Results of passenger aircraft wing optimization with varying number of variables on medium mesh. 61

4.5 Results of passenger aircraft wing optimization with varying number of NURBS degree in stream-wise direction 62

4.6 Results of passenger aircraft wing optimization with varying number of NURBS degree in both stream-wise and span-wise directions . . . 62

4.7 Boundary curve fixation error 66

4.8 EV-55 Outback closed nacelle CF according to area 67

4.9 EV-55 Outback optimization results 69

5.1 Comparison of Enrichment (+1) and NACA0012^a optimization for a hierarchy of FFD lattices with max. NURBS degree^b 72

5.2 Results of CRM wing optimization on M = 0.88 with 242 design variables using NLPQLP using Laplace-Spring on two meshes. 74

5.3 Comparison of medium mesh optimization and multilevel optimization results 75

6.1 Maximal angle of rotation 83

6.2 Maximal angle of rotation test, NACA 0012 Euler mesh 84

6.3 Maximal angle of rotation test, RAE 2822 RANS mesh 84

6.4 CFD results on deformed NACA 0012 airfoil meshes 85

6.5 CFD results on deformed RAE 2822 airfoil meshes 85

6.6 Maximal wing tip section control points elevation 87
6.7 Number of elements that have worse aspect ration and skewness than 0.05 for deformed Euler meshes 88
6.8 Maximal angle of rotation test, RAE 2822 RANS mesh 88
6.9 CFD results on deformed highly swept wing Euler meshes 89
6.10 CFD results on deformed CRM wing RANS meshes 89
6.11 CPU time demands of different mesh deformation techniques 90
6.12 Memory demands of different mesh deformation techniques 93
6.13 Comparison of CRM wing optimization results with different mesh deformation techniques 95
6.14 Comparison of CRM wing optimization results with different mesh deformation techniques 99

LIST OF SYMBOLS, PHYSICAL CONSTANTS AND ABBREVIATIONS

BWB	Blended Wing Body
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CDM	Conjugate Directions Method
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
DES	Detached Eddy Simulation
DFFD	Direct Free-Form Deformation
DNS	Direct Numerical Simulation
DOE	Design Of Experiment
DP	Design Point
FEM	Finite Element Method
FFD	Free-Form Deformation
GOM	Gradient Optimization Method
GPU	Graphics Processing Unit
H-H	Hicks-Henne
LES	Large Eddy Simulation
MG	Multi-Grid
No.	Number
NURBS	Non-Uniform Rational B-Spline
PDE	Partial Differential Equation
RBF	Radial Basic Functions
RSM	Response Surface Method

SDM	Steepest Descent Method
a, b, c	Defines number of FFD lattice points in x, y, z directions
A	Area
Ar	Aspect ratio
C	Continuity
c_D	Drag coefficient
c_L	Lift coefficient
c_m	Pitching moment coefficient
D	Drag force
F	Cost function
G	Matrix of weights of FFD control points
g_r	Gradient
i, j, k	Defines position in knot vectors in x, y, z directions
l	lenght of element edge
L	Lift force
N	NURBS basic function
P	Matrix FFD lattice control points coordinates
p, m, n	Degrees of basic polynomial functions in x, y, z directions
q, r, s	Defines number of NURBS points in x, y, z directions
R	Cartesian coordinate vector of a point [u,v,w] in a parametric space
r_c	Circumscribed triangle radius
S	Search direction
U, V, W	NURBS knot vectors in x, y, z directions
u, v, w	Parametric coordinates x, y, z directions
Vo	Volume

w_i	Weight factor coefficient
y_f	Final object coordinates in y direction
y0	Initial object coordinates in y direction
X	Vector of optimization variables
α_s	Step size
α_{max}	Maximal angle
∇	Differential operator vectors

PUBLICATIONS OF THE AUTHOR

- O. Amoignon, J. Navrátil, and J. Hradil, “Study of parameterizations in the project cedesa,” *AIAA Paper No. 2014-0570*, 2014
- O. Amoignon, J. Hradil, and J. Navrátil, “A numerical study of adaptive ffd in aerodynamic shape optimization,” *AIAA Paper No. 2014-0899*, 2014
- J. Hradil, “Complex geometrical constraints handling in the context of aerodynamic shape optimisation,” *TRANSFER Vyzkum a vyvoj pro letecky prumysl*, vol. 23, 2014
- J. Hradil and J. Navrátil, “Free-form deformation parameterization for aerodynamic shape optimization,” *New Trends in Civil Aviation 2013*, pp. 22–27, 2013
- J. Hradil, “Adaptive free-form deformation parameterization for geometry and mesh morphing in aeronautical applications,” *RESEARCH BULLETIN*, vol. 22, 2012
- P. Dvořák and J. Hradil, “Numerical simulation of cavity valve flow,” *ANSYS 2012 Setkání uživatelu a konference*, pp. 33–39, 2012
- J. Hradil and R. Popela, “Simplified cfd model of synthetic jet actuator (sja),” *Aircraft proceedings*, pp. 48–50, 2010

LIST OF APPENDICES

A Appendix A 123

A.1 Influence of number of RBF centers in x, y and z directions on the fixation error 123

B Appendix B 125

B.1 Deformed 2D CFD Euler meshes of NACA 0012 airfoil 125

B.2 Deformed 2D CFD RANS meshes of RAE 2822 airfoil 127

B.3 Deformed 3D CFD Euler meshes of highly swept wing 128

B.4 Deformed 3D CFD RANS CRM wing meshes 131

A APPENDIX A

A.1 Influence of number of RBF centers in x, y and z directions on the fixation error

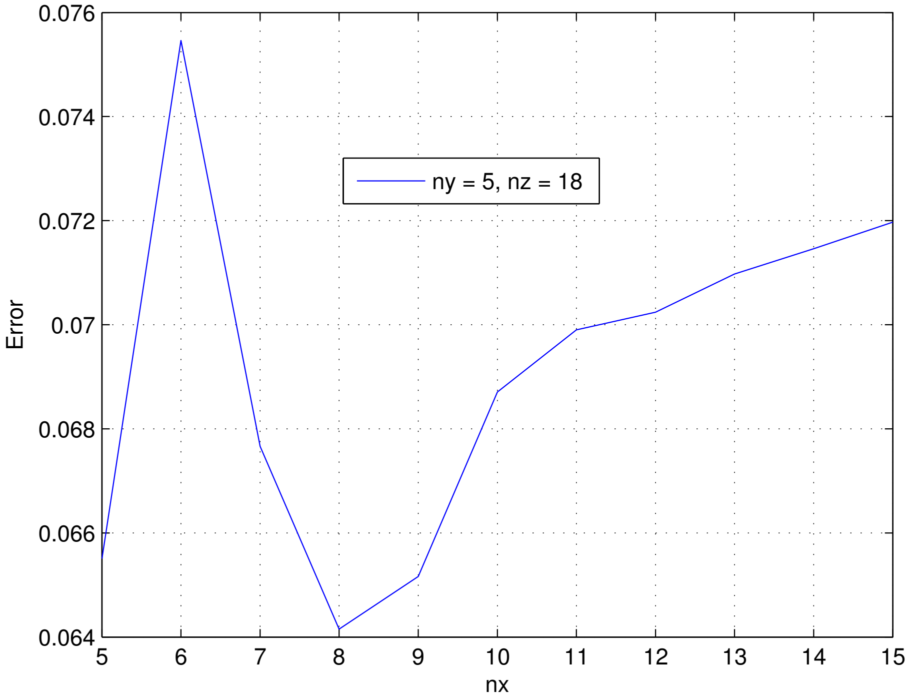


Fig. A.1: Influence of number of RBF parameterization on fixation error in x dirrection

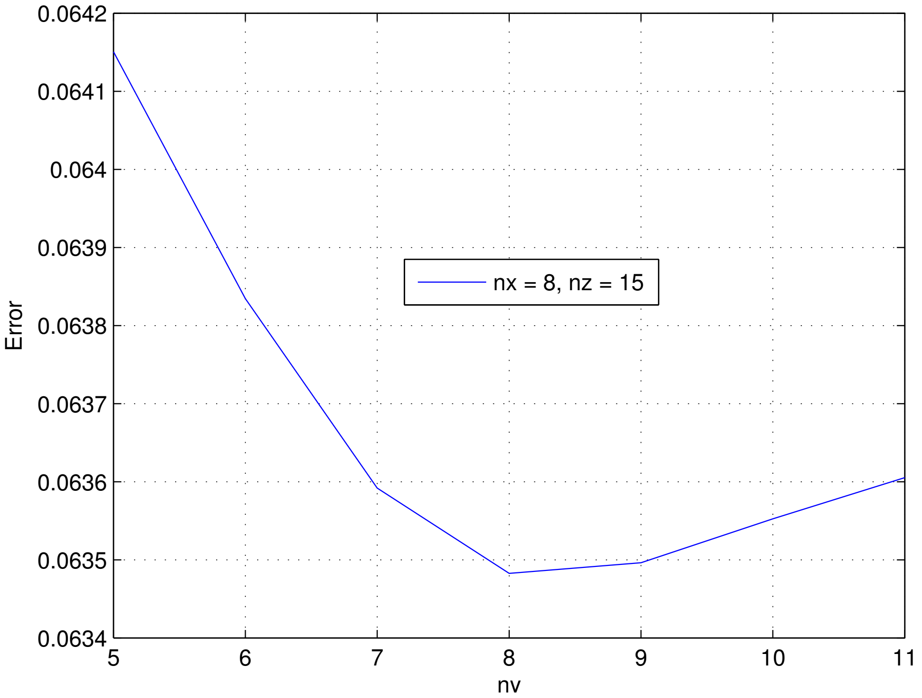


Fig. A.2: Influence of number of RBF parameterization on fixation error in y dirrection

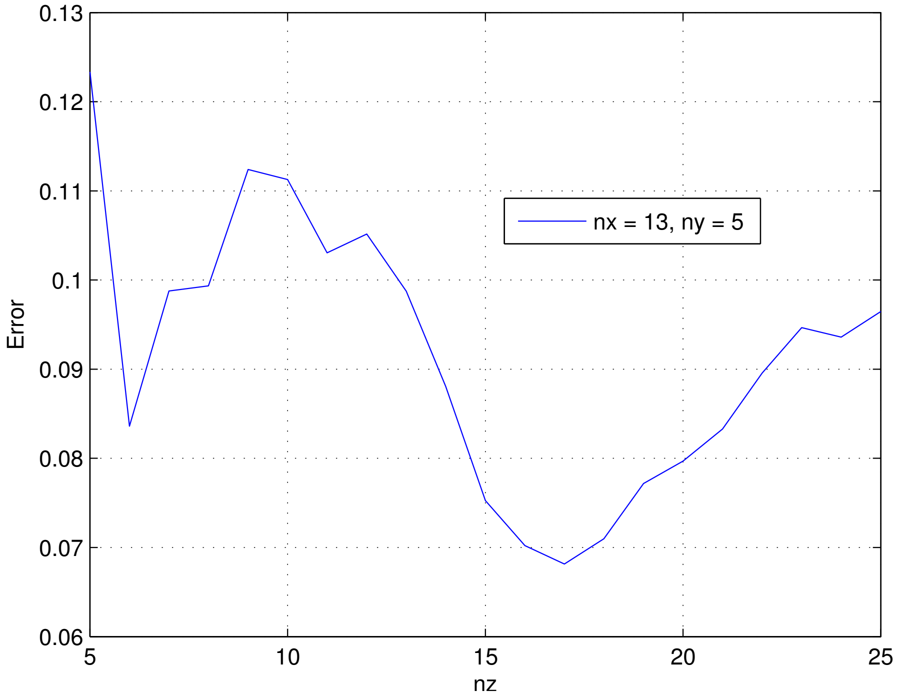


Fig. A.3: Influence of number of RBF parameterization on fixation error in z dirrection

B APPENDIX B

B.1 Deformed 2D CFD Euler meshes of NACA 0012 airfoil

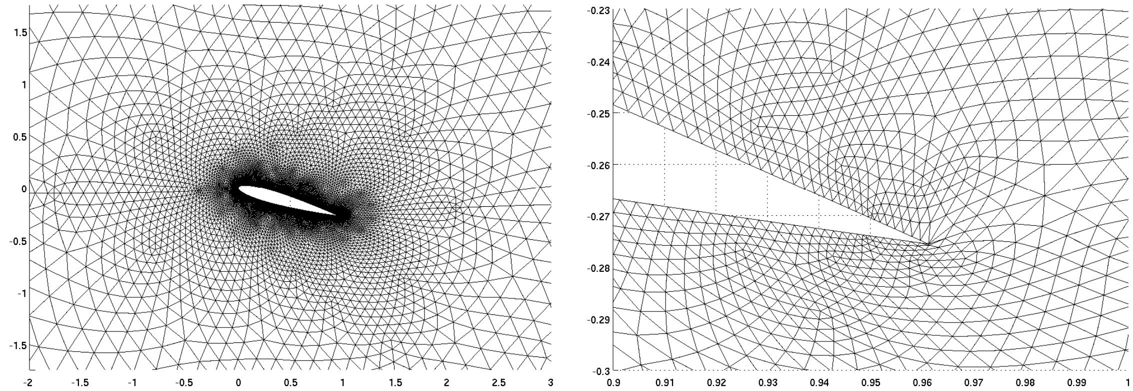


Fig. B.1: 16 degree mesh rotation with Laplace method

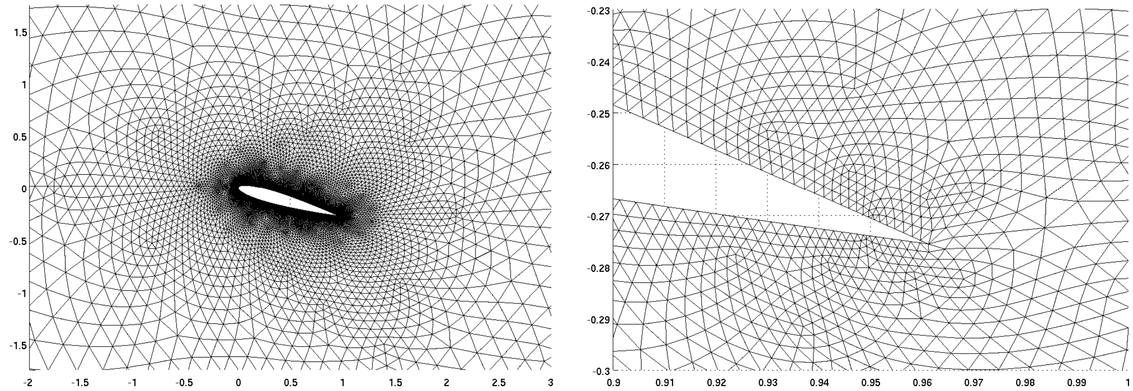


Fig. B.2: 16 degree mesh rotation with Spring analogy method

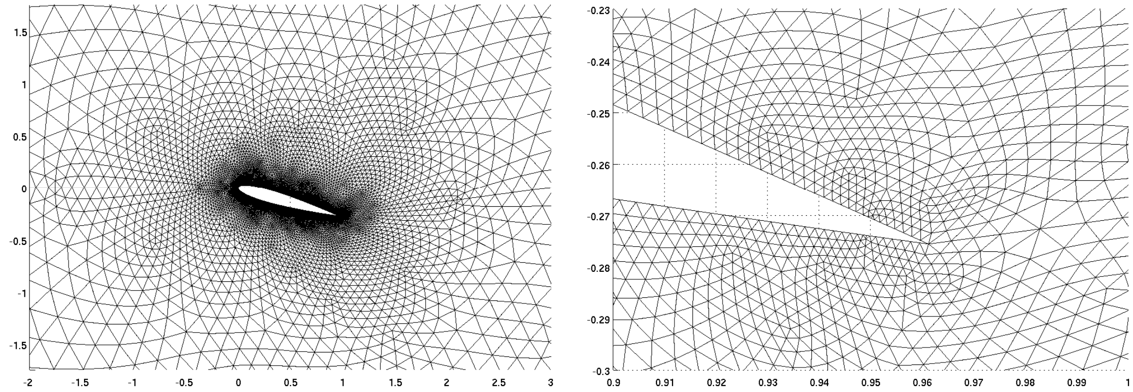


Fig. B.3: 16 degree mesh rotation with FFD method

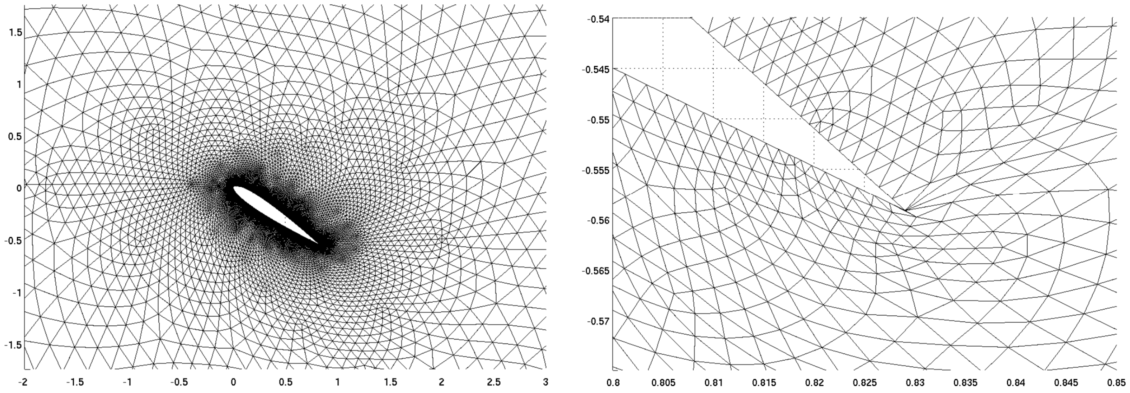


Fig. B.4: 34 degree mesh rotation with Spring analogy method

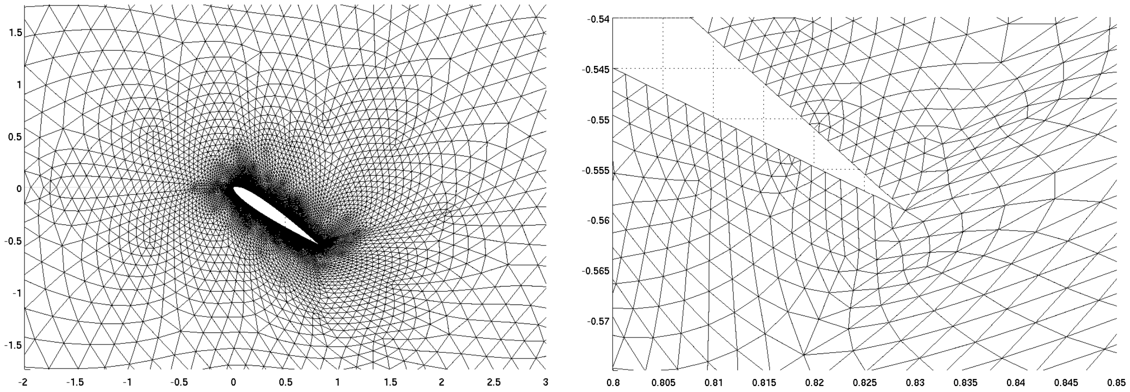


Fig. B.5: 34 degree mesh rotation with FFD method

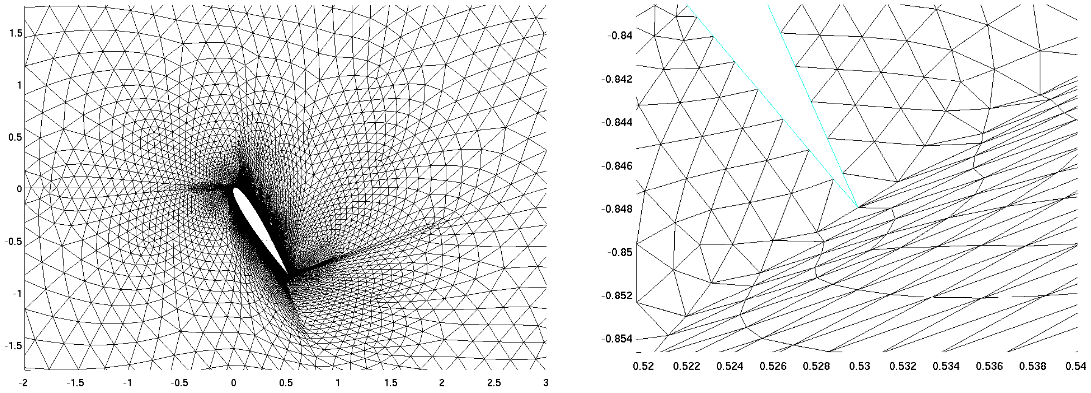


Fig. B.6: 58 degree mesh rotation with FFD method

B.2 Deformed 2D CFD RANS meshes of RAE 2822 airfoil

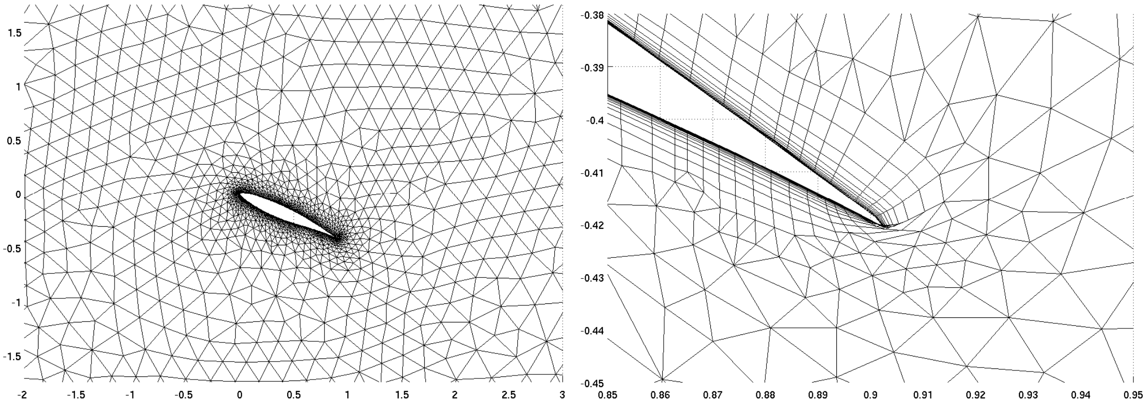


Fig. B.7: 25 degree mesh rotation with Laplace method

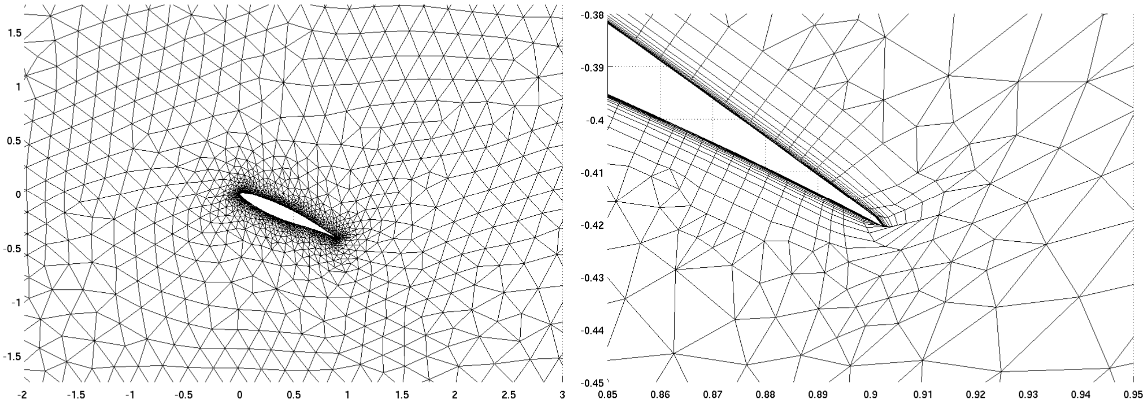


Fig. B.8: 25 degree mesh rotation with FFD method

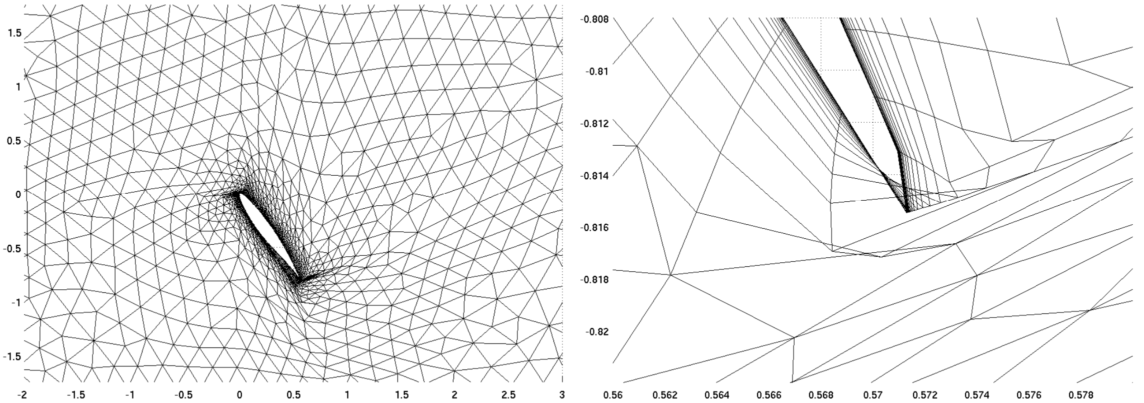


Fig. B.9: 56 degree mesh rotation with FFD method

B.3 Deformed 3D CFD Euler meshes of highly swept wing

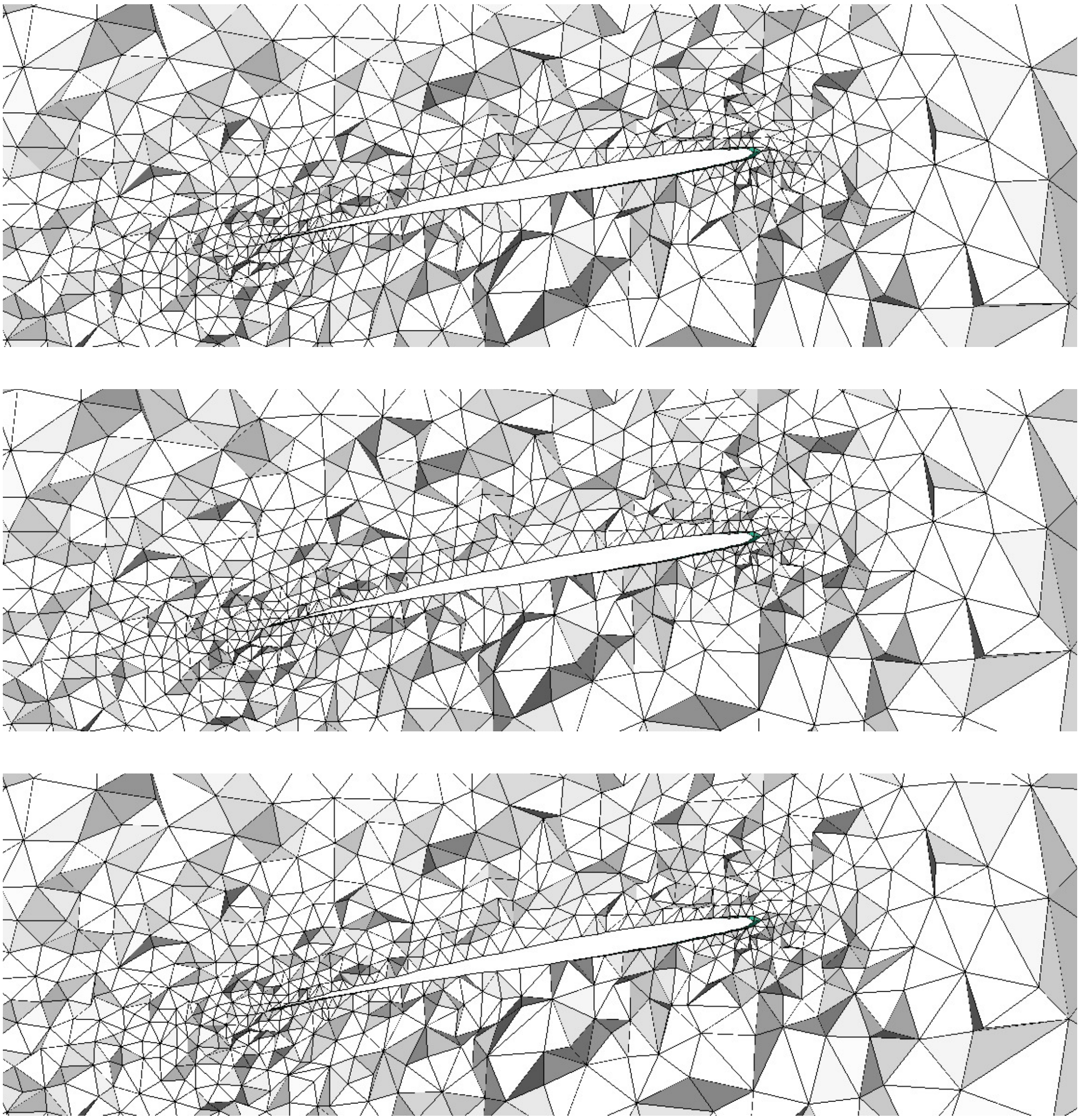


Fig. B.10: Comparison of deformed meshes in the case of 2m elevation of wing tip control points displacement with Laplace (top), Spring analogy (middle) and FFD (bottom) methods in back view of wing tip area

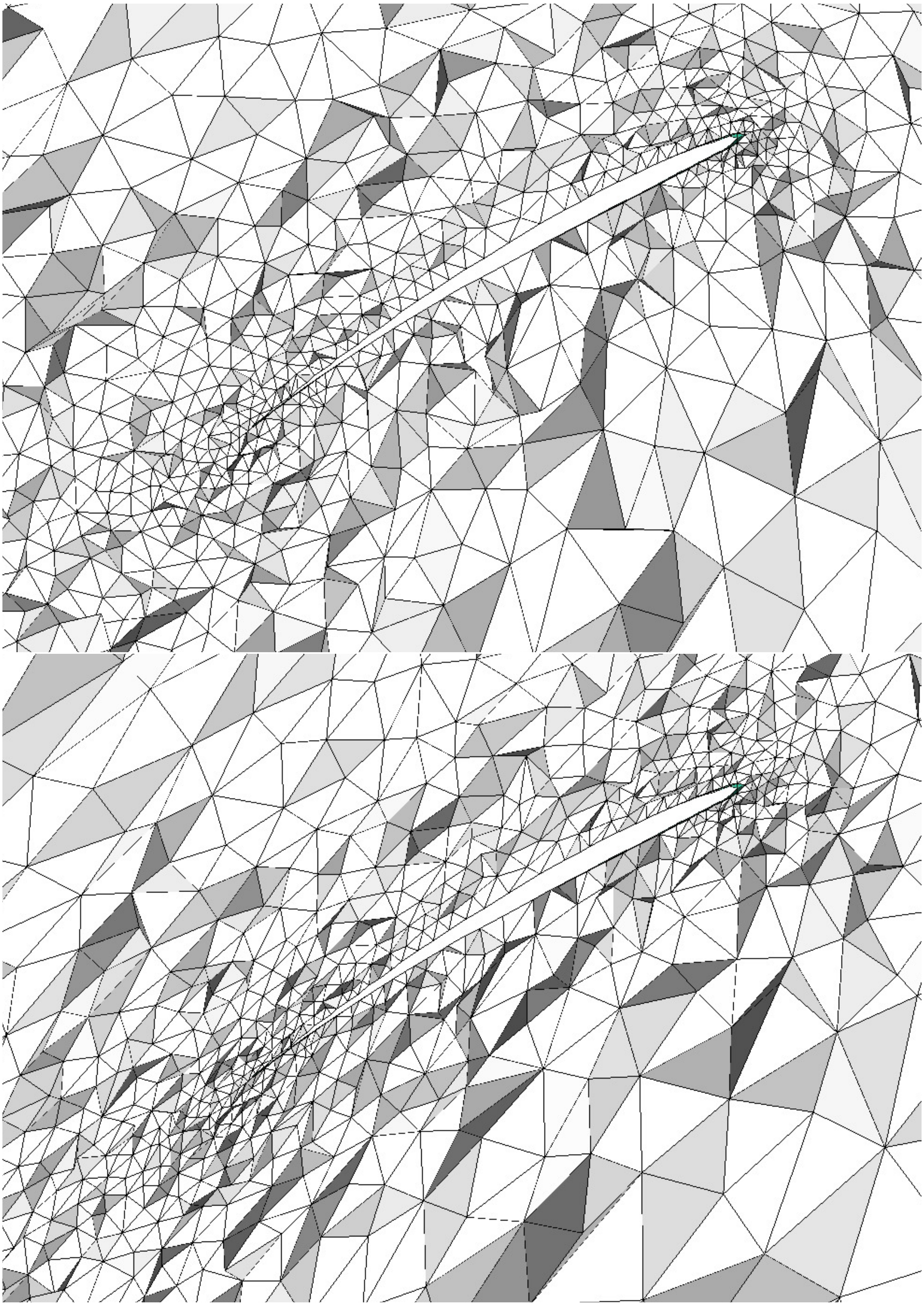


Fig. B.11: Comparison of deformed meshes in the case of 12m elevation of wing tip control points displacement with Spring analogy (middle) and FFD (bottom) methods in back view of wing tip area

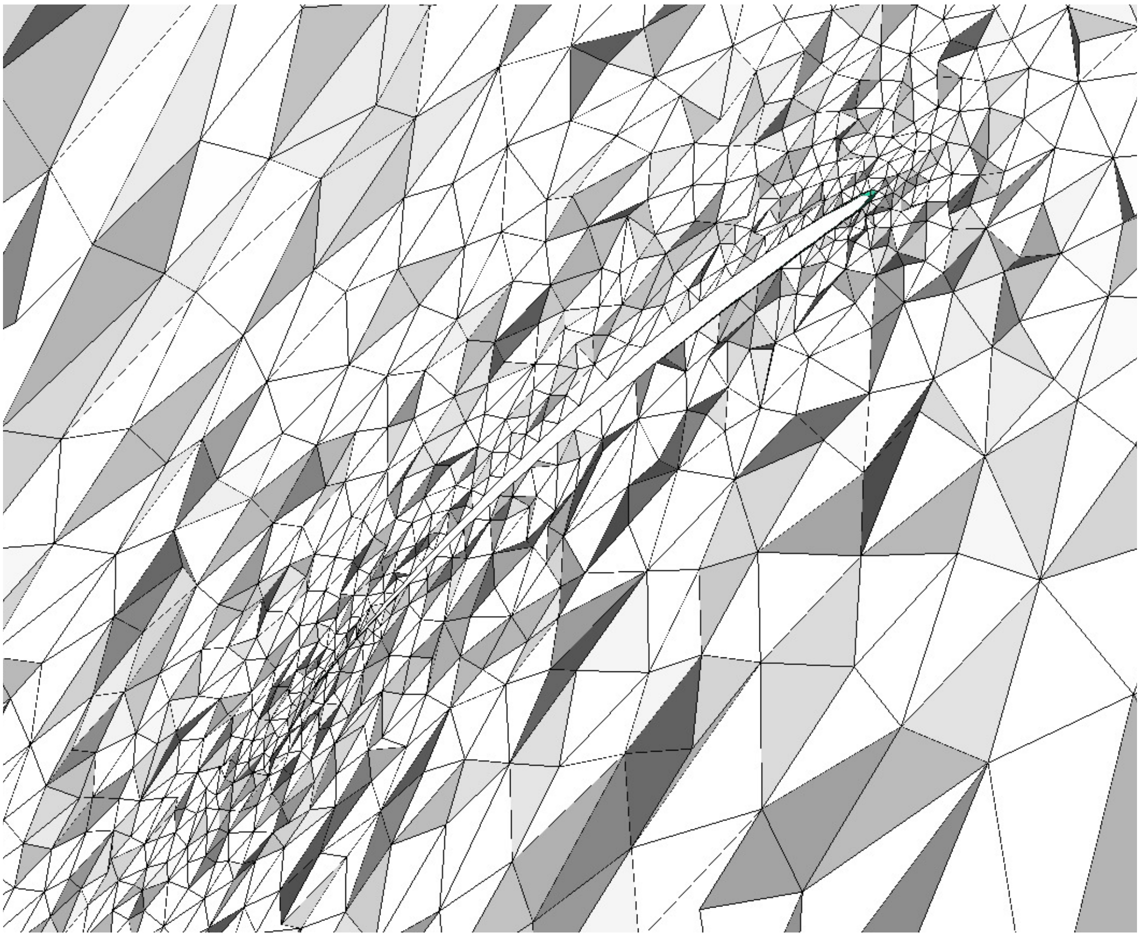


Fig. B.12: Deformed mesh in the case of 19m elevation of wing tip control points displacement with FFD method in back view of wing tip area

B.4 Deformed 3D CFD RANS CRM wing meshes

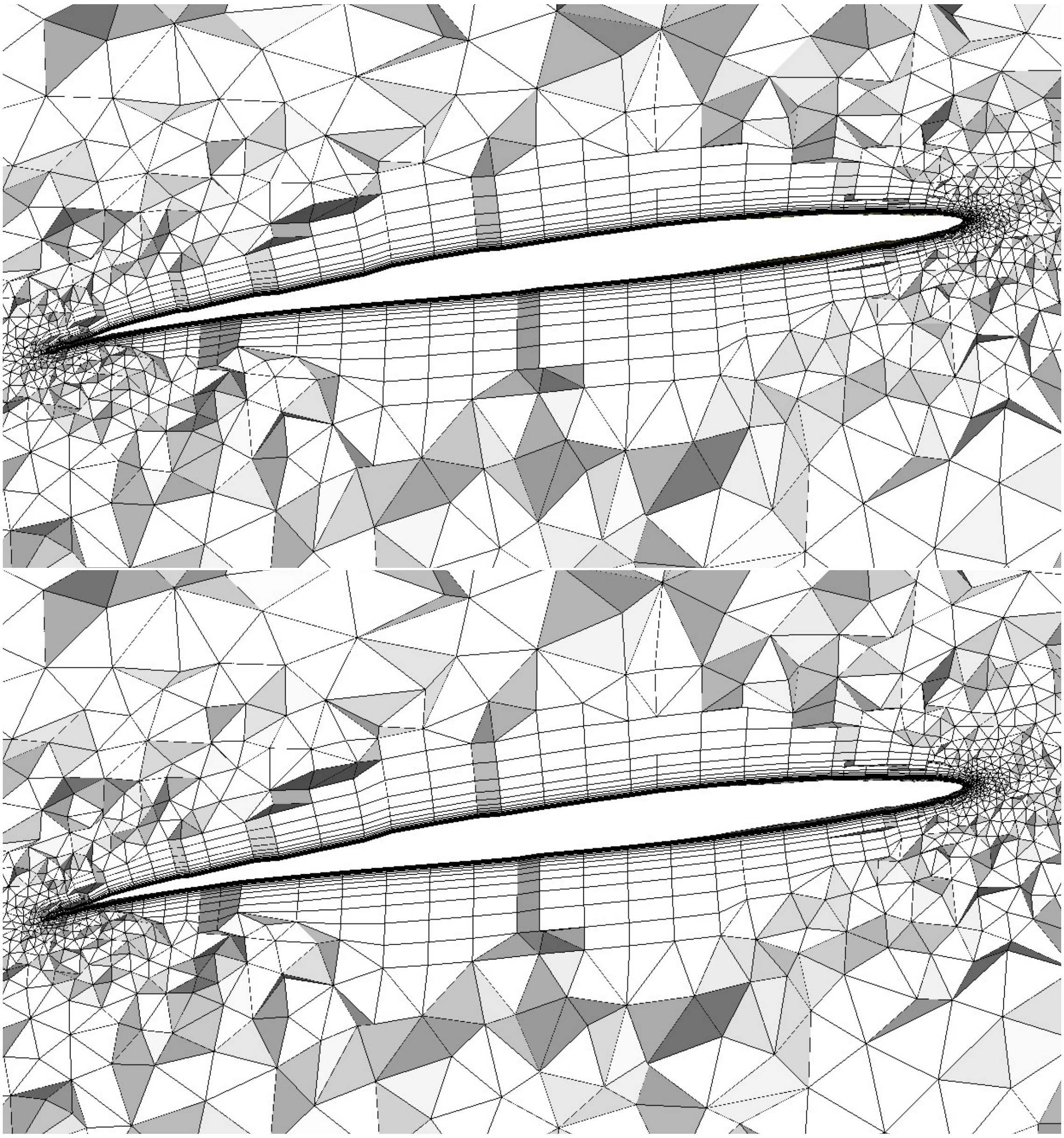


Fig. B.13: Comparison of deformed meshes in the case of 0.3m elevation of wing tip control points displacement with Laplace (top) and FFD (bottom) methods in back view of wing tip area

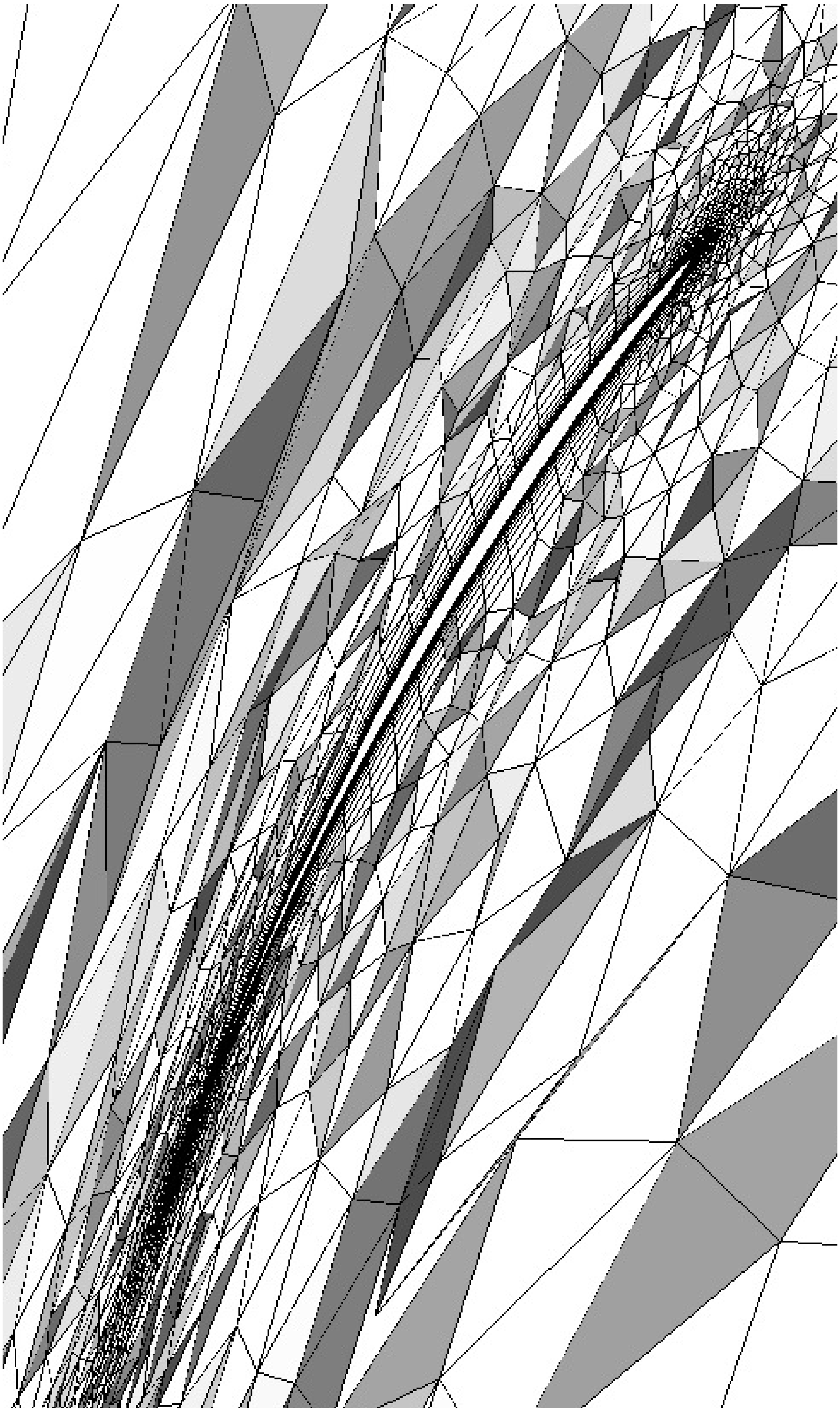


Fig. B.14: Deformed mesh in the case of 4.2m elevation of wing tip control points displacement with FFD method in back view of wing tip area

Curriculum Vitae

Personal information: **Ing. Jiří Hradil**

Date of Birth	4. 10. 1983
Place of Birth	Vyškov
Citizenship	Czech republic
Nationality	Czech
Marital Status	Married
Permanent address:	Komořany 210, 683 01 Rousínov
Contact address:	Brněnská 378/47, 682 01 Vyškov
Cell Phone	+420 733 314 684
E-mail:	hradil.jirka@seznam.cz

Education:

2008 – Present	**Brno University of Technology - Faculty of Mechanical Engineering - Institute of Aerospace Engineering** *Doctoral study programme:* Design and Process Engineering Ph.D. Thesis theme: Adaptive parameterization for aerodynamic shape optimization in aeronautical applications.
2006 – 2008	**Brno University of Technology - Faculty of Mechanical Engineering - Institute of Aerospace Engineering** *Master's study programme:* Aeronautical Engineering *Specialization:* Aircraft Design *Diploma Thesis:* Small Jet Engine Exhaust Temperature and Velocity Flowfield Analysis
2005 – 2006	Brno University of Technology - Faculty of Mechanical Engineering *Master's study programme:* Mechanical Engineering 3rd year
2003 – 2005	Czech Technical University in Prague -Faculty of Mechanical Engineering *Master's study programme:* Mechanical Engineering 1st and 2nd year
1999 – 2003	Secondary Grammar School in Vyškov

Working experience:

2010 – Present — Technical employee at Brno University of Technology - Faculty of Mechanical Engineering - Institute of Aerospace Engineering

- Aerodynamic shape optimization of EV-55 aircraft landing gear nacelle
- Analysis of the distribution of fire extinguishing agent in fire protection system of VUT 061 TURBO aircraft
- Defogging and defrosting system design for VUT 061 TURBO aircraft
- CFD analysis of Synthetic Jet Actuators, Jet Vortex Generators and Vortex Generators for Volkswagen Gmbh (VW XL1).

1.10.2014 – 31.10.2014 Research internship at Swedish Defence Research Agency *(FOI)*
4.8.2013 – 30.10.2013 in the frame of CEDESA project – Stockholm, Sweden.
3.3.2013 – 31.05.2013 Focused on aerodynamic shape optimization and on adaptive
4.5.2011 – 31.10.2011 Free-Form Deformation parameterization

2005 – 2010 — Casual work on Trade Certificate (occasionally abroad) – High load manipulations, assembly, business negotiations, English communication

Conferences, workshops, competitions:

6. September 2014 — Paper presenter at seminar Modelování proudění v leteckých a průmyslových aplikacích, VZLU, Praha

13. - 17. January 2014 — Paper presenter at AIAA SciTech 2014, National Harbor, Maryland, USA

21. November 2014 — Edge workshop 2013, Stockholm, Sweden

21. – 22. June 2013 — Paper presenter at New Trends in Civil Aviation 2013, Žilina, Slovensko

21. – 22. January 2010 — Paper presenter at Aerospace researche center seminar CLKV, Brno, Czech repubic

21. – 23. June 2009 — Participant of Create workshop, Duxford, Great Britain

2008 – 2009 — Participant of Airbus Fly Your Ideas International Competition, qualify into 2nd round (top 84 teams)

Publication:

HRADIL, J. Complex geometrical constraints handling in the context of aerodynamic shape optimisation. Transfer, 2014, vol. 2014, no. 23, p. 23-26. ISSN: 1801- 9315.

AMOIGNON, O.; HRADIL, J.; NAVRÁTIL, J. A numerical study of adaptive FFD in aerodynamic shape optimization. In 52nd Aerospace Sciences Meeting. AIAA 2014-0899, 2014. p. 1-11. ISBN: 978-1-62410-256- 1.

AMOIGNON, O.; NAVRÁTIL, J.; HRADIL, J. Study of parameterizations in project CEDESA. In 52nd Aerospace Sciences Meeting. AIAA 2014-0570, 2014. p.1-19. ISBN: 978-1-62410-256- 1.

HRADIL, J.; NAVRÁTIL, J. Free- Form Deformation parameterization for aerodynamic shape optimization. In New Trends in Civil Aviation 2013. 1. Olomoučany: AKADEMICKÉ NAKLADATELSTVÍ CERM, s. r. o. Brno, 2013. p. 22-27. ISBN: 978-80-7204-843- 4.

HRADIL, J. ADAPTIVE FREE- FORM DEFORMATION PARAMETERIZATION FOR GEOMETRY AND MESH MORPHING IN AERONAUTICAL APPLICATIONS. RESEARCH BULLETIN, 2012, vol. 2012, no. 22, p. 1-7. ISSN: 1425- 2104.

DVOŘÁK, P.; HRADIL, J. Numerical simulation of cavitation valve flow. In ANSYS 2012 Setkání uživatelů a konference. Tesaříkova 1024/ 9 102 00 Praha 10: ARCADEA, 2012. p. 33-39. ISBN: 978-80-905040-1- 1.

HRADIL, J.; POPELA, R. Simplified CFD model of Synthetic Jet Actuator (SJA). Czech Aerospace Proceedings, 2010, vol. 2010, no. 1, p. 48-50. ISSN: 1211- 877X

Professional Qualifications:

Driving Licence	Group B
Special Course Certificate	Certification and Airworthiness

Key Skills:

Operation systems	MS Windows, Linux
Software	Microsoft Office, LaTeX, Glauert III, Xfoil, TORNADO, basic knowledge of FEM programs Ansys Workbench, MSC Patran and MSC Nastran
CAD	AutoCAD, CATIA V5

CFD	Completed a course of CFD during Ph.D. Studies
-Meshing	2D structured, unstructured and hybrid grids in Ansys IcemCFD, TGrid 3D unstructured hybrid grids in Ansys IcemCFD
-Solvers	Fluent V6, Edge Euler, RANS, steady, unsteady, compressible, SJA – periodic BC, Running on Linux cluster, automatization scripts
-Postprocessing	Fluent V6, Tecplot, Paraview, MATLAB-FFA format
-Experience cases	Aircrafts, Cars, Jets, Active flow control, Channel flows
Programming	MATLAB, basic knowledge of C
Parameterization methods	Free-Form Deformation, RBF, Hicks-Henne,
Optimization methods	Steepest descent, Conjugate gradients, Simplex, SQP (Adjoint),
Languages	English – Advanced
	German – Intermediate
Profession interests	Parameterization, Optimization, CFD, Aerodynamics, Airplanes, Cars
Personal interests	Sport, Cosmonautics, Literature, Computers, Travelling, Airplanes, Car races