IsoGeometric analysis and shape optimisation of aircraft compressor blades

Marie Guerder

N° d'ordre NNT : 2022LYSEI069

THÈSE de DOCTORAT de l'UNIVERISTÉ DE LYON

opérée au sein de

l'Institut National des Sciences Appliquées de Lyon

École Doctorale MEGA (ED 162)
Mécanique, Énergétique, Génie Civil, Acoustique

Spécialité de Doctorat
Mécanique

Soutenue publiquement le 20/07/2022 par

Marie Guerder
Ingénieure

IsoGeometric Analysis and Shape Optimisation of Aircraft Compressor Blades

Devant le jury composé de

NOM, Prénom	Grade/Qualité	Établissement	Rôle
BOUCLIER, Robin	Maître de Conférences HDR	INSA Toulouse	Rapporteur
RASSINEUX, Alain	Professeur	UTC	Rapporteur
BOUCARD, Pierre-Alain	Professeur	ENS Paris-Saclay	Président
CHASAPI, Margarita	Docteure	EPFL	Examinatrice
ELGUEDJ, Thomas	Professeur	INSA Lyon	Directeur de thèse
TOUZEAU, Josselyn	Docteur-Ingénieur	Safran Aircraft Engines	Encadrant
DAM, Damien	Ingénieur	Safran Aircraft Engines	Invité et encadrant
DUVAL, Arnaud	Docteur-Ingénieur	CNRS	Invité et encadrant
RANNOU, Johann	Docteur	ONERA	Invité

Hâtez-vous lentement, et, sans perdre courage,
Vingt fois sur le métier remettez votre ouvrage
Polissez-le sans cesse et le repolissez ;
Ajoutez quelquefois, et souvent effacez.

—Nicolas Boileau, *L'Art poétique*

Acknowledgements

THROUGHOUT the years that have been necessary to carry out this work, I have met and have been supported by a tremendous number of persons. It is needless to say that they all have contributed, in their own way, to the accomplishment of this manuscript.

I would like to acknowledge the company Safran Aircraft Engines for having financially supported this work. More particularly, I would like to thank Stéphane Burgburu, head of the Methods department, for taking me on board, and Pierre-Antoine Bénard for including me to his team. Josselyn Touzeau supervised this work the whole time for the industrial side. Thank you for the independence you let me. Paul Féliot and Damien Dam were also involved in the supervising team. I am grateful for your advice.

I thereafter thank Daniel Nélias, head of the LaMCoS, for including me in the laboratory. For the academic side, Thomas Elguedj was my PhD supervisor. I would like to express all my gratitude for your unfailing support — be it scientific or moral —, your availability in spite of being overscheduled, and your ability to always lift my spirits when needed. Arnaud Duval supervised this thesis as well, whom I acknowledge for the valuable help and the time dedicated to further this work.

The home straight constituted by the preparation of the defence is an intense period, not only for the PhD student, but also for the reviewers of his or her work. Robin Bouclier and Alain Rassineux have accepted to play this part for this work, and I am very grateful for their commitment. Thank you both for your insight and valuable advice. Pierre-Alain Boucard kindly accepted to preside over the jury, which involved Margarita Chasapi and Johann Rannou as well. Many thanks to you all for accepting to be part of this jury, and for the interesting and fruitful discussions we had.

It is time to turn to the members of the laboratory. I am very grateful towards all my colleagues, permanent members of staff or PhD students. I have met a lot of wonderful persons, and spent truly great time with a number of them throughout these last years. Thank you all!

Carrying out a research work is a long journey, that is made easier when surrounded by dear ones. I hereby express my gratitude to all my friends who have been, directly or indirectly, involved in this adventure. Even out of sight, you supported me during this one-of-a-kind experience. Special thanks to those who attended my defence, in person or remotely. Last but not least, I would like to thank my family for being there for me at all times.

Résumé

Relever le défi que représente l'incorporation de la modélisation géométrique et l'analyse numérique est un des objectifs de l'Analyse IsoGéométrique. Un des aspects principaux permettant d'adresser cette question est l'utilisation des mêmes fonctions mathématiques – à savoir, les B-Splines Rationnelles Non-Uniformes – pour décrire la géométrie et pour servir de support afin de résoudre l'analyse. L'un des avantages des fonctions NURBS est qu'elles bénéficient d'une continuité supérieure en comparaison aux polynômes de Lagrange, et des maillages plus grossiers peuvent être utilisés, réduisant le temps nécessaire pour l'analyse. Concernant l'optimisation de forme, l'IGA présente l'avantage de fournir un modèle qui est compatible avec les logiciels de Conception Assistée par Ordinateur, sans traitement supplémentaire.

La conception et la fabrication de moteurs d'avions repose largement sur les méthodes numériques, et peut donc bénéficier des propriétés avantageuses de l'IGA. Des problématiques spécifiques émergent dans ce contexte industriel, parmi lesquelles la définition volumique des parties tournantes telles que les aubages. L'objet de ces travaux est de proposer un cadre complet pour la définition, l'analyse et l'optimisation de forme d'aubages de moteurs d'avions, en utilisant l'IGA.

En s'appuyant sur une géométrie d'aubage industriel, nous proposons une procédure permettant d'en construire un modèle B-Spline volumique compatible pour l'analyse, en assurant la précision géométrique de ce dernier ainsi que la régularité du paramétrage. L'optimisation de forme est réalisée en utilisant les coordonnées spatiales des points de contrôle comme variables de *design*. La réponse mécanique de la structure est calculée à l'aide du code IGA *open-source* Yeti. Le reste de l'assemblage, incluant la plateforme et le pied de l'aubage, est pris en compte grâce à une approche de type *mortar* pour le couplage faible des patchs. Une formulation d'élément solide immergé a été développée durant cette étude, permettant de modéliser de façon précise le congé raccordant la pale à sa plateforme. De plus, la compatibilité géométrique est garantie aux interfaces entre patchs adjacents lors de la mise à jour de la forme de la pale durant la procédure d'optimisation de forme.

Les résultats obtenus démontrent l'efficacité de la méthode et sa pertinence pour la conception et l'optimisation de forme d'aubages de moteurs d'avions.

Abstract

Uniting the workflows of geometric design and numerical analysis is one of the challenging aims of IsoGeometric Analysis. Such a goal is addressed by using the same mathematical functions — namely, Non-Uniform Rational B-Splines — to describe the geometry and to serve as a support to solve the analysis. Amongst other advantages, NURBS functions benefit from a higher continuity with regards to Lagrange polynomials, and coarser meshes can be used, reducing the analysis time. When it comes to shape optimisation, IGA offers the advantage of providing a model that is compatible with Computer Aided Design software, without further processing.

The aircraft engine design and manufacturing industry widely uses numerical methods, and hence can benefit from the advantageous features of IGA. Specific concerns arise in this industrial context, the volumetric definition of spinning parts such as blades being a prominent one. The purpose of this work is to propose a complete framework for the design, analysis and shape optimisation of aircraft engine blades using IGA.

Using an actual industrial blade geometry, we propose a procedure to reconstruct a B-spline analysis-suitable volumetric model of the blade, ensuring its geometric accuracy and parametrisation regularity. Shape optimisation is performed using the spatial coordinates of control points as design variables. The mechanical response of the structure is computed using the open-source IGA code Yeti. The rest of the assembly, including the platform and tenon of the blade, is considered using a mortar approach for weak patch coupling. An embedded solid element formulation was developed during this study, enabling accurate modelling of the fillet linking the blade to its platform. In addition, it guarantees the geometric compatibility of the interfaces between adjacent patches during shape updates in the course of the shape optimisation process.

The results demonstrate the efficiency of the method and its relevance for industrial aircraft engine blade design and shape optimisation.

Contents

List of Figures

Chapter 1

Introduction

Numerical methods constitute the fundamentals of a significant part of today's engineering design process, and hence they arouse considerable research interest. In particular, computational mechanics focuses on the use of numerical methods to study the mechanical behaviour of objects. In this work, we place ourselves in the field of computational structural mechanics using IsoGeometric Analysis. We apply this numerical method to a subject of great interest in the aircraft engine industry: the design of aircraft engine blades. Ultimately, this work aims at demonstrating the interest of using a different numerical method than the ones currently used for the design, analysis and shape optimisation of aircraft engine blades.

In the remainder of this chapter, we shall give an overview of the contextual aspects of this research. The research problem will then be addressed, followed by the research aims and objectives. Some significance elements are further given to expose the motivations more broadly, as well as some limitations. Lastly, the structure of the rest of the document is outlined.

IsoGeometric Analysis has been a subject of research for nearly two decades now, the seminal paper of Hughes and co-workers dating back to 2005 (Hughes et al., 2005). The name of this method reflects its philosophy of seamlessly integrating the geometric design and numerical analysis procedures. These two procedures are treated separately in the classic engineering design process: the geometry is usually defined by means of Computer-Aided Design software, and numerical analysis is often performed with the Finite Element Methods. Uniting these two fields is the challenging aim of IGA.

Practically, this goal is addressed by using the same mathematical functions to describe the geometry and to serve as a support to solve the analysis. Consequently, Non-Uniform Rational B-Splines are used as basis functions for FEM, allowing the analysis to benefit from the interesting properties of such functions. Amongst others, we may cite the higher continuity of NURBS functions with regards to Lagrange polynomials, which yields new refinement strategies. Hence, a higher precision can be reached with the same number of degrees of freedom using NURBS.

The geometric description originating from CAD software consists of boundary or

surface representations. Approaches relying solely on surface geometries have thus naturally been explored, as is the case for shell approaches. Hence, there exists substantial on concerning shell formulation in the context of IGA (Kiendl et al., 2009, Benson et al., 2013, Echter et al., 2013, Bouclier et al., 2015, Hirschler, 2019).

The geometric description used for the analysis is made exact by the use of NURBS functions, which is a core element of IGA. The unicity of the "design-and-analysis" model offers new perspectives concerning shape modifications of an initial geometric description, as needed for shape optimisation processes. This field has broadly been explored since the beginning of IGA, as shown by the numerous publications on the subject (Wall et al., 2008, Cho and Ha, 2009, Qian, 2010, Manh et al., 2011, Nørtoft and Gravesen, 2013, Kiendl et al., 2014, Fußeder et al., 2015, Herrema et al., 2017, Wang et al., 2018, Hirschler et al., 2019b,c).

Apart from possessing the previously cited properties, NURBS still constitute parametric descriptions. Amongst other applications, parametric descriptions are used to describe complex shapes related to physical effects. For instance, it is the case of aerodynamic profiles, for which the lift force is driven by the camber of the profile. Hence, NURBS are used for applications such as aerodynamic shape design and optimisation for turbine blades (Miller et al., 1996, Martin et al., 2014, Mykhaskiv et al., 2018), hydrofoils (Kostas et al., 2017), or marine propellers (Pérez-Arribas and Pérez-Fernández, 2018, Arapakopoulos et al., 2019).

Linking the geometric modelling and numerical analysis tools is a crucial goal when it comes to the engineering design process, and IGA offers a way of addressing this challenge. Even though studying shells is highly relevant in the context of IGA, volumetric modelling is still needed for spinning bodies such as blades. The most critical effort acting on such bodies are volumetric forces, which cannot be recovered by classical shell formulations.

Großmann and Jüttler (2010) have addressed the issue of volumetric blade modelling and analysis: starting from point-cloud data describing the blade geometry, they generate a single-patch blade model. Still, shape modification is a crucial aspect for shape optimisation procedures: the airfoil part must be modelled as a distinct patch from the rest of the blade (namely, the fillet and platform). Hence, specific methodologies have to be developed in order to generate volumetric blade models that are suitable for structural analysis and shape optimisation. Generating multiple volumetric patches raises questions of its own, especially in terms of coupling approaches. Using actual industrial geometries leads to highly non-conforming discretisation, increasing the need for genericity in the numerical tools.

Given the described background and research problem, our aim in this work is to propose a complete framework for the design, analysis and shape optimisation of aircraft engine blades using IGA.

Achieving this goal requires various objectives to be met, starting with the generation of a volumetric model of an aircraft engine blade that is suitable for analysis and for shape optimisation, while guaranteeing an initial geometric description that is similar to the input one. Running analysis and shape optimisation requires a suitable discretisation of the generated volumetric model, which must be considered. Apart from the airfoil, a fillet and platform have to be modelled. Hence another objective lies in the coupling of several patches presenting non-conformal parametrisations at their interfaces. The issue of modifying the shape of the airfoil part — through shape optimisation procedures —, while guaranteeing geometric compatibility with the surrounding patches, is to be addressed as well. All these objectives have to be fulfilled while being compatible with current software and numerical tools used in the industrial partner Safran Aircraft Engines. Lastly, in order to clarify the scope of this study, we emphasise that we do not develop specific optimisation algorithms, but rather use open and widely-available implementations as an optimisation black-box.

This study will bring the benefits of IGA — *e.g.*, a reduced overall design time and a higher accuracy — to the analysis and shape optimisation of aircraft engines blades. For instance, the proposed methodology for the volumetric parametrisation of blades can be used as a pre-analysis tool, which could decrease the time needed for the first design stages. Higher accuracy brings new perspectives for the analysis of aircraft engine blades. These favourable features could eventually enhance the performance and reduce the consumption of future aircraft engines. In addition, as the result of shape optimisation using IGA is compatible with CAD software without further modification, time is also saved at this stage of the design process. Ultimately, this study contributes to narrow the gap between academic and industrial methodologies.

Moreover, we seek to provide a convenient framework for the design, analysis, and shape optimisation of complex multi-patch structures. Hence, several aspects of the methodological details herein are applicable to any type of application featuring volumetric NURBS-based IGA inquiries.

This study involves contributions to the open-source IGA code YETI developed at the Contact and Structural Mechanics laboratory at INSA-Lyon, including the specific elements developed in this work.

Our research is targeted at aircraft engine blades, given the work-flow and processes in place at Safran Aircraft Engines. Hence, the methodology for the creation of a volumetric blade model is based on the needs of the industrial partner, and might not fit general purposes.

Using the in-house code YETI presents great advantages, especially regarding flexibility, yet it has some drawbacks as well. On the one hand, its non-intrusive aspect does not allow full integration in the modelling process of the industrial partner, which would have been interesting for comparison purposes. On the other hand, the implementation does not offer as many numerical tools as commercial solutions do (yet!),

which can be limiting as well. However, our goal is to prove that such a framework is feasible, and gives satisfactory results compared with current methodologies. The goal is not to deliver a tool ready for industrial use.

This manuscript is organised as follows. Chapter 2 introduces context and theoretical notions concerning the research subject. The industrial context is first detailed, then numerical optimisation is introduced, and finally the particularities of IGA are specified. Chapter 3 addresses the question of generating a volumetric blade model, given the specific type of initial geometric description provided by the industrial partner. The model originating from the proposed framework is subsequently used for structural analyses and structural shape optimisation. Chapter 4 focuses on more general issues, namely the analysis of complex models consisting of several patches. Coupling of several solid-elements patches is addressed, as well as a nested parametrisation approach used for geometric modelling and shape modifications. Chapter 5 combines the previous questions — and answers — to perform the analysis and structural shape optimisation of a complete (complex) blade model. Finally, the concluding chapter proposes a summary of the work that has been achieved, as well as some perspectives for further research.

Chapter 2

Isogeometric analysis as a powerful tool for the design of aircraft engine blades

Proposing new approaches for both industrial matters and scientific research is a challenging goal. Hence, several notions have to be introduced in order to set the scene for this complex dance. We first give some elements concerning aircraft engines and blades design, as it constitutes the starting point of our work. We subsequently turn to the topic of structural optimisation, giving some theoretical and operational background. Then, IsoGeometric Analysis is introduced. Finally, we set out the convenient framework brought by IGA for gradient-based shape optimisation, in the case of aircraft engines blades design.

Contents

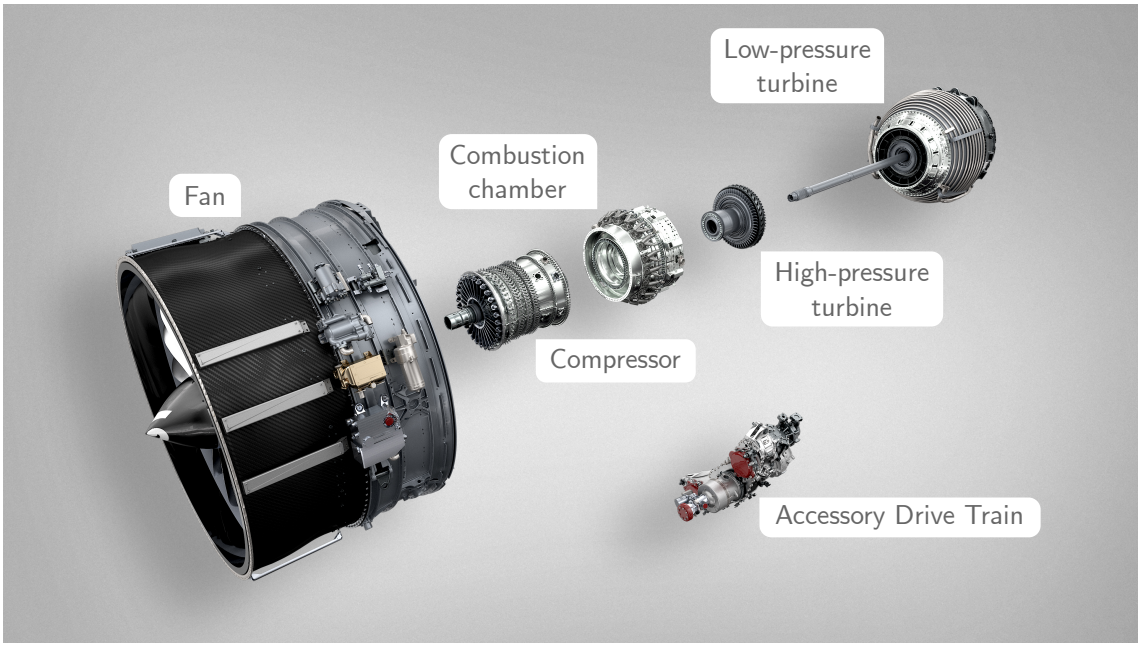


Figure 2.1 – Exploded view of a LEAP 1-A engine (SAFRAN, 2020).

2.1 Aircraft engines blades design

In the following section we introduce several essential elements regarding the industrial context we place ourselves into. These aspects constitute, to a considerable extent, a starting point for the scientific choices this work relies on. Hence, we first start by an introduction about aircraft engines functioning. More details are further provided concerning the axial compressor blades themselves, specifically for design and sizing aspects. Finally, some concerns arising in the search of possible improvements of the design process are suggested.

2.1.1 A compromise between aerodynamic performance and mechanical resistance

2.1.1.1 Aircraft engines functioning

An introduction to aircraft engines functioning is given hereafter. Far from being exhaustive, we simply give the broad lines, identifying the main parts composing aircraft engines. In order to further describe the functioning of the engine, we take the example of the LEAP engine. In particular, figure 2.1 pictures an exploded view of a LEAP 1-A engine. We refer to the document of Thevenin (2004) for a concise reference on the subject.

During flight, the engine pushes the aircraft forward. The air is sucked up by the fan blades, then is expelled at a high velocity at the rear of the engine, thus creating the

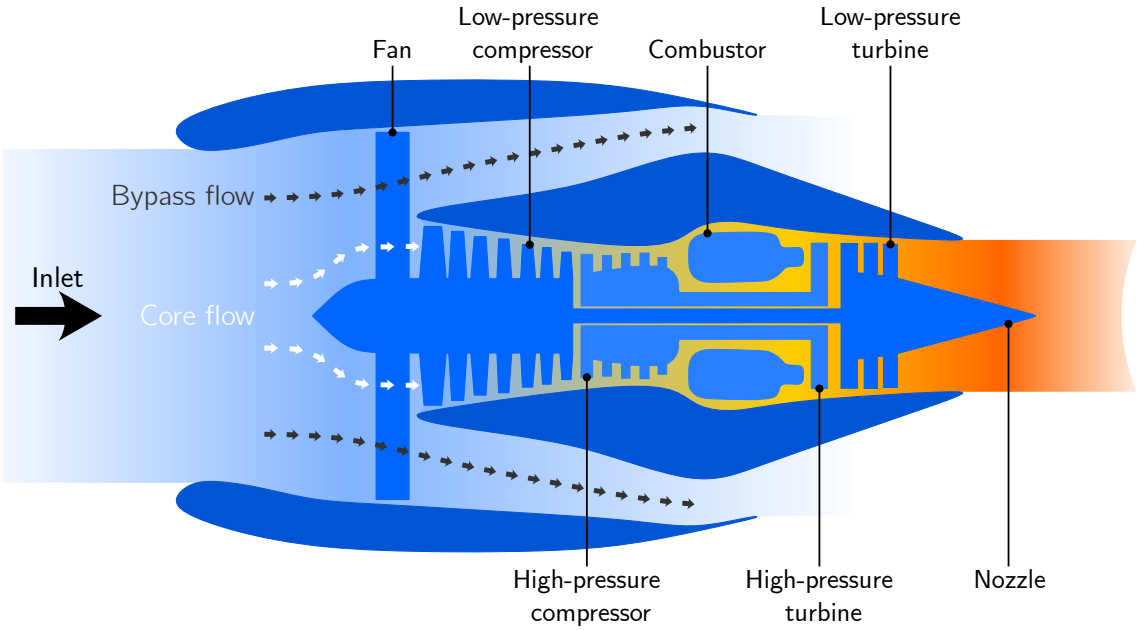


Figure 2.2 – Schematic cross-section view of a turbofan engine. The two types of flow are represented, as well as the main parts of the engine. (Adapted from Garg, 2013).

necessary thrust. It relies on Newton's third law: for each action, there exists an equal and opposed reaction. The fan acts as a propeller: by spinning, the fan blades push the air flow in the engine. The LEAP engine is a *high-bypass turbofan* engine, meaning that a high volume of air bypasses the core of the engine to be ejected at the rear. Hence, the inlet flow is separated into two types: the *primary* (or *core*) flow, and the *secondary* (or *bypass)* flow. Figure 2.2 pictures a cross-section view of a turbofan engine.

The primary flow goes through the machine, consisting of the compressors, the combustion chamber and the turbines. There are two compressors in a row, first the low-pressure compressor, followed by the high-pressure compressor. The compressed air is further guided forward to the combustion chamber. The compressing stages enable to reach optimal temperature and pressure conditions. Before reaching the combustion chamber, the injectors blend the kerosene together with the air, resulting in a homogeneous blend. Having a homogeneous blend reduces the peak of temperature during combustion. The combustion chamber is the core of the engine, where the energy is created by the combustion of the homogeneous blend of air and kerosene. Coming out of the combustion chamber, the flow transfers its energy to the turbines. The pressure and velocity of the high-temperature gases provide the necessary energy to drive the turbines and the shaft, which in turn drives the compressors and the fan.

In summary, the primary flow produces the energy to drive the engine, while the secondary flow is responsible for a vast majority of the engine thrust to push the aircraft forward. Additionally, we mention that compressors and turbines are composed of disks, consisting in an alternation of fixed parts (*stators*) and spinning parts (*rotors*).

2.1.1.2 Axial compressor blades geometric description

Design practices When designing aircraft engines axial compressor blades, a common practice in industry is to tackle aerodynamics and structural mechanics issues separately. Most of the time, these two aspects are actually addressed by distinct specialised teams. Considering that the best possible design from each physics point of view are close to opposite, it results in a conflicting problem to solve. Balance has to be found between aerodynamic performance and mechanical resistance, while satisfying numerous industrial constraints. Meeting this goal is never an elementary task, and considerable effort is dedicated to the convergence of skills and interests. It usually takes the form of an iterative process, where both parties sequentially work on the design. The initial geometry is defined by taking aerodynamic concerns into account, then the "design loops" can take place.

A blade geometry is usually defined by modelling solely its outer surface. It is defined by a series of cross-sections (or *profiles*), that are stacked along the blade height. The number and geometries of the profiles, as well as the way they are stacked, are as many design choices that aerodynamics and structural mechanics teams have to agree upon.

Defining the initial geometry The behaviour of the air flow passing through an axial compressor can be represented by *stream lines* (or *flow lines*, see figure 2.3). They are typically obtained using Euler-2D aerodynamic analyses. A revolution of these flow lines about the rotation axis of the engine is usually performed to define three-dimensional *stream surfaces* (see figure 2.4). In broad outline, the shape of the blades is defined so as to obtain the desired behaviour of the flow in precise locations. Hence, the blades guide the flow path throughout the engine.

The profiles are defined in a planar fashion. The aerodynamic performance of a blade (or any propeller type) is governed by geometric parameters mostly, that are defined in a two-dimensional space, as pictured on figure 2.5. A profile is hence defined for each stream line, in a two-dimensional fashion. A conformal operation of the 2D sections onto the stream surfaces is applied to yield blade sections defined in 3D space. The complexity of shapes is variable, according to the position of the blade in the engine, so is the number of sections used to define the geometry. The three-dimensional blade sections pictured in figure 2.6 for instance feature a high number of profiles and represent a fairly complex geometry.

Finally, once the stacked three-dimensional cross sections have been defined, a *stacking axis* is defined, as pictured in figure 2.3. This axis serves as a guiding curve for the lofting operation yielding the final outer surface of the blade. For more details on surface lofting the interested reader is referred to Piegl and Tiller (1996).

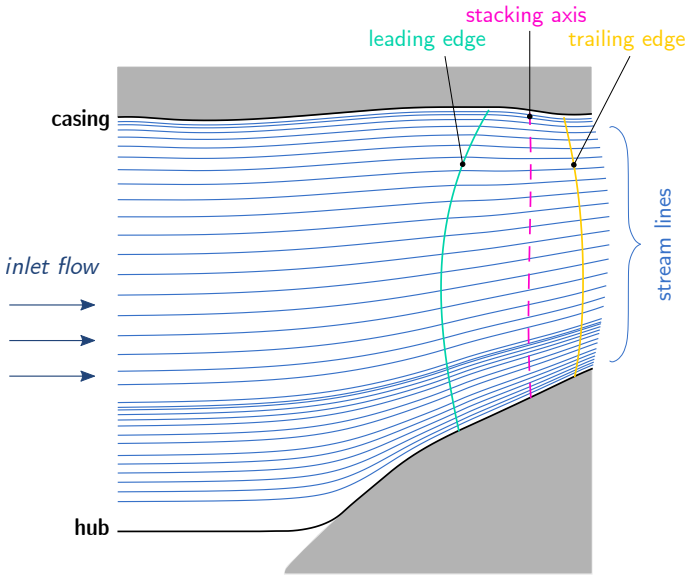


Figure 2.3 – Two-dimensional stream lines in a low-pressure axial compressor. The leading and trailing edges lines of the fan blade are represented as well.

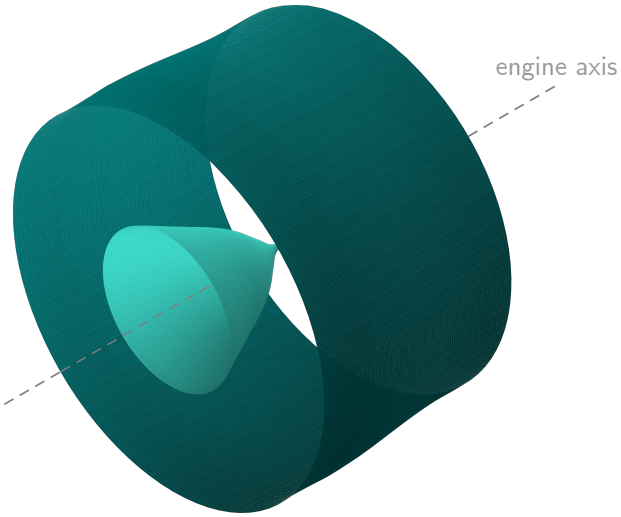


Figure 2.4 – Three-dimensional stream surfaces obtained by rotation about the engine axis.

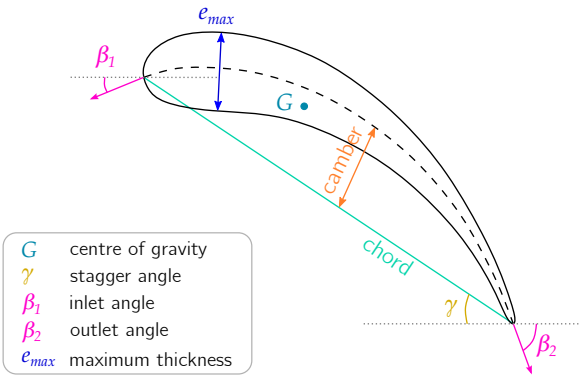


Figure 2.5 – Geometric parameters illustrated onto a schematic two-dimensional aerodynamic profile.

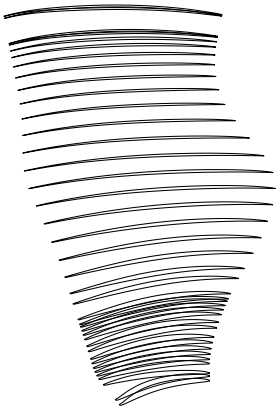


Figure 2.6 – Three-dimensional cross-sections corresponding to the flow lines pictured in figure 2.3.

2.1.1.3 Mechanical sizing

In this work, we are interested in the structural mechanics part of the process. In the following we shall give more details about the mechanical sizing of aircraft engines blades. Several physics come into play. We focus on the main types of loading that are relevant from a structural mechanics point of view.

Static As blades constitute spinning objects, they are subject to *centrifugal force*. Most of the time, centrifugal force is responsible for the major part of the mechanical effort acting on the blade. From a numerical analysis point of view, it requires the use of solid elements, as centrifugal force is a body load. This aspect is important in the context of our work, and motivates the design choices that are further set out.

In addition to centrifugal force, the *surface pressure* acting on the blades also contributes to the overall efforts. According to the position of the blade within the engine, the air is compressed or expanded. Aerodynamic profiles that compose the blade generate a pressure gradient between the so-called *pressure side* and *suction side*, as well as a velocity difference in the flow.

Thermal Then, *temperature loadings* are also relevant contributions to consider for the mechanical analysis of aircraft engines blades. A non-homogeneous temperature distribution is caused by the compression and expansion of the air in the turbine. Hence, a temperature gradient exists in the blades.

Dynamic Finally, *dynamic loadings* have to be considered. The dynamic analysis of compressor blades is crucial for their mechanical design. During their lifetime, both rotor and stator blades are subject to numerous vibration excitement sources due to the rotation of the engine. All phases of a flight imply different velocities, creating various excitement frequencies. Meanwhile, blades must fulfil their function during every flight phase. Their dynamic behaviour is thus extensively studied in practice, and a

series of criteria taking the form of frequency safety margins are used. The first excitation sources to consider are linked to the frequency harmonics of the engine rotation speed : 1N, 2N, 3N... Where N is the operating speed of the engine. These excitation sources are particularly important for blades with low natural frequencies. The second excitation source is related to the harmonic frequencies of the engine corresponding to the number of blades on the same stage and/or on neighbouring stages of the studied blade. Additional rules exist, whether the blade is mobile or fixed, but these are not considered in this work. Blades with high natural frequencies are more subject to this type of excitement. Whichever is the excitement source and the value of the associated frequency, a safety margin of 15% is commonly used (Adam et al., 1998). Therefore, given a rotation speed and an excitation source, a safety range can be defined around the associated frequency. The goal is then to prevent the blades from having any natural frequencies lying in that range. A usual way of visualising the frequency interactions of the blade/engine system is to draw a Campbell diagram (Campbell, 1924), also called an *interference diagram*. Natural frequencies of the blade are drawn along with the rotation speed of the engine for a particular operation regime, and the harmonic(s) of interest. An example diagram is given in figure 2.7.

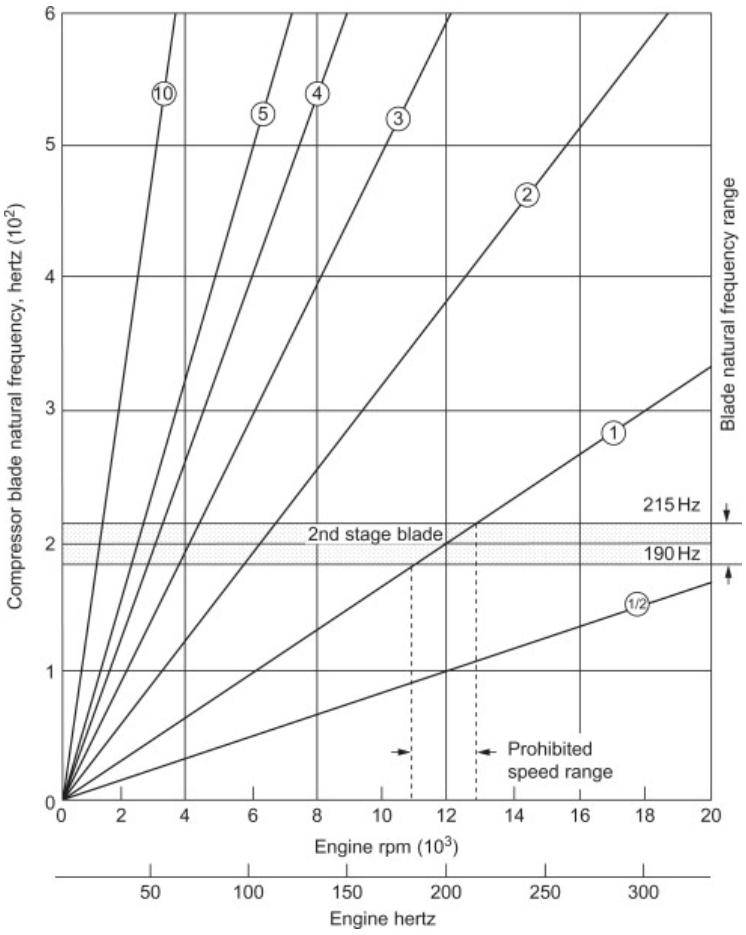


Figure 2.7 – Example of a Campbell diagram. (From Boyce, 2012).

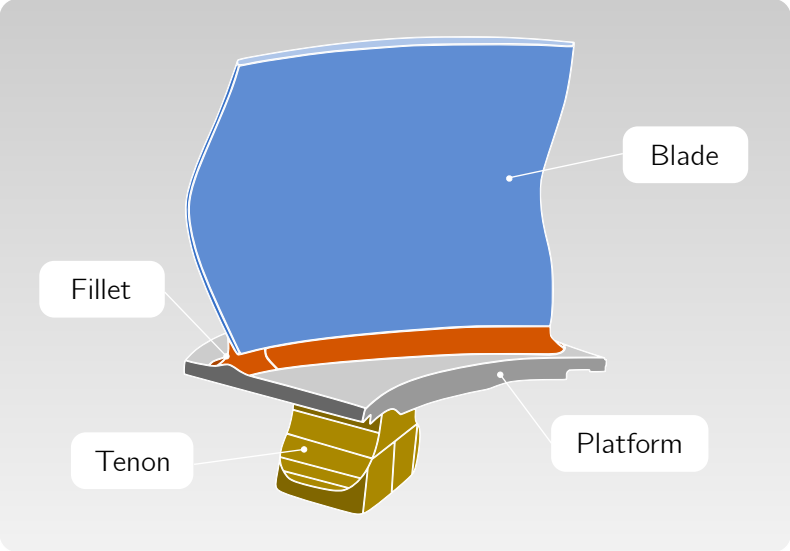


Figure 2.8 – Stylised example of a High-Pressure compressor blade assembly.

2.1.2 From analysis back to CAD

A well-known concern arising in large-scale industrial frameworks is the difficulty of making different entities to work together. Although it surely can refer to people, it applies to the flow of information as well, and more precisely to numerical representations of a same object.

Specifically, in the case of axial compressor blades design, this issue is present in the iterative process taking place between structural mechanics and aerodynamics teams for instance. More generally, the question of the link between Computer-Aided Design -defined geometries and analysis models remains an active subject. The surface definition of blades used for aerodynamic analyses is not suitable for structural mechanics purposes: as stated above, body forces need a solid description of the objects under consideration.

Another design stage that has not been mentioned yet is *structural optimisation*. The forthcoming section is dedicated to this subject, still we can say a few words about this aspect from a design point of view. Structural optimisation relies on the modification of the boundaries of an object in order to enhance its capacities for the desired application. After modifying the boundaries of an object, it has to be integrated anew in its environment. Coming back to blades, it can, for instance, consist in integrating the aerodynamic part back onto the supporting platform it lies on (see figure 2.8). In this context, the fillet has to be generated so as to link these two instances. To present date, this aspect remains a cumbersome manual operation.

2.2 In the search of improved designs

In this section we introduce the concept of structural optimisation, specifically in an industrial framework. We subsequently give more details about the mathematical aspects regarding the optimisation problem, and more precisely in the context of continuous deterministic optimisation. Finally, we depict the main structural optimisation techniques, and later concentrate on parametric shape optimisation, as this method is the one used throughout the presented work.

2.2.1 An essential aspect of the product development cycle

Traditional design processes strongly rely on the experience of the person in charge of the design. Actual technical requirements –among which efficiency, resource consumption, or mass for instance– have to be considered throughout the design process. The development of numerical tools, especially CAD and Finite Element Analysis, enable to improve the defined geometries numerically. Combining the expertise of the designer with mathematical background in order to explore the design space is an efficient approach to meet these goals.

Optimisation is an iterative procedure that aims at minimising (or maximising) a given quantity to improve the subject of the optimisation, eventually considering one or more constraints. In the context of mechanical engineering, it can classically concern the improvement of the rigidity of a structure, while restricting the resulting mass. This problem is especially relevant in an aeronautical context, where the mechanical behaviour is of utmost importance. The mass reduction aspect is a common concern in industry, as a direct way of reducing the raw material consumption –hence reducing the costs and environmental impact of the overall structure.

Structural optimisation refers to the process of defining an optimal configuration of a structure, be it through its size, shape or topology. To optimise structures, numerical techniques like FEA are coupled with optimisation algorithms. More details are given in the forthcoming paragraphs regarding the mathematical background of optimisation. Structural optimisation is subsequently addressed in section 2.2.3.

2.2.2 Optimisation problem

Throughout this subsection we shall give more mathematical background for the notions described previously. We emphasise we are only considering the situation covered by continuous deterministic optimisation. We strongly rely on the book from Nocedal and Wright (2006) for the various concepts that are exposed hereinafter. Cavazzuti (2013) also proposes a description of the theory of optimisation, with an emphasis on applications for concrete engineering structures and technical systems.

2.2.2.1 Mathematical formulation

Optimisation consists in minimising (respectively maximising) a given *objective function* which is subject to *constraints*, acting on its *variables*. The optimisation problem can be written as follows:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \quad \text{subject to} \quad \begin{cases} c_i(\boldsymbol{x}) = 0, & i \in \mathcal{E}, \\ c_i(\boldsymbol{x}) \geq 0, & i \in \mathcal{I}. \end{cases} \tag{2.1}$$

The variables $\boldsymbol{x}$ are also denoted as *unknowns* or *parameters* for the optimisation. They are represented as a vector-valued quantity. The objective function (or *cost function*) f is a scalar-valued function of the variables. Finally, the constraints c_i are scalar functions depending on the variables as well, that enforce equalities or inequalities the variables have to satisfy. The sets $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively.

The optimisation problem (2.1) is written in a particular form. Reformulations are usually needed for solving a specific optimisation problem so as to write it in the same form. Specifically, if maximisation is sought rather than minimisation, the function f can be taken as $-f$.

2.2.2.2 Optimisation algorithms

Stochastic *vs.* deterministic methods Once the optimisation problem is correctly formulated, the type of optimisation algorithm has to be selected. This choice is based on the optimisation problem and its application. Two types of approaches can be identified: *stochastic* and *deterministic* methods.

Modelling real-world situations might induce that several aspects of the model cannot be determined in advance. Instead of "guessing" these unknown quantities, the model can be enriched throughout the optimisation process by using some *a priori* knowledge specified by the person in charge, *e.g.*, an estimate of the probability of a scenario. *Stochastic optimisation* algorithms actually optimise the *expected* performance of the model, based on the provided additional knowledge. This approach is proven useful when considering non-linear problems with high dimensionality, especially. Well-known stochastic optimisation algorithms inspired by biological or physical processes include Genetic Algorithms, Simulated Annealing, or Particle Swarm Optimisation for instance.

In *deterministic optimisation* problems, on the contrary, the model is completely known. Deterministic optimisation is also referred to as *mathematical programming*, as deterministic optimisation algorithms strongly rely on linear algebra. As will be described in the following, this type of algorithms is usually based on the computation of the gradient, and often of the Hessian, of the cost function and the constraints functions, if any. Two great advantages of this approach are that: **1.** it requires less function evaluations, and

2. results drawn from a deterministic optimisation process are replicable. We consider this type of problem further in this manuscript.

Whichever is the approach chosen for solving a given optimisation problem, we emphasise that optimisation algorithms are iterative. Starting from an initial guess for the variables $\boldsymbol{x}$, a sequence of improved estimates is generated, until a criterion is satisfied to stop the iterative process — at which point an optimal value of the objective function is reached, eventually. Optimisation algorithms are distinguished in the methodology applied to advance from one iterate to the next. The quantities in use may classically include the value of the objective function f and constraint functions c_i, and their first and second derivatives.

Global *vs.* local minima The straightforward way of minimising a given cost function f would be to find a value $\boldsymbol{x}^*$ such that:

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \ \forall \boldsymbol{x} \in \mathbb{R}^n, \tag{2.2}$$

where $\boldsymbol{x}^*$ is called the *global minimiser* of f. Finding such a value is however not a simple task, mainly because the knowledge of f is only local in the iterative optimisation process. Hence, the aim of most optimisation algorithms is to reach for a *local minimiser* of the objective function, *i.e.*, a parameter that evaluates to the smallest value in the neighbouring region. The question we shall try to answer in the forthcoming paragraph is: *how to efficiently identify local minima given a smooth function f?*

Identifying a local minimum If the function f under consideration is smooth and twice continuously differentiable, the information given by the gradient ∇f and the Hessian $\nabla^2 f$ is sufficient to ensure that $\boldsymbol{x}^*$ is a local minimum. Taylor's theorem is the appropriate mathematical tool to study minimisers of smooth function. This theorem at hand, it is possible to draw *necessary conditions* for optimality, assuming that $\boldsymbol{x}^*$ is a local minimum of f. Under these assumptions on the variables and the objective function, the first- and second-order necessary conditions relate to the first and second derivatives of the objective function, respectively, and read as:

$$\nabla f(\boldsymbol{x}^*) = 0 \text{ and } \nabla^2 f(\boldsymbol{x}^*) \text{ is positive semi-definite.}$$

In other words, the first-order necessary conditions state that any local minimiser must be a stationary point. The second-order necessary conditions guarantee that the minimiser of f is a strict local minimiser.

Strategies for iterates determination Practically, optimisation algorithms iteratively evaluate the quantities of interest in order to determine if the current value of $\boldsymbol{x}^*$ satisfies the optimality conditions stated above. One concern lies in defining when to stop the iterative process, *i.e.*, when the parameters are sufficiently close to the optimal ones.

Before coming to this question, another matter arises, namely how this iterative process takes place. As previously mentioned, what differentiates one optimisation algorithm from another is the method used to compute the value for the next iterate. Two main techniques exist: the *line search* method and the *trust-region* method. We set out the main components for each approach in the following.

Line search method The line search method consists in finding a descent direction p_k, and searching in this direction for a new iterate $\boldsymbol{x}_{k+1}$ that will improve the current value $\boldsymbol{x}_k$. A step length α_k has to be computed in order to determine the distance to move along p_k. It is computed by minimising a linear problem:

$$\min_{\alpha>0} \quad f(\boldsymbol{x}_k + \alpha_k\, p_k). \tag{2.3}$$

These two parameters are computed for each iterative step of the process (*i.e.*, for each new iterate), until convergence.

Trust-region method The trust-region method uses the previously computed values of the cost function in order to model its behaviour in the region of interest, *i.e.*, some region around $\boldsymbol{x}_k$ (the trusted region). The model function m_k has to be minimised considering the candidate step p:

$$\min_{p} \quad m_k(\boldsymbol{x}_k + p), \tag{2.4}$$

where $\boldsymbol{x}_k + p$ lies in the trust region. If the decreasing in f is too small, the trust-region is reduced and the sub-problem (2.4) is solved once again. The model function m_k usually is defined as a quadratic function of the following form:

$$m_k(\boldsymbol{x}_k + p) = f_k + p^T\, \nabla f_k + \frac{1}{2}\, p^T\, \boldsymbol{B}_k\, p, \tag{2.5}$$

where f_k and ∇f_k denote the values of the function and function gradient respectively, evaluated at the point x_k. The matrix $\boldsymbol{B}_k$ represents the Hessian $\nabla^2 f_k$ or its approximation.

In broad terms, line search and trust-region methods are actually based on comparable quantities: a direction of descent and a step length. For the line search method, the direction is first fixed, then a satisfying step length is computed. For the trust-region method, a radius is first fixed, then the best possible direction is computed. Hence, the major issue for the line search method is to choose the direction p_k; for the trust-region method, it is to compute the Hessian matrix $\boldsymbol{B}_k$. Interestingly, these two concerns meet at some point, where a suitable approximation of the Hessian matrix has to be computed. A popular formula for the Hessian approximation is the *Broyden-Fletcher-Goldfarb-Shanno formula* (BFGS formula in short) :

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k - \frac{\boldsymbol{B}_k\, \boldsymbol{s}_k\, \boldsymbol{s}_k^T\, \boldsymbol{B}_k}{\boldsymbol{s}_k^T\, \boldsymbol{B}_k\, \boldsymbol{s}_k} + \frac{\boldsymbol{y}_k\, \boldsymbol{y}_k^T}{\boldsymbol{y}_k^T\, \boldsymbol{s}_k}, \tag{2.6}$$

where:

$$\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k, \tag{2.7}$$
$$\boldsymbol{y}_k = \nabla f_{k+1} - \nabla f_k. \tag{2.8}$$

Constrained optimisation Up to this point we did not mention how the constraints from the optimisation problem (2.1) are considered. In this case the first-order necessary conditions for $\boldsymbol{x}^*$ to be a local minimiser are written as follows. Assuming that $\boldsymbol{x}^*$ is a local solution of (2.1), that the functions f and c_i in (2.1) are continuously differentiable, and that the set of active constraints gradients is linearly independent at $\boldsymbol{x}^*$. Then, there is a Lagrange multiplier vector $\boldsymbol{\lambda}^*$ with components λ_i^*, $i \in \mathcal{I} \cup \mathcal{E}$, such that the following conditions are satisfied at $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$:

$$\nabla_x \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = 0, \tag{2.9a}$$
$$c_i(\boldsymbol{x}^*) = 0, \quad \forall i \in \mathcal{E}, \tag{2.9b}$$
$$c_i(\boldsymbol{x}^*) \geq 0, \quad \forall i \in \mathcal{I}, \tag{2.9c}$$
$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I}, \tag{2.9d}$$
$$\lambda_i^* \, c_i(\boldsymbol{x}^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}, \tag{2.9e}$$

with the Lagrangian function defined as:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i \, c_i(\boldsymbol{x}). \tag{2.10}$$

The conditions (2.9) are known as the *Karush-Kuhn-Tucker conditions* (KKT conditions in short).

Algorithms for constrained optimisation Three main algorithmic approaches can be identified in the realm of non-linear constrained optimisation:

- penalty and augmented Lagrangian methods,
- Sequential Quadratic Programming (SQP),
- interior-point methods for non-linear programming.

All these techniques make use of *quadratic programming*, namely an optimisation problem consisting in quadratic objective function and linear constraints. *Penalty and augmented Lagrangian* methods use a penalty function associating the cost function and constraints. The problem (2.1) is addressed by solving a series of unconstrained problems. *Sequential Quadratic Programming* methods are based on the approximation of the optimisation problem (2.1) at each iteration, using a quadratic programming sub-problem. The solution of this sub-problem gives the search direction for the next iterate. *Interior-point methods for non-linear programming* can be seen as a prolongation of the interior-point methods for linear programming.

Practical aspects The numerical experiments presented in this work make use of an SQP algorithm based on the implementation from Kraft (1994), using a BFGS update of the $\boldsymbol{B}$ matrix. It is available as the `SLSQP` method from the `optimize` module of the open-source `SciPy` library for non-linear optimisation (Virtanen et al., 2020). This gradient-based algorithm can handle optimisation domains defined using bound constraints, as well as both non-linear inequality and equality constraints.

2.2.3 Structural optimisation

Having detailed the mathematical background for optimisation problems, let us return to design matters. In the following we introduce the various types of structural optimisation, and further concentrate on parametric shape optimisation.

2.2.3.1 Different classes of structural optimisation

Classically, three categories of structural optimisation are identified, each corresponding to a specific need from an engineering point of view. They can be combined, yet are usually used at different stages of the product development process. Figure 2.9 illustrates these categories.

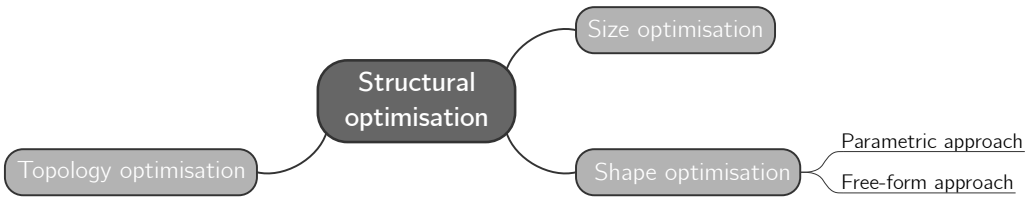


Figure 2.9 – Overview of structural optimisation.

Sizing optimisation *Size optimisation* (see figure 2.10) refers to the modification of a sizing parameter of a structure. For instance, it can be the cross-section of trusses in a truss-structure, the thickness of metal sheets forming a chassis, etc. Thus, sizing optimisation does not modify the overall shape of the structure under consideration, but rather acts on a specific geometric quantity involved in the mechanical response.

Shape optimisation As for *shape optimisation* (see figure 2.11), the goal is to modify the shape of the structure. On the one hand, the *parametric* shape optimisation technique is based on geometric parameters describing the shape of the structure. The geometric parameters must be carefully selected as meaningful quantities for the sought application. Hence, the design possibilities are restricted to the chosen parameters. This aspect can reveal itself limiting in some applications, and drastically reduces the design exploration possibilities. On the other hand, the *free-form* shape optimisation approach does not restrict the investigation to the existing boundary of the structure. The shape is defined

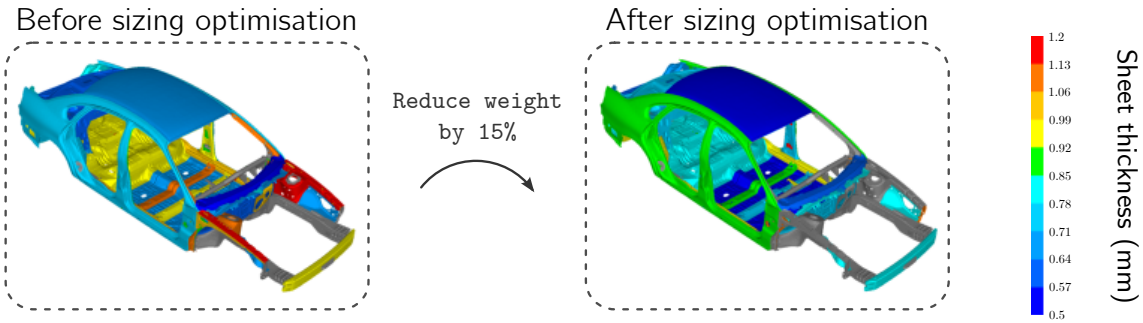


Figure 2.10 – Example of structural optimisation using sizing optimisation. In this example the overall weight of a car chassis is reduced by 15% by adjusting the metal sheets thickness. (Adapted from Keonys).

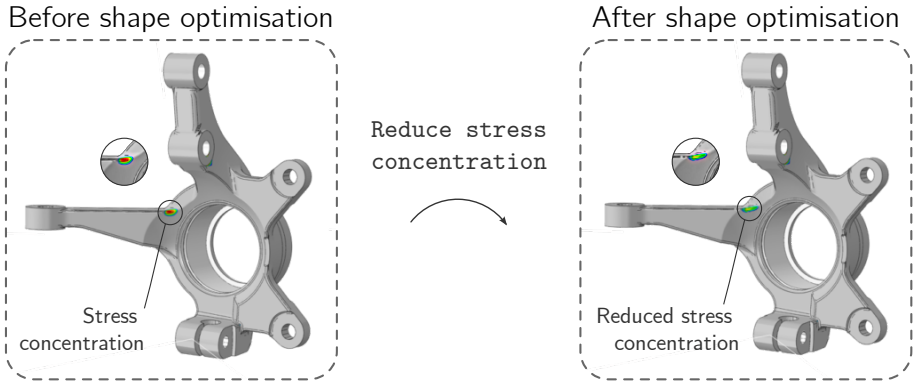


Figure 2.11 – Example of structural optimisation using shape optimisation. The stress concentration pointed out on the left-hand side is reduced. (Adapted from Keonys)

by the boundaries, which can be perturbed freely to attain an optimal geometry. No *a priori* knowledge of the relevant areas is needed in order to perform free-form shape optimisation, allowing for more freedom than its parametric counterpart regarding the design space exploration.

Topology optimisation Finally, *topology optimisation* (see figure 2.12) enables to modify the topology of the domain under consideration. Classically, a domain is defined, inside which the optimisation process can add or withdraw material, based on the provided boundary conditions and loadings. Hence, the topology of the initial geometry is modified as some boundaries and topological elements (*e.g.*, holes) may have appeared or disappeared.

As stated above, these types of structural optimisation are conceptually different, and are used at different times of the design process. For instance, topology optimisation and free-form shape optimisation are typically used as exploratory techniques, since the resulting shape and/or topology might be far different from the initial one. Sizing optimisation may come into play at the end of the design process, as it is gen-

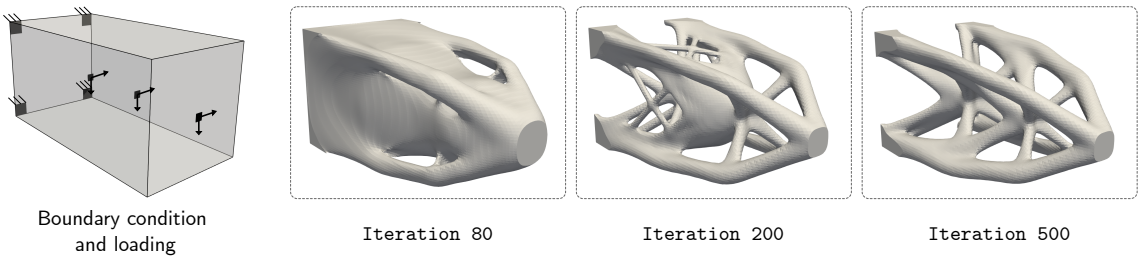


Figure 2.12 – Example of structural optimisation stages, using topology optimisation. Starting from bulk material repartition, the weight is reduced throughout the iterative process. (Courtesy of Djourachkovitch, 2020).

erally involved in the improvement at the assembly scale. Finally, parametric shape optimisation can be used at any stage of the design process. Its relative restrictive nature can be seen as a way of controlling the shape modifications: when a product has reached an advanced stage of the design process, one may want to improve it within a certain (restricted) range of variation.

2.2.3.2 Parametric shape optimisation

Structural optimisation using parametric shape optimisation is a field of study that is now established, the seminal references dating back to the 1970s. Improving structural performance still is an up-to-date concern, hence shape optimisation remains an active research area. Several reviews for this field have been published, *e.g.*, Vanderplaats (1993) for early publications, and Daxini and Prajapati (2017) for later ones. More recently Upadhyay et al. (2021) proposed a review of advances since the 2000s. Classically, shape optimisation relies on three modules (represented on the diagram in figure 2.13):

- geometry description and modification (*design model*),
- structural analysis (*analysis model*),
- optimisation process (*optimisation model*).

A challenging aspect in this framework is the link between those three representations of the structure under consideration. Usually, the geometry is described by means of a CAD software, while the shape modifications are handled by the geometric parameters defined by the user (the *design variables*). Structural analysis is typically carried out using Finite Element Methods or Boundary Element Methods. The optimisation process consists in using information provided by both the design and analysis model. The first provides the current design variables values while the latter gives access to the mechanical behaviour of the structure. These pieces of information are used to compute new design variables that will lead to a better design from the optimisation problem point of view, that is, variables that satisfy the constraint(s) — if any — while minimising the cost function.

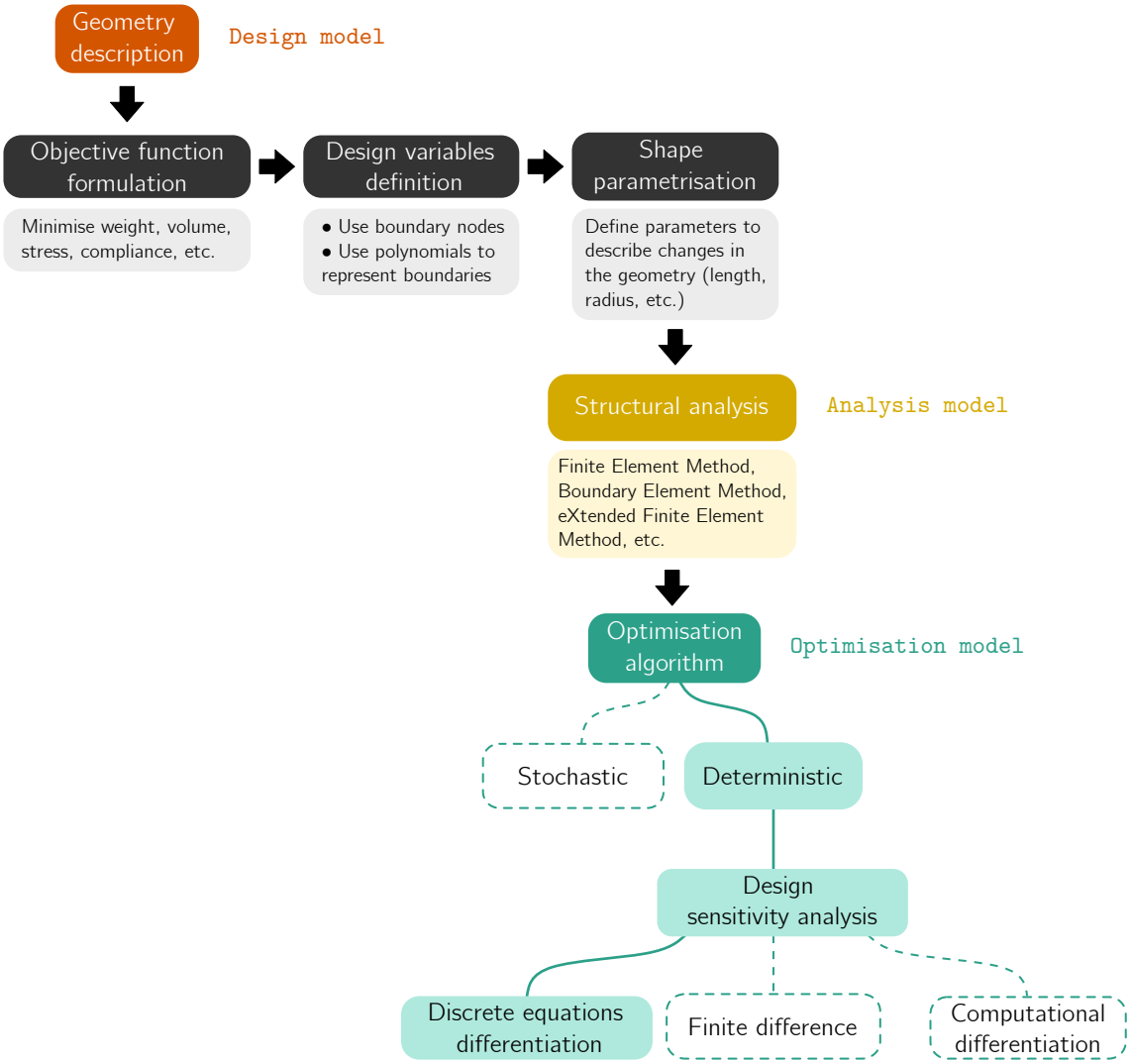


Figure 2.13 – Overview of the structural optimisation framework in the case of parametric shape optimisation. The highlighted steps are the ones used for this work. (Adapted from Daxini and Prajapati, 2017).

In order to ensure the connection from the design to the analysis model, two main strategies have arisen: the first consists in using the CAD definition for the parametrisation of the shape, while the other uses the spatial locations of nodes of a Finite Element mesh as design variables. The CAD-based approach requires the re-meshing of the geometry for every new set of design variables values. The FE-based strategy, apart from presenting a high number of design variables, can lead to distorted geometry. Mesh regularisation techniques can be employed to reduce this tendency, adding computational cost. In addition, mesh morphing or free-form approaches can be used to reduce the number of design variables and reduce mesh distortion. That being said, the reconstruction of a CAD model from a deformed mesh remains a difficult task. Even though both CAD-based and FE-based techniques have drawbacks, they still are widely used in industrial frameworks. Seeking alternative methods to integrate the geometrical definition and the structural analysis is of high interest, especially in the context of structural shape optimisation.

Whichever is the strategy used to convert the geometric description to an analysis model for structural analysis, the optimisation model has to be provided with several pieces of information. As previously mentioned, the analysis model is used to evaluate relevant quantities for the optimisation process to take place, based on the geometric modifications applied to the design model. More precisely, the analysis module supplies the quantities involved in the cost function and constraints functions, if any. As presented in section 2.2.2, the variations of these quantities are needed. This is what sensitivity analysis is about: evaluate the gradients of the quantities involved in the optimisation problem formulation. We subsequently describe three strategies to obtain these quantities in the case of gradient-based optimisation processes.

Sensitivity analysis Using a gradient-based optimisation algorithm necessarily leads to the question of the design sensitivity analysis. The sensitivities consist in the differentiation of the structural expression with respect to the design variables. The information brought by the analysis of sensitivities is of crucial importance as it gives a measure of the performance of the improvement brought by the geometry update during the optimisation process. Several techniques exist in order to compute those variations. We list hereafter the main three approaches. For illustration purposes, we will tackle the example of a smooth function $g : \mathbb{R}^n \mapsto \mathbb{R}$ for which we want to draw the partial derivative with respect to the design variable component x_i (*i.e.*, with respect to the i-th component of the design variables vector $\boldsymbol{x}$), written as $\dfrac{\partial g}{\partial x_i}$.

Discrete equations differentiation The most straightforward approach from a conceptual point of view in order to draw the partial derivatives of g is to calculate them by hand. It can be achieved for self-adjoint problems, resulting in an efficient computation of the sensitivities in terms of computational cost. The main drawback is that analytical

derivation may lead to intricate developments in more complicated cases. Automated methods exist to compute the derivatives as we will see in the following.

Finite difference Finite-difference methods are very popular as they are rather easy to implement. They are based on the approximation of a given partial derivative by using infinitesimal perturbations of the design variables vector x to evaluate the function response. For instance, the *forward-difference* approximation reads as:

$$\frac{\partial g}{\partial x_i}(\boldsymbol{x}) \approx \frac{g(\boldsymbol{x} + \varepsilon\, \boldsymbol{e}_i) - g(\boldsymbol{x})}{\varepsilon}, \tag{2.11}$$

where ε is a small positive scalar, and $\boldsymbol{e}_i$ is the i-th unit vector. Apart from its relative simplicity, the other advantage of this approach is that no analytical knowledge of the function g is needed. Its main drawback concerns the computational cost, as it becomes expensive as the number of design variables increases. Indeed, the number of evaluations of the quantity (2.11) is as numerous as the number of design variables. What is more, since it depends on an evaluation parameter ε, it can suffer accuracy problems.

Computational differentiation Also known as *automated differentiation*, this method is based on software tools that decompose the evaluation of the function g into elementary arithmetic operations. The chain rule is then applied to these operations in order to obtain the expression of the derivatives. It benefits from a good accuracy while avoiding tedious developments of expressing the derivatives by hand.

As a concluding remark regarding the search of optimal design, let us sum up the numerical tools we use in this work. We base our set-up on parametric shape optimisation. As mentioned in section 2.1, shape optimisation is preferentially used as a way of controlling the variations of shape of the final product, and not shatter the efforts of previous design stages regarding the numerous constraints that must be considered for aircraft engines blades. The parametric aspect is an important argument in this context, as most relevant quantities from an aerodynamic point of view are directly linked to geometrical parameters. Hence, controlling the shape of a blade *via* a series of geometric parameters falls impeccably in the framework we situate ourselves into. The approach chosen for the design sensitivity analysis is the discrete equations differentiation, given that we use a gradient-based algorithm. Further in this chapter we give more details about the framework we use for the computation of analytical sensitivities (*i.e.*, how we derive the discrete equations).

2.3 Isogeometric analysis

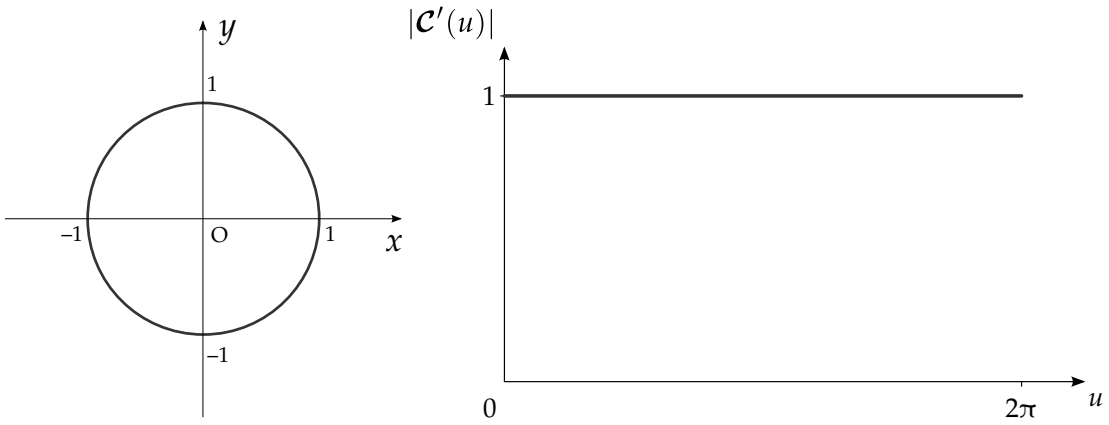
In the following section we first introduce the concept of IsoGeometric Analysis as a promising strategy for integrating the geometric design and analysis. We further give

more elements about the parametric approach for geometric design, which is a crucial aspect when considering NURBS. Definition and properties of B-Splines functions are later exposed, and geometric design using NURBS is depicted. Finally, we give an overview of NURBS-based IGA, with an emphasis on the similarities and differences with respect to classical FE based on Lagrange polynomials. We rely on the book of Piegl and Tiller (1996) for the properties and definitions of B-Splines and NURBS, and on the book of Cottrell et al. (2009) for the theoretical and implementation aspects of IGA.

2.3.1 Integration of the geometric design and analysis as an objective

Introduced by Hughes et al. (2005), IGA is a numerical method that aims at improving the link between geometrical design and analysis. It relies on the use of B-Splines or NURBS functions in order to both represent the geometry and carry out analysis. Hence, as in standard FEA, the isoparametric concept is invoked. The core difference between the two approaches is that IGA yields an exact representation of the geometry to compute the unknown fields during analysis, while in classical FEA the mesh actually approximates the geometry. Another attractive feature of IGA lies in its computational precision, partly brought by the higher continuity of B-Spline functions, in comparison with Lagrange polynomials for instance. B-Spline functions present other interesting properties to be used in the IGA framework, such as efficient refinement procedures.

The precise and smooth unique geometry description is one of the main improvements over previous use of B-spline shape functions in finite elements (see for example Ching-Tze, 1979, Kagan et al., 1998). IGA has been used in many different types of problems where its precise geometric description and smooth higher order basis functions provide interesting results. From the simple single patch geometries solved in the seminal paper of Hughes et al. (2005) for elasticity and advection-diffusion, more advanced physics have been tackled, such as turbulent flows (Bazilevs et al., 2010, Van Opstal et al., 2017), large deformations elasticity and plasticity (Elguedj et al., 2008), higher order PDEs (Gómez et al., 2008, Liu et al., 2013), thick and thin shells (Kiendl et al., 2009, Echter et al., 2013, Bouclier et al., 2015), fluid structure interaction (Bazilevs et al., 2008, Hsu and Bazilevs, 2012), contact and friction (Temizer et al., 2011, Matzen and Bischoff, 2016), etc. Over the years, bottlenecks of IGA were also addressed such as local refinement with alternative splines (Dörfel et al., 2010, Vuong et al., 2011, Johannessen et al., 2014), complex geometries with multiple patches and trimming (Schmidt et al., 2012, Ruess et al., 2014, Coox et al., 2017) as well as generating analysis-suitable geometric models (Xu et al., 2013a, Al Akhras et al., 2016).



(a) Representation of the curve in $\mathbb{R}^2$. **(b)** Plot of the velocity as a function of the parameter value u.

Figure 2.14 – Example of parametric curve in $\mathbb{R}^2$.

2.3.2 Geometric design with B-Splines and NURBS

2.3.2.1 Parametric design

Before entering to the heart of the matter, let us broach the subject of parametric design. Geometric modelling strongly relies on parametric description of curves (respectively surfaces). The parametric form of a curve in $\mathbb{R}^2$ can be defined as:

$$\mathcal{C}(u) = (x(u), y(u)). \tag{2.12}$$

Where u is an independent parameter defined on an arbitrary increasing interval $[a, b]$, $a \leq b$, usually normalised to $[0, 1]$. Hence, each of the coordinates of a point (x, y) on the curve is represented separately as a function of the variable u, and $\mathcal{C}(u)$ can be seen as a vector-valued function.

As an illustrative example, let us take the case of a circle of unit radius spanning in $\mathbb{R}^2$, centred at the origin, as pictured in figure 2.14a. Following the previous curve definition, its parametric form can be written as:

$$\begin{cases} x(u) = \cos(u) \\ y(u) = \sin(u) \end{cases}, \quad u \in [0, 2\pi]. \tag{2.13}$$

The parametric approach is useful for describing a time-dependant quantity. In the case of the circle, let us assume that the position (x, y) varies as a function of the time u. Then, the first and second derivatives of the position respectively represent the velocity and acceleration. The first derivative, yielded by the differentiation of (2.12), reads as:

$$\mathcal{C}'(u) = (x'(u), y'(u)) = (-\sin(u), \cos(u)). \tag{2.14}$$

An interesting observation in that context is that the magnitude of the velocity vector $\mathcal{C}'(u)$ is a constant (see figure 2.14b):

$$|\mathcal{C}'(u)| = \sqrt{\sin^2(u) + \cos^2(u)} = 1, \tag{2.15}$$

meaning that the displacement speed is constant. This property is referred to as *uniform parametrisation*, or in this case *constant-speed parametrisation*. The latter is precisely illustrated by this example. For a more general appreciation, we can reformulate this observations as follows: there is a linear relation between the variation of the parameter u and the value of $\boldsymbol{\mathcal{C}}(u)$. The uniformity of the parametrisation is a notion that is further addressed in this manuscript.

Parametric surfaces are represented as vector-valued functions of two parameters u and v. In $\mathbb{R}^3$, we have:

$$\boldsymbol{\mathcal{S}}(u,v) = (x(u,\, v),\, y(u,\, v),\, z(u,\, v)), \tag{2.16}$$

u and v being defined on arbitrary intervals $[a,\, b]$, $a \leq b$, and $[c,\, d]$, $c \leq d$, respectively. Note that a curve can be defined in $\mathbb{R}^3$ as well, by adding a third z-coordinate to its definition for instance.

We can state some interesting observations and properties of parametric curves and surfaces that will prove themselves useful thereafter, specifically in a geometric design context. First, parametric forms are rather natural ways of representing a geometry in computer graphics. Indeed, parametric curves and surfaces present a natural direction of traversal. As parameters take only increasing values, this direction is determined by the way the curve (or surface) is parametrised. Then, it is a straightforward process to generate ordered sequences of points along a parametric curve. Likewise, a lattice of points can be generated onto a surface. Ultimately, computing a point on a curve or surface only requires a parameter value. This operation is referred to as *point evaluation*. A drawback of parametric definition concerns the reverse operation, *i.e.*, determine if a given point is lying on the curve or surface (*point projection*). A related problem is to determine the parameter corresponding to this given point, known as *point inversion*.

2.3.2.2 Definition and properties of B-Spline basis functions

B-Spline basis functions are further used for the construction of B-Spline curves, surfaces, and volumes (and of their rational counterparts, NURBS). After defining B-Spline basis functions, we list some properties about them. Some terminology is also addressed in the following. Yet again we refer to the *NURBS book* (Piegl and Tiller, 1996) for the forthcoming paragraphs.

Definition B-Spline basis functions are piecewise polynomial basis functions that can be computed recursively. The recursive formula is known as the Cox-de Boor formula, named after its authors (Cox, 1972, De Boor, 1972).

The B-Spline basis functions are computed using a sequence of non-decreasing parameters $\boldsymbol{U} = \{u_1,\, u_2,\, \ldots,\, u_m\}$. The u_i ($i = \{0,\, 1,\, \ldots,\, m-1\}$) are further referred to as the *knots*. The sequence $\boldsymbol{U}$ is called the *knot vector*. The basis function denoted as $N_{i,\,p}(u)$

is the i-th function of degree p. It is defined as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1}, \\ 0 & \text{else,} \end{cases} \tag{2.17a}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - ui} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \tag{2.17b}$$

Hence, in order to compute basis functions, a knot vector $\boldsymbol{U}$ and a degree p are required. From equation (2.17a) one can notice that the $N_{i,0}(u)$ are piecewise constants: they are 0 everywhere except on the interval $[u_i, u_{i+1}]$. The interval $[u_i, u_{i+1}]$ is later referred to as the i-th *knot span*.

Knot vector As stated above, the knot vector $\boldsymbol{U}$ is a set of non-decreasing parameters. Hence, a parameter value can be repeated, *i.e.*, we have $u_i \leq u_{i+1}$. The knot spans are defined as intervals for which $u_i < u_{i+1}$. The *multiplicity* of a knot refers to the number of repeated knot values in $\boldsymbol{U}$. Further terminology for knot vectors concerns the repartition of knot values. A knot vector is said *uniform* if the knot values are uniformly spaced, and *non-uniform* otherwise. Additionally, an *open* knot vector denotes a knot vector that has its first and last knots repeated $p + 1$ times, where p is the degree. It is *closed* otherwise. In the following we only consider the open case. What is more, we consider the parameters sequence as normalised to $[0, 1]$.

Properties Some properties of B-Spline functions are listed hereafter. Since B-Spline curves are based on the latter, their properties derive from those.

Local support The support of each function $N_{i,p}(u)$ is compact and included in the interval $[u_i, u_{i+p+1}]$. That is, $N_{i,p}(u) = 0$ outside this interval.

Non-negativity The $N_{i,p}(u)$ are non-negative, *i.e.*:

$$N_{i,p}(u) \geq 0 \quad \forall\, i,\, p,\, u. \tag{2.18}$$

Partition of unity An interesting property of B-Spline basis functions formed from an open knot vector is that they constitute a partition of unity, *i.e.*:

$$\forall\, u \in \boldsymbol{U}, \quad \sum_{i=1}^{n} N_{i,p}(u) = 1. \tag{2.19}$$

Differentiability All derivatives of $N_{i,p}(u)$ exist in the interior of a knot span. For a given knot u of multiplicity k, $N_{i,p}(u)$ is $p - k$ times continuously differentiable.

Continuity If a knot has a multiplicity of k, basis functions at this knot will have a C^{p-k} continuity. Likewise, the continuity of the derivatives of B-Spline basis functions decreases of k for knots of multiplicity k. Therefore, increasing the degree increases the continuity, while increasing knot multiplicity decreases the continuity.

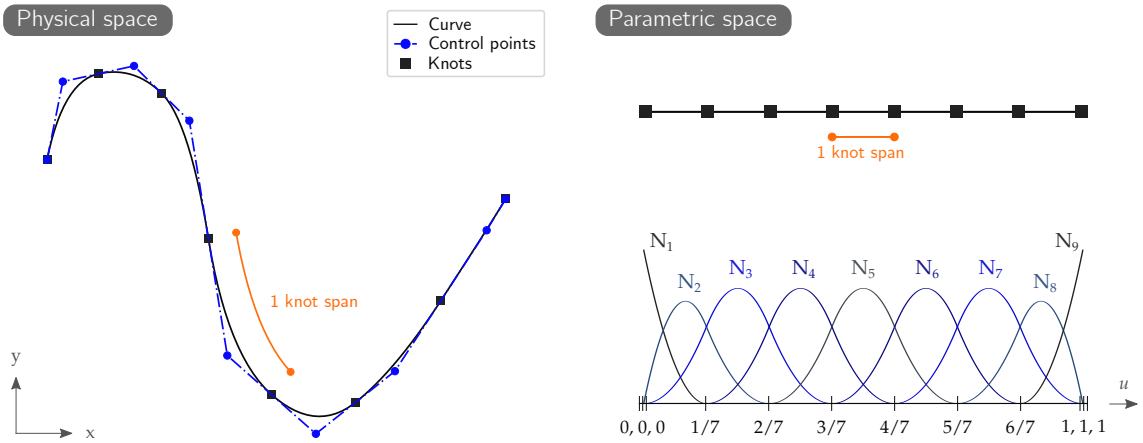


Figure 2.15 – Physical and parametric spaces for a quadratic B-Spline curve in $\mathbb{R}^2$. The associated knot vector is defined as $\left\{0, 0, 0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1, 1, 1\right\}$.

2.3.2.3 B-Spline and NURBS geometries

B-Spline curves A B-Spline curve $\boldsymbol{\mathcal{C}}(u)$ of degree p is defined by linear combination of products of B-Spline basis functions $N_{i,p}(u)$ and control points $\boldsymbol{P}_i$:

$$\boldsymbol{\mathcal{C}}(u) = \sum_{i=1}^{n} N_{i,p}(u)\, \boldsymbol{P}_i, \tag{2.20}$$

where $i \in \{1, 2, \ldots, n\}$, $n \in \mathbb{N}$, denotes the index of the function, u is a parameter on the knot vector $\boldsymbol{U} \subset \mathbb{R}$, and the $\boldsymbol{P}_i$ are vector-valued quantities. If $\boldsymbol{P}_i \in \mathbb{R}^2$, the resulting B-Spline curve will span in $\mathbb{R}^2$. Likewise, it will span in $\mathbb{R}^3$ if the control points are defined in $\mathbb{R}^3$.

We emphasise that B-Spline curves are parametric curves, hence they present the properties described in section 2.3.2.1. As a linear combination of B-Spline basis functions, they inherit most of their properties. This aspect is partly illustrated in figure 2.15, where *physical* and *parametric* spaces are represented. The first is the place where knots are defined, whereas the latter is where the control points are spanned, and hence where the curve lies. The knot spans in parametric space can be taken for elements in a finite-element sense. An interesting observation highlighted therein is the fact that a uniform knot vector will not necessarily span regularly spaced elements in physical space. If we compare this example to the parametric circle represented in figure 2.14, it means we do not traverse the curve at a constant pace by passing from a knot to the following.

B-Spline surfaces and solids For multi-dimensional approximations, the basis functions are defined as a tensor product of one-dimensional basis functions. For example, given a control net $\boldsymbol{P}_{ijk}$ of points in $\mathbb{R}^3$, degrees p, q, and r and intervals $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$

with associated knot vectors, a B-Spline solid is defined on $\boldsymbol{U} \times \boldsymbol{V} \times \boldsymbol{W}$ as:

$$\boldsymbol{\mathcal{V}}(u, v, w) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} N_{i,p}(u)\, M_{j,q}(v)\, L_{k,r}(w)\, \boldsymbol{P}_{ijk}. \tag{2.21}$$

Non-Uniform Rational B-Splines The rational counterparts of B-Splines are known as NURBS. Using the same basis functions, an additional component is introduced: the so-called *weights*. They are used to define the *rational basis functions*:

$$R_{i,p}(u) = \frac{N_{i,p}(u)\, w_i}{\sum_{j=0}^{n} N_{j,p}(u)\, w_j}, \tag{2.22}$$

where w_i are the weights, p is the degree, and $N_{i,p}(u)$ are the afore-defined basis functions. Then, the definition of a p-th degree NURBS curve reads as:

$$\boldsymbol{\mathcal{C}}(u) = \sum_{i=0}^{n} R_{i,p}(u)\, \boldsymbol{P}_i. \tag{2.23}$$

The rational basis functions $R_{i,p}(u)$ are piecewise rational functions on $u \in [0, 1]$. Their properties are similar to those of their non-rational counterparts. It is worth to note that B-Splines are a special case of NURBS, for which all weights are equal to 1. We hereafter list a series of geometric properties of B-Splines, without loss of generality, as they also are valid for NURBS.

2.3.2.4 Geometric properties of B-Splines

Using the previous observations about parametric design and properties of B-Spline basis functions, we subsequently list some properties of B-Spline entities. This list is not exhaustive, yet is aimed at focusing on points that are further used for geometric modelling matters.

Continuity and differentiability The continuity and differentiability of a curve $\boldsymbol{\mathcal{C}}(u)$ arises from those of the B-Spline basis functions $N_{i,p}(u)$. Then, $\boldsymbol{\mathcal{C}}(u)$ is at least $p - k$ times continuously differentiable at a knot of multiplicity k.

Non-interpolatory aspects In the general case, the interior control points are not situated on the curve. The exception lies for curves of degree $p = 1$, which thus interpolate the control points, and for knots of multiplicity $k = p$. For all cases featuring an open knot vector, the endpoint control points are interpolated, *i.e.*: $\boldsymbol{\mathcal{C}}(0) = \boldsymbol{P}_0$ and $\boldsymbol{\mathcal{C}}(1) = \boldsymbol{P}_n$.

Local support The local support property of B-Spline basis functions leads to the *local modification scheme* for B-Spline curves. That is, a given control point $\boldsymbol{P}_i$ has influence on the portion of the curve situated in the interval $\left[u_i,\, u_{i+p+1}\right]$.

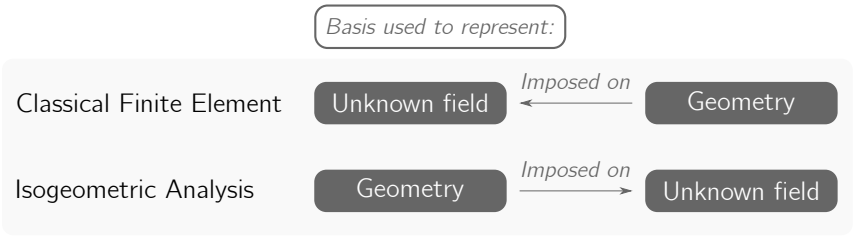


Figure 2.16 – Reversing the isoparametric arrow. (Adapted from Cottrell et al., 2009).

Affine invariance An affine transformation is applied to the curve by applying it to the control points. This property stems of the partition of unity property of B-Spline basis functions.

2.3.3 NURBS-based IGA

Concept As in standard FEM, NURBS-based IGA relies on the *isoparametric concept*, which induces the use of the same shape functions for the approximation of the unknown fields and the geometric description of the domain. Cottrell et al. (2009) used the expression of "reversing the isoparametric arrow" to summarise the core difference between classical FEA and IGA (see figure 2.16). In a nutshell, it consists in using the NURBS basis functions in order to solve the analysis, which happen to be the same functions as the ones used to define the geometry. It is "reversed" in the sense that for classical FEM, the basis functions (*e.g.*, Lagrange polynomials) used for the approximation of the solution are used to describe the geometry. It results in the approximation of the initial geometry.

Practically, in order to build the solution of the problem in the context of IGA, assuming we search for the approximation of the displacement $\boldsymbol{u}^h$ in any point of the structure, we can write that:

$$\boldsymbol{u}^h = \sum_{A=1}^{n} N_A \boldsymbol{u}^A, \tag{2.24}$$

where the N_A are the NURBS basis functions and $\boldsymbol{u}^A$ are the degrees of freedom. The reader who is familiar with finite element methods will certainly get his or her bearings, as the form (2.24) is the same.

Refinement procedures Refinement is an essential aspect when tackling numerical methods, relying on the discretisation of the domain under consideration. Based on the previous paragraph, it is worth to note that the tensor-product structure inherent to NURBS modelling prevents refining locally. This concern has been raised ever since the beginning of isogeometric methods, and several technologies have been developed in order to circumvent it. The most famous technique is the *T-Splines* technology, enabling to create T junctions (Sederberg et al., 2003, Scott et al., 2011, Evans et al., 2014). *LR*

B-Splines have proven to be an efficient alternative for local refinement (see Johannessen et al., 2013, Occelli et al., 2019). *Hierarchical splines* have been proposed as well as a solution (Vuong et al., 2011, Giannelli et al., 2012, Buffa and Giannelli, 2016, Giannelli et al., 2016).

Apart from local refinement, refinement procedures in the context of NURBS modelling have proven to be very efficient. An interesting aspect with respect to the functions used in classical FE is that NURBS basis functions can used with a combination of h- and p-refinement to achieve the so-called *k-refinement*. We remind that h-refinement refers to reducing the element size, while p-refinement consists in increasing the element order. A noticeable aspect for classical FE is that using refinement modifies the geometry, as the spatial location of nodes is altered. In IGA, it is not the case: the geometry remains untouched when the basis is refined, one way or another. The numerical tools used for h- and p-refinement are *knot insertion* and *degree elevation*, respectively.

- *Degree elevation*, or *order elevation* refers to the elevation of the polynomial degree of the shape functions. In this case, the knot values constituting the knot vector must be repeated as many times as the degree is elevated.
- *Knot insertion* consists in adding knots to the knot vector.

Finally, using both theses numerical tools yields the k-refinement. This additional strategy results in a higher polynomial order as well as a higher continuity of the shape functions. Precisely, it is obtained by first applying order elevation onto the coarse geometry, then using knot insertion of multiplicity 1 to obtain the desired fine discretisation. Hence, high continuity is preserved, maintaining initial continuity at knots. This approach is the one used in this work.

To mesh or not to mesh? One of the main advantages of IGA highlighted in its early days (and to present day) is the direct link between the geometric model and the analysis model, hence avoiding the time-consuming meshing operation. While remaining an attractive aspect, this assertion could be moderated. Simply stating that the time dedicated to the generation of a proper meshing of the geometry is reduced to zero does not reflect the reality. To date, several studies have investigated the generation of *analysis-suitable* geometries in the context of IGA. A clear motivation for this approach is to be able to use CAD-defined geometries for the analysis, which usually is not the case. Two prominent reasons are first, that CAD geometries rely on trimmed entities, and second, that volumetric modelling is not made possible as solely surface definitions are used. Another important matter concerns the internal parametrisation of the resulting model (Lipton et al., 2010, Gravesen et al., 2012, Xu et al., 2013b, Wang and Qian, 2014, Farah et al., 2016, Al Akhras et al., 2017, Wu et al., 2017). Since knot spans are used as elements (in a finite element sense) for the analysis, their relative shape and distortion is important. Hence, IGA *does* suffer the need of ensuring a suitable discretisation and

the control of its overall aspect. Incidentally, these matters are further broached in this work.

2.4 On the usefulness of IGA

To conclude this chapter, we hereby propose an overview of the convenient framework that can be drawn for the analysis and shape optimisation of aircraft engines blades, using the isogeometric paradigm. This framework combines the aforementioned tools, methods and requirements to achieve our aims. More precisely, we first introduce the isogeometric shape optimisation framework. Based on the advantages brought by this approach, we finally outline the interest of making use of it, in the context of aircraft engines blades design process.

2.4.1 Isogeometric shape optimisation

The first work on isogeometric shape optimisation was introduced in Wall et al. (2008) and followed by many others considering different physics and optimisation techniques. Among this numerous literature, one could think of the optimisation of curved beams (Nagy et al., 2010, Choi and Cho, 2018, Weeger et al., 2019), plates and shells (Nagy et al., 2013, Kiendl et al., 2014, Bandara and Cirak, 2018, Hirschler et al., 2019b,c), heat conduction (Wang et al., 2017), electromagnetics (Nguyen et al., 2012, Manh et al., 2014), fluids (Park et al., 2013), and fluid-structure interaction (Kostas et al., 2015, 2017), etc. A more detailed literature review on isogeometric shape optimisation can be found in the review paper of Wang et al. (2018).

2.4.1.1 A favourable framework

In the present work, we adopt the now classical multi-level approach (see *e.g.*, Nagy et al., 2013, Kiendl et al., 2014) which consists in choosing different refinement levels of the same spline-based geometry to define the analysis and optimisation spaces. Using this strategy is possible because the geometric description is unchanged in the spline refinement process with h-, p- and k- refinements (see Cottrell et al., 2009). The use of this multi-level approach is summarised in Figure 2.17. A single numerical representation is used to define the geometry and carry out structural analyses: the difference between the design and analysis models is a question of refinement level. The spatial location of the control points of the design model is usually used as design variables, while the analysis model presents an adequate refinement level to guarantee the quality of the solution.

In many works considering isogeometric shape optimisation, gradient-based methods are used, starting from the early work of Wall et al. (2008). This requires sensitivity analysis to express the variation of physical quantities over the change of design

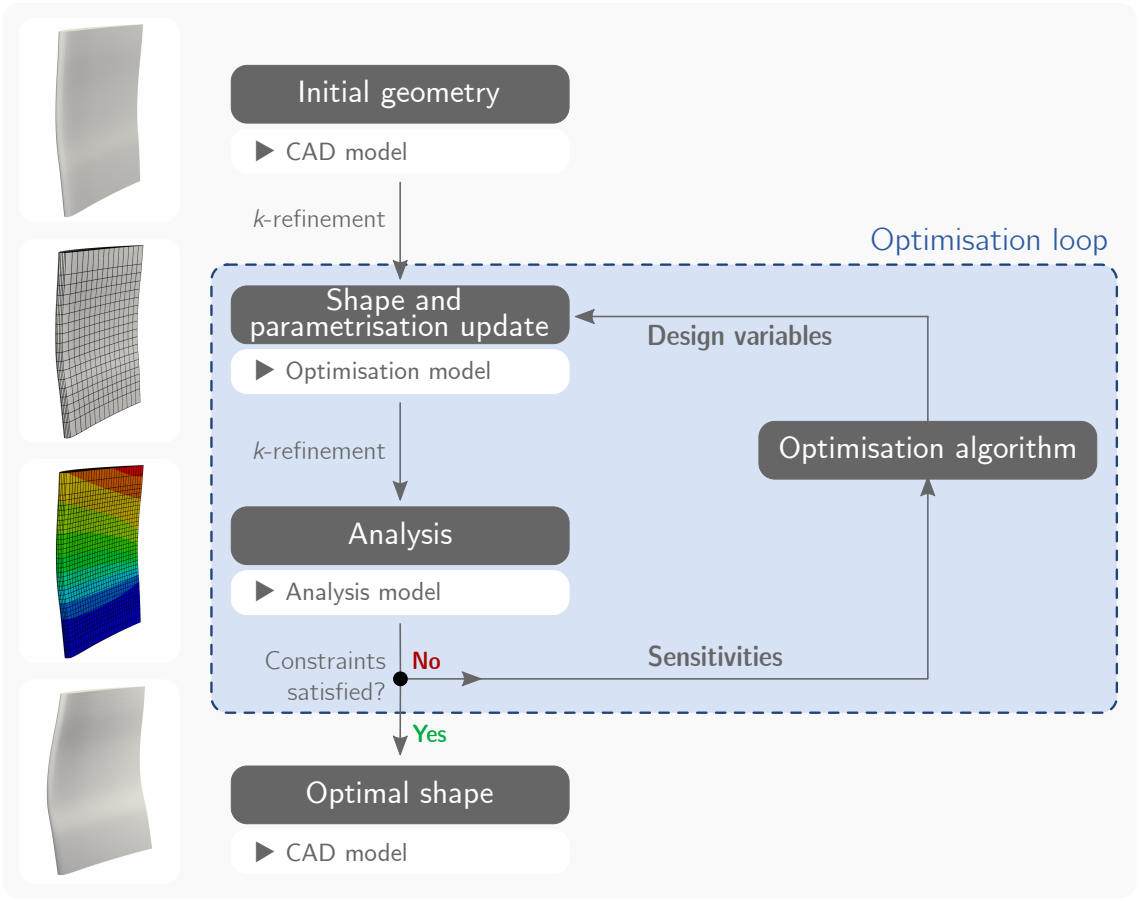


Figure 2.17 – Overview of the main steps of the isogeometric shape optimisation. A unique analysis-suitable geometric representation is used in such framework.

variables. For self-adjoint problems, such as compliance and natural frequencies, analytical sensitivities can be achieved. IGA provides a framework where it is possible to obtain compact and general expressions of analytical sensitivities for a reasonable computational cost, which is an improvement over other methods such as standard FEA. This has been detailed by Qian (2010) and Fußeder et al. (2015), and more recently by Hirschler et al. (2020). Finite difference and semi-analytical approaches have also been used and provide an alternative when analytical derivation leads to intricate developments ((Wang and Turteltaub, 2015, Wang and Poh, 2018)). For complex non-linear problems, a continuum-based configuration Design Sensitivity Analysis can be used (see Cho and Choi, 2000, Choi and Cho, 2000). Alternatively, gradient-free approaches are also interesting when considering isogeometric shape optimisation thanks to the lower computational cost of the analysis for a given accuracy, see for example Kostas et al. (2015) and Herrema et al. (2017). In the present work, we follow the framework introduced by Hirschler et al. (2020) to perform gradient-based optimisation using full analytical sensitivities.

Another important point is the fact that isogeometric shape optimisation is still a finite element-based method, hence it can suffer from the same drawbacks. In particular, mesh distortion is an issue during the optimisation process if the initial parametrisation of the geometry is not well chosen — although IGA is less sensitive to mesh distortion than standard FEA, see Lipton et al. (2010). In other words, the optimal geometry can depend on the initial discretisation. This has been observed and studied in Kiendl et al. (2014) and Wang and Poh (2018), where several ways to circumvent this aspect have been introduced.

2.4.1.2 Sensitivity analysis

Expressing the gradients of the objective function and of the constraints with respect to the design variables of the optimisation problem is a key aspect when using a gradient-based algorithm, which can lead to tedious developments. When these are not available, because they cannot be formally expressed or because it would be too difficult to do so, one way of getting around the issue is to use a finite-differences scheme or an automated differentiation method (Bücker et al., 2006, Bischof et al., 2008). As mentioned above, IGA provides a framework that allows to obtain compact and general expressions of analytical sensitivities for a reasonable computational cost when considering self-adjoint problems. This has been detailed by Qian (2010) and Fußeder et al. (2015), and more recently by Hirschler et al. (2020). The main steps are given in the following.

Objective or constraints functions can be expressed as:

$$f := g\left(x, \boldsymbol{u}(x)\right), \tag{2.25}$$

where $\boldsymbol{u}$ is the solution of the linear system:

$$\boldsymbol{K}(x)\,\boldsymbol{u} = \boldsymbol{F}(x). \tag{2.26}$$

Thus, the gradient of function f with respect to the design variable x_i, $i \in [1, d]$ can be decomposed as:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \frac{\partial g}{\partial \boldsymbol{u}} \cdot \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x_i}. \tag{2.27}$$

Introducing $\boldsymbol{u}^*$ as the solution of the following adjoint problem:

$$\boldsymbol{K}\boldsymbol{u}^* = \frac{\partial g}{\partial \boldsymbol{u}}, \tag{2.28}$$

equation 2.27 can be rewritten as:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \boldsymbol{K}}{\partial x_i}\boldsymbol{u} \right). \tag{2.29}$$

The derivatives of the stiffness matrix $\boldsymbol{K}$ and the load vector $\boldsymbol{F}$ with respect to design variables are computed on the analysis model and propagated to the optimisation model using the refinement operator $\boldsymbol{R}$, kept constant during the optimisation process:

$$\boldsymbol{Q} = \boldsymbol{R}\,\boldsymbol{P}. \tag{2.30}$$

$\boldsymbol{Q}$ and $\boldsymbol{P}$ being the coordinates of the control points respectively on the analysis and optimisation model. Derivatives can be expressed as:

$$\frac{\partial \bullet}{\partial \boldsymbol{P}} = \boldsymbol{R}^T \frac{\partial \bullet}{\partial \boldsymbol{Q}}. \tag{2.31}$$

And the sensitivity reads:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \frac{\partial \boldsymbol{P}}{\partial x_i} : \boldsymbol{R}^T \left(u^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} - \boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{Q}}\boldsymbol{u} \right), \tag{2.32}$$

where $\dfrac{\partial \boldsymbol{P}}{\partial x_i}$ is an operator linking the control points coordinates of the optimisation model with the design variables. In the case of linear elasticity, derivatives of $\boldsymbol{K}$ and $\boldsymbol{F}$ in the analysis model can be expressed as geometric quantities from the element formulation. The interested reader can refer to Hirschler et al. (2020) for more details on how to obtain analytical sensitivities in IGA.

2.4.2 Closing the design loop

Considering the aircraft engines field, integrating design and analysis in a single framework is one of the key aspect that led to the exploration of IGA in the present work. Another crucial feature is the transition from the shape optimisation resulting model back to the CAD environment. This conversion is essential for manufacturing purposes. What is more, describing the initial geometry of blades thanks to parametric curves such as B-Splines or NURBS has already been explored, not only for aircraft engines blades (Großmann and Jüttler 2010, Mykhaskiv et al. 2018), but also wind turbines blades (Stein et al., 2012, Herrema et al., 2017), or marine propellers (Pérez-Arribas and Pérez-Fernández, 2018, Arapakopoulos et al., 2019).

Concluding remarks

As a concluding remark for this chapter, let us broadly sum up the main elements leading to the chosen framework. First, we take advantage of the smoothness and higher-order approximation capacities brought by isogeometric analysis. Then, we consider using this approach for shape optimisation purposes. The sensitivities being expressed analytically, it results in an efficient method for gradient-based optimisation. Last, we take advantage of the interesting property of IGA to yield CAD-compatible models as a result of shape optimisation.

Chapter 3

From construction to shape optimisation of a simplified blade model

The present chapter focuses on applying isogeometric shape optimisation to aircraft engines blades design by integrating geometric description, structural analysis and shape optimisation in a single industrial framework. Starting with stacked cross sections of the blade geometry as an input, we construct a B-spline analysis-suitable volumetric model of the blade, ensuring its geometric accuracy and parametrisation regularity. We perform structural shape optimisation using suitable industrial loading cases and objective functions onto industrial blades shapes. The results demonstrate the efficiency of the method and its relevance for industrial aircraft engine blades design and optimisation.

Contents

3.1 Construction of a trivariate B-Spline blade model

3.1.1 Chosen approach and framework

The first step towards the isogeometric shape optimisation of blades is to generate a trivariate B-Spline blade model that is suitable for analysis and shape optimisation. From an analysis point of view, the repartition of knots spanned in physical space is of crucial importance. A harmonious repartition of knots leads to a regular discretisation of the geometry, and by extension, of the finite element mesh. As for shape optimisation, we aim for using the spatial location of control points as design variables. Hence, keeping the number of control points to a reasonable level (*i.e.*, not too high) is a key point to ensure computational efficiency. *How to comply with the aforementioned requirements while guaranteeing geometrical precision of the model with respect to the initial definition?* In order to answer this question, an overview of the proposed approach is given hereafter.

Single patch model Various strategies arise when it comes to generate a trivariate B-Spline model, in terms of number and spatial organisation of patch(es) in particular. In our case, the topology of a blade lends itself to a single patch representation. For this purpose, we suppose the geometry of the blade as homomorphous to a cube (see figure 3.1). The method we set up consists in identifying the elementary geometric entities that assemble into a cube: four curves can form a closed surface, and six surfaces assemble into a closed volume. In order to identify these entities, we based ourselves on the work from Großmann and Jüttler (2010), that deal with the reconstruction of a blade geometry.

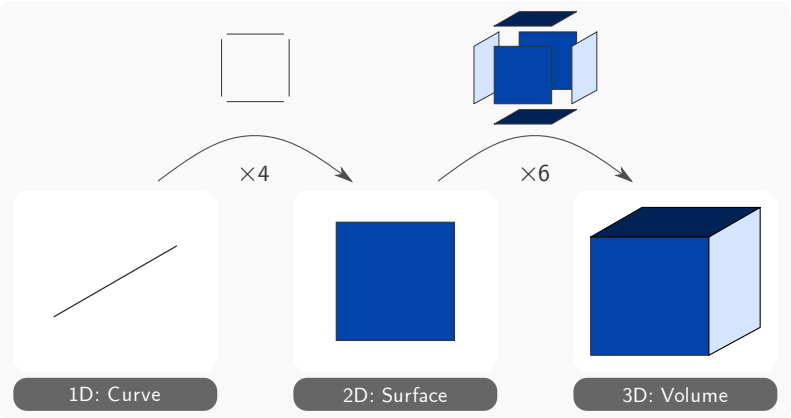


Figure 3.1 – Elementary geometric operations resulting in a volumetric cube.

Overview Figure 3.2 outlines the main steps of the methodology proposed in this work to generate a trivariate B-Spline blade model. The procedure is summed up hereafter.

1) Create the set of 3D B-Spline curves forming the stacked blade profiles.

2) For each of these curves, identify four *wedge points* and split each profile accordingly. This step yields four (stacked) curves groups: leading edge, pressure side, trailing edge and suction side.

3) Adjust the parametrisation of all created curves.

4) Create four surfaces from the four curves groups generated in step 2 (the so-called *contour* surfaces).

5) Adjust the parametrisation of the surfaces.

6) Create the two missing surfaces, *i.e.*, *top* and *bottom* surfaces, by interpolating linearly the curves forming the first and last profiles.

7) Compute the volume control points by linear combination of the control points of the six outer surfaces.

Mock-up blade model Throughout this chapter several geometric operations are detailed. Given the relative complexity of the shapes we have to deal with, several aspects are difficult to illustrate on an actual blade geometry. This is why we chose to build a mock-up blade geometry to clarify several points. This example is further used in chapter 4 for the same reasons.

3.1.2 Proposed methodology

Step 1 – Initial curves

Our starting point for the construction of a trivariate B-Spline blade model is a series of B-Spline curves. The curves actually consist in aerodynamic profiles, that are stacked in a vertical fashion. Hence, the curves solely define the outer surface of the blade.

Step 2 – Splitting curves

The goal for this step is to subdivide every profile in four portions. First, the four points where the curve is to be split have to be identified. Having figure 3.1 at hand, it actually consists in identifying the four corners of the square we reduce the profiles to. In order to achieve this, we adopted the same method as developed by Großmann and Jüttler (2010). The division of each profile is approximatively the one used in aerodynamics: leading edge, suction side, trailing edge, pressure side. We emphasise that our division is not relevant from an aerodynamic point of view, and is used only for geometry purposes.

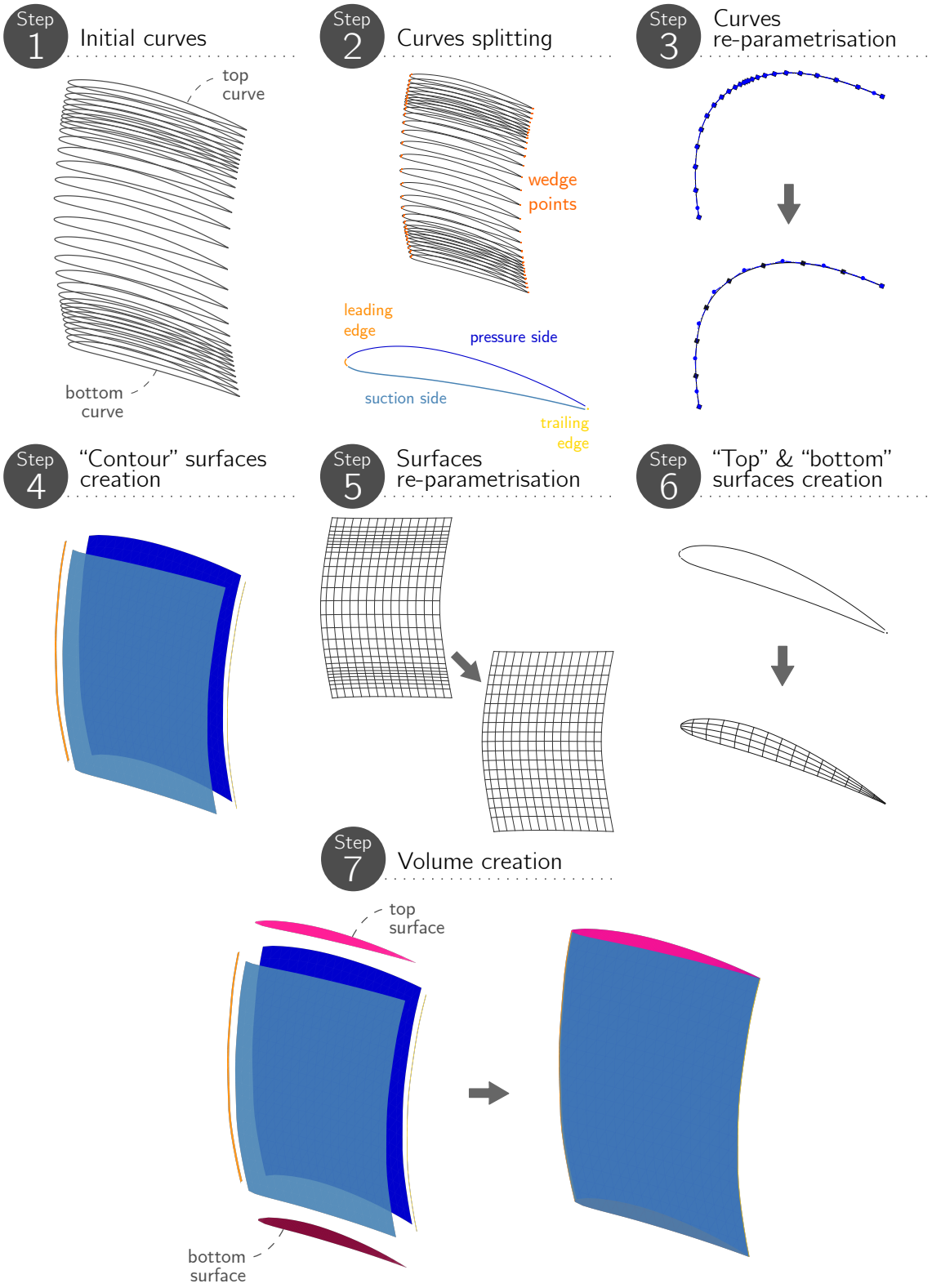


Figure 3.2 – Overview of the framework for the volumetric reconstruction of a blade.

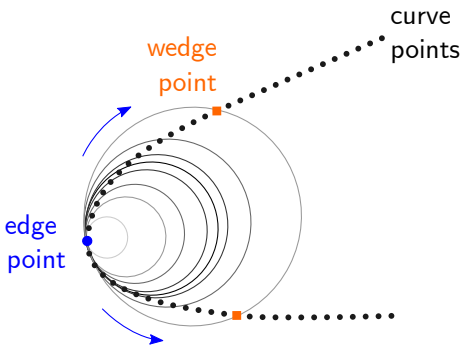


Figure 3.3 – Process of fitting the edge points to a sphere. The operation is illustrated in the two-dimensional case for clarity.

A geometric criterion has been established in order to choose these points, in the leading and trailing edges regions: an iterative method based on fitting the points to a sphere (see figure 3.3). Starting by the foremost point of the profile (the *edge point*), a point is added on either side each iteration and the error corresponding to the sphere fit is computed. The process ends when the computed error exceeds a user-defined value, *i.e.*, when the quality of the sphere fit becomes too poor. The process of fitting the points to a sphere includes an additional constraint, as the centre of the adjusted sphere is enforced to belong to the regression plane formed by the sampled points. A Gauss-Newton scheme taking this constraint into account has been implemented in order to execute the fitting process.

The points where each profile is to be divided (the *wedge points*) are the last points added during the iterative process. Once these points have been identified for each profile, the curves are split by using multiple knot insertion. This step produces four times as much curves as the initial number. In the following, the curves are processed according to their groups (leading edge, pressure side, trailing edge and suction side).

Step 3 – Curves re-parametrisation

A preliminary condition for constructing a surface from curves is to ensure they are compatible in a B-Spline sense. In our case, it implies that all curves belonging to the same group must share a common knot vector. Another requirement in this work is the repartition of the curves' knots in physical space, so that the final isogeometric mesh is regular. We propose a procedure that ensures both conditions at once, by using two different algorithms. An overview of the procedure is presented in figure 3.4.

The first step consists in obtaining a smooth distribution of points being sampled on a given curve. A parameters sequence that complies with this requirement has to be computed in order to evaluate the curve. As a matter of fact, Hernández-Mederos and Estrada-Sarlabous (2003) proposed an iterative algorithm to sample points on parametric curves with control of their distribution. More precisely, the `UniArcLength` algorithm

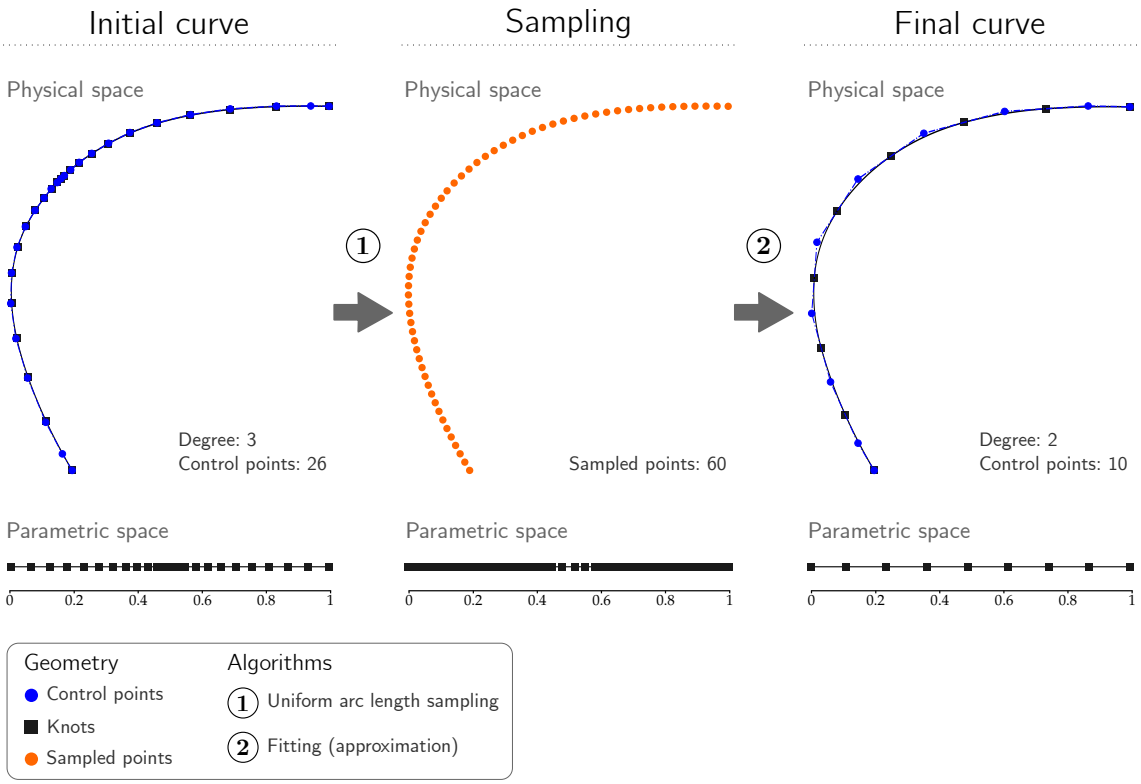


Figure 3.4 – Curves re-parametrisation steps. The process is illustrated for a leading edge curve; here the interest of obtaining a coarser parametrisation appears clearly.

from their work enables to compute a parameters sequence that results in approximately evenly spaced points in physical space. In this algorithm, the parameters distribution is obtained by computing the arc length value. The length L of a parametric curve $\mathcal{C}$ between parameters a and b reads as:

$$L = \int_a^b \|\mathcal{C}'(u)\|_2 \mathrm{d}u \tag{3.1}$$

In the implementation, the arc length value is approximately computed using a Gaussian quadrature. The length of the curve evaluated between two consecutive parameters is computed each iteration. The loop stops when the condition on the parameters repartition is met. The *Sampling* step from figure 3.4 illustrates the resulting sampled points.

Afterwards, the regularly spaced points in physical space are used to carry out a curve fitting using a curve approximation technique. Approximation (as opposed to interpolation) offers the possibility to control the degree and the number of control points of the final curve. These aspects are important in our case, keeping in mind that we run these steps for all curves from a same group so that they are compatible in a B-Spline sense. What is more, it appears that using evenly spaced points for fitting a curve leads to regularly spanned knots in physical space for the resulting curve. Now focusing

on the approximation process in itself, we use an approximation algorithm such as the one presented in Piegl and Tiller (1996, pp. 410-413) — namely a least squares curve approximation. The unknowns for this minimisation problem are the control points coordinates. Let us recall the expression of a non-rational curve of degree p:

$$\boldsymbol{\mathcal{C}}(u) = \sum_{i=1}^{n} N_{i,\,p}(u)\, \boldsymbol{P}_i \quad u \in [0,1]\,, \tag{3.2}$$

with $p \geq 1$, and $n \geq p$. The points to approximate are the $\mathbf{Q}_0, \ldots, \mathbf{Q}_m$ $(m > n)$. The first and last points are interpolated, *i.e.*: $\mathcal{C}(0) = \mathbf{Q}_0$ and $\mathcal{C}(1) = \mathbf{Q}_m$. For the remaining $\mathbf{Q}_k$, the following mean squared distance is to be minimised with respect to the $n+1$ variables $\mathbf{P}_i$:

$$\sum_{k=1}^{m-1} |\mathbf{Q}_k - \boldsymbol{\mathcal{C}}\left(\overline{u}_k\right)|^2, \tag{3.3}$$

where the $\overline{u}_k$ are pre-computed parameter values.

For the computation of the parameter values (*i.e.*, the knot vector of the approximation curve), several options are proposed therein. In our case, we begin by running a first approximation for all curves with the same prescribed number of control points. The knots vectors of all curves, computed by centripetal method (see Piegl and Tiller, 1996, pp. 364-365), are then averaged to compute a unique knot vector. All curves are approximated again with the same number of prescribed control points as the first stage, this time enforcing the knot vector to the averaged one. At the end of the day, all curves from a same group share a common knot vector and present a coarser — yet smoother — parametrisation.

Step 4 – Contour surfaces creation

Creating the contour surfaces of the blade is straightforward. Given a group of compatible curves, the grid of control points of the surface is formed by the control points of the curves — each curve providing a row of control points. In the first parametric direction, the one corresponding to the curves, the parametrisation is smooth and controlled (in terms of degree and number of control points). On the contrary, the parametrisation of the second parametric direction, which corresponds to the stacking direction, is not. For this direction we use a uniform knot vector, that is part of the initial inputs. Hence, another re-parametrisation step is required for the contour surfaces to have a controlled parametrisation in the stacking direction.

Step 5 – Contour surfaces re-parametrisation

Analogously to step 3, we seek to achieve regularly spanned knots in physical space, looking at the contour surfaces this time. In this case we have to handle two parametric directions at once, yet only one of them has to be re-parametrised.

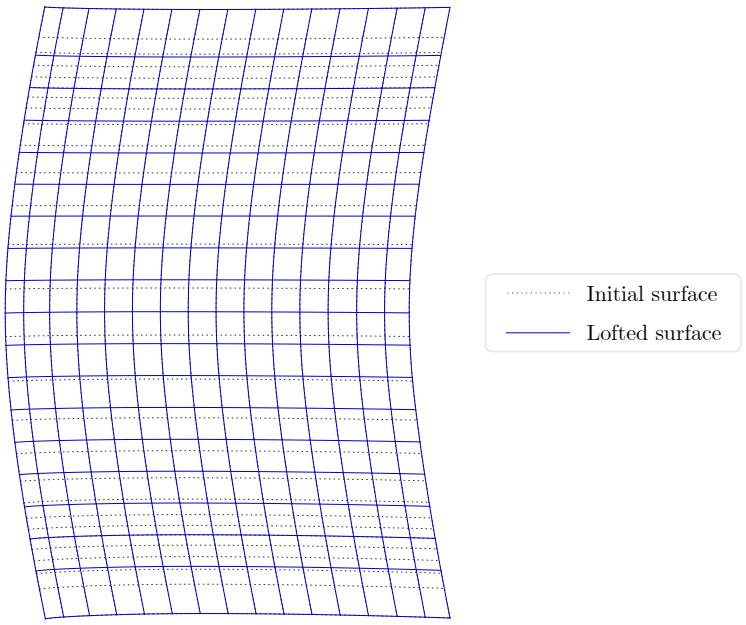


Figure 3.5 – Comparison between initial and re-parametrised contour surface.

Firstly, a parameters sequence is computed once again thanks to the `UniArcLength` algorithm for the direction requiring re-parametrisation. However, unlike step 3, the goal is not to sample a lot of points on the surface, but rather compute the right number of parameters regarding the final number of IGA-elements defined by the user. Then, this sequence is used to perform multiple knot insertion in order to retrieve the parametric curves of the surface corresponding to these parameters. Indeed, knot insertion is useful to extract isoparametric curves on a given surface, (see Piegl and Tiller, 1996, p. 161). "New" section curves are thus obtained, enabling to build the re-parametrised surface.

To that end, we use the surface construction technique called *surface skinning*, or *lofting* (see Piegl and Tiller, 1996, pp.457-471). It consists in blending a series of section curves together: the section curves then form the parametric curves of the resulting surface. The advantage of this method is that we can control the degree and number of control points in the blend direction, which corresponds to the stacking direction in physical space in our application. Another favourable feature is that the control points of the re-parametrised curves from step 3 are *interpolated*. It is an important detail because it ensures the coincidence of the corner control points of the surface, a crucial aspect considering that these surfaces form the boundary of the geometry-to-be. A comparison between an initial and re-parametrised surface is given in figure 3.5.

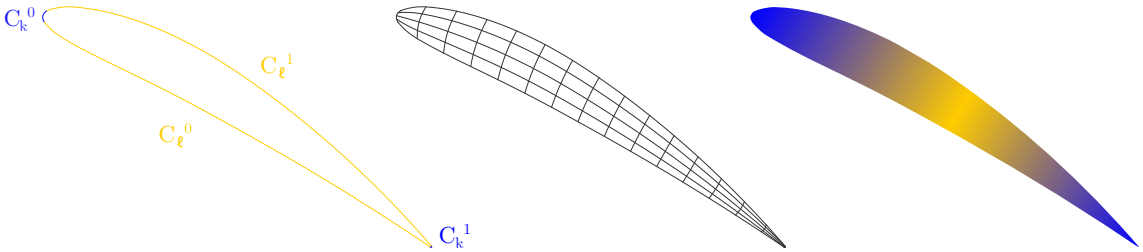


Figure 3.6 – Example of a Coons patch. From left to right: initial boundary curves, parametric lines, and final surface.

Step 6 – Creation of top and bottom surfaces

We finally refer to figure 3.1 to stress our point on boundary surfaces. After creating the contour surfaces, one may notice two more surfaces are needed in order to close the boundaries of the blade geometry — they are referred to as *top* and *bottom* surfaces. We have to use a different construction technique to create them since they only are defined by their boundaries.

A common way of computing interior control points to create a surface in that case is to use the well-known *Coons patch*, (see Coons, 1967). It consists in a bilinear interpolation of boundary curves in order to create a surface. The four curves must define a closed boundary (*i.e.*, their endpoints must be coincident two by two. Creating a Coons patch amounts to create three surfaces: one interpolating the $\mathcal{C}_k$ curves, one interpolating the $\mathcal{C}_\ell$ curves, and one interpolating the four corners formed by the curves. The three surfaces interpolating the $\mathcal{C}_k$ and $\mathcal{C}_\ell$ curves are also known as *ruled surfaces*. These surfaces are then linearly combined to create the final patch. A detailed implementation of Coons surfaces can be found in Piegl and Tiller (1996, pp.496-507) as a special case of the so-called *Gordon surfaces*. This implementation does not require the boundary curves to be compatible in a B-Spline sense.

In our case, the boundary curves are formed by the first and last blade profiles, for the bottom and top surfaces respectively. The boundary curves already are compatible by means of the re-parametrisation step. The final parametrisation of the top and bottom surfaces is therefore controlled, and make them compatible with the contour surfaces. An example is featured in figure 3.6.

Step 7 – Final volume creation

Finally, having the six boundary surfaces at hand, the inner control points of the trivariate B-Spline model can be computed. Their coordinates are computed by linear interpolation of the control points of the six boundary surfaces, using the trivariate generalisation of a Coons patch. The detailed expression can be found in Xu et al. (2011), based on the work from Farin and Hansford (1999). In the following, let us denote the control points as c_{ijk}, with $i \in \{0, \dots, l\}$, $j \in \{0, \dots, m\}$, $k \in \{0, \dots, n\}$. Boundary control

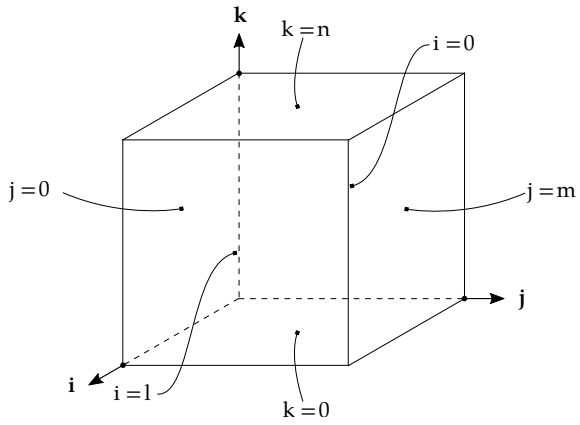


Figure 3.7 – Control points indices for a volumetric Coons patch.

points are then identified by c_{0jk}, c_{ljk}, c_{i0k}, c_{imk}, c_{ij0}, and c_{ijn} (see figure 3.7). Here it is assumed that the boundary surfaces under consideration are B-Spline surfaces, and that opposite surfaces share the same degree and knot vectors. To complete the compatibility assumptions, neighbouring surfaces must share the same control points on their boundaries — which is the two-dimensional version of endpoints compatibility for boundary curves, for using the same explanation as step 6.

Having fulfilled all these requirements beforehand enable to use the following expression to obtain the inner control points:

$$
\begin{aligned}
c_{ijk} = {} & (1 - i/l)\, c_{0jk} + i/l\, c_{ljk} + (1 - j/m)\, c_{i0k} \\
& + j/m\, c_{imk} + (1 - k/n)\, c_{ij0} + k/n\, c_{ijn} \\
& + [1 - i/l,\ i/l] \begin{bmatrix} c_{00k} & c_{0mk} \\ c_{l0k} & c_{lmk} \end{bmatrix} \begin{bmatrix} 1 - j/m \\ j/m \end{bmatrix} \\
& - [1 - j/m,\ j/m] \begin{bmatrix} c_{i00} & c_{i0n} \\ c_{im0} & c_{imn} \end{bmatrix} \begin{bmatrix} 1 - k/n \\ k/n \end{bmatrix} \\
& - [1 - k/n,\ k/n] \begin{bmatrix} c_{0j0} & c_{lj0} \\ c_{0jn} & c_{ljn} \end{bmatrix} \begin{bmatrix} 1 - i/l \\ i/l \end{bmatrix} \\
& + (1 - k/n) \left[[1 - i/l,\ i/l] \begin{bmatrix} c_{000} & c_{0m0} \\ c_{l00} & c_{lm0} \end{bmatrix} \begin{bmatrix} 1 - j/m \\ j/m \end{bmatrix} \right] \\
& + k/n \left[[1 - i/l,\ i/l] \begin{bmatrix} c_{00n} & c_{0mn} \\ c_{l0n} & c_{lmn} \end{bmatrix} \begin{bmatrix} 1 - j/m \\ j/m \end{bmatrix} \right].
\end{aligned}
$$

The position of interior control points can later be adjusted if necessary, see Xu et al. (2011) for instance.

3.1.3 Application to two compressor blades models

The methodology detailed in section 3.1 is now applied on geometries representative of a low-pressure compressor fan blade and of a HP compressor rotor blade. The resulting models are illustrated in figure 3.16. They are used in sections 3.2.1 and 3.2.2 to perform a static and a modal analysis using IGA.

We chose these two blade geometries in order to put emphasis on various aspects of this work. First, the two blades present different levels of geometrical complexity. The HP compressor blade is small in height and does not present a highly "folded" geometry. On the contrary, the fan blade is rather slender and highly twisted. Note in particular the difference of scale between the two geometries, which is highlighted in figure 3.16, and the uniformity of the resulting meshes, which is a product of the re-parametrisation steps detailed in section 3.1. The proposed methodology dealt well with both geometries, which highlights its overall robustness and genericness.

Furthermore, the function and the position of the two blades in the engine will allow us to explore different test cases, with different design criteria in each case, as detailed in section 3.3. Also, the potential for improvement using shape optimisation is not the same for both geometries. The HP compressor blade is an actual blade geometry, so its potential for improvement is not very high since its shape is already quite optimal. The fan blade on the other hand, is still at a development stage and offers more room for improvement using shape optimisation techniques.

3.2 Preliminary mechanical analyses

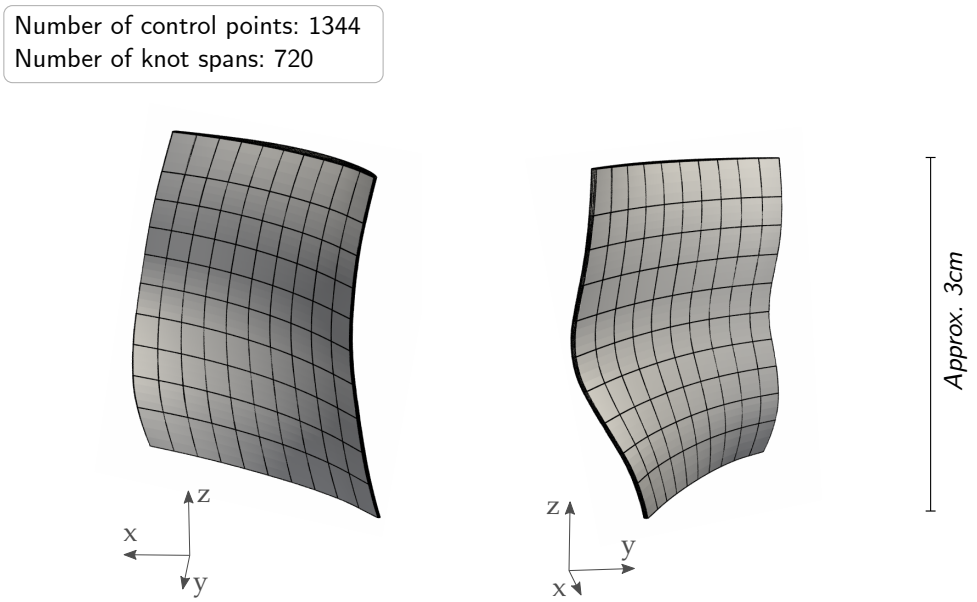
The final objective of this chapter is to run shape optimisation of aircraft engine blades. Prior to that achievement, we propose different sizing cases. The results stemming from these analyses will provide an initial status for several variables, and make possible the comparison with the optimisation results.

3.2.1 Blade under centrifugal force and aerodynamic pressure

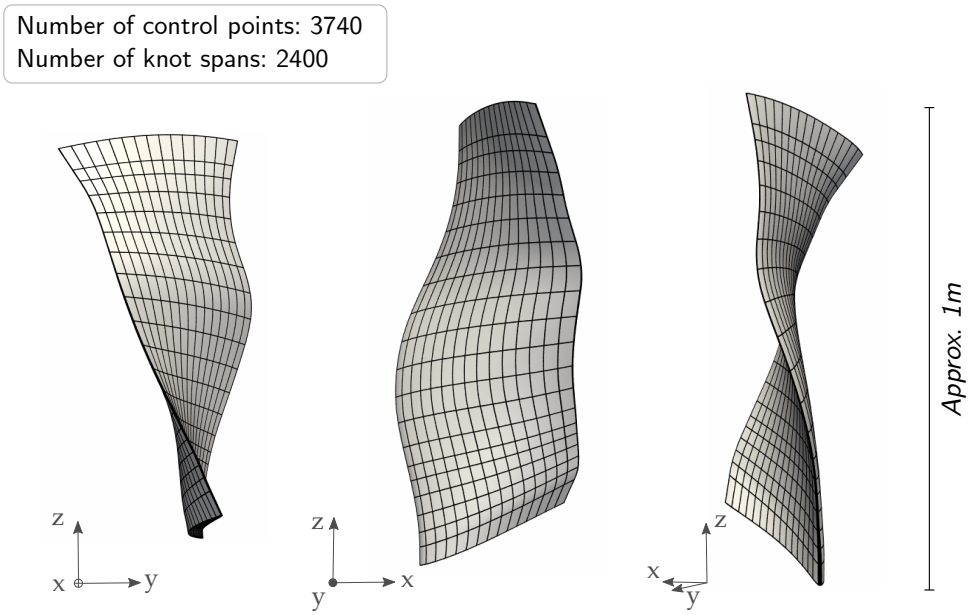
As classically done for the mechanical sizing of aircraft engine blades, we consider centrifugal force and an actual pressure field resulting from aerodynamic simulations, which is illustrated in figure 3.9. The centrifugal force $\mathbf{b}$ is defined by:

$$\mathbf{b} = \rho\,\omega^2\mathbf{r}, \tag{3.4}$$

with ρ the density, ω a given angular velocity, and $\mathbf{r}$ the vector in radial direction. Concerning the pressure field, a discretised pressure distribution is provided by Safran Aircraft Engines aerodynamics department. A continuous pressure distribution is obtained on the outer surface of the blade by approximating the discrete field, using the same parametrisation as the one generated for the blade geometry itself.



(a) HP compressor blade.



(b) Fan blade.

Figure 3.8 – Two trivariate B-Spline blade models with parametric lines.

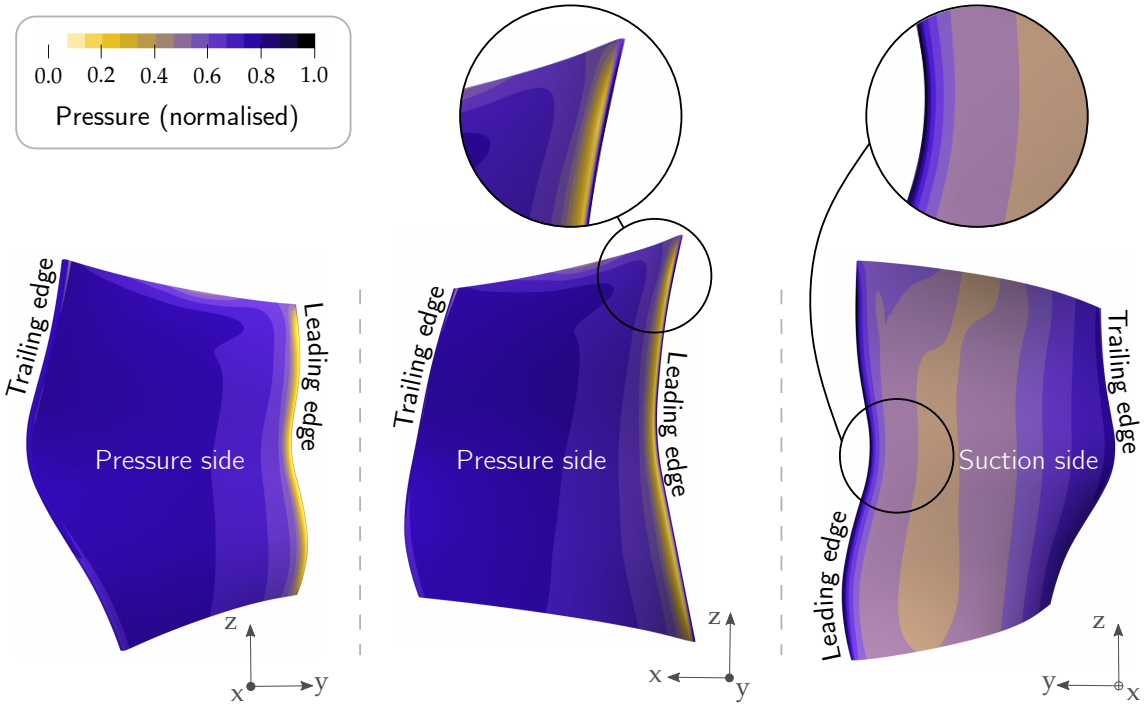


Figure 3.9 – Pressure field applied on the HP compressor blade outer surface.

Only the HP compressor blade is considered here and the analysis set-up is as follows. The bottom face ($z = 0$) of the blade is fixed. The engine rotation speed is set to a value around 16000 rpm, which corresponds to a typical cruise flight condition. The pressure field is the one illustrated in figure 3.9. The computation gives a compliance value equal to 12.62 m/N.

3.2.2 Modal analyses

We now turn to the study of the vibration behaviour of the blades. We first focus on the previously presented HP compressor blade. Its first vibration mode is around 2000 Hz, which can be considered as quite a high value. As mentioned in section 2.1.1, blades with high natural frequencies are more subject to enter into mechanical resonance for harmonic frequencies of the engine corresponding to the number of blades on the same stage and/or on neighbouring stages. In this example we chose an arbitrary harmonic as the excitation frequency, namely $30N$.

A usual way of visualising the frequency interactions of a blade/engine system is to draw an *interference diagram*. It features the natural frequencies of the blade as a function of the rotation speed of the engine. The speed for a particular operation regime is traced (resulting in a vertical line on the graph), as well as the studied harmonic[1]. The

[1]Here we explain the logic for one harmonic, but of course this scheme is applicable for multiple harmonics.

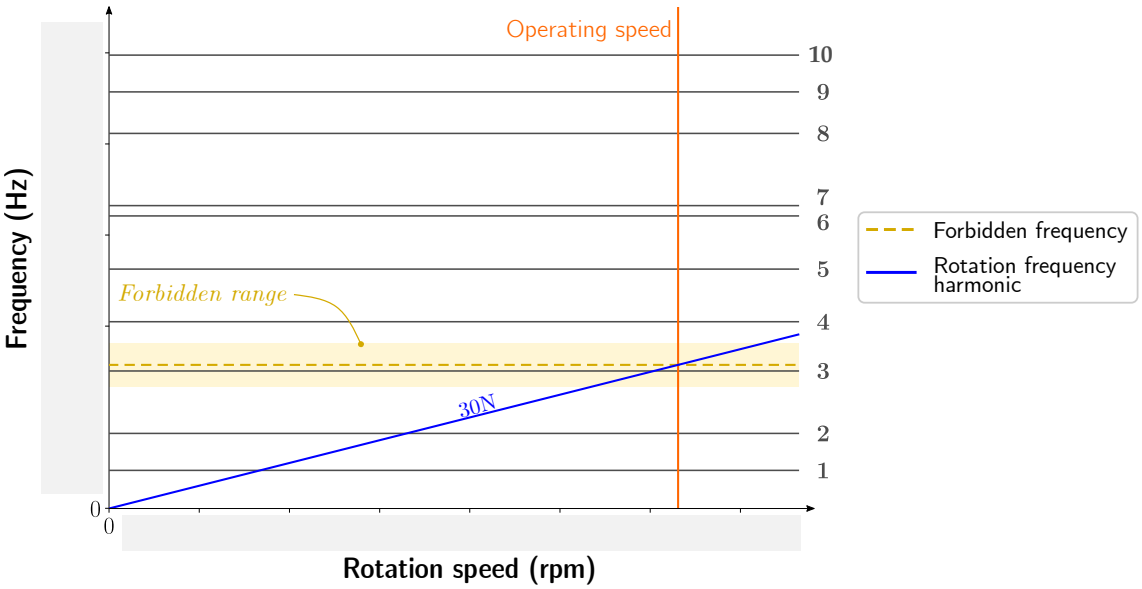


Figure 3.10 – Interference diagram of the high-pressure compressor blade. The indices on the right-hand part of the diagram correspond to the vibration modes indices.

frequency we seek to avoid is materialised by the intersection of the harmonic line and the operating speed line. A safety margin of 15% is commonly used in order to avoid interferences (Adam et al., 1998). The goal is then to prevent the blade from having any natural frequencies lying in that range.

The diagram for the HP compressor blade is presented in figure 3.10. The same engine operating speed is used as for the previous analysis. The shaded area corresponds to the 15% margin around the frequency standing for the $30N$ harmonic. Looking at figure 3.10, one can notice the third natural frequency of the blade is lying in the safety margin associated with this particular harmonic.

Let us now centre on the fan blade. It has a slender geometry, and its first natural frequency is around 40 Hz, which is rather low in comparison with the HP compressor blade. The excitation sources to consider for this blade are the ones corresponding to the first harmonics. The interference diagram is traced in figure 3.11. Here we consider the $1N$ harmonic. It is clear that the fifth natural frequency does not comply with the 15% margin requirement, while the fourth natural frequency is lying at the border of the range.

Remark In the presented results, the natural frequencies of the blade are computed on the deformed configuration of a centrifugal force computation. A general observation is that the natural frequencies values are unaffected by the value of the rotation speed. In practice, this is not the case since blade stiffening due to rotation has a great influence on the vibration behaviour of the system. However, these phenomena are not considered in the analysis set-up we implemented.

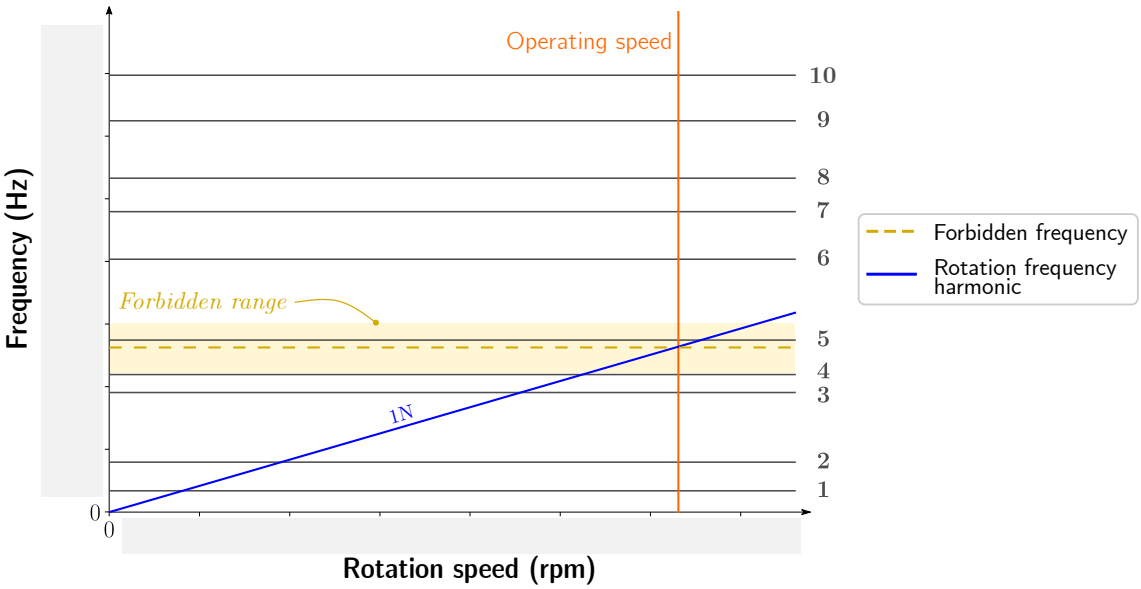


Figure 3.11 – Interference diagram of the compressor fan blade. The indices on the right-hand part of the diagram correspond to the vibration modes indices.

3.3 Shape optimisation of two compressor blades models

3.3.1 Parametric shape modification

In this work, two parametric shape modifications are set up. The first modifies the stacking law of the blade sections. It is used in the compliance minimisation case in section 3.3.2. The second modifies the thickness of the blade sections and is used for the target frequency and forbidden frequency problems of sections 3.3.3 and 3.3.4.

Remark Both shape modifications are based on parametric sections along the height of the blade. Here we consider sections that correspond to rows of control points after re-parametrisation. Therefore, there is no link between the considered sections and the initial ones (see section 3.1.2), that are based on stream surfaces.

3.3.1.1 Stacking law modification

This first parametric shape modification is set up to control the position and orientation of the sections along the height of the blade. It is used on the HP compressor blade in the compliance minimisation case in section 3.3.2.

For each section of the blade, a geometric centroid is computed in order to define the origin of a local coordinate system. Three types of motion are defined: two translations along the x and y axes, and one rotation along the z-axis. These are parametrised using variables a, b and θ respectively, as illustrated in figure 3.12.

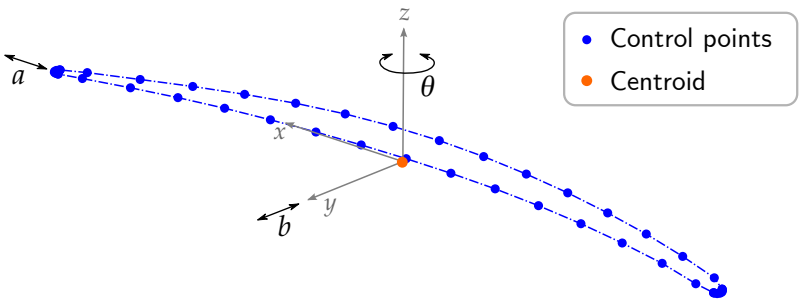


Figure 3.12 – Stacking axis shape parametrisation. The authorised motions are translations a and b, and rotation θ. Only the exterior control points of the section are represented for clarity.

Formulation Let us denote by $\mathbf{R}$ the rotation matrix of angle θ along the z axis,

$$\mathbf{R} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.5}$$

and by $\mathbf{T} = (a, b, 0)^T$ the translation vector. Given an initial control point $\mathbf{P} = (x, y, z)^T$, the application of the shape parametrisation yields a modified control point $\tilde{\mathbf{P}}$ which coordinates are computed as:

$$\tilde{\mathbf{P}} = \mathbf{R} \cdot \mathbf{P} + \mathbf{T}. \tag{3.6}$$

The derivatives of equation 3.6 with respect to the design variables a, b and θ can therefore be computed using the following formulas:

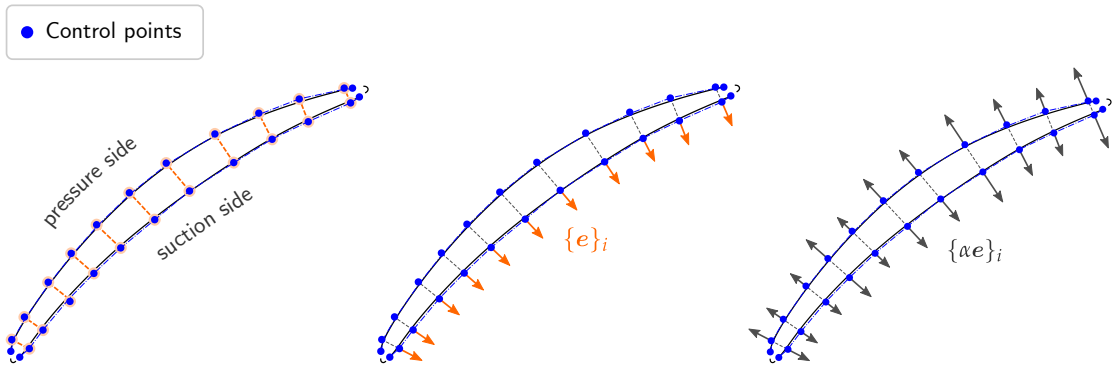
$$\begin{cases} \dfrac{\partial \tilde{\mathbf{P}}}{\partial a} &= (1, 0, 0)^T, \\ \dfrac{\partial \tilde{\mathbf{P}}}{\partial b} &= (0, 1, 0)^T, \\ \dfrac{\partial \tilde{\mathbf{P}}}{\partial \theta} &= \dfrac{\partial \mathbf{R}}{\partial \theta}\mathbf{P}, \end{cases} \tag{3.7}$$

with:

$$\frac{\partial \mathbf{R}}{\partial \theta} = \begin{pmatrix} -\sin\theta & \cos\theta & 0 \\ -\cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{3.8}$$

Looking at equation (3.7), one can note that the derivatives of the shape parametrisation with respect to the translation variables a and b are constant but that its derivatives with respect to θ is not. Therefore, it has to be recomputed for each sensitivity evaluation during the optimisation process.

Bound constraints Usually, when aircraft engines manufacturers choose to act on the stacking law of a given blade to perform shape optimisation, the aerodynamic behaviour



(a) Pairs of control points. **(b)** Direction vectors. **(c)** Control points motion.

Figure 3.13 – Thickness shape parametrisation.

of the blade is considered as well as its mechanical behaviour. Since no aerodynamic constraints are considered in this work, bound constraints are defined in order to avoid consequential shape modifications from an aerodynamics point of view. Hence, for each blade section, bounds are set on all three design variables: the rotation angle θ will be set between ± 2 degrees, while the translations a and b will be set between ± 1 mm.

3.3.1.2 Thickness modification

The second parametric shape modification is set up to control the thickness of the sections along the height of the blade. It is used for the target frequency and forbidden frequency problems of sections 3.3.3 and 3.3.4.

The principle of the parametrisation is illustrated in figure 3.13. The goal is to enable thickness variations by using directly the control points coordinates. In order to vary the thickness, the control points forming the suction and pressure sides of the section are paired, see figure 3.13a. Each pair forms a line from which a unit direction vector $\mathbf{e}$ can be defined as depicted in figure 3.13b. Then, a variable α attributed to each pair is used to control the magnitude of the translation of the pair of control points in opposite directions along this unit vector, see figure 3.13c.

Formulation Formally, the expression of this shape parametrisation can be written as:

$$\begin{cases} \tilde{\mathbf{P}}_1 &= \mathbf{P}_1 + \alpha \mathbf{e}, \\ \tilde{\mathbf{P}}_2 &= \mathbf{P}_2 - \alpha \mathbf{e}, \end{cases} \tag{3.9}$$

where $(\mathbf{P}_1, \mathbf{P}_2)$ is a pair of control points, $(\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2)$ is the same pair after transformation, $\mathbf{e}$ is the unit direction vector associated with the pair, and α is the magnitude of the translation (which is the same for both points in order to preserve the blade's skeleton).

The derivatives of (3.9) with respect to the design variable α can be computed using the following formula, with $k \in \{1, 2\}$:

$$\frac{\mathrm{d}\tilde{\mathbf{P}}_k}{\mathrm{d}\alpha} = \pm\mathbf{e}. \tag{3.10}$$

For each pair of control points, the derivative of the shape parametrisation with respect to the design variable α is the direction vector $\mathbf{e}$. Thus, it only needs to be computed once, at the beginning of the process.

We draw the reader's attention to the fact that some control points are left in their original position in order to ensure the geometric continuity between the boundary parametric surfaces of the blade (corresponding to the *contour surfaces* using our earlier denomination). The first derivatives at the endpoints of a B-Spline curve are affected by two control points: the first two and last two, respectively. Hence, the spatial location of these control points is not modified to ensure the junction of the outer surfaces.

Bound constraints Since the shape parametrisation modifies the thickness of the blade, the bound constraints applied on the α variables for each section and each pair of control points are an important aspect of the optimisation problem set-up. Indeed, it is necessary to avoid unfeasible geometries as much as possible, especially regarding crafting matters. Hence, the bounds on the design variables are defined such that the resulting thickness of the blade in any modified location can neither be greater than the initial highest value nor smaller than the initial lowest value. This bounds definition therefore ensures the non-penetration of boundary surfaces during the optimisation. The interior parametrisation of each section needs to be re-computed at every iteration as we modify the boundaries definition. It is computed by means of bilinear interpolation as in Step 6 of section 3.1.2.

3.3.2 Compliance minimisation

We first consider the minimisation of the compliance under a volume constraint for the HP compressor blade. Its volume V is allowed to vary by at most 10% of its initial value V_0, which is formalised as a single inequality constraint:

$$c(x) = (V(x) - 0.9\,V_0)(V(x) - 1.1\,V_0), \quad x \in \mathbb{X}. \tag{3.11}$$

The parametric stacking law modification described in section 3.3.1.1 is considered. The two bottommost sections are fixed and the remaining ten sections are allowed to vary, resulting in a total of $3 \times 10 = 30$ design variables for this case.

The simulation model is the same as in section 3.2.1. The displacement fields and stacking axis of the blade before and after the optimisation are illustrated in figures 3.14 and 3.15 respectively. The values of θ for each section are also shown on the figure for comparison.

One can observe the displacements have been significantly reduced by the optimisation. As a consequence, the compliance value has dropped by 40%. Even though the final shape is not realistic from an aerodynamic point of view, we believe that this example effectively demonstrates the feasibility of the approach.

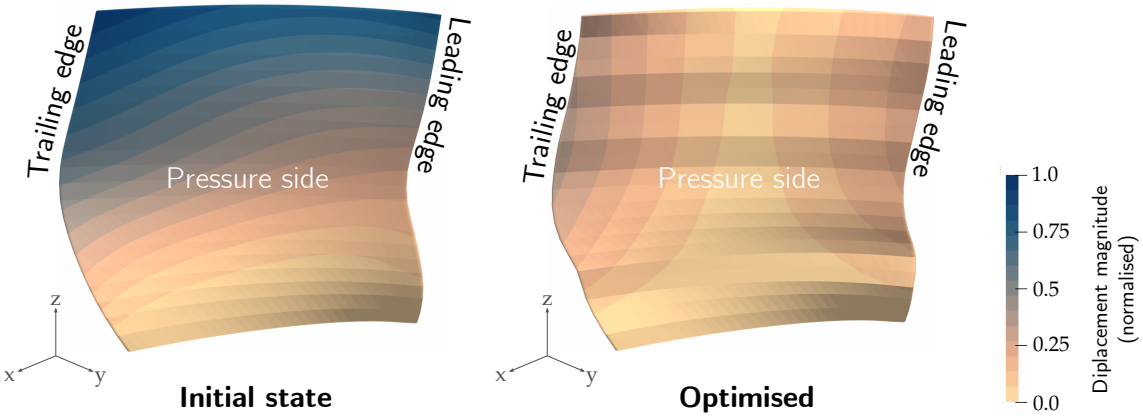


Figure 3.14 – Displacement fields for the initial (left) and optimised (right) geometries after compliance optimisation on the HP compressor blade.

3.3.3 Target frequencies

We now turn to the target frequencies problem. The preliminary analyses of section 3.2.2 revealed two problematic natural modes for each blade, that will be the starting points of our investigations. Table 3.1 sums up the initial and targeted frequency values for the HP compressor blade and for the fan blade. This approach can be related to the work of Manh et al. (2011), which tackles the issue of finding a shape where the first natural frequencies are prescribed. The major difference between the two approaches is the way the natural frequency values are prescribed. In the work cited above, the authors set the frequencies by adding as many constraints as prescribed frequencies in the optimisation problem. In our work we formulate the objective function of the optimisation problem so as to obtain the desired natural frequencies.

Optimisation set-up The following objective function is considered for this problem:

$$f(x) = \sum_{i=1}^{2} \left(\frac{\omega_i(x) - \omega_{i,\text{target}}}{\omega_{i,\text{target}}} \right)^2, \tag{3.12}$$

where the $(\omega_{i,\text{target}})_{i\in\{1,2\}}$ are target eigen pulsation values and the $(\omega_i(x))_{i\in\{1,2\}}$ are the eigen pulsation values of a blade modified according to x. Practically, this objective function favours precise target frequency values for the two selected modes. Besides, the same volume constraint (3.11) as in the compliance minimisation case is enforced. We emphasise that the analysis steps during the optimisation process does not take

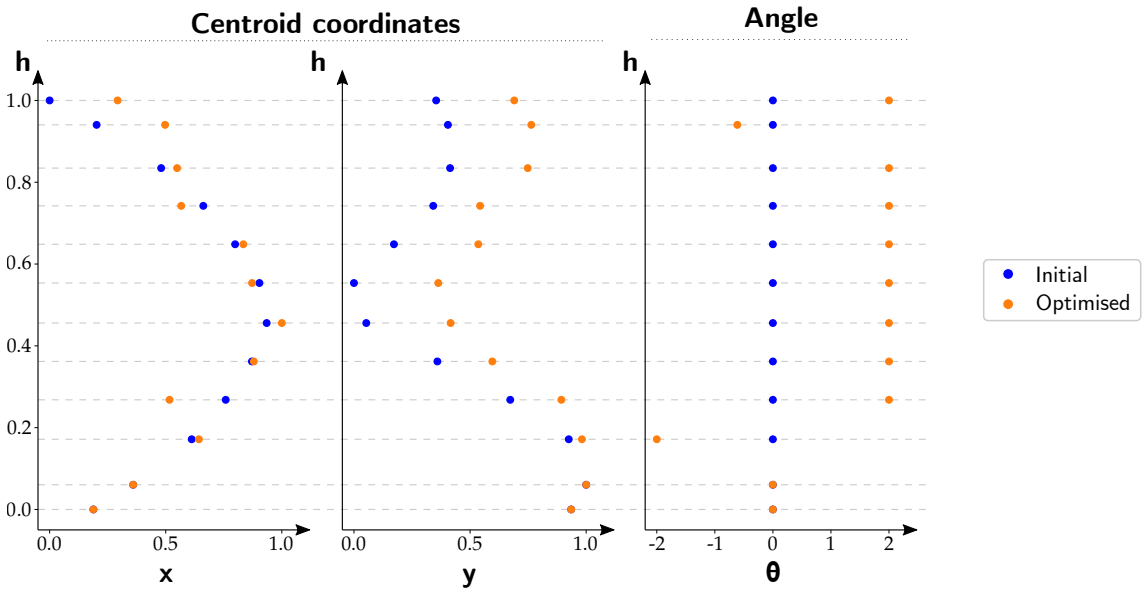


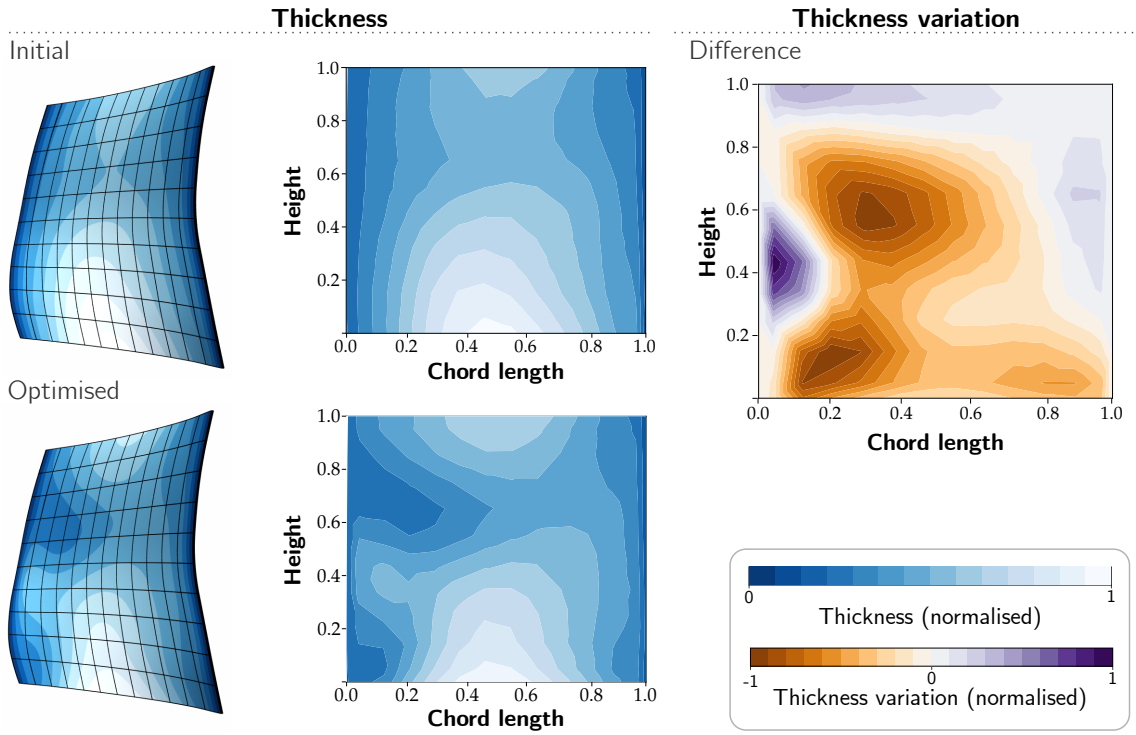
Figure 3.15 – Comparison of the initial and optimised stacking laws after compliance optimisation on the HP compressor blade. All values are plotted along the (normalised) height of the blade.

the rotational speed into account when computing the eigen frequency values. The parametric thickness modification described in section 3.3.1.2 is considered. The HP compressor blade is made of 12 sections, with 12 direction vectors each, which makes a total of 144 design variables. The fan blade on the other hand, is made of 22 sections with 15 direction vectors each, which makes a total of 330 design variables. The simulation model is the same as in section 3.2.2.

Results The final ratio values of the natural frequencies for the modes of interest are reported in table 3.1 and the thickness repartition of the initial and optimised geometries are illustrated in figures 3.16a and 3.16b, for the HP compressor blade and for the fan blade respectively. As expected, the final frequencies are very close to the target values. The optimised HP compressor blade tends to be thinner than the initial one, except for the middle and top sections, which are thicker. As a result, the volume of the blade has

HP compressor blade				**Fan blade**			
Mode no.	Initial	Target	Final	*Mode no.*	Initial	Target	Final
3	1.1862 f_A	f_A	1.0002 f_A	*4*	0.9954 f_C	f_C	1.0000 f_C
4	1.0756 f_B	f_B	1.0002 f_B	*5*	0.9016 f_D	f_D	1.0000 f_D

Table 3.1 – Initial, targeted and final frequencies for the modes of interest for the HP compressor blade and for the fan blade.



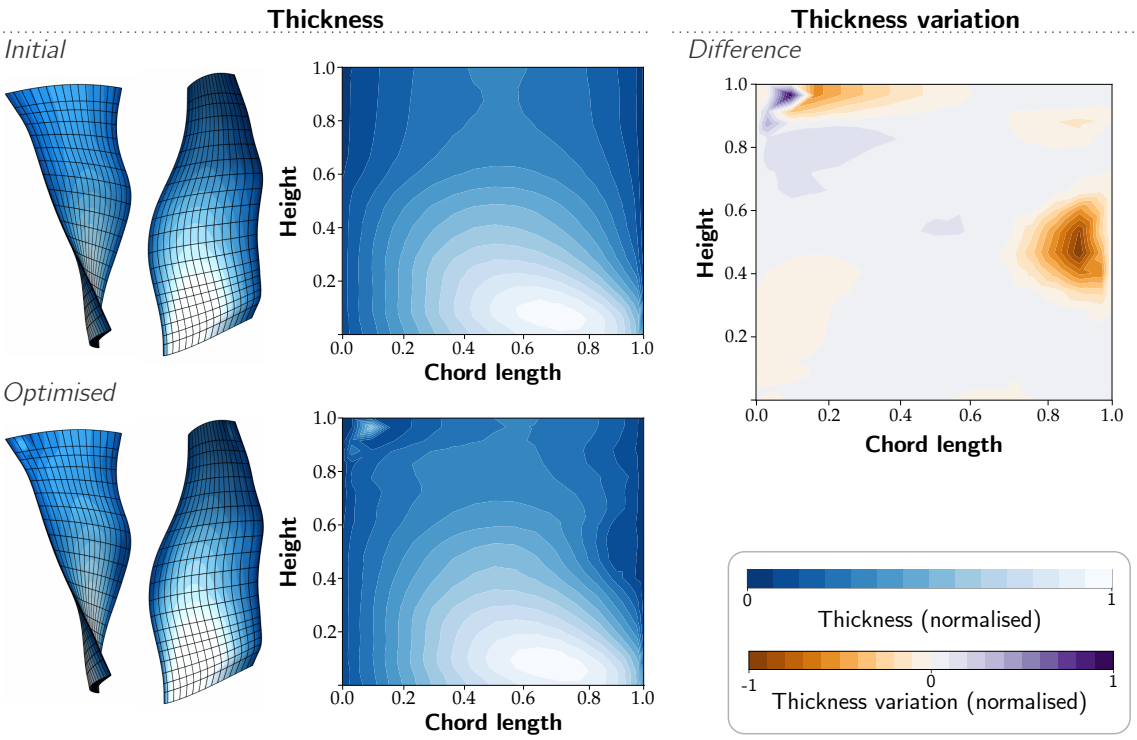
(a) Results for the HP compressor blade.

Figure 3.16 – Results for the target frequencies optimisation. Left: thickness contour plot, plotted onto the geometry and in a flattened space. Right: thickness variation between the two states, represented in a flattened space. The thickness values correspond to the one defined for the shape parametrisation.

dropped by 9%. As regards the fan blade, it is globally slightly thinner, with certain zones of greater thickness, especially on the top sections near the leading edge. Its final volume is 3% greater than the initial one.

3.3.4 Exclusion of frequency ranges

Satisfying results were obtained with the target frequencies formulation of section 3.3.3. However, using target frequencies is difficult to put into practice when a high number of natural frequencies have to be constrained, or when it is desirable to avoid multiple frequency ranges simultaneously. To address these limitations, we propose a second formulation with a different definition for the objective function — which can be seen as a penalty function. As the title of this section states, we seek to exclude certain frequency ranges to avoid resonance with the natural frequencies of a given blade. This example can be compared with works dealing with *frequency gaps*, especially when considering structural optimisation. Frequency gaps (or *frequency separation*) problems usually consist in maximising the separation of two adjacent natural frequencies for a given structure, as in Jensen and Pedersen (2006) or Olhoff et al. (2012). Works concerning



(b) Results for the fan blade.

Figure 3.16 – Results for the target frequencies optimisation *(continued)*. Left: thickness contour plot, plotted onto the geometry and in a flattened space. Right: thickness variation between the two states, represented in a flattened space. The thickness values correspond to the one defined for the shape parametrisation.

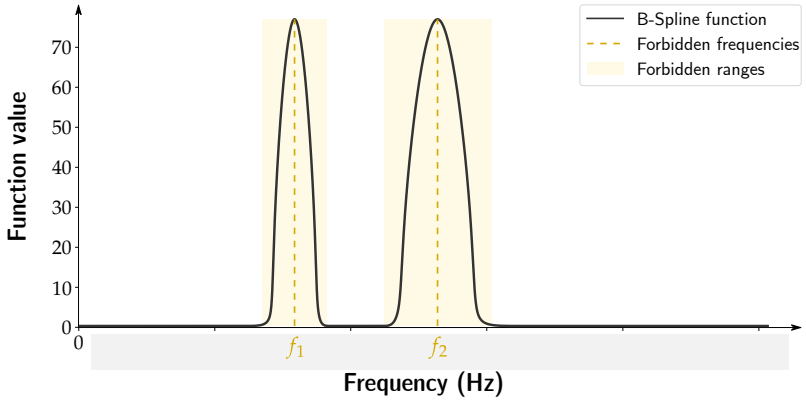


Figure 3.17 – Example of B-Spline penalty function. High function values are set near the forbidden frequencies f_1 and f_2, and low values are set elsewhere.

such problem often consider the maximisation of fundamental frequency as well — see Adali (1984), Duffy and Adali (1991), Adali and Verijenko (2001), Du and Olhoff (2007) for instance. Our approach is however slightly different from a design point of view, since we do not know which natural frequencies need to be separated prior to, and during, the optimisation. Another difference is that we do not seek to optimise the material repartition or its arrangement, but the shape of a whole three-dimensional structure.

Penalty function construction Our proposal is to use a penalty function in order to exclude certain frequency ranges. The idea is to build a function with high values in the vicinity of forbidden frequencies we consider, and low (or zero) values elsewhere. This penalty function is evaluated for each natural mode, and the contributions are summed. The objective function we seek to minimise during the optimisation process consists of this sum. In that fashion, the natural frequencies lying in ranges with high penalty values will be drawn away.

Visually, this penalty function we seek to build is a "bumped" function (see figure 3.17), where the bumps are the zones we do not want natural frequencies to lie in. We thought of using a B-Spline curve as a penalty function, as it offers a great flexibility in terms of penalty values. That way, it is possible to build a function that is adapted to each case without too much difficulty. Another favourable aspect is the continuity and differentiability of B-Spline functions, which allow to compute analytically the gradient values of the resulting penalty function — and hence, of the objective function itself. In the end, using a B-Spline definition is interesting in the analytic framework we set up[2].

Optimisation set-up To demonstrate the effectiveness of this objective function, we consider two forbidden frequency ranges for each blade: the $30N$ and $50N$ harmonics

[2]And it sounds somewhat pleasant to carry out shape optimisation of a B-Spline geometry by means of a B-Spline function.

are chosen for the HP compressor blade, and the $1N$ and $2N$ harmonics are chosen for the fan blade (see figures 3.10 and 3.11 for the initial interference diagrams). The objective constraint can be expressed as follows, given the penalty function $\mathcal{C}_{penalty}$ and a total of n_{freqs} studied natural frequencies:

$$f(x) = \sum_{i}^{n_{freqs}} \mathcal{C}_{penalty}\left(\frac{\omega_i(x)}{2\pi}\right), \tag{3.13}$$

where the $(\omega_i(x))_{i\in\{1,2,\dots,n_{freqs}\}}$ are the eigen pulsation values of a blade modified according to x. Practically, the value of the penalty function for a given eigen pulsation is evaluated by means of a binary search.

The rest of the optimisation set-up is the same as in section 3.3.3 for both blades: the thickness shape parametrisation is used and the volume constraint stated in equation (3.11) is enforced.

Results We first look into the results obtained for the HP compressor blade. The evolution of the values of the first ten natural frequencies during the optimisation is presented in figure 3.18 (top). The frequencies f_1^{HP} and f_2^{HP} are the two forbidden frequency values. In the initial state, two natural frequencies are lying in two different forbidden ranges (iteration 0). An optimum is reached after 11 iterations. No more natural frequency lies in the forbidden ranges, and the volume constraint is respected. The final volume value lies on the upper limit we set for the constraint, *i.e.* 10% greater than the initial one. Figure 3.18 (bottom) shows the evolution of the objective function and constraint values over the iterations. Thickness contour plots are presented in figure 3.19. It can be observed that the objective function reaches a minimum after 4 iterations, and that subsequent iterations are necessary to satisfy the volume constraint.

We now turn to the results for the fan blade. The optimisation histories for this case are presented in figure 3.20. The frequencies f_1^{Fan} and f_2^{Fan} are the two forbidden frequency values. In this case four natural frequencies lie in the forbidden frequency ranges at the beginning. An optimum is reached after 17 iterations, the constraint being respected throughout the whole optimisation procedure. The thickness contour plots are presented in figure 3.21. It can be observed that the magnitude of the thickness modifications is larger in this case than in the target frequencies case — this aspect can be observed by comparing the Thickness variation plots of both cases. The volume is however overall similar to the initial one, the increase being of 2%.

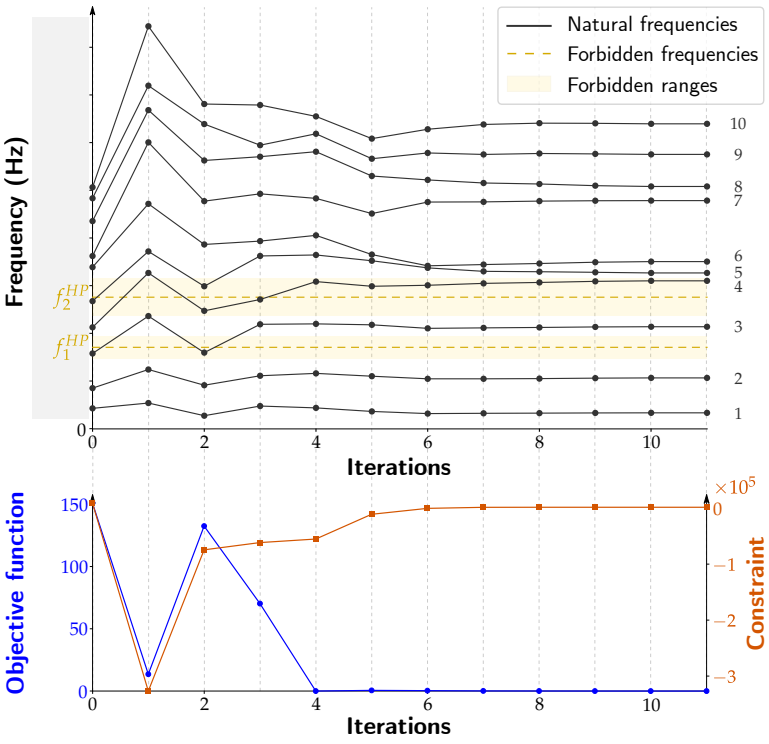


Figure 3.18 – Optimisation histories for the second vibration case on the HP compressor blade. Top: evolution of the first ten natural frequencies over the optimisation. Bottom: evolution of the objective function and constraint values over the optimisation.

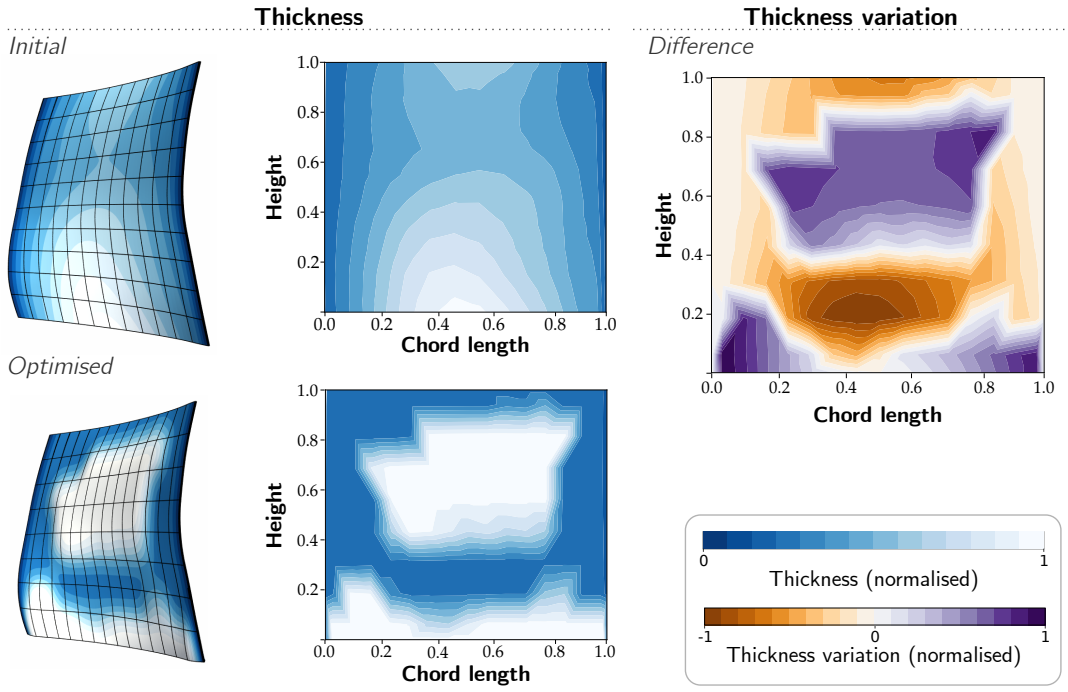


Figure 3.19 – Thickness variations for the second vibration case on the HP compressor blade. Left: thickness contour plot of the blade, plotted onto the geometry and in a flattened space. Right: thickness variation between the two states, represented in a flattened space.

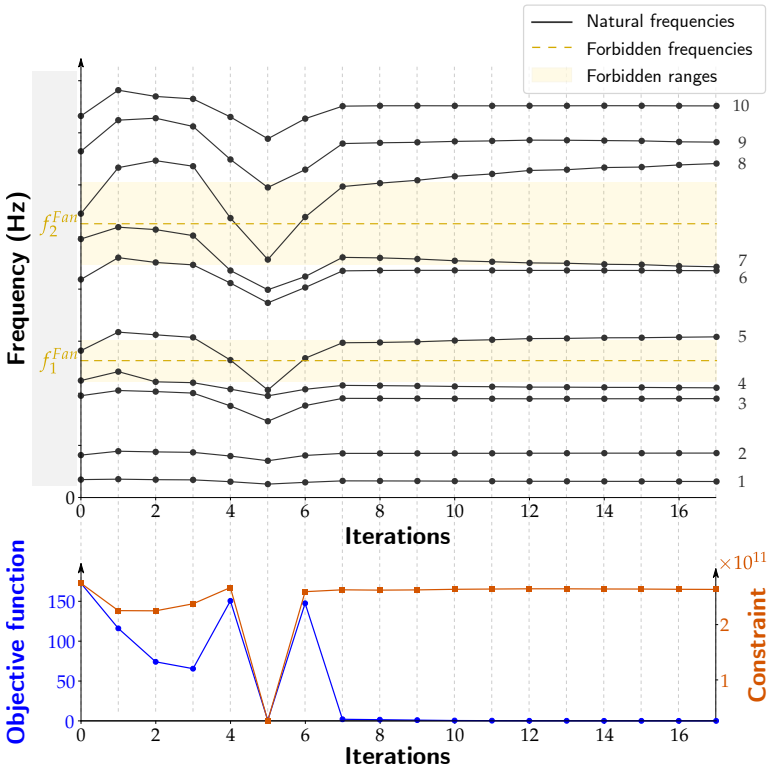


Figure 3.20 – Optimisation histories for the second vibration case on the fan blade. Top: evolution of the first ten natural frequencies over the optimisation. Bottom: evolution of the objective function and constraint values over the optimisation.

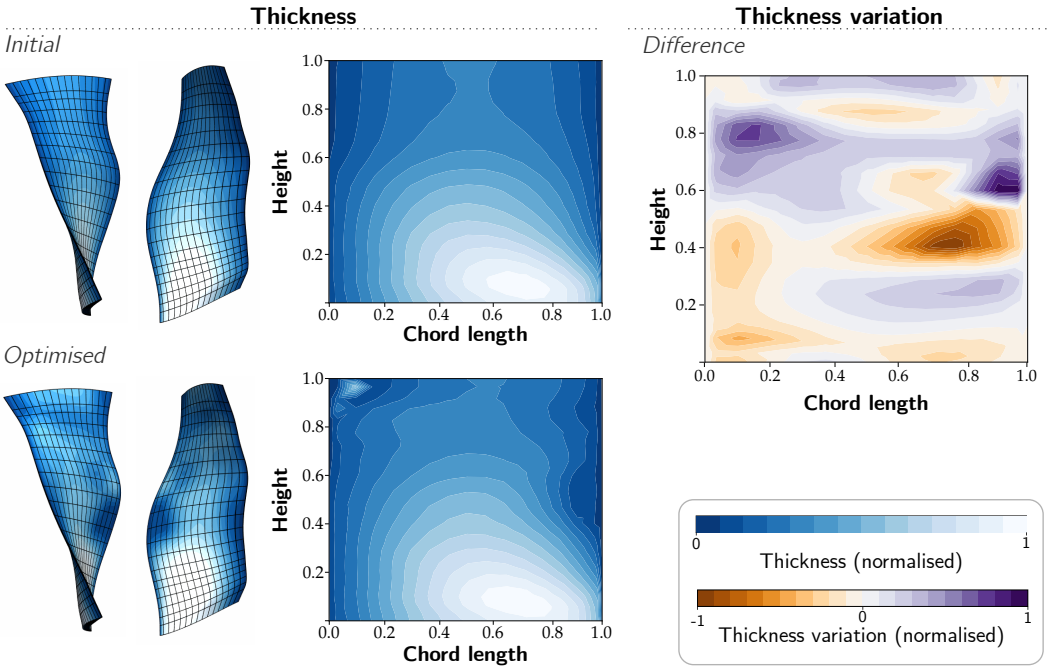


Figure 3.21 – Thickness variations for the second vibration case on the fan blade. Left: thickness contour plot of the blade, plotted onto the geometry and in a flattened space. Right: thickness variation between the two states, represented in a flattened space.

Concluding remarks

This chapter presents first steps towards a methodology for the parametric shape optimisation of aircraft engines axial compressor blades using isogeometric analysis. First, a procedure to generate analysis- and optimisation-suitable volumetric B-spline models starting from stacked blade profiles is proposed. Its effectiveness is demonstrated on a small and thick high-pressure compressor blade and on a slenderer and more twisted low-pressure compressor fan blade. It is our belief that the proposed methodology can generalise to other blade shapes, such as propeller blades or turbine blades.

Secondly, we investigate different optimisation problem formulations to solve two different sizing cases. The first case is based on a static analysis featuring a centrifugal force and an actual surface pressure field resulting from aerodynamic simulations. The objective is to reduce the overall blade displacements in that case. To this end, a differentiable parametric stacking law modification procedure is set-up and a compliance minimisation under volume and bounds constraints formulation is used. It is shown that a 40% compliance drop can be achieved in this case, effectively reducing the overall displacements.

The second case uses a modal analysis and is based on Campbell diagrams. The objective is to concurrently satisfy several frequency margins specifications. A differentiable parametric thickness modification procedure is set-up in this case and two objective function formulations are considered. The first formulation is a classical target attainment formulation. It is shown that target frequencies can indeed be reached if appropriately set. Satisfying results are obtained with this formulation but it has limitations. Indeed, it is not practical when a high number of natural frequencies have to be constrained, or when it is desirable to avoid multiple frequency ranges simultaneously. To circumvent these limitations, a forbidden frequencies ranges formulation with a B-spline based objective function definition is proposed. It is shown that this formulation effectively lifts the aforementioned limitations.

We limited ourselves to embedded blades in this chapter, which did not allow us to study structural constraints. Structural constraints have a major impact on blades life expectancy, and therefore on their sizing. A more involved model integrating both the blade and its platform would be required to accurately simulate the force paths passing through the blade root. This should be the object of the following chapter.

Chapter 4

Isogeometric analysis of complex solid structures

Complex geometric modelling has been a crucial concern since the very beginning of IsoGeometric Analysis: the tensor-product structure of NURBS geometry inevitably raises the question of multiple patches modelling and analysis. In this chapter we make use of several techniques in order to address these matters in an appropriate manner, considering the context of solid elements modelling. We first introduce the embedded approach we set up to address complex geometric modelling with solid elements. Coupling and domain decomposition aspects are tackled afterwards, as essential components of multiple patches modelling. The interest of using both techniques concurrently is subsequently illustrated at the end of this chapter.

Contents

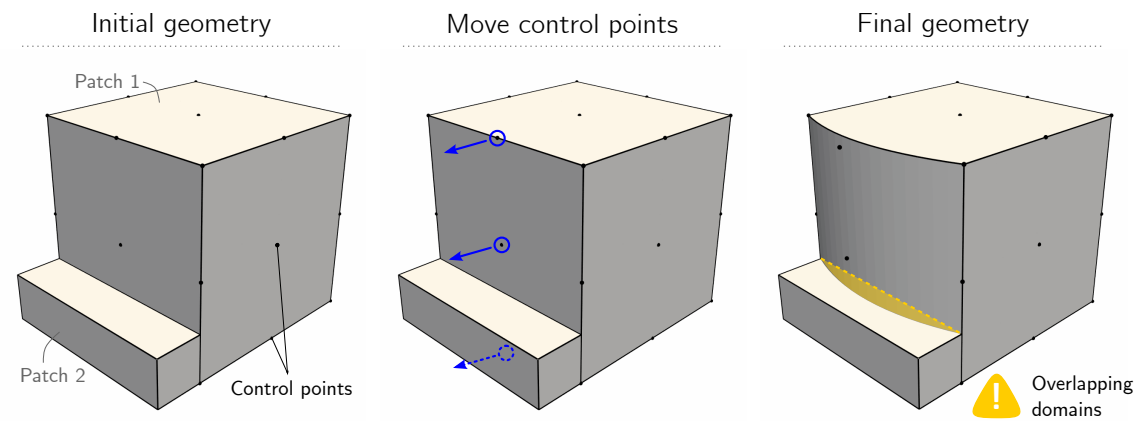


Figure 4.1 – Modifying the control points position of one patch within a two-patches model.

4.1 Parametrisation nesting: the embedded approach

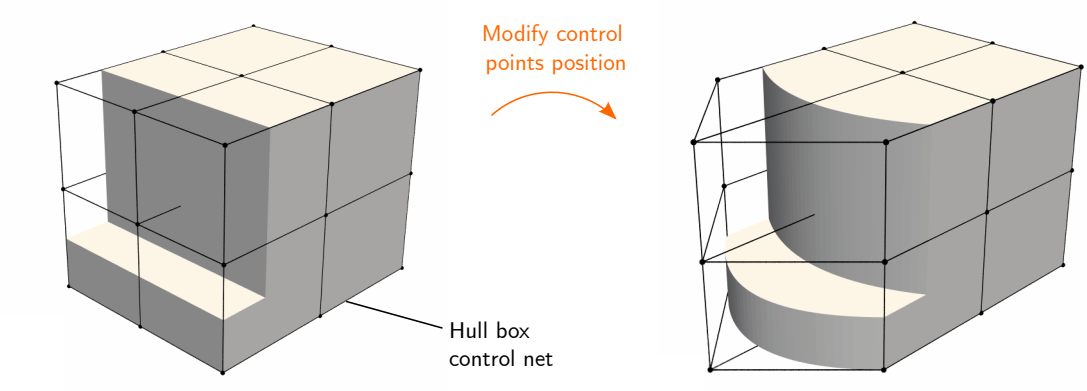
In this section we introduce the so-called *embedded approach* we set up in order to address various difficulties brought by the modelling (and analysis) of complex solid structures. We first outline the main ideas behind this technique, and illustrate its interest for our work. Then we explain with more details how to carry out mechanical analysis using this embedded formulation. We conclude this section by various numerical examples in order to assess the implementation of the proposed method.

4.1.1 Modifying the shape of a multi-patch model

4.1.1.1 A simple observation for multi-patch modelling

We start this section by bringing a simple observation to the reader's attention. For this purpose, we rely on figure 4.1. Let us consider the left part of the figure: our goal is to modify the shape of the main cube (Patch 1), by modifying the position of its control points as pictured in the centre of the figure. The result of this operation is the model pictured on the right-hand side: the shape of the cube is modified, but it now overlaps the domain of the smaller patch (Patch 2). This result is not desirable given our framework and purposes. The question we shall try to answer in this section therefore is: *how to impose shape modifications on the main object while having other patches to "follow" automatically?*

This question is valid for the simple cuboid example, as well as for more complex models. In particular, if we recall our final goal of running structural shape optimisation of a complete blade model, the question arises indirectly. Shape optimisation requires to update the shape of the model under consideration at every optimisation step. Precisely, in our case we look forward to optimise the shape of the blade patch, while having the fillet to follow its shape modifications throughout the optimisation process.



(a) Embedding the whole geometry.

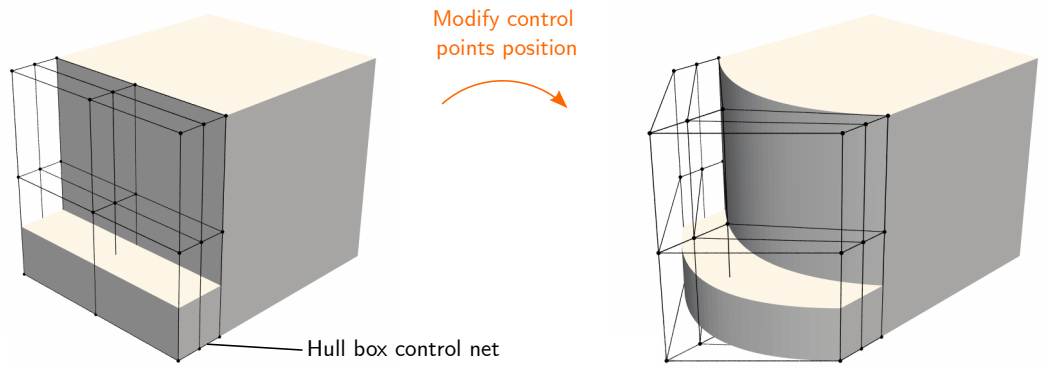
Figure 4.2 – The free-form deformation approach. A modification of the control points position of the hull box leads to the modification of the nested object(s).

4.1.1.2 Free Form Deformation

The approach we chose to set up is close to the well-known *Free Form Deformation* concept. This technique has first been used in computer graphics and geometric modelling (see Sederberg and Parry, 1986, Coquillart, 1989), and later used for shape parametrisation in shape optimisation (Duvigneau, 2006, Manzoni et al., 2012, Koshakji et al., 2013). The core idea behind FFD is to nest a given geometric representation (be it a curve, surface or volume) into a non-complex volumetric parametrisation. The host volume is later referred to as *hull box* or *hull object*. The point of nesting a complex geometric description into a simpler one is straightforward for one who seeks to apply shape modifications. The simple parametrisation of the host volume makes it possible to modify the global shape without dealing with too many control points. This aspect is illustrated in figure 4.2a: the whole model is embedded into a simple cube. Modifying the position of the control points of the hull box results in shape modifications of both embedded patches. In addition to the attractive aspect of dealing with fewer control points for shape modification, we emphasise that geometric coincidence of the embedded patches is automatically guaranteed.

Now turning back to our objective, we use the concept of parametrisation nesting in a similar way as done in works from Bauer et al. (2017) and Hirschler et al. (2019c). They respectively embed curves into surfaces, and surfaces into volumes. We propose to take this idea further by embedding volumes into volumes, *i.e.*, embedding trivariate entities into trivariate hulls.

This approach is particularly adapted to NURBS-based modelling, as the notion of parametric space is already present. The way we put this approach into practice is summarised in figure 4.2b. Only the Patch 2 is embedded into a volumetric hull object. The latter is parametrised in a way that ensures geometric compatibility with the cube patch. Then, when the Patch 1 is modified, the hull object is modified accordingly.



(b) Embedding only one patch. The hull box parametrisation is based on the cube.

Figure 4.2 – The free-form deformation approach (*continued*). A modification of the control points position of the hull box leads to the modification of the nested object(s).

It results in the modification of the shape of the Patch 2, which still is geometrically compatible with the Patch 1. This example gives an insight of the final set-up we shall use for the complete blade model, picturing the Patch 1 as the blade and the Patch 2 as a fillet.

4.1.2 Implementation aspects for the embedded solid element

4.1.2.1 NURBS composition

The core idea of nested parametrisation is to map the geometry (and other local quantities) of each patch into the parameter space of the hull object. Thus, compared to classical NURBS geometry, the nested parametrisation will present an additional mapping step. It results in the composition of the NURBS basis functions from the embedded patch and hull object, respectively.

Let us sum up the various spaces involved in the composition:

1. The embedded object parametric space (parametrised by variables θ_1, θ_2, θ_3);
2. The hull object parametric space (parametrised by variables ξ_1, ξ_2, ξ_3), which can also be seen as the embedded object physical space,
3. The hull object physical space (parametrised by variables x, y, z), which is where the composition comes into play.

The three spaces are pictured in figure 4.3 hereafter for better visualisation. The mathematical formulations of the objects are as follows. The hull object $\mathbf{V}_H$ is defined as a NURBS volume:

$$\mathbf{V}_H(\xi_1, \xi_2, \xi_3) = \sum_{ijk} N_{ijk}(\xi_1, \xi_2, \xi_3)\, \mathbf{Q}_{ijk}, \tag{4.1}$$

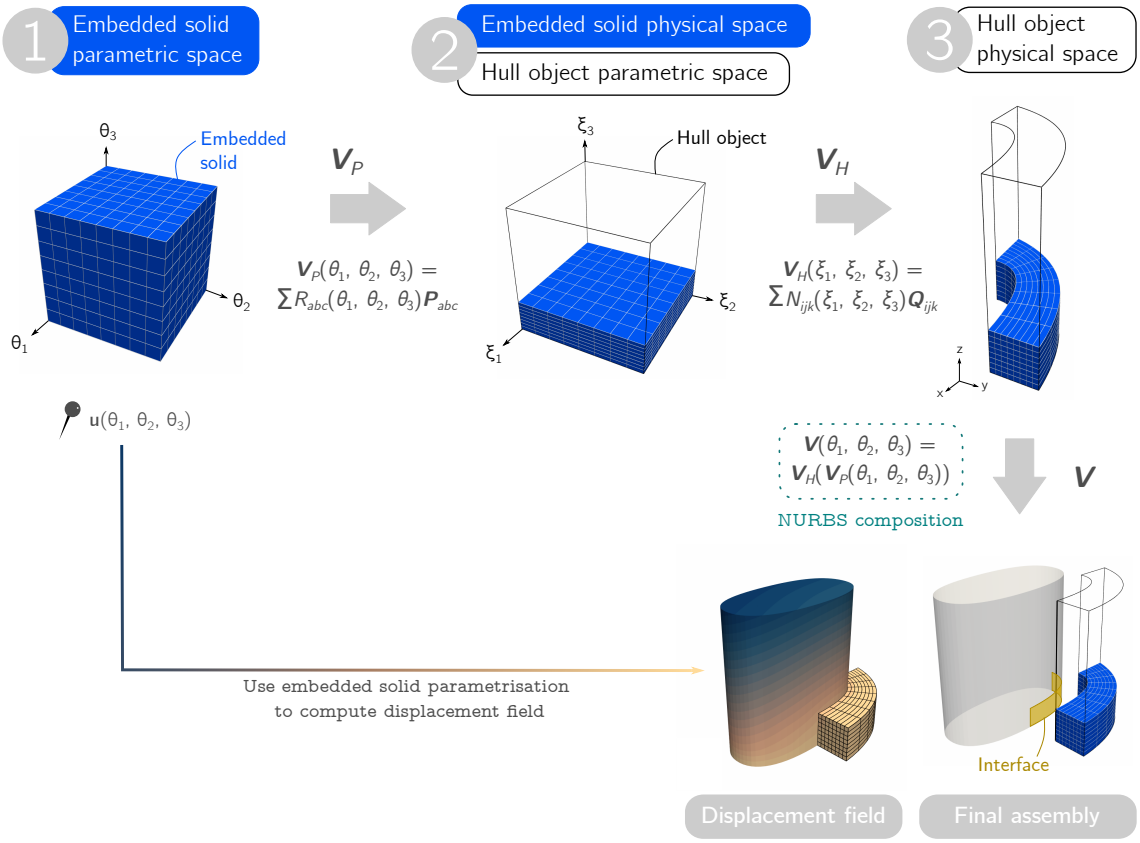


Figure 4.3 – Overview of the embedded solid element.

where the N_{ijk} are the trivariate NURBS basis functions and $\mathbf{Q}_{ijk}$ are the associated control points. The embedded solid $\mathbf{V}_P$ is defined as:

$$\mathbf{V}_P(\theta_1, \theta_2, \theta_3) = \sum_{abc} R_{abc}(\theta_1, \theta_2, \theta_3) \mathbf{P}_{abc}, \tag{4.2}$$

where the R_{abc} are the trivariate NURBS basis functions and $\mathbf{P}_{abc}$ are the associated control points. Finally, we can define the final composition of these two mappings as the following volume $\mathbf{V}$:

$$\mathbf{V}(\theta_1, \theta_2, \theta_3) = \mathbf{V}_H(\mathbf{V}_P(\theta_1, \theta_2, \theta_3)) \tag{4.3}$$

$$\mathbf{V}(\theta_1, \theta_2, \theta_3) = \sum_{ijk} N_{ijk} \left(\sum_{abc} R_{abc}(\theta_1, \theta_2, \theta_3) \mathbf{P}_{abc} \right) \mathbf{Q}_{ijk}. \tag{4.4}$$

From a geometric point of view, using the nested parametrisation enables a simple and direct link between the main patch and the surrounding patches. As illustrated in the simple example of figure 4.2b, we base the parametric description of the hull object onto the main patch. Modifying the geometry of the main patch induces the modification of the hull object, thus automatically guaranteeing the compatibility of the

embedded object. To extend the extensively used metaphor of the *master-slave* paradigm, we could qualify this approach as a *leader-minion* model.

4.1.2.2 Analysis with embedded solid elements

Let us now give some elements about the mechanical analysis in this framework. Having an additional mapping with respect to the classical IGA framework does not lead to any approximation for the geometric description. In that way, the advantages of IGA are not marred by this supplementary layer.

To illustrate this statement, we outline the essential process of computing the shape functions (and derivatives) for the construction of element stiffness matrices for the case of the embedded solid element. We consider in the following we are at the elementary level, looping on integration points. We rely on figure 4.4 for the various spaces that come into play.

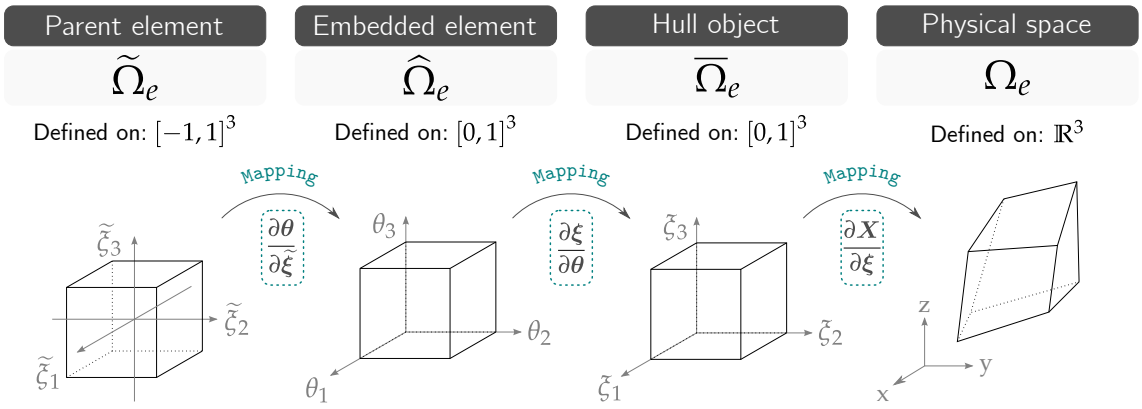


Figure 4.4 – Naming of the various spaces involved in the embedded solid element formulation.

The final objective is to compute two quantities:

- The determinant of the Jacobian matrix for the transformation of integrals from the parent element coordinates system (*i.e.*, in $\widetilde{\Omega}_e$) to the global coordinates system (*i.e.*, in Ω_e) ;
- The derivative of the shape functions used for the displacement approximation with respect to the global coordinates system (spanning in Ω_e, see figure 4.4).

Jacobian matrix In addition to the notations presented in figure 4.4, we shall write the knot vector parameters involved in the current element as $U_i^{(\alpha)}$ and $U_i^{(\alpha+1)}$ with $i = \{1, 2, 3\}$, respectively. The coordinates of the parent element for numerical integration are written as $\widetilde{\xi}_i$ with $i = \{1, 2, 3\}$.

Computing the Jacobian matrix in the case of the embedded solid element

▷ *Coordinates of the embedded solid element in* $\widehat{\Omega}_e$

$$\theta_i = \frac{1}{2}\left(\left(U_i^{(\alpha+1)} - U_i^{(\alpha)}\right)\cdot \widetilde{\xi}_i + U_i^{(\alpha+1)} + U_i^{(\alpha)}\right)$$

▷ *NURBS basis functions and derivatives for the embedded solid element in* $\widehat{\Omega}_e$

$$R,\ \frac{\partial R}{\partial \theta_i}$$

▷ *Coordinates of the hull object in* $\widehat{\Omega}_e$

$$\xi_i = \sum_{abc} R_{abc}(\theta_i) P_{abc}$$

▷ *NURBS basis functions and derivatives for the hull object in* $\overline{\Omega}_e$

$$N,\ \frac{\partial N}{\partial \xi_i}$$

▷ *Gradient of the mapping from* $\widetilde{\Omega}_e$ *to* $\widehat{\Omega}_e$

$$\frac{\partial \theta_i}{\partial \widetilde{\xi}_j} = \frac{1}{2}\left(U_i^{(\alpha+1)} - U_i^{(\alpha)}\right)$$

▷ *Gradient of the mapping from* $\widehat{\Omega}_e$ *to* $\overline{\Omega}_e$

$$\frac{\partial \xi_i}{\partial \theta_j} = \sum_{abc} \frac{\partial R_{abc}}{\partial \theta_j}\, \xi_i$$

▷ *Gradient of the mapping from* $\overline{\Omega}_e$ *to* Ω_e

$$\frac{\partial X_i}{\partial \xi_j} = \sum_{pqr} \frac{\partial N_{pqr}}{\partial \xi_j}\, X_i$$

▷ *Jacobian*

$$\boldsymbol{J} = \frac{\partial \boldsymbol{X}}{\partial \widetilde{\boldsymbol{\xi}}} = \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{\theta}}{\partial \widetilde{\boldsymbol{\xi}}}$$

Approximation space for displacement In order to assemble the elementary stiffness matrix, the derivative of the shape functions used for the displacement approximation with respect to the global coordinates system are needed. Before getting to compute them, the approximation space for the displacement field has to be chosen. Since NURBS composition is used, the isoparametric concept may not be used for the embedded solid

element. We must specify which solution space is used for the discretisation of the displacement field. Two choices present themselves:

1. Use the discretisation of the hull object:

$$u^h(\theta_1, \theta_2, \theta_3) = \sum_{ijk} N_{ijk}\left(\mathbf{V}_P\left(\theta_1,\, \theta_2,\, \theta_3\right)\right) \mathbf{u}_{ijk}, \tag{4.5}$$

2. Use the discretisation of the embedded solid:

$$u^h(\theta_1, \theta_2, \theta_3) = \sum_{abc} R_{abc}\left(\theta_1,\, \theta_2,\, \theta_3\right) \mathbf{u}_{abc}. \tag{4.6}$$

On the one hand, using the discretisation of the hull object for the discretisation of the displacement field could lead to ill-conditioned systems in case of a high proportion of void. This drawback is well-known issue in immersed and finite-cell methods for example, and techniques exist in order to circumvent it (Rank et al., 2012, De Prenter et al., 2019). On the other hand, the second scenario does not need such techniques. What is more, in that case the hull object parametrisation only serves as an operator for geometric modifications. This approach seems more natural for the targeted application, as we precisely seek to use the hull object as a shape modification tool. Another favourable aspect is that this approach is more straightforward to implement during the assembly of the stiffness matrix, as showed hereinafter.

Computing the derivative of the shape functions used for the displacement approximation in the case of the embedded solid element

▷ *Gradient of the mapping from Ω to $\widehat{\Omega}_e$*

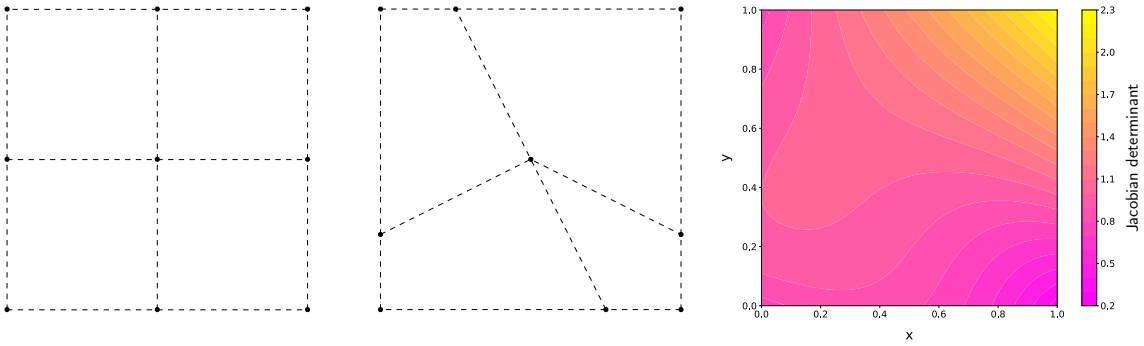
$$\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{X}} = \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)^{-1}$$

▷ *Basis functions composition*

$$\frac{\partial R}{\partial X_i} = \frac{\partial R}{\partial \theta_j} \frac{\partial \theta_j}{\partial X_i}$$

4.1.3 Numerical testing

We assess the implementation of the embedded solid element onto some reference numerical examples. In particular, we investigate the effect of the internal parametrisation of both the hull object and embedded elements on the quality of the result. The effect of the distortion of the control and physical meshes on the quality of the solution has been investigated in work from Lipton et al. (2010) for instance. In the following, we likewise study the effect of perturbing the inner control points repartition. Figure 4.5 illustrates the two-dimensional case of a deteriorated parametrisation: initial and distorted control



(a) Initial control mesh. **(b)** Distorted control mesh.. **(c)** Jacobian determinant values.

Figure 4.5 – Distorted parametrisation of a square quadratic planar surface.

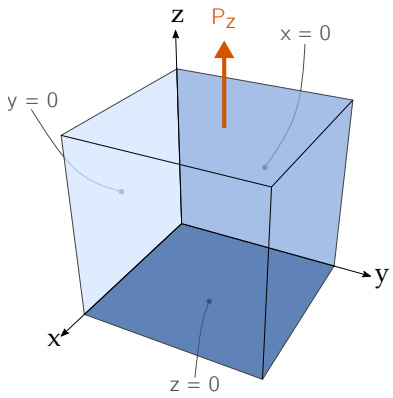


Figure 4.6 – Boundary conditions and loading for the patch test.

meshes are plotted, as well as the resulting Jacobian determinant for the deteriorated case.

We make use of a well-known reference example in order to assess our implementation. Namely, the *patch test* (see Irons and Razzaque, 1972) is a classical test used as an indicator for FE software. It assesses the solution on a simple test case, for which there is a uniform strain solution. The patch test boundary conditions and loading are presented in figure 4.6. We examine various configurations, based on this reference example.

The tackled examples are set up as described in the following:

1. Conventional patch test. The hull object consists in a unit cube. The embedded patch fills all of the parametric space of the hull.

2. Effect of a high quantity of void. The hull object consists in a cube which is 4 units long on each side. The embedded patch fills an eighth of the parametric space of the hull (thus resulting in a unit cube after composition).

3. Distorted inner parametrisation.

   (a) *Distorted hull.* The hull object is a unit cube with a distorted inner parametrisation. The embedded patch fills all of the parametric space of the hull and has a regular parametrisation.

(b) Distorted embedded patch. The hull object consists in a unit cube. The embedded patch fills all of the parametric space of the hull and has a distorted inner parametrisation.

The results obtained in the different cases are pictured in figure 4.7. They are in accordance with the reference solution. These examples confirm the implementation of the embedded solid element. The second example in particular illustrates the interest of choosing the discretisation of the embedded element for the displacement approximation: a high quantity of void does not affect the final result.

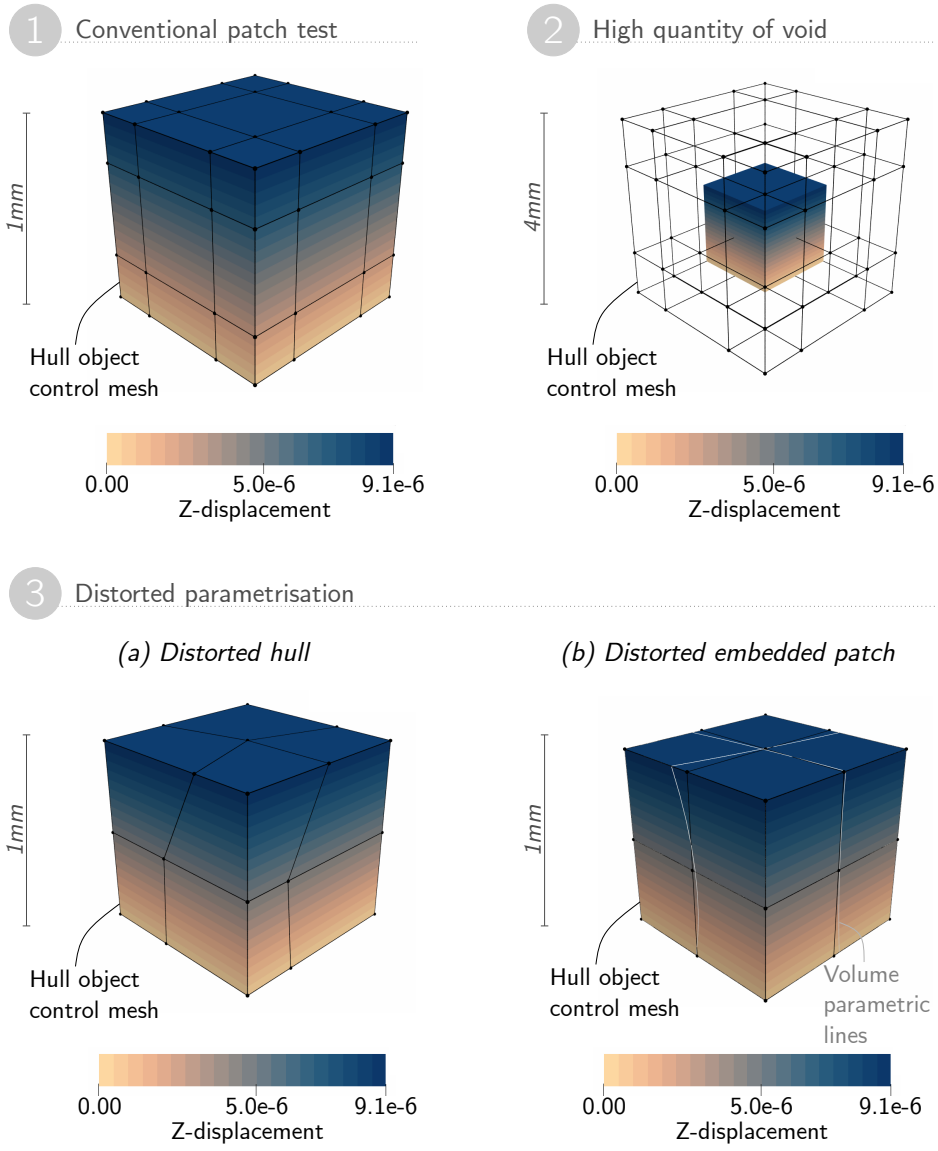


Figure 4.7 – Results of the numerical testing of the embedded solid element.

4.2 Isogeometric multi-patch analysis

In this section we first address the issue of modelling NURBS-based complex geometries. The tensor product structure of the parameter space of a NURBS patch restricts the shape potential to cuboid homomorphisms. In order to represent more complex geometries, multiple patches are required. In that context, several strategies arise in order to run analysis of multiple patches structures. We expose the technique we chose and assess its implementation on several numerical cases, from simple reference examples to more complex geometries. Throughout this section we shall rely on the geometry presented in figure 4.8 to illustrate various notions.

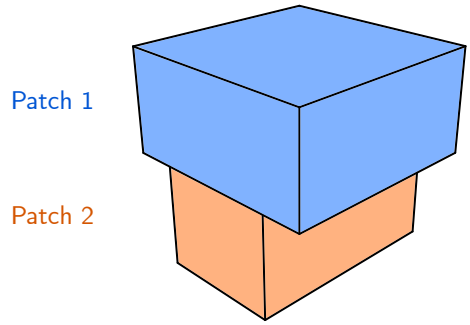


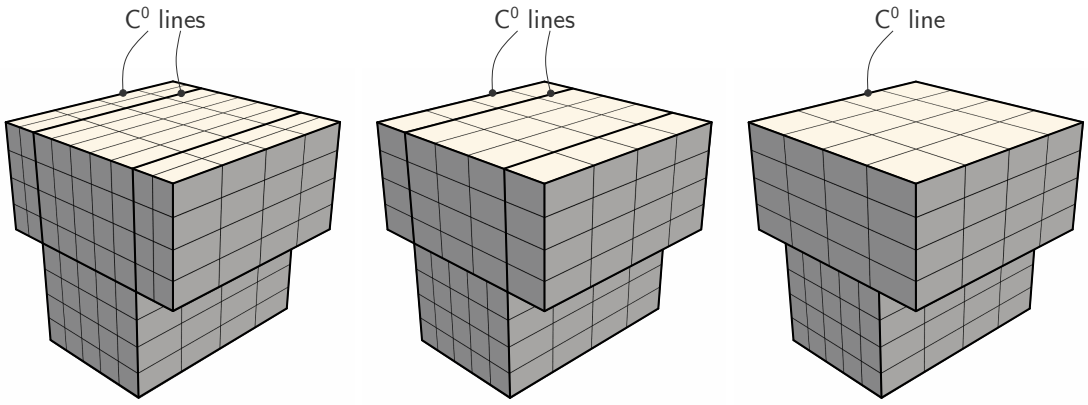
Figure 4.8 – Example geometry for the study of two sub-domains.

4.2.1 Geometric modelling with untrimmed NURBS

4.2.1.1 Tensor product structure and complex geometries

Thus far we exclusively considered geometries consisting of a single patch. Given the tensor product structure of a NURBS patch, the parametric space necessarily results in a structured rectangular mesh. A single patch is therefore suitable for representing topological cuboid shapes only. In order to model real-world structures, the need of several patches is essential.

Besides geometric consideration, we shall bear in mind that the core idea behind IGA is to integrate geometric modelling and analysis. Considering several patches is not trivial as the coupling between two (or more) patches must be ensured. The most straightforward way to achieve such a goal is to consider all patches as parametrically identical in the regions of interface. Then, the repeated control points are dealt with as a single one for the analysis. However, geometric modelling entails the description of patches in an independent manner. That is, the final parametrisation of the patches does not necessarily (and most of the time *does not*) lead to conforming parametrisation in the regions of interface between patches. This is why we review various types of interfaces one might have to deal with when considering multiple domains.



(a) Matching interface. **(b)** Non-matching interface. **(c)** Non-conforming interface.

Figure 4.9 – Types of interfaces.

4.2.1.2 Different levels of interface compatibility

Coupling several sub-domains is nowadays an established issue in numerical analysis. Classically, the following terms are used to describe the various configurations of interfaces one can encounter when modelling geometries consisting of several patches (or domains).

- Interfaces are said to be *matching* when the interface is defined between two C^0 lines on both sub-domains, and the parametric lines of the latter are perfectly aligned in the region of interface (see figure 4.9a).

- A *non-matching* interface would be characterised by the presence of C^0 lines on either side of the interface, without the parametric lines to be perfectly connected on the interior of it (see figure 4.9b).

- Finally, we face the case of a *non-conforming* interface when the edges of the interface are not aligned with elements boundaries on either side, while having no coincidence of the mesh of the two sub-domains in the region of interface (see figure 4.9c).

As mentioned above, geometric modelling does not consider the analysis aspect. As a result, the domains that define the final geometry usually share non-conforming interfaces. Note that in the case of volumetric entities (*i.e.*, trivariate models), the non-conforming case 4.9c easily happens as interfaces can be multi-dimensional. In the following we set out various techniques in order to run analysis of any geometric model consisting of multiple patches.

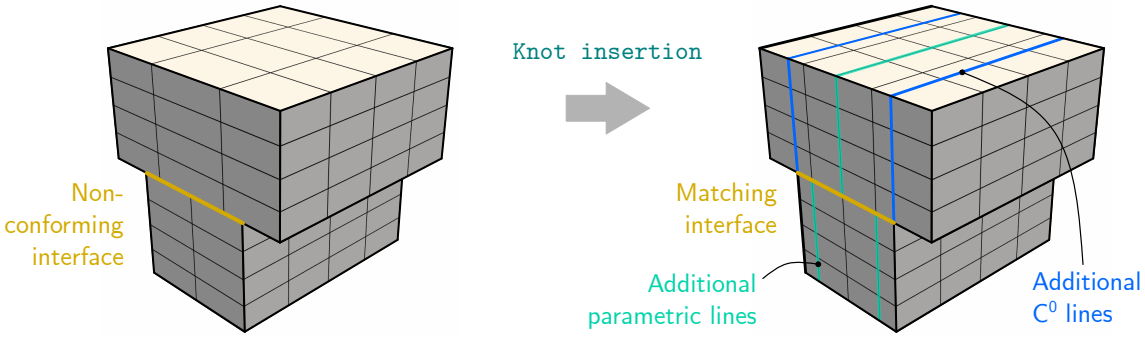


Figure 4.10 – Connecting simple non-conforming patches by using multiple knot insertion.

4.2.2 Modelling strategies and coupling methods

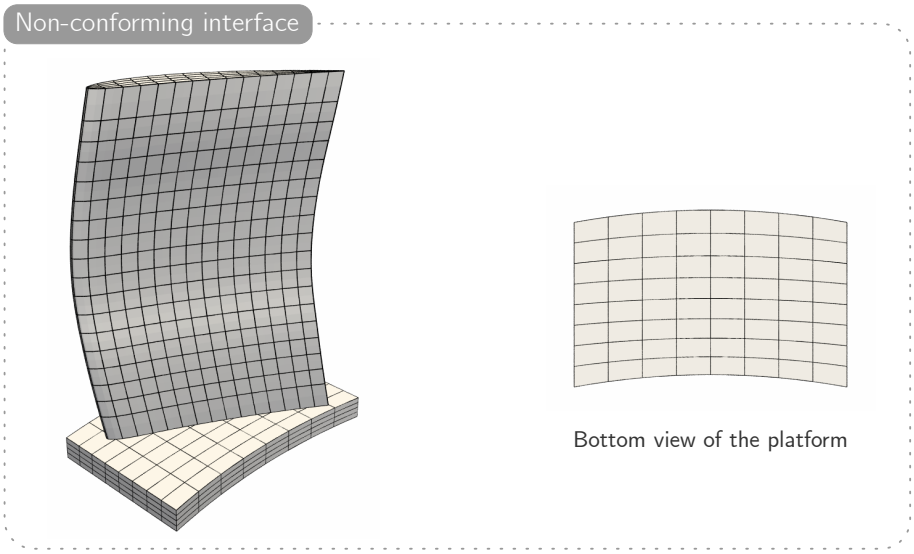
In order to be able to use the geometric model as a support for analysis — or, to use the established term, to make the geometric model *analysis-suitable* —, several techniques have been developed. Two main strategies arise: the first aims at making the interfaces to match (*via* re-parametrisation techniques for instance), while the second consists in developing coupling methods in order to connect non-conforming sub-domains.

4.2.2.1 Analysis-suitable model

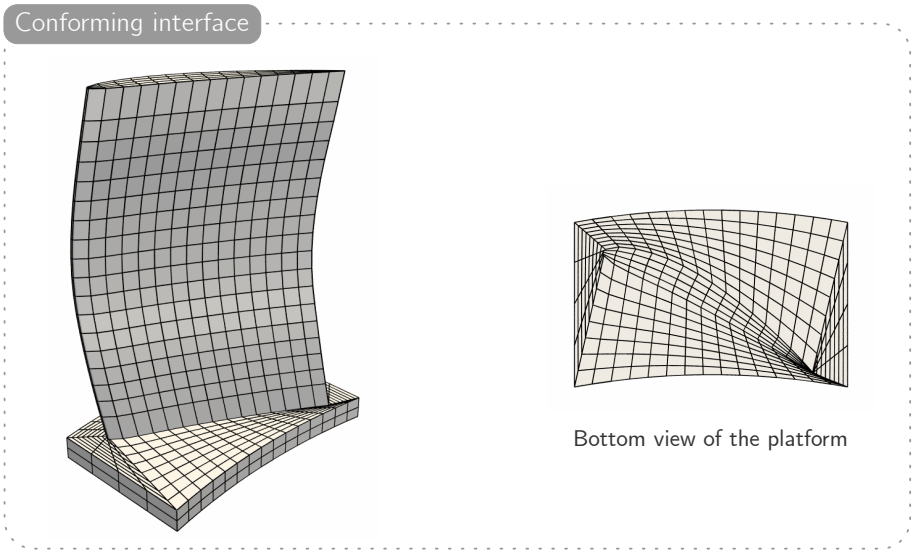
Re-parametrisation procedures The approach of modifying the parametrisation of the model in order to obtain a matching interface is relatively natural. For the analysis, the duplicated control points are concatenated, reducing the operation to connecting the elements on either side on the interface.

First considering the case of non-matching interfaces as illustrated in figure 4.10, one can use a combination of refinement and/or de-refinement techniques in order to achieve matching interfaces. In figure 4.10, knot insertion is used to yield a matching interface. In practice, it amounts to make the basis functions of the two sub-domains identical along the interface. Note in particular that C^0 lines are required on the boundaries of the interface in order to achieve this.

Achieving matching interfaces from non-conforming geometries is more difficult. The case of non-conforming interfaces cannot be dealt with using the same essential geometric operations (knot insertion and removal, degree elevation and reduction, etc.). Considering figure 4.11a for instance, the mock-up blade geometry crosses the elements of the upper surface of the platform. In this case, the basis functions cannot be adjusted in the interface region to become identical. Thus, it would be necessary to re-parametrise the whole domain of the platform in spite of its relatively simple geometric definition, as pictured on figure 4.11b. The platform has been divided into five patches to obtain matching interfaces. This example illustrates the difficulty behind the strategy of obtaining matching interfaces: the tensor product structure of NURBS soon becomes an obstacle, especially when volumetric parametrisations are in play.



(a) Initial parametrisation with non-conforming interface.



(b) Re-parametrised model to obtain matching interface.

Figure 4.11 – Example for the coupling of a mock-up blade and platform.

Coupling methods The second strategy to be able to analyse a multi-patch geometry is to use numerical tools. That way, it is possible to couple non-conforming interfaces without any re-parametrisation step. In the context of our work, it seemed more appropriate to favour this approach. Indeed, our final goal is to perform analysis and shape optimisation of a complete blade structure. Considering the fact that our final structure could involve up to half a dozen patches, choosing a re-parametrisation approach for the multi-patch analysis did not seem appropriate.

4.2.2.2 Coupling non-conforming interfaces

Description of the coupling problem and prevalent methods Before presenting the framework we use for multi-patch analysis, let us shortly review the main numerical methods that have been investigated in the particular case of isogeometric analysis. Three methods stand out: the penalty coupling (Leidinger et al., 2019, Leonetti et al., 2020, Pasch et al., 2021), mortar coupling (Bernardi et al., 1993, Temizer et al., 2011, Hesch and Betsch, 2012, Brivadis et al., 2015, Dornisch et al., 2015, Dornisch and Müller, 2016, Matzen and Bischoff, 2016, Bouclier et al., 2017, Hirschler et al., 2019c, Buffa et al., 2020, Chasapi et al., 2020), and Nitsche coupling (Nitsche, 1971, Sanders et al., 2012, Apostolatos et al., 2014, Nguyen et al., 2014, Ruess et al., 2014, Du et al., 2015, Guo and Ruess, 2015, Schillinger et al., 2016, Hu et al., 2018, Antolin et al., 2019c, Elfverson et al., 2019, Du et al., 2020). Specifically, Apostolatos et al. (2014) discuss these domain decomposition methods for their application in isogeometric analysis.

We hereafter set the holding equations for two non-overlapping sub-domains $\Omega^{(1)}$ and $\Omega^{(2)}$, and the interface $\Gamma_0 = \Omega^{(1)} \cap \Omega^{(2)}$ as represented on the diagram in figure 4.12. For $k \in \{1, 2\}$ and the unknown displacements $\boldsymbol{u}$:

$$\boldsymbol{div}\,\boldsymbol{\sigma}^{(k)} + \boldsymbol{b}^{(k)} = 0 \quad \text{in } \Omega^{(k)}, \tag{4.7}$$

$$\boldsymbol{\sigma}^{(k)} = \boldsymbol{D}^{(k)}\boldsymbol{\varepsilon}^{(k)} \quad \text{in } \Omega^{(k)}, \tag{4.8}$$

$$\boldsymbol{\sigma}^{(k)}\boldsymbol{n}^{(k)} = \boldsymbol{t} \quad \text{in } \Gamma_N \cap \partial\Omega^{(k)}, \tag{4.9}$$

$$\boldsymbol{u}^{(k)} = \boldsymbol{g} \quad \text{on } \Gamma_D \cap \partial\Omega^{(k)}, \tag{4.10}$$

with the body force $\boldsymbol{b}$, a symmetric positive definite matrix of material constants $\boldsymbol{D}$ (corresponding to linear elasticity in the present case), the boundary traction $\boldsymbol{t}$ and unit outward normal $\boldsymbol{n}$ defined on the Neumann boundary Γ_N, and the prescribed displacement $\boldsymbol{g}$ defined on the Dirichlet boundary Γ_D.

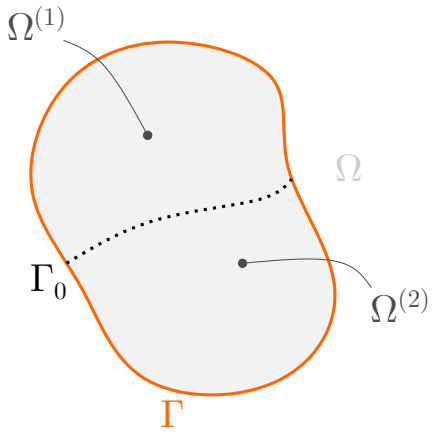


Figure 4.12 – Diagram describing the coupling problem set-up.

The problem described throughout equations (4.7) – (4.10) is divided into two different problems. In order to enforce the equality of mutual displacements along the interface, we must specify the following condition:

$$\boldsymbol{u}^{(1)} = \boldsymbol{u}^{(2)} \quad \text{on } \Gamma_0, \tag{4.11}$$

and the interface tractions have to satisfy:

$$\boldsymbol{t}^{(1)} + \boldsymbol{t}^{(2)} = 0 \quad \text{on } \Gamma_0. \tag{4.12}$$

The weak form of the problem (4.7) – (4.10) consists in the variation of the potential. It reads as:

$$\sum_{k=1}^{2} \left(\int_{\Omega^{(k)}} \delta\boldsymbol{\varepsilon}\, \boldsymbol{\sigma}\, d\Omega - \int_{\Omega^{(k)}} \delta\boldsymbol{u}\, \boldsymbol{b}\, d\Omega - \int_{\Gamma_N \cap \partial\Omega^{(k)}} \delta\boldsymbol{u}\, \boldsymbol{t}\, d\Gamma \right) = 0, \tag{4.13}$$

without consideration of the displacements and tractions conditions (4.11) and (4.12). The coupling conditions cannot be considered in the strong sense in the case of non-conforming interfaces. Hence, they must be imposed weakly. The three coupling approaches described hereafter are different in the way the contribution to the potential is formulated.

Penalty coupling In the penalty approach, the displacement condition (4.11) is enforced in a weak sense by multiplying it by its test twin and multiplying it with a penalty parameter $\frac{\alpha^{pen}}{2}$. The contribution of the penalty term reads as:

$$\int_{\Gamma_0} \frac{\alpha^{pen}}{2} \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) \cdot \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) d\Gamma. \tag{4.14}$$

The accuracy of the penalty approach strongly relies on the value of the penalty parameter, which is problem-dependant. On the one hand, setting a small value for α^{pen} might not be sufficient in order to ensure the compatibility of the constraints at the interface. On the other hand, high values of α^{pen} can lead to ill-conditioned discrete systems. Nevertheless, penalty coupling has successfully been applied to various cases, as recently showed by Herrema et al. (2019) and Johnson and Hsu (2020) for wind turbine blades for instance.

Mortar coupling The mortar coupling technique makes use of an additional field $\boldsymbol{\lambda}$, namely the Lagrange multiplier field. The interface tractions appearing in (4.12) can be re-written as follows:

$$\boldsymbol{t}^{(1)} = \boldsymbol{\sigma}^{(1)} \cdot \boldsymbol{n}^{(1)} = -\boldsymbol{\lambda}, \tag{4.15}$$

$$\boldsymbol{t}^{(2)} = \boldsymbol{\sigma}^{(2)} \cdot \boldsymbol{n}^{(2)} = \boldsymbol{\lambda}. \tag{4.16}$$

Mortar coupling consists in multiplying the displacement condition (4.11) with the unknown field $\boldsymbol{\lambda}$ and integrate it over the interface. The additional term for the potential of the mortar coupling term thus reads as:

$$\int_{\Gamma_0} \boldsymbol{\lambda} \cdot \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) d\Gamma, \quad \boldsymbol{\lambda} \in \mathcal{L}, \tag{4.17}$$

$\mathcal{L}$ being an *ad-hoc* space. An important matter when using Lagrange multiplier-based methods is that the *inf-sup conditions* (see Chapelle and Bathe, 1993) have to be fulfilled as the final problem constitutes a saddle-point problem.

Nitsche coupling In this method, the stresses at the interface are averaged by means of a weighted parameter. When the domains on either side of the interface share the same (or similar) material properties, it is a common practice to set this parameter to $\frac{1}{2}$. In that case, the contribution of Nitsche coupling for the potential reads as:

$$\begin{aligned} &- \int_{\Gamma_0} \left(\frac{1}{2} \left(\boldsymbol{\sigma}^{(1)} + \boldsymbol{\sigma}^{(2)} \right) \cdot \boldsymbol{n} \right) \cdot \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) d\Gamma \\ &+ \int_{\Gamma_0} \frac{\alpha^{Nit}}{2} \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) \cdot \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) d\Gamma, \end{aligned} \tag{4.18}$$

α^{Nit} being the stabilisation parameter, which can be viewed as the spring constant. A generalised eigenvalue problem can be solved over the interface in order to estimate the value of the stabilisation parameter, see Apostolatos et al. (2014) for instance.

Mortar coupling of isogeometric solids In this work we make use of the mortar approach in order to enforce the coupling conditions. Our decision criterion was based on the fact that using mortar coupling does not lead to the modification of the stiffness operators of each sub-domain. This aspect was important in order to incorporate the coupling aspect to an already-existing IGA code.

Complete weak form of the problem and discretisation aspects Let us dwell upon the mortar method in order to specify several aspects. The variation of the potential (4.17) is added to the weak form (4.13). Then, the complete weak form of the coupling problem using the Lagrange multiplier method reads as follows:

$$\begin{aligned} &\sum_{k=1}^{2} \left(\int_{\Omega^{(k)}} \delta\boldsymbol{\varepsilon} \, \boldsymbol{\sigma} \, d\Omega - \int_{\Omega^{(k)}} \delta\boldsymbol{u} \, \boldsymbol{b} \, d\Omega - \int_{\Gamma_N \cap \partial\Omega^{(k)}} \delta\boldsymbol{u} \, \boldsymbol{t} \, d\Gamma \right) \\ &+ \int_{\Gamma_0} \boldsymbol{\lambda} \cdot \left(\delta\boldsymbol{u}^{(1)} - \delta\boldsymbol{u}^{(2)} \right) d\Gamma + \int_{\Gamma_0} \delta\boldsymbol{\lambda} \cdot \left(\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(2)} \right) d\Gamma. \end{aligned} \tag{4.19}$$

Adding the supplementary terms to the potential turns the stationary problem into a saddle-point one.

Having this expression at hand, the isogeometric discretisation can be specified for the unknown displacements $\boldsymbol{u}^{h\,(k)}$, $k \in \{1, 2\}$, and the Lagrange multiplier field $\boldsymbol{\lambda}^h$:

$$\boldsymbol{u}^{h\,(k)} = \sum_i N_i^{(k)} \boldsymbol{u}_i^{(k)}, \quad k \in \{1, 2\} \tag{4.20}$$

$$\boldsymbol{\lambda}^h = \sum_i \overline{N}_i \, \boldsymbol{\lambda}_i, \tag{4.21}$$

using the nodal Lagrange multipliers $\boldsymbol{\lambda}_i$ and the associated NURBS basis functions $\overline{N}_i$.

The discretisation of the linearised weak form results in the following coupled system of equations:

$$\begin{bmatrix} \boldsymbol{K}^{(1)} & \boldsymbol{0} & \boldsymbol{C}^{(1)\,T} \\ \boldsymbol{0} & \boldsymbol{K}^{(2)} & -\boldsymbol{C}^{(2)\,T} \\ \boldsymbol{C}^{(1)} & -\boldsymbol{C}^{(2)} & \boldsymbol{0} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}^{(1)} \\ \boldsymbol{F}^{(2)} \\ \boldsymbol{0} \end{pmatrix}, \tag{4.22}$$

with the local stiffness matrices $\boldsymbol{K}^{(k)}$, load vectors $\boldsymbol{F}^{(k)}$, and the displacement coupling matrices $\boldsymbol{C}^{(k)}$, for $k \in \{1, 2\}$. The coupling matrices contain the following quantities:

$$C_{ij} = \int_{\Gamma_0} \overline{N}_i \, N_j \, d\Gamma, \tag{4.23}$$

which enables to fulfil the displacement condition on the interface (4.11) as follows:

$$\boldsymbol{C}^{(1)} \, \boldsymbol{u}^{(1)} = \boldsymbol{C}^{(2)} \, \boldsymbol{u}^{(2)}. \tag{4.24}$$

Integration points positioning We rely on figure 4.13 throughout this paragraph for clarity. We restrain the scope of our study to a situation where at least one of the domains in play has an entire face[1] involved in the coupling configuration. The geometry of the domains is not restricted. We depict the case of a planar interface in figure 4.13, yet it does not constitute a mandatory condition. As specified earlier, we consider the case of non-overlapping sub-domains with watertight geometries.

Several projections are required for the computation of the coupling matrices. Integrals are computed in the parametric space, whichever is the considered domain. However, under the assumption of non-conforming interfaces and non-linear parametrisation, there is no straightforward relation between the parametric space of both domains. Still, this link is required in order to ensure the correct positioning of integration points on both sides of the interface.

[1] Here we refer to a *parametric* face, considering that trivariate NURBS patches parametrically consist in cuboid entities.

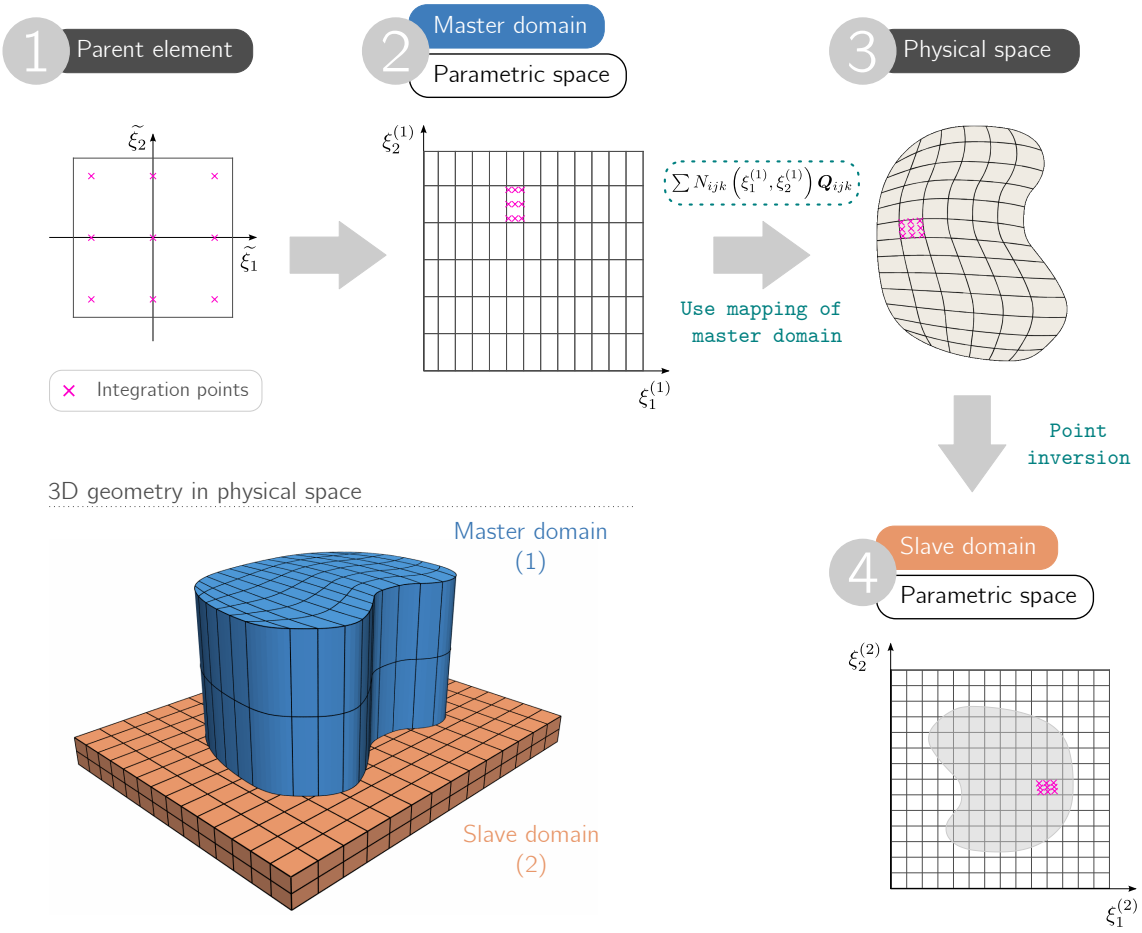


Figure 4.13 – Overview of the spaces and projections involved for the mortar coupling of isogeometric solids.

Formally, the interface can be described in the parametric space of each domain. For the first one, it is rather simple as an entire face of the volume is involved. That way, the interface is described by a square in parametric space. Later on, this domain is defined as the *master* domain. For the second domain, the interface description is not trivial. The borders of the interface region must be projected onto the face involved in order to identify the portion of parametric space that is in play. Conversely, this domain is referred to as the *slave* one.

Several compositions are needed in order to define the position of the integration points in the parametric space of the master domain, as pictured in figure 4.13. Their position in physical space is computed using the master domain mapping. This operation is required in order to define the position of the integration points into the parametric space of the slave domain. Hence, a *point inversion* (see Piegl and Tiller, 1996, pp.229-234) is performed in order to recover the parametric coordinates corresponding to the physical ones, in the slave domain parametric space. This procedure ensures a

consistent position of the integration points in both parametric domains.

Projection procedures add computational cost to the overall set-up. However, we believe that using these numerical tools still offer a significant advantage with respect to re-parametrisation methods, such as presented in figure 4.11 for instance.

4.2.3 Assessment of the implementation

4.2.3.1 Reference examples

We first start by assessing the implemented formulation onto a couple of reference numerical examples, namely the cantilever beam and the patch test. The two examples are tested with three configurations each: one presenting matching interfaces, one with non-matching interfaces, and lastly we address non-conforming interfaces.

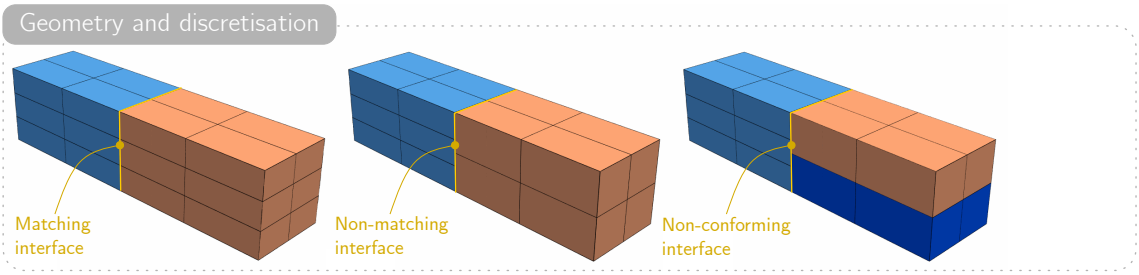
Cantilever beam The cantilever beam test is first considered. We consider a beam of length $L = 4\,\text{mm}$ and height $h = 1\,\text{mm}$. We use quadratic elements, and material properties are as follows: Young's modulus $E = 110 \times 10^3\,\text{Mpa}$ and Poisson's ratio $\nu = 0.34$. One end of the beam is clamped while a tangential uniform load is applied on the other end, with $P = 1\,\text{Mpa}$. The three configurations and associated results are pictured in figure 4.14. The displacement magnitude corresponds to the analytical one in all three cases.

Patch test For the patch test case, the geometry we use is a basic block of unit dimensions, where the boundary conditions and loading are applied as represented on figure 4.6. The tested configurations for the patch test are represented in figure 4.15a. For the non-conforming case three patches are used.

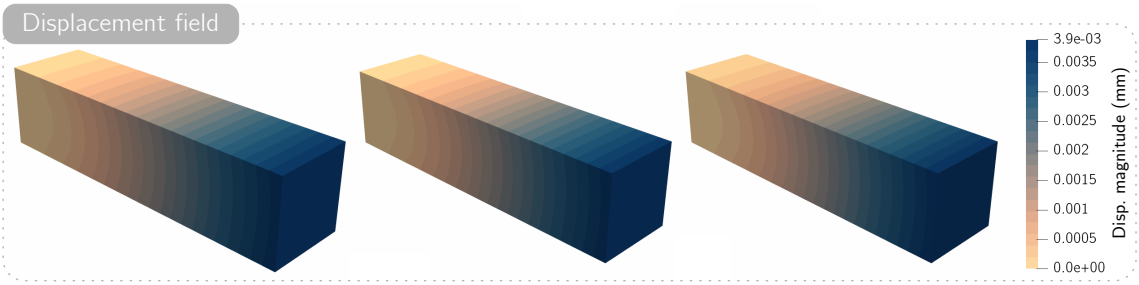
The results obtained are presented on figure 4.15b, where the strain magnitude is plotted. All three cases pass the test. In particular, one can see the strain presents a constant value throughout all domains.

4.2.3.2 Cuboid assembly

To continue the testing process of the coupling implementation, let us now consider a more complex case in terms of number of interfaces. We tackle the geometry that is described in figure 4.16a. It consists in the assembly of various cuboid patches. Our goal is to feature an assembly that would present the same complexity as the final actual blade assembly, including a platform and fillet. We believe this assembly shows sufficient geometric complexity, considering that the four patches surrounding the centre cube actually have to be coupled in corner regions (which are known to be numerically problematic), and given the overall number of interfaces in the model (thirteen in total).

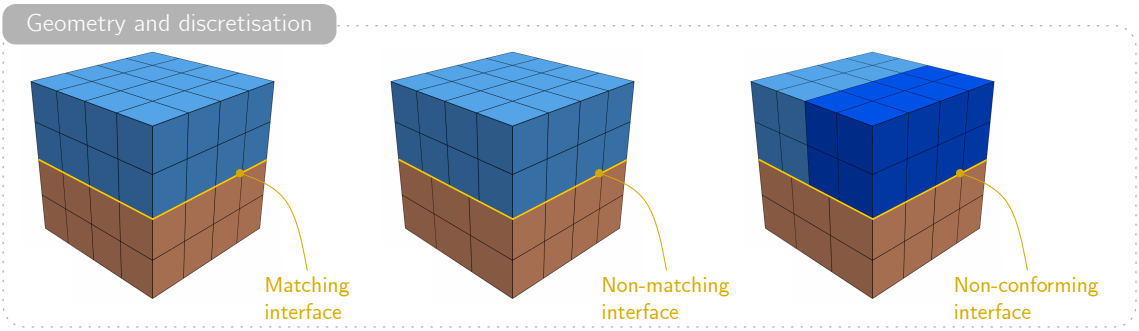


(a) Geometry and discretisation: each color correspond to a patch.

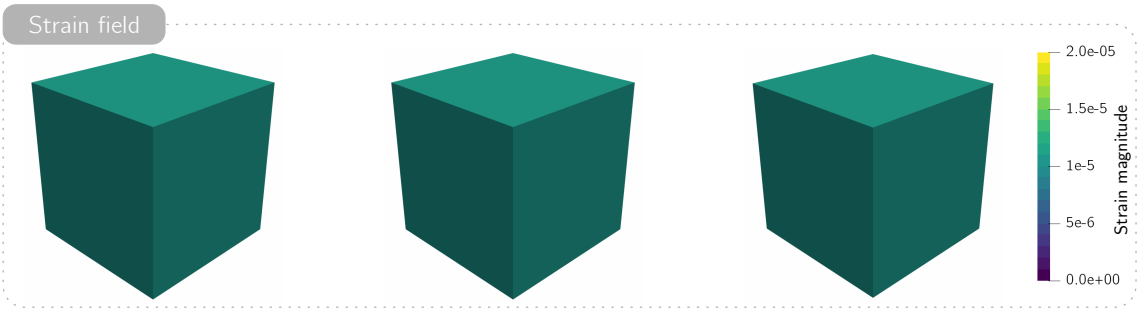


(b) Displacement field: the displacement magnitude is plotted.

Figure 4.14 – Cantilever beam configurations and results.

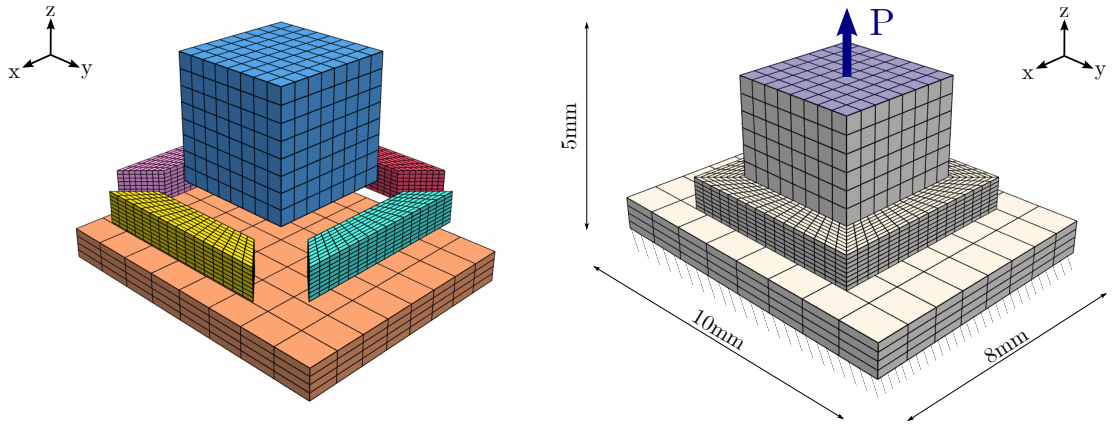


(a) Geometry and discretisation: each color correspond to a patch.



(b) Strain field: the strain magnitude is plotted.

Figure 4.15 – Patch test configurations and results.



(a) Exploded view of the geometry and discretisation. Each color corresponds to a patch.

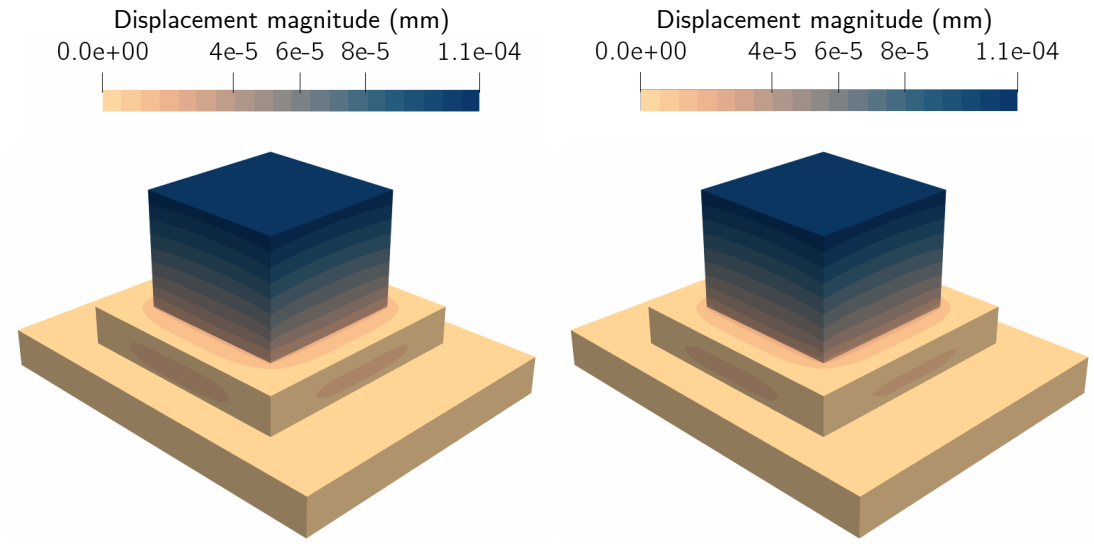
(b) Boundary conditions and loading.

Figure 4.16 – Analysis set-up for the cuboid assembly case.

In order to assess this model, the analysis set-up is as follows. The material properties are the same as described in the previous beam example in section 4.2.3.1. The dimensions of the patches as well as boundary condition and loading are indicated on figure 4.16b, right. The magnitude of the uniform pressure is $P = 3\text{N}/\text{mm}^2$.

Our goal in this case is to ensure that the behaviour of the model is *globally* recovered. Yet we also investigate local performance, especially in the regions of interfaces and in the corners of the patches surrounding the centre cube. We chose to compare our solution to a commercial finite element software (ABAQUS). The set-up is the same as described in figure 4.16. It is however different in terms of number of patches, as solely three cuboid patches are used. They are linked together by means of a means of a `Tie` constraint (Dassault Systèmes SE, 2020).

The global behaviour of the assembly is pictured in figure 4.17. The displacement magnitude is featured for both cases. The magnitude is the same, as well as its repartition. The performance in the regions of interface is analysed in the y-normal mid-plane of the assembly. Each component of the displacement is then plotted for various heights z, along the x directions. The goal of such study is to highlight the possible jumps in displacement when an interface is crossed. We remind here that since we use a mortar approach to solve the coupling problem, we ensure a global behaviour in the regions of interface. Jumps in the displacement values are thus bound to happen in the framework we use. The results are presented in figure 4.18. The x- and z-components of the displacement are plotted along the x direction. The y component is null on the studied paths.



(a) Results obtained with YETI. **(b)** Results obtained with ABAQUS.

Figure 4.17 – Comparison of displacement magnitude for the cuboid assembly.

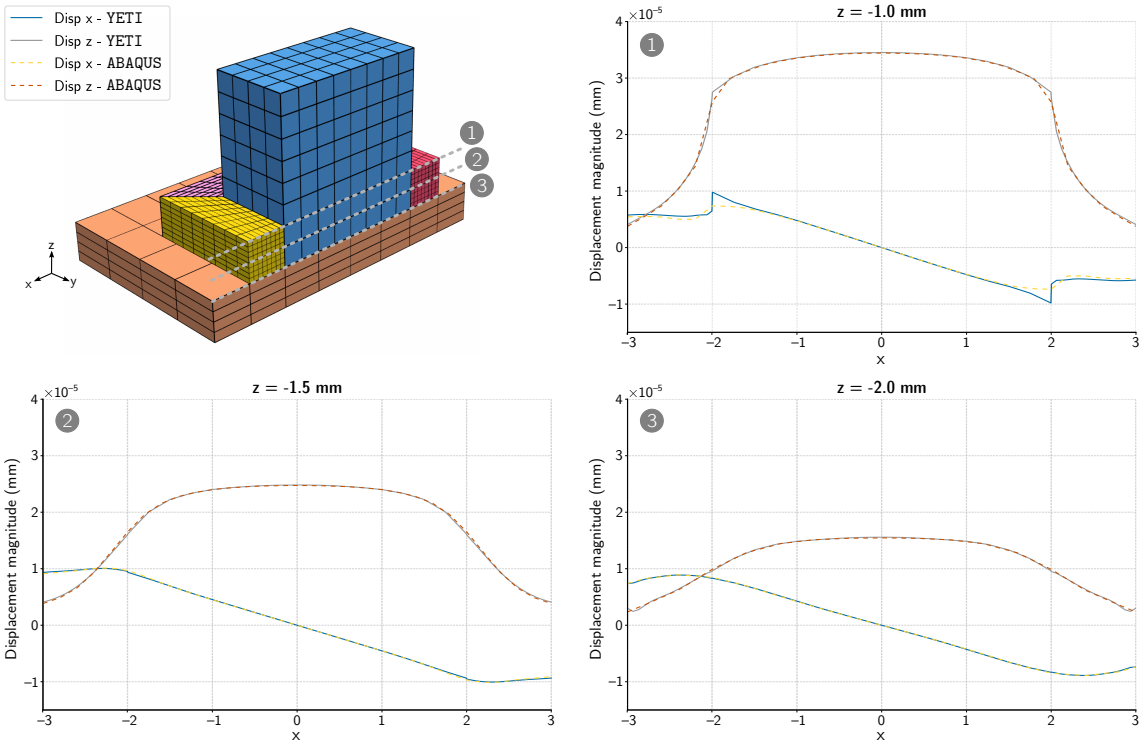


Figure 4.18 – Analysis of the behaviour in the interface region for the cuboid assembly.

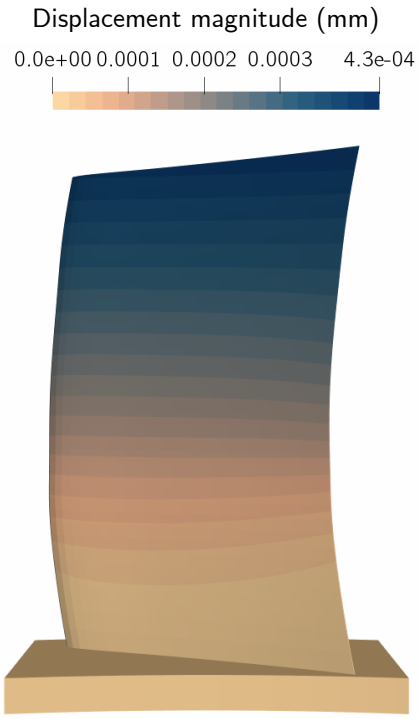


Figure 4.19 – Results for the centrifugal force example on the mock-up blade model.

4.2.3.3 Mock-up blade and platform assembly

As a last example we take the geometry presented in figure 4.11 in order to test our implementation onto a non-conforming interface featuring more realistic geometries. The analysis set-up is as follows. Symmetry conditions are applied onto the platform, and the blade is coupled to the platform. The whole geometry is subject to centrifugal force of arbitrary angular velocity of 3000 rpm. The material properties are the same as used in the previous numerical examples. The resulting displacement field is pictured in figure 4.19. As a matter of fact, in a more realistic application, the connecting zone between the blade and the platform would feature a fillet for a smoother transition. This aspect is addressed in the forthcoming chapter.

4.3 A convenient framework for the design and analysis of complex multi-patch structures

The two previous sections brought the elementary components to assemble a convenient framework for the design and analysis of complex multi-patch structures.

As observed in section 4.2, coupling techniques reach a limit when one seeks to achieve matching interfaces (especially in a trivariate NURBS modelling context). Advanced numerical tools exist in order to couple non-conforming sub-domains. The

parametrisation nesting presented in section 4.1 brings a favourable framework for the shape modification of complex geometries. This aspect is particularly interesting for studies dealing with shape optimisation. We further illustrate this aspect in the following. Still, the question of coupling several domains featuring the proposed embedded solid element formulation remains open.

4.3.1 Dealing with coupled embedded entities

4.3.1.1 Implementation

The proposed coupling approach described in section 4.2, and summed up in figure 4.13, does not offer the possibility of using the embedded solid element formulation. As it certainly did not escape some readers, the point inversion operation yielding step 4 might need some adaptation in the case of an embedded slave domain. Another point not to miss lies in the case of dealing with an embedded master domain: there is an additional NURBS composition step before reaching step 3 so as to obtain the integration points position in physical space. Figure 4.20 pictures the additional steps required, paralleling with figure 4.13.

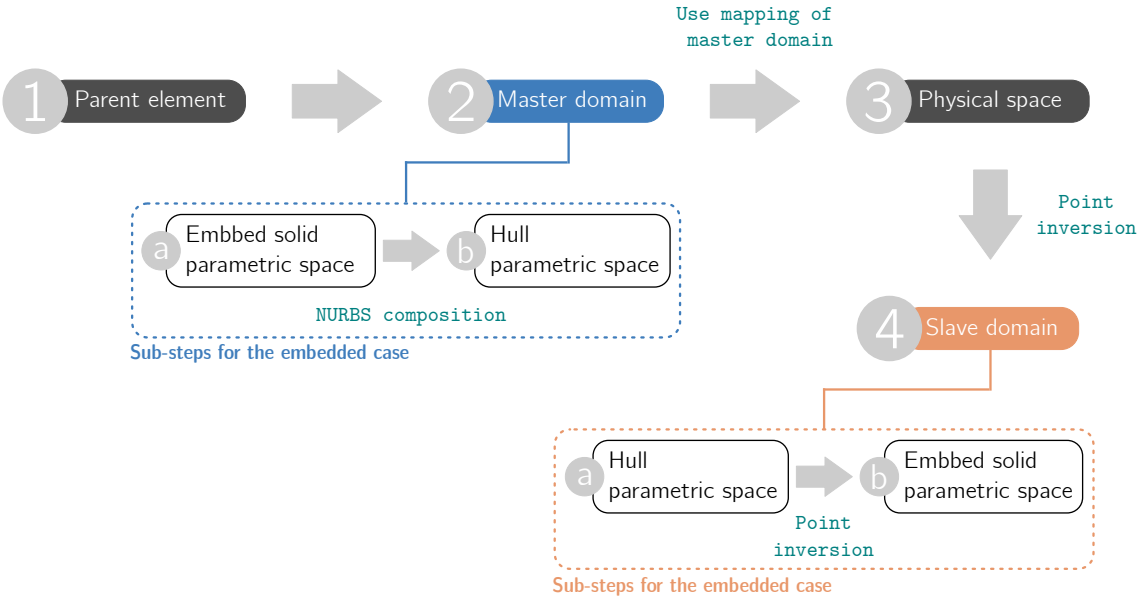


Figure 4.20 – Sum-up of the spaces and projections involved in the coupling of embedded solids.

4.3.1.2 From the user point of view

Considerable effort has been put into providing an implementation that is not too cumbersome for the user point of view. Another focal point was the genericness of the proposed implementation. In that context, a convention that seemed acceptable was to

consider that the hull face number and embedded solid face number were the same, that is, that the parametric directions of an embedded object and its hull are consistent.

4.3.2 Embedded solids for shape design and modification

The numerical examples of section 4.1.3 assess the implementation of the embedded solid element, using various variations of the patch test. The behaviour observed for these examples is as expected. In the context of complex geometric modelling, let us now illustrate the potential of this approach for geometric modelling.

4.3.2.1 Complex geometric design

The embedded approach presents an interesting potential for complex geometric design. In order to illustrate our point, we come back to our earlier example of cuboid assembly from section 4.2.3.2. For this configuration, six patches are used, and their geometry is simplistic. What could be done in this case would be to embed them into more complex hull objects. That way, the data setting is not modified — apart from the embedded formalism —, and complex shapes can be obtained. What is more, a variety of different shapes can be attained from a simple initial layout.

This aspect is summed up in figure 4.21. Two examples are provided herein: first, the main cube is morphed to the potato-shaped volume from section 4.2.2.2. Then, the main cube is shaped as the mock-up blade model. The hull boxes are displayed for both cases.

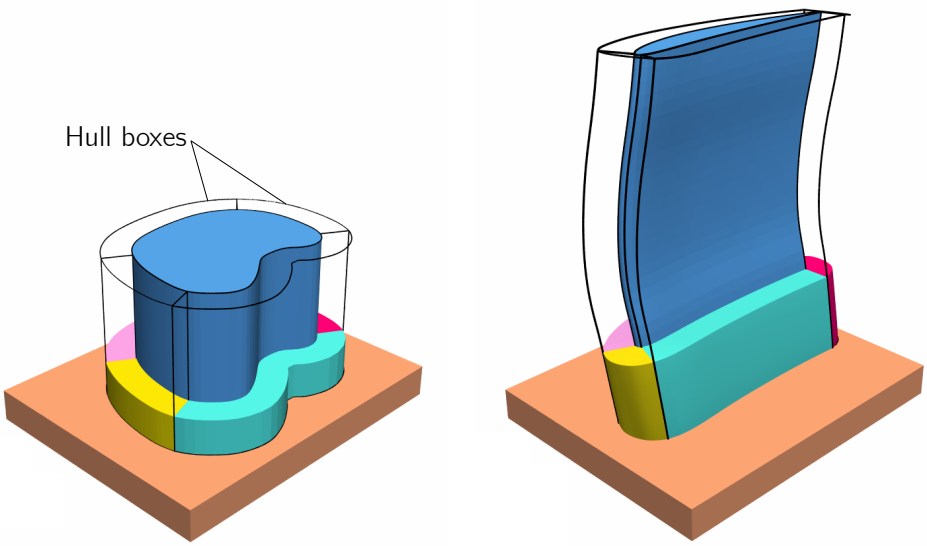


Figure 4.21 – Two examples of complex geometries based on a cuboid assembly, using the embedded solid elements.

4.3.2.2 Specific shape modifications

Another attractive aspect brought by parametrisation nesting is its potential for shape modifications. We illustrate in this paragraph how we consider using it in the context of parametric shape optimisation.

The reasoning lying behind this approach originates in turbine blades sizing methods, yet is not restricted to this application. We refer to section 2.1 for aircraft engines blades design methodology. Here we focus on the existence of a fillet at the bottom of the blade — or any geometry where a fillet exists. Fillets are usually present in CAD representations, yet when passing on to analysis they tend to be neglected as a non-essential feature. Still, when studying stress concentration, the absence of a smooth shape in angles can lead to inconsistent results. Not neglecting these shapes for analysis can thus be of interest.

In particular, in the course of structural shape optimisation, the mechanical response of an assembly is evaluated in order to compute quantities of interest. If stresses are in play, preserving fillet representations is relevant. However, a concern in that case is: *what happens if the shape of the assembly is modified in the fillet region?*

In order to address this question, we propose a first part of an answer based on the embedded approach. We consider cases where the fillet geometry is not targeted by the shape optimisation procedure. Then, if the fillet is embedded into hull boxes that are linked to the geometry under modification, the shape adaptation is straightforward. This aspect is illustrated in figure 4.22 with the potato-shaped solid. The shape of the surrounding hull boxes is modified according to the potato shape modifications. The connection of the surrounding patches is thus ensured, without further worrying about the shape modifications aspects.

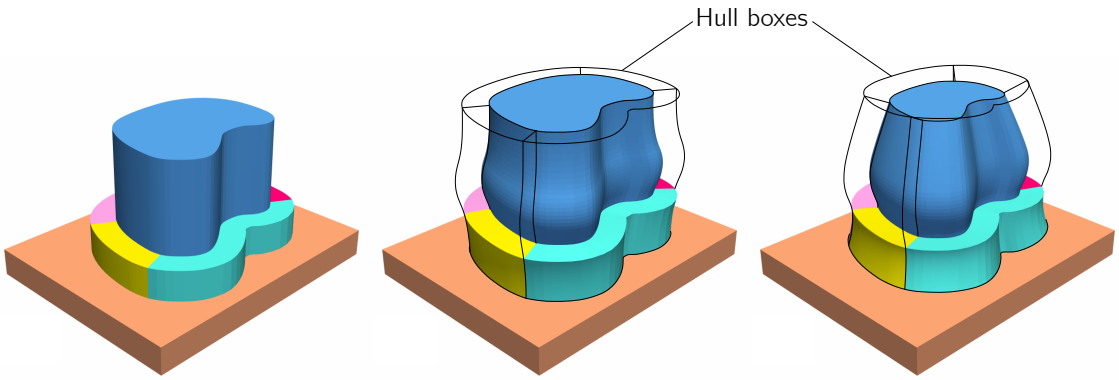


Figure 4.22 – Modifications of the shape of the leader patch induce the modification of the surrounding embedded patches.

Concluding remarks

Using multiple patches is an issue inherent to NURBS modelling, given the tensor-product structure of resulting geometries. We address this question in the context of solid elements modelling, with the ultimate aim of being able to model and run analysis of a complete blade model.

The first aspect we bring to light is the development of a so-called embedded solid element, using parametrisation nesting. NURBS composition is used in order to describe the final geometry. By embedding a given solid geometry into a more complex one, it is possible to handle complex geometric design and to impose specific shape modifications. Secondly, the need for multiple patches leads to the question of the coupling methodology used for the analysis of complex structures. We use a mortar coupling approach, with specific processing for the positioning of the integration points, ensuring robustness and genericness.

The use of both techniques concurrently offers several promising possibilities, among which shape modifications throughout shape optimisation procedures and complex geometric modelling.

Chapter 5

Construction and analysis of a complete blade model

Providing a framework for the analysis of actual industrial blade geometries is one of the final goals of this work. Based on the results and developments presented in the previous chapters, we finally set out the analysis of a complete blade model. We first come back to the notion of analysis-suitable models, providing some context elements. The construction of the complete geometry is then explained. Finally, numerical results are presented.

Contents

5.1 Analysis-suitable geometric descriptions

Before presenting into detail the geometric description used for the analysis of the complete blade model, we shall introduce some (more) concerns that arise in the quest of achieving analysis-suitable models. We already listed some of them in section 4.2, mentioning the need of multiple patches to represent complex geometries. The main argument supporting such claim is the tensor product structure of NURBS parameter space. We also emphasised that our strategy in this work is to use untrimmed geometries.

As we now tackle the question of modelling a whole blade assembly, featuring an actual industrial geometry, we are confronting the issue of describing a geometry that is defined in a CAD software. For the reader to understand better our approach to tackle this issue, let us give some general background concerning two standard concepts in CAD geometric representations: Boundary Representations and trimmed geometries.

5.1.1 Standard concepts in CAD geometric representations

5.1.1.1 Boundary representation

B-Rep is the most common representation of CAD geometries. As its name suggests, this paradigm consists in representing an object by its boundaries. Hence, a volume is represented by its boundary surfaces rather than a volumetric discretisation. In a nutshell, the geometry is represented in terms of its "skin". For example, the simple part represented on figure 5.1 actually is represented through more than a dozen boundary surfaces.

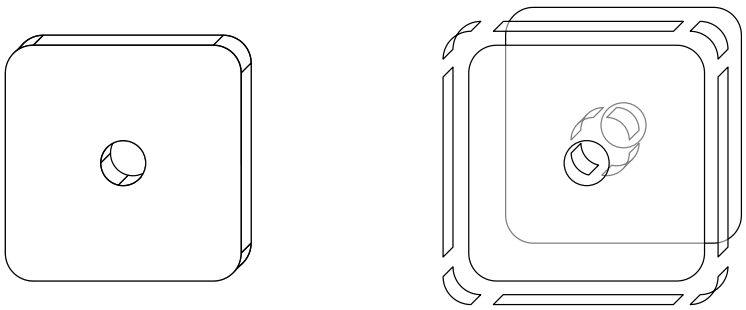
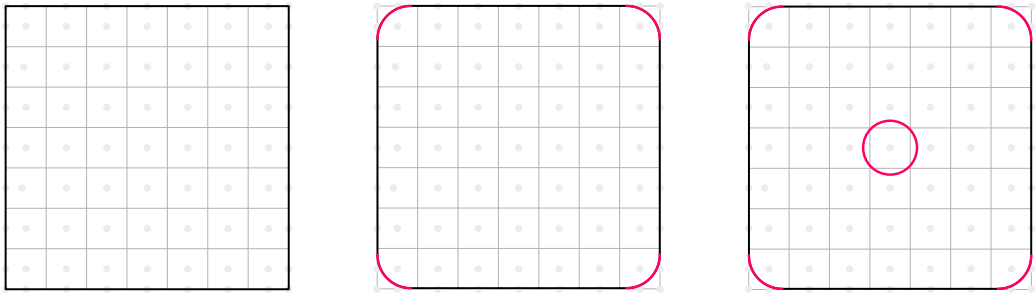


Figure 5.1 – B-Rep model of a simple part.

A B-Rep object can be decomposed in three categories of entities: its *faces* (usually surfaces), its *edges* (usually curves), and its *vertices* (consisting in points). They form the *features* of the B-Rep object. The boundary surfaces are connected, thus defining the limit between exterior and interior points of the geometry. The description of these surfaces depends on the way they are constructed. Most of the time, they actually consist in trimmed surfaces.

5.1.1.2 Trimmed surfaces

Trimmed surfaces can be seen as surfaces which only have a part of their domain that is displayed on screen. Hence, their parametric description does not match their physical existence. It is usually done by adding information to the surface instance: a *trimming curve* is used to define the boundary between the displayed part of the surface and the rest of its domain. The parametric definition of the surface is not modified. As a result, the trimmed surface has the same control points, degree and knot vector as the untrimmed surface it is based on. Figure 5.2 pictures the holed surface from the previous example. One can note that the control points nor the parametric lines are modified between the initial and the final surface.



(a) Initial surface. **(b)** Filleted surface. **(c)** Holed and filleted surface.

Figure 5.2 – Example of a trimmed surface. Grey dots ● represent the control points, grey lines — represent the parametric lines, and magenta curves — represent the trimming curves.

Trimming is of core importance in CAD software, in order to represent the objects obtained by common operations such as extruding, chamfering, filleting, sweeping and so on. Taking the above example again, trimming is necessary in order to represent the hole and the fillets.

5.1.2 Strategies for the analysis

5.1.2.1 Running analysis on trimmed geometries

A large and growing body of literature has investigated isogeometric analysis of trimmed geometries. Several fields are involved on this topic, starting with static analysis on trimmed geometries on single-patch models (Kim et al., 2009, 2010), and further on multiple conforming patches (Schmidt et al., 2012). Further topics have been covered, such as explicit dynamics (Leidinger et al., 2019), coupling aspects (Ruess et al., 2014, Coradello et al., 2021, Pasch et al., 2021), and shape optimisation (Kang and Youn, 2016). Volumetric Representation based on trimmed B-Spline trivariates has been tackled by Massarwi and Elber (2016), and Antolin et al. (2019b) address the analysis of V-Reps. Finally, Marussig and Hughes (2018) propose a thorough review of trimming in isogeometric analysis.

5.1.2.2 Building untrimmed solid representations

Isogeometric analysis of trimmed geometries is a subject of its own, that has not been tackled in the course of this work. Other strategies are available, such as reconstructing untrimmed geometries from trimmed ones. For instance, Hui and Wu (2005) proposed a framework to generate regular surfaces from trimmed geometries. We chose to address the modelling of the geometry from this angle, by reconstructing untrimmed trivariates from B-Reps.

5.2 Construction of the complete blade assembly

In the following we give the main steps yielding a complete blade model, consisting in no less than 11 patches. The platform and tenon part specifically took notable effort in terms of geometry generation. The fillet part is modelled using the previously presented embedded solid element. For the sake of clarity, we hereafter repeat figure 2.8, that pictures the assembly in question.

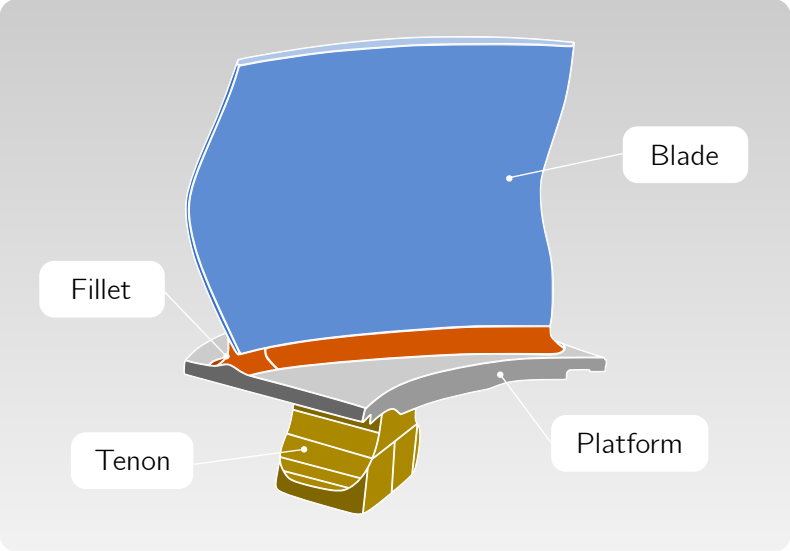


Figure 2.8 – Stylised HP compressor blade assembly (repeated from page 30).

5.2.1 Platform and tenon

In the following we detail the main geometric features of the platform and tenon. The creation of these parts strongly relies on the industrial context (see section 2.1). Additionally, let us specify we use the software Rhino® (McNeel, 2021) for the creation of the geometry.

The platform and tenon forming part of the blade assembly are depicted in figure 2.8. As it may not be blindingly obvious at first glance, this geometry reveals itself quite complex. Consequently, the elementary operations carried out to create the CAD model

are numerous. In order to demonstrate the industrial relevance of this work, it seemed of crucial importance to be able to construct a similar geometry. As showed throughout this manuscript, obtaining an IGA-ready volumetric model is far from being trivial.

5.2.1.1 Overview of prominent geometric aspects

The operations carried out in CAD software to create the geometric model of a given object are strongly related to crafting processes. For the platform and tenon part, the industrial context has a strong role in the definition as well. Let us outline the most important geometric features of those parts. We rely on figure 5.3 to support our explanations.

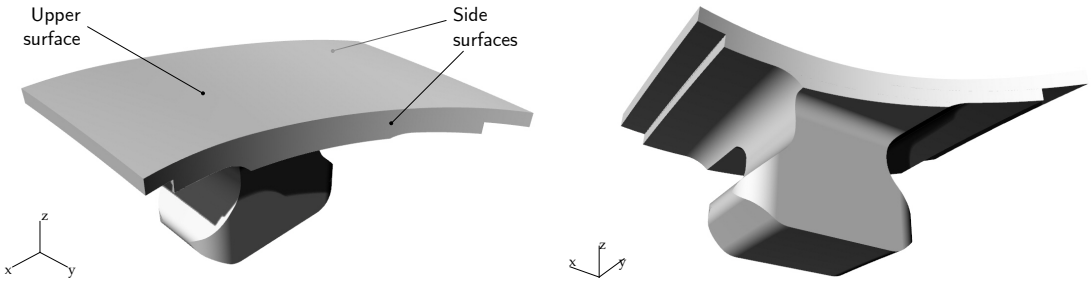


Figure 5.3 – Rendered perspective views of the platform and tenon parts.

Non-planar swept surface The upper surface of the platform – *i.e.*, the surface on which the blade is to lie – consists in a swept surface around the y axis (see figure 5.3). The profile that is swept is not a planar curve, but rather consists in a flow line. Recalling sub-subsection 2.1.1.2 about the definition of the geometry of axial compressor blades, we can say that the upper surface of the platform consists in a portion of stream surface. It is an important aspect to consider for the forthcoming modelling stages, as this surface cannot be described analytically.

Platform sides The platform side surfaces are cut using cylindrical surfaces. To picture it more clearly, seen from above, the sides consist in circular arcs. An important point to note is that the arcs are of the same radius, with different centres. This aspect allows for the right adjustment of the platforms on the complete blade wheel: the platforms slot together. This aspect is illustrated in figure **??**.

5.2.1.2 Techniques employed

Generating the geometry within Rhino In order to achieve the platform and tenon model, the operations carried out within the CAD software have been reproduced using

Rhino. These operations include sweeping, creating pockets and grooves, and so forth. At this stage, the geometry consists in a B-Rep, defined by (trimmed) boundary surfaces. In order to achieve a volumetric definition, the geometry is split into smaller geometric entities that can be represented by a NURBS patch. That is, we sought to identify where to split the geometry so as to obtain parametrically cuboid geometries. The split geometry is pictured in figure 5.4 and results in six B-Reps.

Keeping in mind that Rhino provides trimmed surfaces, the outer surfaces of the cuboid parts cannot be exported as is for the creation of volumetric patches. The chosen approach was eventually to export the edges of the B-Reps, for which the actual description is available. Hence, for each of the cuboid B-Reps pictured in figure 5.4, the edges are exported, with the final goal of using these boundaries in order to recover the (untrimmed) outer surfaces.

Processing the geometry to achieve a volumetric description The geometry processing for the platform and tenon is based on the same techniques and algorithms used as described in section 3.1 for the reconstruction of the volumetric blade. More precisely, the outer surfaces of each patch are generated using Coons patches, given the boundary curves. An exception lies for the two bottommost B-Reps forming the tenon, for which some faces have been reconstructed using lofted surfaces in order to recover their strong curvature. Then, the re-parametrisation framework presented in section 3.1 has been used to yield compatibles sets of surfaces for generating volumetric patches. Going throughout this process thus enables to successfully put to contribution the framework we initially developed for the blade.

The final parametrisation of the platform and tenon is depicted in figure 5.5.

5.2.2 Adjusting the blade geometry

Once the platform geometry has been defined and positioned in space with respect to the rest of the engine, the blade geometry is adjusted. The blade is thereby clipped twice: on its lower side for the intersection with the upper face of the platform, and on its upper side accounting for the clearance with respect to the engine casing. This aspect is pictured in figure 5.6.

From a practical point of view, the clipping objects both consists in surfaces. Hence, Rhino is used to carry out surface-to-surface intersection operations between the blade outer surface and the cutting surfaces. Such operation yields two curves, which are used to update the original input curves depicted in section 2.1.1.2. The framework presented in section 3.1 to build a volumetric description of the blade can further be carried out seamlessly.

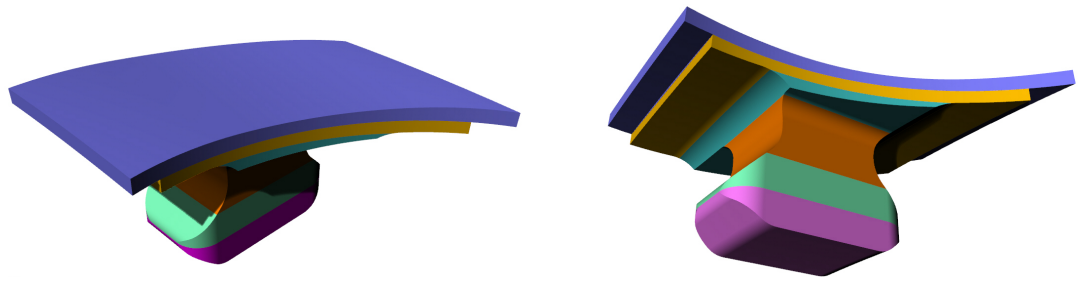


Figure 5.4 – Rendered perspective views of the six B-Reps constituting the platform and tenon.

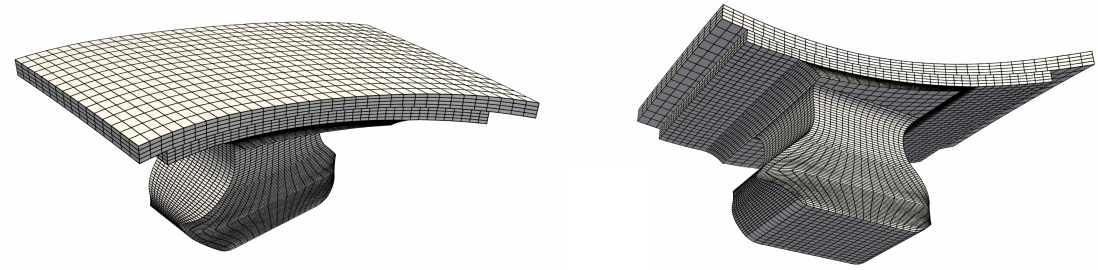
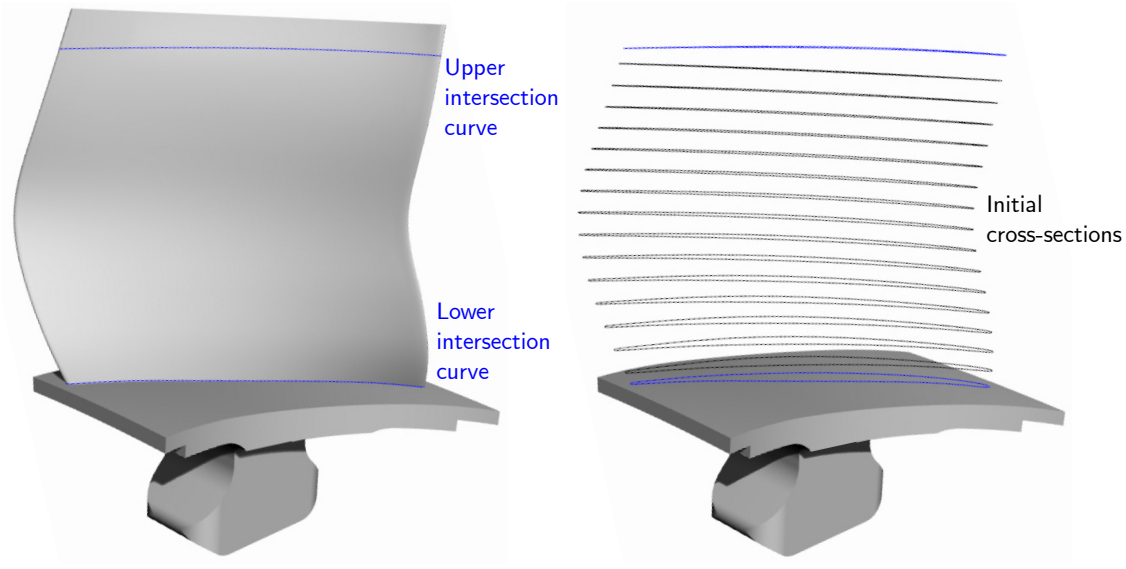


Figure 5.5 – Perspective view of the volumetric geometry of the platform and tenon with parametric lines.



(a) Blade outer surface and platform. **(b)** Updated cross-sections and platform.

Figure 5.6 – Adjustment of the blade geometry.

5.2.3 Fillet

Having the platform at hand, in the following we detail the construction of the fillet part. We make use of the embedded solid element described in section 4.1 in order to ensure the geometric compatibility of the fillet with the blade and platform, respectively. We first set out how the hull boxes are generated, and further detail the chosen geometry and parametrisation for the embedded fillet geometry. The process is summed up in figure 5.7.

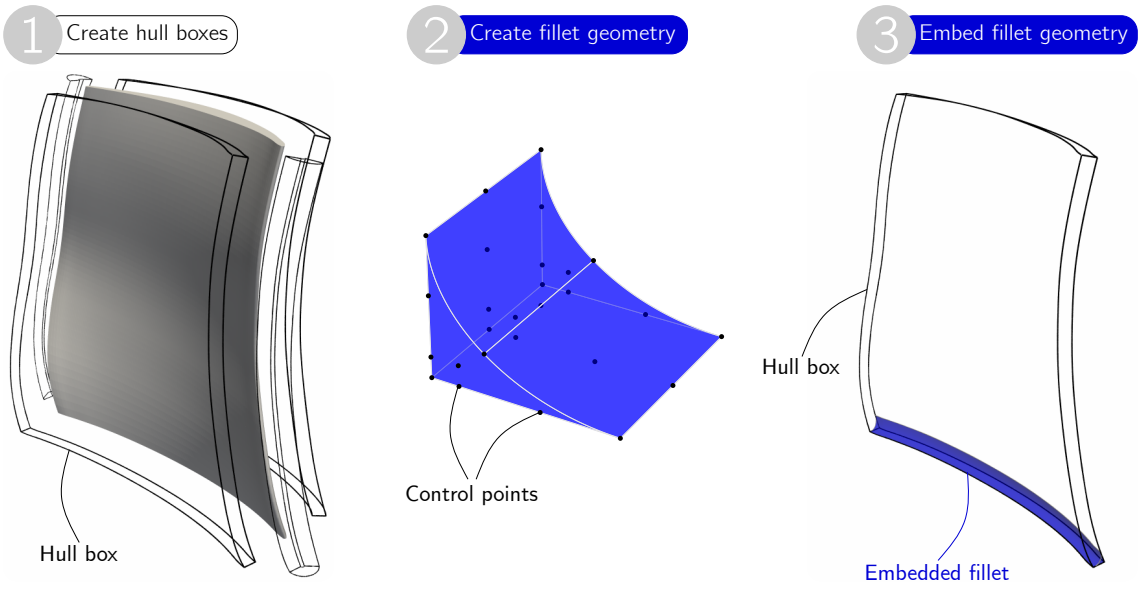


Figure 5.7 – Stages to build the embedded fillet patch for the HP compressor blade.

5.2.3.1 Hull boxes

As indicated in section 4.1, we base the parametrisation of the hull boxes onto the *leader* patch, which in this case happens to be the blade part. We hereafter detail the process for the construction of one box. It can be divided in four main stages, which should be considered with reference to section 3.1. The final goal is to generate six boundary surfaces to assemble a volume. We always start from an outer surface belonging to the leader patch, then go through the following stages. They are illustrated on figure 5.8.

Create the extended surface The so-called *extended surface* is created based on the outer surface under consideration. It defines the limit where the hull box reaches towards the borders of the platform. Without being a surface parallel projection, the developed process can be compared to it. It relies on the use of the control points of the outer surface as a starting point, and then applying modifications to their spatial coordinates. The following stages are applied to each control point $\boldsymbol{P}$.

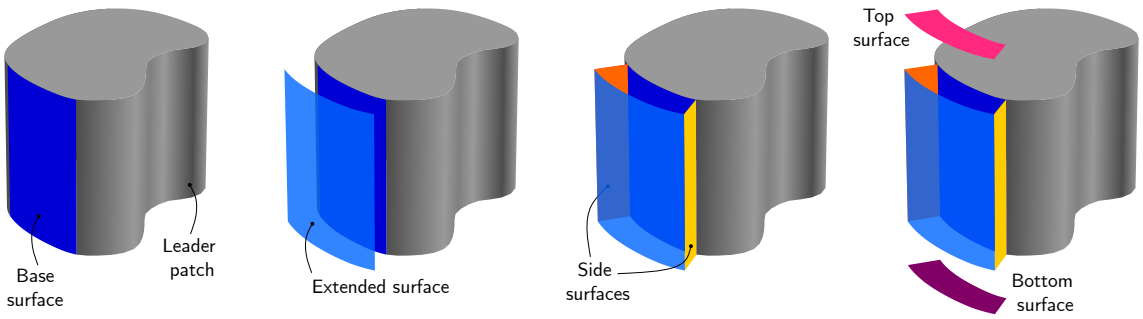


Figure 5.8 – Hull boxes construction steps.

- The control point $\boldsymbol{P}$ is projected onto the surface — as control points do not lie on the surface in the general case —, yielding parameters (u, v),
- the first derivatives of the surface $\mathcal{S}_u$ and $\mathcal{S}_v$ are evaluated at parameters (u, v).
- The normal vector $\boldsymbol{n}$ is computed for these parameter values using the following expression:

$$\boldsymbol{n} = \frac{\mathcal{S}_u \times \mathcal{S}_v}{|\mathcal{S}_u \times \mathcal{S}_v|}, \tag{5.1}$$

  which yields the normalised normal vector.

- The new value of the control point coordinates $\widetilde{\boldsymbol{P}}$ is computed, using the following relation:

$$\widetilde{\boldsymbol{P}} = \boldsymbol{P} + \alpha \cdot \boldsymbol{n}, \tag{5.2}$$

  where α is a user-defined value scaling the actual offset between the initial surface and the extended one.

The extended surface is defined by the newly computed control points, and has the same degrees and knot vectors as the outer surface it is based on.

Create side surfaces The side surfaces are created by linear interpolation of the sides of the two existing surfaces (known as a *ruled surface*). Hence, the side surfaces present the same degree and knot vector in the shared direction as the existing surfaces. Degree elevation is performed in the remaining direction to reach the same degree in both directions.

Create top and bottom surfaces As in chapter 3, the top and bottom surfaces are obtained by using a bilinear interpolation of boundary curves (Coons patch).

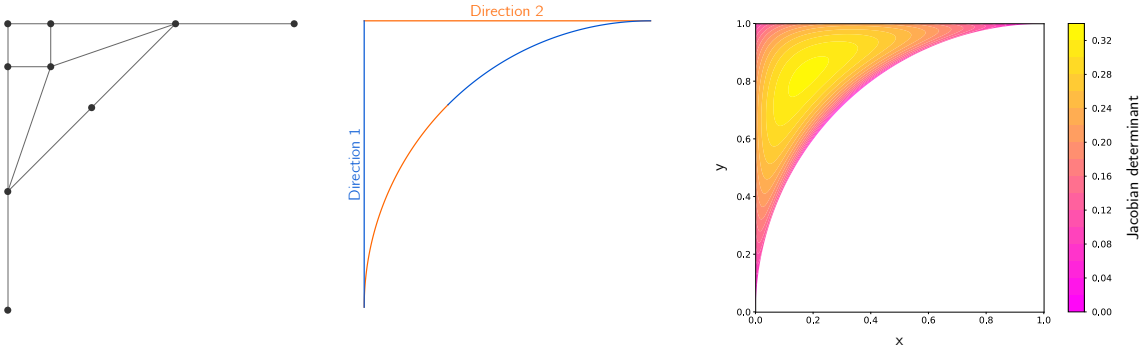
Assemble the final volume Finally, having the six boundary surfaces at hand, the volumetric parametrisation of the hull box is computed using the volumetric Coons patch.

After generating the hull boxes, their bottom control points are adjusted so as to lie on the platform. Putting to good use the *NURBS book* (Piegl and Tiller, 1996) and its projection algorithm once again, we project the bottom control points of the hull boxes onto the upper surface of the platform. This additional step prevents the embedded fillet to lie outside of the platform, as it can sometimes be the case. For instance, see figure 2.8: on left-hand part of the figure, one can note the fillet is slightly cut.

Having the hull boxes at hand, let us now describe the geometry we shall embed into.

5.2.3.2 Embedded geometry

For the geometry of the embedded fillet, we based ourselves on the exact circular fillet patch proposed by Lipton et al. (2010). As mentioned in their work, the geometrical mapping of such patch leads to almost-zero Jacobian determinant at the corners, while having a strictly positive Jacobian determinant in the interior of the patch (see figure 5.9). The numerical investigations carried out herein suggest that the fillet patch performs well. The cited reference contains all the necessary data to build a similar fillet (*i.e.*, control points coordinates and weights, and knot vectors). It consists in a bi-quadratic NURBS with nine control points.



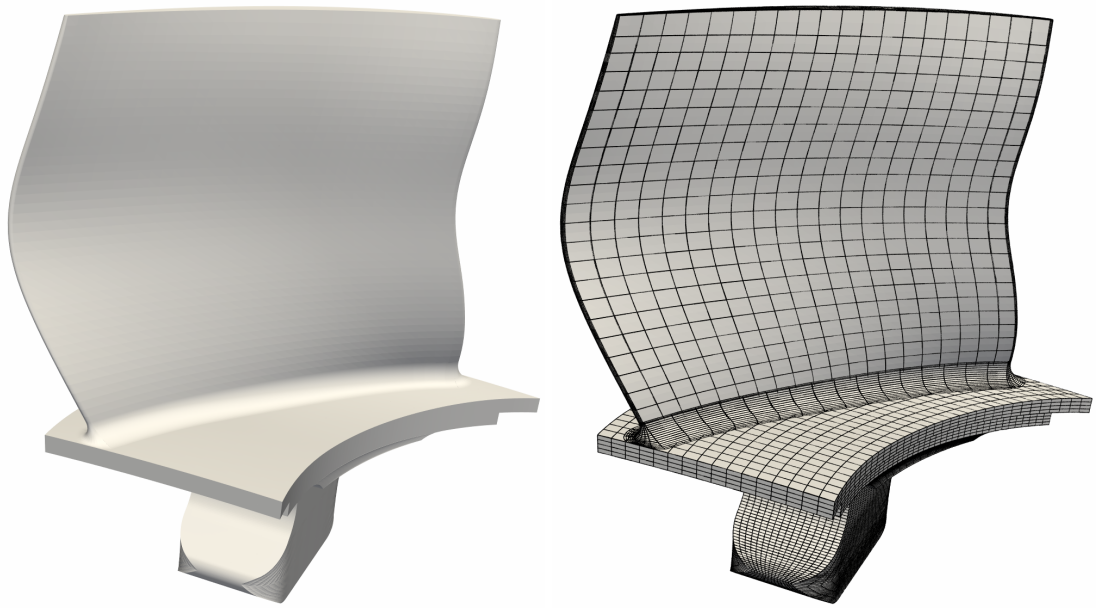
(**a**) Control mesh. (**b**) Physical space. (**c**) Jacobian determinant values.

Figure 5.9 – Parametrisation of the exact circular fillet patch.

Since it is described in a two-dimensional fashion, we extend this parametrisation for 3D cases by adding a third dimension. Since we seek to use this patch as an embedded solid, we settled for a straight linear interpolation between two cross-sections made of the previously depicted geometry. The final volumetric patch is presented in step 2 of figure 5.7. All the elementary components are now available to model a complete blade assembly, consisting in a blade, a platform and tenon, and an embedded fillet linking the blade to the platform part.

5.2.4 Complete assembly

Finally, the complete assembly is pictured in figure 5.10. It consists in a total of seven solid patches, and four embedded patches (which makes four hull objects and four embedded objects). Finally, the geometry consists of 61400 control points, while all patches are of degree two in all three parametric directions. Figure 5.11 pictures a portion of the blade wheel.



(a) Rendered view. **(b)** Geometry with parametric lines displayed.

Figure 5.10 – Final blade assembly.

5.3 Numerical results

In the following, we illustrate how the framework we developed throughout this work can be put together and implemented for the analysis of a complete blade model.

5.3.1 Configuration

5.3.1.1 Geometry

The geometry used for this case is the one presented in figure 5.10. The blade is modelled by a single patch, as described in section 3.1. The platform geometry, as described above, consists in six patches. The blade and platform are modelled using solid elements. Then, the fillet is modelled in four parts using the embedded solid element presented in section 4.1. Lastly, considering the coupling aspects of the model, let us specify that this geometry counts up to 17 interfaces. In particular, the platform has

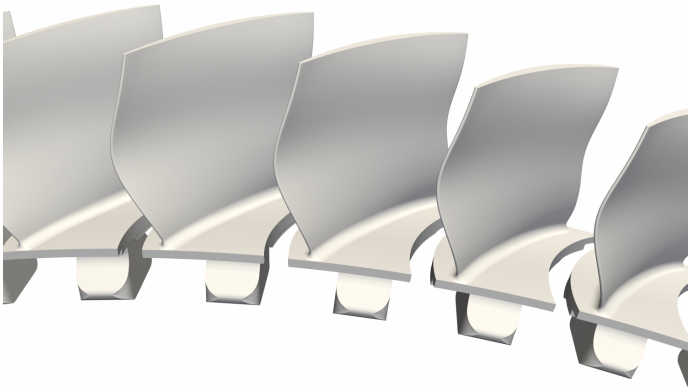


Figure 5.11 – Close-up of the blade wheel at the platform junctions.

to be "assembled" using the coupling approach presented in section 4.2, and so do the fillet parts.

As stated above, this model counts no less than 61400 control points, making a total of 169000 degrees of freedom for the numerical analysis – removing those associated with the hull objects, as they do not come into play for the analysis. As for the coupling problem stemming from such configuration, 8000 degrees of freedom are involved in the coupling problem.

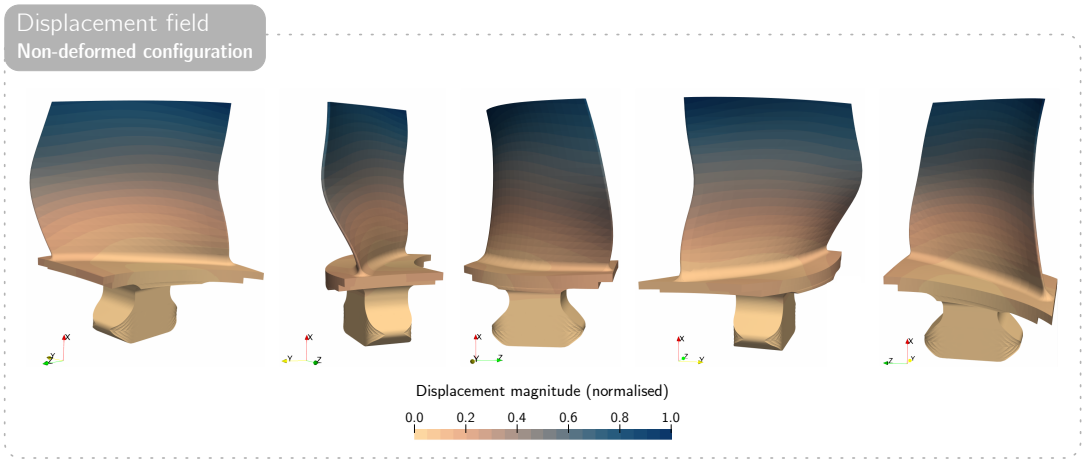
5.3.1.2 Boundary conditions and loading

Centrifugal force and surface pressure are the two loadings applied for this case. All solid patches are subject to the body force, with identical magnitude as presented in section 3.2.1. The surface pressure presented in the same section is used, and applied on the blade solely. As the blade geometry has been modified with respect to the one presented in figure 3.9, the repartition of the pressure field has been adapted. First, the blade itself does not present the same geometry and second, the pressure field is not applied on the filleted part of the assembly, thus justifying for further adjustments with respect to figure 3.9.

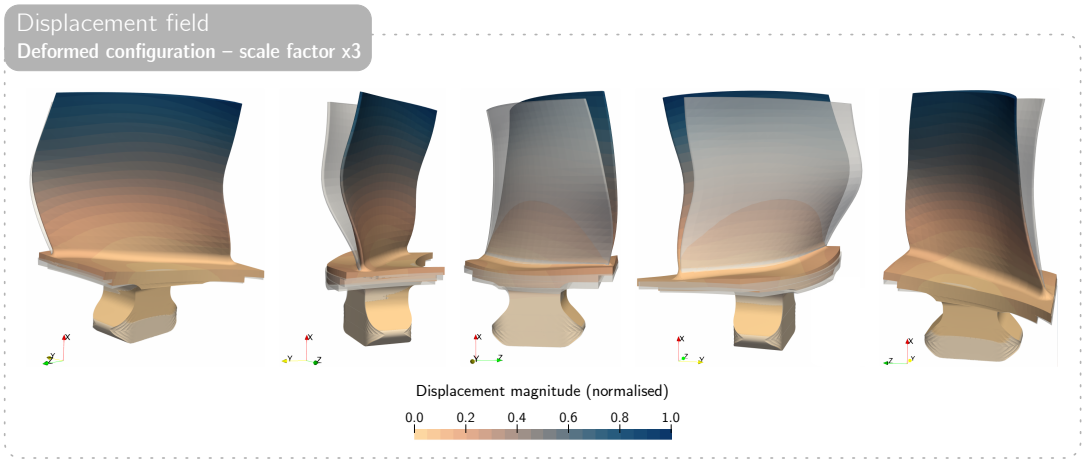
Regarding boundary conditions, the bottommost part of the tenon is kept fixed. It is not exactly identical to the industrial numerical set-up for this type of analysis, yet enables to recover the displacement and strain fields on the blade, which is the part we are interested in.

5.3.2 Mechanical analysis

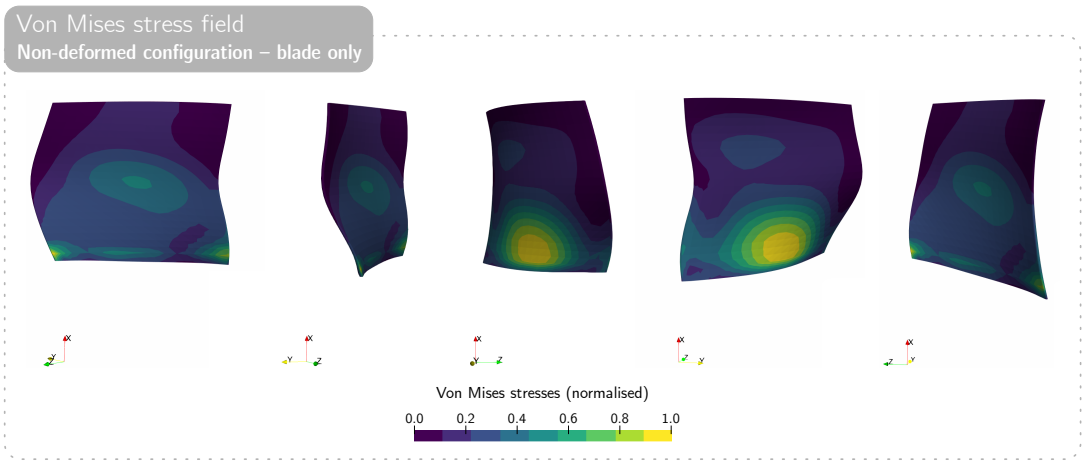
The results are presented in figure 5.12. More specifically, figures 5.12a and 5.12b picture the displacement field, while figure 5.12c presents the Von Mises stress field, plotted on the blade geometry only. One can note the displacement field is continuous, even when



(a) Displacement field: the displacement magnitude is plotted on the no-deformed geometry.



(b) Displacement field: the displacement is plotted on the deformed geometry, with a scale factor of 3. The non-deformed state is plotted in transparency.



(c) Stress field: the Von Mises stress field is plotted on the non-deformed geometry of the blade.

Figure 5.12 – Results for the analysis of the complete blade model.

crossing interfaces between patches, be it embedded or not. As for the stresses, they concentrate at the bottom of the blade. As one can note from the deformed configurations plotted in figure 5.12b, the blade bends under the effect of the centrifugal force, resulting in high efforts located at the foot of the blade.

For this case, the stiffness matrix is assembled in less than 1 minute, and the coupling matrix assembly takes less than 10 minutes. The time for solving, using the `splu` function from the sparse linear algebra python package `scipy.linalg` (Virtanen et al., 2020), is less than 40 minutes. For reference, let us precise the analysis has been run on 16 cores, using a processor of type Intel Xeon E5-2650 v4.

5.3.3 Perspectives for structural shape optimisation

Very promising results have been achieved on the complete blade assembly regarding mechanical analysis. Few more developments would be needed in order to reach one step further, by carrying out the shape optimisation of the assembled blade.

Apart from the complete geometry, an essential element to reach this goal is the expression of analytical sensitivities in the context of the embedded solid element. As succinctly introduced in section 2.4.1.2, IGA brings a favourable framework for expressing the derived quantities involved in gradient-based optimisation. Amongst these quantities appears the element formulation used for the analysis, which has a direct impact on the expression of the derivatives of the Jacobian, its determinant, as well as the derivatives of the strains and stresses. In particular, Hirschler et al. (2020) thoroughly give the main steps for the computation of these quantities in the case of solid-shell elements. Following the same steps, without using covariant and contravariant metrics, we were able to give the expression for those quantities in the case of the embedded solid element formulation. Appendix B gathers the analytical developments carried out in that context.

Concluding remarks

Modelling complex geometric structures in the context of IGA still remains a huge challenge, especially when considering volumetric parametrisations. Apart from B-rep design paradigms, one of the main obstacles is the lack of pre-processing tools. We specifically refer to the tedious procedure that has been set up in order to rebuild a trivariate platform and tenon model.

Still, these efforts have shown to be fruitful as we were able to run analysis on a complete blade assembly. This example is an achievement demonstrating the feasibility of the approaches developed in the previous chapters. As stated above, they offer interesting new perspectives for the structural shape optimisation of a complete blade assembly.

Chapter 6

Conclusion

This concluding chapter will summarise the key findings of the presented work, in relation with the research aims and questions considered at the beginning of the manuscript. The contributions will be discussed, while considering the limitations of our work. Opportunities for future research and closing remarks will finally be provided.

We sought to investigate the possibilities offered by IsoGeometric Analysis in the context of aircraft engine blade design, analysis and shape optimisation.

Several elements led to the choice of using IGA to address the research aims and questions, one of them being the smoothness and higher-order approximation capacities IGA brings. The possibility to express analytical sensitivities for gradient-based shape optimisation was another attractive feature we were interested in. Being able to provide a shape-optimised model that is compatible with CAD software environments seemed of high interest as well. In the following, we summarise the main findings and developments that have been carried out during this work.

Starting with the core subject of interest, namely compressor blades, we sought to propose a framework for reconstructing trivariate industrial blade models. Mechanical analysis and shape optimisation procedures have successfully been demonstrated on such models. Turning towards our final goal of analysing a complete blade assembly, numerical solutions have subsequently been investigated, and a two-pronged strategy has been established. Firstly, we proposed a new element formulation, the so-called *embedded solid element*. Implementing this element eventually offered the possibility to handle the geometric compatibility of various patches through shape modifications of a complex assembly. Secondly, the issue of non-conforming parametrisation at interfaces of multi-patch models is resolved by setting up a coupling method for non-overlapping sub-domains, thus enabling numerical analysis on such models. In addition, the proposed implementation of the coupling is valid for both classical and embedded solid elements. Finally, the numerical tools and strategies developed in the course of this work have been effectively applied to model a complete high-pressure compressor blade assembly.

Over the course of this work, research fields ranging from geometric modelling, mechanical analysis, finite element methods, optimisation, and coupling have been investigated. Elements from these fields were then combined in the development of our numerical approach. While this may not appear ground-breaking, we believe that assembling pieces of the puzzle from such varied academic and industrial fields constitutes in itself a noteworthy contribution. We also hope that by demonstrating the feasibility of the developed approach on an actual industrial geometry, we take a step towards better integration of new numerical methods into real-world applications.

Spinning bodies such as blades call for volumetric modelling, as does any object subject to body force. Consequently, the modelling and analysis of trivariates was a central topic of this study, in which it was shown that addressing volumetric modelling is not necessarily a straightforward task. Although there exists substantial literature dealing with isogeometric analysis of shells and plates, few papers tackle the case of volumetric parametrisations. Studies dealing with bivariate parametrisation are often generalisable to trivariates, yet it is not a trivial process to actually implement it.

Another issue in industrial applications was the compatibility of adjacent patches belonging to a same model through shape modifications, which can typically occur during the course of a shape optimisation process. We were able to propose an effective solution by formulating the embedded solid element. Its relevance to geometric modelling in the context of multi-patch models goes far beyond the specific application we studied. The same applies to the strategy we developed for the coupling of non-overlapping solids in the case of non-conformal parametrisations at interfaces.

Taking a step back from this work, we can say that narrowing the gap between academic research and real-world applications is itself a non-negligible task, that contributes to the permeability of both environments. Furthermore, we believe that contributing to an open-source code and open research, as we did in this work, is a point that should be highlighted.

Several aspects have not been tackled in this work. In the following we shall acknowledge some points that we did not investigate.

First, from an industrial point of view, a number of important parameters have not been considered. For instance, in the mechanical analysis of blades, we did not include the temperature loading in our simulations. Considering that the temperature reaches around $300°C$ for the high-pressure compressor blade, its effect is significant on the mechanical response of the assembly. As for shape modifications proposed for the shape optimisation of the airfoil part, more work would be needed in order to formulate more actual shape parametrisations. In fact, being able to formulate aerodynamics-related shape modifications would have been of great interest. Not taking this aspect into account limits the potential comparison of our work with industrial results.

On a more theoretical point of view, we emphasised the interest of using analytical sensitivities in the context of gradient-based shape optimisation using IGA. Incidentally, this is what led us to express the analytical sensitivities of the embedded solid element.

A limitation that is worth noting is that these expressions were achieved for the specific case of auto-adjoint problems. More developments are needed to handle cases that do not fall into this category. Furthermore, we proposed a weak coupling strategy for the simulation of multiple patches with non-conformal parametrisations at interfaces. This coupling strategy is based on point inversion and projection procedures, adding computational cost to the coupling problem. Last, we relied on widely available geometric algorithms for the generation of trivariate parametrisations. While those have proven to be efficient in our framework, more advanced solutions probably exist, among the geometry community for one.

By linking the contributions and limitations we highlighted, we can identify various prospects for future work.

To further investigate isogeometric shape optimisation of volumetric parametrisations, further studies could address the use of multi-start procedures in the case of gradient-based algorithms. Moreover, other algorithms such as the ones used through this work – namely Sequential Least Squares Programming – could be tested. Over the years, different local gradient-based algorithms have been used in the context of isogeometric structural shape optimisation, such as the Method of Moving Asymptotes or the Modified Method of Feasible Directions, and it could be interesting to investigate how they perform in our case.

The question of multi-patch coupling offers a wealth of possibilities. The first element we would like to bring to light is the possibility of using domain decomposition methods to reduce computation time. In fact, this aspect has been tackled by Hirschler et al. (2019a). They showed that there exists a natural link between FETI approaches for domain decomposition, and used the additional unknowns introduced to apply a mortar approach (*i.e.*, those introduced to define the Lagrange multipliers field) to formulate an interface problem. Another aspect that could be considered is the use of dual basis functions for the mortar method. As showed by Wohlmuth (2000), replacing the Lagrange multiplier space by a dual space facilitates the realisation of the matching condition at the interface. For instance, Dornisch et al. (2017) developed approximate dual basis functions for B-Spline and NURBS that enable an accurate mortar formulation in the context of isogeometric analysis. We believe this approach could lead to better results for the resolution of the coupling problem. Finally, in this work we specifically investigated the coupling of non-overlapping subdomains. Studying the possibility of coupling overlapping domains could be of interest, especially for the application of the interface between the blade and platform parts, as it would allow skipping the phase of computing the surface-to-surface intersection between the two parts.

Turning to parametrisation nesting, we believe this approach can be far more versatile. This aspect actually has already been shown by Antolin et al. (2019a), who used a similar concept as the embedded solid to run shape optimisation of micro-tiles in micro-structures, with an application to porous objects. To describe their work using the terminology in this work, they embedded a volumetric pattern (micro-tiles) in a pe-

riodic fashion inside a volumetric description of the considered structure. This approach enables adjusting the shape of the micro-tiles as well as the macro shape of the structure during the design and/or optimisation stages, and opens up a variety of exciting new possibilities. As for the embedded approach we developed, a possibility one could envision for the design of turbine blades is the positioning of cooling tunnels inside the bulk material.

From a broader perspective, one last consideration that should be mentioned is the need for more widely available solutions for IGA pre-processing. This work has shown that it is clearly possible to model complex industrial geometries without commercially developed tools, yet their existence would offer a more complete link between geometry and analysis, fulfilling the original goal of IsoGeometric Analysis.

Appendix A

Résumé étendu des travaux en français

Ce chapitre reprend de façon condensée les principaux éléments présentés dans le corps du manuscrit, en langue française. Après avoir introduit le sujet et les motivations sous-jacentes, nous donnons les éléments de contexte nécessaires, d'un point de vue académique et industriel. Puis, nous présentons notre méthodologie pour la construction d'un modèle de pale B-Spline volumique, et nous présentons un résultat d'optimisation de forme sur cette géométrie. Ensuite, la question de la modélisation et l'analyse d'un modèle d'aubage complet est considérée. Les outils numériques présentés sont utilisés pour réaliser l'analyse statique d'une géométrie complète, sous chargements industriels réels. Une conclusion dresse une vue d'ensemble finale et propose des pistes de recherche pour de futurs travaux.

Contents

A.1 Introduction

Dans ces travaux, nous nous plaçons dans le domaine la mécanique des structures numérique, en utilisant l'Analyse Isogéométrique. Nous appliquons cette méthode à la conception d'aubages de moteur d'avion. Ces travaux visent à démontrer l'intérêt d'utiliser des méthodes numériques différentes de celles utilisées actuellement pour la définition, l'analyse et l'optimisation de forme d'aubes de moteur d'avion.

Le nom de l'Analyse Isogéométrique reflète sa philosophie d'intégrer de façon fluide la définition géométrique et les procédures d'analyse numérique. Ces deux problématiques sont traditionnellement traitées séparément lors du processus de conception : la géométrie est définie en utilisant des logiciels de Conception Assistée par Ordinateur, tandis que l'analyse numérique est réalisée à l'aide de la Méthode des Éléments Finis. En pratique, il s'agit d'utiliser les mêmes fonctions mathématiques pour décrire la géométrie et pour servir de support pour résoudre l'analyse. Ainsi, les B-Splines Rationnelles Non-Uniformes (NURBS) sont utilisées comme bases de fonctions pour la MEF, ce qui permet de bénéficier des propriétés de ces fonctions pour l'analyse.

La représentation volumique est nécessaire pour les structures tournantes telles que les aubages, car les efforts les plus critiques agissant sur ces objets sont des efforts volumiques. Par ailleurs, une méthodologie spécifique doit être développée afin de générer des modèles d'aubages volumiques qui soient adaptés pour l'analyse et l'optimisation de forme. Pour permettre les modifications de forme, la pale doit être modélisée par un patch distinct du reste de l'aubage. Générer un modèle basé sur une géométrie industrielle complexe, et comportant plusieurs patchs, mène à des discrétisations hautement non-conformes, ce qui nécessite un traitement particulier d'un point de vue numérique.

Pour atteindre les objectifs mentionnés plus haut, plusieurs aspects doivent être adressés. En commençant par la génération d'un modèle volumique de pale, celui-ci doit à la fois présenter une description géométrique précise, tout en étant compatible pour l'analyse et pour l'optimisation de forme, ce qui suppose une discrétisation spatiale adaptée. En dehors de la pale, un congé de raccordement et une plateforme doivent être modélisés. La gestion du couplage de ces différents patchs est un autre objectif, en prenant en compte des paramétrages non-conformes de part et d'autre de l'interface. Modifier la forme de la pale au sein de l'assemblage complet suppose de garantir la compatibilité géométrique entre les patchs. Tous ces objectifs doivent être remplis en étant conciliables avec les outils numériques du service Méthodes du partenaire industriel.

Cette étude bénéficiera des propriétés de l'IGA, que sont un temps de conception réduit et une meilleure précision, pour l'analyse et l'optimisation forme d'aubes de moteur d'avion. Ces aspects avantageux peuvent mener à l'amélioration des performances et réduire l'impact environnemental des futurs moteurs d'avion. Par ailleurs, ce sujet contribue à réduire l'écart entre technologies académiques et industrielles. Ces travaux ont permis de contribuer au code IGA open-source YETI.

Dans la suite de ce chapitre, nous donnerons dans un premier temps des éléments de contexte permettant d'identifier certaines des problématiques ayant mené à notre sujet de recherche, tant en termes de concepts scientifiques que de contexte industriel. Ensuite, nous présenterons une première partie des travaux réalisés, qui ont consisté à reconstruire la géométrie d'une pale de façon trivariable en trois dimensions, puis à en réaliser l'analyse et l'optimisation de forme. Puis, nous présenterons un autre volet des travaux de recherche, cette fois-ci portant sur la construction et l'analyse d'un modèle d'aubage complet, comprenant une pale, une plateforme, et un congé de raccordement liant la première à la seconde.

A.2 Contexte

Proposer de nouvelles approches du point de vue de l'industriel et de celui de la recherche scientifique est un défi exigeant. Aussi, plusieurs notions doivent être introduites afin de planter le décor de cette chorégraphie complexe. Après avoir introduit les concepts essentiels en ce qui concerne les B-Splines et l'Analyse Isogéométrique, l'optimisation paramétrique de forme sera brièvement présentée. Enfin, le contexte industriel sera donné, en particulier en ce qui concerne la conception d'aubages de moteurs d'avions.

A.2.1 B-Splines et Analyse Isogéométrique

Plusieurs aspects concernant les B-Splines et leurs propriétés sont présentées ci-après. Le lecteur ou la lectrice intéressé·e peut se référer, *e.g.*, au *NURBS book* (Piegl and Tiller, 1996) pour de plus amples détails.

A.2.1.1 Introduction aux B-Splines

Une courbe B-Spline $\boldsymbol{\mathcal{C}}$ de degré p dans $\mathbb{R}^{2,3}$ est définie comme suit :

$$\boldsymbol{\mathcal{C}}(u) = \sum_{i=1}^{n} N_{i,p}(u)\boldsymbol{P}_i, \tag{A.1}$$

où les $\left\{N_{i,p}\right\}_{1\leq i\leq n}$ sont les n bases de fonctions B-Spline, avec $n \in \mathbb{N}$, les $\left\{\boldsymbol{P}_i\right\}_{1\leq i\leq n}$ sont n points de contrôle dans $\mathbb{R}^{2,3}$ et u est un paramètre sur un intervalle $U \subset \mathbb{R}$. Les bases de fonctions B-Spline sont calculées de façon récursive à l'aide de la formule de Cox-de Boor. Une fonction B-Spline est définie dans l'espace paramétrique par son vecteur de nœuds (*knot vector*), qui est une liste ordonnée de paramètres. Le nombre de nœuds est lié au degré p et au nombre de points de contrôle n de la courbe. Si le *knot vector* présente des valeurs uniformément espacées, alors il est dit *uniforme* ; dans le cas contraire, il est dit *non-uniforme*.

Pour des approximations multidimensionnelles, les bases de fonctions sont définies comme un produit tensoriel de fonctions unidimensionnelles. Par exemple, étant donné

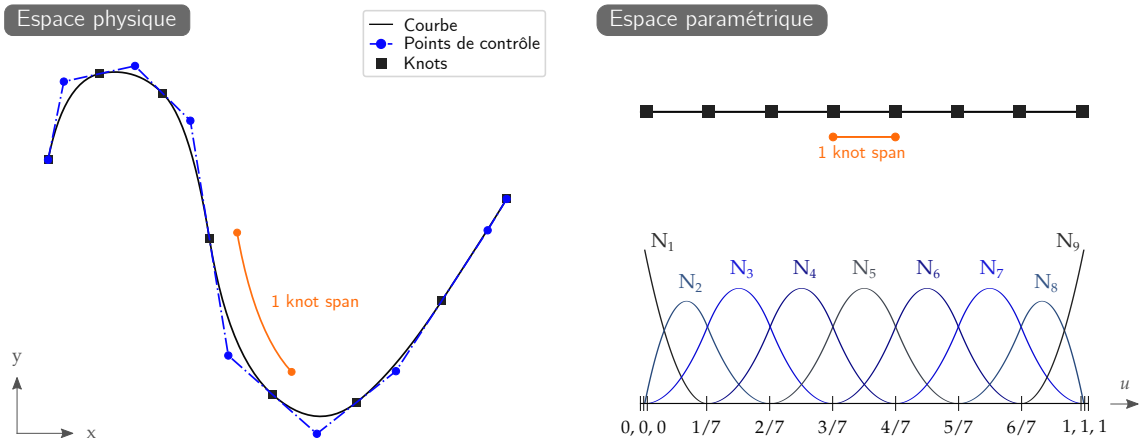


Figure A.1 – Espaces physique et paramétrique pour une courbe B-Spline quadratique dans $\mathbb{R}^2$. Le vecteur de nœuds associé est défini par $\left\{0, 0, 0, \frac{1}{7}, \frac{2}{7}, \frac{3}{7}, \frac{4}{7}, \frac{5}{7}, \frac{6}{7}, 1, 1, 1\right\}$.

un filet de contrôle $\left\{\boldsymbol{P}_{i,j,k}\right\}_{i,j,k}$ de points dans $\mathbb{R}^3$, les degrés p, q, et r, et les intervalles U, V, et W avec les vecteurs de nœuds associés, un solide B-Spline est définit sur $U \times V \times W$ comme suit :

$$\boldsymbol{\mathcal{S}}(u,v,w) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} N_{i,p}(u) M_{j,q}(v) L_{k,r}(w) \boldsymbol{P}_{i,j,k}. \tag{A.2}$$

Un aspect important à prendre en compte en ce qui concerne la formulation d'une géométrie B-Spline est la différenciation entre les espaces *paramétrique* et *physique*. Le premier est l'espace où les *knots* sont définis, tandis que le second est l'espace où les points de contrôle s'étendent (et donc l'espace dans lequel la courbe est représentée). Les *knots* délimitent les *knot spans* dans l'espace paramétrique, qui peuvent être vus comme des éléments au sens des éléments finis. Cependant, un point important à souligner est qu'un vecteur de nœuds présentant des valeurs de paramètres uniformément réparties ne génèrera pas nécessairement des éléments régulièrement espacés dans l'espace physique, comme le montre la figure A.1.

A.2.1.2 Analyse Isogéométrique

Introduite par Hughes et al. (2005), l'Analyse Isogéométrique est une méthode numérique ayant pour objectif d'améliorer le lien entre description géométrique et analyse. Elle repose sur l'utilisation de fonctions B-Splines ou NURBS, à la fois pour représenter la géométrie et pour réaliser l'analyse. Ainsi, comme dans le cas de l'analyse par Éléments Finis, le concept isoparamétrique est invoqué. La différence fondamentale entre les deux approches est que l'IGA conduit à une représentation exacte de la géométrie pour le calcul des inconnues durant l'analyse, là où l'analyse par EF classique ne fait qu'approximer la géométrie. Une autre propriété intéressante de l'IGA réside dans sa meilleure précision pour le calcul, ce qui est en partie dû à la plus grande continu-

ité des fonctions B-Splines par rapport à leurs homologues polynomiaux par exemple. Les fonctions B-Spline présentent également d'autres propriétés intéressantes dans le cadre de l'IGA, comme des procédures de raffinement efficaces. De plus amples détails peuvent être trouvés dans le livre de Cottrell et al. (2009).

A.2.2 Optimisation paramétrique de forme

Dans ces travaux, nous utilisons des algorithmes à gradient pour réaliser l'optimisation de forme de pales. Nous introduisons dans cette section quelques concepts utiles pour la suite de l'exposé en ce qui concerne l'optimisation paramétrique.

A.2.2.1 Problème d'optimisation

Formellement, étant donné un *espace de design* d-dimensionnel $\mathbb{X} \subset \mathbb{R}^d$ et une *fonction objectif* $f : \mathbb{X} \to \mathbb{R}$ à minimiser, un problème d'optimisation paramétrique peut être défini comme suit :

$$\text{Trouver } x^\star = \underset{x \in C}{\operatorname{argmin}}\, f(x), \tag{A.3}$$

où $C = \{x \in \mathbb{X} \,|\, c_i(x) \leq 0,\, e_j(x) = 0,\, (i,j) \in \mathcal{I} \times \mathcal{J}\}$ est un sous-domaine admissible de $\mathbb{X}$, définit par des contraintes à valeurs réelles définies sur $\mathbb{X}$, $(c_i)_{i\in\mathcal{I}}$ étant des contraintes d'inégalité et $(e_j)_{j\in\mathcal{J}}$ des contraintes d'égalité, où $\mathcal{I}$ et $\mathcal{J}$ sont des ensembles d'indices.

Plusieurs méthodes peuvent être utilisées pour résoudre le problème d'optimisation (A.3). Le lecteur ou la lectrice intéressé·e peut se référer au livre de Nocedal and Wright (2006) pour une revue complète en ce qui concerne l'optimisation à gradient. Les exemples numériques présentés plus loin ont été réalisés à l'aide d'un algorithme de type Sequential Quadratic Programming, disponible dans la méthode `SLSQP` du module `optimize` de la libraire open-source `scipy` pour l'optimisation non-linéaire. Cet algorithme à gradient peut prendre en compte des domaines d'optimisation définis par des contraintes de bords ainsi que des contraintes non-linéaires d'égalité et d'inégalité.

A.2.2.2 Analyse de sensibilités

Dans le cas de l'optimisation à gradient, exprimer les gradients de la fonction objectif et des contraintes du problème d'optimisation par rapport aux variables de design est une étape clé. Cette étape peut mener à des développements pénibles ; quand les gradients ne peuvent pas être exprimés de façon analytique, une possibilité est d'utiliser des schémas par différences finies, ou une méthode de différentiation automatique. Ceci étant, l'IGA fournit un cadre qui permet d'obtenir des expressions générales et compactes des sensibilités analytiques, pour un coût de calcul raisonnable, dans le cas des problèmes autoadjoints. Cet aspect a été détaillé par Qian (2010) et Fußeder et al. (2015), et plus récemment par Hirschler et al. (2020).

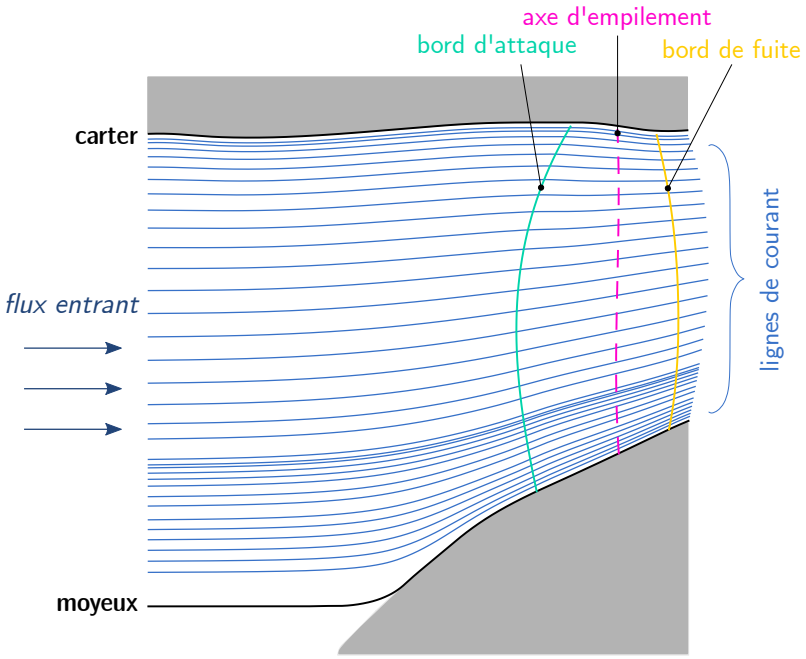


Figure A.2 – Lignes de courant bidimensionnelles dans un compresseur axial.

A.2.3 Conception d'aubages de moteur d'avion

Le contexte industriel est indispensable afin de motiver les raisons de ces travaux, les questions de recherche associées ainsi que les solutions proposées. Nous en brossons un portrait rapide dans ce qui suit.

A.2.3.1 Description géométrique d'aubes de compresseur axiales

Pour la conception d'aubages, une pratique courante dans l'industrie est de considérer les aspects aérodynamiques et de mécanique des structures de façon séparée. Deux équipes spécialisées travaillent donc en parallèle, ce qui donne lieu à un processus itératif afin d'arriver à une description finale de l'aubage. Chaque équipe modifie tour à tour le design afin de se rapprocher de ses propres objectifs.

La géométrie d'une pale est en général définie par sa surface extérieure, qui est décrite par une série de section (ou *profils*), qui sont empilés le long de l'axe de celle-ci. La définition initiale de la géométrie de la pale est définie par des critères aérodynamiques. Pour cela, les aérodynamicien·ne·s se basent sur le comportement du flux dans le moteur, qui est représenté grâce à des *lignes de courant*, comme représenté en figure A.2. La géométrie de la pale est définie de sorte à guider le flux correctement pour obtenir les performances désirées. Une fois la géométrie des profils déterminée, un *axe d'empilement* est défini. Ce dernier permet de créer la surface extérieure de la pale par la création d'une surface par section (*loft surface*).

A.2.3.2 Dimensionnement mécanique

Dans ces travaux, nous nous intéressons au volet mécanique des structures du processus. Nous donnons ci-après quelques éléments de dimensionnement mécanique d'aubages de moteur d'avion. Nous nous concentrons sur les principales sollicitations qui entrent en jeu d'un point de vue de la mécanique des structures.

L'effort centrifuge constitue le chargement prépondérant influant sur le comportement mécanique des aubages. Une pression de surface induite par le flux d'air contribue également au chargement statique de la pale. D'autre part, des chargements thermiques entrent en jeu, et prennent part aux efforts agissant sur la pale. Enfin, du point de vue dynamique, les vibrations induites par le fonctionnement du moteur constituent un objet d'étude important. Les différentes vitesses de fonctionnement du moteur induisent différentes fréquences d'excitation, et les aubages doivent remplir leur fonction à tous les stades de vol. Aussi, leur comportement vibratoire est étudié en détail.

A.3 De la construction à l'optimisation de forme d'un modèle de pale seule

A.3.1 Construction d'un modèle de pale solide

Comme énoncé dans la section A.2.3, les pales nécessitent une modélisation volumique afin d'étudier les effets des efforts centrifuges sur celles-ci. L'enjeu de cette partie des travaux est de générer un modèle de pale B-Spline volumique qui soit à la fois fidèle à la géométrie initialement définie, tout en présentant une discrétisation qui soit convenable pour l'analyse mécanique et l'optimisation de forme.

Pour construire un modèle de pale volumique, nous considérons la géométrie homéomorphe à un cube. Cette hypothèse va guider les étapes nécessaires à la reconstruction volumique, que nous résumons ici :

1. Créer l'ensemble des courbes B-Spline formant les profils de pale empilés.
2. Pour chaque courbe, identifier quatre points formant les coins, et séparer les courbes en ces points. Cette étape permet de générer quatre groupes de courbes : bord d'attaque, intrados, bord de fuite, et extrados.
3. Ajuster le paramétrage de toutes les courbes créées.
4. Créer quatre surfaces à partir des quatre groupes de courbes créés à l'étape **2** (générant les surfaces contours).
5. Créer les deux surfaces manquantes, *i.e.*, les surfaces dessus et dessous par interpolation bilinéaire des courbes formant le premier et le dernier profil.
6. Calculer les coordonnées des points de contrôle du volume par combinaison linéaire des points de contrôle des six surfaces extérieures.

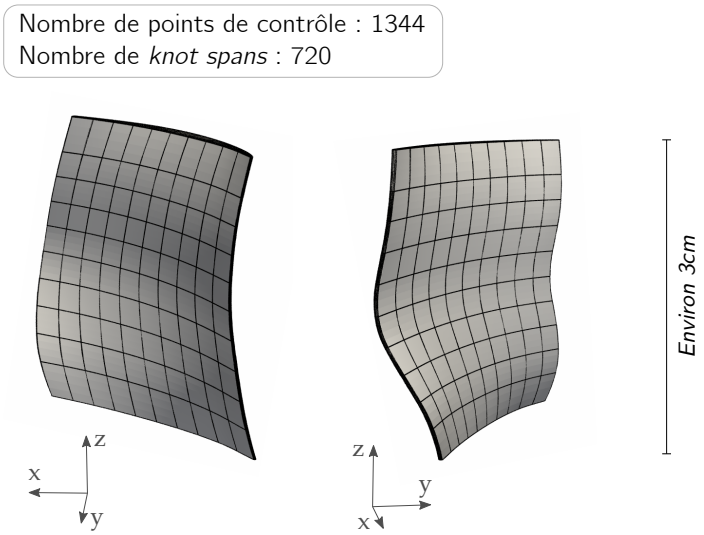


Figure A.3 – Modèle B-Spline trivariable d'une pale de compresseur haute pression, avec lignes paramétriques.

L'étape **3** est une étape essentielle pour assurer la régularité du paramétrage du modèle final. Nous détaillons ci-dessous les principaux éléments permettant de réaliser cette étape.

Reparamétrage des courbes Dans un premier temps, nous calculons une séquence de paramètres permettant d'obtenir des points régulièrement espacés dans l'espace physique en évaluant la courbe en ces paramètres. Pour cela, nous utilisons l'algorithme `UniArcLength` de Hernández-Mederos and Estrada-Sarlabous (2003), qui correspond exactement à cette attente. Dans cet algorithme, les valeurs paramètres sont calculées en utilisant un paramétrage par longueur d'arc. Ensuite, les points régulièrement espacés dans l'espace physique sont utilisés pour approximer une courbe, ici grâce à l'algorithme d'approximation présenté dans le livre de Piegl and Tiller (1996). Utiliser une approximation nous permet de choisir le degré et le nombre de points de contrôle de la courbe résultante, qui possède alors un paramétrage harmonieux dans l'espace physique. Une approche similaire est utilisée pour garantir un paramétrage compatible des surfaces extérieures aux étapes **4** et **5**.

Création du modèle final Finalement, ayant construit des surfaces extérieures compatibles en termes de paramétrage, il est possible de calculer la position des points de contrôle permettant de générer un modèle volumique, comme présenté en figure A.3.

A.3.2 Optimisation de forme d'un modèle de pale seule

Paramétrage de forme La paramétrage de forme consiste à définir de quelle façon les points de contrôle de la pale vont être modifiés au cours du processus d'optimisation de forme. Nous présentons ici un paramétrage de forme permettant de modifier l'épaisseur de la pale. Les variables de design sont donc liées à la position des points de contrôle,

ou plus particulièrement leur déplacement relatif, afin de modifier l'épaisseur de la pale. Les contraintes sur les variables de design sont formulées de façon à ne pas obtenir une épaisseur trop faible, par rapport à des contraintes de fabrication, ni trop élevée, afin de ne pas influer de façon trop importante sur les performances aérodynamiques de la pale.

Analyse modale Comme énoncé dans le paragraphe A.2.3, le comportement vibratoire des aubages doit être étudié en détail afin de prévenir toute interférence entre les modes propres de celles-ci et les différentes fréquences d'excitation ayant lieu au cours du fonctionnement du moteur. Une façon répandue de représenter les interactions entre les différentes fréquences en jeu est d'utiliser un diagramme de Campbell, aussi appelé diagramme d'interférence (Campbell, 1924).

Dans notre cas, nous traçons dans une premier temps les dix premiers modes propres de la pale en fonction de la vitesse de rotation. La vitesse de fonctionnement de croisière du moteur est ensuite représentée. Puis, des plages de fréquence à éviter sont identifiées : l'objet de l'optimisation de forme sera alors d'agir sur les fréquences propres de la pale afin que celles-ci ne se trouvent pas dans les plages interdites identifiées. Pour ce faire, une fonction objectif est formulée de façon à pénaliser les plages de fréquences à éviter. Ainsi, la minimisation de cette fonction conduit à l'exclusion de ces fréquences. La fonction contrainte utilisée ici vise à garantir une modification de volume comprise entre plus et moins 10% du volume initial de la pale.

Résultats Les résultats de l'optimisation de forme sont présentés en figures A.4 et A.5. L'évolution des valeurs de fréquences propres en fonction des itérations est tracée, de même que l'évolution des valeurs des fonctions objectif et contrainte au cours des itérations en figure A.4. Enfin, la carte d'épaisseur (normalisée) de la pale est représentée en figure A.5. La variation d'épaisseur est également représentée, afin de mieux identifier les zones de forte modification. On peut noter que la fonction objectif atteint un minimum après 4 itérations, et que les itérations suivantes sont nécessaires afin de satisfaire la contrainte sur le volume final de la pale.

Dans cette section, nous avons tout d'abord donné les étapes principales permettant de reconstruire une géométrie de pale trivariable en trois dimensions, à partir d'une définition sous forme de sections empilées. Puis, nous avons proposé un paramétrage de forme, associé à des fonctions objectif et contrainte, afin de réaliser un calcul d'optimisation de forme sur cette pale. Dans la section suivante nous décrivons les solutions que nous proposons – et mettons en œuvre – pour la construction et l'analyse d'un modèle complet d'aubage de compresseur.

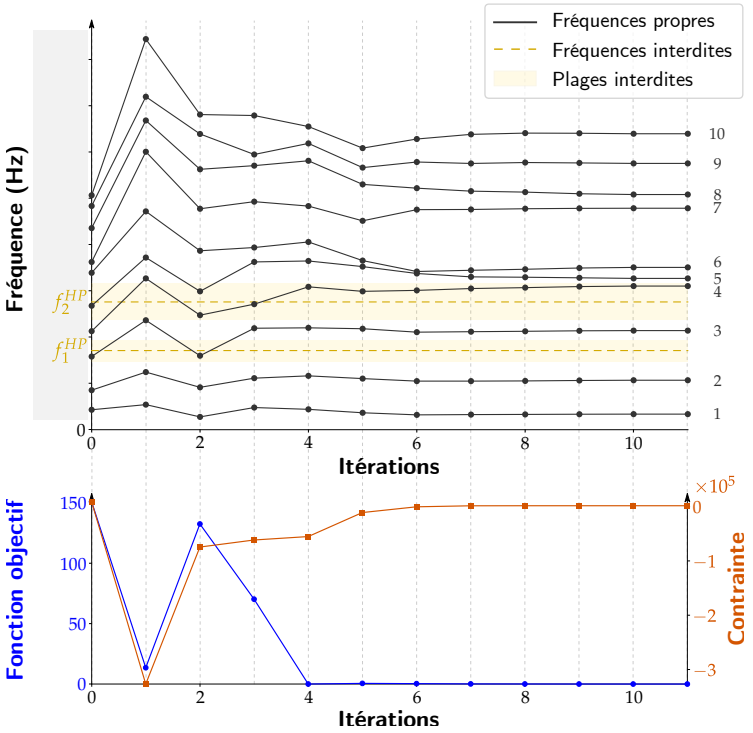


Figure A.4 – Historiques de l'optimisation de la pale de compresseur haute pression. Dessus : évolution des dix premières fréquences propres au cours de l'optimisation. Dessous : évolution de la fonction objectif et de la fonction contrainte en fonction des itérations.

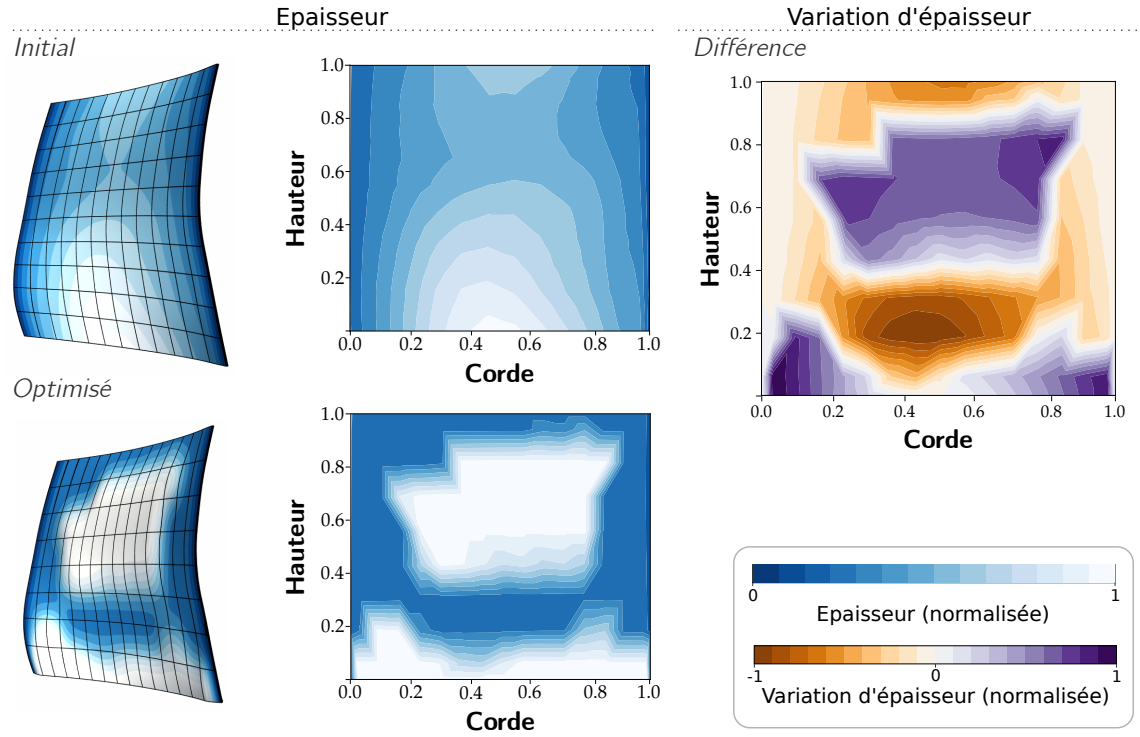


Figure A.5 – Variations d'épaisseur avant et après optimisation sur la pale de compresseur HP. Gauche : cartes d'épaisseur de la pale, tracées sur la géométrie et dans un espace aplati. Droite : variations d'épaisseur entre les deux états, représentées dans un espace aplati.

A.4 Construction et analyse d'un modèle complet d'aubage

Comme énoncé dans l'introduction de ce chapitre, nous cherchons à réaliser l'optimisation de la forme d'un seul patch NURBS (la pale), faisant partie d'un ensemble complexe de plusieurs patchs (composé de la plateforme et du congé de raccordement). La géométrie légendée est présentée en figure A.6.

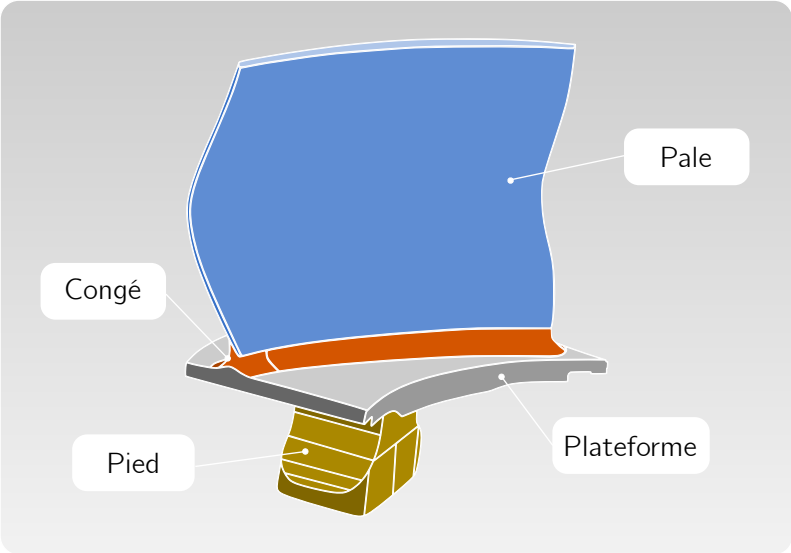


Figure A.6 – Exemple stylisé d'un aubage de compresseur Haute Pression.

Deux problématiques se posent dans ce cas. Premièrement, du point de vue géométrique : notre objectif est de faire varier la forme de la pale seulement, tout en garantissant la compatibilité géométrique de ce patch avec le congé de raccordement – au cours des modifications de forme induites lors d'un processus d'optimisation de forme par exemple. Deuxièmement, d'un point de vue de l'analyse : réaliser l'analyse et l'optimisation de forme de structures complexes requiert un traitement supplémentaire par rapport à des structures composées d'un seul patch.

A.4.1 Modèle géométrique complexe

La modification de la géométrie d'un assemblage de plusieurs patchs pose la question de la compatibilité géométrique entre ceux-ci. Ici, nous illustrons cette difficulté, puis nous proposons des éléments de réponse à cette problématique.

A.4.1.1 Modifications de forme

Cette question surgit dans le cadre particulier de l'application que nous visons. Faisant référence à la figure A.6, notre objectif final est de pouvoir modifier la géométrie de la pale en garantissant de manière concomitante la compatibilité géométrique du congé avec le reste de l'assemblage. Plus particulièrement, ces modifications de forme auraient lieu lors d'un processus d'optimisation de forme : la forme de la pale est mise à jour

à chaque itération. Intégrer une modification de géométrie à la volée constituerait un avantage conséquent d'un point de vue industriel. En effet, le retour vers les logiciels de Conception Assistée par Ordinateur (CAO) après des modifications de forme basées sur les méthodes numériques est une question à part entière.

A.4.1.2 *Free Form Deformation*

L'approche que nous avons mise en place s'inspire du concept désormais bien connu de *Free Form Deformation* (FFD) (Sederberg and Parry, 1986, Coquillart, 1989). Cette approche a d'abord été utilisée en infographie et modélisation géométrique, puis a été utilisée pour du paramétrage de forme pour de l'optimisation de forme (Duvigneau, 2006, Manzoni et al., 2012, Koshakji et al., 2013). Le principe de la FFD est d'imbriquer une représentation géométrique dans un paramétrage volumique peu complexe. Le volume d'accueil sera qualifié d'*enveloppe* par la suite. L'intérêt d'imbriquer une géométrie complexe dans une autre, plus simple, est direct pour celui ou celle qui cherche à modifier la forme globale d'un assemblage. Le paramétrage simple du volume d'accueil rend possible les modifications de forme globales sans avoir à manier un nombre de points de contrôle trop important. Cet aspect est illustré sur la figure A.7.

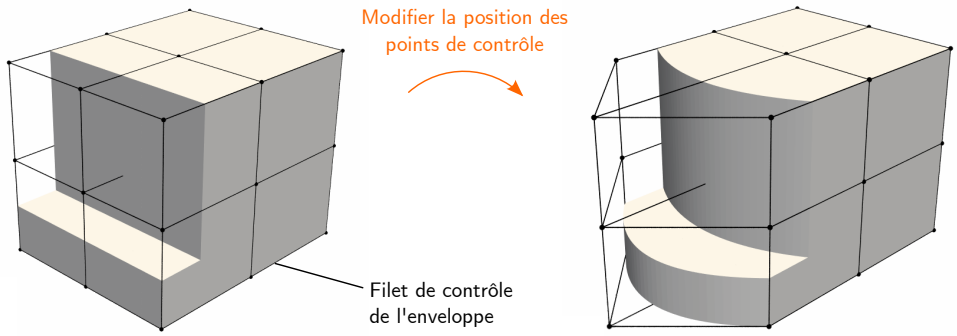


Figure A.7 – L'approche *free-form deformation*. La modification de la position des points de contrôle de l'enveloppe entraîne la modification des objets imbriqués.

En plus de l'attrait que représente le maniement d'un nombre peu important de points de contrôle pour la modification de forme, soulignons que la compatibilité géométrique des patchs adjacents est automatiquement garantie.

A.4.2 Analyse de structures complexes

A.4.2.1 Élément solide immergé

L'idée au cœur de l'imbrication de paramétrages est de décrire la géométrie de chaque patch dans l'espace paramétrique de l'enveloppe. L'imbrication de paramétrages comporte ainsi une étape de transformation supplémentaire par rapport à une géométrie

NURBS "classique". Cela se traduit par la composition des fonctions de base NURBS du patch immergé et de l'enveloppe, respectivement. Nous résumons ci-après les différents espaces qui entrent en jeu pour la composition :

1. L'espace paramétrique de l'objet immergé (décrit par les variables θ_1, θ_2, θ_3) ;
2. L'espace paramétrique de l'enveloppe (décrit par les variables ξ_1, ξ_2, ξ_3), qui peut aussi être vu comme l'espace physique de l'objet immergé ;
3. L'espace physique de l'enveloppe (décrit par les variables x, y, z), qui est le lieu de la composition de fonctions.

Description mathématique Les descriptions mathématiques des objets sont données ci-après. L'enveloppe $\mathbf{V}_H$ est définie comme un volume NURBS :

$$\mathbf{V}_H\left(\xi_1,\, \xi_2,\, \xi_3\right) = \sum_{ijk} N_{ijk}\left(\xi_1,\, \xi_2,\, \xi_3\right) \mathbf{Q}_{ijk}, \tag{A.4}$$

où les N_{ijk} sont les fonctions NURBS trivariables et $\mathbf{Q}_{ijk}$ les points de contrôle associés. Le solide immergé $\mathbf{V}_P$ est définit par :

$$\mathbf{V}_P\left(\theta_1,\, \theta_2,\, \theta_3\right) = \sum_{abc} R_{abc}\left(\theta_1,\, \theta_2,\, \theta_3\right) \mathbf{P}_{abc}, \tag{A.5}$$

où les R_{abc} sont les fonctions NURBS trivariables et $\mathbf{P}_{abc}$ les points de contrôle associés. Finalement, la composition de ces deux descriptions peut s'écrire comme le volume $\mathbf{V}$ suivant :

$$\mathbf{V}\left(\theta_1,\, \theta_2,\, \theta_3\right) = \mathbf{V}_H\left(\mathbf{V}_P\left(\theta_1,\, \theta_2,\, \theta_3\right)\right) = \sum_{ijk} N_{ijk} \left(\sum_{abc} R_{abc}\left(\theta_1,\, \theta_2,\, \theta_3\right) \mathbf{P}_{abc} \right) \mathbf{Q}_{ijk}. \tag{A.6}$$

Choix de l'espace d'approximation pour le calcul des déplacements Étant donné que la composition de NURBS est utilisée, le concept isoparamétrique n'est pas forcément applicable dans le cas de l'élément solide immergé. L'espace solution pour la discrétisation du champ de déplacement doit être précisé, les deux choix étant d'utiliser la discrétisation de l'enveloppe ou bien celle du solide immergé. Dans ces travaux, c'est la discrétisation du solide immergée qui est utilisée. Cette approché évite le risque d'une grande proportion de vide dans le paramétrage, qui pourrait mener à des systèmes mal conditionnés. Ainsi, le paramétrage de l'enveloppe n'est alors qu'un outil intermédiaire permettant de façonner et modifier la géométrie. Un autre aspect engageant de cette approche est qu'elle est plus directe à implémenter lors de l'assemblage de la matrice de raideur élémentaire.

A.4.2.2 Couplage

Utiliser l'imbrication de paramétrage n'écarte pas la problématique résidant dans l'analyse de structures composées de plusieurs patchs NURBS. La question du couplage de do-

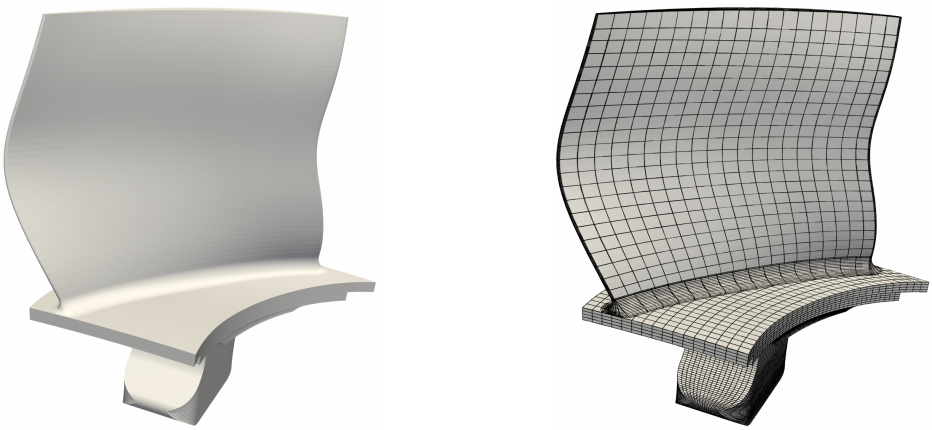
maines se pose nécessairement lorsqu'on s'attèle à la modélisation de structures complexes. En effet, rien n'indique que les différents patchs présentent une discrétisation conforme dans le cas général. Dans le cas d'une interface non-conforme, les conditions permettant de raccorder les domaines Ω_1 et Ω_2 situés de part et d'autre de celle-ci sont imposées faiblement. Nous nous plaçons dans le cas où les domaines considérés sont solides, et où il n'y a pas de recouvrement. L'interface entre les domaines est alors décrite par la surface Γ, et la condition de raccord s'écrit :

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{sur } \Gamma, \tag{A.7}$$

qui correspond à la continuité des déplacements à l'interface. Pour imposer cette condition faiblement, plusieurs méthodes numériques existent. Citons par exemple les méthodes de pénalisation, les méthodes utilisant les multiplicateurs de Lagrange, ou encore les méthodes de Nitsche (Apostolatos et al., 2014). Dans ces travaux, la méthode par multiplicateurs de Lagrange est utilisée, avec un formalisme de type *mortar*.

A.4.3 Un cadre approprié pour la modélisation et l'analyse d'un modèle d'aubage complet

Nous illustrons la mise en œuvre de l'approche immergée présentée plus haut, associée au couplage de domaines. L'objectif de cet exemple est de construire, et analyser, un modèle d'aubage complet. Le géométrie est représentée en figure A.8.



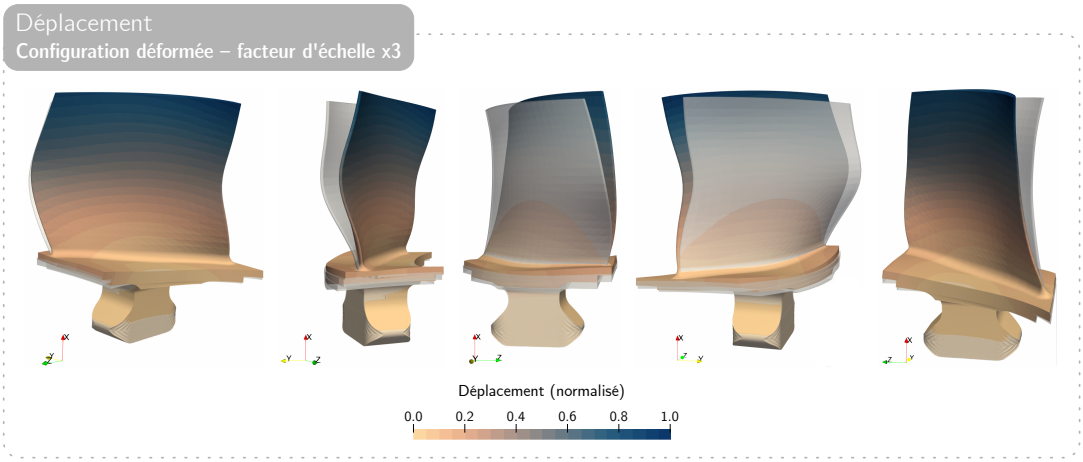
(a) Vue rendue. **(b)** Affichage des lignes paramétriques.

Figure A.8 – Géométrie de l'assemblage final.

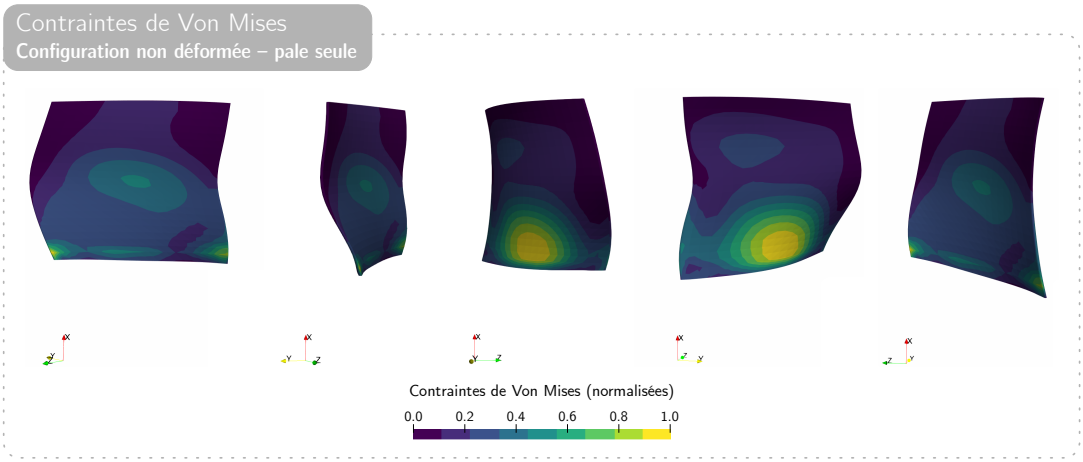
Géométrie du modèle La géométrie de la pale a été construite en utilisant la méthodologie présentée plus haut. La plateforme a été modélisée grâce au logiciel Rhino® (McNeel, 2021). L'élément solide immergé a été utilisé afin d'obtenir le congé de raccordement.

Condition aux limites et chargements Les chargements appliqués sur l'aubage sont un effort centrifuge correspondant à une vitesse de croisière du moteur, ainsi qu'une pression aérodynamique appliquée sur la pale. Les portées latérales du pied sont bloquées.

Résultats Les résultats du calcul statique sont présentés en figure A.9 . On peut noter la bonne continuité des champs de déplacement aux interfaces entre les différents patchs. La pale a tendance à fléchir sous l'effet des chargements, résultant en des concentrations de contraintes en pied de pale.



(a) Champ de déplacement: les déplacements sont tracés sur la géométrie déformée, avec un facteur d'échelle de 3. La configuration non-déformée est tracée en transparence.



(b) Champ de contraintes: les contraintes de Von Mises sont tracées sur la pale non-déformée.

Figure A.9 – Résultats pour l'analyse du modèle d'aubage complet.

A.5 Conclusion

Ces travaux visaient à étudier les possibilités offertes par l'Analyse IsoGéométrique dans le contexte de la conception, de l'analyse et de l'optimisation de forme des aubes de moteurs d'avion.

Dans un premier temps, nous avons proposé un cadre pour reconstruire des modèles d'aubes de compresseur par un paramétrage trivarié. Des procédures d'analyse mécanique et d'optimisation de forme ont été réalisées avec succès sur de tels modèles. En ce qui concerne l'analyse d'un aubage complet, des solutions numériques ont ensuite été étudiées et une stratégie en deux volets a été établie. Tout d'abord, nous avons proposé un élément solide immergé, permettant de traiter la compatibilité géométrique de plusieurs patchs au cours de modifications de forme d'un assemblage complexe. Deuxièmement, le problème du paramétrage non conforme aux interfaces a été résolu par la mise en place d'une méthode de couplage pour les sous-domaines sans recouvrement. Enfin, les outils et stratégies numériques développés au cours de ce travail ont été appliqués à un assemblage complet d'aubage de compresseur haute pression.

Au cours de ces travaux, des domaines allant de la modélisation géométrique à l'analyse mécanique, en passant par les méthodes éléments finis, l'optimisation et le couplage, ont été étudiés. Rassembler ces briques issues des milieux académiques et industriels constitue en soi une contribution. Nous espérons qu'en démontrant la faisabilité de l'approche développée sur une géométrie industrielle réelle, nous contribuons à une meilleure intégration des nouvelles méthodes numériques dans les applications industrielles. La modélisation et l'analyse de géométries trivariées était un sujet central de cette étude, dans laquelle il a été montré que la prise en compte de la modélisation volumique n'est pas nécessairement une tâche simple.

Nous pouvons identifier différentes perspectives pour les travaux à venir. Afin d'approfondir l'optimisation de forme isogéométrique, d'autres études pourraient porter sur l'utilisation de procédures de type *multi-start* dans le cas d'algorithmes à gradient, et d'autres algorithmes que ceux utilisés dans le cadre de ce travail pourraient être testés. En ce qui concerne le couplage multi-patch, la possibilité d'utiliser des méthodes de décomposition de domaine pour réduire le temps de calcul nous semble prometteuse. Traiter le couplage de domaines avec recouvrement pourrait également être intéressant. Pour ce qui est de l'approche immergée, nous pensons que cette approche peut être beaucoup plus polyvalente, d'un point de vue de la modélisation géométrique en particulier. D'un point de vue plus général, nous mettons en avant le besoin de solutions plus largement disponibles pour le pré-traitement de l'IGA. Ces travaux ont montré qu'il est possible de modéliser des géométries industrielles complexes sans outils développés commercialement, pourtant leur existence offrirait un lien plus complet entre la géométrie et l'analyse, répondant ainsi à l'objectif initial de l'Analyse IsoGéométrique.

Appendix B

Differentiation of the embedded solid element formulation

Using analytical sensitivities in the case of auto-adjoint problems is an advantage brought by isogeometric shape optimisation, which we emphasised in the course of this work. As mentioned in section 4.1, one motivation for the introduction of the so-called embedded solid element was to be able to use it in the case of shape optimisation. Even though we did not perform any shape optimisation example where an embedded entity is subject to shape optimisation, we believe that providing the necessary elements to do so is an important aspect.

Hence, in this appendix we propose to set out the main analytical expressions that come into play for the differentiation of the embedded solid element formulation. Starting with linear elasticity, we give the general framework for a classical isogeometric solid element. Then, we turn to sensitivity analysis strictly speaking, in order to motivate our choices and to give an overview of the derived quantities coming into play. Last, we express those derived quantities, first in a general case, and then specifically in the case of the embedded solid and for the hull object.

B.1 Linear elasticity

In the following section we summarise some key elements about linear elasticity. We emphasise we address the case of the *classical* isogeometric solid elements for the discrete expressions. The specific case of the embedded solid element is addressed further in this appendix.

B.1.1 Continuum formulation

Let us consider a body $\Omega \subset \mathbb{R}^3$ with boundary Γ. The three-dimensional solid is subject to prescribed mechanical loads, namely *body force* $\boldsymbol{b}$, *boundary traction* $\boldsymbol{t}$, and *prescribed displacement* $\boldsymbol{g}$. Then, assuming an isotropic and linear elastic behaviour, the boundary

value problem can be expressed as follows.
Find the displacements $\boldsymbol{u} : \Omega \longrightarrow \mathbb{R}^3$ such that:

$$\boldsymbol{div}\,\boldsymbol{\sigma} + \boldsymbol{b} = 0 \quad \text{in } \Omega, \tag{B.1}$$

$$\boldsymbol{u} = \boldsymbol{g} \quad \text{on } \Gamma_D, \tag{B.2}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t} \quad \text{in } \Gamma_N, \tag{B.3}$$

with $\Gamma_D \cup \Gamma_N = \Gamma$, and $\boldsymbol{n}$ the unit outward normal on the boundary Γ. The stress tensor $\boldsymbol{\sigma}$ is related to the strain tensor $\boldsymbol{\varepsilon}$ by means of the constitutive relation :

$$\boldsymbol{\sigma} = \boldsymbol{D}\,\boldsymbol{\varepsilon}, \tag{B.4}$$

where :

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\boldsymbol{\nabla} \boldsymbol{u}^T + \boldsymbol{\nabla} \boldsymbol{u}\right) \tag{B.5}$$

and $\boldsymbol{D}$ is a symmetric positive definite matrix of material constants – which corresponds to linear elasticity in the present case, and is expressed in equation (B.10). Integrating the Galerkin weak form leads to the following expression:

$$\int_\Omega \delta\boldsymbol{\varepsilon}^T\,\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \delta\boldsymbol{u}^T\,\boldsymbol{b}\,\mathrm{d}\Omega - \int_\Gamma \delta\boldsymbol{u}\,\boldsymbol{t}\,\mathrm{d}\Gamma = 0, \tag{B.6}$$

from which the discrete equations can be drawn.

B.1.2 Discrete formulation

The discrete equilibrium equation reads as:

$$\boldsymbol{K}\,\boldsymbol{u} = \boldsymbol{F}, \tag{B.7}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{u}$ the control points displacement vector, and $\boldsymbol{F}$ the load vector that contains $\boldsymbol{b}$ and $\boldsymbol{t}$. The stiffness matrix $\boldsymbol{K}$ can be assembled from the element stiffness matrix $\boldsymbol{K}^e$. In the same way, the force vector $\boldsymbol{F}$ can be assembled from the element force vector $\boldsymbol{F}^e$.

The element stiffness matrix can be computed through the following relation:

$$\boldsymbol{K}^e_{kl} = \int_{\overline{\Omega^e}} \boldsymbol{B}_k^T\,\boldsymbol{D}\,\boldsymbol{B}_l\,|\boldsymbol{J}|\,\mathrm{d}\overline{\Omega}, \tag{B.8}$$

where k and l denote the control points indices of the current element e, $\boldsymbol{B}_i$ is the strain-displacement matrix, $|\boldsymbol{J}|$ is the Jacobian matrix determinant, and $\overline{\Omega}$ is the parametric

domain of the body. The strain-displacement matrix $\boldsymbol{B}_i$ reads as:

$$\boldsymbol{B}_i = \begin{bmatrix} \dfrac{\partial N_i}{\partial X_1} & 0 & 0 \\ 0 & \dfrac{\partial N_i}{\partial X_2} & 0 \\ 0 & 0 & \dfrac{\partial N_i}{\partial X_3} \\ \dfrac{\partial N_i}{\partial X_2} & \dfrac{\partial N_i}{\partial X_1} & 0 \\ 0 & \dfrac{\partial N_i}{\partial X_3} & \dfrac{\partial N_i}{\partial X_2} \\ \dfrac{\partial N_i}{\partial X_3} & 0 & \dfrac{\partial N_i}{\partial X_1} \end{bmatrix}, \tag{B.9}$$

where N_i is the NURBS basis function associated to control point i. The strain-stress matrix $\boldsymbol{D}$ is:

$$\boldsymbol{D} = \begin{bmatrix} \lambda+2\nu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\nu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}, \tag{B.10}$$

where λ and μ are the Lamé material parameters, which are linked to Young's modulus E and Poisson's ratio ν as follows:

$$\lambda = \frac{\nu\, E}{(1+\nu)\,(1-2\nu)}, \tag{B.11}$$

$$\mu = \frac{E}{2\,(1+\nu)}. \tag{B.12}$$

For the classic isogeometric solid element, the Jacobian matrix reads as:

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial X_1}{\partial \xi_1} & \dfrac{\partial X_2}{\partial \xi_1} & \dfrac{\partial X_3}{\partial \xi_1} \\ \dfrac{\partial X_1}{\partial \xi_2} & \dfrac{\partial X_2}{\partial \xi_2} & \dfrac{\partial X_3}{\partial \xi_2} \\ \dfrac{\partial X_1}{\partial \xi_3} & \dfrac{\partial X_2}{\partial \xi_3} & \dfrac{\partial X_3}{\partial \xi_3} \end{bmatrix} \tag{B.13}$$

which is the mapping from the parametric space to the physical space.

The force vector on element e reads as:

$$\boldsymbol{F}_k^e = \int_{\overline{\Omega}^e} N_k\, \boldsymbol{b}\, |\boldsymbol{J}|\, \mathrm{d}\overline{\Omega} + \int_{\overline{\Gamma}_N^e} N_k\, \boldsymbol{t}\, |\boldsymbol{J}|\, \mathrm{d}\overline{\Gamma}, \tag{B.14}$$

where $\overline{\Gamma}$ is the parametric domain of the boundary.

B.2 Sensitivity analysis

B.2.1 Existing works and motivations

In the context of gradient-based isogeometric shape optimisation, in chapter 2 we emphasised the interest of expressing the sensitivities analytically. In fact, the framework brought by IGA is convenient to express the derived quantities that are needed. In the following, we detail how such quantities are computed for the embedded solid element presented in section 4.1.

We strongly relied on works from Hirschler et al. (2020) to be able to derive the quantities involved. The main difference in our case is that we do not make use of the covariant basis vectors. Even though the authors emphasise the practicality of using such notations, we motivate our choice by two main reasons.

- First, we seek to express the derived quantities for the embedded solid element, for which compositions are involved. Using covariant basis vectors "conceals" the explicit contribution of the composed terms, which did not seem convenient from our point of view.
- Second, let us note that Hirschler et al. mainly tackle cases involving shell formulations, for which covariant basis vectors have a great interest, and definitely simplify the notations. Since we only are interested in (composed) solid formulations, this advantage is less valid in our case.

Apart from this aspect, the steps are very similar, yet include supplementary steps due to the mappings composition. This is why we do not develop the sensitivity propagation aspects here – as those steps are no different from the cited work – but rather concentrate on the differentiation of the embedded solid element formulation.

B.2.2 Adjoint sensitivities

The following developments have already been set out in paragraph 2.4.1.2. We repeat them hereafter for completeness, starting from an objective function f expressed using the state variable $\boldsymbol{u}$:

$$f := g\left(x, \boldsymbol{u}(x)\right), \tag{B.15}$$

where $\boldsymbol{u}$ is the solution of the linear system (B.7). Thus, the gradient of function f with respect to the design variable x_i, $i \in [1, d]$ can be decomposed as:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \frac{\partial g}{\partial \boldsymbol{u}} \cdot \frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}x_i}. \tag{B.16}$$

Introducing $\boldsymbol{u}^*$ as the solution of the following adjoint problem:

$$\boldsymbol{K}\boldsymbol{u}^* = \frac{\partial g}{\partial \boldsymbol{u}}, \tag{B.17}$$

equation B.16 can be rewritten as:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \boldsymbol{K}}{\partial x_i} \boldsymbol{u} \right). \tag{B.18}$$

The derivatives of the stiffness matrix $\boldsymbol{K}$ and the load vector $\boldsymbol{F}$ with respect to design variables are computed on the analysis model and propagated to the optimisation model using the refinement operator $\boldsymbol{R}$, kept constant during the optimisation process:

$$\widetilde{\boldsymbol{Q}} = \boldsymbol{R}\, \widetilde{\boldsymbol{P}}. \tag{B.19}$$

$\widetilde{\boldsymbol{Q}}$ and $\widetilde{\boldsymbol{P}}$ being the coordinates of the control points respectively on the analysis and optimisation model. Derivatives can be expressed as:

$$\frac{\partial \bullet}{\partial \widetilde{\boldsymbol{P}}} = \boldsymbol{R}^T \frac{\partial \bullet}{\partial \widetilde{\boldsymbol{Q}}}. \tag{B.20}$$

And the sensitivity reads:

$$\frac{\mathrm{d}f}{\mathrm{d}x_i} = \frac{\partial g}{\partial x_i} + \frac{\partial \widetilde{\boldsymbol{P}}}{\partial x_i} : \boldsymbol{R}^T \left(u^* \cdot \frac{\partial \boldsymbol{F}}{\partial \widetilde{\boldsymbol{Q}}} - \boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{K}}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u} \right), \tag{B.21}$$

where $\dfrac{\partial \widetilde{\boldsymbol{P}}}{\partial x_i}$ is an operator linking the control points coordinates of the optimisation model with the design variables. In the case of linear elasticity, derivatives of $\boldsymbol{K}$ and $\boldsymbol{F}$ in the analysis model can be expressed as geometric quantities from the element formulation.

At this point, we have to express the terms between brackets, that is:

$$\boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \widetilde{\boldsymbol{Q}}} \quad \text{and} \quad \boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{K}}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}. \tag{B.22}$$

In the following we address the derivation of the stiffness matrix, as it constitutes the most challenging part of the differentiation process. As stated in subsection B.1.2, the stiffness matrix is built element-wise. Hence, the above quantities are computed in the same fashion. Then, for element e we have:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = \sum_k \sum_l \boldsymbol{u}_k^{e*} \cdot \frac{\partial \boldsymbol{K}_{kl}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}_l^e, \tag{B.23}$$

where k and l denote the control points indices of the current element e.

In the course of the optimisation, the shape of the physical domain Ω changes under the modification of the control points $\widetilde{\boldsymbol{Q}}$. The parametric domain $\overline{\Omega}$ does not change. Differentiating equation (B.8) with respect to the control points location reads as:

$$\frac{\partial \boldsymbol{K}_{kl}^e}{\partial \widetilde{\boldsymbol{Q}}} = \int_{\overline{\Omega}_e} \frac{\partial}{\partial \widetilde{\boldsymbol{Q}}} \left(\boldsymbol{B}_k^T \boldsymbol{D}\, \boldsymbol{B}_l \, |\boldsymbol{J}| \right) \mathrm{d}\overline{\Omega}, \tag{B.24}$$

keeping in mind that $\boldsymbol{D}$ does not depend upon the design variables, and applying the chain rule yields:

$$\frac{\partial \boldsymbol{K}^e_{kl}}{\partial \widetilde{\boldsymbol{Q}}} = \int_{\overline{\Omega^e}} \left\{ \left(\frac{\partial \boldsymbol{B}_k^T}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{D}\, \boldsymbol{B}_l + \boldsymbol{B}_k^T \boldsymbol{D}\, \frac{\partial \boldsymbol{B}_l}{\partial \widetilde{\boldsymbol{Q}}} \right) |\boldsymbol{J}| + \boldsymbol{B}_k^T \boldsymbol{D}\, \boldsymbol{B}_l \frac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}} \right\} \mathrm{d}\overline{\Omega}. \tag{B.25}$$

Replacing the expression of the derivative of the elementary stiffness matrix in equation (B.23) gives:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = \sum_k \sum_l \boldsymbol{u}_k^{e*} \cdot \int_{\overline{\Omega^e}} \left\{ \left(\frac{\partial \boldsymbol{B}_k^T}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{D}\, \boldsymbol{B}_l + \boldsymbol{B}_k^T \boldsymbol{D}\, \frac{\partial \boldsymbol{B}_l}{\partial \widetilde{\boldsymbol{Q}}} \right) |\boldsymbol{J}| + \boldsymbol{B}_k^T \boldsymbol{D}\, \boldsymbol{B}_l \frac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}} \right\} \mathrm{d}\overline{\Omega}\, \boldsymbol{u}_l^e. \tag{B.26}$$

Hence, the above expression is of the form:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = \sum_k \sum_l \boldsymbol{u}_k^{e*} \int_{\overline{\Omega^e}} \{\cdots\}\, \mathrm{d}\overline{\Omega}\, \boldsymbol{u}_l^e. \tag{B.27}$$

By commuting the double sum with the integral, we can get to an expression of the form:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = \int_{\overline{\Omega^e}} \left\{ \sum_k \sum_l \boldsymbol{u}_k^{e*} \{\cdots\}\, \boldsymbol{u}_l^e \right\} \mathrm{d}\overline{\Omega}. \tag{B.28}$$

Developing the term inside the integral yields:

$$\begin{aligned} \sum_k \sum_l \boldsymbol{u}_k^{e*} \{\cdots\}\, \boldsymbol{u}_l^e = & \sum_k \left\{ \boldsymbol{u}_k^{e*} \frac{\partial \boldsymbol{B}_k^T}{\partial \widetilde{\boldsymbol{Q}}} \right\} \sum_l \{\boldsymbol{D}\, \boldsymbol{B}_l\, \boldsymbol{u}_l^e\}\, |\boldsymbol{J}| \\ & + \sum_k \left\{ \boldsymbol{u}_k^{e*}\, \boldsymbol{B}_k^T \right\} \sum_l \left\{ \boldsymbol{D}\, \frac{\partial \boldsymbol{B}_l^T}{\partial \widetilde{\boldsymbol{Q}}}\, \boldsymbol{u}_l^e \right\} |\boldsymbol{J}| \\ & + \sum_k \left\{ \boldsymbol{u}_k^{e*}\, \boldsymbol{B}_k^T \right\} \sum_l \left\{ \boldsymbol{D}\, \boldsymbol{B}_l^T\, \boldsymbol{u}_l^e \right\} \frac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}}. \end{aligned} \tag{B.29}$$

To identify the above terms, let us recall that the discrete expression of the adjoint strains can be written as:

$$\boldsymbol{\varepsilon}^* = \sum_k \boldsymbol{B}_k\, \boldsymbol{u}_k^*, \tag{B.30}$$

from which we can draw the expression of the derivative *w.r.t.* the control points:

$$\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \widetilde{\boldsymbol{Q}}} = \sum_k \frac{\partial \boldsymbol{B}_k}{\partial \widetilde{\boldsymbol{Q}}}\, \boldsymbol{u}^*. \tag{B.31}$$

Likewise, the stresses read as:

$$\boldsymbol{\sigma} = \sum_l \boldsymbol{D}\, \boldsymbol{B}_l\, \boldsymbol{u}_l, \tag{B.32}$$

and their derivative is written as:

$$\frac{\partial \boldsymbol{\sigma}}{\partial \widetilde{\boldsymbol{Q}}} = \sum_l \boldsymbol{D}\, \frac{\partial \boldsymbol{B}_l}{\partial \widetilde{\boldsymbol{Q}}}\, \boldsymbol{u}_l. \tag{B.33}$$

Identifying the expression of the strains and stresses and their derivatives in expression (B.29), and replacing it inside the integral written in (B.28), we finally have:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = \int_{\overline{\Omega^e}} \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \widetilde{\boldsymbol{Q}}} : \boldsymbol{\sigma} \right) |\boldsymbol{J}| \, \mathrm{d}\overline{\Omega} + \int_{\overline{\Omega^e}} \left(\boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}^*}{\partial \widetilde{\boldsymbol{Q}}} \right) |\boldsymbol{J}| \, \mathrm{d}\overline{\Omega} + \int_{\overline{\Omega^e}} (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}} \mathrm{d}\overline{\Omega}, \tag{B.34}$$

which will be starting point for the next developments.

We emphasise that up to this point, all presented developments are similar to the ones presented in Hirschler et al. (2020). The following section is the one that addresses the specific case of differentiating the operators in the case of the embedded solid formulation.

B.3 Differentiation of the IGA operators

Before expressing the derivatives of the quantities involved in equation (B.34), let us remind the main notations used in the case of the embedded solid element. For convenience we repeat figure 4.4 hereafter. In addition, table B.1 sums up the notations we use.

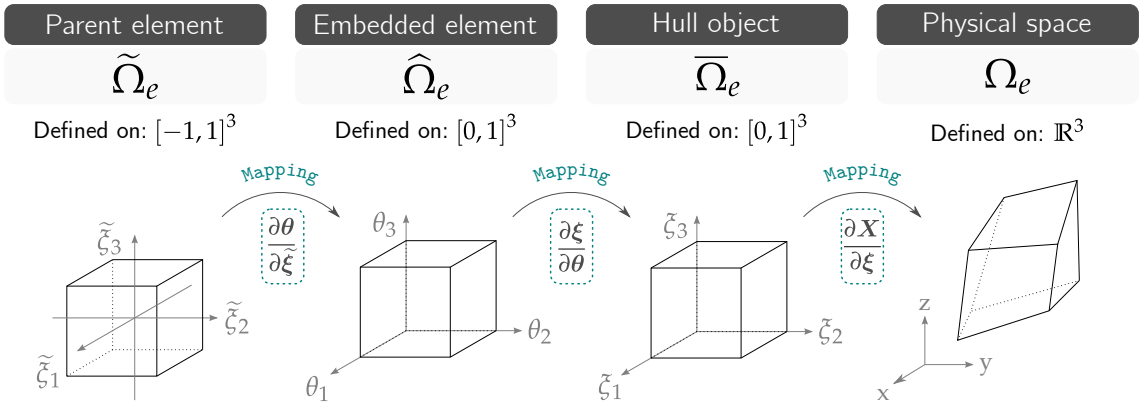


Figure 4.4 – Naming of the various spaces involved in the embedded solid element formulation (repeated from page 88).

B.3.1 Preliminary quantities for the computation of sensitivities

In order to back our developments, let us express some preliminary notations and quantities that will be appearing. In the following, subscript a denotes the index of the active control point, hence we have that $a \in [1, n_{cp}]$, n_{cp} denoting the number of control points

	Embedded solid	Hull object
Basis functions	R	N
Parameters	$\boldsymbol{\theta}$	$\boldsymbol{\xi}$
Physical points	$\boldsymbol{\xi}$	$\boldsymbol{X}$
Control points	$\boldsymbol{P}$	$\boldsymbol{Q}$

Table B.1 – Notations summary for the embedded solid element.

of the embedded solid or the hull object, according to the case. Figure 4.4 recapitulates the notations used for the embedded solid and hull object.

In the following, we express the derivative of the various mappings that will be involved in the expression of the differentiation of the embedded solid element formulation. The final quantities are shaded to facilitate the identification for further use.

Embedded entity Recalling section 4.1, we give the expression of the physical position $\boldsymbol{\xi}$ of the embedded entity:

$$\boldsymbol{\xi} = \sum_i R_i \boldsymbol{P}_i, \tag{B.35}$$

were R are the NURBS basis functions, and $\boldsymbol{P}$ the control points. The derivative *w.r.t.* the embedded parameters $\boldsymbol{\theta}$ gives:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} = \sum_i \frac{\partial R_i}{\partial \boldsymbol{\theta}} \boldsymbol{P}_i, \tag{B.36}$$

which is the mapping from the embedded parameter space to the embedded physical space. Expressing the components indices, the previous relation reads as:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} = \frac{\partial \xi_j}{\partial \theta_i} \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{B.37}$$

The general expression for the derivative of the mapping with respect to the embedded physical coordinates is written as follows:

$$\frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) = \frac{\partial}{\partial \boldsymbol{P}} \left(\sum_i \frac{\partial R_i}{\partial \boldsymbol{\theta}} \boldsymbol{P}_i \right), \tag{B.38}$$

which is a fourth-rank tensor. In order to handle a more convenient quantity, *i.e.*, a third-rank tensor, let us express the previous derivative with respect to a single control point indexed a, written $\boldsymbol{P}_a$. In particular, writing the components explicitly gives:

$$\frac{\partial}{\partial \boldsymbol{P}_a} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) = \frac{\partial}{\partial \boldsymbol{P}_{a_i}} \left(\frac{\partial \xi_k}{\partial \theta_j} \right) \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k. \tag{B.39}$$

Using equation (B.38), we can draw the following relation:

$$\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) = \delta_{ik}\frac{\partial R_a}{\partial \theta_j}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k, \tag{B.40}$$

where δ_{ik} is the Kronecker delta. Hence, this tensor contains the derivatives of the embedded basis functions if $i = k$, ans is 0 otherwise.

Hull object Following the same stages for the hull object, we first express the hull physical points $\boldsymbol{X}$ as:

$$\boldsymbol{X} = \sum_i N_i \boldsymbol{Q}_i, \tag{B.41}$$

with N the NURBS basis functions, and $\boldsymbol{Q}$ the control points. Differentiating this expression *w.r.t.* the hull parameters $\boldsymbol{\xi}$ gives:

$$\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} = \sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}}\boldsymbol{Q}_i, \tag{B.42}$$

which is the mapping from the hull object parameter space to the hull object physical space. Expressing the components of the above quantity gives:

$$\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} = \frac{\partial X_j}{\partial \xi_i}\boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{B.43}$$

The derivative of the mapping with respect to the hull object control points reads as:

$$\frac{\partial}{\partial \boldsymbol{Q}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = \frac{\partial}{\partial \boldsymbol{Q}}\left(\sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}}\boldsymbol{Q}_i\right), \tag{B.44}$$

which is a fourth-rank tensor. As done for the embedded case, we express the derivative of the mapping *w.r.t.* one control point $\boldsymbol{Q}_a$ to lower the rank of the tensor we handle. It then reads as:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = \frac{\partial}{\partial Q_{a_i}}\left(\frac{\partial X_k}{\partial \xi_j}\right)\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k. \tag{B.45}$$

Integrating the result expressed in equation (B.42), we can express the same quantity as follows:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = \delta_{ik}\frac{\partial N_a}{\partial \xi_j}\boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k, \tag{B.46}$$

where δ_{ik} is the Kronecker delta. Likewise, this tensor contains the derivatives of the basis functions of the hull object if $i = k$ and is 0 otherwise.

Composition Having the previous results for the embedded solid and hull object at hand, it is now possible to express the same type of quantity for the composition of both descriptions. In that case, expressing the derivative of the physical points $\boldsymbol{X}$ (see equation (B.42)) by making explicitly appear the contribution of the embedded parametrisation gives:

$$\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} = \sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \left(\sum_j R_j \boldsymbol{P}_j \right) \boldsymbol{Q}_i. \tag{B.47}$$

The derivative with respect to the embedded control points $\boldsymbol{P}$ is obtained *via* the chain rule as follows:

$$\frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) = \frac{\partial}{\partial \boldsymbol{P}} \left(\sum_j R_j \boldsymbol{P}_j \right) \cdot \frac{\partial}{\partial \boldsymbol{\xi}} \left(\sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \boldsymbol{Q}_i \right), \tag{B.48}$$

and in particular, for one control point $\boldsymbol{P}_a$, we have:

$$\frac{\partial}{\partial \boldsymbol{P}_a} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) = \frac{\partial}{\partial \boldsymbol{P}_a} \left(\sum_j R_j \boldsymbol{P}_j \right) \cdot \frac{\partial}{\partial \boldsymbol{\xi}} \left(\sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \boldsymbol{Q}_i \right). \tag{B.49}$$

Let us express the first term of the expression above. Recalling the physical position of the embedded entity expressed in equation (B.35), we have that:

$$\frac{\partial}{\partial \boldsymbol{P}_a} \left(\sum_j R_j \boldsymbol{P}_j \right) = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{P}_a}, \tag{B.50}$$

with:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{P}_a} = \delta_{ij} R_a \, \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{B.51}$$

Let us now turn to the second term of equation (B.49). Recalling equation (B.42), we have:

$$\frac{\partial}{\partial \boldsymbol{\xi}} \left(\sum_i \frac{\partial N_i}{\partial \boldsymbol{\xi}} \boldsymbol{Q}_i \right) = \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right). \tag{B.52}$$

Expressing explicitly the components of this quantity gives:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) &= \frac{\partial}{\partial \xi_i} \left(\frac{\partial X_k}{\partial \xi_j} \right) \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k \\ &= \frac{\partial^2 X_k}{\partial \xi_i \partial \xi_j} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k. \end{aligned} \tag{B.53}$$

Combining equations (B.51) and (B.53), we have that:

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{P}_a} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) &= \left(\delta_{ij} R_a \, \boldsymbol{e}_i \otimes \boldsymbol{e}_j \right) \left(\frac{\partial^2 X_m}{\partial \xi_k \partial \xi_\ell} \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m \right) \\ &= \delta_{ij} R_a \frac{\partial^2 X_m}{\partial \xi_j \partial \xi_\ell} \boldsymbol{e}_i \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m \\ &= R_a \frac{\partial^2 X_m}{\partial \xi_i \partial \xi_\ell} \boldsymbol{e}_i \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m, \end{aligned} \tag{B.54}$$

from which we can eventually draw the following relation:

$$\frac{\partial}{\partial P_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = R_a \frac{\partial^2 X_k}{\partial \xi_i \partial \xi_j} \boldsymbol{e}_i \otimes \boldsymbol{e}_j \otimes \boldsymbol{e}_k. \tag{B.55}$$

In the above, we expressed the terms of the derivative of the mapping from the hull object physical space to the hull object parametric space with respect to the embedded control points. The converse quantity should be expressed as well, *i.e.*, the derivative of the mapping from the embedded solid physical space to the embedded solid parametric space with respect to the hull object control points. If expressed in the same fashion as previously done by making appear explicitly the embedded parametrisation, this quantity would read as:

$$\frac{\partial}{\partial \boldsymbol{Q}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) = \frac{\partial}{\partial \boldsymbol{Q}}\left(\sum_i \frac{\partial R_i}{\partial \boldsymbol{\theta}} \boldsymbol{P}_i\right). \tag{B.56}$$

Since the quantity betweens brackets does not depend on the hull control points position, its derivative *w.r.t.* the control points $\boldsymbol{Q}$ is zero. For further quick reference, we write it explicitly hereafter:

$$\frac{\partial}{\partial Q_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) = \boldsymbol{0}. \tag{B.57}$$

To summarise the previous steps, we expressed the derivatives of the mapping from the embedded solid parametric space to the embedded physical space, written as $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}$, and of the mapping from the hull object parametric space to the hull object physical space, written as $\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}$, both with respect to the embedded solid control points $\boldsymbol{P}$ and the hull control points $\boldsymbol{Q}$. These quantities are involved in the computation of analytical sensitivities, as is exposed hereinafter.

B.3.2 Quantities of interest

Having the preliminary quantities at hand, we can eventually express the quantities involved in equation (B.34), that we repeat for convenience:

$$\begin{aligned}\boldsymbol{u}^{e*} \cdot \frac{\partial \boldsymbol{K}^e}{\partial \widetilde{\boldsymbol{Q}}} \boldsymbol{u}^e = &\int_{\overline{\Omega^e}} \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \widetilde{\boldsymbol{Q}}} : \boldsymbol{\sigma}\right) |\boldsymbol{J}| \, \mathrm{d}\overline{\Omega} + \int_{\overline{\Omega^e}} \left(\boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}^*}{\partial \widetilde{\boldsymbol{Q}}}\right) |\boldsymbol{J}| \, \mathrm{d}\overline{\Omega} \\ &+ \int_{\overline{\Omega^e}} (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}} \mathrm{d}\overline{\Omega},\end{aligned} \tag{B.34}$$

where, for reminder, $\widetilde{\boldsymbol{Q}}$ designates any control point, embedded of hull. Three specific terms can be identified in this expression, that are listed hereafter. Those are the terms we shall express in the remainder of this appendix.

1. Derivative of the Jacobian determinant: $\dfrac{\partial |\boldsymbol{J}|}{\partial \widetilde{\boldsymbol{Q}}}$.

2. Derivative of the adjoint strains: $\dfrac{\partial \boldsymbol{\varepsilon}^*}{\partial \widetilde{\boldsymbol{Q}}}$.

3. Derivative of the stresses: $\dfrac{\partial \boldsymbol{\sigma}^*}{\partial \widetilde{\boldsymbol{Q}}}$.

For each quantity, the derivatives have to be expressed with respect to the control points. We will first give a general expression, denoting the control points as $\boldsymbol{\Lambda}$, before specifying how the expression develops, for the embedded control points $\boldsymbol{P}$ and hull control points $\boldsymbol{Q}$.

B.3.2.1 Derivative of the Jacobian determinant

In the case of the NURBS composition, the Jacobian matrix reads as:

$$\boldsymbol{J} = \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\theta}} = \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}, \tag{B.58}$$

then the expression of the Jacobian determinant reads as:

$$|\boldsymbol{J}| = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right|. \tag{B.59}$$

The general expression of the derivative of the determinant of the Jacobian matrix with respect to the general control points $\boldsymbol{\Lambda}$ is expressed as follows:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{\Lambda}} = \frac{\partial}{\partial \boldsymbol{\Lambda}}\left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right| + \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \frac{\partial}{\partial \boldsymbol{\Lambda}}\left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right|. \tag{B.60}$$

In order to express the derivatives of the form $\dfrac{\partial |\bullet|}{\partial \boldsymbol{\Lambda}}$, Jacobi's formula might be used.

Jacobi's formula

Jacobi's formula expresses the derivative of the determinant of a non-singular matrix function A depending on parameter t, and reads as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\det\left(A\left(t\right)\right) = \det\left(A\left(t\right)\right) \cdot \mathrm{tr}\left(A\left(t\right)^{-1} \cdot \frac{\mathrm{d}A\left(t\right)}{\mathrm{d}t}\right). \tag{B.61}$$

Having this formula in mind, and generalising for a multi-dimensional case, we can now express the quantities appearing in equation (B.60). The derivative of the determinant of the mapping from the embedded solid parametric space to the embedded solid physical space *w.r.t.* the general control points is expressed as:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}} \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| = \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| \cdot \mathrm{tr} \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial}{\partial \boldsymbol{\Lambda}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) \right), \tag{B.62}$$

and the derivative of the determinant of the mapping from the hull object parametric space to the hull object physical space *w.r.t.* the control points reads as:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}} \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right| = \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right| \cdot \mathrm{tr} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{\Lambda}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) \right), \tag{B.63}$$

which we will use to compute the final expressions of the derivatives of the Jacobian determinant with respect to the embedded control points and the hull object control points.

Embedded control points Let us first write equation (B.60) for the embedded control points case:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}} = \frac{\partial}{\partial \boldsymbol{P}} \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| \cdot \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right| + \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| \cdot \frac{\partial}{\partial \boldsymbol{P}} \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right|. \tag{B.64}$$

Then, replacing the quantities expressed in equations (B.63) and (B.62) gives:

$$\begin{aligned} \frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}} = & \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| \cdot \mathrm{tr} \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) \right) \cdot \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right| \\ & + \left| \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right| \cdot \left| \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right| \cdot \mathrm{tr} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) \right). \end{aligned} \tag{B.65}$$

Identifying the expression of the Jacobian determinant, see equation (B.59), in the above expression yields:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}} = |\boldsymbol{J}| \left[\mathrm{tr} \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) \right) + \mathrm{tr} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{P}} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) \right) \right], \tag{B.66}$$

hence the expression of the derivative with respect to the a-th control point $\boldsymbol{P}_a$ reads as:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}_a} = |\boldsymbol{J}| \left[\mathrm{tr} \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial}{\partial \boldsymbol{P}_a} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} \right) \right) + \mathrm{tr} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{P}_a} \left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} \right) \right) \right], \tag{B.67}$$

for which we shall detail how the two terms between brackets are computed.

First term For the first term, we begin by expressing the following product, basing ourselves on the preliminary quantity computed in equation (B.40), which expresses

the derivative of the mapping from the embedded parametric space to the embedded physical space with respect to embedded control points:

$$\begin{aligned}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) &= \left(\frac{\partial \theta_p}{\partial \xi_n}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_p\right)\left(\delta_{km}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m\right) \\ &= \delta_{km}\, \frac{\partial \theta_k}{\partial \xi_n}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m \\ &= \frac{\partial \theta_m}{\partial \xi_n}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m.\end{aligned} \tag{B.68}$$

Reordering the indices for readability purposes, we have that:

$$\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) = \frac{\partial \theta_m}{\partial \xi_k}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m. \tag{B.69}$$

Referring to equation (B.67), which is the final expression we seek to compute, we shall now express the trace of the above result. However, we are dealing with a third-rank tensor, and hence cannot rigorously call it the *trace* of such quantity. Keeping in mind that our goal is to provide an expression that facilitates the implementation, we emphasise that this third-rank tensor in fact corresponds to three square matrices of size 3 that are "stacked", each sub-matrix corresponding to a component of the active control point $\boldsymbol{P}_a$. In practice, we compute the trace of each sub-matrix composing this tensor, and store the values in a vector: each vector component corresponds to the trace of one sub-matrix. We shall denote this modified trace operator as $\mathring{\mathrm{tr}}$, for readability issues. If we denote by α_{ijk} the coefficients of a third-rank tensor $\boldsymbol{A}$, the *modified trace* thus reads as:

$$\mathring{\mathrm{tr}}\left(\boldsymbol{A}\right) = \mathring{\mathrm{tr}}\left(\alpha_{ijk}\right) = \left[\alpha_{11k} + \alpha_{22k} + \alpha_{33k}\right]\, \boldsymbol{e}_k = \alpha_{iik}\, \boldsymbol{e}_k \tag{B.70}$$

Having this distinction in mind, we can write:

$$\mathring{\mathrm{tr}}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\right) = \mathring{\mathrm{tr}}\left(\frac{\partial \theta_m}{\partial \xi_k}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m\right) = \frac{\partial \theta_k}{\partial \xi_\ell}\, \frac{\partial R_a}{\partial \theta_\ell}\, \boldsymbol{e}_k. \tag{B.71}$$

Second term We now turn to the second term between brackets, for which we need the derivative of the mapping from the hull parametric space to hull physical space *w.r.t.* the embedded control points, which has been computed in equation (B.55). We have that:

$$\begin{aligned}\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) &= \left(\frac{\partial \xi_p}{\partial X_n}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_p\right)\left(R_a\, \frac{\partial^2 X_m}{\partial \xi_k \partial \xi_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m\right) \\ &= R_a\, \frac{\partial \xi_k}{\partial X_n}\, \frac{\partial^2 X_m}{\partial \xi_k \partial \xi_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m.\end{aligned} \tag{B.72}$$

Reordering the indices, we finally have:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = R_a\, \frac{\partial \xi_k}{\partial X_\ell}\, \frac{\partial^2 X_n}{\partial \xi_k \partial \xi_m}\, \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m \otimes \boldsymbol{e}_n \tag{B.73}$$

Then, using the modified trace operator previously introduced, we have that:

$$\mathring{\mathrm{tr}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\right) = R_a \frac{\partial \xi_k}{\partial X_\ell}\frac{\partial^2 X_m}{\partial \xi_k \partial \xi_\ell}\boldsymbol{e}_m. \tag{B.74}$$

Finally, we can write the expression of the derivative of the Jacobian determinant with respect to the a-th embedded control point $\boldsymbol{P}_a$:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}_a} = |\boldsymbol{J}| \left(\frac{\partial \theta_m}{\partial \xi_k}\frac{\partial R_a}{\partial \theta_k} + R_a \frac{\partial \xi_k}{\partial X_\ell}\frac{\partial^2 X_m}{\partial \xi_k \partial \xi_\ell}\right)\boldsymbol{e}_m. \tag{B.75}$$

Expressing the p-th component of the above expression to facilitate implementation gives:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{P}_{a_p}} = |\boldsymbol{J}| \left(\sum_\ell \sum_k \left(\frac{\partial \theta_p}{\partial \xi_k}\frac{\partial R_a}{\partial \theta_k} + R_a \frac{\partial \xi_k}{\partial X_\ell}\frac{\partial^2 X_p}{\partial \xi_k \partial \xi_\ell}\right)\right). \tag{B.76}$$

Let us now turn to the expression of the derivative of the Jacobian determinant with respect to the hull control points.

Hull control points Going through the same steps for the hull control points is rather shorter. We start by expressing equation (B.60) for the hull control points case:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{Q}} = \frac{\partial}{\partial \boldsymbol{Q}}\left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right| + \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \frac{\partial}{\partial \boldsymbol{Q}}\left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right|. \tag{B.77}$$

Recalling that the derivative of the mapping from the embedded parametric space to the embedded physical space *w.r.t.* the hull control points is zero – as written in equation (B.57) –, we know that the first term of the above expression is zero. Then, we have that:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{Q}} = \left|\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right| \cdot \frac{\partial}{\partial \boldsymbol{Q}}\left|\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right|. \tag{B.78}$$

Introducing the expression of the determinant of the embedded mapping and the derivative of the hull mapping, see equations (B.63) and (B.62), and identifying $|\boldsymbol{J}|$ gives:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{Q}} = |\boldsymbol{J}| \cdot \mathrm{tr}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{Q}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\right). \tag{B.79}$$

In particular, the derivative of the determinant of the Jacobian matrix with respect to the a-th control point $\boldsymbol{Q}_a$ reads as:

$$\frac{\partial |\boldsymbol{J}|}{\boldsymbol{Q}_a} = |\boldsymbol{J}| \cdot \mathrm{tr}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \cdot \frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\right). \tag{B.80}$$

Recalling the preliminary quantity written in equation (B.46), which is the derivative of the hull mapping with respect to the hull control points, we have that:

$$\begin{aligned}\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) &= \left(\frac{\partial \xi_p}{\partial X_n}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_p\right)\left(\delta_{km}\, \frac{\partial N_a}{\partial \xi_\ell}\, \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m\right) \\ &= \delta_{km}\, \frac{\partial \xi_k}{\partial X_n}\, \frac{\partial N_a}{\partial \xi_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m \\ &= \frac{\partial \xi_m}{\partial X_n}\, \frac{\partial N_a}{\partial \xi_\ell}\, \boldsymbol{e}_n \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m.\end{aligned} \tag{B.81}$$

Reordering the indices gives:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = \frac{\partial \xi_m}{\partial X_k}\, \frac{\partial N_a}{\partial \xi_\ell}\, \boldsymbol{e}_k \otimes \boldsymbol{e}_\ell \otimes \boldsymbol{e}_m. \tag{B.82}$$

Using the modified trace operator introduced in equation (B.70) and applying it to the above expression yields:

$$\mathring{\mathrm{tr}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\right) = \frac{\partial \xi_\ell}{\partial X_k}\, \frac{\partial N_a}{\partial \xi_k}\, \boldsymbol{e}_\ell. \tag{B.83}$$

The expression of the derivative of the Jacobian determinant with respect to the a-th hull control point $\boldsymbol{Q}_a$ finally reads as:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{Q}_a} = |\boldsymbol{J}|\, \frac{\partial \xi_\ell}{\partial X_k}\, \frac{\partial N_a}{\partial \xi_k}\, \boldsymbol{e}_\ell, \tag{B.84}$$

and in particular, for the p-th component:

$$\frac{\partial |\boldsymbol{J}|}{\partial \boldsymbol{Q}_{a_p}} = |\boldsymbol{J}| \sum_k \frac{\partial \xi_p}{\partial X_k}\, \frac{\partial N_a}{\partial \xi_k}. \tag{B.85}$$

Now that the derivatives of the Jacobian determinant with respect to the embedded and the hull control points have been expressed, let us turn to the derivative of the adjoint strains.

B.3.2.2 Derivative of the adjoint strains

In order to express the derivative of the adjoint strains, we begin by introducing the expression of the adjoint strains in the case of the embedded element.

Expression of the adjoint strains Basing ourselves on equation (B.5), the expression of the adjoint strains can be written as follows:

$$\boldsymbol{\varepsilon}^* = \frac{1}{2}\left[\left(\frac{\partial \boldsymbol{u}^*}{\partial \boldsymbol{X}}\right)^T + \left(\frac{\partial \boldsymbol{u}^*}{\partial \boldsymbol{X}}\right)\right], \tag{B.86}$$

where $\boldsymbol{u}^*$ is the solution of the adjoint problem (B.17). Dropping the matrix notation for clarity, we have:

$$\varepsilon_{ij}^* = \frac{1}{2}\left[\left(\frac{\partial u_i^*}{\partial X_j}\right)^T + \frac{\partial u_i^*}{\partial X_j}\right] = \frac{1}{2}\left[\frac{\partial u_j^*}{\partial X_i} + \frac{\partial u_i^*}{\partial X_j}\right]. \tag{B.87}$$

Recalling the following, which is nothing more than writing explicitly the embedded solid mapping:

$$\frac{\partial u_j^*}{\partial X_i} = \frac{\partial \boldsymbol{\theta}}{\partial X_i}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}, \tag{B.88}$$

the expression of the adjoint strains in the case of the embedded solid element is written as:

$$\varepsilon_{ij}^* = \frac{1}{2}\left[\frac{\partial \boldsymbol{\theta}}{\partial X_i}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\theta}}{\partial X_j}\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\right]. \tag{B.89}$$

Introducing the composition in the above expression, we finally have:

$$\varepsilon_{ij}^* = \frac{1}{2}\left[\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\xi}}{\partial X_j}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\right]. \tag{B.90}$$

Expression of the derivative of the adjoint strains Having the preceding expression in hand, we can give the general expression for the derivative of the adjoint strains with respect to the control points $\boldsymbol{\Lambda}$:

$$\frac{\partial \varepsilon_{ij}^*}{\partial \boldsymbol{\Lambda}} = \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_j}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right)\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\right]. \tag{B.91}$$

One can note that the differentiated term in the above expression is not known. Let us express this quantity in the following, by applying the chain rule:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right) = \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_i}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} + \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right). \tag{B.92}$$

This step makes appear more unknown differentiated quantities, that we clarify hereafter :

1. and the derivative of the mapping from the hull object parametric space to the hull object physical space,
$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\right),$$
2. the derivative of the inverse of the mapping from the embedded solid parametric space to the embedded solid physical space,
$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right).$$

Let us begin with the derivative of the inverse of the mapping from the embedded solid parametric space to the embedded solid physical space. We have that:

$$\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} = \left[\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right]^{-1}, \tag{B.93}$$

and hence, the following relation holds:

$$\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} = \boldsymbol{I}, \tag{B.94}$$

where $\boldsymbol{I}$ is the identity matrix. Differentiating the above expression thus yields:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right) \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) = \mathbf{0}, \tag{B.95}$$

where $\mathbf{0}$ is a zero matrix. We can finally draw the expression of the unknown quantity as follows:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right) = -\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}} \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right) \frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}, \tag{B.96}$$

recalling that the differentiated quantity in play in the above has already been encountered for the differentiation of the determinant of the Jacobian matrix.

We now turn to the derivative of the mapping from the hull object parametric space to the hull object physical space. Proceeding as we just did above, let us start with the trivial equality:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} = \left[\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right]^{-1}. \tag{B.97}$$

The following relation then holds:

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} = \boldsymbol{I}. \tag{B.98}$$

As carried out above, we differentiate this expression *w.r.t.* the control points in the general case:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\right) \frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}} + \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) = \mathbf{0}, \tag{B.99}$$

from which we finally draw:

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\right) = -\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}} \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}. \tag{B.100}$$

In this case as well, the differentiated quantity that takes part in the final expression was needed to express the derivative of the determinant of the Jacobian matrix. More specifically, we can write the following relation in order to gather all necessary terms to develop equation (B.92):

$$\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_i}\right) = -\frac{\partial \boldsymbol{\xi}}{\partial X_i} \frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right) \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}. \tag{B.101}$$

Finally, using the relations given above to develop the expression of the derivative of the adjoint strains *w.r.t.* control points in the general case (B.91) gives:

$$\begin{aligned}\frac{\partial \varepsilon_{ij}^*}{\partial \boldsymbol{\Lambda}} = \frac{1}{2}\Bigg[&- \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} \\ &- \frac{\partial \boldsymbol{\xi}}{\partial X_j}\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_i^*}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{\xi}}{\partial X_j}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{\Lambda}}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\Bigg].\end{aligned} \tag{B.102}$$

Two quantities can already be pre-computed, as they are the same for both the embedded and hull object control points. Taking $\iota = i, j$, we have:

$$\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_\iota^*}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \theta_m}{\partial \xi_\ell}\,\boldsymbol{e}_\ell \otimes \boldsymbol{e}_m\right)\left(\frac{u_\iota^*}{\partial \theta_k}\,\boldsymbol{e}_k\right) = \frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_\iota^*}{\partial \theta_k}\,\boldsymbol{e}_l, \tag{B.103}$$

and

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_\iota^*}{\partial \boldsymbol{\theta}} = \left(\frac{\partial \xi_n}{\partial X_m}\,\boldsymbol{e}_m \otimes \boldsymbol{e}_n\right)\left(\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_\iota^*}{\partial \theta_k}\,\boldsymbol{e}_l\right) = \frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_\iota^*}{\partial \theta_k}\,\boldsymbol{e}_m. \tag{B.104}$$

Let us now tackle the expression of the derivative of the adjoint strains with respect to the embedded control points.

Embedded control points Basing ourselves on equation (B.91), the derivative of the adjoint strains with respect to the embedded control points reads as:

$$\frac{\partial \varepsilon_{ij}^*}{\partial \boldsymbol{P}} = \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{P}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial}{\partial \boldsymbol{P}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_j}\right)\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\right]. \tag{B.105}$$

In the remainder of this paragraph we will only focus on the first term between brackets of the above equation, as the second is computed in the same fashion. Moreover, we consider the derivatives with respect to the a-th control point $\boldsymbol{P}_a$. Hence, using equation (B.92), we can express the first term as follows:

$$\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_i}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}, \tag{B.106}$$

which in turn, using equations (B.101) and (B.96) for the first and second sub-term respectively, can be developed as:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = &- \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} \\ &- \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}.\end{aligned} \tag{B.107}$$

In the following we express the two sub-terms of this expression step by step.

First sub-term Reminding the preliminary quantity expressed in equation (B.55), we can compute the following:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(Ra\,\frac{\partial^2 X_q}{\partial \xi_n \partial \xi_p}\boldsymbol{e}_n\otimes\boldsymbol{e}_p\otimes\boldsymbol{e}_q\right)\left(\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\right)\\ &= R_a\,\frac{\partial^2 X_m}{\partial \xi_n \partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_p,\end{aligned} \tag{B.108}$$

and finally express the first sub-term as:

$$\begin{aligned}\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\frac{\partial \xi_q}{\partial X_i}\boldsymbol{e}_q\right)\left(R_a\,\frac{\partial^2 X_m}{\partial \xi_n \partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_p\right)\\ &= R_a\,\frac{\partial \xi_n}{\partial X_i}\frac{\partial^2 X_m}{\partial \xi_n \partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_p\end{aligned} \tag{B.109}$$

Second sub-term Using the preliminary quantity written in equation (B.40), the first step is to compute:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\delta_{mp}\,\frac{\partial R_a}{\partial \theta_n}\boldsymbol{e}_m\otimes\boldsymbol{e}_n\otimes\boldsymbol{e}_p\right)\left(\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_l\right)\\ &= \delta_{ml}\,\frac{\partial R_a}{\partial \theta_n}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\otimes\boldsymbol{e}_n\\ &= \frac{\partial R_a}{\partial \theta_n}\frac{\partial \theta_k}{\partial \xi_m}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\otimes\boldsymbol{e}_n.\end{aligned} \tag{B.110}$$

Then, we have:

$$\begin{aligned}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\frac{\partial \theta_p}{\partial \xi_n}\boldsymbol{e}_n\otimes\boldsymbol{e}_p\right)\left(\frac{\partial R_a}{\partial \theta_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_\ell\otimes\boldsymbol{e}_m\right)\\ &= \frac{\partial \theta_\ell}{\partial \xi_n}\frac{\partial R_a}{\partial \theta_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_m,\end{aligned} \tag{B.111}$$

from which we can finally draw the second sub-term:

$$\begin{aligned}\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{P}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\frac{\partial \xi_p}{\partial X_i}\boldsymbol{e}_p\right)\left(\frac{\partial \theta_\ell}{\partial \xi_n}\frac{\partial R_a}{\partial \theta_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_m\right)\\ &= \frac{\partial \xi_n}{\partial X_i}\frac{\partial \theta_\ell}{\partial \xi_n}\frac{\partial R_a}{\partial \theta_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m.\end{aligned} \tag{B.112}$$

In order to give a convenient expression for coding and implementation, we express hereafter the p-th component of the first term between brackets of equation (B.105) for the a-th control point. Hence, joining results form equations (B.109) and (B.112), we have:

$$\frac{\partial}{\partial P_{a_p}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = -R_a \sum_n \sum_m \sum_\ell \sum_k \frac{\partial \xi_n}{\partial X_i}\frac{\partial^2 X_m}{\partial \xi_n \partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k} - \sum_m \sum_\ell \sum_k \frac{\partial \xi_m}{\partial X_i}\frac{\partial \theta_\ell}{\partial \xi_m}\frac{\partial R_a}{\partial \theta_\ell}\frac{\partial \theta_k}{\partial \xi_p}\frac{\partial u_j^*}{\partial \theta_k}. \quad \text{(B.113)}$$

The p-th component of other term between brackets of equation (B.105) for the a-th control point is expressed conversely, by permuting the i and j indices. Let us now turn to the expression of the derivative of the adjoint strains with respect to the embedded control points.

Hull control points The derivative of the adjoint strains with respect to the hull control points reads as:

$$\frac{\partial \varepsilon_{ij}^*}{\partial \boldsymbol{Q}} = \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{Q}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial}{\partial \boldsymbol{Q}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_j}\right)\frac{\partial u_i^*}{\partial \boldsymbol{\theta}}\right]. \quad \text{(B.114)}$$

As for the embedded control points case, we will only focus on the first term between brackets of the above equation. We also consider the derivatives with respect to the a-th control point $\boldsymbol{Q}_a$. Hence, using equation (B.92), we can express the first term as follows:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = \frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial X_i}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} + \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}. \quad \text{(B.115)}$$

Using equations (B.96) and (B.101), we can write the following:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = -\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} - \frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\theta}}\right)\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}. \quad \text{(B.116)}$$

We draw the reader's attention to the second term of the above expression, since the derivative of the mapping from the embedded solid physical space to the embedded solid parametric space *w.r.t.* the hull object control points comes into play. As written in (B.57), this quantity is zero, and hence equation (B.116) simplifies as follows:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = -\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}}. \quad \text{(B.117)}$$

Let us express this quantity step by step. Using equation (B.46), we have that:

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\delta_{nq}\frac{\partial N_a}{\partial \xi_p}\boldsymbol{e}_n\otimes\boldsymbol{e}_p\otimes\boldsymbol{e}_q\right)\left(\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\right)\\ &= \delta_{nm}\frac{\partial N_a}{\partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_p\\ &= \frac{\partial N_a}{\partial \xi_p}\frac{\partial \xi_\ell}{\partial X_n}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\otimes\boldsymbol{e}_p.\end{aligned} \tag{B.118}$$

Reordering the indices gives:

$$\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = \frac{\partial N_a}{\partial \xi_n}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\otimes\boldsymbol{e}_n. \tag{B.119}$$

We can eventually draw the complete sub-term from the above expression:

$$\begin{aligned}\frac{\partial \boldsymbol{\xi}}{\partial X_i}\frac{\partial}{\partial \boldsymbol{Q}_a}\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right)\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{X}}\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\xi}}\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} &= \left(\frac{\partial \xi_p}{\partial X_i}\boldsymbol{e}_p\right)\left(\frac{\partial N_a}{\partial \xi_n}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_m\otimes\boldsymbol{e}_n\right)\\ &= \frac{\partial \xi_m}{\partial X_i}\frac{\partial N_a}{\partial \xi_n}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}\boldsymbol{e}_n\end{aligned} \tag{B.120}$$

Finally, the p-th component of the first term between brackets of equation (B.114) for the a-th control point $\boldsymbol{Q}_a$ can be expressed as:

$$\frac{\partial}{\partial Q_{a_p}}\left(\frac{\partial \boldsymbol{\theta}}{\partial X_i}\right)\frac{\partial u_j^*}{\partial \boldsymbol{\theta}} = -\sum_m\sum_\ell\sum_k \frac{\partial \xi_m}{\partial X_i}\frac{\partial N_a}{\partial \xi_p}\frac{\partial \xi_\ell}{\partial X_m}\frac{\partial \theta_k}{\partial \xi_\ell}\frac{\partial u_j^*}{\partial \theta_k}. \tag{B.121}$$

The p-th component of the other term between brackets of equation (B.114) for the a-th control point is expressed conversely, by permuting the i and j indices.

B.3.2.3 Derivative of the stresses

Now turning ourselves to the expression of the stresses, let us first write Hooke's law for isotropic materials in direct tensor notation:

$$\boldsymbol{\sigma} = \lambda \mathrm{tr}\left(\boldsymbol{\varepsilon}\right)\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}, \tag{B.122}$$

where $\boldsymbol{I}$ is the identity matrix and λ and μ are the Lamé parameters. In index notation, the expression of the stresses reads as:

$$\sigma_{ij} = \lambda\,\varepsilon_{kk}\,\delta_{ij} + 2\,\mu\,\varepsilon_{ij}, \tag{B.123}$$

where δ_{ij} is the Kronecker delta. With this expression at hand, let us give the general expression for the derivative of the stresses with respect to the control points:

$$\frac{\partial \sigma_{ij}}{\partial \boldsymbol{\Lambda}} = \lambda\frac{\partial \varepsilon_{kk}}{\partial \boldsymbol{\Lambda}}\delta_{ij} + 2\,\mu\frac{\partial \varepsilon_{ij}}{\partial \boldsymbol{\Lambda}}, \tag{B.124}$$

where the derivative of the trace of the stresses can be written as:

$$\frac{\partial \varepsilon_{kk}}{\partial \mathbf{\Lambda}} = \frac{\partial}{\partial \mathbf{\Lambda}} \left(\mathrm{tr} \left(\varepsilon_{ij} \right) \right). \tag{B.125}$$

The trace being a linear operator, it commutes with the derivative. Hence, we can write:

$$\frac{\partial}{\partial \mathbf{\Lambda}} \left(\mathrm{tr} \left(\varepsilon_{ij} \right) \right) = \mathrm{tr} \left(\frac{\partial \varepsilon_{ij}}{\partial \mathbf{\Lambda}} \right). \tag{B.126}$$

In the above expression we need to express the trace of the derivative of the state strains with respect to the control points, which is a third-rank tensor. Recalling the modified trace operator introduced in equation (B.70), the expression of the derivative of the stresses with respect to control points reads as:

$$\frac{\partial \sigma_{ij}}{\partial \mathbf{\Lambda}} = \lambda \, \mathring{\mathrm{tr}} \left(\frac{\partial \varepsilon_{ij}}{\partial \mathbf{\Lambda}} \right) \delta_{ij} + 2 \, \mu \, \frac{\partial \varepsilon_{ij}}{\partial \mathbf{\Lambda}}. \tag{B.127}$$

One can note that the derivative of the state strains with respect to the control points is needed to compute the derivative of the stresses. The derivative of the state strains is expressed in a similar fashion as for the adjoint strains, replacing the derivative of the adjoint displacement *w.r.t.* the embedded control points with this of the state displacement as follows:

$$\frac{\partial \varepsilon}{\partial \mathbf{\Lambda}} = \frac{1}{2} \left[\frac{\partial}{\partial \mathbf{\Lambda}} \left(\frac{\partial \boldsymbol{\theta}}{\partial X_i} \right) \frac{\partial u_j}{\partial \boldsymbol{\theta}} + \frac{\partial}{\partial \mathbf{\Lambda}} \left(\frac{\partial \boldsymbol{\theta}}{\partial X_j} \right) \frac{\partial u_i}{\partial \boldsymbol{\theta}} \right]. \tag{B.128}$$

The terms of this equation can be computed using the state counterparts of the adjoint expressions detailed in equations (B.113) and (B.121) for the embedded and hull control points, respectively.

Hence, the derivative of the state strains has to be computed prior to computing the expression of the derivative of the stresses *w.r.t.* control points, for the embedded solid and hull object cases. Apart from this additional step, all necessary material to compute equation (B.127) has been expressed in the above.

Concluding remarks

The main expressions for the computation of analytical sensitivities in the case of the embedded solid element have been developed. Although those results have not been used for numerical experiments in this manuscript, we believe it completes the efforts carried out to build an analytical framework for shape optimisation.

References

Keonys. *TOSCA, de puissantes possibilités d'optimisation FEA et CFD*. `https://www.keonys.com/logiciels-simulations/toca-optimisation-fea-cfd-solution-simulation-dassault-systemes/`. (Accessed: 2021-11-28).

W. Campbell. Protection of Turbine Disk Wheels from Axial Vibration. General electric Company, 1924.

S. A. Coons. Surfaces for Computer-Aided Design of space forms. Technical report, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1967.

J. A. Nitsche. Uber ein Variationsprinzip zur Lösung Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen uneworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971.

M. G. Cox. The numerical evaluation of B-Splines. *IMA Journal of Applied Mathematics*, 10(2):134–149, 1972.

C. De Boor. On Calculating with B-splines. *Journal of Approximation Theory*, 1(6):50–62, 1972.

B. M. Irons and A. Razzaque. Experience with the patch test for convergence of finite elements. In *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*, pages 557–587. Academic Press, jan 1972.

S. Ching-Tze. On spline finite element method. *Mathematica Numerica Sinica*, 1, 1979.

S. Adali. Design of shear-deformable antisymmetric angle-ply laminates to maximize the fundamental frequency and frequency separation. *Composite Structures*, 2(4): 349–369, 1984.

T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1986*, 20(4):151–160, 1986.

S. Coquillart. Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. In *Proceedings of the 17th Annual Conference on Computer Graphics and*

Interactive Techniques, volume 24, pages 187–196, Dallas, TX, USA, 1989. Association for Computing Machinery.

K. J. Duffy and S. Adali. Optimal Hybrid Fibre Orientation for and of Antisymmetric Fundamental Separation Laminates Maximum Frequency. *Journal of Sound and Vibration*, 146(2):181–190, 1991.

C. Bernardi, Y. Maday, and A. T. Patera. Domain Decomposition by the Mortar Element Method. In H. G. Kaper, M. Garbey, and G. W. Pieper, editors, *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*, pages 269–286. Springer Netherlands, Dordrecht, 1993.

D. Chapelle and K. J. Bathe. The inf-sup test. *Computers and Structures*, 47(4-5):537–545, 1993.

G. N. Vanderplaats. Thirty years of modern structural optimization. *Advances in Engineering Software*, 16:81–88, 1993.

D. Kraft. Algorithm 733: TOMP-Fortran Modules for Optimal Control Calculations. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):262–281, 1994.

P. L. Miller, J. H. Oliver, D. P. Miller, and D. L. Tweedt. BladeCAD: An interactive geometric design tool for turbomachinery blades. In *ASME 1996 International Gas Turbine and Aeroengine Congress and Exhibition, GT 1996*, volume 1, page 8, Birmingham, United Kingdom, 1996.

L. Piegl and W. Tiller. *The NURBS book*. Springer Science & Business Media, 1996.

O. Adam, F. Marin, C. Thonon, J.-A. Essers, J.-C. Golinval, and O. Léonard. Developpement d'une méthodologie d'optimisation aérodynamique et mécanique d'aubes de compresseurs. *Revue Française de Mécanique*, 4:247–253, 1998.

P. Kagan, A. Fischer, and P. Z. Bar-Yoseph. New B-Spline Finite Element approach for geometrical design and mechanical analysis. *International Journal for Numerical Methods in Engineering*, 41(3):435–458, 1998.

G. Farin and D. Hansford. Discrete Coons patches. *Computer Aided Geometric Design*, 16 (7):691–700, 1999.

S. Cho and K. K. Choi. Design sensitivity analysis and optimization of non-linear transient dynamics. Part II-configuration design. *International Journal for Numerical Methods in Engineering*, 48:375–399, 2000.

K. K. Choi and S. Cho. Design sensitivity analysis and optimization of non-linear transient dynamics. Part I-sizing design. *International Journal for Numerical Methods in Engineering*, 48:351–373, 2000.

B. I. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM Journal on Numerical Analysis*, 38(3):989–1012, 2000.

S. Adali and V. E. Verijenko. Optimum stacking sequence design of symmetric hybrid laminates undergoing free vibrations. *Composite Structures*, 54(2-3):131–138, 2001.

V. Hernández-Mederos and J. Estrada-Sarlabous. Sampling points on regular parametric curves with control of their distribution. *Computer Aided Geometric Design*, 20(6):363–382, 2003.

T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

J.-C. Thevenin. *Le turboréacteur, moteur des avions à réaction*. AAF, 2004.

T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.

K. C. Hui and Y. B. Wu. Feature-based decomposition of trimmed surface. *Computer-Aided Design*, 37(8):859–867, 2005.

M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris, T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen, D. Roose, and T. Schlick, editors. *Automatic Differentiation: Applications, Theory, and Implementations*, volume 50. Springer Science & Business Media, 2006.

R. Duvigneau. Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization. Technical report, 2006.

J. S. Jensen and N. L. Pedersen. On maximal eigenfrequency separation in two-material structures: The 1D and 2D scalar cases. *Journal of Sound and Vibration*, 289(4-5): 967–986, 2006.

J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

J. Du and N. Olhoff. Topological design of freely vibrating continuum structures for maximum values of simple and multiple eigenfrequencies and frequency gaps. *Structural and Multidisciplinary Optimization*, 34(2):91–110, 2007.

Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid-structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43(1): 3–37, 2008.

C. H. Bischof, H. M. Bücker, P. Hovland, U. Naumann, and J. Utke, editors. *Advances in automatic differentiation*. Springer, 2008.

T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes. B and F projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2732–2762, 2008.

H. Gómez, V. M. Calo, Y. Bazilevs, and T. J. R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197(49-50):4333–4352, 2008.

W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33-40):2976–2988, 2008.

S. Cho and S. H. Ha. Isogeometric shape design optimization: Exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38(1):53–70, 2009.

J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis*. John Wiley & Sons, Ltd, Chichester, UK, 2009.

J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198 (49-52):3902–3914, 2009.

H. J. Kim, Y. D. Seo, and S. K. Youn. Isogeometric analysis for trimmed CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 198(37-40):2982–2995, 2009.

Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):264–275, 2010.

D. Großmann and B. Jüttler. Volumetric Geometry Reconstruction of Turbine Blades for Aircraft Engines. In *International Conference on Curves and Surfaces*, pages 280–295. Springer, 2010.

H. J. Kim, Y. D. Seo, and S. K. Youn. Isogeometric analysis with trimming technique for problems of arbitrary complex topology. *Computer Methods in Applied Mechanics and Engineering*, 199(45-48):2796–2812, 2010.

S. Lipton, J. A. Evans, Y. Bazilevs, T. Elguedj, and T. J. R. Hughes. Robustness of isogeometric structural discretizations under severe mesh distortion. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):357–373, 2010.

A. P. Nagy, M. M. Abdalla, and Z. Gürdal. Isogeometric sizing and shape optimisation of beam structures. *Computer Methods in Applied Mechanics and Engineering*, 199 (17-20):1216–1230, 2010.

X. Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2059–2071, 2010.

N. D. Manh, A. Evgrafov, A. R. Gersborg, and J. Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200(13-16):1343–1353, 2011.

M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213-216:206–222, 2011.

I. Temizer, P. Wriggers, and T. J. R. Hughes. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12): 1100–1112, 2011.

A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3554–3567, 2011.

G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Computer Methods in Applied Mechanics and Engineering*, 200(23-24):2021–2031, 2011.

M. P. Boyce. 5 - Rotor Dynamics. In M. P. Boyce, editor, *Gas Turbine Engineering Handbook (Fourth Edition)*, chapter 5, pages 215–250. Butterworth-Heinemann, fourth edi edition, 2012.

C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

J. Gravesen, A. Evgrafov, D.-M. Nguyen, and P. Nørtoft. Planar Parametrization in Isogeometric Analysis. In *International conference on mathematical methods for curves and surfaces*, pages 189–212. Springer, 2012.

C. Hesch and P. Betsch. Isogeometric analysis and domain decomposition methods. *Computer Methods in Applied Mechanics and Engineering*, 213-216:104–112, 2012.

M.-C. Hsu and Y. Bazilevs. Fluid-structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50(6):821–833, 2012.

A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, 2012.

D. M. Nguyen, A. Evgrafov, and J. Gravesen. Isogeometric shape optimization for electromagnetic scattering problems. *Progress In Electromagnetics Research*, 45:117–146, 2012.

N. Olhoff, B. Niu, and G. Cheng. Optimum design of band-gap beam structures. *International Journal of Solids and Structures*, 49(22):3158–3169, 2012.

E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249-252:104–115, 2012.

J. D. Sanders, T. A. Laursen, and M. A. Puso. A Nitsche Embedded Mesh Method. *Computational Mechanics*, 42(2):243–257, 2012.

R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric analysis of trimmed NURBS geometries. *Computer Methods in Applied Mechanics and Engineering*, 241: 93–111, 2012.

P. Stein, M.-C. Hsu, Y. Bazilevs, and K. E. Beucke. Operator- and template-based modeling of solid geometry for Isogeometric Analysis with application to Vertical Axis Wind Turbine simulation. *Computer Methods in Applied Mechanics and Engineering*, 213-216:71–83, 2012.

D. J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255: 133–146, 2013.

M. Cavazzuti. *Optimization Methods: From Theory to Design Scientific and Technological Aspects in Mechanics*. Springer Science & Business Media, 2013.

R. Echter, B. Oesterle, and M. Bischoff. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 254:170–180, 2013.

S. Garg. Aircraft Turbine Engine Control Research at NASA Glenn Research Center. *Journal of Aerospace Engineering*, 26(2):422–438, 2013.

K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2013.

A. Koshakji, A. Quarteroni, and G. Rozza. Free Form Deformation techniques applied to 3D shape optimization problems. *Communications in Applied and Industrial Mathematics*, 4:1–26, 2013.

J. Liu, L. Dedè, J. A. Evans, M. J. Borden, and T. J. R. Hughes. Isogeometric analysis of the advective Cahn-Hilliard equation: Spinodal decomposition under shear flow. *Journal of Computational Physics*, 242:321–350, 2013.

A. P. Nagy, S. T. IJsselmuiden, and M. M. Abdalla. Isogeometric design of anisotropic shells: Optimal form and material distribution. *Computer Methods in Applied Mechanics and Engineering*, 264:145–162, 2013.

P. Nørtoft and J. Gravesen. Isogeometric shape optimization in fluid mechanics. *Structural and Multidisciplinary Optimization*, 48(5):909–925, 2013.

B. U. Park, Y. D. Seo, O. Sigmund, and S. K. Youn. Shape optimization of the stokes flow problem based on isogeometric analysis. *Structural and Multidisciplinary Optimization*, 48(5):965–977, 2013.

G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Analysis-suitable volume parameterization of multi-block computational domain in isogeometric applications. *Computer-Aided Design*, 45(2):395–404, 2013a.

G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis. *Computer-Aided Design*, 45:812–821, 2013b.

A. Apostolatos, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 97 (7):473–504, 2014.

E. J. Evans, M. A. Scott, X. Li, and D. C. Thomas. Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284: 1–20, 2014.

K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.

J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274:148–167, 2014.

N. D. Manh, A. Evgrafov, J. Gravesen, and D. Lahaye. Isogeometric shape optimization of magnetic density separators. *COMPEL:The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, 33(4):1416–1433, 2014.

M. J. Martin, E. Andres, C. Lozano, and E. Valero. Volumetric B-Splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, 37: 26–36, 2014.

V. P. Nguyen, P. Kerfriden, M. Brino, S. P. A. Bordas, and E. Bonisoli. Nitsche's method for two and three dimensional NURBS patch coupling. *Computational Mechanics volume*, 53:1163–1182, 2014.

M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank. Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries. *Computer Methods in Applied Mechanics and Engineering*, 269:46–71, 2014.

X. Wang and X. Qian. An optimization approach for constructing trivariate B-spline solids. *Computer-Aided Design*, 46(1):179–191, 2014.

R. Bouclier, T. Elguedj, and A. Combescure. An isogeometric locking-free NURBS-based solid-shell element for geometrically nonlinear analysis. *International Journal for Numerical Methods in Engineering*, 101(10):774–808, 2015.

E. Brivadis, A. Buffa, B. Wohlmuth, and L. Wunderlich. Isogeometric mortar methods. *Computer Methods in Applied Mechanics and Engineering*, 284:292–319, 2015.

W. Dornisch, G. Vitucci, and S. Klinkel. The weak substitution method – an application of the mortar method for patch coupling in NURBS-based isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 103(3):205–234, 2015.

X. Du, G. Zhao, and W. Wang. Nitsche method for isogeometric analysis of Reissner-Mindlin plate with non-conforming multi-patches. *Computer Aided Geometric Design*, 35:121–136, 2015.

D. Fußeder, B. Simeon, and A.-V. Vuong. Fundamental aspects of shape optimization in the context of isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 286:313–331, 2015.

Y. Guo and M. Ruess. Nitsche's method for a coupling of isogeometric thin shells and blended shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284: 881–905, 2015.

K. V. Kostas, A. I. Ginnis, C. G. Politis, and P. D. Kaklis. Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, 284:611–622, 2015.

Z.-P. Wang and S. Turteltaub. Isogeometric shape optimization for quasi-static processes. *International Journal for Numerical Methods in Engineering*, 104(5):347–371, 2015.

H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. Isogeometric analysis-suitable trivariate NURBS models from standard B-Rep models. *Computer Methods in Applied Mechanics and Engineering*, 307:256–274, 2016.

A. Buffa and C. Giannelli. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. *Mathematical Models and Methods in Applied Sciences*, 26(01):1–25, 2016.

W. Dornisch and R. Müller. Patch coupling with the isogeometric dual mortar approach. *Proceedings in Applied Mathematics and Mechanics*, 16(1):193–194, 2016.

P. Farah, A. T. Vuong, W. A. Wall, and A. Popp. Volumetric coupling approaches for multiphysics simulations on non-matching meshes. *International Journal for Numerical Methods in Engineering*, 108(12):1550–1576, 2016.

C. Giannelli, B. Jüttler, S. K. Kleiss, A. Mantzaflaris, B. Simeon, and J. Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 299:337–365, 2016.

P. Kang and S. K. Youn. Isogeometric shape optimization of trimmed shell structures. *Structural and Multidisciplinary Optimization*, 53(4):825–845, 2016.

F. Massarwi and G. Elber. A B-spline based framework for volumetric object modeling. *Computer Aided Design*, 78:36–47, 2016.

M. E. Matzen and M. Bischoff. A weighted point-based formulation for isogeometric contact. *Computer Methods in Applied Mechanics and Engineering*, 308:73–95, 2016.

D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S. K. F. Stoter, Y. Yu, and Y. Zhao. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.

H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. Towards an automatic isogeometric analysis suitable trivariate models generation—Application to geometric parametric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:623–645, 2017.

A. M. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, and K.-U. Bletzinger. Embedded structural entities in NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 325:198–218, 2017.

R. Bouclier, J. C. Passieux, and M. Salaün. Development of a new, more regular, mortar method for the coupling of NURBS subdomains within a NURBS patch: Application to a non-intrusive local enrichment of NURBS patches. *Computer Methods in Applied Mechanics and Engineering*, 316:123–150, 2017.

L. Coox, F. Greco, O. Atak, D. Vandepitte, and W. Desmet. A robust patch coupling method for NURBS-based isogeometric analysis of non-conforming multipatch surfaces. *Computer Methods in Applied Mechanics and Engineering*, 316:235–260, 2017.

S. D. Daxini and J. Prajapati. Parametric shape optimization techniques based on Meshless methods: A review. 56(5):1197–1214, 2017.

W. Dornisch, J. Stöckler, and R. Müller. Dual and approximate dual basis functions for B-splines and NURBS-Comparison and application for an efficient coupling of patches with the isogeometric mortar method. *Computer Methods in Applied Mechanics and Engineering*, 316:449–496, 2017.

A. J. Herrema, N. M. Wiese, C. N. Darling, C. Ganapathysubramanian, A. Krishnamurthy, and M.-C. Hsu. A framework for parametric design optimization using isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316:944–965, 2017.

K. V. Kostas, A. I. Ginnis, C. G. Politis, and P. D. Kaklis. Shape-optimization of 2D hydrofoils using an Isogeometric BEM solver. *Computer-Aided Design*, 82:79–87, 2017.

T. M. Van Opstal, J. Yan, C. Coley, J. A. Evans, T. Kvamsdal, and Y. Bazilevs. Isogeometric divergence-conforming variational multiscale formulation of incompressible turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 316:859–879, 2017.

Z.-P. Wang, S. Turteltaub, and M. M. Abdalla. Shape optimization and optimal control for transient heat conduction problems using an isogeometric approach. *Computers and Structures*, 185:59–74, 2017.

M. Wu, B. Mourrain, A. Galligo, and B. Nkonga. H1-parametrizations of complex planar physical domains in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 318:296–318, 2017.

K. Bandara and F. Cirak. Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces. *Computer-Aided Design*, 95:62–71, 2018.

M.-J. Choi and S. Cho. Isogeometric configuration design optimization of shape memory polymer curved beam structures for extremal negative Poisson's ratio. *Structural and Multidisciplinary Optimization*, 58:1861–1883, 2018.

Q. Hu, F. Chouly, P. Hu, G. Cheng, and S. P. A. Bordas. Skew-symmetric Nitsche's formulation in isogeometric analysis: Dirichlet and symmetry conditions, patch coupling and frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 341:188–220, 2018.

B. Marussig and T. Hughes. A Review of Trimming in Isogeometric Analysis: Challenges, Data Exchange and Simulation Aspects. *Archives of Computational Methods in Engineering*, 25(4):1059–1127, 2018.

O. Mykhaskiv, M. Banovic, S. Auriemma, P. Mohanamuraly, A. Walther, H. Legrand, and J.-D. Müller. NURBS-based and Parametric-based Shape Optimisation with Differentiated CAD Kernel. *Computer-Aided Design and Applications*, 15(1), 2018.

F. Pérez-Arribas and R. Pérez-Fernández. A B-spline design model for propeller blades. *Advances in Engineering Software*, 118:35–44, 2018.

Y. Wang, Z. Wang, Z. Xia, and L. Hien Poh. Structural Design Optimization Using Isogeometric Analysis: A Comprehensive Review. *Computer Modeling in Engineering and Sciences*, 117(3):455–507, 2018.

Z.-P. Wang and L. H. Poh. Optimal form and size characterization of planar isotropic petal-shaped auxetics with tunable effective properties using IGA. *Composite Structures*, 201:486–502, 2018.

P. Antolin, A. Buffa, E. Cohen, J. F. Dannenhoffer, G. Elber, S. Elgeti, R. Haimes, and R. Riesenfeld. Optimizing Micro-Tiles in Micro-Structures as a Design Paradigm. *Computer-Aided Design*, 115:23–33, 2019a.

P. Antolin, A. Buffa, and M. Martinelli. Isogeometric Analysis on V-reps: First results. *Computer Methods in Applied Mechanics and Engineering*, 355:976–1002, 2019b.

P. Antolin, A. Buffa, R. Puppi, and X. Wei. Overlapping multi-patch isogeometric method with minimal stabilization. *arXiv*, pages 1–23, 2019c.

A. Arapakopoulos, R. Polichshuk, Z. Segizbayev, S. Ospanov, A. I. Ginnis, and K. V. Kostas. Parametric models for marine propellers. *Ocean Engineering journal*, 192: 106595, 2019.

F. De Prenter, C. V. Verhoosel, and E. H. Van Brummelen. Preconditioning immersed isogeometric finite element methods with application to flow problems. *Computer Methods in Applied Mechanics and Engineering*, 348:604–631, 2019.

D. Elfverson, M. Larson, and K. Larsson. A new least squares stabilized Nitsche method for cut isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 349:1–16, 2019.

A. J. Herrema, E. L. Johnson, D. Proserpio, M. C. H. Wu, J. Kiendl, and M.-C. Hsu. Penalty coupling of non-matching isogeometric Kirchhoff-Love shell patches with application to composite wind turbine blades. *Computer Methods in Applied Mechanics and Engineering*, 346:810–840, 2019.

T. Hirschler. *IsoGeometric Modeling for the Optimal Design of Aerostructures*. PhD thesis, Université de Lyon, 2019.

T. Hirschler, R. Bouclier, D. Dureisseix, A. Duval, T. Elguedj, and J. Morlier. A dual domain decomposition algorithm for the analysis of non-conforming isogeometric Kirchhoff-Love shells. *Computer Methods in Applied Mechanics and Engineering*, 357: 112578, 2019a.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. Isogeometric sizing and shape optimization of thin structures with a solid-shell approach. *Structural and Multidisciplinary Optimization*, 59(3):767–785, 2019b.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. The embedded isogeometric Kirchhoff–Love shell: From design to shape optimization of non-conforming stiffened multipatch structures. *Computer Methods in Applied Mechanics and Engineering*, 349:774–797, 2019c.

L. F. Leidinger, M. Breitenberger, A. M. Bauer, S. Hartmann, R. Wüchner, K.-U. Bletzinger, F. Duddeck, and L. Song. Explicit dynamic isogeometric B-Rep analysis of penalty-coupled trimmed NURBS shells. *Computer Methods in Applied Mechanics and Engineering*, 351:891–927, 2019.

M. Occelli, T. Elguedj, S. Bouabdallah, and L. Morançay. LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis. *Advances in Engineering Software*, 131:166–185, 2019.

O. Weeger, B. Narayanan, and M. L. Dunn. Isogeometric shape optimization of nonlinear, curved 3D beams and beam structures. *Computer Methods in Applied Mechanics and Engineering*, 345:26–51, 2019.

A. Buffa, J. Corno, C. De Falco, S. Schöps, and R. V. Hernández. Isogeometric mortar coupling for electromagnetic problems. *SIAM Journal on Scientific Computing*, 42(1): B80–B104, 2020.

M. Chasapi, W. Dornisch, and S. Klinkel. Patch coupling in isogeometric analysis of solids in boundary representation using a mortar approach. *International Journal for Numerical Methods in Engineering*, 121(14):3206–3226, 2020.

Dassault Systèmes SE. *Mesh tie constraints*. `https://abaqus-docs.mit.edu/2017/English/SIMACAECSTRefMap/simacst-c-tiedconstraint.htm`, 2020. (Accessed: 2022-01-06).

T. Djourachkovitch. *Conception de matériaux micro-architecturés innovants : Application à l'optimisation topologique multi-échelle*. PhD thesis, Université de Lyon, 2020.

X. Du, G. Zhao, W. Wang, and H. Fang. Nitsche's method for non-conforming multipatch coupling in hyperelastic isogeometric analysis. *Computational Mechanics*, 65(3):687–710, 2020.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. A New Lighting on Analytical Discrete Sensitivities in the Context of IsoGeometric Shape Optimization. *Archives of Computational Methods in Engineering*, 28:2371–2408, 2020.

E. L. Johnson and M.-C. Hsu. Isogeometric analysis of ice accretion on wind turbine blades. *Computational Mechanics*, 66:311–322, 2020.

L. Leonetti, F. S. Liguori, D. Magisano, J. Kiendl, A. Reali, G. Garcea, and H. Chi. A robust penalty coupling of non-matching isogeometric Kirchhoff-Love shell patches in large deformations. *Computer Methods in Applied Mechanics and Engineering*, 371: 113289, 2020.

SAFRAN. *LEAP-1A*, safran media library photo. `https://medialibrary.safran-group.com/Photos/media/282159`, 2020. (Accessed: 2021-12-09).

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, . Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, and P. van Mulbregt. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3):261–272, 2020.

L. Coradello, J. Kiendl, and A. Buffa. Coupling of non-conforming trimmed isogeometric Kirchhoff–Love shells via a projected super-penalty approach. *Computer Methods in Applied Mechanics and Engineering*, 387:114187, 2021.

McNeel. *Rhino - Rhinoceros 3D*. `https://www.rhino3d.com/`, 2021. (Accessed: 2022-02-09).

T. Pasch, L. F. Leidinger, A. Apostolatos, R. Wüchner, K.-U. Bletzinger, and F. Duddeck. A priori penalty factor determination for (trimmed) NURBS-based shells with Dirichlet and coupling constraints in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 377:113688, 2021.

B. D. Upadhyay, S. S. Sonigra, and S. D. Daxini. Numerical analysis perspective in structural shape optimization: A review post 2000. *Advances in Engineering Software*, 155, 2021.

Acronyms

BEM Boundary Element Method.

BFGS Broyden-Fletcher-Goldfarb-Shanno.

CAD Computer-Aided Design.

DSA Design Sensitivity Analysis.

FE Finite Element.

FEA Finite Element Analysis.

FEM Finite Element Method.

FFD Free-Form Deformation.

GA Genetic Algorithm.

HP High Pressure.

IGA IsoGeometric Analysis.

KKT Karush-Kuhn-Tucker.

LEAP Leading Edge Aviation Propulsion.

NURBS Non-Uniform Rational B-Splines.

PSO Particle Swarm Algorithm.

SA Simulated Annealing.

SQP Sequential Quadratic Programming.

Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **https://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	**M. Stéphane DANIELE** C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **https://edeea.universite-lyon.fr** Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	**M. Philippe DELACHARTRE** INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 **philippe.delachartre@insa-lyon.fr**
E2M2	**ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION** **http://e2m2.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	**M. Philippe NORMAND** Université Claude Bernard Lyon 1 UMR 5557 Lab. d'Ecologie Microbienne Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://ediss.universite-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	**Mme Sylvie RICARD-BLUM** Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 **sylvie.ricard-blum@univ-lyon1.fr**
INFOMATHS	**INFORMATIQUE ET MATHÉMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	**M. Hamamache KHEDDOUCI** Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 **hamamache.kheddouci@univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	**M. Stéphane BENAYOUN** Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 **stephane.benayoun@ec-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	**M. Jocelyn BONJOUR** INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **https://edsciencessociales.universite-lyon.fr** Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	**M. Christian MONTES** Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

FOLIO ADMINISTRATIF

THÈSE DE L'UNIVERSITÉ DE LYON OPERÉE AU SEIN DE L'INSA LYON

NOM : GUERDER — DATE de SOUTENANCE : 20/07/2022

Prénom : Marie

TITRE : **IsoGeometric Analysis and Shape Optimisation of Aircraft Compressor Blades**

NATURE : Doctorat — Numéro d'ordre : 2022LYSEI069

École doctorale : MEGA (ED 162) – Mécanique, Énergétique, Génie Civil, Acoustique

Spécialité : Mécanique

RÉSUMÉ

Relever le défi que représente l'incorporation de la modélisation géométrique et l'analyse numérique est un des objectifs de l'Analyse IsoGéométrique. Un des aspects principaux permettant d'adresser cette question est l'utilisation des mêmes fonctions mathématiques – à savoir, les B-Splines Rationnelles Non-Uniformes – pour décrire la géométrie et pour servir de support afin de résoudre l'analyse. L'un des avantages des fonctions NURBS est qu'elles bénéficient d'une continuité supérieure en comparaison aux polynômes de Lagrange, et des maillages plus grossiers peuvent être utilisés, réduisant le temps nécessaire pour l'analyse. Concernant l'optimisation de forme, l'IGA présente l'avantage de fournir un modèle qui est compatible avec les logiciels de Conception Assistée par Ordinateur, sans traitement supplémentaire. La conception et la fabrication de moteurs d'avions repose largement sur les méthodes numériques, et peut donc bénéficier des propriétés avantageuses de l'IGA. Des problématiques spécifiques émergent dans ce contexte industriel, parmi lesquelles la définition volumique des parties tournantes telles que les aubages. L'objet de ces travaux est de proposer un cadre complet pour la définition, l'analyse et l'optimisation de forme d'aubages de moteurs d'avions, en utilisant l'IGA. En s'appuyant sur une géométrie d'aubage industriel, nous proposons une procédure permettant d'en construire un modèle B-Spline volumique compatible pour l'analyse, en assurant la précision géométrique de ce dernier ainsi que la régularité du paramétrage. L'optimisation de forme est réalisée en utilisant les coordonnées spatiales des points de contrôle comme variables de *design*. La réponse mécanique de la structure est calculée à l'aide du code IGA *open-source* Yeti. Le reste de l'assemblage, incluant la plateforme et le pied de l'aubage, est pris en compte grâce à une approche de type *mortar* pour le couplage faible des patchs. Une formulation d'élément solide immergé a été développée durant cette étude, permettant de modéliser de façon précise le congé raccordant la pale à sa plateforme. De plus, la compatibilité géométrique est garantie aux interfaces entre patchs adjacents lors de la mise à jour de la forme de la pale durant la procédure d'optimisation de forme. Les résultats obtenus démontrent l'efficacité de la méthode et sa pertinence pour la conception et l'optimisation de forme d'aubages de moteurs d'avions.

Mots clés : Analyse IsoGéométrique, Optimisation de Forme, Aubages de Compresseur, B-Splines Trivariables, Méthode Immergée, Couplage.

ABSTRACT

Uniting the workflows of geometric design and numerical analysis is one of the challenging aims of IsoGeometric Analysis. Such a goal is addressed by using the same mathematical functions — namely, Non-Uniform Rational B-Splines — to describe the geometry and to serve as a support to solve the analysis. Amongst other advantages, NURBS functions benefit from a higher continuity with regards to Lagrange polynomials, and coarser meshes can be used, reducing the analysis time. When it comes to shape optimisation, IGA offers the advantage of providing a model that is compatible with Computer Aided Design software, without further processing. The aircraft engine design and manufacturing industry widely uses numerical methods, and hence can benefit from the advantageous features of IGA. Specific concerns arise in this industrial context, the volumetric definition of spinning parts such as blades being a prominent one. The purpose of this work is to propose a complete framework for the design, analysis and shape optimisation of aircraft engine blades using IGA. Using an actual industrial blade geometry, we propose a procedure to reconstruct a B-spline analysis-suitable volumetric model of the blade, ensuring its geometric accuracy and parametrisation regularity. Shape optimisation is performed using the spatial coordinates of control points as design variables. The mechanical response of the structure is computed using the open-source IGA code Yeti. The rest of the assembly, including the platform and tenon of the blade, is considered using a mortar approach for weak patch coupling. An embedded solid element formulation was developed during this study, enabling accurate modelling of the fillet linking the blade to its platform. In addition, it guarantees the geometric compatibility of the interfaces between adjacent patches during shape updates in the course of the shape optimisation process. The results demonstrate the efficiency of the method and its relevance for industrial aircraft engine blade design and shape optimisation.

Keywords: IsoGeometric Analysis, Shape Optimisation, Compressor Blades, Trivariate B-Splines, Embedded Method, Coupling.

Laboratoire de recherche : Laboratoire de Mécanique des Contacts et des Structures (LaMCoS)

Directeur de thèse : ELGUEDJ, Thomas

Président du jury : BOUCARD, Pierre-Alain

Composition du jury : BOUCLIER, Robin — RASSINEUX, Alain — BOUCARD, Pierre-Alain
CHASAPI, Margarita — ELGUEDJ, Thomas — TOUZEAU, Josselyn
DUVAL, Arnaud — DAM, Damien — RANNOU, Johann