Isogeometric modeling for the optimal design of aerostructures

Thibaut Hirschler

N° d'ordre NNT : 2019LYSEI092

THÈSE de DOCTORAT de l'UNIVERSITÉ 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 14/11/2019 par

Thibaut Hirschler
Ingénieur ENSEIRB-MATMECA

IsoGeometric Modeling for the Optimal Design of Aerostructures

Devant le jury composé de

RIXEN, Daniel J.	Professor	TUM Munich	Président
KIENDL, Josef	Associate Professor	NTNU	Rapporteur
BOUCARD, Pierre-Alain	Professeur	ENS Paris-Saclay	Rapporteur
COLLIN, Annabelle	Maître de Conférences	Enseirb-Matmeca	Examinatrice
EYHERAMENDY Dominique	Professeur	Centrale Marseille	Examinateur
ELGUEDJ, Thomas	Professeur	INSA Lyon	Directeur de thèse
MORLIER, Joseph	Professeur	ISAE Supaéro	Invité et encadrant
BOUCLIER, Robin	Maître de Conférences	INSA Toulouse	Invité et encadrant
DUVAL, Arnaud	Ingénieur de Recherche	INSA Lyon	Invité et encadrant
TOUZEAU, Josselyn	Ingénieur-Docteur	Safran Aircraft Engines	Invité

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage **secretariat@edchimie-lyon.fr** INSA : R. GOURDON	M. Stéphane DANIELE Institut de recherches sur la catalyse et l'environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 Avenue Albert EINSTEIN 69 626 Villeurbanne CEDEX **directeur@edchimie-lyon.fr**
E.E.A.	**ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec. : M.C. HAVGOUDOUKIAN **ecole-doctorale.eea@ec-lyon.fr**	M. Gérard SCORLETTI École Centrale de Lyon 36 Avenue Guy DE COLLONGUE 69 134 Écully Tél : 04.72.18.60.97 Fax 04.78.43.37.17 **gerard.scorletti@ec-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 INSA : H. CHARLES **secretariat.e2m2@univ-lyon1.fr**	M. Philippe NORMAND UMR 5557 Lab. d'Ecologie Microbienne Université Claude Bernard Lyon 1 Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69 622 Villeurbanne CEDEX **philippe.normand@univ-lyon1.fr**
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTÉ** **http://www.ediss-lyon.fr** Sec. : Sylvie ROBERJOT Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 INSA : M. LAGARDE **secretariat.ediss@univ-lyon1.fr**	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 Avenue Jean CAPELLE INSA de Lyon 69 621 Villeurbanne Tél : 04.72.68.49.09 Fax : 04.72.68.49.16 **emmanuelle.canet@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. Luca ZAMBONI Bât. Braconnier 43 Boulevard du 11 novembre 1918 69 622 Villeurbanne CEDEX Tél : 04.26.23.45.52 **zamboni@maths.univ-lyon1.fr**
Matériaux	**MATÉRIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction **ed.materiaux@insa-lyon.fr**	M. Jean-Yves BUFFIÈRE INSA de Lyon MATEIS - Bât. Saint-Exupéry 7 Avenue Jean CAPELLE 69 621 Villeurbanne CEDEX Tél : 04.72.43.71.70 Fax : 04.72.43.85.28 **jean-yves.buffiere@insa-lyon.fr**
MEGA	**MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr** Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bât. Direction **mega@insa-lyon.fr**	M. Jocelyn BONJOUR INSA de Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69 621 Villeurbanne CEDEX **jocelyn.bonjour@insa-lyon.fr**
ScSo	**ScSo*** **http://ed483.univ-lyon2.fr** Sec. : Véronique GUICHARD INSA : J.Y. TOUSSAINT Tél : 04.78.69.72.76 **veronique.cervantes@univ-lyon2.fr**	M. Christian MONTES Université Lyon 2 86 Rue Pasteur 69 365 Lyon CEDEX 07 **christian.montes@univ-lyon2.fr**

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Remerciement

Merci est le premier mot je souhaite laisser dans ce manuscrit. Tant de personnes ont contribué à ces travaux, et cela, bien de différentes manières ; je me dois de les remercier chaleureusement. J'ai été marqué par la gentillesse et les encouragements portés à mon égard tout au long de cette belle aventure qu'est le doctorat.

Merci infiniment à vous mes encadrants ; Thomas Elguedj, Joseph Morlier, Robin Bouclier, et Arnaud Duval. Votre bienveillance m'a permis de m'épanouir pleinement. Merci pour la confiance que vous m'avez accordée rapidement. J'ai apprécié travailler avec chacun d'entre vous. Merci pour ce que vous avez apporté, chacun à votre manière. Vous avez su me pousser à donner le meilleur de moi sans pour autant m'étouffer sous une charge trop importante de travail. Bien au contraire, vous m'avez encouragé à prendre des vacances, à faire des breaks, à changer d'air lorsque cela était nécessaire. Voilà un peu un témoignage de votre bienveillance. Merci pour l'autonomie que vous m'avez accordée ; j'ai pleinement apprécié pouvoir proposer mes idées et pouvoir explorer de nouvelles pistes de façon autonome. Un grand Merci pour tout.

Merci à chaque personne venue lors de la soutenance. J'adresse un merci particulier aux membres du jury. Tout d'abord, je remercie profondément Pierre-Alain Boucard et Josef Kiendl d'avoir si gentiment accepté de rapporter cette thèse. C'est avec beaucoup de reconnaissance, d'humilité, et de fierté que j'ai reçu vos rapports. Merci d'avoir apporté vos regards expérimentés sur mes travaux. Un grand merci à Daniel Rixen d'avoir si bien endossé le rôle de Président du jury ; merci pour l'enthousiasme et la gentillesse témoignés durant la soutenance. Je remercie également Annabelle Collin, Dominique Eyheramendy, et Josselyn Touzeau d'avoir fait le déplacement jusqu'à Lyon. Merci pour votre intérêt concernant mes travaux, pour vos contributions lors de la soutenance, merci pour les échanges. Vous avez tous participé au succès de cette journée. Finalement, je souhaite également te remercier Damien Crozes d'avoir été l'initiateur de ce doctorat en me mettant en relation avec Robin et Joseph lors que mon stage de fin d'études à Toulouse, merci de m'avoir encouragé à m'orienter vers le doctorat.

Merci à toutes les personnes côtoyées durant ces trois années de thèse. J'ai vraiment apprécié l'atmosphère qui règne au LaMCoS, et cela, en premier lieu, grâce à toutes les belles personnes qui remplissent ce laboratoire. Vous, les doctorants, avez été précieux. Je crois qu'on s'est bien *marré !* Merci à mes deux compagnons Matthieu et Tristan (*a.k.a.* Sergent Mar-quart'). Ca a été un bonheur de passer ces trois ans avec vous. Matthieu,

merci pour ta joie de vivre inarrêtable. Tristan, il est difficile de te décrire tellement tu es spécifique ; donc merci pour qui tu es, pour ton franc-parler qui nous fait tant sourire. Marie, je te souhaite beaucoup de bonheur et de réussite pour ta thèse. Ne te braque pas si tes collègues ne veulent pas entendre que l'IGA c'est top. Merci à tous les autres doctorants, même si l'IGA ne vous convainc pas. Merci à Florian, Nicolas, et avant vous, Arthur ainsi que l'imprévisible Pipo, d'avoir partagé votre îlot de bureaux avec moi. Nico et Florian, merci pour les fous rires indénombrables. Merci Médéric, Zhaofeng, Haoming, Pierre, MHD, et avant vous, Fatima, et Ye, d'avoir merveilleusement complété ce beau bureau (sens figuré bien-entendu, *rires*) et d'en avoir fait un lieu sympathique. Un grand merci à toi Yvan pour la motivation que tu communiques, merci pour ta simplicité. Tes capacités de travail sur-humaines imposent le respect. Ca a été chouette de commencer et finir nos thèses ensemble. Merci à vous les doctorants du bureau *d'à-côté*, j'ai nommé Zi, Tristan, Thomas, Efoé, Ethel, Alexis, Du, et Meng. J'ai apprécié vous rencontrer et passer de bons moments ensemble. Meng, ça a été sympa de finir ce doctorat en même temps que toi. Ca fait du bien de voir qu'on est pas le seul à se prendre la tête sur les slides à quelques jours (heures ?) de la soutenance. Merci à chacun pour tout ; la bonne ambiance et l'entraide ont pleinement participé à mon épanouissement et la réussite de cette thèse. Ca a été précieux.

Merci à vous, toutes les personnes extraordinaires rencontrées tout au long de mon parcours, depuis le Geisberg en Alsace (littéralement, la montagne des chèvres) jusqu'à Lyon. Merci aux Enseirbiens-Matmecanos, Julien, Luc, et Gautier d'être venu à ma soutenance. Ca m'a fait extrêmement plaisir de vous voir. Merci à toi Valentin pour ton amitié malgré la distance. Steven, ça a été sympa de te croiser au CSMA à Giens, on aura bien rigolé ! Merci également à vous, Léon et Aude, ça a été précieux de vous avoir. Merci pour les pauses au parc *avant la tête d'Or* ; Léon, ton café est extra. Ca a été bon de pouvoir s'encourager et se soutenir. Merci à vous, les personnes rencontrées au GBU, pour ces bons moments partagés, pour ces discussions qui donnent du sens à la vie. J'aurais du mal à tous vous nommer. Vos prières, vos encouragements, vos marques d'attention m'ont porté. Je ne peux qu'être reconnaissant pour tout cela. Simon, tu es un ami précieux, merci pour ton amitié qui dure. Tu es exceptionnel et je suis honoré de te compter parmi mes proches amis. Acsa, Ando, maintenant Yannick, et vous tous les autres amis de l'église de la rue Louis, merci beaucoup pour vos marques d'attention que vous m'avez adressées tout au long de cette thèse ; cela m'a porté. Comme la vie en général, la thèse est faite de hauts et de bas. Merci à chacun pour votre présence qui m'a permis de sourire malgré les difficultés, et de partager les bons moments. J'espère pouvoir vous rendre un peu de tout ce que vous m'avez apporté.

Merci à vous mes proches, ma famille. Je vous dédicace ce travail. Papa, Maman, merci pour tout ce que vous m'avez transmis. Vous m'avez toujours encouragé et soutenu, cela aussi est précieux. Merci à vous mes frangins, Benoît, Quentin et Lucas, pour votre amour et tous ces moments vécus ensemble. Merci Quentin pour tes *tips* de graphistes, cela m'a permis d'embellir mes figures. Merci à la jeune femme la plus extra

de cette planète, ma Sedera, pour tout ce que tu m'as apporté. Tu as été exceptionnelle durant cette thèse et plus particulièrement à la fin, dans les moments plus délicats. Ta présence à mes côtés m'a porté, tu m'as donné force et courage. Merci pour tout l'amour que tu me témoignes chaque jour, tu es la meilleure de toutes.

Finalement, la thèse a été très formatrice, et cela bien au-delà des compétences purement scientifiques. Je n'estime pas pouvoir mesurer tout ce que cela m'a apporté. Merci Emmanuel pour ta fidélité que j'ai pu expérimenter tout au long de cette thèse. On te croit souvent lointain ou absent mais tu es bien plus près de nous que nous le pensons. Le doctorat a été une super expérience, j'encourage quiconque se posant la question à se lancer dans cette belle aventure !

Thibaut Hirschler

Résumé

Concevoir des structures au regard de leur comportement mécanique est une tâche importante en ingénierie. Cependant, cette phase de conception peut s'avérer longue et fastidieuse lorsque les relations de cause à effet ne sont pas correctement identifiées. Plus spécifiquement, il est généralement difficile d'obtenir la meilleure forme d'une structure (la forme optimale), car cela fait appel à de nombreuses compétences. En effet, des outils avancés pour la modélisation géométrique sont nécessaires pour représenter fidèlement la structure et pour explorer une large variété de formes. A cela s'ajoute le besoin d'une méthode de calcul de structures compétente et rapide afin de réduire la durée du processus d'optimisation. Un lien étroit entre le modèle géométrique et celui d'analyse est souhaité car tous deux sont amenés à communiquer successivement. Ainsi, l'Analyse IsoGéométrique apparaît comme un outil adéquat pour les problèmes d'optimisation paramétrique de formes. En effet, cette méthode s'appuie sur un modèle unique pour décrire fidèlement la géométrie et pour effectuer l'analyse. Entièrement basée sur le concept de l'AIG, nous construisons une stratégie de conception optimale de la forme pour les structures raidies, omniprésentes en aéronautique. Nous présentons une formulation massif coque permettant d'imposer des variations continues d'épaisseur à une structure mince. Nous formulons des sensibilités analytiques pour les éléments isogéométriques standards et coques. Aussi, nous présentons une approche immergée pour appliquer des changements de forme aux structures raidies. Du point de vue de l'analyse, nous introduisons une formulation coque adaptée aux compositions géométriques et nous construisons un algorithme dédié aux cas de discrétisations non-conformes en s'appuyant sur les méthodes de couplage Mortar et de décomposition de domaine. *In fine*, tous ces développements sont fusionnés et permettent de traiter divers exemples avec des niveaux de complexité croissants.

Note : *ce document est intégralement écrit en anglais. Un résumé détaillé écrit en français est donné à la fin du document, en annexe A.*

Abstract

Designing structural parts against the material limits, the impact of loads, and many other constraints, is of standard interest in engineering. However, improving the design of a structure can be long and drawn out, especially when a clear understanding of cause-effect relationships is missing. Finding the best possible design, namely the optimal design, is a complex task because it requires several competences. Usually, efficient geometric modeling is needed to accurately represent the structure. Conjointly, the geometric model should provide high flexibility during the design exploration. In addition, structural analysis must be fast enough to shorten the overall process. Besides, for the sake of compactness, a close connection between the geometric model and the structural analysis seems essential. Finally, all modeling choices are deeply related, and thus, they should be thought and built accordingly to the others. Therefore, IsoGeometric Analysis appears as a powerful tool for structural optimization since it uses a unique model with both high quality geometric and analysis properties. Here, we present a compact framework built on the core idea of IGA. We strive to construct unified models with new opportunities for structural design with a direct application to stiffened Aerostructures. More specifically, we present a solid-shell approach to impose continuous thickness variations. We formulate analytical sensitivities for standard and shell formulations. Then, we introduce an embedded technique that enables to impose complex shape updates. From the analysis point of view, we build a shell formulation dedicated to geometric compositions and we design a specific solver based on Domain Decomposition methods and Mortar approach for the coupling of non-conforming discretizations. Different examples with increasing level of complexity show the performances of the adopted methodologies.

Contents

Chapter 1

Introduction

1.1 Motivations and Objectives

Numerical simulation occupies an important place in the engineering design process. The numerical results guide numerous decisions regarding a final product. The impact of these numerical methods on the final product is important and the choice of a particular method against others can lead to a very unique design. If one gives the task of designing a specific part, defined by a design brief, to two different engineers, then one inevitably ends up with two different final products. Of course it varies for example with the experience of each designer. A confirmed engineer will not make the same choices than a junior engineer. However, this is not the only source of the differences. The numerical methods used by both of them can be one of the major sources of these differences. In the case where two different numerical approaches are used by each of them, then one could be certain that the final parts would differ. The efficiency of the employed method is crucial. For example, if one numerical method is very fast and the other is slow (but giving the same result), then using the fastest method enables to explore much more configurations. At the end, it leads to a better final design. The employed numerical methods need to be versatile such that the overall design process is shorten. They also need to be accurate such that the final product has the desired behavior.

In all industrial sectors, constant improvements are sought. This is particularly true in aeronautics where the design of aircraft competes in a race for technological innovation. A particular interest is to design aircraft with lower fuel consumption. Potential improvements can be provided in many ways and we set out in this thesis to explore one of them. It concerns the shape of the structural parts, namely the stiffeners. In fact, stiffened structures are omnipresent in aeronautics. These structures are obtained by the assembly of a main part (usually denoted the skin or the panel) with sub-parts as stringers and ribs (the stiffeners). The fastening of the stiffeners with the skin gives a final structure with a high stiffness to weight ratio. The construction of the wings or of the fuselage are good examples of stiffened structures. Classical designs contain a lot of

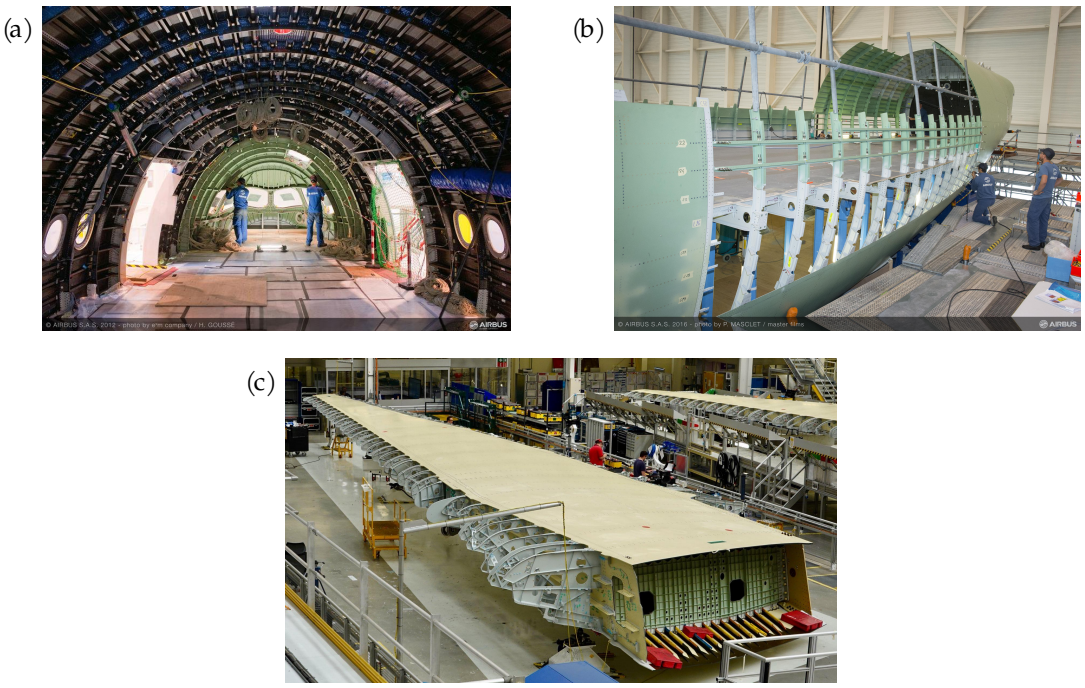
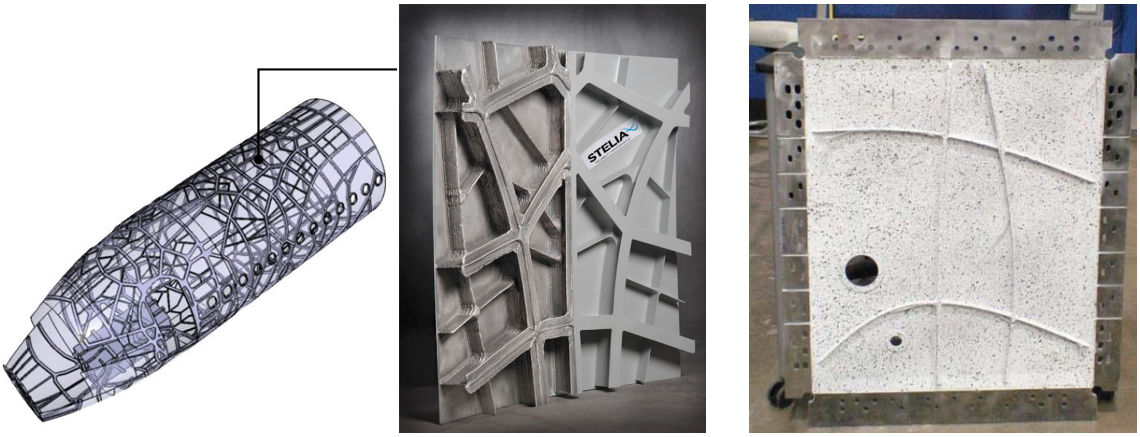


Figure 1.1 – Stiffened structures are omnipresent in aeronautics: the fuselage (photo (a) from [1] and photo (b) from [2]) and the wing (photo (c) from [3]) are built with this principle.

straight and aligned stiffeners uniformly distributed along the main structure (wing or fuselage). Examples are shown in figure 1.1.

Recent works motivate the development of new innovative stiffened structures. It seems promising to design stiffeners with new types of shapes. Particularly, defining curvilinear stiffeners instead of straight ones can further improve the mechanical behavior of the overall stiffened structures [4–14]. Figure 1.2 highlights what could be, in the future, the aerostructures based on free-form stiffeners. The shape of a structure has a high influence on its mechanical behavior. For example, the shape of an egg enables to show great resistance when subjected to internal pressure. Possible great improvements regarding the structures of airplanes could be obtained by designing stiffeners with new innovative shapes.

In order to design these new stiffened structures, dedicated numerical methods are required. This thesis focuses on the development of new methods that enable to design these innovative stiffened structures by improving, for example, the shape and the position of the stiffeners. The goal is to built a general framework which is flexible and robust. It should be flexible in order to tackle a lot of different examples using the same strategy. For example, the method should be applicable to the design of the fuselage, the wing, or any other part. The setting of the process should be easy to put in place such that numerous cases can be performed. Robustness is sought in order to get trustful results. It is also required to get a framework that is general and usable for as many



(a) (b)

Figure 1.2 – Innovative stiffened structures with curvilinear stiffeners: (a) from [15], and (b) from [16].

cases as possible.

We choose to build our framework based on IsoGeometric Analysis. IGA is a novel numerical method for structural analysis (and numerical simulation in general) which relies on geometric models to perform the simulation. It has gained relevance since its introduction by Hughes et al. [17] in 2005. We will start by giving an overview of this approach along with justifications for using it in the presented context. Then step by step we introduce new strategies for the shape optimization of aerostructures. We believe that we end up with a framework that answers to several of the initial objectives. Apart from the application to stiffened structures, this thesis deals with the shape optimization of shell structures. Contributions to this research field can be found throughout the document (see the list of publications and scientific presentation in the following section 1.3).

1.2 Outline

More specifically, the document is organized as follows. Chapter 2 motivates IsoGeometric Analysis in the perspective of structural shape optimization. Basics regarding both IGA and shape optimization are presented. Then, more theoretical content is given in chapter 3 which focuses on the shape optimization of shells. We present the use of a solid-shell formulation for this purpose. It is presented in conjunction with an isogeometric Kirchhoff–Love shell. Analytical sensitivities are developed for both types of shell. We show the different benefits brought by each of them and how they can be combined. In order to tackled more complex structures, we deal with non-conforming multipatch models in chapter 4. We introduce a Mortar coupling for Kirchhoff–Love shells. We design a dual Domain Decomposition solver to efficiently analyze these mul-

tipatch models with complex shell junctions. Chapter 5 deals with the difficulty in handling these non-conforming multipatch models for performing shape optimization. We present an embedded methodology which gives the possibility to generate stiffened structures in a suitable way. It offers great flexibility regarding the imposition of the shape variations. A specific shell formulation, namely the embedded Kirchhoff–Love shell, is established. Several examples of shape optimization of stiffened structures are presented in chapter 6. It enables to show how all the developed approaches are related and how they can be concatenated into one final framework. In the final chapter, we summarize the major results of the thesis. Further prospects and possible improvements of the study are discussed.

1.3 Contributions

This thesis led to the following publications and presentations. Additional resources will probably be added to this list, especially regarding the analytical sensitivities presented in this thesis.

Articles in scientific journals

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Isogeometric sizing and shape optimization of thin structures with a solid-shell approach". In: *Structural and Multidisciplinary Optimization* 59.3 (2019), pp. 767–785.

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". In: *Computer Methods in Applied Mechanics and Engineering* 349 (2019), pp. 774–797.

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". In: *Computer Methods in Applied Mechanics and Engineering* 357 (2019), p. 112578.

Conference Articles

T. Hirschler, R. Bouclier, A. Duval, D. Crozes, T. Elguedj, and J. Morlier. "Analyse Isogéométrique pour les problèmes d'Optimisation de Forme des Structures Coques". In: *13e colloque national en calcul des structures*. Giens, Var, France, 2017.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Un modèle isogéométrique aux discrétisations non-conformes pour la conception optimale des structures raidies". In: *14e colloque national en calcul des structures*. Giens, Var, France, 2019.

Other Conference presentations

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Isogeometric Shape Optimization in the context of Solid-shell Models". In: *5th International Conference on Isogeometric Analysis*. Pavia, Italy, 2017.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Efficient Modelling for the Shape Optimisation of Shells and Stiffened Structures". In: *6th European Conference on Computational Mechanics*. Glasgow, Scotland, 2018.

T. Elguedj, T. Hirschler, R. Bouclier, A. Duval, and J. Morlier. "Isogeometric Shape Optimization for Innovative Design of Stiffened Structures". In: *13th World Congress in Computational Mechanics*. New-York city, USA, 2018.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Isogeometric Shape Optimization of Curvilinearly Stiffened Aerostructures". In: *6th International Conference on Isogeometric Analysis*. Austin, Texas, USA, 2018.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Coupling of non-conforming embedded isogeometric Kirchhoff-Love shells by a Mortar approach: towards the efficient shape optimization of Aeronautical structures". In: *8th International Conference on Coupled Problems in Science and Engineering*. Sitges (Barcelona), Spain, 2019.

T. Elguedj, T. Hirschler, R. Bouclier, A. Duval, and J. Morlier. "An Embedded Isogeometric Kirchhoff–Love Shell Formulation for the Shape Optimization of Non-conforming Multi-patch Structures". In: *15th U.S. National Congress on Computational Mechanics*. Austin, Texas, USA, 2019.

R. Bouclier, T. Hirschler, A. Duval, T. Elguedj, and J. Morlier. "An embedded isogeometric Kirchhoff–Love shell formulation for the shape optimization of non-conforming multi-patch and stiffened structures". In: *7th International Conference on Isogeometric Analysis*. Munich, Germany, 2019.

T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "A dual Domain Decomposition solver for analyzing non-conforming multipatch Kirchhoff–Love models based on Mortar coupling". In: *7th International Conference on Isogeometric Analysis*. Munich, Germany, 2019.

Chapter 2

Isogeometric Analysis as a powerful approach for Shape Optimization

The original idea behind IsoGeometric Analysis is to rely on geometric models to perform structural analysis. All this work follows this innovative mindset. This first chapter aims at motivating our interest for IGA. The theoretical background of this method along with underlying examples are presented. More precisely, the first task is to introduce main aspects regarding geometric modeling. We outline the basis of Isogeometric Analysis with the goal of motivating its use for shape optimization.

Contents

2.1 Design Optimization

2.1.1 A crucial objective in Engineering Design

Design optimization is a very common task during the product development even if it may not be called this way. In fact, when an engineer seeks to design a new product, he tries to develop it accordingly to a design brief. The final product should match the specifications in the best possible way. To that purpose, the engineer goes through several design stages in order to get a final product that satisfies all the requirements. In practice, it may be long and drawn out. The engineer often tries a lot of configurations before the final design is obtained. Usually, a long succession of attempts is needed: the final product is obtained by a trial-and-error learning process. The difficulty lies in the fact that the cause-effect relationships are sometimes hard to identify. The problem becomes more important when the engineer has no adequate resources to carry out the product development.

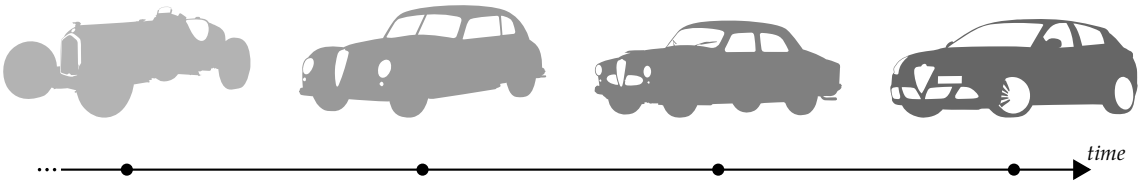


Figure 2.1 – Evolution of cars over time.

Fortunately, the engineer can rely on all the know-how acquired over time. For example, the engineer does not start from a blank page when designing a new car. All the works done for the previous car models provide valuable knowledge to the engineer. It helps him to take better decision during the design of the new car. The engineer also considers the defaults of the existing cars, in order to improve the new one. This process leads to the improvement of the product over time. In the case of cars, this evolution is clearly visible as illustrated in figure 2.1. Cars have been improved regarding a lot of criteria as comfort, security, fuel economy and efficiency, etc. Finally, this can be viewed as an Optimization Process: cars are continuously ameliorated to perform better than the precedents. At the end of the process, one should get the optimal product, *i.e.* the best car ever, the one that cannot be improved in any sense. Of course, for complex product as a car, getting the optimal one may not be possible. Firstly, several criteria can be inconsistent: a car can hardly be the most comfortable but also the fastest or the most practical. The consumer wishes may also vary between two people, and also over time. The technology is also evolving which constantly gives new ways to improve the product. Thus, getting the optimal car is quite utopian. Design improvement could be a more realistic term than design optimization.

Numerical simulation has participated to very significant improvements in every industry sector. It consists in building numerical models of a product and simulating

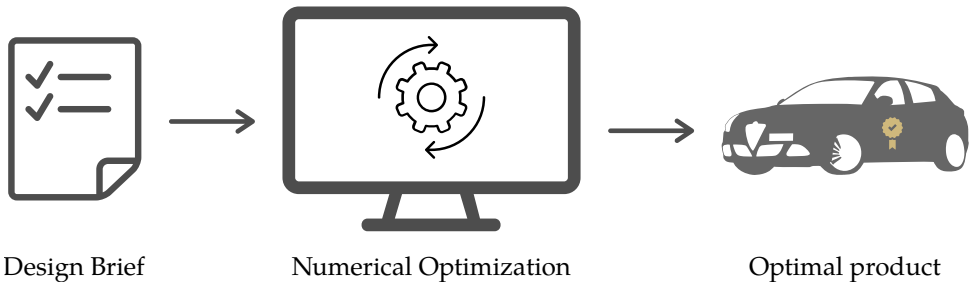


Figure 2.2 – Very schematic representation of numerical optimization.

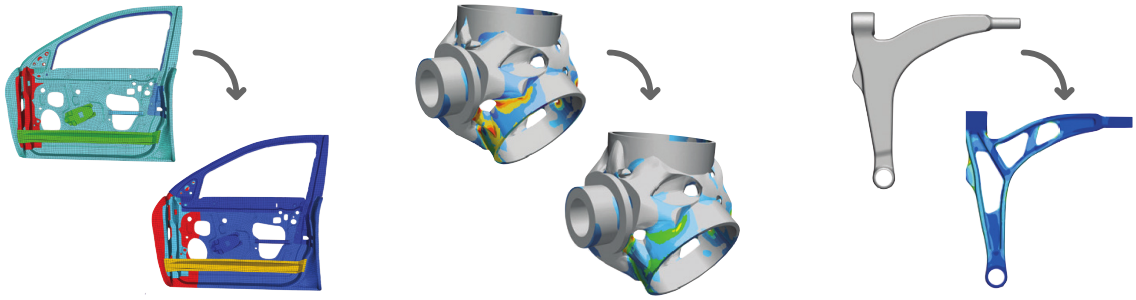
its behavior with the help of a computer. It avoids to build real prototypes which are replaced by virtual ones. In addition to be generally cheaper, using numerical models enables to study the influence of a large variety of parameters in a reduced amount of time. The trial-and-error learning process is done much faster. But even more interesting, since the numerical models enable *a priori* to get the response of a product with respect to any given set of input parameters, they give the possibility to find the optimal configuration.

It gives rise to numerical methods which aim at solving optimization problem in engineering. Obviously, even with a numerical model in hand, it is not feasible to try every set of parameters out in order to identify the best configuration. Indeed, the cost (in terms of time, money, etc.) of a numerical simulation is not null. Thus, specific optimization methods have been developed in order to appropriately use the numerical model to find the optimal configuration. The general idea of numerical optimization is illustrated in figure 2.2. As of now, we understand that the modeling choices made during the built of the numerical model are crucial. This work explores new modeling choices to deal with a very specific optimization problem which is commonly referred to as structural shape optimization. It consists in finding the optimal shape of a structure regarding its mechanical behavior.

2.1.2 Current methods for Structural Optimization

The modeling options of a structure are multiple. The most common technique to analyze an industrial structure is based on Finite Element Modeling [31–34]. But even in this framework of Finite Element Analysis, a multitude of modeling choices can be made which lead to different models of a same structure. Thus, several methods exist to perform structural optimization.

Types of structural optimization The different methods employed in structural optimization are usually categorized into one of the three principal classes, namely material or sizing optimization, shape optimization, and topology optimization [36]. Figure 2.3 presents several optimization problems solved with the commercial software Tosca [35]. Topology optimization aims at searching the best material distribution within a given



Sizing Optimization | Shape Optimization | Topology Optimization

Figure 2.3 – The three most common types of structural optimization problems [35].

domain. It is an interesting method to be used in the conceptual design stage since it enables to investigate large design configurations [36]. The rise of additive manufacturing gives the opportunity to directly manufacture the final design given by the topology optimization process. This leads to final product with stunning shapes and great mechanical performance [37, 38]. However, in case of more common manufacturing processes, additional geometric modeling steps are necessary to create a design that can effectively be manufactured. Sizing optimization comes into play during the detailed design stage, *i.e.* when the overall shape of the structure is already known. It deals with the optimization of internal parameters that does not change the geometry [36]. The most common problems solved by sizing optimization concerns the thickness distribution of shell structures [39], the cross sections of truss structure [40], or the material properties of a structure [41]. Shape optimization is in between sizing and topology optimization and can be performed at every stage of the design chain. The goal is to optimize the shape of a structure without changing its topology [36].

Obviously, structural optimization methods play a crucial role in the design of aeronautic structures [37, 42–48]. Sizing, shape and topology optimization problems are involved. An example of the design stages for the optimal design of the A380 wing box ribs are presented in [42, 44] and repeated in figure 2.4. By looking at the design chain in figure 2.4, one can predict that the overall process can be quite long. Especially, in this case the topology optimal result is only taken as a conceptual design. It is reinterpreted as a stiffened structure. This geometric modeling task can hardly be automatized and requires human intervention. The same step applies after the shape and sizing optimization where the height of the stiffeners and the thickness distribution are optimized. Human intervention is needed once again to generate the final geometric model of the ribs. The modeling choices at each step of the process have great impacts on the overall time. A close link between the models involved during for example, the structural analysis, the geometric modeling, the optimization processes, etc. is highly beneficial. In the study presented in figure 2.4, different software were repeatedly used (Altair OptiStruct, Altair HyperMesh, and ICAD, see [42] for more information).

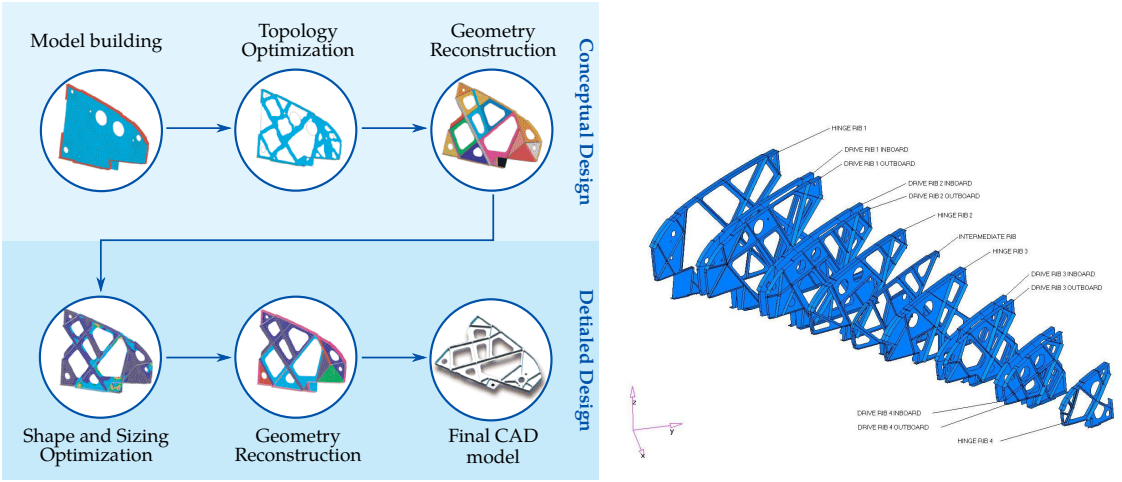


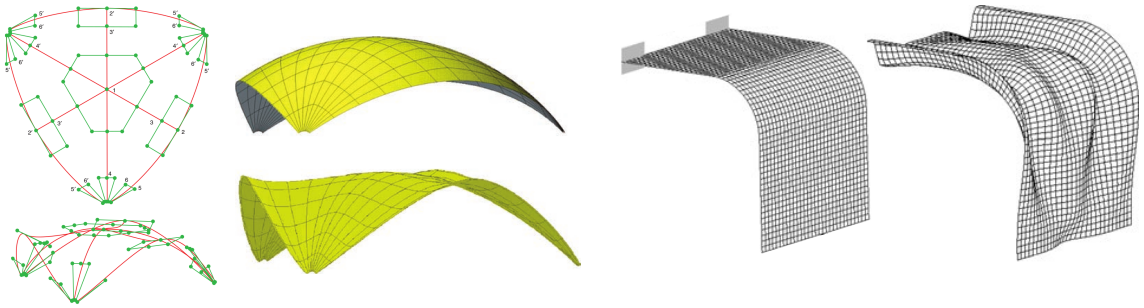
Figure 2.4 – Optimization during design stages of an aircraft component as presented by Krog et al. [42].

Structural shape optimization In this work we focus on shape optimization methods. Indeed, these approaches have shown great potential for designing stiffened structures [4–14]. Shape optimization has also been successfully applied to design specific aircraft parts as for example pressurized bulkheads [49]. Shape optimization methods are also intensively used in aircraft design [50].

However, the full potential of shape optimization for structural optimization seems not to have been yet fully tackled. For instance, if we go back to the optimization example of the wing box ribs (see figure 2.4 and source document [42]), one can notice that the shape optimization concerns the height of the stiffeners only. The curvature of the stiffeners, or the shape of the holes could have been additional features to integrate into the optimization problem. Krog et al. [42] mention that this was not implemented due to complexity and time restrictions. Thus, improvements can be done in many ways and this work seeks to address some remaining difficulties of shape optimization techniques.

Numerical approaches for the shape optimization of structures are not new and robust strategies have been presented in the eighties and early nineties [51–56]. These developments have led to two principal classes of methods: the node-based approaches, and the CAD-based approaches. The node-based method uses the nodes of an analysis model (*i.e.* the finite element mesh) as design parameters. The CAD-based method uses two models of the same structure: the Computer-Aided Design model that describes the geometry, and an analysis model to perform the structural analysis. Figure 2.5 presents examples (taken from [36]) of shape optimization problems solved with both approaches.

The first CAD-based approach has been presented by Imam [51] in order to circumvent mesh distortion problems arising when moving independently FEM nodes, and



CAD-based shape optimization | Node-based shape optimization

Figure 2.5 – Examples of CAD-based and node-based shape optimization as found in [36].

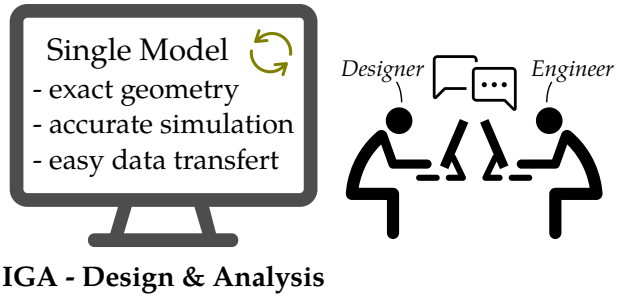
to reduce the number of design variables in comparison to node-based method. Having two separated models (one design model and one analysis model) has shown great benefits [52] and has been preferred over FEM-based methods for a specified time [56]. However, a major drawback of the method has restricted its deployment in design offices. Indeed, it requires a close link between the design model and the analysis model. To this purpose, very specific program needs to be dedicated [57]. For complex structures, the link between the geometric model and the finite element mesh is far from straightforward. Also, the repeated mesh generations during the resolution burdens this optimization process. Finally, CAD-based approaches are highly sensitive to the definition of the design model and the design parameters. Thus, more recently, node-based methods regain interest and efficient methods have been developed [58–61]. Interestingly, they reuse similar ingredients as CAD-based methods (for example spline functions in the sensitivity filtering). According to Bletzinger [36], both methods appear to merge. This is even more true with the contribution of IsoGeometric Analysis as shown latter on.

2.2 General Purposes and Motivations for IGA

2.2.1 The need for CAD-CAE integration

IsoGeometric Analysis has been introduced by Hughes et al. [17] in 2005 with a principal motivation: unify the fiels of Computer-Aided Design and Finite Element Analysis. The initial issue being raised during the introduction of IGA concerns the existing gap between these two disciplines. Following Cottrell et al. [62], the problem emerges from the fact that FEA and CAD have evolved independently during past decades even if they are closely linked during engineering designs. Engineering design involves, at one side, the designer who generates geometric models through CAD file, and at the other side, the stress analyst who builds the inputs accordingly to a FEA code. However, the increasing complexity of both geometric and analysis models is beginning to cause some drawbacks as they are based on different technologies.

(a) Traditional Engineering Design



(b) Better Engineering Design through CAD/CAE Integration

Figure 2.6 – IGA aims at improving engineering design by integrating Design and Analysis into a single model: designers and engineers can better collaborate.

The translation of the CAD file into the FEA inputs is not straightforward as shown in figure 2.6(a). It does not only concern the mesh generation but the integration of CAD and FEA requires an efficient creation of appropriate *simulation-specific* geometries [62]. In fact, the designer endeavors to represent as faithfully as possible a conceptual geometry. It means that the generated CAD model usually presents a lot of details as holes, fillets, and a lot of sub-parts as bearings, etc. Highly detailed CAD models are rarely exploitable for the simulation since it usually contains irrelevant details as viewed from the analyst. Thus, additionally to the mesh generation step, a lot of time is spent to generate analysis-suitable geometries where the initial CAD model is *defeatured* (see figure 2.6(a)). For sophisticated engineering systems, things turn out to be very tricky. Different simulations as for example structural mechanics, fluid dynamics, heat transfer, and many others, are performed. Each type of analysis has its own particular method, and thus different models are required for each of them. Figure 2.7 illustrates this point: for sophisticated engineering systems, for instance here an airplane, there are different visions of the same product. Thus, it follows that appropriate *simulation-specific* geometries are needed. For example, the aerodynamicist may focus on the outer geometry of the aircraft whereas the stress analyst is mainly concerned by the internal structure. Therefore, there is not an unique CAD model of the airplane but a lot of geometric models are required depending on the simulation to be performed.

According to Cottrell et al. [62], the translation to analysis-suitable geometries in addition to the meshing and the input setting for simulation codes, take *de facto* over 80%

Figure 2.7 – Engineering systems: each engineer has its own vision and requirement of the same product [adapted from: www.ruthmalan.com, artwork by C.W. Miller]

of the overall analysis time. It is clear that a stronger dialog between Computer-Aided Design and Computer-Aided Engineering is crucial. The engineer should be aware of what the designer does and vice versa. IGA brings this issue back to the table and has fostered fruitful discussions regarding a better CAD-CAE integration. As explained in figure 2.6(b), the goal of IGA can be seen as an attempt to ease the collaboration between the designers and the engineers by integrating into a single model, efficient geometric modeling possibilities and accurate simulation capabilities.

After almost 15 years, the initial paper [17] has been cited by more than 8000 research works and the interest for IGA keeps going up. It proves that it has raised a significant impact, at least in research. However, one may wonder if we achieve the process depicted in figure 2.6(b)? Today, the answer is still no since it does not exist commercial software that enable a fully integration of Design and Analysis. But things are starting to move forward in the industry. IGA has started to be integrated to well-known Finite Element codes as for example LS-DYNA [63] or RADIOSS [64]. However, a key point remains a challenge: no robust preprocessing tools are available yet. The difficulty lies in some major choices that have been made regarding CAD geometries in the past. We can list at least two sources of trouble: trimming procedures and Boundary-Representation. Both procedures have received special attention from the analysis side [65] and are, among others, not yet fully tackled. The issue of integrating CAD and CAE requires a lot of effort of researchers from both fields. More than ever, both disciplines need to work together in order to handle this common goal and defining the future of engineering design.

2.2.2 A suitable approach for parametric Shape Optimization

So far, we have not given theoretical explanations on how goes an optimization process. We present all the key ingredients in chapter 3. However, if one is familiar with numerical simulation and structural analysis, then one can easily suspect the benefit of combining high quality geometric representation and efficient analysis capabilities into one single model. Indeed, without providing details yet, shape optimization involves an iterative process with three principal steps as explained in Bletzinger et al. [66] and

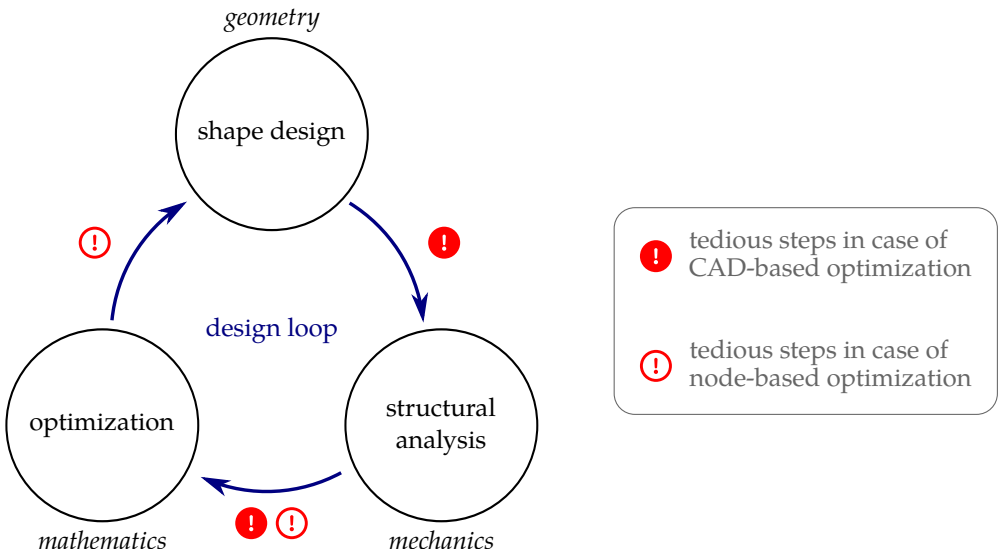


Figure 2.8 – Main steps of shape optimization (inspired from [66]): difficulties can occur during the transition from one step to the next when either CAD-based or node-based optimization is employed.

shown in figure 2.8. Depending on the modeling choices, the transition between these steps in the design loop can become quite tedious.

In case of the CAD-based optimization, the link between the CAD-model and the analysis model is not straightforward. A meshing step separates both models. This creates also drawbacks when exploiting the results from the structural analysis. These results come from the analysis model and should be transferred back to the design model which has a different description (FEM/CAD models). In case of node-based optimization, the difficulties lie in the treatments of the large number of design parameters. Indeed, the number of nodes of a FEM mesh can be significant. Thus, adequate mathematical strategies should be put in place in order to process all the data. Special care is needed to exploit the results coming from the FEA such that appropriate shape updates are imposed.

That is why IGA seems to be a great framework to perform shape optimization of structures. Wall et al. [67] were the first to show the benefit of IGA for this purpose. It has been followed by an increasing number of papers dealing with IGA-based shape optimization [18, 67–82]. It concerns not only structural shape optimization but also other fields as heat conduction [83], electromagnetics [84, 85], fluid mechanics [86], and many other optimization problems. A general procedure, which has been improved over the years, is commonly adopted [87, 88]. But before we can highlight the potential of IGA for shape optimization, we need to introduce key theoretical ingredients regarding this new analysis methodology.

2.3 Geometric Modeling

IGA as introduced by Hughes et al. [17] relies on NURBS modeling. The principal idea is to perform the analysis directly on a model that has been generated using traditional CAD modeling tools. NURBS is the acronym for Non-Uniform Rational Basis Spline which is a powerful tool used in computer graphics for generating and representing curves and surfaces. In this section, we give key element regarding this geometric modeling technique, as it composes the basis of the presented strategy.

2.3.1 Parametric representation of complex geometries

NURBS, and spline functions in general, are parametric functions. Describing a curve in the parametric form consists in expressing each coordinates of a point on the curve separately as an explicit function of an independent parameter [89]:

$$\boldsymbol{C}(u) = \big(x(u),\, y(u)\big) \quad \forall u \in \mathcal{U}, \tag{2.1}$$

where $\mathcal{U}$ defines the space of variation of the parameter u. It is often the normalized interval $[0, 1]$.

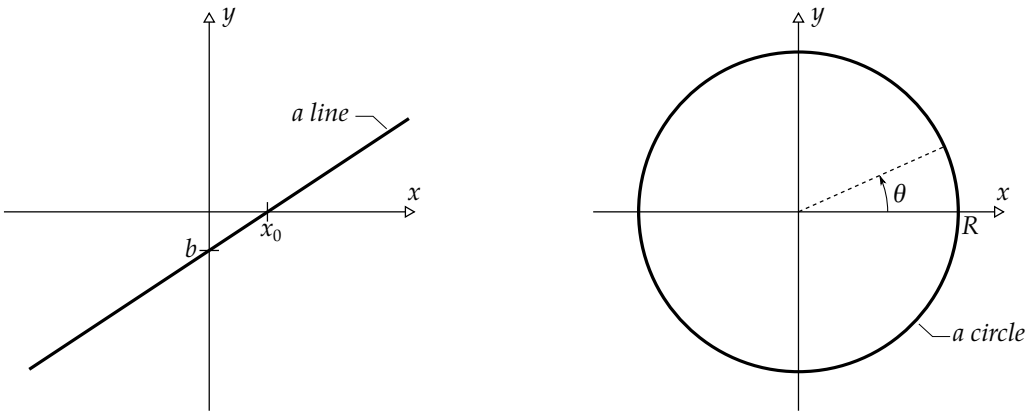


Figure 2.9 – Two simple planar curves.

Parametric description of simple curves as the line or the circle depicted in figure 2.9 can be done. For example, the line is defined as:

$$x(u) = u \quad \text{and} \quad y(u) = -\frac{b}{x_0}u + b. \tag{2.2}$$

The parameter u takes any real value to represent the complete line or is bounded to represent a segment of the line. Similarly, the curve of the circle can be parametrized as follows:

$$x(\theta) = R\cos\theta \quad \text{and} \quad y(\theta) = R\sin\theta \qquad \forall\theta \in [0, 2\pi]. \tag{2.3}$$

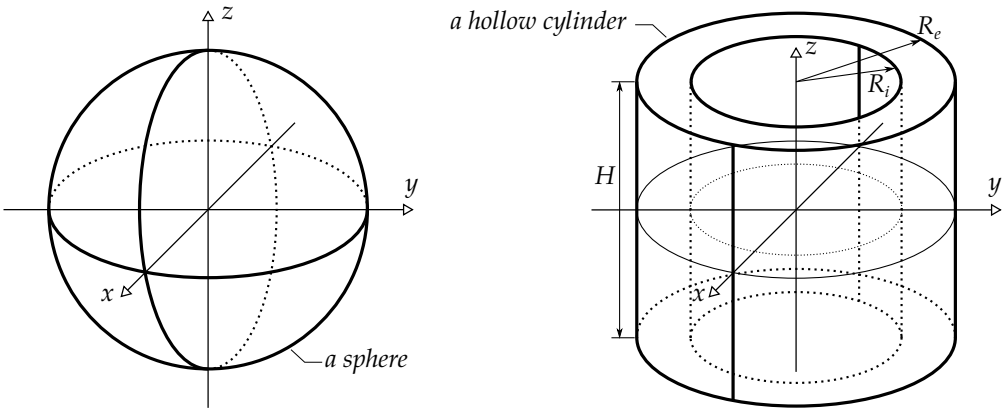


Figure 2.10 – Two simple geometric objects.

The parametrization is not unique and parametric forms are not the only way of representing curves. Implicit equations are also a common method to describe curves. For instance, the circle of radius R and centered at the origin can be described by the equation:

$$x^2 + y^2 - R^2 = 0. \tag{2.4}$$

Both techniques have different advantages. The parametric form gives an easy way to compute a point on a curve. On the other hand with the implicit form it is easy to determine if a given point is on the curve or not.

We have presented the case of a curve lying in the xy plane but it can be extended to three dimensional cases and also surfaces and volumes. There exist typical examples as those represented in figure 2.10. For instance, the unit sphere can be parametrized as:

$$\boldsymbol{S}(\theta, \varphi) = \big(x(\theta, \varphi),\, y(\theta, \varphi),\, z(\theta, \varphi)\big), \tag{2.5}$$

with

$$\begin{aligned} x(\theta, \varphi) &= \sin\theta \cos\varphi \\ y(\theta, \varphi) &= \sin\theta \sin\varphi \qquad \forall\theta \in [0, \pi],\ \forall\varphi \in [0, 2\pi]. \\ z(\theta, \varphi) &= \cos\theta \end{aligned} \tag{2.6}$$

Two parameters are needed to describe a surface (θ and φ for the sphere). For describing a volume, three parameters are required. For example, the hollow cylinder from figure 2.10 is described by:

$$\boldsymbol{V}(r, \theta, h) = \big(x(r, \theta, h),\, y(r, \theta, h),\, z(r, \theta, h)\big), \tag{2.7}$$

with

$$\begin{aligned} x(r, \theta, h) &= r\cos\theta \\ y(r, \theta, h) &= r\sin\theta \qquad \forall r \in [R_i, R_e],\ \forall\theta \in [0, 2\pi],\ \forall h \in [-H/2, H/2]. \\ z(r, \theta, h) &= h \end{aligned} \tag{2.8}$$

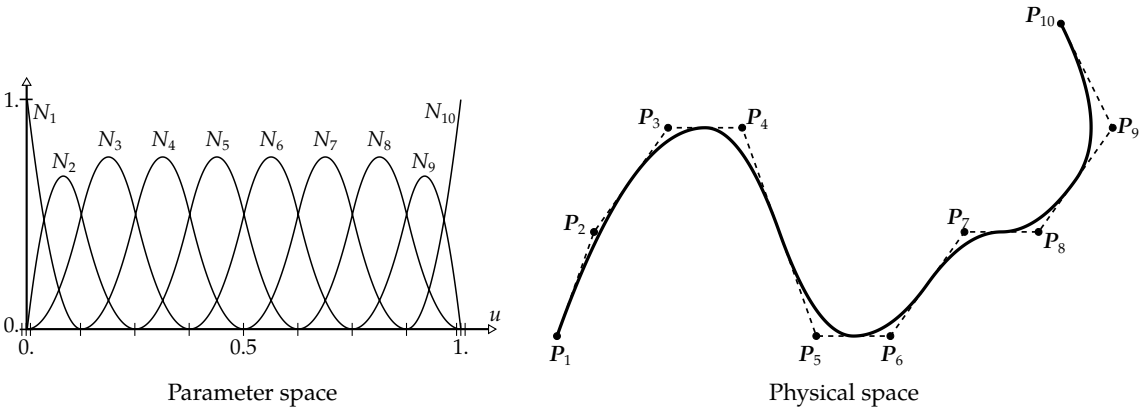


Figure 2.11 – A quadratic B-Spline curve on $U = \{0, 0, 0, \frac{1}{8}, \frac{2}{8}, \ldots, \frac{7}{8}, 1, 1, 1\}$.

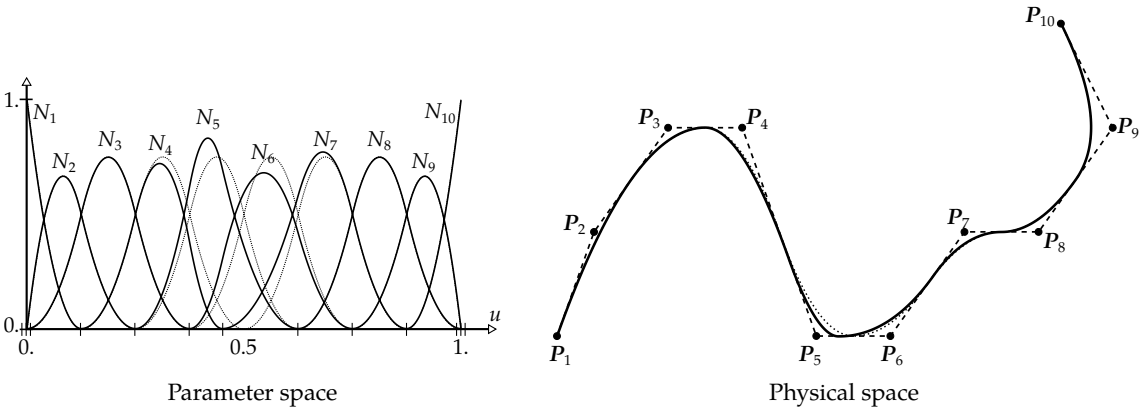


Figure 2.12 – A quadratic B-Spline curve on a non-uniform knot vector. The initial knot $u_4 = \frac{4}{8}$ used in figure 2.11 has been changed by $u_4 = \frac{9}{20}$ (the dotted lines repeat the uniform case).

2.3.2 B-Spline and NURBS modeling

B-Spline curves One possibility to describe complex curves is the use of piecewise polynomial curves. A popular technique is the use of B-Spline curves [89, 90]. It expresses a curve in the parametric form as:

$$\boldsymbol{C}(u) = \sum_{i=1}^{n} N_i^p(u)\boldsymbol{P}_i \qquad u \in [a,\, b] \tag{2.9}$$

where $\boldsymbol{P}_i$ is the control point associated to the pth-degree B-Spline basis function N_i^p. The n B-Spline basis functions are piecewise polynomials. Their expression is obtained by specifying a knot vector U and the degree p. Thus, three inputs is required to define B-Spline curve: the degree, the knot vector, and the control points.

The degree p takes an integer value and defines the degree of the basis functions. The knot vector $U = \{u_1, \ldots, u_{n+p+1}\}$ is a non-decreasing sequence of $n + p + 1$ real numbers (*i.e.* the knots). For non-periodic curve, the first $p + 1$ knots are identical and

equal to the parameter lower bound a, and the last $p+1$ knots are identical and equal to the parameter upper bound b. It leads to knot vectors of the form:

$$U = \{ \underbrace{a, \ldots, a}_{p+1}, u_{p+2}, \ldots, u_n, \underbrace{b, \ldots, b}_{p+1} \}, \tag{2.10}$$

where the interior knots verify the conditions $u_i \leq u_{i+1}$ and $a < u_i < b$.

Figure 2.11 presents a first example of a B-Spline curve. We depict both the B-Spline basis functions and the curve. The curve is quadratic and the knot vector was chosen as:

$$U = \{0, 0, 0, \tfrac{1}{8}, \tfrac{2}{8}, \tfrac{3}{8}, \tfrac{4}{8}, \tfrac{5}{8}, \tfrac{6}{8}, \tfrac{7}{8}, 1, 1, 1\}. \tag{2.11}$$

This knot vector is defined as uniform because all interior knots are equally spaced. The influence of the knot vector is shown in figure 2.12. We only change one knot of the vector such that:

$$U = \{0, 0, 0, \tfrac{1}{8}, \tfrac{2}{8}, \tfrac{3}{8}, \tfrac{9}{20}, \tfrac{5}{8}, \tfrac{6}{8}, \tfrac{7}{8}, 1, 1, 1\}. \tag{2.12}$$

This knot vector is defined as non-uniform because the interior knots are not equally spaced.

B-Spline basis functions So far, we did not mention how are formulated the B-Spline basis functions involved in equation (2.9) and already depicted in figures 2.11 and 2.12.

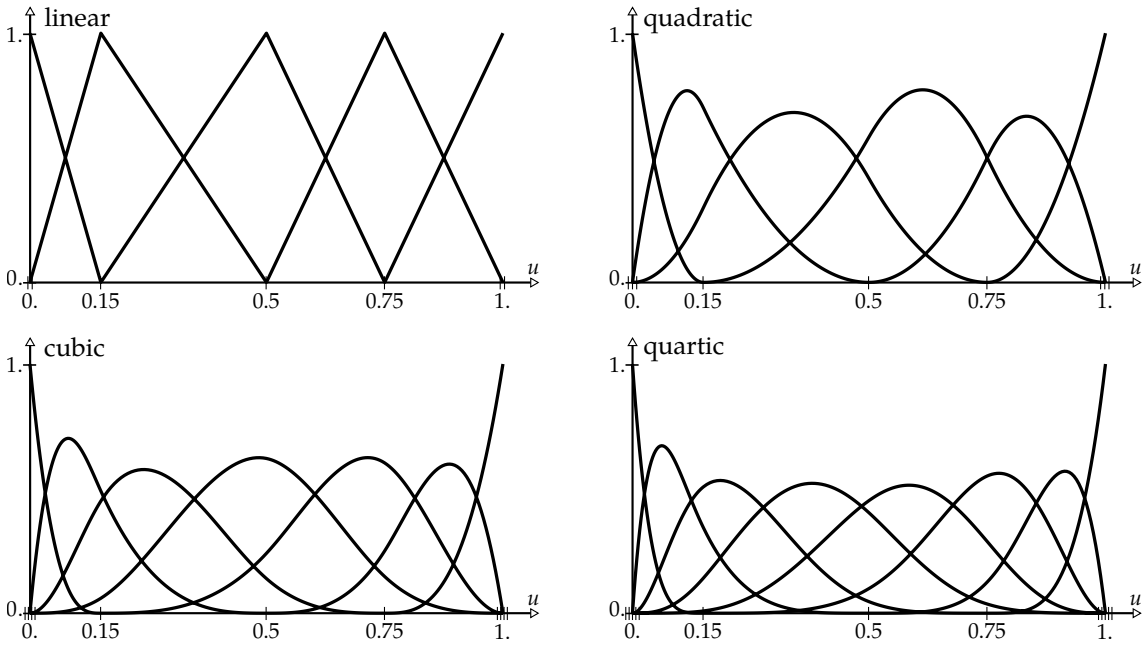


Figure 2.13 – Examples of B-Spline basis functions of different degree. Same interior knots were used (0.15, 0.50, and 0.75).

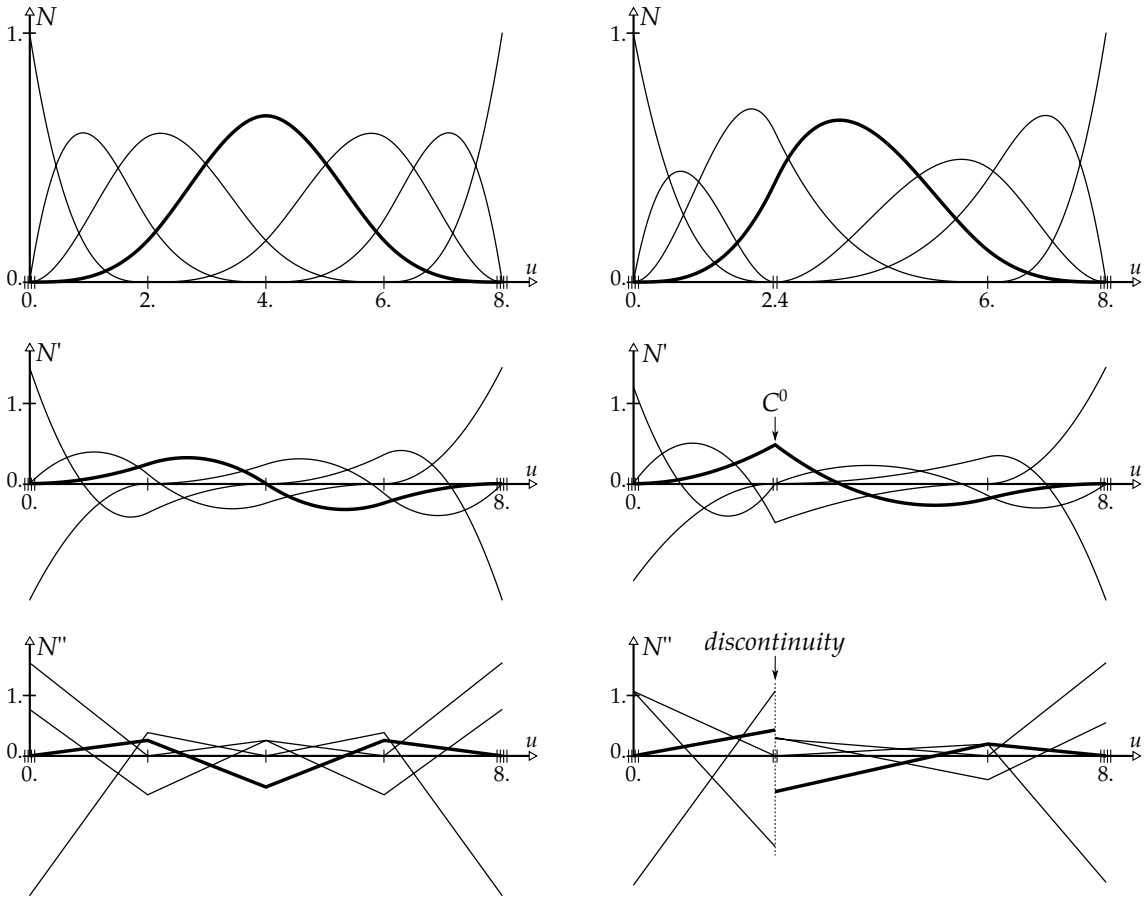


Figure 2.14 – Differentiability of cubic basis functions in case of uniform knot vector (left), and in case of double knots (right).

There exist different ways to define them [89]. We call on the recurrence formula:

$$
\begin{aligned}
N_i^0(u) &= \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \\
N_i^p(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u).
\end{aligned}
\tag{2.13}
$$

This recurrence formula is often known as the Cox-de Boor formula with reference to the inventors (C. de Boor and M.G. Cox, see [89] for more information). The initial functions N_i^0 are step functions. The function N_i^p is expressed as a linear combination of two $(p-1)$-degree basis functions.

We display several examples of B-Spline basis functions in figure 2.13. One can observed the influence of the degree. Many properties of these B-Spline basis functions are given in dedicated works [89, 91, 92]. In particular, let us mention the differentiability properties of the B-Spline basis functions. All derivatives of N_i^p exist in the interior of a

knot span. At a knot u_j, N_i^p is $p-k$ times continuously differentiable, where k denotes the multiplicity of the knot [89]. Figure 2.14 highlights this point: the second derivatives of the cubic basis functions are discontinuous at the double knot. One existing way to compute the kth derivatives is given by:

$$\frac{d^k}{du^k}\left(N_i^p\right) = \frac{p}{u_{i+p}-u_i}\frac{d^{k-1}}{du^{k-1}}\left(N_i^{p-1}\right) - \frac{p}{u_{i+p+1}-u_{i+1}}\frac{d^{k-1}}{du^{k-1}}\left(N_{i+1}^{p-1}\right). \tag{2.14}$$

Again, we refer to Piegl et al. [89] regarding theoretical and implementation aspects.

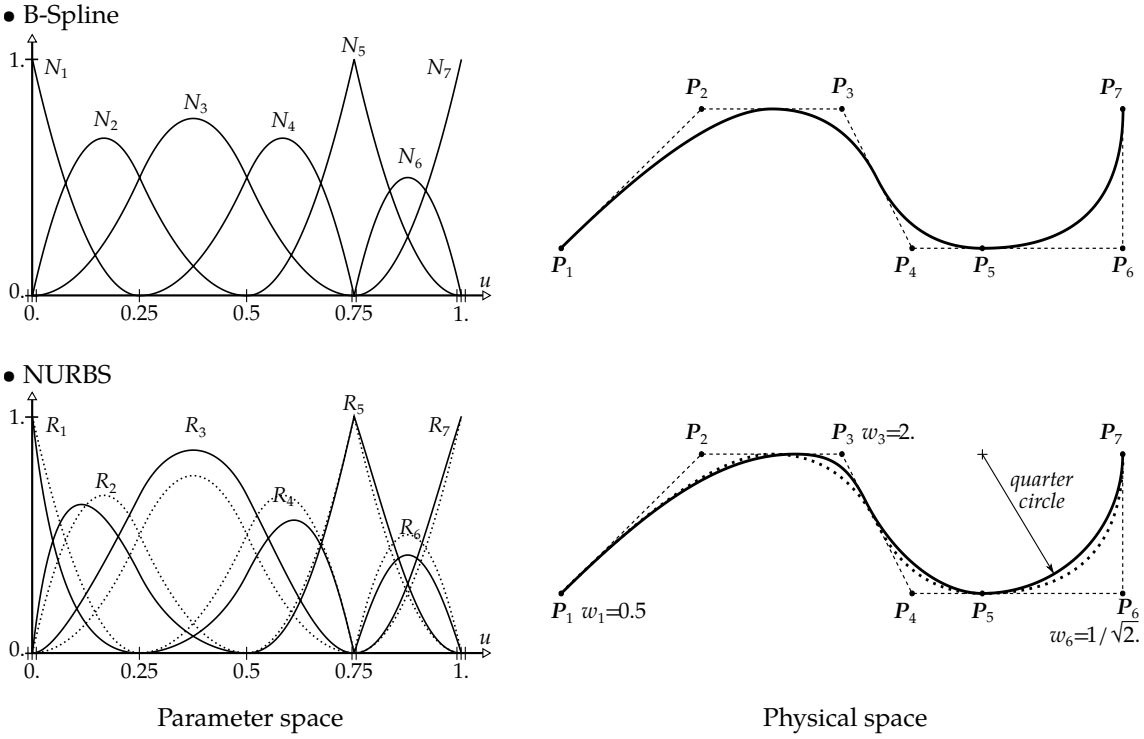


Figure 2.15 – An example of a NURBS curve: influence of the weights on the basis functions and on the final curve with comparison to the B-Spline case.

NURBS curves Non-Uniform Rational B-Spline curves are very similar to B-Spline curves and can be seen as generalized B-Splines. A NURBS curve takes the same form than a B-Spline curve. Instead of equation (2.9), we now have:

$$\boldsymbol{C}(u) = \sum_{i=1}^{n} R_i^p(u)\boldsymbol{P}_i \qquad u \in [a,\, b] \tag{2.15}$$

where the R_i^p are piecewise rational basis functions. These NURBS basis functions are expressed by adding weights to the B-Spline basis functions:

$$R_i^p(u) = \frac{N_i^p(u)w_i}{W(u)} \qquad \text{with} \qquad W(u) = \sum_{j=1}^{n} N_j^p(u)w_j. \tag{2.16}$$

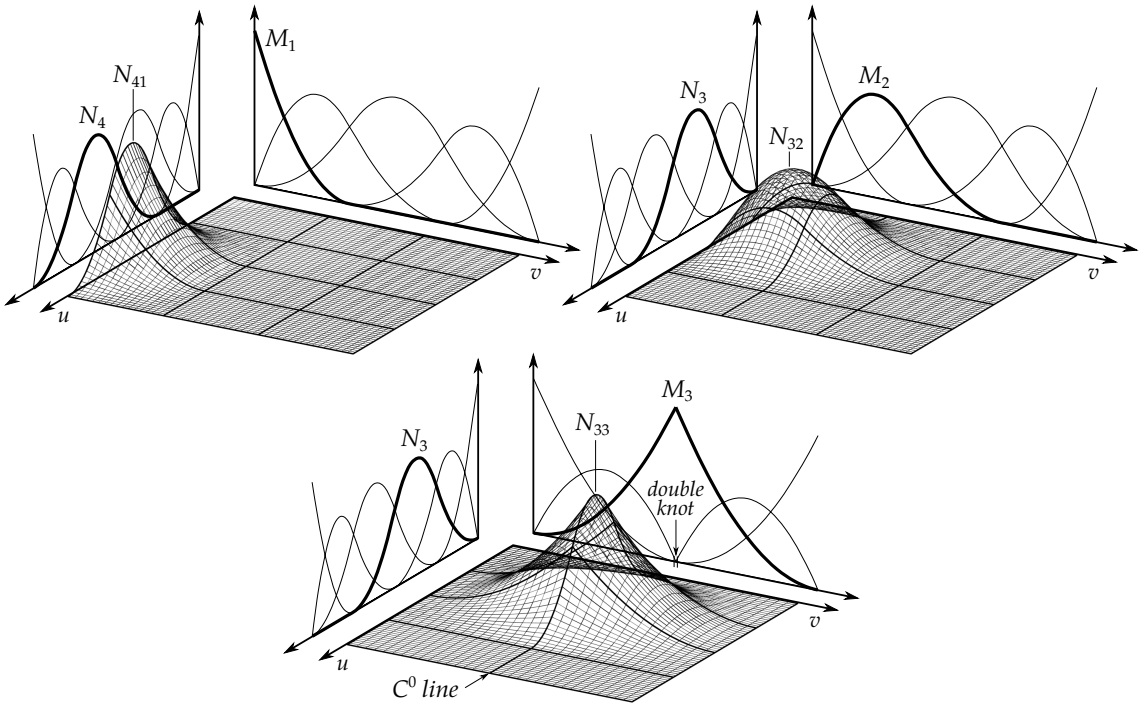


Figure 2.16 – Example of bivariate B-Spline basis functions. The lower continuity at repeated knots of the univariate basis functions propagates along the knot lines due to the tensor product.

Thus, a NURBS curve takes the same inputs than a B-Spline curve, *i.e.* a degree p, a knot vector U, and control points $\boldsymbol{P}_i$, plus an additional set of n weights w_i (one per control point). Taking all the weights equal to one reverts to a B-Spline.

The weights give more shape control to the designer. In fact, by changing their values, the curve is modified as shown in figure 2.15. It also enables to represent exactly conic sections as circles or ellipses.

Surfaces and Volumes B-Spline and NURBS are not restricted to parametric curves. Surfaces and volumes can also been described in the parametric form by using these functions. The construction principle is identical to curves. In particular, a B-Spline surface is a bivariate function of the form:

$$\boldsymbol{S}(u,v) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} N_{ij}^{pq}(u,v)\boldsymbol{P}_{ij}. \tag{2.17}$$

The bivariate B-Spline basis functions N_{ij}^{pq} take degree p in the u-direction and degree q in the v-direction. They are obtained by the tensor product of univariate B-Splines:

$$N_{ij}^{pq}(u,v) = N_i^p(u)M_j^q(v). \tag{2.18}$$

The basis functions $N_i^p(u)$ and $M_j^q(v)$ are B-Spline basis functions of degree p and q, respectively. They correspond to two different knot vectors U and V of size $n_u + p +$

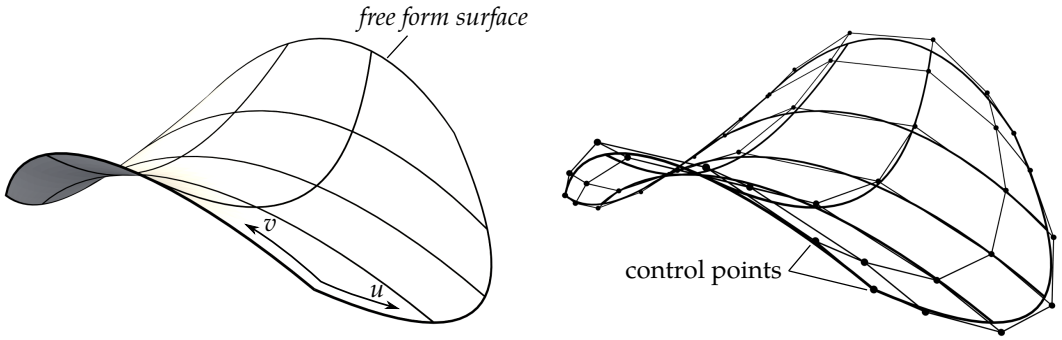


Figure 2.17 – Example of free-form surface described with a cubic B-Spline surface.

1 and $n_v + q + 1$, respectively. Examples of 2D B-Spline basis functions are given in figure 2.16. One can observe that the lower continuity due to double knot leads to a complete C^0-line of the bivariate basis function. This is due to the tensor product nature of B-Spline surfaces. Finally, in order to build of B-Spline surface, one should define two degrees, two knots vectors, and $n_u \times n_v$ control points. An example of a B-Spline surface is shown in figure 2.17.

The construction of a NURBS surface requires the additional definition of weights. As for the NURBS curve, one weight per control points is set. More specifically, a NURBS surface reads as:

$$\boldsymbol{S}(u,v) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} R_{ij}^{pq}(u,v) \boldsymbol{P}_{ij}, \tag{2.19}$$

where the bivariate piecewise rational basis functions R_{ij}^{pq} are given by:

$$R_{ij}^{pq}(u,v) = \frac{N_{ij}^{pq}(u,v) w_{ij}}{W(u,v)} \qquad \text{with} \qquad W(u,v) = \sum_{k=1}^{n_u} \sum_{l=1}^{n_v} N_{kl}^{pq}(u,v) w_{kl}. \tag{2.20}$$

The same tensor product operation is involved for NURBS surfaces. As for NURBS curves, the weights give more shape control on the surface and allow to describe exactly, for example cylinders and spheres.

Finally, let us introduce the case of parametric volumes. These are trivariate functions obtained with an additional tensor product. A NURBS volume can be expressed as follows:

$$\boldsymbol{V}(u,v,w) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} \sum_{k=1}^{n_w} R_{ijl}^{pqr}(u,v,w) \boldsymbol{P}_{ijk}, \tag{2.21}$$

where the trivariate piecewise rational basis functions R_{ijl}^{pqr} are given by:

$$R_{ijk}^{pqr}(u,v,w) = \frac{N_{ijk}^{pqr}(u,v,w) w_{ijk}}{W(u,v,w)} \quad \text{with} \quad W(u,v,w) = \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} \sum_{k=1}^{n_w} N_{ijk}^{pqr}(u,v,w) w_{ijk}. \tag{2.22}$$

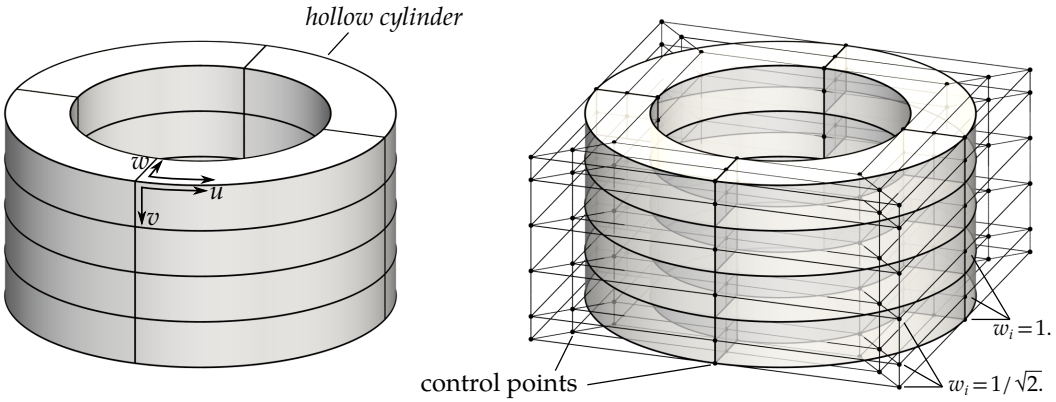


Figure 2.18 – Example of a hollow cylinder described with a quadratic NURBS volume.

The trivariate piecewise polynomial basis functions are the generalizations of those for B-Spline surfaces (equation 2.18). Thus, a volume is equipped with three knot vectors for each u-, v-, and w-direction. Figure 2.18 depicts the case of a hollow cylinder. Thanks to the weights, the circular part can be exactly represented.

2.3.3 Design features and shape parametrization

B-Spline and NURBS are interesting tools for geometric modeling because they offer great flexibility in terms of shape control. Thus, a lot of CAD software are based on these technologies. We will benefit of some of their interesting properties in this work, especially the refinement strategies and the shape manipulation.

Refinement In fact, B-Spline and NURBS can be enriched while leaving the underlying geometry intact. There are different refinement possibilities. The first possibility consists in inserting new knots in the knot vectors. This process is called *knot insertion*. More precisely, let us take the case of a p-degree B-Spline curve on $U = \{u_1, \cdots, u_{n+p+1}\}$ with n control points $\boldsymbol{P}_i$. The knot insertion consists in adding m new knots $\bar{u}_1, \ldots, \bar{u}_m$ in the knot vector (interior knots). The new refined version of the B-Spline curve takes the same degree p but is built on the new knot vector which concatenate the initial knots u_i with the inserted ones $\bar{u}_j$ (in ascending order). The new knot vector $\bar{U}$ counts $n + m + p + 1$ knots. Thus, no computation step is required regarding the construction of the new (refined) basis functions. The only calculation step concerns the control points. We need to compute the coordinates of the $n + m$ control points denoted here $\bar{\boldsymbol{P}}_i$. Interestingly, it takes the form of a linear application:

$$\bar{\mathbf{P}} = \mathbf{R}^p \mathbf{P} \tag{2.23}$$

where $\mathbf{R}^p$ is matrix of size $(m + n) \times n$, and the control points are stored in agreement: $\mathbf{P} = (\boldsymbol{P}_1, \cdots, \boldsymbol{P}_n)^T$ and $\bar{\mathbf{P}} = (\bar{\boldsymbol{P}}_1, \cdots, \bar{\boldsymbol{P}}_{n+m})^T$. The refinement matrix is sparse and is

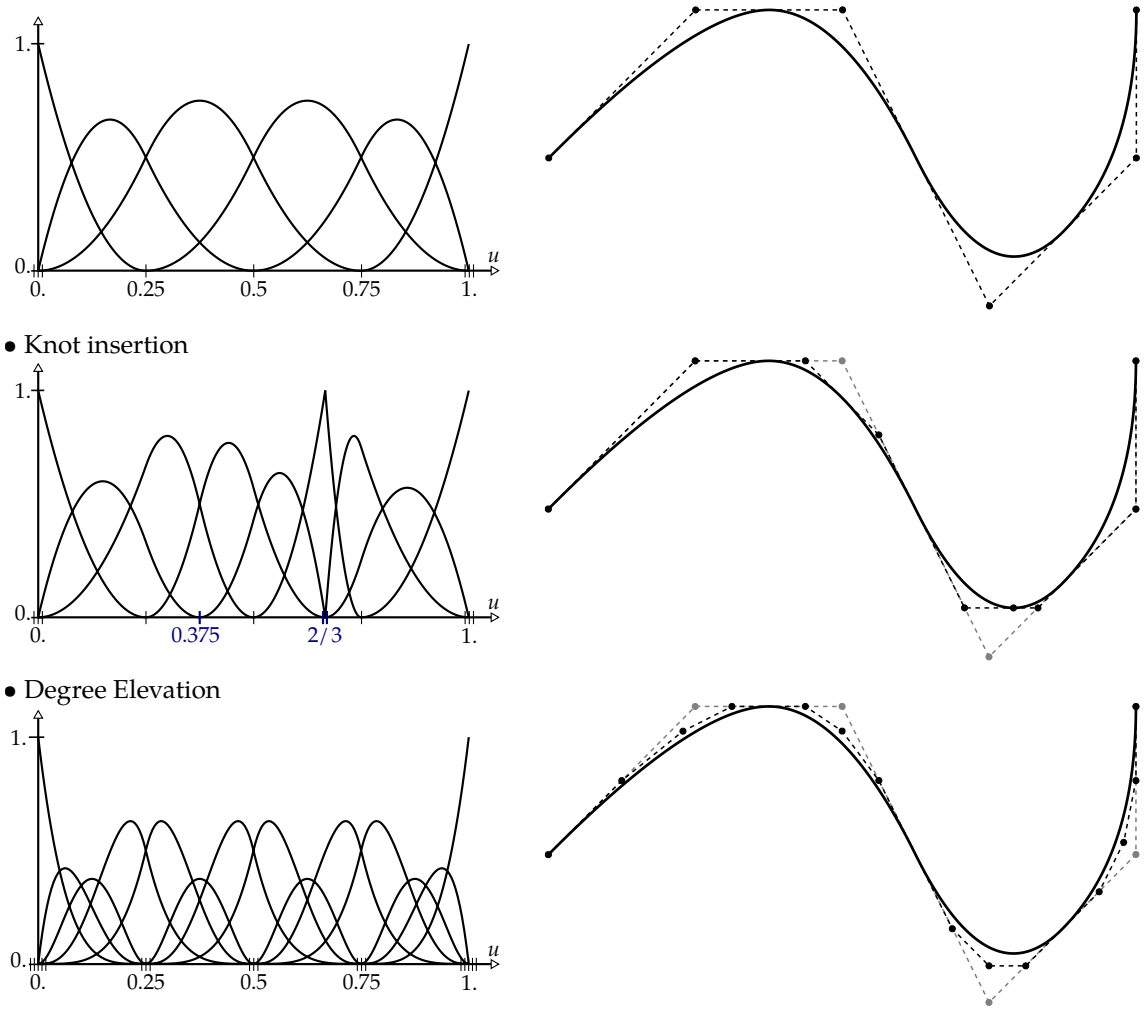


Figure 2.19 – Refinement procedures apply to a B-Spline curve.

obtained recursively [62]:

$$\mathbf{R}_{ij}^{0} = \begin{cases} 1 & \text{if } u_j \leq \bar{u}_i < u_{j+1} \\ 0 & \text{otherwise} \end{cases}, \tag{2.24}$$

$$\mathbf{R}_{ij}^{q+1} = \frac{\bar{u}_{i+q} - u_j}{u_{j+q} - u_j}\mathbf{R}_{ij}^{q} + \frac{u_{j+q+1} - \bar{u}_{i+q}}{u_{j+q+1} - u_{j+1}}\mathbf{R}_{ij+1}^{q}, \qquad q = 0, 1, \ldots, p-1.$$

An example of refinement trough knot insertion is presented in figure 2.19. One can see the changes in the basis functions and in the control points while leaving the initial geometry intact.

The other refinement possibility is to elevate the degree of the B-Spline. The polynomial degree of the basis functions is raised. In order to preserve the same level of continuity at the knots, the multiplicity of each knot value is increased as much as the

degree. Let us take the example from figure 2.19. The initial knot vector U and the refined knot vector $\bar{U}$ are:

$$\begin{aligned} U &= \{0,0,0,\ 0.25,0.50,0.75,\ 1,1,1\} \\ \bar{U} &= \{0,0,0,0,0,\ 0.25,0.25,0.25,\ 0.50,0.50,0.50,\ 0.75,0.75,0.75,\ 1,1,1,1,1\} \end{aligned} \tag{2.25}$$

At the knot location, the continuity of the refined curve is still one: $p - m = 4 - 3 = 1$ (m is the multiplicity of the interior knots). The computation of the new control points can be perform in three steps [89]:

1. subdivide the curve into many Bézier curves by using the knot insertion algorithm,
2. elevate the degree on each of these segments,
3. remove exceeding knots to combine the Bézier curves into one B-Spline curve.

Steps 1 and 3 are done using the knot insertion procedure. The formula for the degree elevation of Bézier curve can be found for example in Piegl et al. [89]. It is possible to elevate the degree for p to $p + t$ in one step:

$$\boldsymbol{Q}_i^t = \sum_{j=\max(0,i-t)}^{\min(p,i)} \frac{\binom{p}{j}\binom{t}{i-j}}{\binom{p+t}{i}} \boldsymbol{Q}_j \qquad i = 0,\ldots,p+t \tag{2.26}$$

where $\boldsymbol{Q}_i$ and $\boldsymbol{Q}_i^t$ are the control points of the initial Bézier curve of degree p and of the new Bézier curve of degree $p + t$ (numbering starts at zero here). The interesting point to notice is that the degree elevation procedure takes also the form of a linear application. A refinement matrix can be built to express the new control points of the degree-elevated B-Spline in relation to the initial ones [93]:

$$\bar{\mathbf{P}} = \left[\mathbf{S}^r\mathbf{T}^t\mathbf{R}^i\right]\mathbf{P}, \tag{2.27}$$

where $\mathbf{R}^i$ is the matrix that decomposes the curve in multiple Bézier curves, $\mathbf{T}^t$ is the matrix that elevate the degree of the Bézier curves, and $\mathbf{S}^r$ is the matrix that recompose the B-Spline from the multiple Bézier curves. Matrix $\mathbf{S}^r$ can be seen as a lest-squares inverse of an operator used for knot insertion.

The refinement procedures (knot insertion and degree elevation) can be combined which offers great flexibility regarding the parametrization of the B-Spline curve. In this case, degree elevation is performed first, and then the new knot values are inserted. It leads to a curve with higher continuity level at these new knots, this the so-called k-refinement [62]. Finally, we will denote by $\mathbf{R}$ the refinement matrix throughout the rest of this document independently of the refinement procedure. Let us also mention that same refinement operators can be used for NURBS curve by using the homogeneous coordinates:

$$\bar{\mathbf{P}}^w = \mathbf{R}\mathbf{P}^w, \tag{2.28}$$

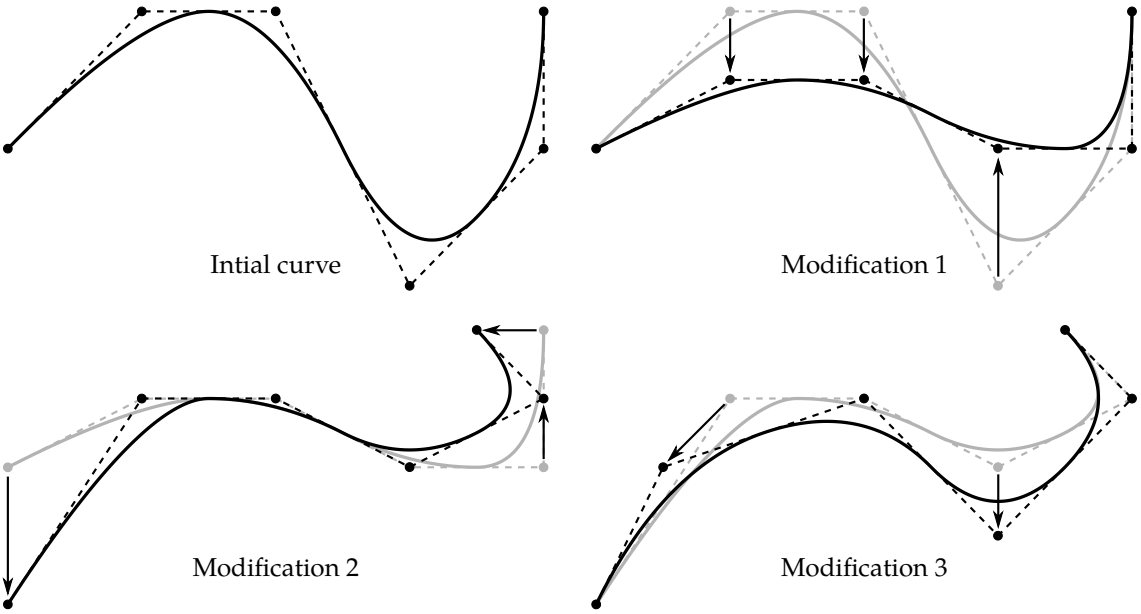


Figure 2.20 – Example of shape modification by altering the spatial location of the control points.

where $\mathbf{P}^w$ has one more column than $\mathbf{P}$: considering $\boldsymbol{P}_i = (x_i, y_i, z_i)$ with associated weight w_i, then $\boldsymbol{P}_i^w = (x_i w_i, y_i w_i, z_i w_i, w_i)$. The refinement strategy can also be extended to surfaces and volumes: the refinement is performed direction per direction.

Geometric modification B-Spline and NURBS give to the designer great flexibility over the shape. We have already observed the influence of the knot values and the influence of the weights. The designer can also operate shape changes by modifying the coordinates of the control points as shown in figure 2.20. This possibility will be of great interest for performing shape optimization. In the framework of isogeometric shape optimization, the shape modifications can be imposed by altering the spatial location of the control points, and in some case the weights.

2.4 NURBS-based Finite Element Analysis

2.4.1 Isogeometric Principle

General idea The idea behind isogeometric analysis is to use the spline-based parametrization to build approximation spaces when applying the Galerkin's method [17, 62]. Here we restrict ourselves to NURBS-based models. Thus, the solution space is defined as all linear combinations of a given set of NURBS basis functions. This is finally the major difference with the classical Finite Element Method from the analysis point of view. IGA relies on same weak formulations than the standard FEM. The isoparametric NURBS basis turns these weak statements into a system of algebraic equations. In linear elasticity,

it means that the displacement field is approximated as:

$$u^h = \sum_{A=1}^{n} R_A \boldsymbol{u}_A, \tag{2.29}$$

where the R_A are the NURBS basis functions originated from the geometric model of the structure. With this approximation solution space in hand, the Principle of Virtual Work takes the form of the well-known linear system:

$$\mathbf{K}\boldsymbol{u} = \boldsymbol{F}, \tag{2.30}$$

where $\mathbf{K}$ is the so-called stiffness matrix, $\boldsymbol{u}$ is the displacement vectors (vector of Degrees Of Freedom), and $\boldsymbol{F}$ is the force vector. We do not go into the details here regarding their expressions. Theoretical developments are presented in the next chapters with a focus on shell analysis. This little introduction merely aims at linking IGA with the standard method for structural analysis (*i.e.* FEM).

Performance IGA has shown great performance in term of computational results making IGA often seen as a high-performance computational method. Many researchers working in computational mechanics have been interested in this new technology. IGA has been successfully applied to numerous mechanical problems. One can give a non-exhaustive list of applications: shell analysis [94–96], contact problems [64, 97], fluid-structure interaction [98, 99], shape optimization problems [18, 67, 71], material with non-linear behaviors [100–102], fluid dynamics [103, 104], structural vibration [105, 106], etc. Even more, IGA has even outperformed classical FEA in specific fields as for example in the simulation of aortic valve closure where significant reduction of the analysis time was obtained [107]. It has been observed that the computational geometry technologies (spline functions) used in IGA framework have great mathematical properties for numerical simulation. These smooth high-order functions enable to handle severe mesh deformations that occur during non-linear simulations [108]. It also leads to greater accuracy (per Degree Of Freedom) in comparison to standard approaches [109].

2.4.2 Multi-level approach for Shape Optimization

The framework of this study draws on research dealing with isogeometric shape optimization. A general procedure, which as been improved over the years is commonly adopted [87, 88].

Figure 2.21 describes the main steps of the optimization process. It gets an initial CAD model as a main input. CAD model has here to be understood as an analysis suitable NURBS geometry. As an output the optimal shape is given and, an interesting point to notice is that this shape is a NURBS geometry too. Therefore, the result is directly usable by designers and, thus, for production. There is no need to re-design the final geometry in a CAD environment as it would be the case with FE-based optimization.

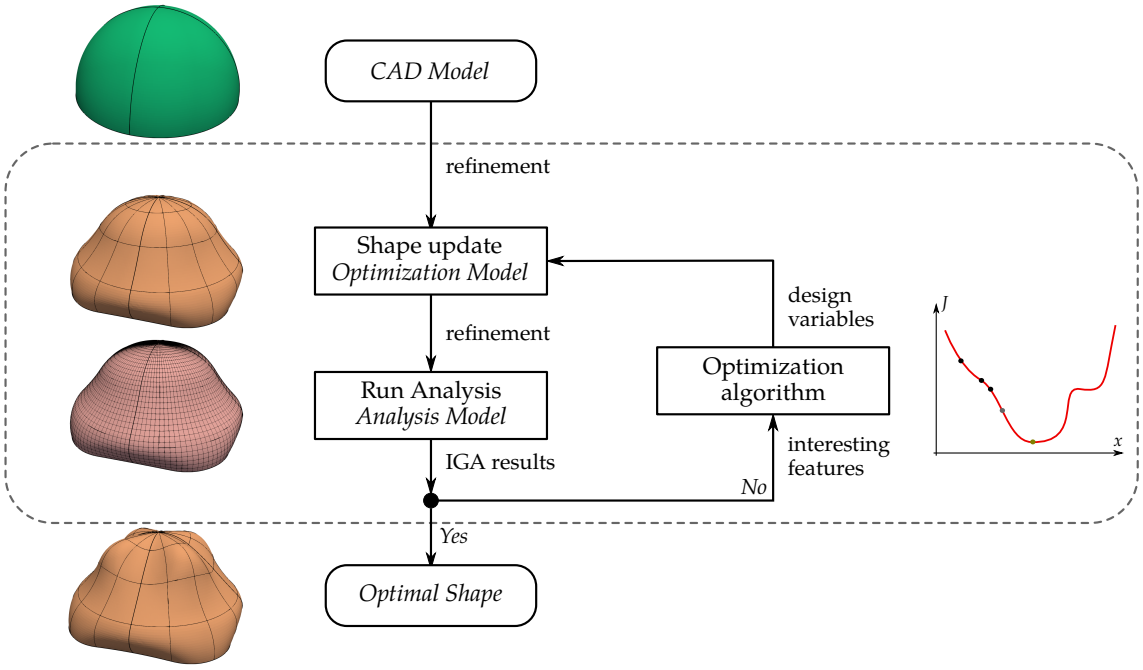


Figure 2.21 – Isogeometric shape optimization flowchart: overview of the main steps.

The optimal NURBS geometry is obtained from an iterative process with three main steps. Each iteration consists in:

1. updating the current shape in a better one by altering the spatial location of the control points and in some case the weights,
2. running the analysis for this new geometry and then inferring mechanical and/or geometrical properties of interest,
3. computing new design variations to further improve the structure.

A major asset of isogeometric shape optimization is the possibility to properly choose both optimization and analysis spaces [68–72, 75, 84]. A fine NURBS discretization is introduced as the analysis model in order to ensure good quality computations. Conversely, the optimization model is defined to impose suitable shape variations. Both spaces describe the exact same geometry and are initially obtained through different refinement levels of the CAD model as shown in figure 2.22.

It is clear that, the finer the analysis model, the better the results. The choice of the refinement level used for the analysis model is principally dictated by the need for realistic computational cost. When it comes to the optimization model, the situation is quite different. In this case, the choice of the refinement level has an impact on the complexity of the optimal shape reached. A coarse optimization model may provide a simpler optimal shape than a fine optimization model. Therefore, few questions can be asked in order to accurately define the refinement levels for the analysis and optimization models:

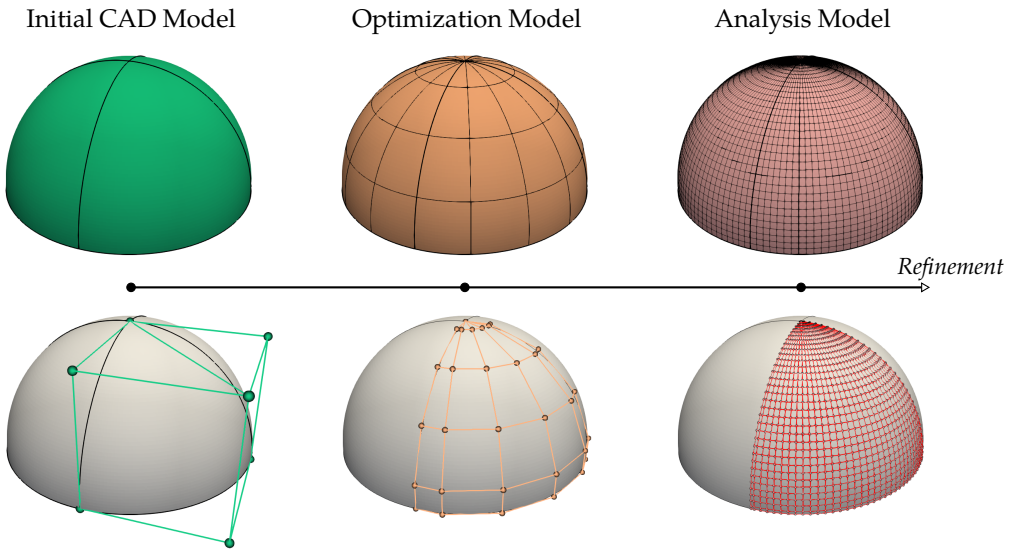


Figure 2.22 – Multilevel design approach: optimization and analysis spaces describe the exact same geometry and initially obtained through different refinement levels.

— How different from the initial geometry the optimal shape is allowed to be ?

— Depending on the intended level of shape variation complexity, how fine should the analysis model be to ensure reliable results ?

These points are illustrated and discussed in more details in all the following examples of this work.

Chapter 3

Isogeometric Shape Optimization of Shell Structures

Designing structural parts against the material limits, the impact of loads, and many other constraints, is of standard interest in engineering. However, improving the design of a structure can be long and drawn out, especially when a clear understanding of cause-effect relationships is missing. Finding the best possible design, namely the optimal design, is a complex discipline that combines mathematics and mechanics. This chapter deals with this issue: it gives theoretical insights on structural shape optimization with application to shell design. The principal goal is to study and highlight the benefit of IGA for this particular issue.

Contents

3.1 Optimization Problems

3.1.1 Mathematical Formulation

The mathematical formulation of an optimization problem involves different quantities. Firstly, we need to quantify the performance of the studied system. For instance, in case of design optimization, we require a quantity that tell us what is the best design between several possibilities. This quantity is called the **objective** and takes in practice the form of a scalar function denoted here f. The objective is usually formulated such that the best solution is the one which returns the smallest value assessed by function f. At this point, we understand that the selection of the objective is a crucial point. In fact, it will drive all the decisions made during the design process. The choice of a good objective is not a simple task regarding complex real-world problems. The objective depends on several characteristics of the system, called variables or unknowns. Once again, in the specific case of design optimization, we need to set design parameters that enable to vary the geometry of the structure. We designate these parameters as **design variables** and we represent them as a vector of unknowns x. In this work, we only consider the case of continuum variables. Finally, we need to describe the space in which we are searching the optimal solution. This space is delimited through a combination of restrictions called the **constraints** c_i. These constraints act on the variables x and define the domain where they live in. Mathematically, the constraints are defined as scalar functions of x, and commonly takes the form of implicit equations or inequalities. With these notations in hand, an optimization problem can be formulated as follows:

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{w. r. t.} \quad & x \in \mathbb{R}^n \\ \text{subjected to} \quad & c_i(x) = 0, \ \forall i \in \mathcal{E} \\ & c_i(x) \leq 0, \ \forall i \in \mathcal{I} \end{aligned} \tag{3.1}$$

where $\mathcal{E}$ and $\mathcal{I}$ are set of indices for equality and inequality constraints, respectively.

Such a definition of an optimization problem can appear restrictive, but in fact it offers great possibilities. Indeed, we can imagine an infinity of optimization problems that can be written in the presented form. In the particular context of design optimization of structures, a multitude of choices can be made for defining both the objective, the design variables, and the constraints. The definition of these quantities ought to be done by the designer: this is the modeling step.

Structural Optimization Let us describe the most common modeling choices in case of structural optimization. A classification exists as discussed in the previous chapter (see section 2.1). The differences between sizing, shape, and topology optimization reflect in the definition of the design variables [110]. Figure 3.1 highlights this classification and depicts what could be the design variables for each type of structural optimization

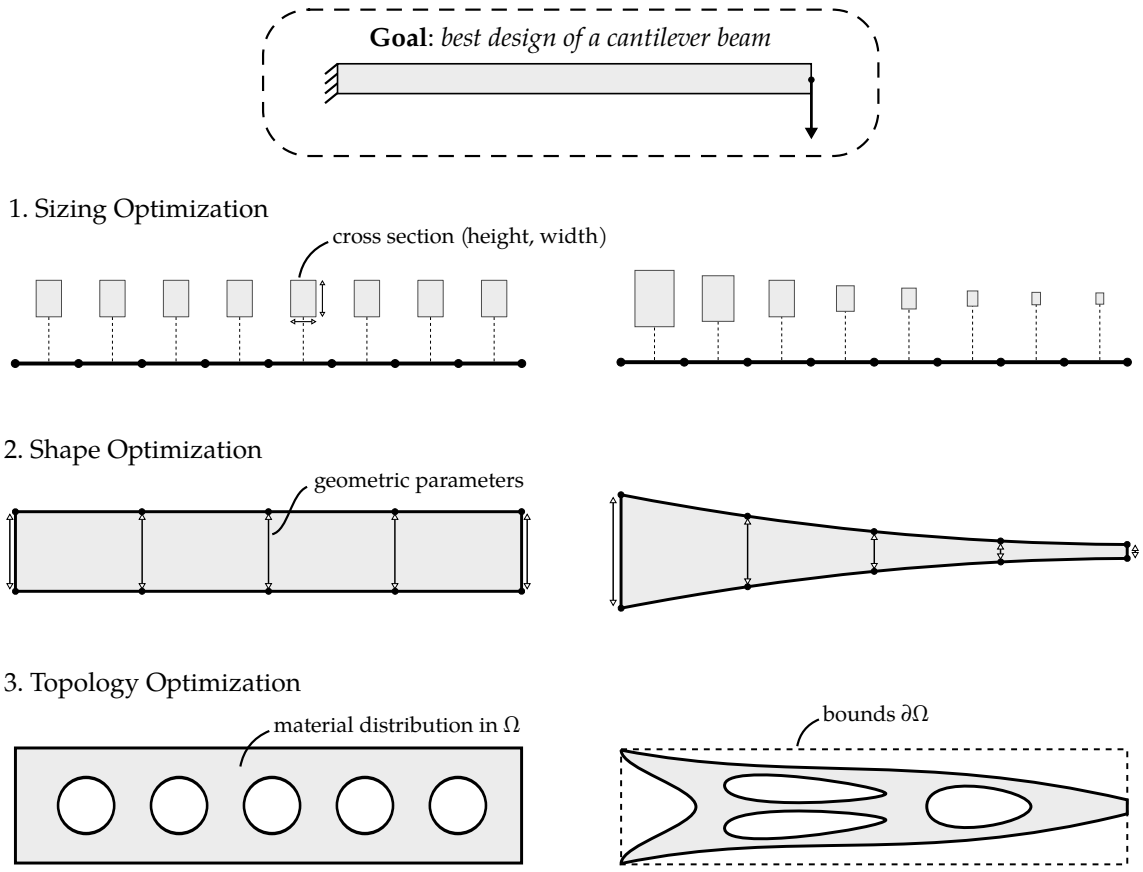


Figure 3.1 – Type of optimization

problems for a simple case of cantilever beam. We can give a non-exhaustive list of typical design variables associated to each category of optimization problems:

— *material and sizing optimization*
fiber angles or number of plies in a composite material, cross section of a beam (area, moment of inertia), thickness of a plate,
— *shape optimization*
geometry variables such as nodal coordinates of a finite element mesh, control point coordinates of a spline, CAD dimensions (length, width, radius),
— *topology optimization*
material distribution in a design space.

The boundary between all these categories is however not distinct as mentioned by Bletzinger [36]. For example, the method called SIMP (Solid Isotropic Material with Penalization) which deals with topology optimization consists in modifying the Young's modulus of each finite element (low value is associated to void and high value expresses the presence of material). In this case, the design variables are similar to those involved in material optimization or sizing optimization. Choosing the type of the design vari-

ables usually depends on the stage of product development. According to Bletzinger [36], topology optimization is mainly used during the conceptual stage. Shape optimization comes into play in preliminary design and in the detailed design. Sizing is mainly used in the detailed design stage.

As for the design variables, there are also typical response functions that are commonly involved in structural optimization problems. We can list some of them:

— linear compliance,
— eigenfrequency,
— stress,
— mass,
— linear buckling,
— displacement at a specific location.

These response functions are chosen as the objective or the constraints in the problem (3.1). The choice of a response function is independent of the choice of the design variables. Thus, same response function can be used, for instance, during a sizing optimization and a shape optimization. Interested readers can refer to [111] for general contents regarding structural optimization. More specialized works exist as for instance [112] for composite material, and [113, 114] for topology optimization, etc.

3.1.2 Gradient-based Algorithm

There exists a wide number of algorithms that enable to solve the constraint optimization problem (3.1). They share one similarity: each algorithm tends to find the solution in an iterative way. Starting with an initial guess x_0, the variables are modified iteratively in order to reduce the value of the objective function, and also in order to verify the constraints (which can induce the increase of the objective function). After a certain number of iterations, the algorithm hopefully stops because the solution x^* has be founded.

Global and Local optimization A question is how to determine whether a given set of variables x^* is the solution of the optimization problem (3.1) or not. Firstly, let us consider the case of an unconstrained problem (*i.e.* $\mathcal{E} \cup \mathcal{I} = \emptyset$). In this case, one can qualify x^* as the global minimizer if the following relation holds (definition taken from Nocedal et al. [115]):

$$f(x^*) \leq f(x) \quad \text{for all } x. \tag{3.2}$$

Of course, this definition cannot be used in practice to identify an optimum since it is not possible to evaluate the objective function for an infinite set of variables.

Thus, we require other criteria to identify the minimum. Fortunately, one can formulate necessary and sufficient conditions satisfied by the minimizer. Particularly, in case of a smooth objective function (*i.e.* continuously differentiable function), a necessary

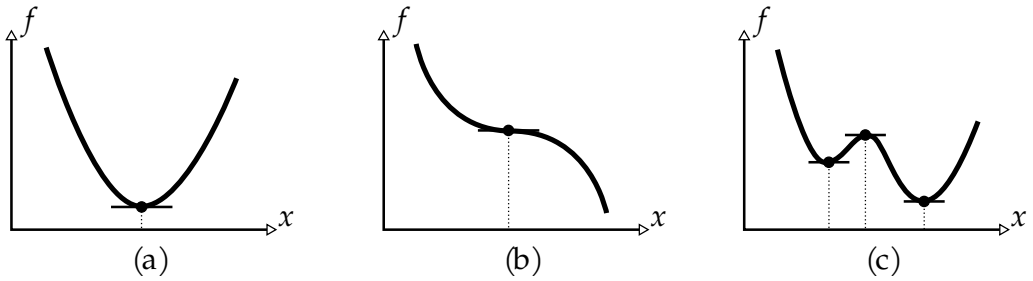


Figure 3.2 – Recognizing a minimizer of a smooth function: (a) the first derivative equals to zero at the global minimum, but this condition is not sufficient to identify the global minimizer as highlight in cases (b) and (c).

condition to fulfill is:

$$\nabla f(x^*) = 0. \tag{3.3}$$

This equation is often called the *First-Order Necessary Condition* because it involves the first derivatives. It is however not sufficient to identify x^* as the global minimizer of function f as described in figure 3.2. The points for which the equation (3.3) holds are named as stationary points. In order to fully recognize a minimizer, one need to look at the Hessian $\nabla^2 f$. Under the consideration that the Hessian is continuous, a minimizer x^* of function f can be recognize if the following condition is satisfied:

$$\nabla f(x^*) = 0, \quad \text{and} \quad \nabla^2 f(x^*) \text{ is positive definite.} \tag{3.4}$$

It is the *Second-Order Sufficient Condition* and one can find the associated proof in, for example, the book of Nocedal et al. [115]. However, we cannot conclude that a given point x^* which verifies the sufficient condition 3.4 is the global minimizer of our function f. In fact, in case of a non-convex function as the one depicted in figure 3.2(c), there might exist multiple local minimums where this condition holds. Recognizing and locating a global minimizer turns out to be very difficult in practice, especially for large and complex engineering problems where the objective function is inevitably non-convex. Many algorithms for nonlinear optimization problems enable to find local solutions only. A local solution is the best solution in a neighborhood of feasible points. Although most of the time local solutions are founded, these solutions highly improve in many times the initial configuration.

Before describing the algorithms, let us mention that equivalent conditions to (3.3) and (3.4) are available in case of constrained optimization problems. If x^* denotes a local solution of the constrained problem (3.1), then there is a Lagrange multiplier vector λ^*

such that:

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0, \tag{3.5a}$$
$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E}, \tag{3.5b}$$
$$c_i(x^*) \leq 0, \quad \forall i \in \mathcal{I}, \tag{3.5c}$$
$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I}, \tag{3.5d}$$
$$\lambda_i^* c_i(x^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}, \tag{3.5e}$$

where the Lagrangian function denoted $\mathcal{L}$ is defined as:

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

The necessary conditions (3.5) for an optimum is usually called the *Karush–Kuhn–Tucker* conditions. Equivalently to the sufficient condition given in the specific case of unconstrained problem, *e.g.* equation (3.4), one can formulate sufficient condition verified by a local solution to problem (3.1). Additionally to the KKT conditions, the Hessian $\nabla^2_{xx} \mathcal{L}(x^*, \lambda^*)$ should be positive definite in the tangent space of the active constraints (sometimes called the critical cone). Interested readers can refer to Nocedal et al. [115], for example.

We stop here giving details regarding the theory of optimization problems. We give here the main results only. We mention these points because they underlie the construction of optimization algorithms.

Optimization algorithms There exists a large number of algorithms that enable to solve optimization problems of the form (3.1). Each algorithm is usually designed into a specific framework. It means that there is not a single generic algorithm, but instead there exists a large number of algorithms that are more or less efficient for specific classes of optimization problems. We can distinguish different classes of problems depending on the presence of constraints or not, the number of variables (small or large), the convexity and the smoothness (linear, highly non-linear, continuity, differentiability) of the objective function and constraints, etc. A good algorithm should however work well on a wide variety of problems. Its efficiency is measured in view of the required computational resources (time and storage). A good algorithm should converge fast with a low storage usage. It should also give the solution with precision and this without being too sensitive to arithmetic rounding errors [115].

As a consequence, another additional task to the modeling step (formulation of the optimization problem) consists in choosing the appropriate optimization algorithm. This task is far from trivial. In case of structural shape optimization, *gradient-based* algorithms are mainly used. One principal reason for this, is that the commonly adopted functions (objective and constraints as mentioned previously in section 3.1.1) are usually differentiable *w.r.t.* the design variables. Thus, gradients can be computed (we will see

how later on). For large problem with hundred to thousand design variables, the use of gradients is quasi-inevitable to build an algorithm that converges in an acceptable amount of time. The information brought back by the gradients enables the algorithm to make suitable decisions during the resolution. For large scale problems, gradient-free algorithm may require much more evaluation of the objective and constraint functions which drastically increase the overall computational time.

Without going into detail, let us mention the two fundamental strategies used by optimization algorithms: the *line search* strategy, and the *trust-region*. These two strategies enable to solve unconstrained optimization problems. They aim at updating the current iterate x^k into a better one x^{k+1}, as follows:

— *line search*
  — the algorithm computes firstly a direction p^k,
  — it searches along this direction to find a new iterate which improves the solution: this give a scalar α^k named the step length,
  — then the variables are updated, *i.e.* $x^{k+1} = x^k + \alpha^k p^k$.

— *trust-region*
  — the algorithm builds a model m^k that approximates the problem near the current point x^k,
  — it computes the minimizer of m^k in a region close to x^k, *i.e.* the trust region,
  — if the solution of the approximate sub-problem improves sufficiently the solution, it is used as the new iterate.

Furthermore, a close link between both strategies can be found (see again Nocedal et al. [115]). In fact, the model m^k used in trust-region is usually a quadratic function involving the function gradient and an approximation of the Hessian B^k:

$$m^k(x^k + p) = f(x^k) + p^T \nabla f(x^k) + \frac{1}{2} p^T B^k p. \tag{3.6}$$

For example, if the matrix B^k is set to zero, then minimizing the approximation (3.6) in the trust-region gives a steepest descent step in which the step length correspond to the trust-region radius [115]. Finally, let us mention the case of quasi-Newton methods where the Hessian matrix is approximated trough the gradient variations during the resolution. These methods are appealing because the Hessian is not directly computed, which is particularly demanding in case of structural optimization. A popular method is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) Hessian update strategy.

Algorithms for non-linear constrained optimization problem add other ingredients in order to handle the inequality and equality constraints. We can list three main categories of algorithms:

— Penalty and Augmented Lagrangian methods

— Sequential Quadratic Programming
— Interior-Point methods

In the first category, the original constrained problem is replaced by a sequence of unconstrained problems where terms are added to the objective function. SQP methods generates iteratively quadratic programming sub-problems to be solved. Interior-Point methods provide efficient algorithm to solve large-scale constrained optimization problems. These approaches use barrier functions to reformulate the optimization problems as barrier problems. More information on optimization gradient-based algorithm from both theoretical and practical sides can be found in Nocedal et al. [115].

Nowadays, robust implementations of optimization algorithms are provided in scientific computing libraries. For instance, the optimization module of the python open source package `scipy` offers a large choice of algorithms [116]. In this work, we do not develop a specific optimization algorithm, even it could be of great interest. For example, Nagy *et al.* [71, 117] introduce an efficient algorithm for sizing and shape optimization in the IGA framework. Instead, in this work, we use the SQP algorithm from Kraft [118] which is available in several python packages as for instance `scipy` [116] or `NLopt` [119]. More specifically, the algorithm is denoted `SLSQP` in the aforementioned packages, which stands for Sequential Least-Squares Quadratic Programming.

3.1.3 Methods for Sensitivity Analysis

When gradient-based optimization algorithm is used, a crucial issue is therefore the computation of the gradients. This step is commonly named as *sensitivity analysis* where the goal is to study the effect of an input variation on the quantities of interest (the output). Once again, there are different possibilities with different levels of accuracy and numerical cost (computational and implementation). Thus, the different options for sensitivities can either be [120]:

— Discrete or Variational,
— Direct or Adjoint,
— Analytical or Approximate.

A review of these options for structural design sensitivities can found in Keulen et al. [120]. They highlight the differences in the framework of linear static analysis. The difference between discrete derivatives and variational derivatives lies in the discretization step. Continuum derivatives are obtained by differentiating firstly the continuum equations that governed the structural behavior. Then a discretization step enables to solve the resulting sensitivity continuum equations which finally gives the sensitivities. The discrete sensitivities are obtained by differentiating the discrete governing equation versus the design variables. Thus, the discretization step occurs in this case before the differentiation. In the context of IGA-based shape optimization, Fußeder et al. [75] tackles this issue by comparing these two schemes: *optimize first–discretize then* and, reversely, *discretize first–optimize then*. Equivalence between these possibilities can be shown in

several cases [75]. The notion of adjoint or direct sensitivities concerns both variational and discrete derivatives. Adjoint sensitivity analysis is numerically more efficient if the number of response functions involved in the optimization problem (objective and constraints) is lower than the number of design variables. In structural optimization, many problems are self-adjoint [36], which means the adjoint sensitivities do not involve additional equations to be solved. Therefore adjoint sensitivities constitute a natural choice in this case and the notion of adjoint or direct sensitivities is set aside. Finally, approximation steps can be introduced during the sensitivity analysis in order to avoid, for example, complex and long formulas which are tedious to implement without any mistakes. More specifically, the so-called semi-analytical discrete sensitivities are largely used in structural analysis codes because they require very little programming effort and almost no element-dependent sensitivity routines [36, 120].

Let us now give mathematical expression of some of these sensitivities in the context of linear static analysis. We focus here on discrete sensitivities. In this context, we have the discrete governing equations that takes the well-known form of a linear system of equation:

$$\mathbf{K}(x)\boldsymbol{u} = \boldsymbol{F}(x), \tag{3.7}$$

where the stiffness matrix $\mathbf{K}$ and the load vector $\boldsymbol{F}$ are formulated through domain integrals. It means that their expressions depend on the design variables x, and therefore the displacement DOFs $\boldsymbol{u}$ does also. Classical structural responses can generally be written explicitly in terms of the design variables and the state variables $\boldsymbol{u}$:

$$f = f(x, \boldsymbol{u}(x)). \tag{3.8}$$

As an example, one can take the case of the compliance which can be written in the discrete form as:

$$f_{\text{cpl}} = \frac{1}{2}\boldsymbol{F}(x) \cdot \boldsymbol{u}(x). \tag{3.9}$$

Global Finite Difference A simple way to perform the sensitivity analysis consists in perturbing each design variables with a small variation and evaluate the response of the structure for each of these perturbations. The use of Finite Difference formula enables then to obtain the derivatives. The most popular remains the forward difference:

$$\frac{df}{dx_i}(x) \approx \frac{f(x + \varepsilon\, e_i) - f(x)}{\varepsilon}, \tag{3.10}$$

where the perturbation ε takes a small scalar value, *e.g.* $\varepsilon = 10^{-6}$. Vector e_i has the same length than x, with all components equal to zero expected the component i which is set to one. The accuracy depends on the parameter ε. In theory, the exact derivative is obtain when it tends to zero. However, in practice the step size is limited due to arithmetic accuracy and a too small value gives wrong results. Finally, one can understand the importance of scaling the problem when approximated sensitivities are used. In order

to choose a step length which is less sensitive to the dimension of the problem, one can scale the design variable so they vary between 0 and 1 for example.

If the function f depends on the state variable as for instance the compliance, then the state equations (3.7) have to be solved for each perturbation which turns to be very computational demanding in case of a high number of design variables. To limit the computational cost, one can employ fast re-analysis techniques. For example, if an iterative solver is used, a unique preconditioner for each perturbation is sufficient, and the solution obtained without the shape perturbation can be used has a good starting point. More discussion on this issue is tackled by Keulen et al. [120].

Analytical Discrete Sensitivity One can obtain analytical sensitivities by applying a chain rule of differentiation. The starting point is the differentiation of the response function as follows:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \frac{d\boldsymbol{u}}{dx_i}. \tag{3.11}$$

The term $d\boldsymbol{u}/dx_i$ is not explicitly known. We can identify this term by differentiating the discrete state equations (3.7):

$$\frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u} + \mathbf{K}\frac{d\boldsymbol{u}}{dx_i} = \frac{\partial \boldsymbol{F}}{\partial x_i}. \tag{3.12}$$

Solving this equation enables to get the required term $d\boldsymbol{u}/dx_i$. It reads as:

$$\frac{d\boldsymbol{u}}{dx_i} = \mathbf{K}^{-1}\left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}\right). \tag{3.13}$$

Finally, we can substitute equation (3.13) into equation (3.11):

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \mathbf{K}^{-1} \underbrace{\left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}\right)}_{\text{pseudo-load vector}}. \tag{3.14}$$

One can see that the inverse $\mathbf{K}^{-1}$ is involved in the expression of the derivative. To a certain extent, it means that a resolution is required. There are two ways of dealing with this issue which are actually the adjoint and the direct approaches we discussed previously. In the adjoint method, one solves firstly the adjoint problem. The adjoint problem does not depend on the number of design variable. Thus, for each design variable, the adjoint solution is reused. This result reads as follows:

$$\mathbf{K}\boldsymbol{u}^* = \frac{\partial f}{\partial \boldsymbol{u}} \quad \Rightarrow \quad \frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u}\right). \tag{3.15}$$

The direct method solves several linear systems with the so called pseudo-load vectors as the right-hand sides:

$$\mathbf{K}\boldsymbol{v}_i = \frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u} \quad \Rightarrow \quad \frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \boldsymbol{v}_i. \tag{3.16}$$

In direct sensitivity analysis there are potentially as many systems to be solved as design variables. Thus, adjoint sensitivity analysis is often preferred as long as the number of response function involved in the optimization problems is smaller than the number of design variables. If not, there would be a large number of adjoint problems to be solved (*i.e.* more than the number of systems involved in the direct approach).

Finally, let us take the case of the structural compliance expressed in equation (3.9). We can specify the expression of the different terms involved in the general formulation of the derivative (3.14):

$$\frac{\partial f_{\text{cpl}}}{\partial x_i} = \frac{1}{2}\frac{\partial \boldsymbol{F}}{\partial x_i} \cdot \boldsymbol{u}, \qquad \text{and} \qquad \frac{\partial f_{\text{cpl}}}{\partial \boldsymbol{u}} = \frac{1}{2}\boldsymbol{F}. \tag{3.17}$$

In this case, the problem is self-adjoint. In fact, the adjoint problem reads here as:

$$\mathbf{K}\boldsymbol{u}^* = \frac{1}{2}\boldsymbol{F} \quad \Rightarrow \quad \boldsymbol{u}^* = \frac{1}{2}\boldsymbol{u}. \tag{3.18}$$

Thus, the adjoint solution is deduced from the state solution and no additional resolution is needed. Finally, with these development in hand, one can get the following derivative of the compliance:

$$\frac{df_{\text{cpl}}}{dx_i} = \boldsymbol{u} \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{1}{2}\frac{\partial \mathbf{K}}{\partial x_i}\boldsymbol{u} \right). \tag{3.19}$$

The very last point consists now in the computation of the derivative of the stiffness matrix and the load vector with respect to the design variables. This last step is the most challenging. Their expressions depend on the nature of the design variables and also on the finite element formulation. We will show in this work how it is achievable especially in the context of isogeometric shape optimization of shells (see section 3.3).

Semi-Analytical Discrete Sensitivity An efficient way to simplify the calculation of the derivatives of the stiffness matrix and the load vector with respect to the design variables consists in approximating them using finite differences:

$$\begin{aligned} \frac{\partial \boldsymbol{F}}{\partial x_i} &\approx \frac{\Delta \boldsymbol{F}}{\Delta x_i} = \frac{\boldsymbol{F}(x + \varepsilon\, e_i) - \boldsymbol{F}(x)}{\varepsilon}, \\ \frac{\partial \mathbf{K}}{\partial x_i} &\approx \frac{\Delta \mathbf{K}}{\Delta x_i} = \frac{\mathbf{K}(x + \varepsilon\, e_i) - \mathbf{K}(x)}{\varepsilon}, \end{aligned} \tag{3.20}$$

where step length is a small scalar value ε takes is small value as for the global Finite Difference sensitivities (3.10). With semi-analytical discrete sensitivities, one has to build as many stiffness matrices as the number of design variables. However, in contrast with global Finite Difference sensitivities, there is no need to solve the state governing equations for each design variable. Moreover, the stiffness matrix is defined as a sum of elementary matrices of smaller size. One single design variable may not be related to

every elementary matrices. Thus, the stiffness variations are preferably approximated at the elementary level. For the compliance, it gives something of the form:

$$\frac{df_{\text{cpl}}}{dx_i} = \sum_{e \in \mathcal{E}_i} \boldsymbol{u}^e \cdot \left(\frac{\Delta \boldsymbol{F}^e}{\Delta x_i} - \frac{1}{2} \frac{\Delta \mathbf{K}^e}{\Delta x_i} \boldsymbol{u}^e \right), \tag{3.21}$$

where $\mathcal{E}_i$ is the set of active elements accordingly to the design variable x_i.

One interesting feature of semi-analytical sensitivities, besides the appearing simplicity, is the independence to the element formulation. This is very convenient from an implementation point of view. No element dependent routines has to be formulated. Thus, adding a new element formulation to an existing code is straightforward. However, it has been shown that this type of derivatives suffers from a very specific type of approximation error [36, 55, 74, 110, 121]. In fact, they lead to inconsistent derivatives regarding a rigid-body test, as discussed in [36]. Methods have been suggested in order to correct the derivatives, see for example [55, 74, 121]. Without any correction, semi-analytical derivatives are sensible to the mesh size and to the step size. This is especially true for shell formulations as we will show in the following (see numerical experiments in section 3.3).

Others issues There are still much to say about sensitivity analysis for structural shape optimization. Let us lastly tackled few points that appears important to mention even if we will not go into precise details here.

The *influence of the discretization* has been observed in the context of IGA-based shape optimization in the work of Kiendl et al. [72] and theoretical insights on this issue has been performed by Z.-P. Wang et al. [78] and similar results are addressed in [36, 61]. The discrete design derivatives do not represent the continuous shape gradient. This issue can be observed on simple example tacking a rectangular plate for example [72, 78]. They found that an inconsistency originated from the discretization which induces a discrete quadratic norm to represent the continuous Euclidean norm (more information in [78]). However, the discretization-dependency is not specific to isogeometric shape optimization but concerns every Finite Element-based shape optimization [61]. Simple approaches have been suggested to overcome this inconsistency. The idea is to introduce of weighting step, sometimes denoted as the sensitivity weighting. The components of the search direction p are scaled with respect to the geometric influence of the design variables:

$$p_w = \mathbf{B}^{-1} p. \tag{3.22}$$

where $\mathbf{B}$ is the geometric influence matrix. This matrix takes a similar form than classical mass matrices. In order to avoid the resolution of a linear system, diagonally lumped mapping matrices are preferred in practice [78]. If the problem is convex, then taking the normalization or not leads to the same result. However, in case of highly non-convex optimization problems, ignoring the sensitivity weighting possibly results in poorer local optimums [72].

A common task in Node-based shape optimization is the *sensitivity filtering* (or gradient smoothing). Non-smoothness of the gradients appears substantially when the geometry is discretized with low-order function (typically a linear FEM mesh) [58, 59]. In the IGA framework, this task is, as it were, automatically taken into account. The multilevel approach presented previously in section 2.4.2 enables to impose the shape update on a coarser mesh than the one used for the analysis. This construction automatically smooths the shape variations. Moreover, a multi-resolution strategy can be easily put in place within IGA framework. Starting with a very coarse discretization, the optimization model is refined successively each time a minimum is reached. It results in a efficient filtering strategy as shown by Bandara et al. [122, 123]. Based on these observation and our own numerical experiments, we will omit in this work the introduction of a filtering step during the sensitivity analysis. It seems necessary in the case of a very large number of design variables only, *e.g.* up to 10 000 in the IGA framework.

Finally, there still exists other useful tools for the sensitivity analysis that we did not mention yet. For instance, *Automatic Differentiation* has gained popularity [120]. It refers to the differentiation of the computer code itself. If one has access to the source routines, then there are tools that can generate new routines which correspond to the differentiation of the initial ones. Basically, a routine is a collection of elementary operations for which is possible to define partial derivatives. The program applies a long chain rule of differentiation, starting from these elementary operations [124]. Application of these techniques for structural shape optimization can be found in the literature, see for example [125]. Let us also mention the *complex-step* approach which is an accurate alternative to the global Finite Difference sensitivities. It is interesting for cases where a response function is defined as a black-box function only (no access to the source code). The only requirement for the Complex-step approach is that the black-box should handle inputs and outputs of complex type (instead of real). If so, this approach is more accurate than the global Finite Difference method for an identical level of computational cost (and simplicity). Interested readers can refer to [126, 127].

3.2 Shell Analysis

The aim of this work is to build a numerical tool for the design of aeronautical structures, and more specifically stiffened structures. These types of structures are usually referred as thin-walled structures. They are geometrically characterized by one very small dimension, namely the thickness, in comparison with the two others. Since the early development of numerical simulation, specific Finite Element formulations have been introduced for these thin structures in order to analyze them in a accurate and proper way. Efficient shell formulations enable a fast analysis. The analysis time with shell elements can drastically be lower than with standard 3D Finite Elements. Thus, shell elements are widely used by stress engineers and every commercial Finite Element code offers these specific elements. In the present work, we seek to employ two types of

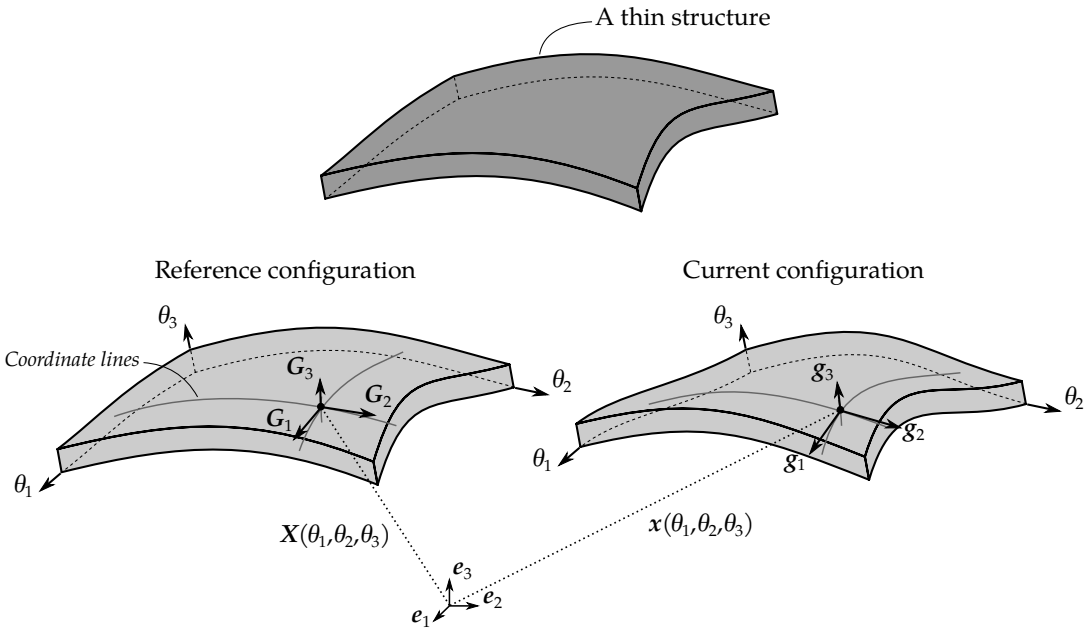


Figure 3.3 – A shell body

shell formulations: a *Kirchhoff–Love shell* formulation and a *solid-shell* formulation.

For isogeometric shape optimization of shells, it seems that Kirchhoff–Love NURBS elements have been mainly considered [71, 72, 122], while few research works adopt a Reissner–Mindlin NURBS shell formulation [76]. However, the influence of the shell formulations on the optimization results is not well-known, and, unfortunately, the choice of one shell formulation rather than another is usually dictated by numerical needs (which should not). Thus, the Kirchhoff–Love NURBS element is attractive due to its low computational cost, but it should be borne in mind that it is restricted to thin shells. In order to provide a broader framework that is also suitable for thicker shells, we propose to consider in this work solid-shell NURBS-based element as well. We will highlight that these elements can bring interesting features in the shell shape optimization framework from the design point of view, in particular for the design of shells with variable thicknesses.

3.2.1 An Isogeometric solid-shell formulation

The solid-shell NURBS element used in this work is the classical one introduced by Bouclier et al. [128]. It is based on a 3D continuum mechanics-type formulation. The idea is to take a 3D solid continuum element and, in order to limit the computational costs, discretize the shell using a single layer of elements through the thickness. This element is used in its raw form and no particular treatment (strain-projection, reduced integration, etc.) is put in place here. To alleviate locking, the basic strategy of increasing the polynomial degree of the analysis model is performed. Despite the simplicity of

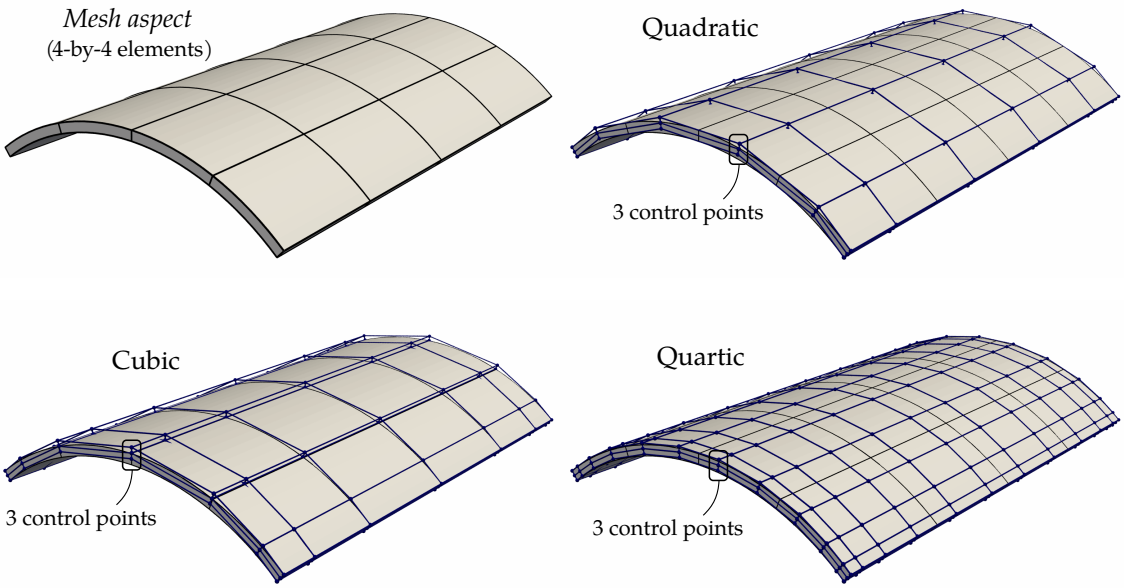


Figure 3.4 – Discretization with the isogeometric standard solid-shell element. Degree 2 is always chosen through the thickness, no matter the degree in the two principal directions.

the formulation, this element has shown similar behaviours to those of Kirchhoff–Love or of Reissner-Mindlin NURBS shell elements through several shell benchmarks [94, 128–130]. Other efficient NURBS-based solid-shell elements have been developed to overcome locking phenomena [128, 131] and to analyze structures with geometric [132, 133] and material [101, 134] non-linearities.

3.2.1.1 The standard solid-shell NURBS element

Since the standard solid-shell formulation is nothing else than the classical 3D solid element, the theoretical aspects are essentially the same than those already presented in section 2.4. The only consideration made here, consists in always selecting degree 2 through the thickness. Based on the observations of Bouclier et al. [128], there is no need to consider higher degree in this direction. Thus, when speaking of solid-shell elements of degree p, the reader should understand that degree p is used in the main directions and that quadratic approximation is set through the thickness. Figure 3.4 depicts this discretization specificity.

Using curvilinear coordinates However, let us re-write the element formulation based on curvilinear coordinates instead of Cartesian coordinates. Using curvilinear coordinates is very convenient to describe complex shapes with curvatures, and it is therefore widely used in shell theory. The position vector associated to a material point is constructed from curvilinear coordinates θ_i as shown in figure 3.3. The spatial mapping

defined through the curvilinear coordinates should preserve distinctness (*i.e.* injective functions). Cylindrical and spherical coordinate systems are typical examples of curvilinear coordinate systems. For instance, locating a point P belonging to a cylinder is simplified with the use of cylindrical coordinates, *i.e.* a radial coordinate r, an angular coordinate θ, and a height z:

$$\boldsymbol{X}(r,\theta,z) = r\cos\theta\boldsymbol{e}_1 + r\sin\theta\boldsymbol{e}_2 + z\boldsymbol{e}_3, \tag{3.23}$$

where $\boldsymbol{X}$ is the vector position associated to the material point P. Curvilinear coordinates can finally be seen as a generalization of this type of geometric transformations.

We denote with capital and lowercase letters all quantities of the reference and deformed configuration, respectively. The displacement field of a material point is defined as the difference vector of the position vectors, $\boldsymbol{x}$ and $\boldsymbol{X}$, of the deformed and reference configuration:

$$\boldsymbol{u}(\theta_1,\theta_2,\theta_3) = \boldsymbol{x}(\theta_1,\theta_2,\theta_3) - \boldsymbol{X}(\theta_1,\theta_2,\theta_3), \tag{3.24}$$

Hence, the displacement vector is defined as a function of the curvilinear coordinates. Once the displacement field is defined, one can measure the deformation using different strain quantities. More specifically, the second order Green–Lagrange strain tensor $\mathbf{E}$ is formulated in curvilinear coordinates as the difference of the covariant metric coefficients, G_{ij} and g_{ij}, of both reference and current configurations [135, 136]:

$$\mathbf{E} = \frac{1}{2}\left(g_{ij} - G_{ij}\right)\boldsymbol{G}^i \otimes \boldsymbol{G}^j, \tag{3.25}$$

There are quantities involved in the strain field (3.25) that have not been introduced yet. We need especially to describe the covariant and contravariant basis vectors. The covariant vectors are represented in figure 3.3. These vectors are tangent to the coordinate lines and are defined as:

$$\boldsymbol{G}_i = \frac{\partial \boldsymbol{X}}{\partial \theta_i} = \boldsymbol{X}_{,i} \quad i = 1,\ldots,3. \tag{3.26}$$

It is commonly adopted to use superscripts for contravariant quantities and subscripts for covariant quantities. The covariant metric coefficients involved in equation (3.25) are computed by the scalar product of covariant basis vectors:

$$G_{ij} = \boldsymbol{G}_i \cdot \boldsymbol{G}_j. \tag{3.27}$$

The covariant metric coefficients for the current configuration is defined equivalently. The contravariant metric coefficients are obtained by the inverse of the covariant coefficient matrix:

$$\left[\, G^{ij} \,\right] = \left[\, G_{ij} \,\right]^{-1}. \tag{3.28}$$

We have now the possibility to express the contravariant basis vectors:

$$\boldsymbol{G}^i = G^{ij}\boldsymbol{G}_j. \tag{3.29}$$

Much more information on these geometric quantities can be found in Kiendl [135] and Echter [136], and in the underlying references cited therein. Finally, for small displacements, the linearized Green-Lagrange strain tensor $\boldsymbol{\varepsilon}$ reads as:

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\boldsymbol{u}_{,i} \cdot \boldsymbol{G}_j + \boldsymbol{u}_{,j} \cdot \boldsymbol{G}_i\right) \boldsymbol{G}^i \otimes \boldsymbol{G}^j. \tag{3.30}$$

In this work, we limit ourselves to this infinitesimal strain tensor.

Under the assumptions of a linear elasticity behavior of the material, the Saint-Venant-Kirchhoff material model is commonly adopted:

$$\mathbf{S} = \mathbf{C} : \mathbf{E}, \tag{3.31}$$

where $\mathbf{S}$ denotes the second Piola-Kirchhoff stress tensor, and $\mathbf{C}$ denotes the elasticity tensor. In the case of small displacements, the stress tensor $\mathbf{S}$ is approximated by the linearized Cauchy stresses $\boldsymbol{\sigma}$. Furthermore, with these assumptions the Saint-Venant-Kirchhoff material law is given as the Hooke's law:

$$\boldsymbol{\sigma} = \mathbf{C} : \varepsilon, \tag{3.32}$$

where the material tensor is defined in the covariant basis:

$$\begin{aligned} \mathbf{C} &= C^{ijkl} \boldsymbol{G}_i \otimes \boldsymbol{G}_j \otimes \boldsymbol{G}_k \otimes \boldsymbol{G}_l, \\ C^{ijkl} &= \lambda G^{ij} G^{kl} + \mu \left(G^{ik} G^{jl} + G^{il} G^{jk} \right). \end{aligned} \tag{3.33}$$

The material parameters λ and μ are the Lamé constants.

Isogeometric formulation The isogeometric framework sticks well with the use of curvilinear coordinates. One can define those curvilinear coordinates as being the geometric parameters of the NURBS. The position vector $\boldsymbol{X}$ is defined by a tri-variate mapping:

$$\boldsymbol{X}(\theta_1, \theta_2, \theta_3) = \sum_k R_k(\theta_1, \theta_2, \theta_3) \boldsymbol{P}_k \quad \text{with} \quad (\theta_1, \theta_2, \theta_3) \in \bar{\Omega}, \tag{3.34}$$

accordingly to the definition of a NURBS volume previously introduced (see section 2.3). The covariant basis vectors are then obtain by partial derivation with respect to the parameters θ_i which gives:

$$\boldsymbol{G}_i(\theta_1, \theta_2, \theta_3) = \sum_k R_{k,i}(\theta_1, \theta_2, \theta_3) \boldsymbol{P}_k. \tag{3.35}$$

Once these vectors in hand, one can compute the covariant metric coefficients G_{ij} using equation (3.27). Then, the contravariant metric coefficients G^{ij} are obtained by inverting a 3-by-3 matrix as given by equation (3.28). Those are the main calculation steps of the formulation. Hence, using curvilinear coordinates is actually not a complex task, especially with IGA. Finally, the internal virtual work reads as:

$$\delta W_{\text{int}}^{\text{SSh}} = -\int_{\bar{\Omega}} \left(\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\right) |J| \, \mathrm{d}\bar{\Omega} \tag{3.36}$$

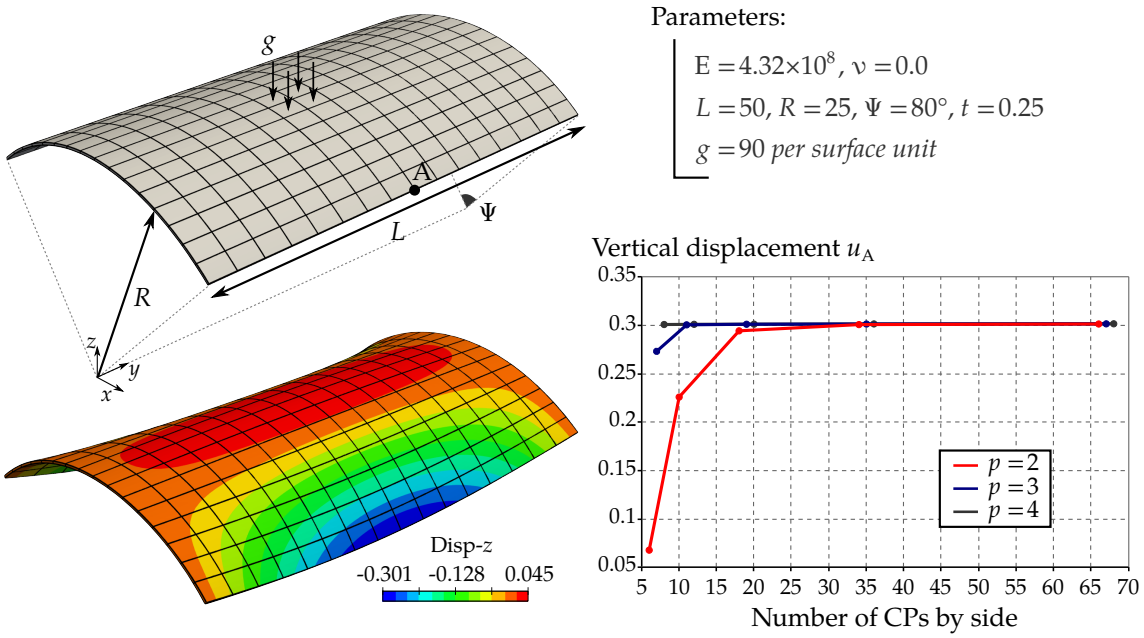


Figure 3.5 – Scordelis-Lo roof benchmark problem: performance study for the standard solid-shell element.

where a change of variable has been made in order to integrate over the parameter space $\bar{\Omega}$. This way, the integration variables are the curvilinear coordinates θ_i (*i.e.* we have $\mathrm{d}\bar{\Omega} = \mathrm{d}\theta_1\mathrm{d}\theta_2\mathrm{d}\theta_3$ in equation (3.36)). The Jacobian $|J|$ of this change of variable can be computed through the following triple product:

$$|J| = (\boldsymbol{G}_1 \times \boldsymbol{G}_2) \cdot \boldsymbol{G}_3. \tag{3.37}$$

For the rest, everything is identical to classical Finite Element formulations: *i.e.* approximation of the displacement field, numerical integration using quadrature rule, etc. Even if the intermediary steps are a bit different, one should keep in mind that the standard solid-shell element written in the curvilinear fashion is strictly the same than the 3D solid continuum element. At the end of the day, the stiffness matrices obtained with both approaches are identical.

We present the solid-shell formulation in the curvilinear framework because it will enable to write the analytical sensitivities in a convenient way (see section 3.3). Moreover, more elaborate solid-shells that integrate additional procedures to alleviate undesirable phenomenon, as locking behaviors, are formulated based on this kinematic [101, 128, 131–134]. We depict in figure 3.5 a short performance study of this standard solid-shell for the Scordelis-Lo roof benchmark example. Same results than Bouclier et al. [128] are obtained.

Imposing shape variations With solid-shell elements, the structure is described by its volume and not by its mid-surface. Thus, the standard strategy depicted in section 2.4.2

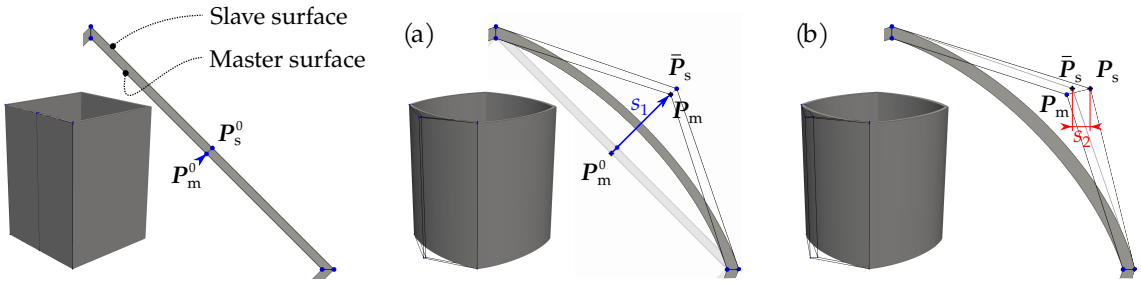


Figure 3.6 – Definition of the design variables in the context of solid-shell isogeometric elements. (a) A first set of design variables updates the master surface and (b) the slave surface follows the shape variation through a second set of variables.

needs to be further improved: the shape control has to deal with volume. A solid-shell is characterized by two outer skins. A natural idea is to define one of the outer skin as the master surface and the other as the slave surface. This is carried out in this work in two steps. A first set of design variables updates the master surface as it is commonly done with shell structure and as already discussed. Then, a second set of design variables moves the slave surface starting from the master surface. This idea is illustrated in figure 3.6 which shows the shape parametrization of a tube. A master-slave shape parametrization avoids adding a lot of geometric constraints to prevent mesh distortion and self penetration (*e.g.* overlapping of the outer skins).

Each control point from the master surface has its equivalent on the slave surface. If a variable s_1 moves the master point from initial position $\boldsymbol{P}_m^0$ to $\boldsymbol{P}_m$ in direction $\boldsymbol{n}_1$ and a second variable s_2 further updates the neighbor slave point from $\boldsymbol{P}_s^0$ to $\boldsymbol{P}_s$ in direction $\boldsymbol{n}_2$, then the design update follows:

$$\begin{aligned} \boldsymbol{P}_m &= \boldsymbol{P}_m^0 + s_1 \boldsymbol{n}_1, \\ \boldsymbol{P}_s &= \boldsymbol{P}_s^0 + s_2 \boldsymbol{n}_1 + s_2 \boldsymbol{n}_2. \end{aligned} \tag{3.38}$$

By setting additional bounds to s_2 we avoid undesirable shapes as for example overlapping of the outer skins. Generally, in shape optimization of shells, the thickness is kept constant. For this purpose, we apply a simple approach where the second set of design variables is not included in the optimization. In other words, same shape variations are applied on both outer skins. If the optimization model has a higher degree through the thickness than 1, then the intermediate control points are interpolated between both outer skin. Finally, the use of solid-shell NURBS element offers the possibility to optimize through the thickness. The thickness optimization issue seems to be promising [137], and it is investigated in the present work (see section 3.4.2).

3.2.2 The Isogeometric Kirchhoff–Love shell

At the beginning of NURBS-based shell formulations, structural model approach was firstly developed (see, *e.g.*, the initial works of Kiendl et al. [94] on the Kirchhoff–

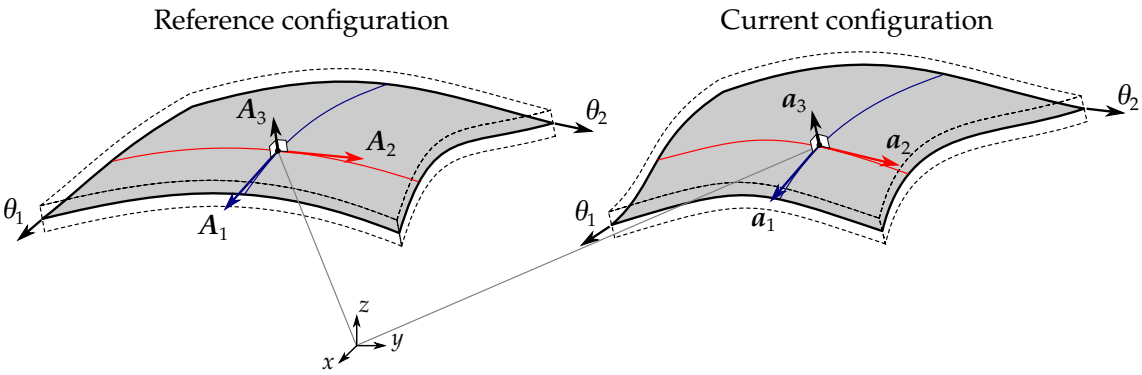


Figure 3.7 – Kirchhoff–Love shell formulation:

Love theory and Benson et al. [129] on the Reissner–Mindlin theory, and then, more recently, Echter et al. [95] and Dornisch et al. [138]). The structural model approach is based on a discretization of the mean surface alone. It leads to fewer degrees of freedom per element in comparison with solid-shell element. This is particularly true with the Kirchhoff–Love formulation where no rotational degrees of freedom are needed. In return, Kirchhoff–Love shells require C^1 continuity but thanks to NURBS surfaces, this condition is easily satisfied. IGA based on NURBS surfaces offers C^{p-1} continuity throughout the element. Finally, one should keep in mind that Kirchhoff–Love shells neglect transverse shear deformations, and thus they are accurate for analyzing thin structures only.

Kinematics The starting point of the Kirchhoff–Love shell formulation consists in invoking specific kinematic assumptions, namely the Kirchhoff–Love hypotheses. These hypotheses introduced by Kirchhoff [139] and Love [140] state that the normals to the mid-surface in the reference configuration remain normal and unstretched in the deformed configuration. Figure 3.7 illustrates this point.

There are different strategies to impose the kinematic assumptions. Principally, shell models can either be obtained via the *direct approach* or either the *derivation from three-dimensional continuum mechanics* [141]. The Kirchhoff–Love shell formulation introduced by Kiendl et al. [94] is based on the direct approach. It means that the shell is regarded from the beginning as a two-dimensional surface (often named as a Cosserat surface) and proper kinematic assumptions, representing the three-dimensional behavior, are postulated [135]. Within this approach, the shell continuum is described by its mid-surface $\boldsymbol{S}$ and director vectors:

$$\boldsymbol{X}(\theta_1,\theta_2,\theta_3) = \boldsymbol{S}(\theta_1,\theta_2) + \theta_3 \boldsymbol{A}_3(\theta_1,\theta_2), \quad \theta_3 \in [-\tfrac{t}{2},\tfrac{t}{2}], \tag{3.39}$$

where t is the shell thickness. In the case of Kirchhoff–Love kinematic, the director vector is taken as the normal at each point of the mid-surface:

$$\boldsymbol{A}_3 = \frac{1}{A}\boldsymbol{A}_1 \times \boldsymbol{A}_2, \quad \text{with} \quad A = |\boldsymbol{A}_1 \times \boldsymbol{A}_2|. \tag{3.40}$$

Putting equation (3.39) into equation (3.26) gives the expression of the covariant basis vectors within Kirchhoff–Love theory:

$$\begin{aligned} \boldsymbol{G}_\alpha &= \boldsymbol{A}_\alpha + \theta_3 \boldsymbol{A}_{3,\alpha} \ , \\ \boldsymbol{G}_3 &= \boldsymbol{A}_3 \ . \end{aligned} \tag{3.41}$$

One can operate identically to describe the current configuration. We denote by $\boldsymbol{v}$ the displacement field of the mid-surface. Thus, the position vector of the current configuration reads as:

$$\boldsymbol{x}(\theta_1, \theta_2, \theta_3) = \big[\boldsymbol{S} + \boldsymbol{v}\big](\theta_1, \theta_2) \ + \ \theta_3 \boldsymbol{a}_3(\theta_1, \theta_2). \tag{3.42}$$

We can write the displacement field accordingly to equation (3.24):

$$\boldsymbol{u}(\theta_1, \theta_2, \theta_3) = \boldsymbol{v}(\theta_1, \theta_2) + \theta_3 \big[\boldsymbol{a}_3 - \boldsymbol{A}_3\big](\theta_1, \theta_2). \tag{3.43}$$

Since we make the assumption of small displacements, we will neglect the quadratic term *w.r.t.* the mid-surface displacement $\boldsymbol{v}$ in the difference vector $\mathbf{w} = \boldsymbol{a}_3 - \boldsymbol{A}_3$. To this purpose, let us write the Taylor's theorem in Banach spaces:

$$\mathbf{w}(\boldsymbol{v}) = \mathbf{w}(0) + \mathbf{w}^{\text{lin}}(\boldsymbol{v}) + \mathcal{O}(||\boldsymbol{v}||^2), \tag{3.44}$$

where $\mathbf{w}(0) = 0$, and the linearized vector $\mathbf{w}^{\text{lin}}$ is given by:

$$\mathbf{w}^{\text{lin}}(\boldsymbol{v}) = \frac{1}{A}\left\{ \Big(\boldsymbol{v}_{,1} \times \boldsymbol{A}_2 + \boldsymbol{A}_1 \times \boldsymbol{v}_{,2}\Big) - \boldsymbol{A}_3 \Big[\boldsymbol{A}_3 \cdot \Big(\boldsymbol{v}_{,1} \times \boldsymbol{A}_2 + \boldsymbol{A}_1 \times \boldsymbol{v}_{,2}\Big)\Big] \right\}. \tag{3.45}$$

This quantity is obtained by differentiating the vector $\mathbf{w}$ *w.r.t.* the mid-surface displacement $\boldsymbol{v}$. Mathematically, it is the Fréchet derivative of $\mathbf{w}$ at $\boldsymbol{v} = 0$. Equivalent derivative steps are done in the work of Kiendl [135, see for instance equation (5.26)]. It is also interesting to make the link with the expressions presented by Echter [95, 136]. They write the linearized difference vector as:

$$\mathbf{w}^{[95]} = \boldsymbol{\Phi} \times \boldsymbol{A}_3, \tag{3.46}$$

where the rotation vector $\boldsymbol{\Phi}$ is given by:

$$\boldsymbol{\Phi} = \varphi_1 \boldsymbol{A}_1 + \varphi_2 \boldsymbol{A}_2, \tag{3.47}$$

and the rotation angles φ_α are expressed as follows:

$$\varphi_1 = \frac{1}{A} \boldsymbol{v}_{,2} \cdot \boldsymbol{A}_3 \qquad \text{and} \qquad \varphi_2 = -\frac{1}{A} \boldsymbol{v}_{,1} \cdot \boldsymbol{A}_3. \tag{3.48}$$

It possible to prove that it leads to identical formulas (*i.e.* $\mathbf{w}^{\text{lin}} = \mathbf{w}^{[95]}$). In order to prove it, one can compute the scalar product between these linearized difference vectors and

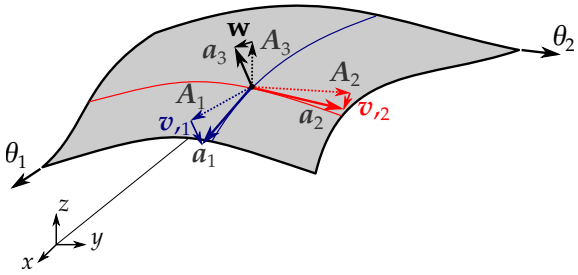


Figure 3.8 – Link between the local basis vectors of the reference and the current configurations.

the surface covariant basis vectors $\boldsymbol{A}_i$. Let us write few steps of the calculation with the vector $\boldsymbol{A}_1$:

$$\begin{aligned}\mathbf{w}^{\text{lin}} \cdot \boldsymbol{A}_1 &= \frac{1}{A}\left(\boldsymbol{v}_{,1} \times \boldsymbol{A}_2\right) \cdot \boldsymbol{A}_1 \\ &= \frac{1}{A}\left(\boldsymbol{A}_2 \times \boldsymbol{A}_1\right) \cdot \boldsymbol{v}_{,1} \\ &= -\boldsymbol{v}_{,1} \cdot \boldsymbol{A}_3\end{aligned} \qquad \begin{aligned}\mathbf{w}^{[95]} \cdot \boldsymbol{A}_1 &= \left(\boldsymbol{\Phi} \times \boldsymbol{A}_3\right) \cdot \boldsymbol{A}_1 \\ &= \varphi_2 \left(\boldsymbol{A}_2 \times \boldsymbol{A}_3\right) \cdot \boldsymbol{A}_1 \\ &= -\boldsymbol{v}_{,1} \cdot \boldsymbol{A}_3\end{aligned}$$

We obtain the same result in both cases. For the two other vectors, $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$, we proceed the same way. One should get:

$$\mathbf{w}^{\text{lin}} \cdot \boldsymbol{A}_2 = \mathbf{w}^{[95]} \cdot \boldsymbol{A}_2 = -\boldsymbol{v}_{,2} \cdot \boldsymbol{A}_3 \qquad \text{and} \qquad \mathbf{w}^{\text{lin}} \cdot \boldsymbol{A}_3 = \mathbf{w}^{[95]} \cdot \boldsymbol{A}_3 = 0.$$

Based on these observations, we conclude that both definitions of the linearized difference vectors (*i.e.* equations (3.45) and (3.46)) are identical. This is very interesting to notice since the numerical results obtained by Echter et al. [95] are slightly different than those obtained by Kiendl et al. [94]. The authors in [95] mention these differences without understanding where they come from [95, see page 179]. By using $\mathbf{w}^{\text{lin}}$ in the following, we will manage to recover the formulation introduced by Kiendl et al. [94]. This makes us think that the theory presented in both papers [94, 95] is identical.

Let us continue our developments. To summarize, we wrote the shell body displacement field as:

$$\boldsymbol{u}(\theta_1, \theta_2, \theta_3) = \boldsymbol{v}(\theta_1, \theta_2) + \theta_3 \mathbf{w}^{\text{lin}}(\theta_1, \theta_2). \tag{3.49}$$

In order to express the deformation field, one can put equations (3.49) and (3.41) into the expression of the linearized Green-Lagrange strains (3.30). For instance, we can take a closer look at the component ε_{11}:

$$\begin{aligned}\varepsilon_{11} &= \boldsymbol{u}_{,1} \cdot \boldsymbol{G}_1 \\ &= \left(\boldsymbol{v}_{,1} + \theta_3 \mathbf{w}^{\text{lin}}{}_{,1}\right) \cdot \left(\boldsymbol{A}_1 + \theta_3 \boldsymbol{A}_{3,1}\right) \\ &= \boldsymbol{v}_{,1} \cdot \boldsymbol{A}_1 \;+\; \theta_3 \left(\boldsymbol{v}_{,1} \cdot \boldsymbol{A}_{3,1} + \mathbf{w}^{\text{lin}}{}_{,1} \cdot \boldsymbol{A}_1\right) \;+\; (\theta_3)^2 \mathbf{w}^{\text{lin}}{}_{,1} \cdot \boldsymbol{A}_{3,1}\,.\end{aligned} \tag{3.50}$$

The thickness of the shell is considered thin in the Kirchhoff–Love shell theory, thus the last term which is quadratic *w.r.t.* the thickness coordinate θ_3 is neglected. The

same assumption is made on the other strain components. This way, the strains can be separated into a constant part, namely the *membrane strains* $\boldsymbol{e}$, and a linear part, namely the *bending strains* $\boldsymbol{\kappa}$. Furthermore, all the transversal strains vanish, *i.e.* $\varepsilon_{i3} = 0$. Mathematically, the strains is decomposed as follows:

$$\boldsymbol{\varepsilon} = \boldsymbol{e} + \theta_3 \boldsymbol{\kappa}, \tag{3.51}$$

where the covariant membrane components are:

$$\boldsymbol{e}_{\alpha\beta} = \frac{1}{2}\left(\boldsymbol{v}_{,\alpha} \cdot \boldsymbol{A}_\beta + \boldsymbol{v}_{,\beta} \cdot \boldsymbol{A}_\alpha\right). \tag{3.52}$$

Greek indices (α, β) takes on values 1 or 2. The covariant bending components are:

$$\begin{aligned}\boldsymbol{\kappa}_{\alpha\beta} = -\boldsymbol{v}_{,\alpha\beta} \cdot \boldsymbol{A}_3 + \frac{1}{A}\Big(\boldsymbol{v}_{,1} \cdot (\boldsymbol{A}_{\alpha,\beta} \times \boldsymbol{A}_2) + \boldsymbol{v}_{,2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha,\beta})\Big) \\ + \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha,\beta}}{A}\Big(\boldsymbol{v}_{,1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \boldsymbol{v}_{,2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1)\Big).\end{aligned} \tag{3.53}$$

We do not give all the mathematical developments one has to do in order to recover these expressions. However, having equation (3.50) in hand almost directly leads to the final expressions (3.52) and (3.53). One can also verify that these expressions are equivalent to what is presented in [94, 135], among many others.

Material tensor The constitutive equations are obtained under some simplifications. Here, we do not introduce the expressions of the vectors $\boldsymbol{G}_i$, *i.e.* equation (3.41), into the 3D continuum material law (3.33). In fact, when deriving shell theories by the direct approach, two-dimensional material equations are postulated [141]. The constitutive tensor will rely on the mid-surface metrics $A^{\alpha\beta}$ instead of G^{ij}. More specifically, the isotropic material tensor is obtained by replacing G^{ij} by $A^{\alpha\beta}$ in equation 3.33. It reads as:

$$\bar{C}^{\alpha\beta\gamma\delta} = \bar{\lambda} A^{\alpha\beta} A^{\gamma\delta} + \mu\left(A^{\alpha\gamma} A^{\beta\delta} + A^{\alpha\delta} A^{\beta\gamma}\right), \quad \bar{\lambda} = \frac{2\lambda\mu}{\lambda + 2\mu}. \tag{3.54}$$

The material parameter $\bar{\lambda}$ appears instead of the Lamé's first parameter λ. An additional condensation of the material equations has been operated in order to include the plane-stress condition. This is a common measure to eliminate the contradiction of having $\varepsilon_{33} = 0$ and $\boldsymbol{\sigma}^{33} = 0$ at the same time (without having a Poisson's ratio $\nu = 0$) [141]. One can also rewrite the material tensor in terms of Young's modulus and Poisson's ratio:

$$\bar{C}^{\alpha\beta\gamma\delta} = \frac{E}{1-\nu^2}\left[\nu A^{\alpha\beta} A^{\gamma\delta} + \frac{1}{2}(1-\nu)\left(A^{\alpha\gamma} A^{\beta\delta} + A^{\alpha\delta} A^{\beta\gamma}\right)\right]. \tag{3.55}$$

Stress resultants We already mention that plane-stress state is postulated. Moreover, with the constitutive tensor $\bar{\mathbf{C}}$ just introduced, the stress field is also written as the sum of membrane stresses and bending stresses:

$$\boldsymbol{\sigma} = \bar{\mathbf{C}} : \boldsymbol{e} + \theta_3 \bar{\mathbf{C}} : \boldsymbol{\kappa}. \tag{3.56}$$

Let us rewrite the internal virtual work for the Kirchhoff–Love shell:

$$\begin{aligned}\delta W_{\text{int}} &= -\int_{\bar{\Omega}} \left(\boldsymbol{\sigma} : \delta\boldsymbol{\varepsilon}\right) |J| \, \mathrm{d}\bar{\Omega} \\ &= -\int_{\bar{\Omega}} \left(\boldsymbol{e} : \bar{\mathbf{C}} : \delta\boldsymbol{e} + \theta_3 \left[\boldsymbol{e} : \bar{\mathbf{C}} : \delta\boldsymbol{\kappa} + \boldsymbol{\kappa} : \bar{\mathbf{C}} : \delta\boldsymbol{e}\right] + (\theta_3)^2 \boldsymbol{\kappa} : \bar{\mathbf{C}} : \delta\boldsymbol{\kappa}\right) |J| \, \mathrm{d}\bar{\Omega}\,.\end{aligned} \tag{3.57}$$

At this point, it is quasi possible to integrate analytically through the thickness. Indeed, the membrane strains, the bending strains, and the material tensor do not depend on the thickness coordinates θ_3. In order to integrate analytically, we assume this equally:

$$\int_{\bar{\Omega}} (\quad) |J| \mathrm{d}\theta_1 \mathrm{d}\theta_2 \mathrm{d}\theta_2 = \int_{\bar{\Omega}_0} \left\{ \int_{-t/2}^{t/2} (\quad) \, \mathrm{d}\theta_3 \right\} A \mathrm{d}\theta_1 \mathrm{d}\theta_2, \tag{3.58}$$

which is exactly satisfied in case of a plate with no curvature. The assumption made here is identical to the one involved in the constitutive law (see discussion associated to equation (3.54)). Now let us perform three intermediary computations:

$$\int_{-t/2}^{t/2} 1 \, \mathrm{d}\theta_3 = t, \qquad \int_{-t/2}^{t/2} \theta_3 \, \mathrm{d}\theta_3 = 0, \qquad \int_{-t/2}^{t/2} (\theta_3)^2 \, \mathrm{d}\theta_3 = \frac{t^3}{12}.$$

Finally, the internal virtual work for the Kirchhoff–Love shell formulation reads as:

$$\delta W_{\text{int}}^{\text{KL}} = -\int_{\bar{\Omega}_0} \left(\boldsymbol{n} : \delta\boldsymbol{e} + \boldsymbol{m} : \delta\boldsymbol{\kappa}\right) A \, \mathrm{d}\bar{\Omega}_0, \tag{3.59}$$

where $\boldsymbol{n}$ and $\boldsymbol{m}$ denote the normal forces and the bending moments respectively. They are expressed as follows:

$$\boldsymbol{n} = t\,\bar{\mathbf{C}} : \boldsymbol{e} \qquad\qquad \boldsymbol{n}^{\alpha\beta} = \frac{tE}{1-\nu^2} H^{\alpha\beta\gamma\delta} \boldsymbol{e}_{\gamma\delta}, \tag{3.60}$$

$$\boldsymbol{m} = \frac{t^3}{12}\bar{\mathbf{C}} : \boldsymbol{\kappa} \qquad\qquad \boldsymbol{m}^{\alpha\beta} = \frac{t^3 E}{12(1-\nu^2)} H^{\alpha\beta\gamma\delta} \boldsymbol{\kappa}_{\gamma\delta}, \tag{3.61}$$

with:

$$H^{\alpha\beta\gamma\delta} = \nu A^{\alpha\beta} A^{\gamma\delta} + \frac{1}{2}(1-\nu)\left(A^{\alpha\gamma} A^{\beta\delta} + A^{\alpha\delta} A^{\beta\gamma}\right). \tag{3.62}$$

Isogeometric formulation The Kirchhoff–Love NURBS element is obtained by discretizing the mid-surface with a NURBS surface. This discretization is also used to approximate the mid-surface displacement field:

$$\boldsymbol{S}(\theta_1, \theta_2) = \sum_k R_k(\theta_1, \theta_2) \boldsymbol{P}_k, \quad \text{and} \quad \boldsymbol{v}^h(\theta_1, \theta_2) = \sum_k R_k(\theta_1, \theta_2) \boldsymbol{v}_k. \tag{3.63}$$

By putting these two discretized quantities in the equations presented before, *i.e.* (3.39) to (3.62), one can get the so-called elementary stiffness matrices. For instance, the discretized membrane strains read as:

$$\boldsymbol{e}^h = \sum_k \mathbf{B}_k^m \, \boldsymbol{v}_k \quad \rightarrow \quad \begin{pmatrix} \boldsymbol{e}_{11}^h \\ \boldsymbol{e}_{22}^h \\ 2\boldsymbol{e}_{12}^h \end{pmatrix} = \sum_k \begin{bmatrix} R_{k,1} \, \boldsymbol{A}_1^T \\ R_{k,2} \, \boldsymbol{A}_2^T \\ R_{k,1} \, \boldsymbol{A}_1^T + R_{k,2} \, \boldsymbol{A}_2^T \end{bmatrix} \begin{pmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \end{pmatrix}, \tag{3.64}$$

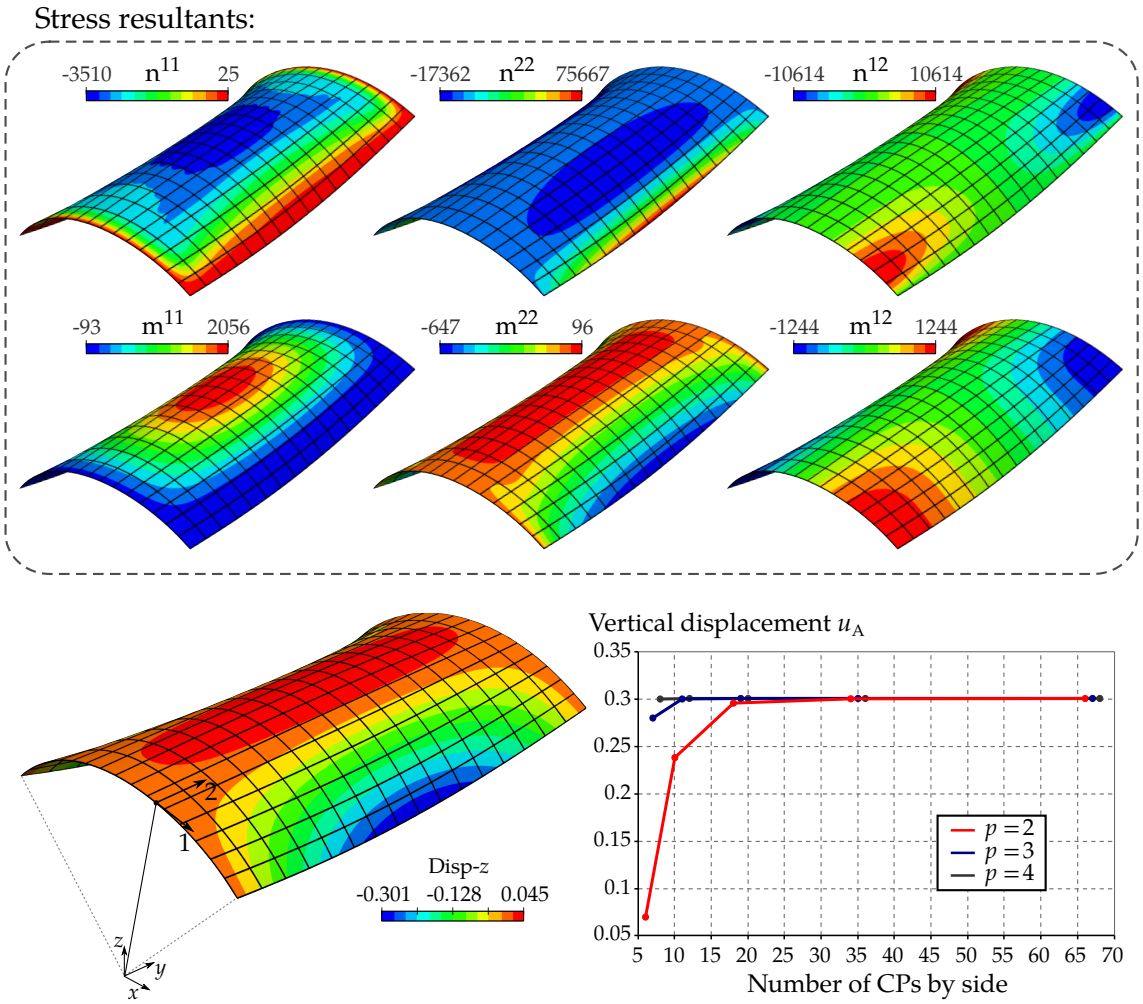


Figure 3.9 – Scordelis-Lo roof benchmark problem: performance study for the isogeometric Kirchhoff–Love element.

where $v_{1ik} = \boldsymbol{v}_k \cdot \boldsymbol{e}_i$ denotes the i^{th} component of the displacement DOFs associated to the control point $\boldsymbol{P}_k$. Identically to the membrane matrices $\mathbf{B}_k^m$, bending matrices $\mathbf{B}_k^b$ enables to describe the discrete bending strains as follow:

$$\boldsymbol{\kappa}^h = \sum_k \mathbf{B}_k^b \boldsymbol{v}_k \quad \rightarrow \quad \begin{pmatrix} \boldsymbol{\kappa}_{11}^h \\ \boldsymbol{\kappa}_{22}^h \\ 2\boldsymbol{\kappa}_{12}^h \end{pmatrix} = \sum_k \begin{bmatrix} \boldsymbol{B}_{k,11}{}^T \\ \boldsymbol{B}_{k,22}{}^T \\ 2\boldsymbol{B}_{k,12}{}^T \end{bmatrix} \begin{pmatrix} v_{1k} \\ v_{2k} \\ v_{3k} \end{pmatrix}, \tag{3.65}$$

with:

$$\begin{aligned} \boldsymbol{B}_{k,\alpha\beta} = -R_{k,\alpha\beta}\boldsymbol{A}_3 + \frac{1}{A}\Big(R_{k,1}\boldsymbol{A}_{\alpha,\beta} \times \boldsymbol{A}_2 + R_{k,2}\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha,\beta}\Big) \\ + \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha,\beta}}{A}\Big(R_{k,1}\boldsymbol{A}_2 \times \boldsymbol{A}_3 + R_{k,2}\boldsymbol{A}_3 \times \boldsymbol{A}_1\Big). \end{aligned} \tag{3.66}$$

The material tensor can be written as a 3-by-3 symmetric matrix using the Voigt notation. For this purpose, one can build the following matrix:

$$\mathbf{H} = \frac{E}{1-\nu^2}\begin{bmatrix} H^{11\,11} & H^{11\,22} & H^{11\,12} \\ * & H^{22\,22} & H^{22\,12} \\ * & * & H^{12\,12} \end{bmatrix}, \tag{3.67}$$

where the components $H^{\alpha\beta\gamma\delta}$ are given by equation (3.62). Then, the stiffness matrix of the Kirchhoff–Love shell formulation can be built through 3-by-3 matrices of the form:

$$\mathbf{K}^e_{kl} = \int_{\bar{\Omega}^e_0} \left[t\mathbf{B}^{m\,T}_k \mathbf{H}\mathbf{B}^m_l + \frac{t^3}{12}\mathbf{B}^{b\,T}_k \mathbf{H}\mathbf{B}^b_l \right] A \mathrm{d}\theta_1 \mathrm{d}\theta_2, \tag{3.68}$$

where k and l are indices of two control points related to element e. The integral is later computed using numerical integration. For the external loads, three types are considered: distributed loads $\boldsymbol{p}$ per unit of area applied on the mid-surface, axial forces $\boldsymbol{t}$ per unit of length applied on the edges of the patch, and concentrated loads applied at the vertices of the patch. The load vector, which expresses the external virtual work once the displacement field discretized, reads as:

$$\boldsymbol{F}_k = \int_{\Omega^p_0} R_k \boldsymbol{p}\, \mathrm{d}\Omega_0 + \int_{\Gamma^t_0} R_k \boldsymbol{t}\, \mathrm{d}\Gamma_0 \,. \tag{3.69}$$

Once the element is implemented, one can go through different benchmark examples. For instance, we depict in figure 3.9 the results obtained for the Scordelis-Lo roof shell problem. We get the exact same results than Kiendl et al. [94]. One can also noticed that the performance of the isogeometric Kirchhoff–Love shell is very similar to the standard isogeometric solid-shell presented in figure 3.5. In order to post-process the results, the stress resultants commonly are transformed into a local Cartesian basis $\boldsymbol{E}_i$ to get physical values [135]:

$$\hat{\boldsymbol{n}}^{\alpha\beta} = \boldsymbol{n}^{\gamma\delta}\left(\boldsymbol{E}_\alpha \cdot \boldsymbol{A}_\gamma\right)\left(\boldsymbol{A}_\delta \cdot \boldsymbol{E}_\beta\right). \tag{3.70}$$

The local Cartesian basis can for example be constructed as follows:

$$\boldsymbol{E}_1 = \frac{\boldsymbol{A}_1}{|\boldsymbol{A}_1|}, \qquad \boldsymbol{E}_3 = \boldsymbol{A}_3, \qquad \boldsymbol{E}_2 = \boldsymbol{E}_3 \times \boldsymbol{E}_1. \tag{3.71}$$

3.3 Analytical Sensitivity in IGA

At present, we know how to perform the shell analysis and we gave the first development steps for the sensitivity analysis (see section 3.1.3). Now, we show how it is possible to perform analytical sensitivity analysis in the context of linear elasticity problems. More specifically, we deal with the standard IGA solid-shell and the IGA

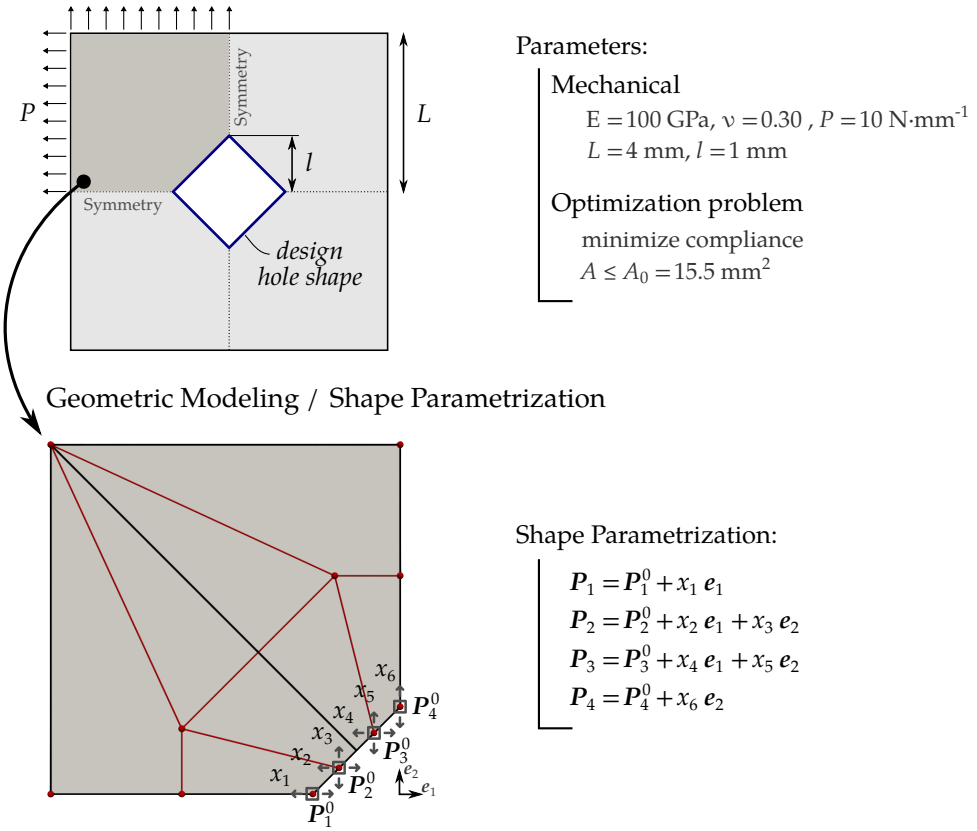


Figure 3.10 – Shape parametrization for the plate with a hole and settings of the problem.

Kirchhoff–Love shell. For the sake of clarity, we repeat equation (3.15) which is our starting point here:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i} \boldsymbol{u} \right).$$

In this section we show how the quantities $\partial \boldsymbol{F}/\partial x_i$ and $\partial \mathbf{K}/\partial x_i$ can be computed.

3.3.1 Sensitivity propagation

3.3.1.1 Linking the Design Variables to the Optimization model

The first step consists in changing the derivatives *w.r.t.* the design variables into derivatives *w.r.t.* the control points of the optimization model. To this end, we will use the relations between the design variables and these control points. This link is given by the so-called *parametrization* step. The coordinates of the control points are expressed as functions of several parameters (the design variables). A simple parametrization lies in defining one independent design variable by movable control point. Each variable moves its associated control point in a specific direction:

$$\boldsymbol{P}_k = \boldsymbol{P}_k^0 + x_k \boldsymbol{n}_k. \tag{3.72}$$

Identically, several design variables can be associated to a single control point. For example, one variable moves the control point in x-direction and the second variable in direction y. Furthermore, it can be interesting to link one specific variable to multiple control points in order, for example, to preserve a geometric continuity or a symmetry. This more general shape parametrization takes the form:

$$\boldsymbol{P}_k = \boldsymbol{P}_k^0 + \sum_{i \in \mathcal{D}_k} x_i \boldsymbol{n}_i^k \,, \tag{3.73}$$

where $\mathcal{D}_k$ is the set of design variables acting on the k^{th} control point. An example of design parametrization is given in figure 3.10 for the problem of a plate with a hole.

By abuse of notation, let us express the derivatives *w.r.t.* the design variables through the derivatives *w.r.t.* the control points of the optimization model:

$$\frac{\partial \bullet}{\partial x_i} = \frac{\partial \boldsymbol{P}}{\partial x_i} \frac{\partial \bullet}{\partial \boldsymbol{P}} \tag{3.74}$$

where the symbol $\bullet$ denotes either the stiffness matrix or the load vector. The derivative of the parametrization is given by:

$$\frac{\partial \boldsymbol{P}}{\partial x_i} = \left(\frac{\partial \boldsymbol{P}_1}{\partial x_i}, \frac{\partial \boldsymbol{P}_2}{\partial x_i}, \dots, \frac{\partial \boldsymbol{P}_{n_{cp}}}{\partial x_i} \right) \qquad \text{with} \qquad \frac{\partial \boldsymbol{P}_k}{\partial x_i} = \begin{pmatrix} \frac{\partial P_{1k}}{\partial x_i} & \frac{\partial P_{2k}}{\partial x_i} & \frac{\partial P_{3k}}{\partial x_i} \end{pmatrix}, \tag{3.75}$$

where P_{1k}, P_{2k}, and P_{3k} are the three Cartesian components (in case of 3D problems) of control point $\boldsymbol{P}_k$. With similar notations, the derivative *w.r.t.* the control points reads as:

$$\frac{\partial \bullet}{\partial \boldsymbol{P}} = \left(\frac{\partial \bullet}{\partial \boldsymbol{P}_1}, \frac{\partial \bullet}{\partial \boldsymbol{P}_2}, \dots, \frac{\partial \bullet}{\partial \boldsymbol{P}_{n_{cp}}} \right) \qquad \text{with} \qquad \frac{\partial \bullet}{\partial \boldsymbol{P}_k} = \begin{pmatrix} \frac{\partial \bullet}{\partial P_{1k}} & \frac{\partial \bullet}{\partial P_{2k}} & \frac{\partial \bullet}{\partial P_{3k}} \end{pmatrix}. \tag{3.76}$$

Let us rewrite the equation (3.74) and give the complete expression of the derivative:

$$\frac{\partial \bullet}{\partial x_i} = \sum_{k=1}^{n_{cp}} \sum_{j=1}^{3} \frac{\partial P_{jk}}{\partial x_i} \frac{\partial \bullet}{\partial P_{jk}}. \tag{3.77}$$

Therefore, the first step to compute the analytical sensitivities is the derivation of the control point coordinates *w.r.t.* the design variables. In case of the shape parametrization of the form (3.73), this first step gives:

$$\frac{\partial \boldsymbol{P}}{\partial x_i} = \left(\boldsymbol{n}_i^1, \boldsymbol{n}_i^2, \dots, \boldsymbol{n}_i^{n_{cp}} \right). \tag{3.78}$$

Thus, for each design variable x_i, one has to build the operator $\partial \boldsymbol{P} / \partial x_i$. It can be defined as a matrix with size $n_{cp} \times 3$ or a vector with length $3n_{cp}$ (n_{cp} being the number of control points of the optimization model). Furthermore, these operators are very sparse and one should take advantages of this sparsity for numerical efficiency. In fact, one design variable moves only few control points. Finally, with the shape parametrization of the form (3.73), the operators remain unchanged during the whole resolution of the

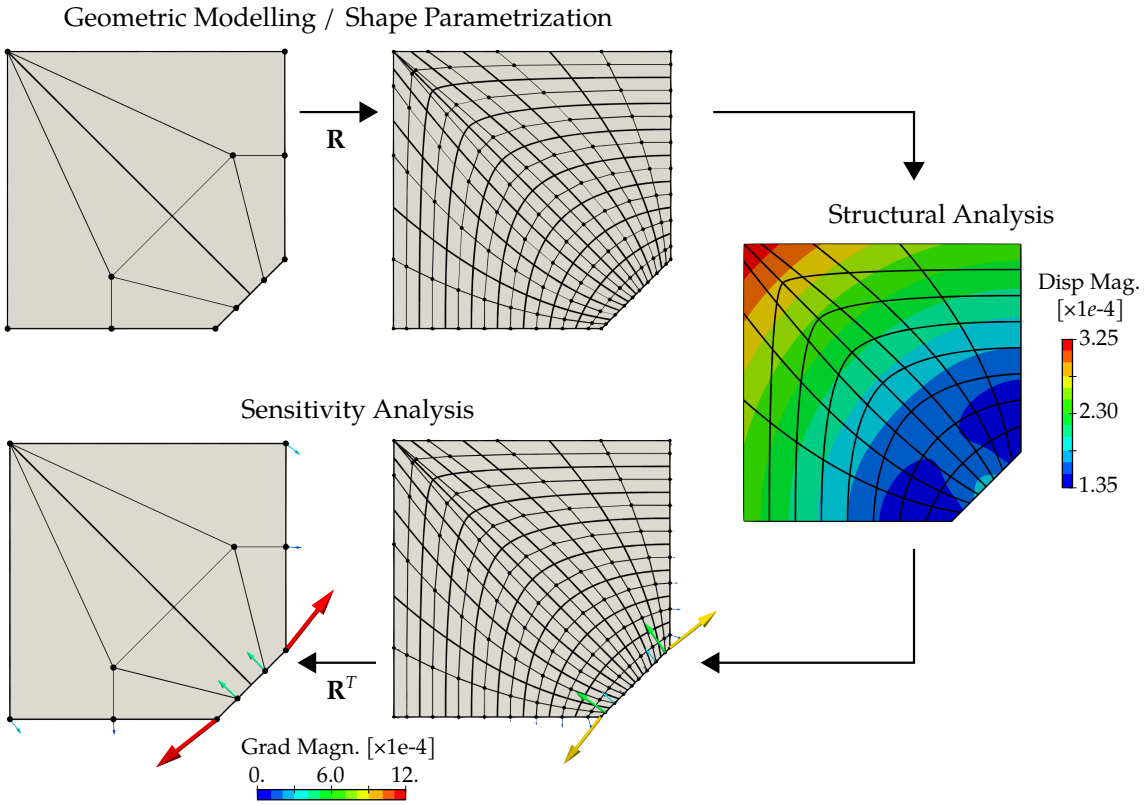


Figure 3.11 – Performing the sensitivity analysis in IGA based shape optimization.

optimization problem (see equation (3.78)). They are build once and for all at the beginning. For the example of the plate with a hole (see figure 3.10), the derivatives of the shape parametrization are the following:

$$\frac{\partial \boldsymbol{P}}{\partial x_1} = \left(\boldsymbol{e}_1, 0, 0, 0, \cdots\right) \qquad \frac{\partial \boldsymbol{P}}{\partial x_3} = \left(0, \boldsymbol{e}_2, 0, 0, \cdots\right) \qquad \frac{\partial \boldsymbol{P}}{\partial x_5} = \left(0, 0, \boldsymbol{e}_2, 0, \cdots\right)$$
$$\frac{\partial \boldsymbol{P}}{\partial x_2} = \left(0, \boldsymbol{e}_1, 0, 0, \cdots\right) \qquad \frac{\partial \boldsymbol{P}}{\partial x_4} = \left(0, 0, \boldsymbol{e}_1, 0, \cdots\right) \qquad \frac{\partial \boldsymbol{P}}{\partial x_6} = \left(0, 0, 0, \boldsymbol{e}_2, \cdots\right)$$

where the dots symbol $(\cdots)$ indicates that the remaining components are filled with zero. It corresponds to the control points that are fixed.

3.3.1.2 From the Optimization model to the Analysis model

The stiffness matrix and the load vector are constructed using the analysis model. Therefore, the derivatives *w.r.t.* the control points of the optimization need to be transformed into derivatives *w.r.t.* the control points of the analysis model. To that purpose, we operate the chain rule:

$$\frac{\partial \bullet}{\partial \boldsymbol{P}} = \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{P}} \frac{\partial \bullet}{\partial \boldsymbol{Q}}. \tag{3.79}$$

The link between the control points $\boldsymbol{P}$ of the optimization model and the control points $\boldsymbol{Q}$ of the analysis model is given by the refinement matrix $\mathbf{R}$ (see the refinement procedures presented in section 2.3.2). It gives:

$$\boldsymbol{Q} = \mathbf{R}\boldsymbol{P} \qquad \rightarrow \qquad \frac{\partial \boldsymbol{Q}}{\partial \boldsymbol{P}} = \mathbf{R}. \tag{3.80}$$

Thus, the chain rule (3.79) takes the form:

$$\frac{\partial \bullet}{\partial \boldsymbol{P}} = \mathbf{R}^T \frac{\partial \bullet}{\partial \boldsymbol{Q}}. \tag{3.81}$$

The refinement matrix $\mathbf{R}$ is a sparse rectangular matrix with size $m_{cp} \times n_{cp}$ (m_{cp} being the number of control points of the analysis model). Hence, the gradients $\partial\bullet/\partial\boldsymbol{P}$ and $\partial\bullet/\partial\boldsymbol{Q}$ should have appropriate shapes so that equation (3.81) makes sense. Thus, similarly to equation (3.76), the derivatives *w.r.t.* the control points of the analysis model takes the form of a matrix of size $m_{cp} \times 3$:

$$\frac{\partial \bullet}{\partial \boldsymbol{Q}} = \left(\frac{\partial \bullet}{\partial \boldsymbol{Q}_1}, \frac{\partial \bullet}{\partial \boldsymbol{Q}_2}, \dots, \frac{\partial \bullet}{\partial \boldsymbol{Q}_{m_{cp}}} \right) \qquad \text{with} \qquad \frac{\partial \bullet}{\partial \boldsymbol{Q}_k} = \begin{pmatrix} \frac{\partial \bullet}{\partial Q_{1k}} & \frac{\partial \bullet}{\partial Q_{2k}} & \frac{\partial \bullet}{\partial Q_{3k}} \end{pmatrix}. \tag{3.82}$$

Finally, let us notice that the refinement matrix does not change, as long as the refinement levels of both optimization and analysis models remain unchanged. Thus, the matrix is commonly built once and for all at the beginning, and can be reused during the optimization process. In addition, this matrix is sparse and of moderate size (in comparison with the stiffness matrix for instance). As a result, the transition between both models is computationally cheap.

3.3.2 Differentiating IGA operators *w.r.t.* the Control Points

3.3.2.1 Differentiating standard IGA operators

By putting equation (3.74) and equation (3.81) into the formula of the derivative (3.15), we are at a point where we need to compute these kind of quantities:

$$\boldsymbol{u}^* \cdot \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} \qquad \text{and} \qquad \boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u}. \tag{3.83}$$

Element-wise loop The stiffness matrix and the load vector are built element-wise. The sensitivity analysis will be perform similarly, and we focus here mainly on the stiffness matrix because it is the most challenging part. Summing over each element gives:

$$\boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u} = \sum_e \left(\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}^e \right). \tag{3.84}$$

An important fact to notice is that only the control points associated to the current element e give non-zero derivatives in the term $\partial \mathbf{K}^e / \partial \boldsymbol{Q}$. Thus, for each element of the analysis model, only the few components of the gradient are updated.

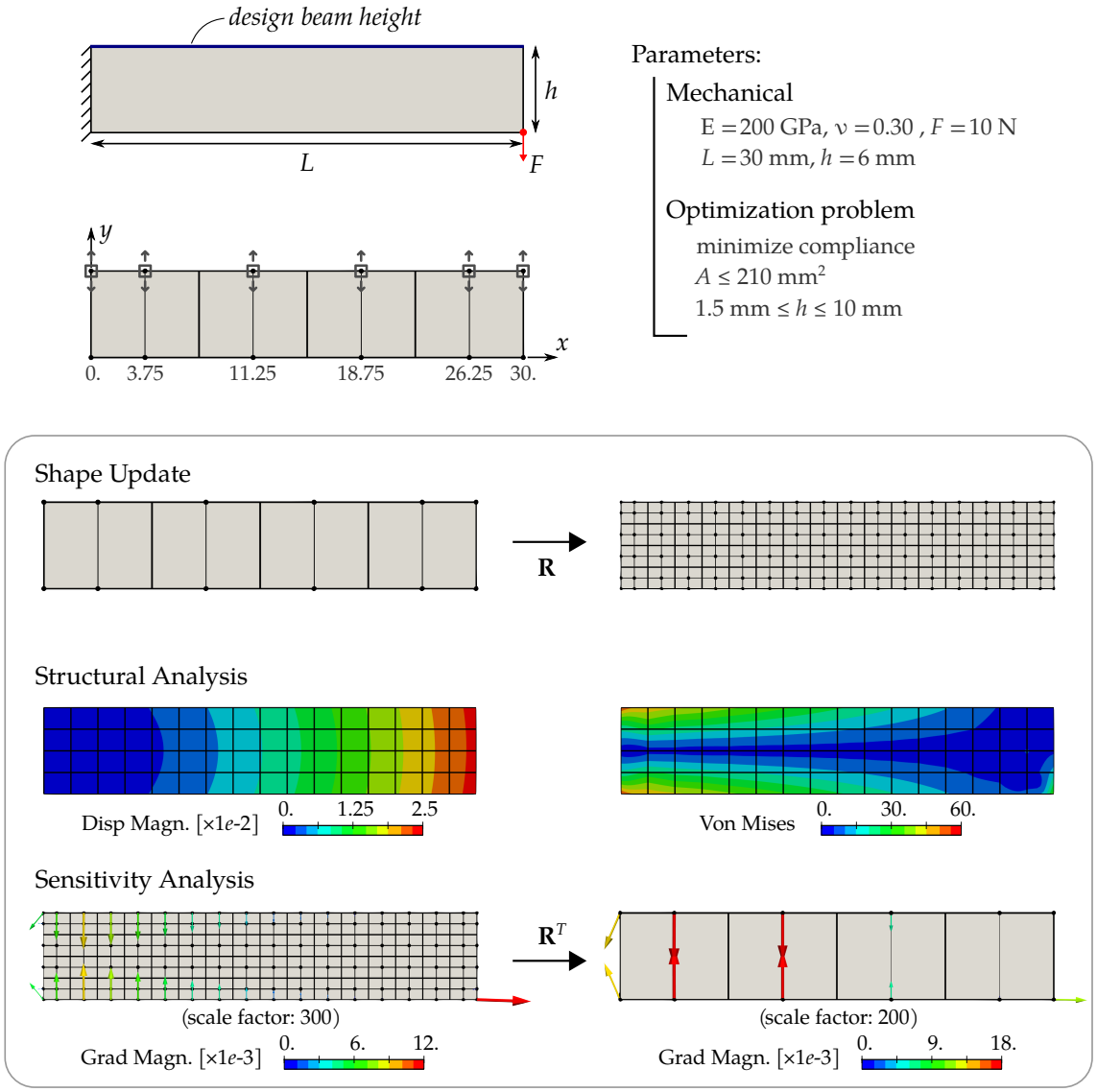


Figure 3.12 – Settings and optimization results for the cantilever beam problem.

Moreover, for each element, it is numerically not efficient to build a large matrix $\partial \mathbf{K}^e / \partial \boldsymbol{Q}$. Instead, we use the following development:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}^e = \sum_k \sum_l \boldsymbol{u}_k^{e*} \cdot \frac{\partial \mathbf{K}_{kl}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}_l^e, \tag{3.85}$$

where k and l denote two control point indices of the current element e. Finally, the stiffness matrix of the standard IGA elements (discretization of equation (3.36)), can be expressed as follows:

$$\mathbf{K}_{kl}^e = \int_{\bar{\Omega}^e} \mathbf{B}_k{}^T \mathbf{H} \mathbf{B}_l |J| \, \mathrm{d}\bar{\Omega} \tag{3.86}$$

Thus, the derivatives of the components of the elementary stiffness matrices *w.r.t.* the control points are given by:

$$\frac{\partial \mathbf{K}^e_{kl}}{\partial \boldsymbol{Q}} = \int_{\bar{\Omega}^e} \left\{ \left(\frac{\partial \mathbf{B}_k{}^T}{\partial \boldsymbol{Q}} \mathbf{H} \mathbf{B}_l + \mathbf{B}_k{}^T \frac{\partial \mathbf{H}}{\partial \boldsymbol{Q}} \mathbf{B}_l + \mathbf{B}_k{}^T \mathbf{H} \frac{\partial \mathbf{B}_l}{\partial \boldsymbol{Q}} \right) |J| \; + \; \mathbf{B}_k{}^T \mathbf{H} \mathbf{B}_l \frac{\partial |J|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega} \,. \tag{3.87}$$

We can put this last equation into equation (3.85). By commuting the double sum and the integral, we get:

$$\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}^e = \int_{\bar{\Omega}^e} \left\{ \left(\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}} : \boldsymbol{\sigma} + \boldsymbol{\varepsilon}^* : \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}} \right) |J| \; + \; (\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega} \,. \tag{3.88}$$

Three terms can be identified:

1. *Derivative of the Jacobian*

$$(\boldsymbol{\varepsilon}^* : \boldsymbol{\sigma}) \frac{\partial |J|}{\partial \boldsymbol{Q}} \qquad \text{with} \qquad \boldsymbol{\varepsilon}^* = \sum_k \mathbf{B}_k \boldsymbol{u}^*_k, \quad \boldsymbol{\sigma} = \sum_l \mathbf{H} \mathbf{B}_l \boldsymbol{u}_l, \tag{3.89}$$

2. *Derivative of the (adjoint) Strains*

$$\frac{\partial \boldsymbol{\varepsilon}^*}{\partial \boldsymbol{Q}} = \sum_k \frac{\partial \mathbf{B}_k}{\partial \boldsymbol{Q}} \boldsymbol{u}^*_k, \tag{3.90}$$

3. *Derivative of the Stresses*

$$\frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{Q}} = \sum_l \left(\frac{\partial \mathbf{H}}{\partial \boldsymbol{Q}} \mathbf{B}_l + \mathbf{H} \frac{\partial \mathbf{B}_l}{\partial \boldsymbol{Q}} \right) \boldsymbol{u}_l. \tag{3.91}$$

We finally end up with a generic expression of the analytical sensitivities. However, the partial derivatives *w.r.t.* the control points of the analysis model are achievable in the context of element formulation based on curvilinear coordinates. This is also a reason why we formulate the standard solid-shell with the use of curvilinear coordinates.

Derivative of the Jacobian In the case of the standard solid-shell formulation, the Jacobian is given by equation (3.37) that we repeat here for the sake of clarity:

$$|J| = (\boldsymbol{G}_1 \times \boldsymbol{G}_2) \cdot \boldsymbol{G}_3.$$

Differentiating the Jacobian *w.r.t.* the control points reads as:

$$\frac{\partial |J|}{\partial \boldsymbol{Q}} = (\boldsymbol{G}_2 \times \boldsymbol{G}_3) \cdot \frac{\partial \boldsymbol{G}_1}{\partial \boldsymbol{Q}} \; + \; (\boldsymbol{G}_3 \times \boldsymbol{G}_1) \cdot \frac{\partial \boldsymbol{G}_2}{\partial \boldsymbol{Q}} \; + \; (\boldsymbol{G}_1 \times \boldsymbol{G}_2) \cdot \frac{\partial \boldsymbol{G}_3}{\partial \boldsymbol{Q}}. \tag{3.92}$$

We remind that the partial derivative $\partial / \partial \boldsymbol{Q}$ contains the partial derivatives *w.r.t.* the three components of the all active control points (denoted previously Q_{ja} which correspond to the j^{th} component of the control point number a). Differentiating the covariant basis vectors $\boldsymbol{G}_i$ gives:

$$\frac{\partial \boldsymbol{G}_i}{\partial Q_{ja}} = R_{a,i} \, \boldsymbol{e}_j, \tag{3.93}$$

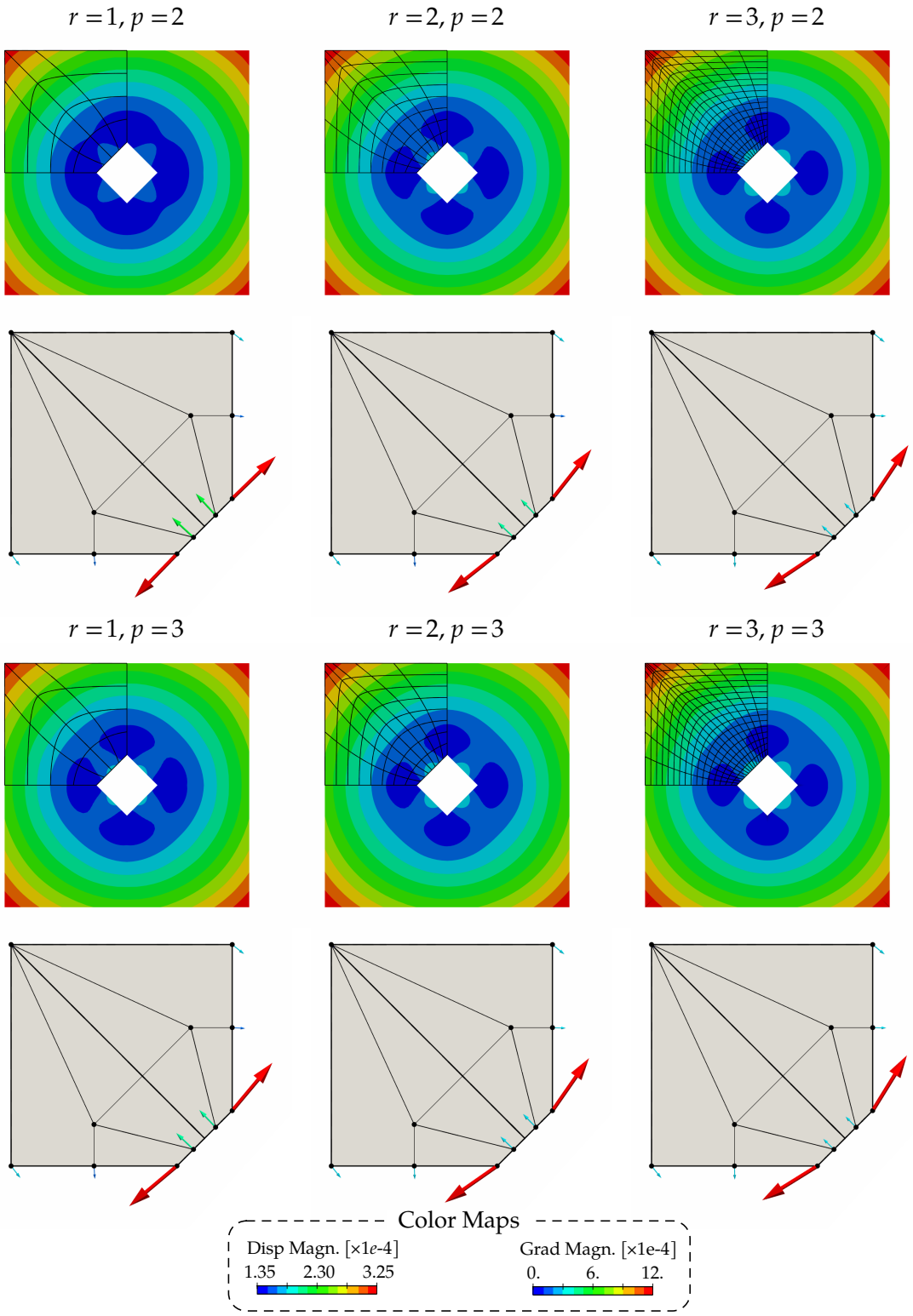


Figure 3.13 – Influence of the refinement level of the analysis model on the analytical sensitivity for the plate with a hole (case of the compliance).

and finally, the complete expression for the derivatives of the Jacobian is:

$$\frac{\partial |J|}{\partial \boldsymbol{Q}_a} = R_{a,1}\boldsymbol{G}_2 \times \boldsymbol{G}_3 \; + \; R_{a,2}\boldsymbol{G}_3 \times \boldsymbol{G}_1 \; + \; R_{a,3}\boldsymbol{G}_1 \times \boldsymbol{G}_2. \tag{3.94}$$

The derivative $\partial|J|/\partial\boldsymbol{Q}_a$ takes the form of a vector with three components. The gradient $\partial|J|/\partial\boldsymbol{Q}$ in equation (3.88) collects these derivatives where the index a corresponds to the active control points (those from the current element e).

One can follow the same approach in case of surfaces, as for instance 2D problems or shells. The Jacobian is given by equation (3.40). One should obtain the following derivatives:

$$\frac{\partial A}{\partial \boldsymbol{Q}_a} = R_{a,1}\boldsymbol{A}_2 \times \boldsymbol{A}_3 \; + \; R_{a,2}\boldsymbol{A}_3 \times \boldsymbol{A}_1. \tag{3.95}$$

We will also use these results in order to compute the analytical sensitivities when a volume or a surface area is involved in the optimization problems. From the perspective of implementing the analytical sensitivities as presented in this work, we advice to start with the case of the volume and surface area. It is a good preparation for the more complex case of the compliance for example. Let us also notice that for the volume/area, it is not necessary to perform the sensitivity analysis on the analysis model but one could use the optimization model only since the geometry is exactly captured at any level of refinement.

Derivative of the Strains In order to express the partial derivatives of the strains *w.r.t.* the control points, let us repeat their expressions accordingly to equation (3.30) for the sake of clarity:

$$\varepsilon_{ij} = \frac{1}{2}\left(\boldsymbol{u}_{,i} \cdot \boldsymbol{G}_j + \boldsymbol{G}_i \cdot \boldsymbol{u}_{,j}\right).$$

Therefore, the formal expression of the derivatives is simply given by:

$$\frac{\partial \varepsilon_{ij}}{\partial \boldsymbol{Q}} = \frac{1}{2}\left(\boldsymbol{u}_{,i} \cdot \frac{\partial \boldsymbol{G}_j}{\partial \boldsymbol{Q}} + \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{Q}} \cdot \boldsymbol{u}_{,j}\right). \tag{3.96}$$

We already know the derivative of the covariant vectors *w.r.t.* the control points, *e.g.* see equation (3.93). Finally the term in equation (3.88) with the derivative of the (adjoint) strains is not as hard as it may seem. One has to compute quantities of the following form:

$$\frac{\partial \varepsilon^*}{\partial \boldsymbol{Q}_a} : \boldsymbol{\sigma} = \sum_{ij} \frac{1}{2}\sigma^{ij}\left(\boldsymbol{u}^*{}_{,i} R_{a,j} + R_{a,i}\boldsymbol{u}^*{}_{,j}\right), \tag{3.97}$$

where the index a corresponds to the active control points. The result given by equation (3.97) should be seen as a vector with three components (derivation *w.r.t.* each of the three components of $\boldsymbol{Q}_a$).

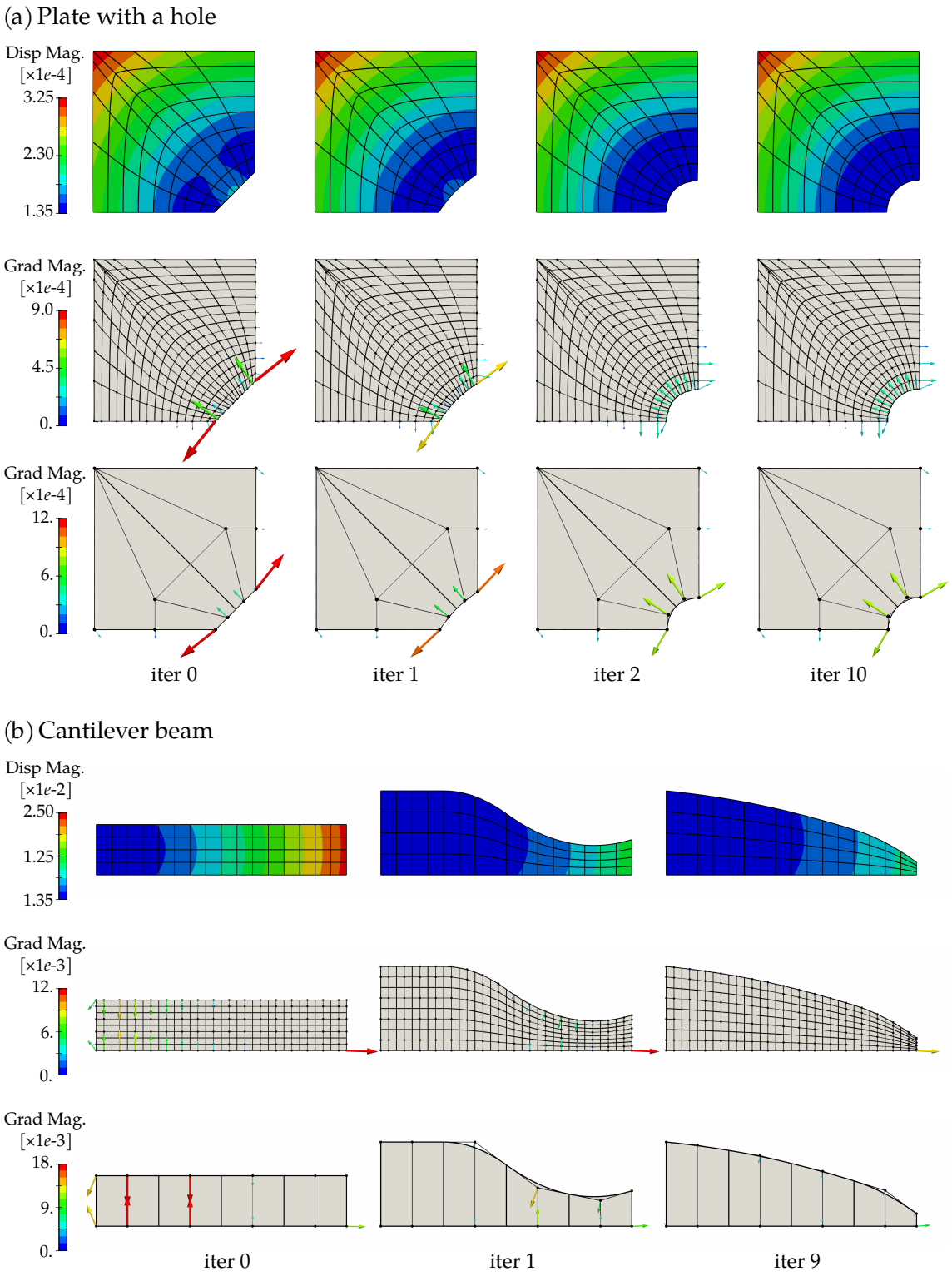


Figure 3.14 – History of structural analysis and sensitivity analysis during optimization of (a) the plate with a hole (see figure 3.10), and (b) the cantilever beam (see figure 3.12).

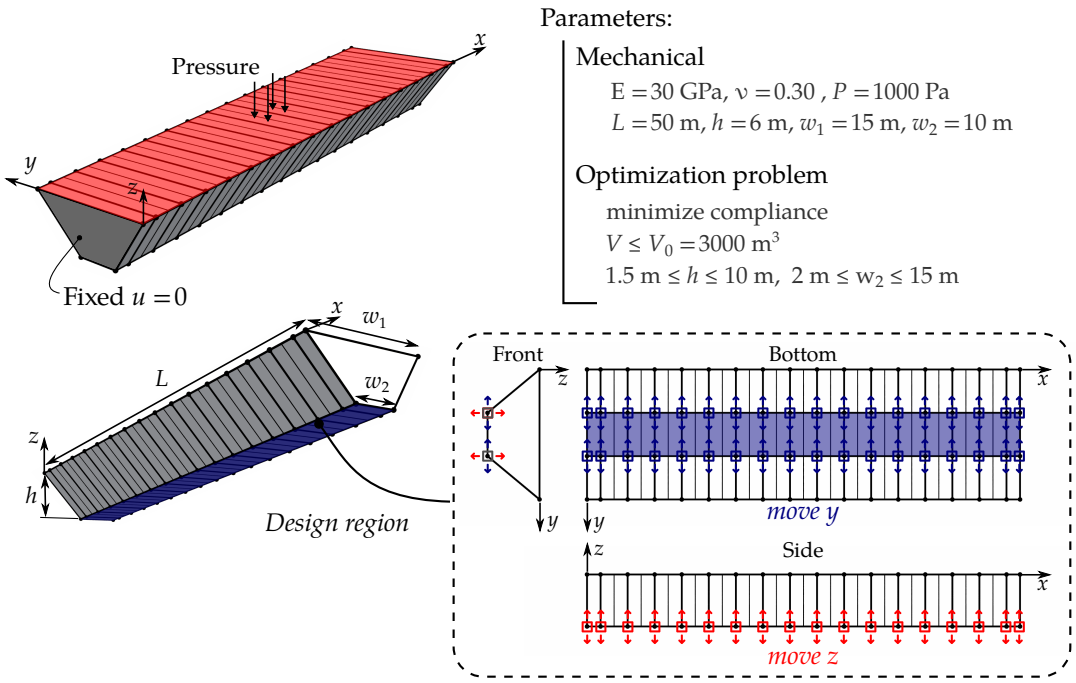


Figure 3.15 – Problem settings for the shape optimization of the 3D beam.

Derivative of the Stresses Having the derivatives of the strains in hand is already a first step to compute those of the stresses. Indeed, they are obtained by using the constitutive equations (3.32). The differentiation *w.r.t.* the control points gives:

$$\frac{\partial \sigma^{ij}}{\partial \boldsymbol{Q}} = \frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}} \varepsilon_{kl} + C^{ijkl} \frac{\partial \varepsilon_{kl}}{\partial \boldsymbol{Q}}, \tag{3.98}$$

where the derivatives of the strains $\partial \varepsilon_{kl} / \partial \boldsymbol{Q}$ is already known (see equations (3.96)-(3.97)).

The derivatives of the material tensor $\partial C^{ijkl} / \partial \boldsymbol{Q}$ are obtained by the derivation of the equation (3.33) which is repeated here for sake of clarity:

$$C^{ijkl} = \lambda G^{ij} G^{kl} + \mu \left(G^{ik} G^{jl} + G^{il} G^{jk} \right).$$

A chain rule of differentiation leads to:

$$\frac{\partial C^{ijkl}}{\partial \boldsymbol{Q}} = \lambda \left(\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} G^{kl} + G^{ij} \frac{\partial G^{kl}}{\partial \boldsymbol{Q}} \right) + \mu \left(\frac{\partial G^{ik}}{\partial \boldsymbol{Q}} G^{jl} + G^{ik} \frac{\partial G^{jl}}{\partial \boldsymbol{Q}} + \frac{\partial G^{il}}{\partial \boldsymbol{Q}} G^{jk} + G^{il} \frac{\partial G^{jk}}{\partial \boldsymbol{Q}} \right). \tag{3.99}$$

Thus, the derivatives of the contravariant metrics *w.r.t.* to the control points are involved. We have not yet computed these derivatives. To this purpose, let us remind how we get these metrics in this work (3.28):

$$\left[G^{ij} \right] = \left[G_{ij} \right]^{-1}.$$

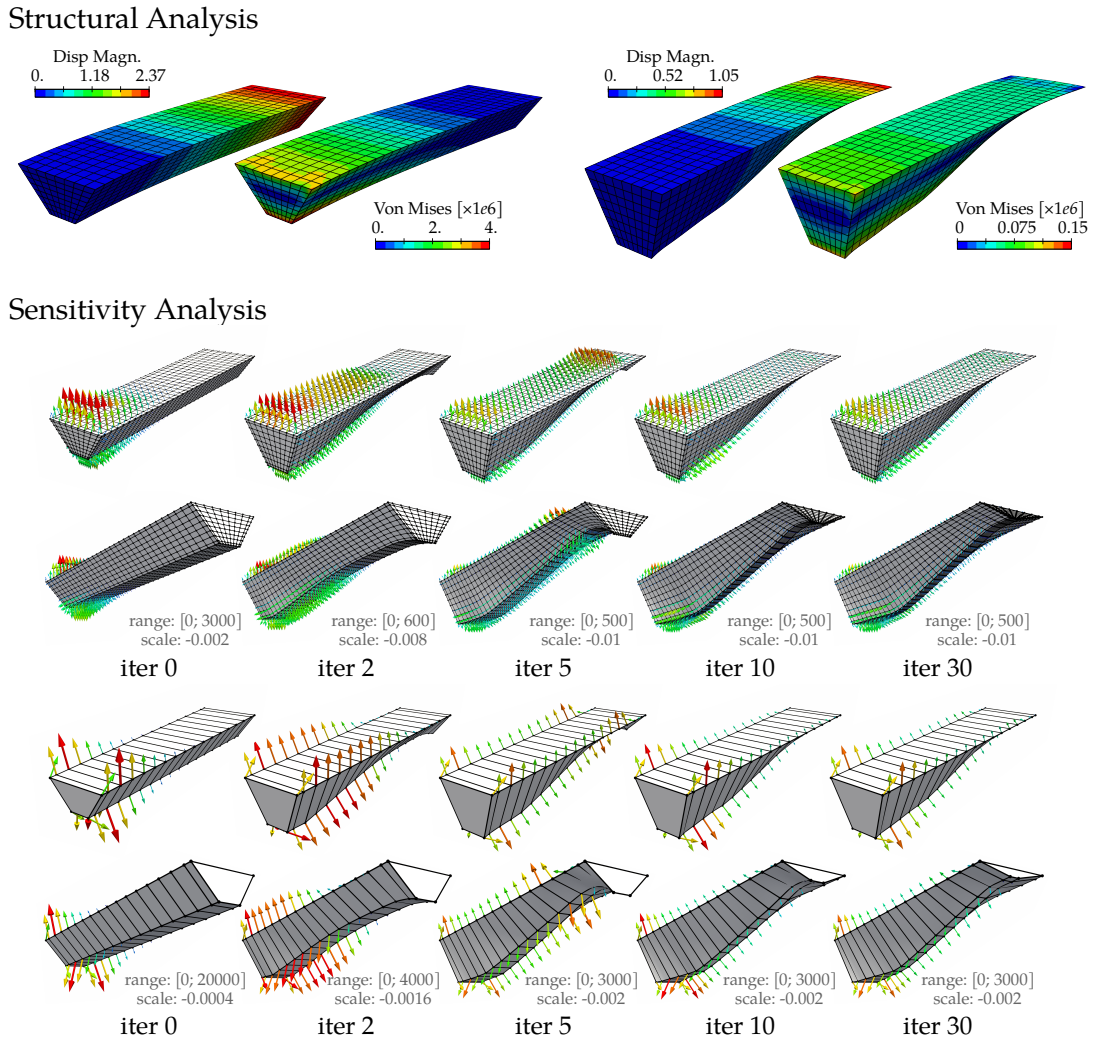


Figure 3.16 – Sensitivity analysis and optimization results for the 3D beam problem.

They are obtained by inverting a 3-by-3 matrix which takes the covariant metrics as components. One can notice that the following relation holds true:

$$\left[\, G^{ij} \,\right] \left[\, G_{ij} \,\right] = \mathbf{I}, \tag{3.100}$$

where $\mathbf{I}$ denotes here the identity matrix of size 3. Differentiating this last equation leads to:

$$\left[\frac{\partial G^{ij}}{\partial \boldsymbol{Q}}\right] \left[\, G_{ij} \,\right] + \left[\, G^{ij} \,\right] \left[\frac{\partial G_{ij}}{\partial \boldsymbol{Q}}\right] = \left[\, 0 \,\right]. \tag{3.101}$$

Then, we multiply this result by the right with the matrix containing the contravariant metrics in order to identify the derivatives of the contravariant metrics. Using equation (3.100) finally gives:

$$\left[\frac{\partial G^{ij}}{\partial \boldsymbol{Q}}\right] = -\left[\, G^{ij} \,\right] \left[\frac{\partial G_{ij}}{\partial \boldsymbol{Q}}\right] \left[\, G^{ij} \,\right]. \tag{3.102}$$

This enables to express the derivatives of the contravariant metrics as functions of the derivatives of the covariant metrics:

$$\frac{\partial G^{ij}}{\partial \boldsymbol{Q}} = -\sum_{kl} G^{ik} G^{lj} \frac{\partial G_{kl}}{\partial \boldsymbol{Q}} \,. \tag{3.103}$$

The covariant metrics are obtained by dot products between the covariant vectors as given in equation (3.27). Thus, the derivatives of the covariant metrics *w.r.t.* the control points is quite straightforward:

$$\frac{\partial G_{kl}}{\partial \boldsymbol{Q}} = \frac{\partial \boldsymbol{G}_k}{\partial \boldsymbol{Q}} \cdot \boldsymbol{G}_l + \boldsymbol{G}_k \cdot \frac{\partial \boldsymbol{G}_l}{\partial \boldsymbol{Q}}, \tag{3.104}$$

and by introducing equation (3.93) one can get the results we were interested in:

$$\frac{\partial G_{kl}}{\partial \boldsymbol{Q}_a} = R_{a,k} \boldsymbol{G}_l + R_{a,l} \boldsymbol{G}_k. \tag{3.105}$$

We now have all the ingredients to compute the derivatives of the stresses *w.r.t.* the control points of the analysis model.

Discussion of the examples We depict several examples along with the theoretical formulas in order to highlight the computation of the analytical sensitivities. The interested readers can use these three test cases to validate the implementation for example. More specifically, we present firstly two well-known 2D problems: the plate with a hole (see figures 3.10, 3.11, 3.13, and 3.14(a)) and the cantilever beam (see figures 3.12 and 3.14(b)). To show that the presented analytical sensitivities work also for general 3D problem (and not only for the special case of the solid-shell formulation), we deal with the optimal design of a thick beam subjected to bending deformation (see figures 3.15 and 3.16).

For each problem, we show the main steps involved during the optimization, namely the structural analysis performed on the finest discretization (the analysis model), and the sensitivity analysis. As explained, the sensitivity analysis is firstly calculated using the analysis model, and then, it is transferred on the coarsest discretization (the optimization model). For instance, one can observe this multilevel sensitivity computation in figure 3.11 for plate with a hole and in figure 3.12 for the cantilever beam. All the gradients presented in the mentioned figures represent the analytical sensitivities of the compliance (3.19). Moreover, we omit the derivation of the load vector because it does not come into play for these examples. Thus, the gradients that are displayed on the analytical models correspond to:

$$\frac{d f_{\text{cpl}}}{d \boldsymbol{Q}} = -\frac{1}{2} \boldsymbol{u}^T \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u}, \tag{3.106}$$

and are computed accordingly to equation (3.88). Thereafter, the gradients that are displayed on the optimization models are obtained using equation (3.81).

Type	df_{cpl}/dx_1	df_{cpl}/dx_2	df_{cpl}/dx_3	df_{cpl}/dx_4	df_{cpl}/dx_5	df_{cpl}/dx_6
FD (1e-4)	-8.91674872	-2.90020134	-2.75637409	2.75637409	2.90020134	8.91674872
FD (1e-6)	-8.91674878	-2.90020142	-2.75637401	2.75637402	2.90020121	8.91674871
sAN (1e-6)	-8.91675602	-2.90020546	-2.75638323	2.75636483	2.90019719	8.91674136
sAN (1e-8)	-8.91675076	-2.90020354	-2.75637295	2.75636794	2.90019289	8.91674717
AN (-)	-8.91674869	-2.90020130	-2.75637405	2.75637405	2.90020130	8.91674869

Table 3.1 – Different types of gradients for the plate with a square hole as given in figure 3.11.

Going into more detail, figure 3.13 show the influence of the refinement level of the analysis model. The refinement influences not only the solution (displacement field) but also the sensitivities. Thus, the refinement level of the analysis model has to be chosen so that both the objective function and the gradient are precisely computed. It is also possible to compare the analytical sensitivities with approximated sensitivities (semi-AN, or global FD). We give in table 3.1 the gradients for the case of the plate with a hole as described in figure 3.10 and 3.11. Here, we use central Finite Differences for the FD approximation, and forward semi-Analytical scheme for the sAN approximation. One can observe the impact of the step length. Up to a certain precision, the result differs. This is especially true in case of the semi-AN calculation. Using a central difference during the approximation helps but it is unfortunately more expensive. In practice one may compromise between precision and numerical cost. For this simple case of the plate with a hole, the influence of the step length remains limited. However, for more complex cases, it can become difficult to set this parameter. For instance, the design variables can have a fully different level of influence on the solution and may require different step length. Even if every quantities are scaled (range of the design variables between 0 and 1, normalization of the objective, etc.), it does not ensure that the influence of the step length vanishes. Comparing the analytical sensitivities with the central FD results confirms the correctness of the presented formulation (and implementation).

Lastly, let us mention that the numerical cost of the analytical sensitivities is lower than the other types of gradients. It is especially true for larger problem as the thick beam (figure 3.16). Another interesting point is that the computation cost is independent of the number of design variables, since the main step is the differentiation of the stiffness matrix *w.r.t.* to the control points of the analysis model. This is done once and for all for each partial derivative *w.r.t.* to the design variables. Based on our observation, the presented analytical sensitivity analysis is numerically cheaper in addition to being more precise.

3.3.2.2 Differentiating Kirchhoff–Love operators

For the Kirchhoff–Love shell formulation, we follow the same logic as the standard IGA solid-shell formulation we just deal with (see section 3.3.2.1). Thus, here we skip re-

dundant calculation steps. Especially, one can obtain the counterpart of equation (3.88) for the Kirchhoff–Love shell formulation by applying the same reasoning. In that respect, we can show that:

$$\begin{aligned}\boldsymbol{u}^{e*} \cdot \frac{\partial \mathbf{K}^e}{\partial \boldsymbol{Q}} \boldsymbol{u}^e = &\int_{\bar{\Omega}_0^e} \left\{ \left(\frac{\partial \boldsymbol{e}^*}{\partial \boldsymbol{Q}} : \boldsymbol{n} + \boldsymbol{e}^* : \frac{\partial \boldsymbol{n}}{\partial \boldsymbol{Q}} \right) A \; + \; (\boldsymbol{e}^* : \boldsymbol{n}) \frac{\partial A}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega}_0 \\ &+ \int_{\bar{\Omega}_0^e} \left\{ \left(\frac{\partial \boldsymbol{\kappa}^*}{\partial \boldsymbol{Q}} : \boldsymbol{m} + \boldsymbol{\kappa}^* : \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{Q}} \right) A \; + \; (\boldsymbol{\kappa}^* : \boldsymbol{m}) \frac{\partial A}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega}_0 \,.\end{aligned} \tag{3.107}$$

We have two principal terms identified as a membrane and a bending contribution. As for the solid-shell, we need to compute the derivatives of the Jacobian, the derivatives of the (adjoint) strains (membrane and bending), and the derivatives of the stress resultants (membrane and bending).

Derivative of the Jacobian We already introduced the expression of the derivatives of the Jacobian *w.r.t.* the control points, see equation (3.95). We simply repeat the expression here:

$$\frac{\partial A}{\partial \boldsymbol{Q}_a} = R_{a,1} \boldsymbol{A}_2 \times \boldsymbol{A}_3 \; + \; R_{a,2} \boldsymbol{A}_3 \times \boldsymbol{A}_1 .$$

Derivative of the Membrane Strains and Stresses The expressions of the membrane strains $\boldsymbol{e}_{\alpha\beta}$ and the membrane forces $\boldsymbol{n}^{\alpha\beta}$ involved in the Kirchhoff–Love shell are essentially similar to the strain and stress fields of the standard solid elements (2D problem). Thus, we give here only the final results. One can recover the following equations by going through what has been presented for the solid-shell formulation. For the derivatives of the membrane strains, we obtain:

$$\frac{\partial \boldsymbol{e}^*_{\alpha\beta}}{\partial \boldsymbol{Q}_a} = \frac{1}{2} \boldsymbol{u}^*{}_{,\alpha} R_{a,\beta} + R_{a,\alpha} \boldsymbol{u}^*{}_{,\beta} . \tag{3.108}$$

The derivatives of the membrane forces is obtained using the constitutive equation (3.60). We have:

$$\frac{\partial \boldsymbol{n}^{\alpha\beta}}{\partial \boldsymbol{Q}} = t \, \frac{\partial \bar{C}^{\alpha\beta\gamma\delta}}{\partial \boldsymbol{Q}} \boldsymbol{e}_{\gamma\delta} \; + \; t \, \bar{C}^{\alpha\beta\gamma\delta} \frac{\partial \boldsymbol{e}_{\gamma\delta}}{\partial \boldsymbol{Q}} . \tag{3.109}$$

The derivatives of the material tensor $\partial \bar{C}^{\alpha\beta\gamma\delta} / \partial \boldsymbol{Q}$ can be computed similarly to what has been done for the 3D constitutive law (3.99).

Since the membrane part involved in the analytical sensitivities for the Kirchhoff–Love formulation is very similar to classical 2D problems (*e.g.* by tacking thickness equals to one and assuming plane-stress state), the interested readers can start by implementing the analytical sensitivities in that context. One can refer to the simple benchmark problems already presented (see section 3.3.2.1).

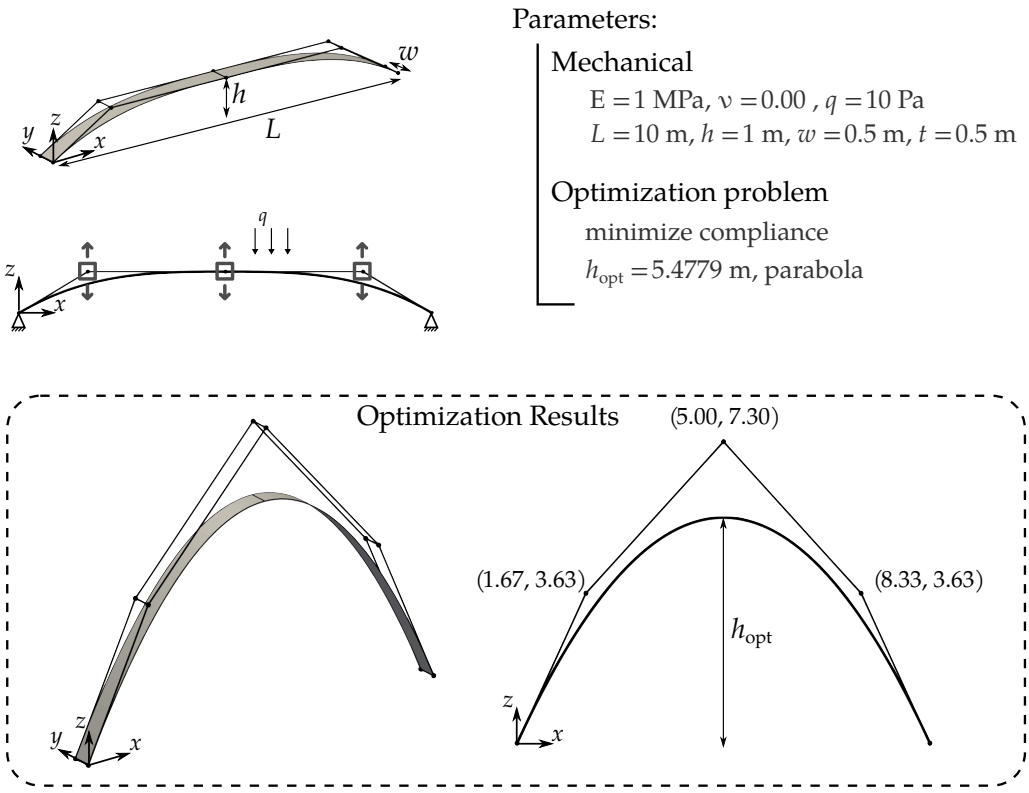


Figure 3.17 – Settings for the shape optimization of an arch under constant load. The optimal result is a parabola with a specific height to length ratio [72].

Derivative of the Bending Strains and Stresses The bending part involved in the sensitivity (3.107) requires additional developments. However, the core idea remains the same. A chain rule is applied until we get an expression with quantities that we know how to derive *w.r.t.* the control points. Let us repeat the expression of the bending strain (3.53):

$$\boldsymbol{\kappa}_{\alpha\beta} = -\boldsymbol{u}_{,\alpha\beta} \cdot \boldsymbol{A}_3 + \frac{1}{A}\Big(\boldsymbol{u}_{,1} \cdot (\boldsymbol{A}_{\alpha,\beta} \times \boldsymbol{A}_2) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{A}_1 \times \boldsymbol{A}_{\alpha,\beta})\Big) + \frac{\boldsymbol{A}_3 \cdot \boldsymbol{A}_{\alpha,\beta}}{A}\Big(\boldsymbol{u}_{,1} \cdot (\boldsymbol{A}_2 \times \boldsymbol{A}_3) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{A}_3 \times \boldsymbol{A}_1)\Big),$$

which is split into three terms as follows in order to describe the derivatives:

$$\boldsymbol{\kappa}_{\alpha\beta} = -\boldsymbol{u}_{,\alpha\beta} \cdot \boldsymbol{A}_3 \; + \; \boldsymbol{\kappa}^1_{\alpha\beta} \; + \; \boldsymbol{\kappa}^2_{\alpha\beta}. \tag{3.110}$$

Thus, the derivatives of the bending strains *w.r.t.* the control points are given by:

$$\frac{\partial \boldsymbol{\kappa}_{\alpha\beta}}{\partial \boldsymbol{Q}} = -\boldsymbol{u}_{,\alpha\beta} \cdot \frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}} \; + \; \frac{\partial \boldsymbol{\kappa}^1_{\alpha\beta}}{\partial \boldsymbol{Q}} \; + \; \frac{\partial \boldsymbol{\kappa}^2_{\alpha\beta}}{\partial \boldsymbol{Q}}. \tag{3.111}$$

The derivatives $\partial \boldsymbol{\kappa}^1_{\alpha\beta}/\partial \boldsymbol{Q}$ can be written as follows:

$$\begin{aligned}\frac{\partial \boldsymbol{\kappa}^1_{\alpha\beta}}{\partial \boldsymbol{Q}} &= \frac{\partial A^{-1}}{\partial \boldsymbol{Q}}\Big(\boldsymbol{u}_{,1}\cdot(\boldsymbol{A}_{\alpha,\beta}\times\boldsymbol{A}_2)+\boldsymbol{u}_{,2}\cdot(\boldsymbol{A}_1\times\boldsymbol{A}_{\alpha,\beta})\Big)\\ &+\frac{1}{A}\Big((\boldsymbol{u}_{,1}\times\boldsymbol{A}_{\alpha,\beta})\cdot\frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}}-(\boldsymbol{u}_{,1}\times\boldsymbol{A}_2)\cdot\frac{\partial \boldsymbol{A}_{\alpha,\beta}}{\partial \boldsymbol{Q}}\Big)\\ &+\frac{1}{A}\Big((\boldsymbol{A}_{\alpha,\beta}\times\boldsymbol{u}_{,2})\cdot\frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}}-(\boldsymbol{A}_1\times\boldsymbol{u}_{,2})\cdot\frac{\partial \boldsymbol{A}_{\alpha,\beta}}{\partial \boldsymbol{Q}}\Big),\end{aligned} \tag{3.112}$$

where some circular shifts have been performed in the scalar triple products. Identically, the other term $\partial \boldsymbol{\kappa}^2_{\alpha\beta}/\partial \boldsymbol{Q}$ is given by:

$$\begin{aligned}\frac{\partial \boldsymbol{\kappa}^2_{\alpha\beta}}{\partial \boldsymbol{Q}} &= \frac{\partial \tilde{A}_{\alpha\beta}}{\partial \boldsymbol{Q}}\Big(\boldsymbol{u}_{,1}\cdot(\boldsymbol{A}_2\times\boldsymbol{A}_3)+\boldsymbol{u}_{,2}\cdot(\boldsymbol{A}_3\times\boldsymbol{A}_1)\Big)\\ &+\frac{\boldsymbol{A}_3\cdot\boldsymbol{A}_{\alpha,\beta}}{A}\Big((\boldsymbol{A}_3\times\boldsymbol{u}_{,1})\cdot\frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}}+(\boldsymbol{u}_{,1}\times\boldsymbol{A}_2)\cdot\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}}\Big)\\ &+\frac{\boldsymbol{A}_3\cdot\boldsymbol{A}_{\alpha,\beta}}{A}\Big((\boldsymbol{u}_{,2}\times\boldsymbol{A}_3)\cdot\frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}}+(\boldsymbol{A}_1\times\boldsymbol{u}_{,2})\cdot\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}}\Big),\end{aligned} \tag{3.113}$$

with:

$$\frac{\partial \tilde{A}_{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{\partial A^{-1}}{\partial \boldsymbol{Q}}\boldsymbol{A}_3\cdot\boldsymbol{A}_{\alpha,\beta}+\frac{1}{A}\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}}\cdot\boldsymbol{A}_{\alpha,\beta}+\frac{1}{A}\boldsymbol{A}_3\cdot\frac{\partial \boldsymbol{A}_{\alpha,\beta}}{\partial \boldsymbol{Q}}. \tag{3.114}$$

We already know how are expressed the derivatives of the covariant basis vectors $\partial \boldsymbol{A}_\alpha/\partial \boldsymbol{Q}$ (see equation (3.93)) and the derivatives of the Jacobian $\partial A/\partial \boldsymbol{Q}$ (see equation (3.95)). Nonetheless, there are some additional derivatives that are involved in the differentiation of the bending strains. Regarding the inverse of the Jacobian, the derivatives read as:

$$\frac{\partial A^{-1}}{\partial \boldsymbol{Q}} = -\frac{1}{A^2}\frac{\partial A}{\partial \boldsymbol{Q}} \tag{3.115}$$

The derivatives of the director vector appears multiple times. After few developments, one should obtain the following formula:

$$\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}} = -\frac{1}{A}\boldsymbol{A}_3\frac{\partial A}{\partial \boldsymbol{Q}}+\frac{1}{A}\left(\frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}}\times\boldsymbol{A}_2\right)+\frac{1}{A}\left(\boldsymbol{A}_1\times\frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}}\right). \tag{3.116}$$

Scalar products between these derivatives $\partial \boldsymbol{A}_3/\partial \boldsymbol{Q}$ and different vectors are involved. Let us give a general expression of this type of quantities:

$$\mathbf{v}\cdot\frac{\partial \boldsymbol{A}_3}{\partial \boldsymbol{Q}_a} = -\frac{1}{A}\left((\mathbf{v}\cdot\boldsymbol{A}_3)\frac{\partial A}{\partial \boldsymbol{Q}_a}+(\mathbf{v}\times\boldsymbol{A}_2)R_{a,1}+(\boldsymbol{A}_1\times\mathbf{v})R_{a,2}\right), \tag{3.117}$$

where $\mathbf{v}$ denotes any required vector. Lastly, let us give the following results:

$$\mathbf{v}\cdot\frac{\partial \boldsymbol{A}_{\alpha,\beta}}{\partial \boldsymbol{Q}_a} = R_{a,\alpha\beta}\,\mathbf{v}. \tag{3.118}$$

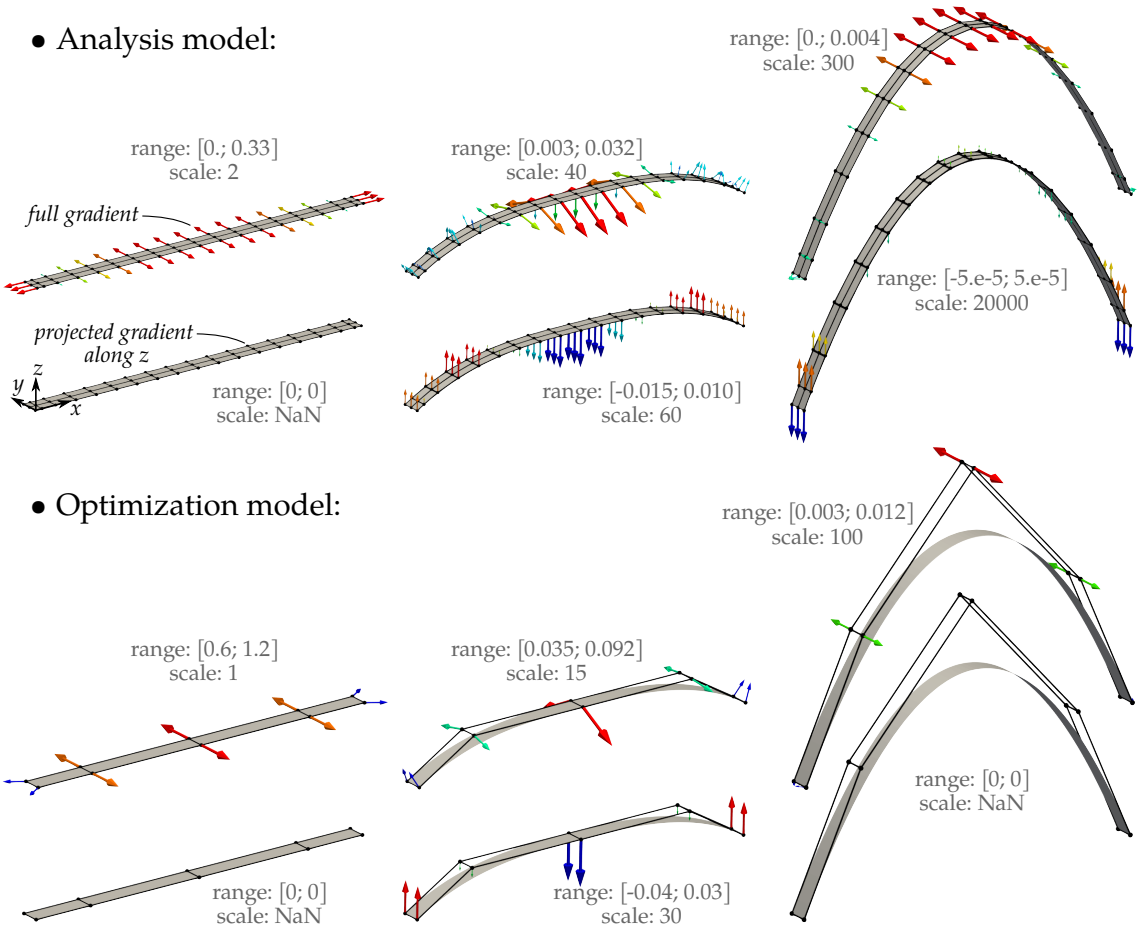


Figure 3.18 – Sensitivity analysis for the arch. The planar beam (left) leads to a zero gradient and thus cannot be taken as the initial configuration. In the optimal configuration (right), the z-component of the gradient is also null which reflects that a minimum has been found.

We now have all the ingredients in order to compute the derivatives of the bending strains *w.r.t.* the control points of the analysis model. It contains quite a lot of terms. Hence, it is worth spending some time to identify repetitive terms in order to make numerical savings in the implementation. For instance, there are multiple cross and dot products that seem better to compute once and for all at the beginning instead of computing them for each control points $\boldsymbol{Q}_a$.

Finally, the derivatives of the bending moments are computed through the derivation of the constitutive equation (3.61):

$$\frac{\partial \boldsymbol{m}^{\alpha\beta}}{\partial \boldsymbol{Q}} = \frac{t^3}{12}\frac{\partial \bar{C}^{\alpha\beta\gamma\delta}}{\partial \boldsymbol{Q}}\boldsymbol{\kappa}_{\gamma\delta} + \frac{t^3}{12}\bar{C}^{\alpha\beta\gamma\delta}\frac{\partial \boldsymbol{\kappa}_{\gamma\delta}}{\partial \boldsymbol{Q}}. \tag{3.119}$$

At this point, we now how to compute every terms, *i.e.* the derivatives of the material tensor and the derivatives of the bending strains.

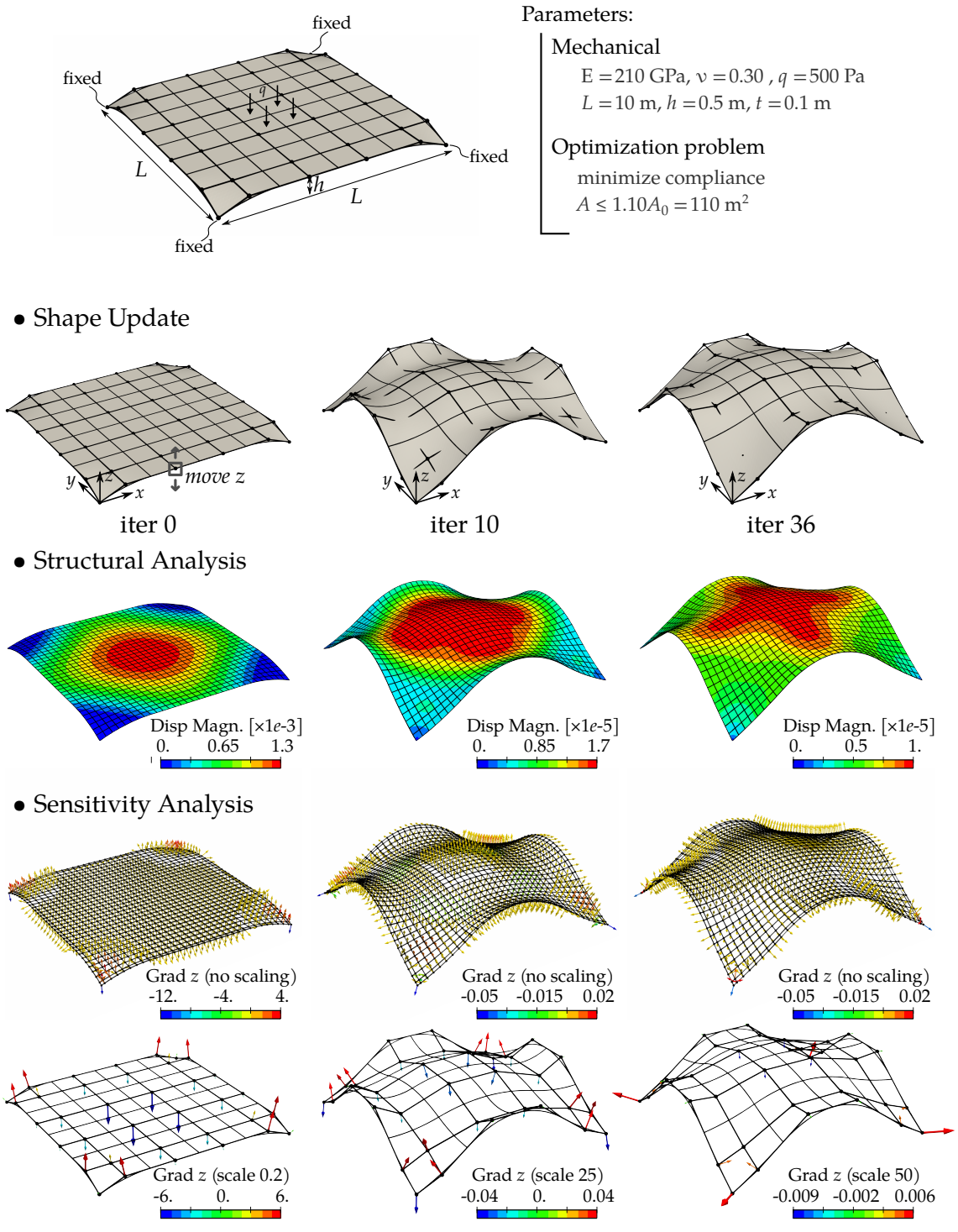


Figure 3.19 – Settings and optimization results for the square roof problem.

Type	df_{cpl}/dx_1	df_{cpl}/dx_2	df_{cpl}/dx_3
FD (1e-6)	-1.6957568	-8.4787011	-1.6957568
central FD (1e-6)	-1.6959047	-8.4787644	-1.6959047
sAN (1e-6)	-1.6959225	-8.4788029	-1.6959224
central sAN (1e-6)	-1.6959047	-8.4787644	-1.6959046
full AN (-)	-1.6959046	-8.4787644	-1.6959046

Table 3.2 – Different types of gradients for the arch as given in figure 3.18 ($\times 1e$-2).

Discussion of the examples As for the standard IGA element formulation, we give several examples in order to demonstrate the good performance of the analytical sensitivities for the Kirchhoff–Love shell. We deal with two examples: the arch problem as depicted in figures 3.17 and 3.18, and the square roof as given in figure 3.19. For each example, the external load is formulated so that it is constant for each shape configuration. The external virtual work is given by:

$$\delta W_{\text{ext}} = \int_{\bar{\Omega}_0} q\boldsymbol{z} \cdot \delta\boldsymbol{u}\, A_z \, \mathrm{d}\bar{\Omega}_0 \quad \text{with} \quad A_z = (\boldsymbol{A}_1 \times \boldsymbol{A}_2) \cdot \boldsymbol{z}, \tag{3.120}$$

where $\boldsymbol{z}$ denotes the director vector of the global coordinate basis as shown on figure 3.17 for the arch and figure 3.19 for the roof. Even if the load vector is constant when modifying the shape in direction z, we decide to include its derivative during the sensitivity analysis in order to provide a general framework. Indeed, if the shape parametrization allows a shape modification in the other directions, the derivative of the load vector should be taken into account. It means that the gradients represented along with the fine parametrization in figures 3.18 and 3.19 are computed by:

$$\frac{df_{\text{cpl}}}{d\boldsymbol{Q}} = \boldsymbol{u}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} - \frac{1}{2}\boldsymbol{u}^T \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}}\boldsymbol{u}, \tag{3.121}$$

where the derivative of the stiffness matrix is done according to equation (3.107). The derivative of the load vector is calculated as follow:

$$\boldsymbol{u}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} = \int_{\bar{\Omega}_0} q\boldsymbol{z} \cdot \delta\boldsymbol{u}\, \frac{\partial A_z}{\partial \boldsymbol{Q}} \, \mathrm{d}\bar{\Omega}_0, \tag{3.122}$$

where:

$$\frac{\partial A_z}{\partial \boldsymbol{Q}_a} = (\boldsymbol{A}_2 \times \boldsymbol{z})R_{a,1} + (\boldsymbol{z} \times \boldsymbol{A}_1)R_{a,2}. \tag{3.123}$$

It is straightforward to see that this term does not add components in z-direction. Finally, the gradients represented over the coarse parametrizations in figures 3.18 and 3.19 are computed using equation (3.81).

In order to validate the presented analytical sensitivity for the Kirchhoff–Love shell formulation, we compare the different types of gradient for the initial (curved) arch as

set in figure 3.17. The results are given in table 3.2. The optimization model is built with 2 elements with degree 3 in the main direction and degree 1 in the other. The analysis model is then obtained by a level of refinement of 3 in the main direction (16 elements). We also elevate the degree such that degree 2 is defined in the second direction. In table 3.2 we present the semi-Analytical and global Finite Differences approximation using both forward and central schemes. The results obtain with the central schemes show the correctness of the analytical sensitivities. For both semi-AN and global FD, the forward scheme induces inaccuracy up to a certain precision.

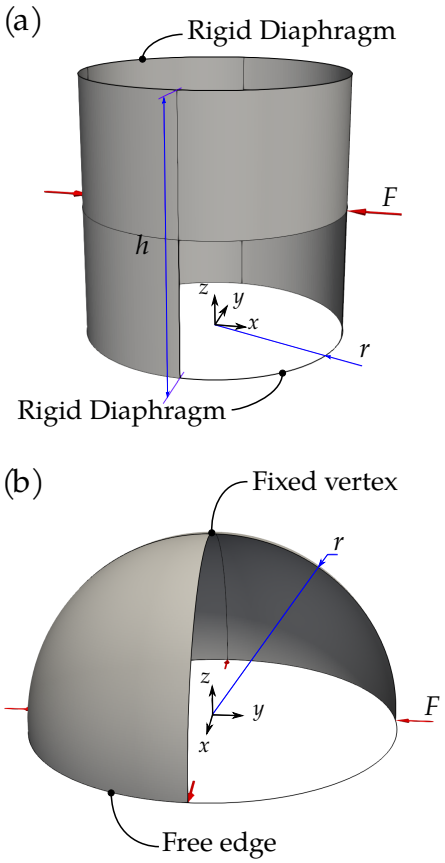
Concerning the numerical cost, we conclude the same as for the standard IGA formulation. The computational cost of the presented analytical sensitivities is independent of the number of design variables and is cheaper than the other approximate sensitivities, especially for complex models.

3.4 Optimal Design of Shell Structures

Most results presented in this section have been published in Hirschler et al. [18]. This study has been done prior to the development of the analytical sensitivities. Semi-analytical sensitivities with a forward Finite Difference scheme were used instead. However, this issue is transparent here. The goal is to explore the use of solid-shell elements in the framework of isogeometric shape optimization of shells. We first compare the solid-shell strategy with the commonly adopted strategy based on Kirchhoff–Love shell. Then we take advantage of the geometric representation of the thickness brought by solid-shell elements. It offers the interesting possibility of imposing continuous sizing variations.

3.4.1 Solid-shell vs. Kirchhoff–Love

In order to investigate the use of isogeometric solid-shell elements for shape optimization of shell structures, two optimization examples depicted in figure 3.20 are first studied. Both the pinched cylinder and the pinched hemisphere are known from the shell obstacle course [130]. These optimization problems are non-convex and are numerically challenging. In fact, the pinched hemisphere and the pinched cylinder are commonly studied to test high-efficient shell elements. The hemisphere is challenging in terms of locking: it exhibits almost no membrane strains. The pinched cylinder is even more severe. The loading and the displacement boundary conditions lead to a highly localized strain state. The local loading associated with the curved geometry make the problem challenging numerically in terms of both locking and representation of complex membrane states. Optimizing this set of problems automatically requires accurate analysis results.



	Cylinder (a)	**Hemisphere** (b)
Height h	600	-
Radius r	300	10
Thickness t	3.0	0.04
Modulus E	3.0×10^6	6.825×10^7
Ratio ν	0.30	0.30
Loading	Local loads $F = 1$	Local loads $F = 2$

Figure 3.20 – Numerical examples to investigate the use of isogeometric solid-shell elements for shape optimization of shell structures: (a) pinched cylinder and (b) pinched hemisphere.

Pinched cylinder We first deal with the test case of the pinched cylinder (see figure 3.20(a)). The optimization model is chosen relatively coarse in order to keep the shape simple. Because of the complexity of the strain state generated by the local loading, a fine optimization model would lead to a too complex final shape. Due to the symmetry of the problem, one eighth of the cylinder is considered, hence only one NURBS patch is needed. The symmetric conditions are imposed through static condensation. The optimization model is discretized by 3×4 quadratic elements in the circumferential and the axial direction respectively. With such an optimization model in hand, 6 design variables update the geometry as depicted in figure 3.21(a). Variables named x_i move some control points in the x-direction, whereas variables denoted r_i move some control points in the radial direction. At the patch boundaries, groups of two or four control points are set in order to preserve a smooth G^1-continuous shape. All variables are constrained by bounds of $[0, 60]$. Analysis is performed with cubic elements. Refinement level of 4 and 3 are set in the circumferential and the axial direction respectively, leading to an analysis model of 1536 elements.

We run the shape optimization with the Kirchhoff–Love shell and with the standard

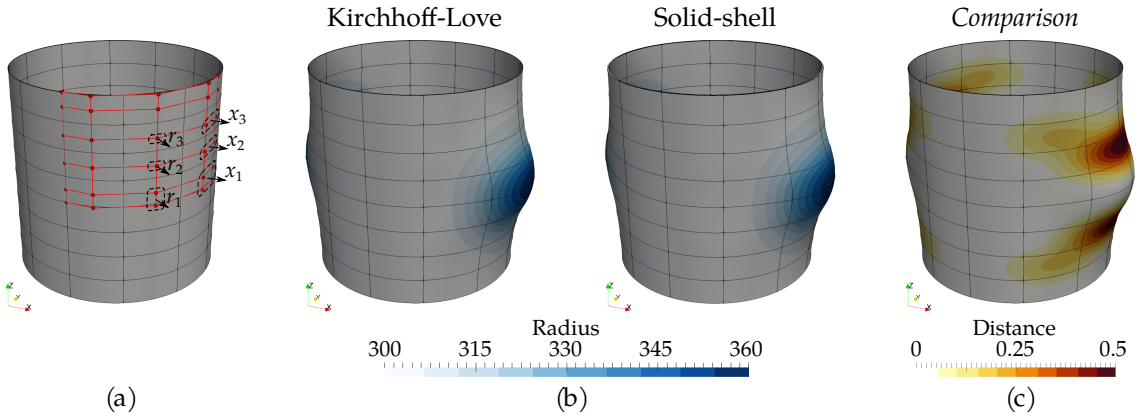


Figure 3.21 – Shape optimization of the pinched cylinder: (a) 6 design variables update the optimization model, (b) final shapes obtained by both shell formulations, and (c) comparison of the final shapes depicted as the distance between their mid-surface.

	x_1	x_2	x_3	r_1	r_2	r_3
Kirchhoff–Love	60.0	22.65	7.52	0.0	1.74	0.23
solid-shell	60.0	23.31	7.58	0.0	1.72	0.51
Relative Gap [%]	0.0	1.10	0.10	0.0	0.02	0.47

Table 3.3 – Optimal values of the design variables for the shape optimization of the pinched cylinder.

solid-shell. Final shapes are identical as shown in figure 3.21. The difference between both results is not visible when looking at the design modifications. To further compare the results, the distance between the mid-surfaces is presented. The maximum gap between the optimal surface obtained with Kirchhoff–Love elements and the final mid-surface obtained with solid-shell elements is about 0.5 which is, by comparison with the range of allowed design variation, very small. The maximum gap normalized by 60 is lower than 1%. Table 3.3 gives the optimal values of the design variables for both shell formulations. Results are similar and the relative gap (defined as $|s^{\mathrm{KL}} - s^{\mathrm{Solid}}|/60$) for each variable does not exceed 1%.

The small difference in final design when using either Kirchhoff–Love elements or solid-shell elements may be a result of the difference in converged solutions. It has be shown that the converged solution using isogeometric solid-shell elements for the pinched cylinder problem is a little softer than the reference solution [17, 128]. This little difference in solution may influence the shape updates and finally lead to a slightly different optimal design. Beyond that, the shape optimization of the pinched cylinder reveals very similar behavior for both shell formulations. During the optimization process, the shape updates are similar and it has been observed that the convergence is reached after the same amount of iterations.

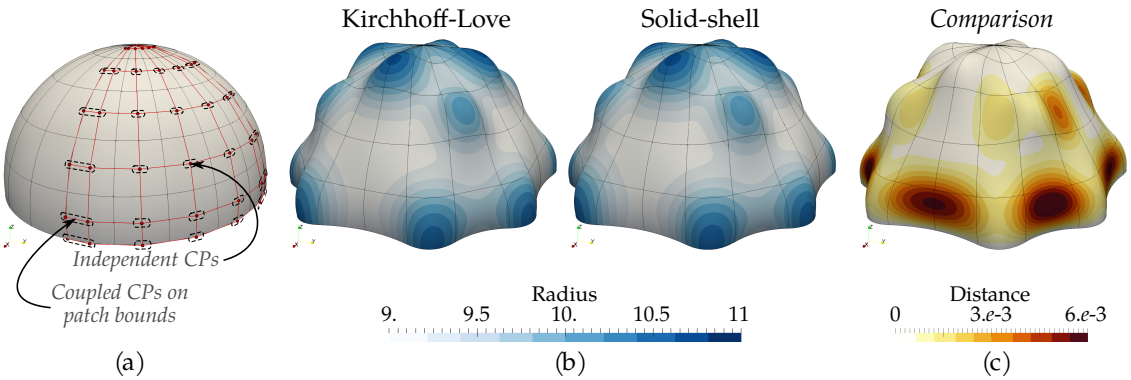


Figure 3.22 – Shape optimization of the pinched hemisphere: (a) 25 design variables update the optimization model, (b) final shapes obtained by both shell formulations, and (c) comparison of the final shapes depicted as the distance between their mid-surface.

Pinched hemisphere Similarity with the second optimization example of the pinched hemisphere can be found in Kiendl et al. [72]. Again, the symmetry of the problem allows to consider only one quarter of the structure. Hence, only one patch is needed. The optimization model is built with uniform knot vectors which discretize the geometry in 5 quadratic elements per parametric direction. 25 design variables depicted in figure 3.22(a) are defined in order to prescribe the shape modifications in radial direction. Once again, the boundary control points are coupled in order the preserve a smooth G^1-continuous shape. All variables are constrained by bounds of $[-1, 1]$. Further k-refinement is applied to perform the analysis, leading to an analysis model with 400 cubic elements. Tests showed that this refinement level was adequate to converge to the good looking optimal design.

Once again, we run the optimization with the solid-shell and compare the results with those provided by the Kirchhoff–Love shell. Figure 3.22(b) presents the optimal design obtained by both formulations. Despite the complexity of this optimization problem, the final shapes are very similar: no visible difference can be observed. Similar hollows and bumps are located on both optimal designs. To compare more in detail the results, the distance between the two mid-surfaces is computed (see figure 3.22(c)). The maximum gap in shape is about $6.34 \cdot 10^{-3}$ which is very small in comparison with the allowed range of design modification. Normalized by 2, the maximum relative distance between the optimal mid-surfaces is lower than 0.5%. Final values of the cost function also demonstrate the similarity between the optimal design obtained with the standard solid-shell and the one obtained with the Kirchhoff–Love shell. The final compliance is $1.34 \cdot 10^{-3}$ for the solid-shell case and $1.30 \cdot 10^{-3}$ for the Kirchhoff–Love case. However, the final shape obtained by Kirchhoff–Love shell is not better. Note that in order to correctly compare the final shapes, the compliance has to be evaluated on the same analysis model [72]. Strain energy is evaluated for the optimized shapes in a post-processing step using a fine mesh with quartic Kirchhoff–Love NURBS elements. This post-processing

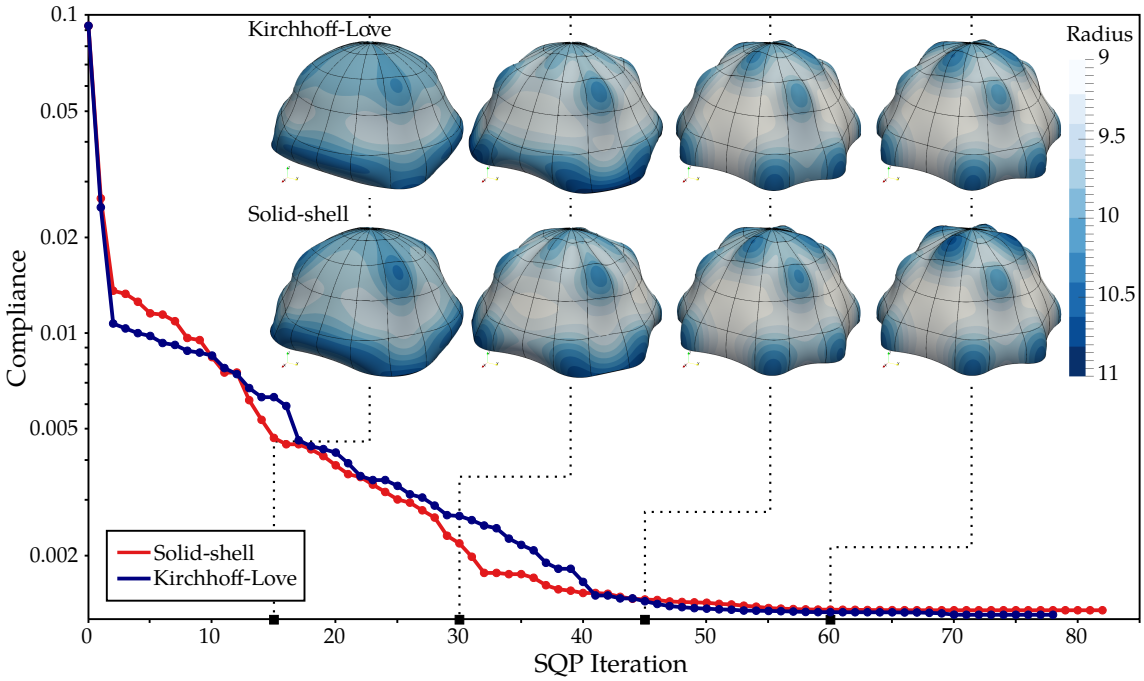


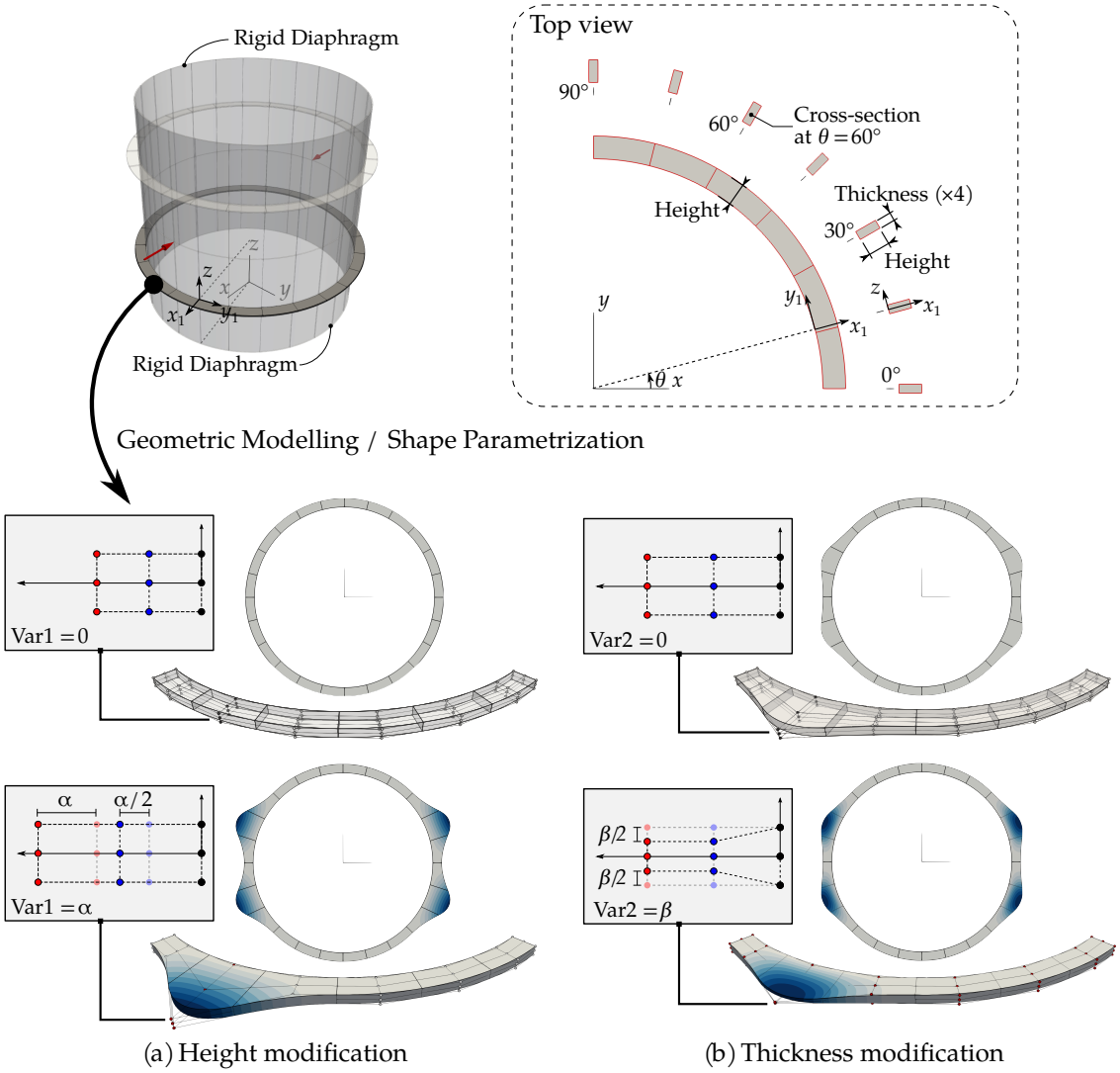
Figure 3.23 – History of compliance and shape during the optimization of the pinched hemisphere when using Kirchhoff–Love NURBS shell elements and NURBS-based Solid-shell elements.

procedure reveals that both shapes give a final compliance of $1.34 \cdot 10^{-3}$ which is 1.5% of the initial value $9.24 \cdot 10^{-2}$.

Beyond the similarity in final shape, the whole optimization histories look alike. Figure 3.23 shows the convergence of the objective function in both cases either with solid-shell NURBS elements or with Kirchhoff–Love NURBS elements. Optimization processes converge at the same speed and stop after an equivalent number of iterations within a given tolerance on function value. Some of the intermediary shapes are also shown in figure 3.23. Shape updates are very similar and the different hollows and bumps appear at the same iteration during the optimization.

3.4.2 Sizing Optimization with a Solid-shell approach

Stiffened cylinder problem The volume representation of the structure offers the possibility to optimize the thickness profile continuously. In order to highlight this feature, we present the optimization of a stiffened cylinder. This examples is derived from the pinched cylinder problem. Two stiffeners are added to the cylinder as depicted on figure 3.24. The stiffeners are located at heights 150 and 450. Initially, these stiffeners have a constant rectangular cross-section of height 30 and thickness 3. The other geometric and material parameters are detailed in figure 3.20. The optimization problem consists in modifying the cross section along the stiffeners in order to improve the global stiff-



(a) Height modification (b) Thickness modification

Figure 3.24 – The stiffened cylinder: two stiffeners are added to the initial pinched cylinder test case. The cross-section of the stiffener can be geometrically parametrized and thus optimized. Here, for each section of control points, two variables impose local design modifications in (a) the height and in (b) the thickness directions respectively.

ness of the structure. A volume constrain is set in order to keep the final volume lower or equal to the initial one V_0. Due to the symmetry of the problem, only an eighth of the structure is considered.

Bi-step approach Since the thickness is much less sensitive than the height, combining both quantities in a single optimization problem may lead to mostly modify the height of the stiffener. That is why we came up with the idea of performing the optimization in sequences of two steps, *i.e.* the *shape step* where we optimize the height, and the *sizing*

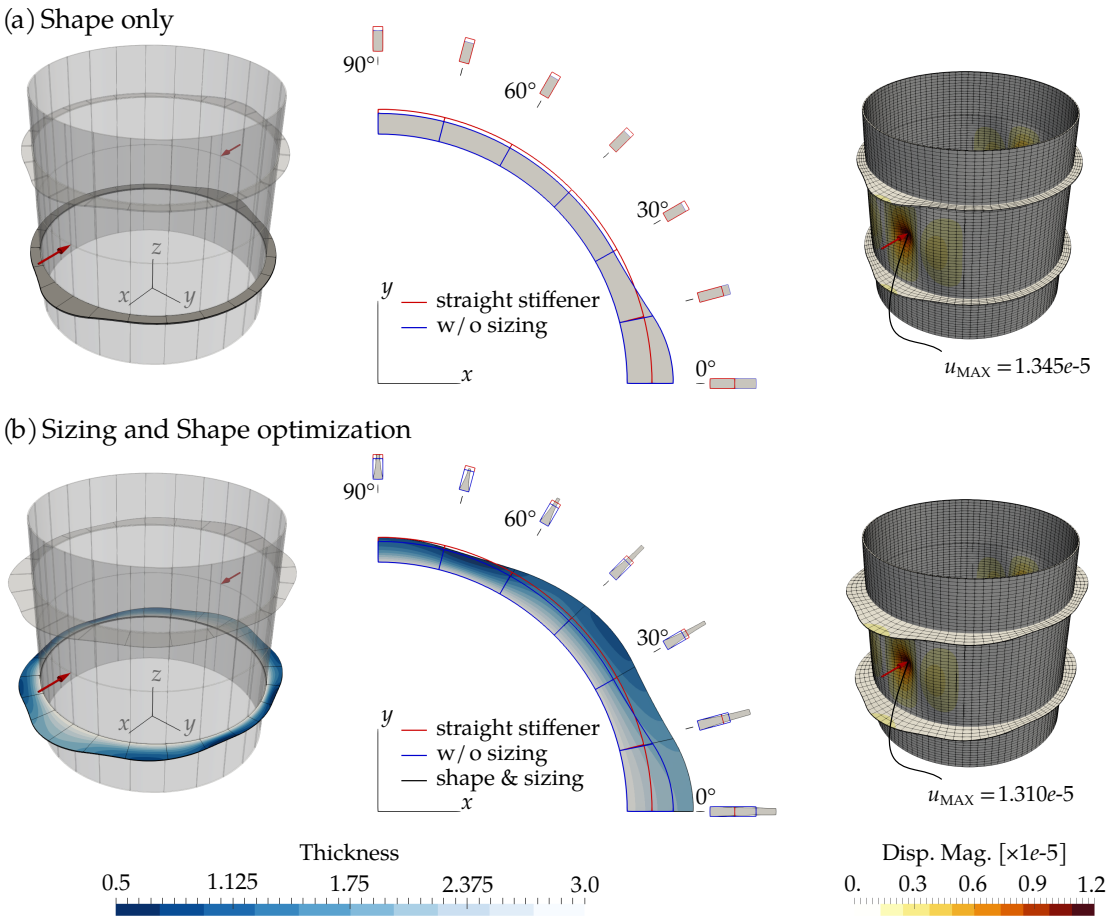


Figure 3.25 – Optimization results of the stiffened cylinder: (a) final shape when only the stiffener height is modified, and (b) final shape when both height and thickness are optimized (annotations on figure 3.24 can be helpful to understand the detailed views of the stiffener).

step where we optimize the thickness.

For this purpose, we define a first set of design variables $\boldsymbol{h}$ that update the height of the stiffener as depicted on figure 3.24(a). A second set of design variables $\boldsymbol{t}$ modifies the thickness as shown in figure 3.24(b). The shape step consists in minimizing the compliance with a maximal volume constrain. An additional bound is set in order to limit the height of the stiffener h in the range $[25, 100]$. The sizing step minimizes the volume V under the constrain that the final compliance is lower or equal to the one obtained from the shape step c^*. The design space for the thickness t is limited to $[0.5, 3.0]$. The global optimization process is made of sequences of a shape step followed by a sizing step. The fixed point process iterates until the global convergence is reached; that is, convergence on the design variables, the compliance and the volume.

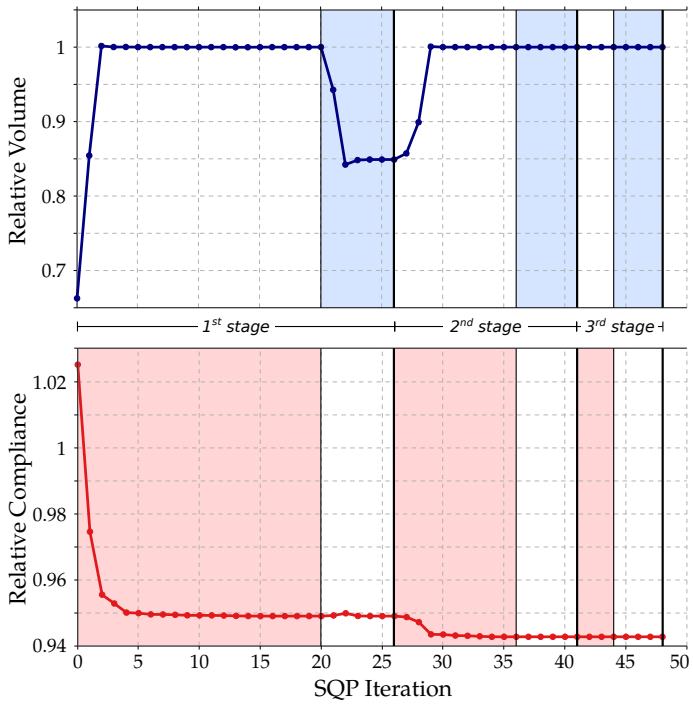


Figure 3.26 – Optimization history of the bi-step approach applied to the stiffened cylinder problem. Three global iterations with successive shape and sizing steps are preformed until convergence. Red zones denote the shape steps where the compliance is minimized and blue zones denote the sizing step where the volume is minimized.

The k^{th} sequence takes the following form:

$$\boldsymbol{h}^k, \boldsymbol{t}^k \longrightarrow \underbrace{\begin{array}{rl} \text{minimize} & f_{\text{cpl}} \\ \text{w.r.t.} & \boldsymbol{h} \in \mathbb{R}^n \\ \text{such that} & 25 \leq h \leq 100 \\ & V \leq V_0 \end{array}}_{\text{1. Shape Step}} \longrightarrow \boldsymbol{h}^{k+1}, \boldsymbol{t}^k \longrightarrow \underbrace{\begin{array}{rl} \text{minimize} & V \\ \text{w.r.t.} & \boldsymbol{t} \in \mathbb{R}^n \\ \text{such that} & 0.5 \leq t \leq 3.0 \\ & f_{\text{cpl}} \leq c^* \end{array}}_{\text{2. Sizing Step}} \longrightarrow \boldsymbol{h}^{k+1}, \boldsymbol{t}^{k+1} \quad (3.124)$$

Both optimization problems are complementary. The shape step will use all the available material in order to minimize the compliance. The sizing step rearranges the material distribution through the thickness in order to save material. Then, this saving material can be further use in the shape step in order to continue minimizing the compliance. This sequential optimization strategy is sensitive to the initial starting point and may lead to local optimum. The height and the thickness design variables are decoupled. Thus, to ensure a good quality result, we performed a multi-start algorithm. On top of that, we notice that multidisciplinary optimization strategies could be of interest to properly solve this problem and we refer the interested reader to [142] and [143] for futher details regarding this point.

Results The results of this study are given in figure 3.25. Optimizing through the thickness leads to a better design in the sense of the compliance. It helps to make a better use of the material volume. By reducing the thickness in some locations, the height of the stiffener can be increased in others. The initial compliance in the case of a straight stiffener is about $c_0 = 1.73e\text{-}06$. When only shape optimization is employed, the final compliance is about $c_1 = 1.68e\text{-}06$. The final compliance when both shape and sizing optimizations are performed is about $c_2 = 1.64e\text{-}06$. Therefore, the compliance gain when adding the sizing optimization is almost twice the compliance gain when only the height of the stiffener is taken into the optimization process.

Figure 3.26 depicts the optimization history. It takes three global iterations to converge. The main design changes are imposed during the first stage (i.e. the first succession of a shape and a sizing step) and the second shape step. For this optimization case, these steps may be enough to get an optimal geometry. Finally, this example shows that the solid-shell NURBS element offers an interesting way to deal with sizing optimization. It extends all the advantages of the NURBS parametrization to sizing optimization problem.

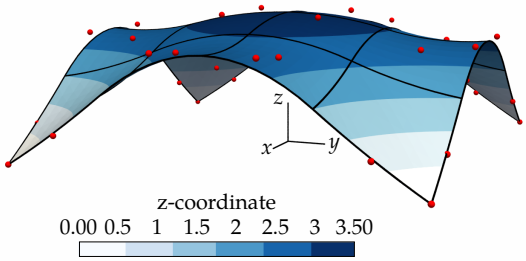
3.4.3 A multi-model optimization process

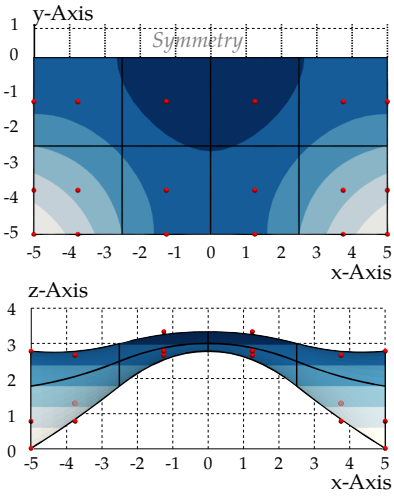
We present here a multi-model optimization. The idea is to combine the potential of both Kirchhoff–Love and solid-shell formulations. In a first stage, the Kirchhoff–Love formulation is used to perform a shape optimization in which the shell structure is varied in the out-of-plane direction. Then, the optimal surface is used to generate a volume model of the structure. In the second stage, thickness optimization is done on the volume model by using the developed solid-shell approach. We notice that this multi-model optimization appears consistent since it corresponds to the design process of structures. The first stage is the prototyping in which the influence of many parameters is studied. It requires a model with low computational cost in order to perform many simulations. Once the global shape of the structure is obtained, an high-fidelity volume model is created. Instead of directly using this model to generate the CAD drawings, one can perform a last optimization to apply final adjustments. Thank to the isogeometric approach, the final geometry can directly be transferred to the manufacturing step since no conversion in a CAD format is required.

We show the reliability of the procedure to design the variable thickness roof. Figure 3.27 presents the main steps and results for this test case. The global shape (step 1 in figure 3.27) is obtained by solving the already tackled example of the square roof problem (see figures 3.19 and associated discussion). This example is based on the work of [144] and similar studies can be found for example in [145, 146]. The optimal positions of the control points are given on the top view and on the side view of the structure on figure 3.27.

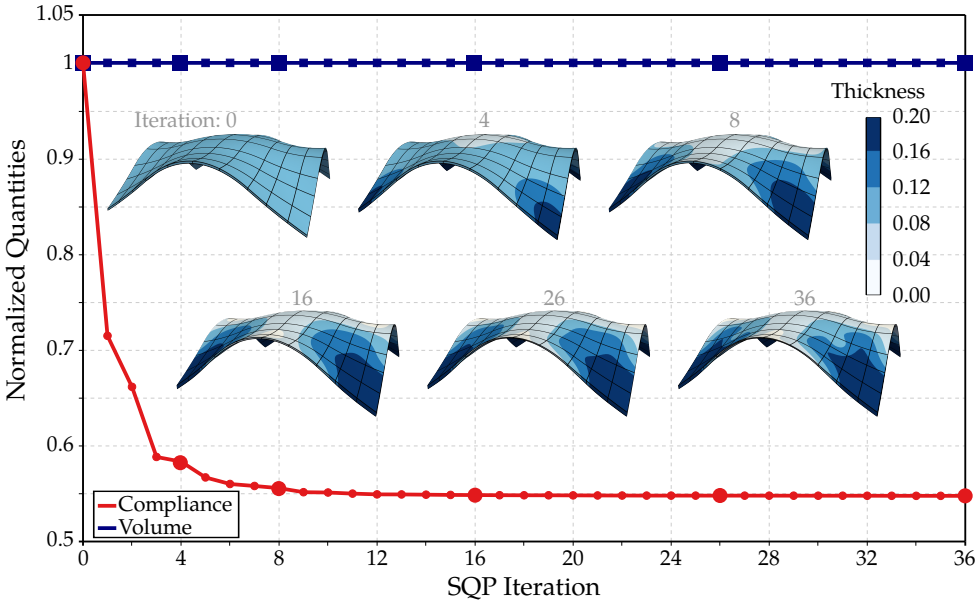
Once the global shape of the structure is obtained, one can build a volume model by

Step 1. Global shape optimization using KL shell





Step 2. Detailed thickness optimization using solid-shell



☑ Final optimal design

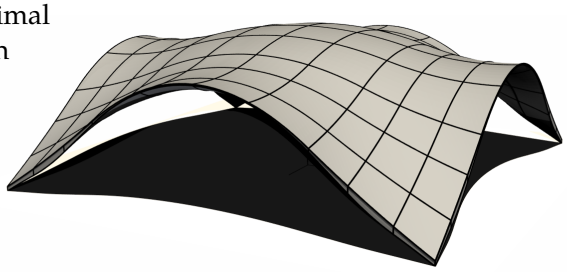


Figure 3.27 – Multi-model process

offsetting the previous surface in z-direction with a distance of $t = 0.10$. We consider only one quarter of the structure. The in-plane parametric directions are defined by the same knot vector $U = \{0\ 0\ 0\ \frac{1}{6}\ \frac{2}{6}\ \frac{1}{2}\ \frac{3}{4}\ 1\ 1\ 1\}$. Degree one is chosen in the thickness direction for building the optimization model. Thus, the optimization model counts 25 elements. The design variables modify the distance between the upper and lower surfaces. They parametrize the distance between neighboring control points of both upper and lower surfaces. The design variables are bounded so that the distance between the related control points varies in range $t_{\min} = 0.025$ and $t_{\max} = 0.20$. The control points located on the symmetric planes are coupled in order to preserve the geometric continuity which leads to a total of 36 design variables. Finally, we choose to keep the upper surface unchanged by moving only the lower control points. For the analysis, a refinement level of two is performed: it discretizes the structure with 20-by-20 quadratic elements.

The sizing optimization (step 2) consists in minimizing the compliance under the constraint of keeping the volume lower than the initial one. The optimization history is presented in figure 3.27 (see step 2). The final ratio of the compliance is below 0.55. Therefore, the thickness optimization drastically improves the behaviour of the structure. It can be observed that the thickness distribution of the final geometry is quite complex. In case of discrete thicknesses, being able to accurately represent such a distribution may require a high number of elements. Thus, the number of design variables would be higher. Finally, the optimal design (see figure 3.27, bottom) has regions with thin thickness and regions with thick thickness. At the corners, the thickness is equal to 0.20 which is, regarding the slenderness of the structures, relatively important. Using Kirchhoff-Love elements would not be relevant to analyze such a structure.

Chapter 4

Domain Decomposition Methods

Efficient solvers for large-scale systems have been intensely studied over the past three decades. In structural mechanics, non-overlapping domain decomposition methods have shown strong potentials, and are now considered as robust strategies for solving complex problems. In this chapter we take advantage of these established performances for analyzing non-conforming isogeometric multipatch shell structures. To this purpose, we first introduce a Mortar coupling for Kirchhoff–Love shells. Then, step-by-step we build a dual domain decomposition algorithm. Relevant theoretical and practical key points are presented, along with numerical investigations.

Contents

4.1 Non-Conforming Parametrization

4.1.1 Multipatch Geometric Models

NURBS are recognized as a powerful tool for geometric design. Thus, it is the predominant technology in many CAD software for designing high-resolution geometries. We now know that NURBS are also powerful for the analysis. Unfortunately, the prerequisites for the geometric modeling and for the numerical simulation are different. In other words, an automatic transfer from a NURBS-based CAD model to an analysis-suitable model is not straightforward.

More specifically, a single NURBS patch cannot represent complex industrial geometries. Instead, multiple NURBS patches are required where each of them describes a specific part of the overall structure. Thus, it is crucial that IGA deals with multipatch models if the purpose is to provide a new versatile method in engineering design. The issue concerns the junctions between those multiple patches: how the coupling between the patches is tackled during both geometric modeling and analysis steps ? Figure 4.1 illustrates this challenge. The difficulty lies in the fact that geometric modeling does not pay a special attention to the presence of multiple patches: each patch is defined inde-

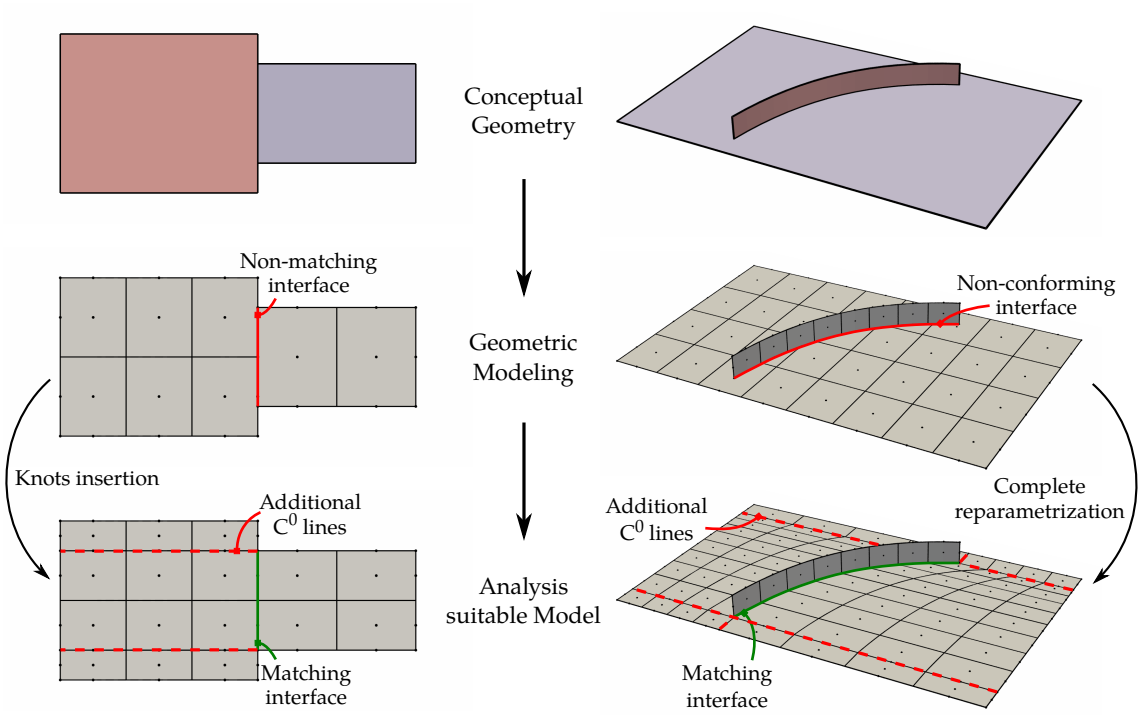
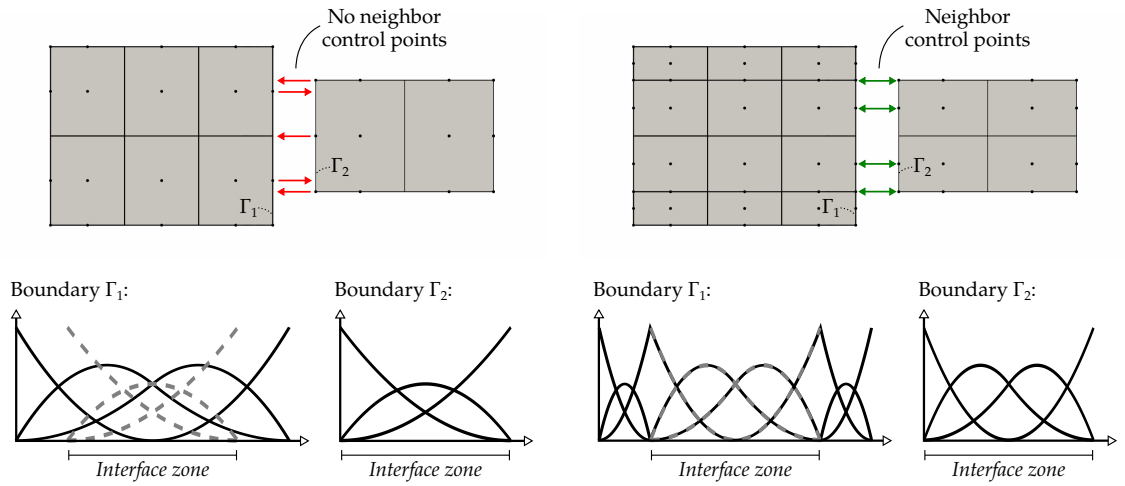


Figure 4.1 – NURBS modeling generates multipatch geometries where each patch represents a specific part of the complete structure. Naturally, these patches present non-matching or even non-conforming parametrization at their interfaces. Additional procedures are required to get a final model with conforming and matching interfaces.



(a) Non-matching interface (b) Matching interface

Figure 4.2 – Connecting conforming patches that are initially non-matching (a): the basis functions of both patches need to be modified in order to become identical over the interface zone (b).

pendently. The parametrization of one patch is not defined under a particular constraint due to any other patches. It leads to patch interfaces with non-matching or even non-conforming properties. From here on, we use the following terminology to characterize those interfaces [147]:

— it is **non-conforming** when some elements are overlapped by the interface,

— it is conforming but **non-matching** when the interface is aligned with the edges of the patches but the control points on both sides do not match,

— it is **matching** when the discretizations are exactly the same on both sides of the interface.

NURBS-based CAD models seldom, if ever, have matching interfaces. Depending on the geometry, NURBS modeling can lead to non-matching interfaces as depicted in figure 4.1, left. But in the more general case, non-conforming interfaces between the patches are encountered (see figure 4.1, right).

4.1.1.1 Analysis suitable models

For the analysis of multipatch models, two approaches can be classified. The first approach consists in calling on **re-parametrization procedures** to transform the patch junctions into fully matching interfaces [148]. One can operate this way because matching interfaces are easily tackled during the analysis by concatenating the duplicated control points at the interface. By analogy with the standard FEM framework, it can be seen as connecting two finite elements. However, generating matching interfaces between NURBS patches is not a trivial task. In the simplest case of a non-matching but conforming interface (see figure 4.1, left), knots insertion (or removal) and degree

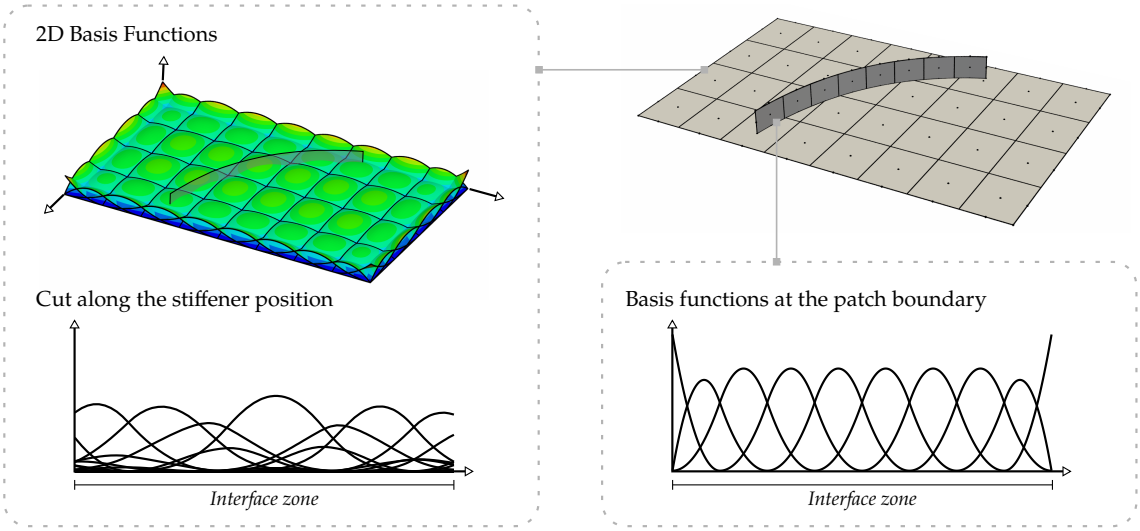


Figure 4.3 – Example of a stiffened panel with non-conforming parametrizations: the interface crosses the elements from the panel. The trace spaces of both sides are particularly different.

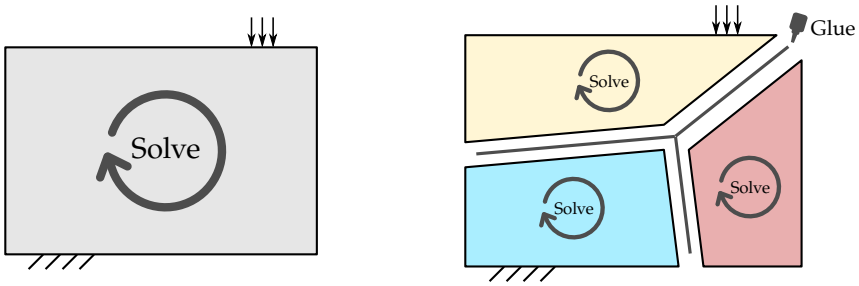
elevation (or reduction) allow to comply the requirement for a matching interface. More specifically, this re-parametrization procedures transform the basis functions of both sides so that they become identical along the interface. The process is highlighted in figure 4.2.

Transforming non-conforming interfaces into fully matching interfaces is trickier. It cannot be done by invoking the fundamental geometric algorithms (knots insertion, degree elevation, etc.) as for the non-matching case. The difficulty comes partly from the tensor-product nature of NURBS. Let us take the example of the stiffened panel displayed in figure 4.1(right). We depicted the trace spaces for this example in figure 4.3. From the panel side, the interface crosses several elements. When the whole cut is not along one parametric direction, it is not possible to obtain a matching junction by modifying the basis functions at the knot spans level. Instead, a complete re-parametrization has to be performed if a matching interface is required. Despite its simple rectangular shape, the panel needs to be decomposed into six patches to get an overall parametrization with matching interfaces as shown in figure 4.1. The problem stems from the fact that this re-parametrization can hardly be automated, especially for volume models [149].

In contrast with the first approach based on re-parametrization procedures, the second option for analyzing multipatch models consists in using **advanced numerical tools** for coupling the non-conforming patches. In this work, we follow this methodology because it appears to us more appropriate for an optimization process. In particular, we want to avoid to modify the parametrization of the model at every iteration of the resolution. This would lead to a too cumbersome method.

4.1.1.2 Domain Decomposition

The examples depicted in figure 4.1 reveal an additional point: NURBS modeling generates a **natural decomposition** of the domain. As already observed, we know that multiple patches are needed to model a large structure with, for example, multiple parts. From an analysis point of view, each patch can be seen as a sub-domain of the computational domain. This observation motivates us to focus on an efficient class of solvers: the non-overlapping Domain Decomposition Methods [150]. Kleiss et al. [151] initially noticed this natural link between IGA and DDM, and they developed an isogeometric solver for matching multipatch models following this Domain Decomposition strategy.



(a) Reference problem (b) Domain Decomposition

Figure 4.4 – Non-overlapping Domain Decomposition Methods: (a) instead of solving one large system defined over the whole the computational domain Ω, (b) the domain is subdivided into multiple sub-domains $\Omega^{(s)}$. The resolution is performed by solving local systems defined per sub-domain, and interface condition are recovered through an iterative process.

The point of interest lies in the potential of DDM for solving large-scale systems. In fact, those non-overlapping DDM have been actively studied over the past decades as efficient parallel solvers for systems with possible millions of DOF. The idea is to subdivide the computational domain into non-overlapping sub-domains: instead of solving one large system, multiple independent local systems are solved per sub-domain and interface conditions are recovered through an iterative process (see figure 4.4). More specifically, the Finite Element Tearing and Interconnecting [152] and the Balancing Domain Decomposition [153] are often seen as originators of the two main branches of those non-overlapping DDM. BDD is usually referred to as a primal approach since it chooses the interface displacement field as main unknowns, whereas FETI is usually referred to as a dual approach because it privileges the interface loads. Since then, a large number of papers and variants have been developed and successfully applied in many context: e.g. plate and shells [154–156], heterogeneous problems [157, 158], contact problems [159], etc.

As stated above, the key aspect regarding how non-overlapping Domain Decomposition Methods concerns the interface conditions between the sub-domains. In the more general dual approach, the sub-domains are glued together through Lagrange multipliers. In our case, the interfaces between the patches are possibly non-conform.

Interestingly, there exists a method to couple non-conforming meshes through Lagrange multipliers: the so-called Mortar method [160]. Thus, Mortar coupling and dual DDM are strongly related, and this is why FETI methods with Mortar finite elements have been studied in the classical FEM framework [161].

Based on all these discussions, we construct in the following a dual Domain Decomposition algorithm for analyzing non-conforming multipatch models. The starting point is the introduction of a Mortar method for the coupling of non-conforming Kirchhoff–Love shells. Then, step-by-step we build the solver by calling upon the core strategy of non-overlapping DDM.

4.1.2 Treatment of non-conforming interface

Before dealing with the particular case of Kirchhoff–Love shells, we shortly review the main methods for coupling non-conforming patches in the context of classical second-order linear elasticity problems. There exist three principal methods, namely penalty coupling [162–164], Mortar coupling [19, 147, 165–170], and Nitsche coupling [171–176]. Especially, Apostolatos et al. [171] compare these three methods for 2D linear elasticity problems in the IGA framework.

Let us take the case of two patches occupying, respectively, sub-domains $\Omega^{(1)}$ and $\Omega^{(2)}$, and joining at interface Γ. On each sub-domains (s), the following system of equations holds:

$$\operatorname{div} \sigma^{(s)} + f^{(s)} = 0 \quad \text{in } \Omega^{(s)}, \tag{4.1}$$

$$\sigma^{(s)} = C^{(s)} : \varepsilon\left(u^{(s)}\right) \quad \text{in } \Omega^{(s)}, \tag{4.2}$$

$$\sigma^{(s)} n^{(s)} = g \quad \text{on } \Gamma_g \cap \partial\Omega^{(s)}, \tag{4.3}$$

$$u^{(s)} = u_0 \quad \text{on } \Gamma_u \cap \partial\Omega^{(s)}, \tag{4.4}$$

where the notations stand for as follows: volume effort f, elasticity tensors C, surface effort g and outward unit normal vector n defined on the Neumann boundary Γ_g, surface displacement u_0 imposed on the Dirichlet boundary Γ_u. The virtual work formulation of this uncoupled problem can be written as:

$$\delta W_{\text{int}}^{(1)} + \delta W_{\text{int}}^{(2)} - \delta W_{\text{ext}}^{(1)} - \delta W_{\text{ext}}^{(2)} = 0, \tag{4.5}$$

where W_{int} and W_{ext} denote the internal and external works, respectively.

In order to get the coupled problem, additional conditions are set on the interface. The continuity of the displacement and the equilibrium of the tractions need to be imposed through:

$$u^{(1)} - u^{(2)} = 0 \quad \text{on } \Gamma, \tag{4.6}$$

$$\sigma^{(1)} n^{(1)} + \sigma^{(2)} n^{(2)} = 0 \quad \text{on } \Gamma. \tag{4.7}$$

Combining these conditions (4.6) and (4.7) with the locals systems (4.1)–(4.4) is strictly equivalent to a linear elasticity problem posed on the whole initial domain Ω. In case of non-conforming interfaces, the coupling conditions cannot be imposed in the strong way. Instead, those coupling conditions are prescribed in the weak sense, and thus, the different coupling methods differ by the way those weak conditions are formulated. Those coupling methods augment the virtual work formulation (4.5) by different contributions.

Penalty coupling The penalty method enforces the displacement continuity (4.6) by introducing the following penalty virtual work:

$$\delta W^{\text{penalty}} = \int_\Gamma \alpha_{\text{pen}} \left(u^{(1)} - u^{(2)}\right) \cdot \left(\delta u^{(1)} - \delta u^{(2)}\right) \text{dl}, \tag{4.8}$$

where α_{pen} is a penalty parameter of large magnitude. The accuracy of the method depends on the value of this parameter. If its value is too large, then the discrete coupled system is ill-conditioned. Conversely, a too small value does not ensure the constraints to be satisfied. Furthermore, the penalty value is often problem-dependent and thus, a given value of α_{pen} may not be appropriated for a variety of problems. Finally, note that the basic penalty method as expressed in (4.8) cannot recover the interface conditions of the strong form, and more specifically the effort equilibrium (4.7).

Mortar coupling The Mortar method introduces an additional field, the Lagrange multiplier field $\lambda \in \mathcal{M}$ ($\mathcal{M}$ being an ad-hoc space), which represents both of the interface traction forces:

$$\lambda = -\sigma^{(1)} n_\Gamma = -\sigma^{(2)} n_\Gamma, \quad \text{with} \quad n_\Gamma = n^{(1)} = -n^{(2)}. \tag{4.9}$$

The effort equilibrium (4.7) is included into the strong form of the problem through (4.9) whereas the displacement continuity (4.6) is imposed in a weak sense. It results in a mixed formulation where the additional virtual work performed by the Lagrange multiplier field reads:

$$\delta W^{\text{mortar}} = \int_\Gamma \left(u^{(1)} - u^{(2)}\right) \cdot \delta\lambda \, \text{dl} \; + \; \int_\Gamma \left(\delta u^{(1)} - \delta u^{(2)}\right) \cdot \lambda \, \text{dl}\,. \tag{4.10}$$

Because the communication between the sub-domains is performed via the Lagrange multiplier only, the Mortar approach enables to keep separated and unmodified the local stiffness operators (*i.e.* associated to each sub-domain). This feature is the basis of the non-overlapping DDM which will be used in this work to build our dual Domain Decomposition solver. Finally, let us recall that a drawback of the Mortar approach is the construction of the approximation space of $\mathcal{M}$. Indeed, an *inf–sup* condition needs to be satisfied in order to avoid undesirable energy-free oscillations [165].

Nitsche coupling The Nitsche method also enforces the Neumann interface condition (4.7) in the weak sense. Starting with the Mortar coupling, the idea is to replace the Lagrange multiplier by the mean interface force [177]. When the neighbor sub-domains have close material properties, the mean weights are usually set identical and equal to $1/2$. In this case, the contribution of the interface work for the Nitsche approach is given by:

$$\begin{aligned}\delta W^{\text{nitsche}} = &- \int_\Gamma \left(u^{(1)} - u^{(2)}\right) \cdot \frac{1}{2} \left(\delta\sigma^{(1)} + \delta\sigma^{(2)}\right) n_\Gamma \,\mathrm{dl} \\ &- \int_\Gamma \left(\delta u^{(1)} - \delta u^{(2)}\right) \cdot \frac{1}{2} \left(\sigma^{(1)} + \sigma^{(2)}\right) n_\Gamma \,\mathrm{dl}\,.\end{aligned} \tag{4.11}$$

In addition, this formulation needs to be stabilized in order to ensure the ellipticity of the Boundary Value Problem. This can be done by adding a term of form (4.8) and by solving a generalized eigenvalue problem to appropriately estimate the stabilization parameter [171, 172]. However, let us notice that a non-symmetric version of the Nitsche method exist. This formulation does not require the introduction of additional stabilization terms: the criterion for stability is weakly satisfied and the method converges with optimal rates in the strain energy norm [173, 176, 178].

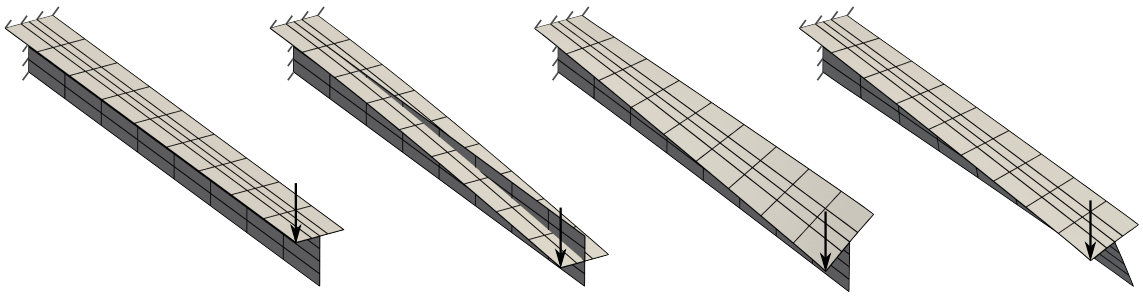
4.2 Mortar coupling for Kirchhoff–Love shells

In the particular context of isogeometric Kirchhoff–Love shells, both of the three coupling methods (penalty, Mortar, and Nitsche) have been successfully applied for large and complex industrial structures, even for non-linear analysis [164, 170, 179]. Among all methods, various arguments are put forward regarding different criteria as for example the simplicity, the accuracy, the robustness, or even the range of applications. Nevertheless, a great potential of the Mortar method seems not to have been yet fully exploited: as already mentioned, there exists a natural link between this class of coupling methods and the family of Domain Decomposition algorithms. This point deserves particular attention, and we attempt in this work to motivate researches in that direction.

4.2.1 Kinematic coupling conditions

We are interested in the analysis of stiffened structures as illustrated in figure 4.1, right. The junction between the panel and the stiffener has only C^0 continuity along the non-conforming patch interface. We present a coupling method which enables to track this type of interface.

In fact, few works dealing with the Mortar coupling of isogeometric Kirchhoff–Love shells that consider the case of G^1-geometric continuity interface can be listed [162, 170]. However, the formulation in the context of linear elasticity analysis of multipatch



(a) Initial Configuration (b) No coupling (c) Displacement only (d) Full coupling

Figure 4.5 – Kinematic constraints for Kirchhoff–Love shells: at the patch junction, the three components of the displacement fields need to be coupled (c). Additionally, the continuity of the rotation in the tangent direction of this interface curve is imposed (d).

Kirchhoff–Love shells which connect with arbitrary angles is not highly studied yet. Some aspects can be found in Duong et al. [180] but the applicability for linear analysis may not appear obvious: the kinematic constraints are built by dot and cross products between quantities of neighboring patches which results in quadratic terms (*w.r.t.* the displacement). It thus requires a linearization and the resolution is done by the Newton-Raphson method [see appendix D.2. in 180].

For the sake of simplicity, the method is presented in the case of two non-overlapping sub-domains $\Omega^{(1)}$ and $\Omega^{(2)}$ without loss of generality. At the common interface denoted by Γ, the continuity of the displacement has to be ensured. Furthermore, in case of a shell junction with rigid hinge, an additional constraint is required which enforces the continuity of the rotation in the tangential direction associated to the interface curve [181]. Thus, we formulate the kinematic constraints as:

$$u^{(1)} = u^{(2)} \quad \text{on } \Gamma, \tag{4.12}$$

$$\Phi^{(1)} \cdot \boldsymbol{t} = \Phi^{(2)} \cdot \boldsymbol{t} \quad \text{on } \Gamma, \tag{4.13}$$

where $\boldsymbol{t}$ is a unit tangent vector associated to the interface curve. Figure 4.5 highlights the effect of these constraints. The T-Beam is built with two non-conforming patches. Thus, without the coupling, there is no load transfer from the upper patch to the lower one. With the imposition of the displacement continuity at the interface, the junction acts as a hinge. Finally, the continuity of the rotation leads to a complete coupling between the patches (see figure 4.5). The continuity of the displacement (4.12) leads to three equations (each Cartesian components). The continuity of the rotation (4.13) leads to one equation because we only enforce the rotation in a given direction (the expression of the rotation vector Φ is given in equation (3.47)).

4.2.2 Weak coupling with a Mortar approach

In order to impose the continuity of the displacement (4.12) and the continuity of the rotation (4.13), we make the use of a Mortar coupling. To this end, we introduce

two Lagrange multipliers, one for each kinematic constraints as also done by Apostolatos et al. [171]. Let us denote $\tilde{\lambda} \in \mathcal{L}_d$ and $\hat{\lambda} \in \mathcal{L}_r$ ($\mathcal{L}_d$ and $\mathcal{L}_r$ being ad-hoc spaces) the Lagrange multipliers associated to the displacement and the rotation continuity constraints, respectively. The virtual work performed by these Lagrange multiplier fields is given by:

$$\begin{aligned}\delta W^{\text{mortarKL}} = &\int_\Gamma \left(u^{(1)} - u^{(2)}\right) \cdot \delta\tilde{\lambda} \,\mathrm{dl} \;+\; \int_\Gamma \left(\delta u^{(1)} - \delta u^{(2)}\right) \cdot \tilde{\lambda} \,\mathrm{dl} \\ &+ \int_\Gamma \left(\Phi^{(1)} \cdot \boldsymbol{t} - \Phi^{(2)} \cdot \boldsymbol{t}\right) \delta\hat{\lambda} \,\mathrm{dl} \;+\; \int_\Gamma \left(\delta\Phi^{(1)} \cdot \boldsymbol{t} - \delta\Phi^{(2)} \cdot \boldsymbol{t}\right) \hat{\lambda} \,\mathrm{dl}\,.\end{aligned} \tag{4.14}$$

We can now introduce the discretizations $u^{h(1)}$ and $u^{h(2)}$ of the displacement fields, and $\tilde{\lambda}^h$ and $\hat{\lambda}^h$ of the Lagrange multipliers as follows:

$$\begin{aligned} u^{h(s)}(\theta_1, \theta_2) &= \sum_i R_i^{(s)}(\theta_1, \theta_2)\, \boldsymbol{u}_i^{(s)} \quad s = 1, 2, \\ \tilde{\lambda}^h(\xi) &= \sum_i \tilde{R}_i(\xi)\, \tilde{\boldsymbol{\lambda}}_i, \\ \hat{\lambda}^h(\xi) &= \sum_i \hat{R}_i(\xi)\, \hat{\boldsymbol{\lambda}}_i. \end{aligned} \tag{4.15}$$

With these discretizations in hand, the variational principle written in the discrete form gives the following coupled linear system to be solved:

$$\begin{bmatrix} \mathbf{K}^{(1)} & 0 & \mathbf{C}^{(1)T} & \mathbf{Z}^{(1)T} \\ 0 & \mathbf{K}^{(2)} & \mathbf{C}^{(2)T} & \mathbf{Z}^{(2)T} \\ \mathbf{C}^{(1)} & \mathbf{C}^{(2)} & 0 & 0 \\ \mathbf{Z}^{(1)} & \mathbf{Z}^{(2)} & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \tilde{\boldsymbol{\lambda}} \\ \hat{\boldsymbol{\lambda}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}^{(1)} \\ \boldsymbol{F}^{(2)} \\ 0 \\ 0 \end{pmatrix}, \tag{4.16}$$

where $\mathbf{K}^{(s)}$ and $\boldsymbol{F}^{(s)}$ denote the local stiffness matrices and load vectors, respectively. The displacement coupling matrices $\mathbf{C}^{(s)}$ and the rotation coupling matrices $\mathbf{Z}^{(s)}$ consist in sparse signed rectangular operators. The two last sets of equations of system (4.16) weakly ensure the continuity of the displacement fields (see (4.12)) and of the rotations in the tangent direction (see (4.13)):

$$\mathbf{C}^{(1)}\boldsymbol{u}^{(1)} + \mathbf{C}^{(2)}\boldsymbol{u}^{(2)} = 0, \tag{4.17}$$

$$\mathbf{Z}^{(1)}\boldsymbol{u}^{(1)} + \mathbf{Z}^{(2)}\boldsymbol{u}^{(2)} = 0. \tag{4.18}$$

Discretization The linear system (4.16) consists in a saddle point problem, implying that the approximation spaces of the Lagrange multipliers must be carefully chosen in order to satisfy an *inf–sup* condition. Recent works present optimal approximation spaces for Mortar coupling in isogeometric analysis [165, 168–170]. In particular, Schuß

et al. [170] deals with the G^1 coupling of non-matching Kirchhoff–Love shells. Based on these works, one could build optimal approximation spaces for the Mortar coupling of shells that intersect in an arbitrary, non-conforming manner (*i.e.* where the junction crosses a patch, or where the patches intersect with an angle, etc.). This issue is not tackled in this work. Instead, the strategy adopted here is only based on numerical experiments. Let p denote the smallest degree of the discrete displacement field of both subdomains $\Omega^{(1)}$ and $\Omega^{(2)}$. We define:

— for the displacement constraints, $\tilde{\lambda}^h$ as a vector-valued B-Spline function with degree $p-1$ since it is mainly related to traction forces,

— for the rotation constraint, $\hat{\lambda}^h$ as a scalar-valued B-Spline function with degree $p-2$ because it transfers a bending moment,

— an identical mesh refinement for both Lagrange multipliers. We discretize these fields using as many elements as the coarsest of the trace of domains $\Omega^{(1)}$ and $\Omega^{(2)}$ along the interface.

We assert that with such a choice, we never encountered instabilities in our computations. All numerical examples presented in the this work highlight the good behavior of this Mortar coupling. One can also notice that the choice of the approximation spaces for the Lagrange multipliers is completely decoupled from the overall approach applied in this work to build the Domain Decomposition solver. With other choices in hand, the remaining part of the method performs identically.

4.2.3 Implementation aspects

Since now, we have not detailed yet the **expression of the coupling operators** $\mathbf{C}^{(s)}$ and $\mathbf{Z}^{(s)}$. For this purpose, we opt for the same notation as done for example in Herrema et al. [164]. We denote by l the global DOF number of the j^{th}-component associated to the displacement vector $\boldsymbol{u}_b$, such that $u_l = \boldsymbol{u}_b \cdot \mathbf{e}_j$ (where $\mathbf{e}_j$ refers to the basis of the global Cartesian coordinate system). Similarly, let k denote the global DOF number of the Lagrange multipliers. With these notations in hand, one can express the following variations:

$$\frac{\partial u^h}{\partial u_l} = R_b \mathbf{e}_j, \quad \frac{\partial \tilde{\lambda}^h}{\partial \tilde{\lambda}_k} = \tilde{R}_a \mathbf{e}_i, \quad \frac{\partial \hat{\lambda}^h}{\partial \hat{\lambda}_k} = \hat{R}_a. \tag{4.19}$$

The coupling matrices are obtained by deriving the Mortar virtual work (4.14) with respect to the DOFs. The coupling matrix $\mathbf{C}$ that enforces the displacement continuity is given by:

$$\mathbf{C}_{kl} \;=\; \pm \int_\Gamma \frac{\partial \tilde{\lambda}^h}{\partial \tilde{\lambda}_k} \cdot \frac{\partial u^h}{\partial u_l} \,\mathrm{dl} \;=\; \pm \int_\Gamma \tilde{R}_a R_b \delta_{ij} \,\mathrm{dl}\,. \tag{4.20}$$

Thus, the displacement coupling matrix has a similar structure than classical mass matrices. It only brings into play products between the basis functions from the displacement

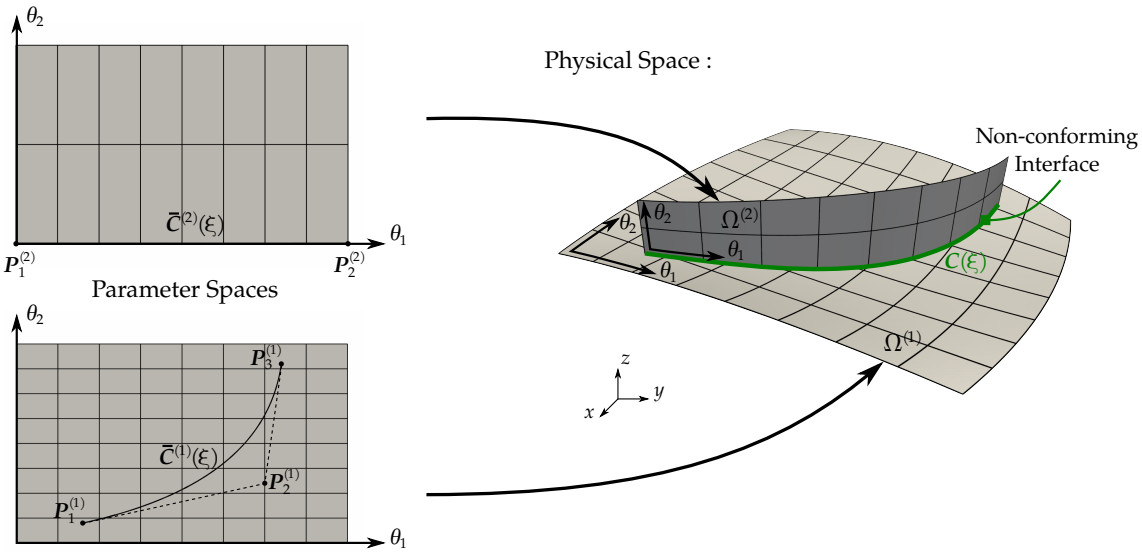


Figure 4.6 – Non-conforming parametrization: the interface between both surfaces is represented in their parameter spaces by using NURBS curves $\bar{C}^{(s)}$. These curves are generated in order to ease the building of the coupling operators for the Mortar approach.

and from the Lagrange multiplier fields. The symbol $\pm$ indicates whether the sub-domain is the master $(+)$ or the slave $(-)$. Accordingly to equation (4.14), in this case the sub-domain $\Omega^{(1)}$ is the master patch and the sub-domain $\Omega^{(2)}$ is the slave patch.

The expression of the coupling matrix $\mathbf{Z}$ associated to the continuity of the rotation is expressed as:

$$\mathbf{Z}_{kl} = \pm \int_\Gamma \frac{\partial \hat{\lambda}^h}{\partial \hat{\lambda}_k} \left(\frac{\partial \Phi}{\partial u_l} \cdot \boldsymbol{t} \right) \mathrm{dl} = \pm \int_\Gamma \hat{R}_a \left[\frac{\mathbf{a}_3}{J} \left(R_{b,\theta_2}(\mathbf{a}_1 \cdot \boldsymbol{t}) - R_{b,\theta_1}(\mathbf{a}_2 \cdot \boldsymbol{t}) \right) \right] \cdot \mathbf{e}_j \, \mathrm{dl}\,. \quad (4.21)$$

It involves the same geometric quantities as for the stiffness matrix. More specifically, one has to compute the basis functions and their derivatives, and then compute the local basis vectors. Finally, the only new quantity is the tangent unit vector $\boldsymbol{t}$ to the interface curve. This vector is deducted during the numerical integration over the coupling interface Γ as explain in the following.

Indeed, we still need to clarify how to **integrate numerically over the interface**. For general non-conforming interface, the methodology adopted in this work is depicted in figure 4.6. We manage to represent the interface in the parameter spaces through NURBS curves. As mentioned by Bauer et al. [182], it results from a surface-to-surface intersection problem which is a common task in CAD. Under the consideration that no approximation is involved during the generation of the curves, we have two definitions of the interface curve (one per subdomain). In other words, the following equation applies:

$$\boldsymbol{S}^{(1)}\left(\bar{\boldsymbol{C}}^{(1)}(\xi)\right) = \boldsymbol{S}^{(2)}\left(\bar{\boldsymbol{C}}^{(2)}(\xi)\right) = \boldsymbol{C}(\xi) \qquad \forall \xi \in [0,1]. \quad (4.22)$$

The embedded curves $\bar{C}$ allows to perform a change of variables which results in the following treatment to compute the coupling matrices. For instance, for matrix $\mathbf{C}$, we get:

$$\mathbf{C}_{kl} = \pm \int_{[0,1]} \tilde{R}_a(\xi)\, R_b\left(\bar{\boldsymbol{C}}(\xi)\right)\, \delta_{ij}\, |\boldsymbol{T}(\xi)|\, \mathrm{d}\xi, \tag{4.23}$$

and we apply a similar treatment for matrix $\mathbf{Z}$. The vector $\boldsymbol{T}$ is given by derivation as follows:

$$\boldsymbol{T}(\xi) = \frac{\mathrm{d}}{\mathrm{d}\xi}\boldsymbol{S}\left(\bar{\boldsymbol{C}}(\xi)\right) = \left(\bar{\boldsymbol{C}}_{,\xi}(\xi)\cdot\mathbf{e}_1\right)\mathbf{a}_1 \;+\; \left(\bar{\boldsymbol{C}}_{,\xi}(\xi)\cdot\mathbf{e}_2\right)\mathbf{a}_2. \tag{4.24}$$

Vector $\boldsymbol{T}$ is tangential to the interface curve. Therefore, once normalized, one can obtain the unit tangent vector $\boldsymbol{t}$ which comes into play in the rotation coupling matrix (4.21). Finally, numerical integration is performed using a Gaussian quadrature rule. A refinement by knot insertion of the embedded NURBS curves discretizes the interface in a finite number of segments. The number of segments and the number of Gauss points per-segment should be chosen such that the numerical error due to the quadrature rule remains negligible. At the end, the assembly of the coupling matrices is performed with a loop on the Gauss points of each segment:

$$\int_0^1 f(\xi)\,\mathrm{d}\xi \;=\; \sum_I \int_{\xi_I}^{\xi_{I+1}} f(\xi)\,\mathrm{d}\xi \;\approx\; \sum_I \sum_{gp} f(\xi_{I_{gp}}) J_{I_{gp}} \omega_{gp}, \tag{4.25}$$

where $\xi_{I_{gp}}$ denotes the position of the Gauss point on the parameter space of the embedded curve, $J_{I_{gp}}$ denotes the Jacobian of the last mapping such that the integral over $[\xi_I, \xi_{I+1}]$ is changed into an integral over [-1, 1], and ω_{gp} denotes the quadrature weight.

4.2.4 Monolithic resolution

Prior to the development of the dual Domain Decomposition algorithm, we present here preliminary numerical examples in order to highlight the correctness of our Mortar coupling for Kirchhoff–Love shells. The resolution is done with a direct solver applied to the coupled linear system (4.16).

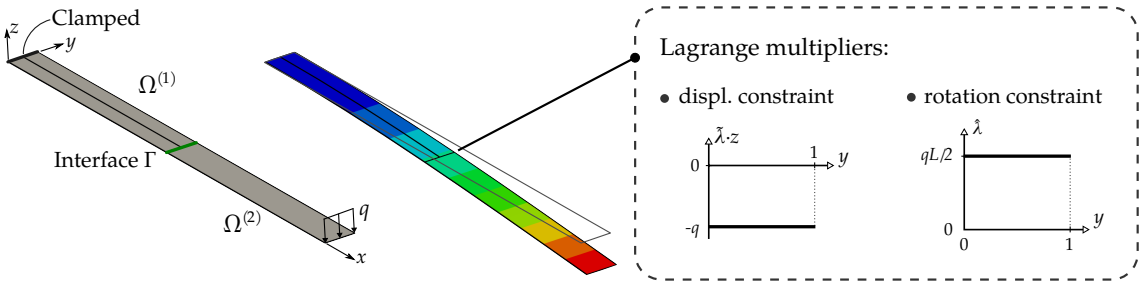


Figure 4.7 – Cantilever beam with end load: the beam is decomposed in two non-matching pacthes and coupled with the Mortar approach. The shape of Lagrange multipliers for the displacement and the rotation continuity constraints are shown one the right.

Beam bending In order to investigate the good behavior of the coupling, we start by considering simple test cases where analytical results are known. For instance, here we study the case of a cantilever beam with end load as described in figure 4.7. Based on the Euler–Bernoulli beam theory, it is possible to express along the beam, the vertical displacement and the stress resultants. We adopt the following parameters: beam length $L = 20\,\mathrm{m}$, width $w = 1\,\mathrm{m}$, and thickness $t = 0.1\,\mathrm{m}$, line load $q = 100\,\mathrm{N\,m^{-1}}$, Young's Modulus $E = 60\,\mathrm{GPa}$, and Poisson's ratio $\nu = 0$.

The beam is decomposed in two non-matching cubic patches of same size (*i.e.* the interface is located at half the length) as shown in figure 4.7. Since the deflection is expressed as a polynomial of degree 3, we should recover the analytical solution. Interestingly, we can link the Lagrange multiplier fields to the analytical values of the bending moment and the shear force. In fact, the z-component of the Lagrange multiplier $\tilde{\lambda}$ that imposes the continuity of the displacement represents the shear force, and the Lagrange multiplier $\hat{\lambda}$ that imposes the continuity of the rotation represents the bending moment. Thus, we can verify that we numerically obtain the following values:

$$\tilde{\lambda}_{\mathrm{AN}} = -q = -100\,\mathrm{N\,m^{-1}}, \qquad \text{and} \qquad \hat{\lambda}_{\mathrm{AN}} = \frac{qL}{2} = 1000\,\mathrm{N}. \tag{4.26}$$

The Lagrange multipliers are discretized with respect to the strategy suggested in section 4.2.2: one quadratic B-Spline element for the displacement constraint, and one linear B-Spline element for the rotation constraint. We run the analysis and we verify that the theoretical results are exactly recovered. The error is of magnitude of the computer precision. One should pick the patch $\Omega^{(2)}$ as the master, and the patch $\Omega^{(1)}$ as the slave in order to get the same sign as in equation (4.26). This test case highlights the mechanical meaning of the proposed Lagrange multiplier formulation. It is also useful for validating the implementation of the coupling operators. Finally, instead of putting the interface midway, one can cut the beam at different locations and investigate if the analytical expressions are recovered (which should be the case).

T-Shape beam The T-beam problem is built using two planar patches that are connected in a non-matching way as depicted in figure 4.8. The interface crosses the middle of the upper patch. The upper patch has one more element than the other in the beam direction, and both patches are discretized with cubic elements. We take the same numerical setting as in Herrema et al. [164] which are also given in figure 4.8.

We obtain a similar deformation as in Herrema et al. [164] where penalty coupling was performed. In order to show that the rotation constraint is well prescribed, we plot in figure 4.8(c) the resulting angle between the patches along the interface. It can be seen that it remains equal to 90° after deformation. Without the rotation constraint, the angle between the patches was found to be equal to 86.6 at the end of the beam because the interface acts as a hinge [164]. The presence of small oscillations for the coarser parametrization may be due to the non-optimality of the chosen spaces for the Lagrangian fields. Once again, it could be interesting to investigate the *inf–sup* condition

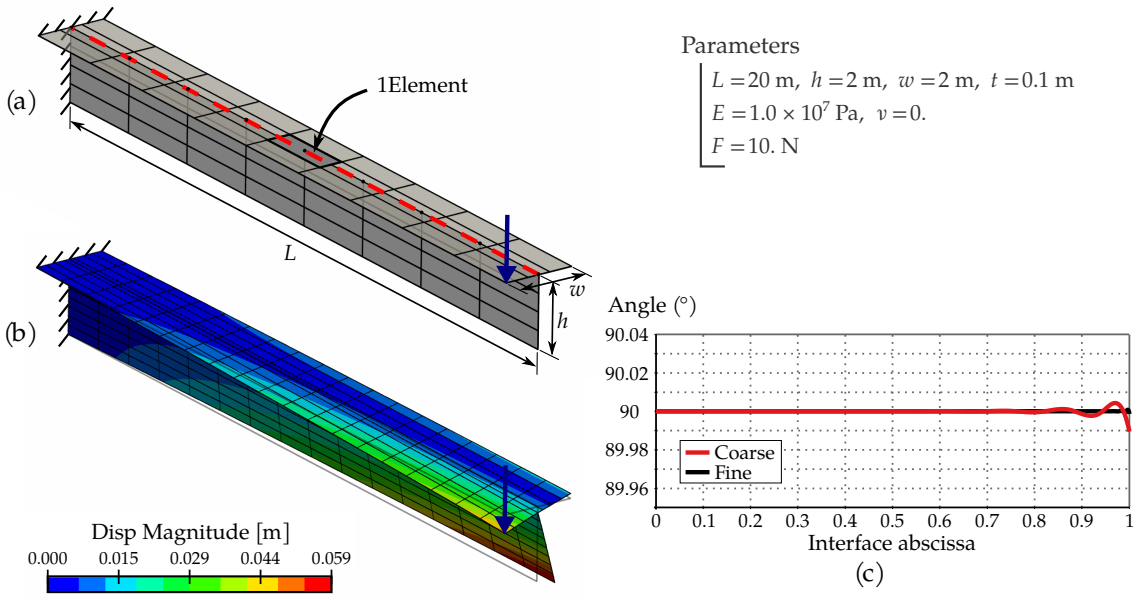


Figure 4.8 – T-Beam non-matching coupling: (a) Description of the problem: the interface cuts the upper patch at the middle of some elements. (b) Deformed configuration with scale factor of 10 and the magnitude of the displacement field. (c) Angle between the patches along the interface after deformation. Coarse case corresponds to the mesh depicted in (b). Fine mesh is three times refined.

to clarify the stability of the method [165]. However, these oscillations vanish with a finer parametrization which is satisfying.

Boundary conditions The coupling operators can be useful to impose boundary conditions, especially for the rotation along the edges of a patch. In fact, since the Kirchhoff–Love shell formulation does not have rotational DOFs, symmetric boundary conditions cannot be imposed in a direct manner in general (*e.g.* eliminating some DOFs).

As an example, one can take the case of the tube with a square cross-section as depicted in figure 4.9. Due to the symmetry of the problem, it is possible to consider only one quarter of the tube, and thus to model the tube with a single patch. To that purpose, additional boundary conditions have to be imposed as depicted in figure 4.9. Along two edges of the patch, the rotation in the z-direction has to be fixed. One can used the coupling matrix $\mathbf{Z}$ and solve the following linear system:

$$\begin{bmatrix} \mathbf{K} & \mathbf{Z}^T \\ \mathbf{Z} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ \hat{\boldsymbol{\lambda}}_{\mathrm{BC}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ 0 \end{pmatrix}. \tag{4.27}$$

Here, we force the other displacement constraints ($u_x = 0$ and $u_y = 0$ as mentioned on figure 4.9) strongly but one could also use the Mortar operator $\mathbf{C}$ introduced previously

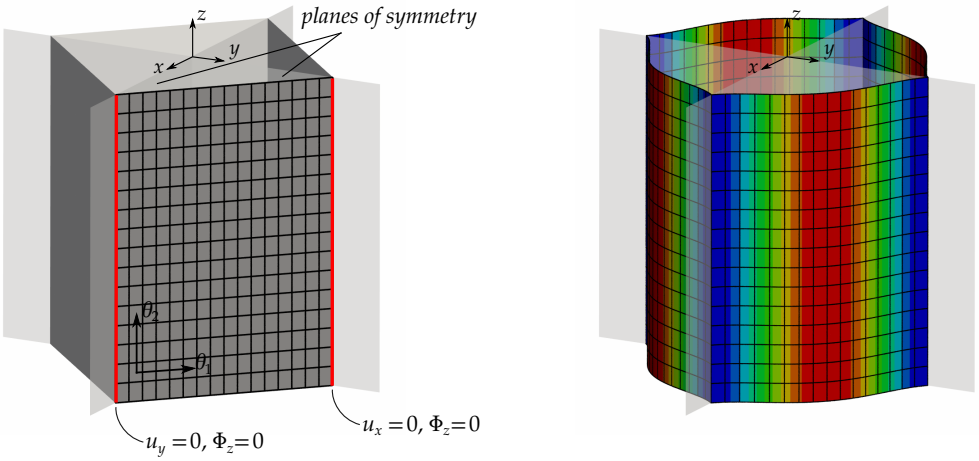


Figure 4.9 – Single patch model of a square shape tube under internal pressure: the constraints for the rotation on the patch edges are imposed through the Mortar coupling operators. Looking at the deformed configuration shows that the angles between the patch and the planes of symmetry are kept fixed (right side, color map corresponds to the displacement magnitude).

for the displacement continuity (see equation 4.20). Figure 4.9(right) displays the deformed configuration of the tube and highlights the good imposition of the boundary conditions. Finally, one can look at the expression of the rotation vector for this present example of the tube. With the parametric directions as given in figure 4.9, the tangent component of the rotation vector reads as follows:

$$\Phi \cdot \boldsymbol{t} = \alpha \, u_{3,\theta_1}$$

where α is a constant and u_3 is the displacement component normal to the shell. In other words, we impose at the patch edges the kinematic condition: $u_{3,\theta_1} = 0$. Similar conditions can be found in the literature, for instance in Schöllhammer et al. [183].

The rotation coupling matrix $\mathbf{Z}$ can also be used for imposing a bending moment at the edge of a shell. As an example, we consider the case of a beam with one fixed end

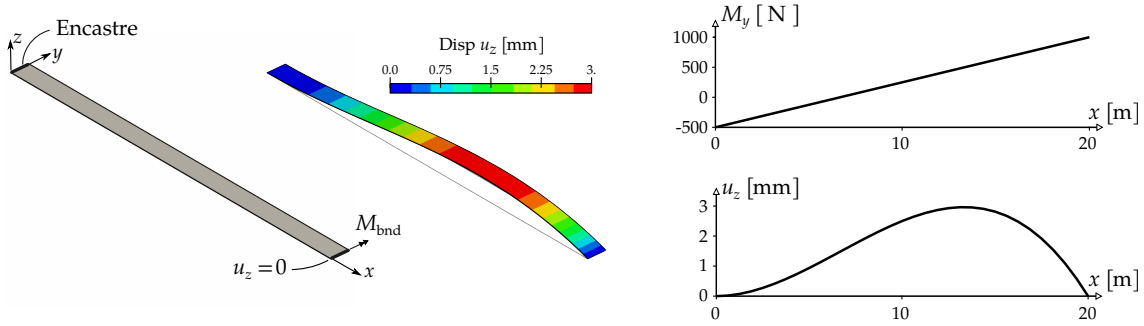


Figure 4.10 – Beam with one end fixed and a bending moment applied on the other edge. The bending moment can be applied using the Mortar operator $\mathbf{Z}$.

and with a moment applied to the other end as depicted in figure 4.10. The geometric parameters of the beam are the same as the cantilever beam with end load described in figure 4.7. We impose a bending moment (per length unit) equals to $M_{\text{bnd}} = 1000\,\text{N}$. To this purpose, we solve the following system:

$$\mathbf{K}\boldsymbol{u} = \mathbf{Z}^T \hat{\boldsymbol{\lambda}}_{\text{bnd}}, \tag{4.28}$$

where all the DOFs associated to the Lagrange multiplier $\hat{\boldsymbol{\lambda}}_{\text{bnd}}$ are equal to M_{bnd}. Based on the Euler–Bernoulli beam theory, one can express the vertical displacement and the bending moment along the beam. The bending moment is a linear function of the beam position, and the displacement is of cubic order (see figure 4.10, right):

$$M_y = \frac{M_{\text{bnd}}}{2L}\,(3x - L)\,, \qquad u_z(x) = \frac{w\,M_{\text{bnd}}}{4EIL}\,(-x^3 + x^2 L)\,,$$

where $I = w\,t^3/12$ denotes the second moment of inertia. Thus, a patch with a single cubic element is sufficient to recover the analytical results. This has been verified numerically (see figure 4.10, right).

4.3 A dual Domain Decomposition algorithm

Henceforth, we address the general case of n_s sub-domains (*i.e.* n_s patches) $\Omega^{(s)}$, $s = 1 \dots n_s$. We define as $\mathbf{K} = \text{diag}(\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \dots, \mathbf{K}^{(n_s)})$ the block diagonal matrix that stores all the stiffness matrices of each sub-domain. We define the global load vector $\boldsymbol{F} = (\boldsymbol{F}^{(1)T}, \boldsymbol{F}^{(2)T}, \dots, \boldsymbol{F}^{(n_s)T})^T$ as well. The total displacement and Lagrange multiplier DOFs are stored in column vectors $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ respectively. We concatenate the displacement and the rotation Mortar coupling matrices into single coupling matrices $\mathbf{B}^{(s)}$ such that:

$$\mathbf{B}^{(s)} = \left[\mathbf{C}^{(s)T},\ \mathbf{Z}^{(s)T}\right]^T. \tag{4.29}$$

The matrices $\mathbf{B}^{(s)}$ are of size $N_\lambda \times N_u^{(s)}$, where N_λ denotes the size of $\boldsymbol{\lambda}$ and $N_u^{(s)}$ denotes the size of $\boldsymbol{u}^{(s)}$. Thus, we introduce the global coupling operator $\mathbf{B} = [\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \dots, \mathbf{B}^{(n_s)}]$. With these notations, the generalization of the coupled system (4.16) reads:

$$\begin{bmatrix} \mathbf{K} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F} \\ 0 \end{pmatrix}. \tag{4.30}$$

We are now looking for an efficient algorithm to solve the coupled linear system (4.30) based on the Domain Decomposition Methods, as motivated previously (see section 4.1.1.2). The proposed solver belongs to the family of dual Domain Decomposition algorithms, originally introduced by Farhat et al. [152] with the one-level FETI algorithm. The following developments are based on these researches and provide an efficient algorithm for the parallel computation of non-conforming multipatch Kirchhoff-Love shells.

4.3.1 The one-level FETI problem

The first step of the approach consists in splitting problem (4.30) into the following coupled set of equations:

$$\mathbf{K}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda}, \quad \text{for } s = 1, \dots, n_s \tag{4.31}$$

$$\sum_{s=1}^{n_s} \mathbf{B}^{(s)}\boldsymbol{u}^{(s)} = 0. \tag{4.32}$$

The goal is then to formulate a dual interface problem with the Lagrange multiplier DOF as the only unknowns. This interface problem is obtained by introducing the local equilibriums (4.31) into the coupling condition (4.32).

Equilibrium of floating sub-domains Generally, the local stiffness matrices $\mathbf{K}^{(s)}$ are not invertible. Indeed, if no Dirichlet boundary conditions are imposed on the floating sub-domain $\Omega^{(s)}$ then corresponding matrix $\mathbf{K}^{(s)}$ is singular. Therefore, $\mathbf{K}^{(s)}$ being symmetric, equation (4.31) has a solution if and only if the load vector $\boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda}$ belongs to the image of operator $\mathbf{K}^{(s)}$. It leads to an additional equation, called the admissibility condition, which is:

$$\mathbf{R}^{(s)T}\left(\boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda}\right) = 0, \quad \text{for } s = 1, \dots, n_s \tag{4.33}$$

where $\mathbf{R}^{(s)}$ is a rectangular matrix. Its columns describe a basis of the null space $\ker(\mathbf{K}^{(s)})$ of matrix $\mathbf{K}^{(s)}$. From a mechanical point of view, these vectors are the local rigid body modes. Then, for each sub-domain, the equilibrium is given by:

$$\boldsymbol{u}^{(s)} = \mathbf{K}^{\dagger(s)}\left(\boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda}\right) + \mathbf{R}^{(s)}\boldsymbol{\alpha}^{(s)}, \tag{4.34}$$

where $\boldsymbol{\alpha}^{(s)}$ is a vector that collects the different amplitudes of the rigid body modes (or null-energy modes) for sub-domain s and $\mathbf{K}^{\dagger(s)}$ is a pseudo-inverse of the stiffness matrix, *i.e.* that satisfies $\mathbf{K}^{(s)}\mathbf{K}^{\dagger(s)}\mathbf{K}^{(s)} = \mathbf{K}^{(s)}$. If the stiffness matrix is non-singular, then the pseudo-inverse is unique and equal the inverse of matrix $\mathbf{K}^{(s)}$, *i.e.* $\mathbf{K}^{\dagger(s)} = \mathbf{K}^{(s)-1}$.

Substitution Now, the local solutions (4.34) can be introduced into the compatibility condition (4.32). It yields the following equation:

$$\sum_{s=1}^{n_s} \mathbf{B}^{(s)}\mathbf{K}^{\dagger(s)}\mathbf{B}^{(s)T}\,\boldsymbol{\lambda} \;-\; \sum_{s=1}^{n_s} \mathbf{B}^{(s)}\mathbf{R}^{(s)}\,\boldsymbol{\alpha}^{(s)} \;=\; \sum_{s=1}^{n_s} \mathbf{B}^{(s)}\mathbf{K}^{\dagger(s)}\boldsymbol{F}^{(s)}. \tag{4.35}$$

Finally, combining this last equation with the admissibility condition given by (4.33), we end up with the interface problem. The resulting system is generally called the FETI system and is given by:

$$\begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{pmatrix}, \tag{4.36}$$

where $\boldsymbol{\alpha}$ concatenates the rigid body unknowns of each sub-domain. The dual Schur complement $\mathbf{F}$, the constraint matrix $\mathbf{G}$, and the right-hand sides $\boldsymbol{d}$ and $\boldsymbol{e}$ are expressed as:

$$\begin{aligned}
\mathbf{F} &= \sum_{s=1}^{n_s} \mathbf{B}^{(s)} \mathbf{K}^{\dagger(s)} \mathbf{B}^{(s)T}, \\
\mathbf{G} &= -\begin{bmatrix} \mathbf{B}^{(1)}\mathbf{R}^{(1)} & \mathbf{B}^{(2)}\mathbf{R}^{(2)} & \cdots & \mathbf{B}^{(n_s)}\mathbf{R}^{(n_s)} \end{bmatrix}, \\
\boldsymbol{d} &= \sum_{s=1}^{n_s} \mathbf{B}^{(s)} \mathbf{K}^{\dagger(s)} \boldsymbol{F}^{(s)}, \\
\boldsymbol{e} &= -\begin{pmatrix} \boldsymbol{F}^{(1)T}\mathbf{R}^{(1)} & \boldsymbol{F}^{(2)T}\mathbf{R}^{(2)} & \cdots & \boldsymbol{F}^{(n_s)T}\mathbf{R}^{(n_s)} \end{pmatrix}^T.
\end{aligned}$$

4.3.2 Solving the interface problem

The resolution of system (4.36) relies on an iterative solver where only matrix-vector products are performed. This way, there is no need for assembling the dual Schur complement which would be computationally very demanding since pseudo-inverses $\mathbf{K}^{\dagger(s)}$ are involved. Moreover, it avoids to inverse the dual Schur complement which has a dense structure. More precisely, system (4.36) is solved in the following using a Preconditioned Conjugate Projected Gradient algorithm. The idea is to iteratively solve a symmetric positive definite linear system (involving $\mathbf{F}$) under the constraint (4.33).

Projection In order to do so, an additional projection step enables to eliminate the DOF associated to the rigid body motions of the floating sub-domains. More specifically, a projector $\mathbf{P}$ onto the null space of $\mathbf{G}^T$, *i.e.* onto $\ker(\mathbf{G}^T)$, is introduced to satisfy (4.33). Thus, by definition, we have the relations:

$$\mathbf{G}^T\mathbf{P} = 0, \quad \text{and} \quad \mathbf{P}^T\mathbf{G} = 0.$$

It allows to eliminate the unknowns $\boldsymbol{\alpha}$ from the first set of equations of system (4.36). To this end, we multiply these equations to the left by $\mathbf{P}^T$:

$$\mathbf{P}^T\Big[\quad \mathbf{F}\boldsymbol{\lambda} + \mathbf{G}\boldsymbol{\alpha} = \boldsymbol{d} \quad \Big] \quad \Rightarrow \quad \mathbf{P}^T\mathbf{F}\boldsymbol{\lambda} = \mathbf{P}^T\boldsymbol{d}. \tag{4.37}$$

The second equation $\mathbf{G}^T\boldsymbol{\lambda} = \boldsymbol{e}$ of system (4.36) is explicitly fulfilled by introducing the splitting:

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^0 + \mathbf{P}\bar{\boldsymbol{\lambda}}, \tag{4.38}$$

where the initialization $\boldsymbol{\lambda}^0$ satisfies the constraint $\mathbf{G}^T\boldsymbol{\lambda}^0 = \boldsymbol{e}$. This splitting is operated during the resolution by firstly choosing $\boldsymbol{\lambda}^0$ as the starting guess, and secondly by projecting the gradient during the update. Finally, these two projection steps transform the interface problem (4.36) into the projected interface problem:

$$\mathbf{P}^T\mathbf{F}\mathbf{P}\,\bar{\boldsymbol{\lambda}} = \mathbf{P}^T\left(\boldsymbol{d} - \mathbf{F}\boldsymbol{\lambda}^0\right), \tag{4.39}$$

Initialize			
$\boldsymbol{\lambda}^0$	$=$	$\mathbf{Q}\mathbf{G}(\mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}\boldsymbol{e}$	
$\boldsymbol{r}^0$	$=$	$\boldsymbol{d}-\mathbf{F}\boldsymbol{\lambda}^0$	
Iterate $k=1,2,\ldots$ until convergence			
$\boldsymbol{w}^{k-1}$	$=$	$\mathbf{P}^T\boldsymbol{r}^{k-1}$	Project
$\boldsymbol{z}^{k-1}$	$=$	$\mathbf{M}\boldsymbol{w}^{k-1}$	Precondition
$\boldsymbol{y}^{k-1}$	$=$	$\mathbf{P}\boldsymbol{z}^{k-1}$	Re-project
$\boldsymbol{p}^k$	$=$	$\boldsymbol{y}^{k-1}-\sum_{i=1}^{k-1}\beta_i\boldsymbol{p}^i \quad (\boldsymbol{p}^1=\boldsymbol{y}^0)$	Conjugate
with β_i	$=$	$\dfrac{\boldsymbol{y}^{k-1}\cdot\mathbf{F}\boldsymbol{p}^i}{\boldsymbol{p}^i\cdot\mathbf{F}\boldsymbol{p}^i}$	
α_k	$=$	$\dfrac{\boldsymbol{y}^{k-1}\cdot\boldsymbol{w}^{k-1}}{\boldsymbol{p}^k\cdot\mathbf{F}\boldsymbol{p}^k}$	Minimize
$\boldsymbol{\lambda}^k$	$=$	$\boldsymbol{\lambda}^{k-1}+\alpha_k\boldsymbol{p}^k$	Update
$\boldsymbol{r}^k$	$=$	$\boldsymbol{r}^{k-1}-\alpha_k\mathbf{F}\boldsymbol{p}^k$	Residual

Table 4.1 – The Preconditioned Conjugate Projected Gradient algorithm with reorthogonalization. Operators **P** and **M** denote the projector and the preconditioner, respectively.

which is solved using the PCPG algorithm as described in table 4.1.

We still need to express the projector and the initialization. It brings into play a symmetric matrix denoted $\mathbf{Q}$ for which the product $\mathbf{G}^T\mathbf{Q}\mathbf{G}$ is invertible. Matrix $\mathbf{Q}$ can be taken as being the preconditioner, identity or a scaling matrix [184]. Finally, we have:

$$\boldsymbol{\lambda}^0 = \mathbf{Q}\mathbf{G}\left(\mathbf{G}^T\mathbf{Q}\mathbf{G}\right)^{-1}\boldsymbol{e}, \tag{4.40}$$

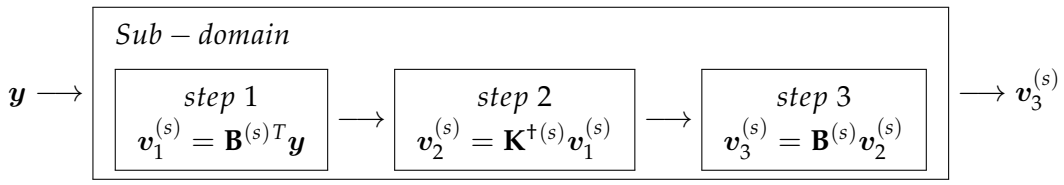
$$\mathbf{P} = \mathbf{I}-\mathbf{Q}\mathbf{G}\left(\mathbf{G}^T\mathbf{Q}\mathbf{G}\right)^{-1}\mathbf{G}^T. \tag{4.41}$$

For most problems, the simplest choice $\mathbf{Q}=\mathbf{I}$ is the most computationally efficient, and many studies reported in the literature have been performed with this choice [184, 185]. We apply the same treatment herein. Finally, full or partial re-orthogonalization of the gradient at every iteration is commonly adopted to overcome the poor conditioning of the dual Schur complement [150]. In this work, a full re-orthogonalization is performed.

Note: parallel computing A crucial feature of this algorithm is that it fits well with parallel computing. This is not surprising as it was originally intended for that particular purpose [152]. To understand this point, one can observe what happens during the product between a vector $\boldsymbol{y}$ and the dual Schur complement $\mathbf{F}$:

$$\mathbf{F}\boldsymbol{y} = \left[\sum_s \mathbf{B}^{(s)}\mathbf{K}^{\dagger(s)}\mathbf{B}^{(s)T}\right]\boldsymbol{y} = \sum_s\left(\mathbf{B}^{(s)}\mathbf{K}^{\dagger(s)}\mathbf{B}^{(s)T}\boldsymbol{y}\right). \tag{4.42}$$

It results in a sum of local contributions that can be computed independently. Each local contribution involves the local pseudo-inverse $\mathbf{K}^{\dagger(s)}$ and the coupling matrix $\mathbf{B}^{(s)}$ which remain unchanged during the iterative resolution, and thus are build once and for all at the beginning. Each local contribution is computed into three steps as follows:



Thus, a sub-domain can be seen as a black box: tacking as an input a vector $\boldsymbol{y}$, it returns the local contribution ${\boldsymbol{v}_3}^{(s)}$. Summing all these local contributions give the desired product $\mathbf{F}\boldsymbol{y}$. The most time consuming step is the product by the pseudo-inverse (*i.e. step 2*). As discussed in the next section, the pseudo-inverse is generally given as a factorization (as a classical inverse). As a result, this step can be seen as the resolution of a linear system where the matrix is already factorized. Finally, let us notice that a sub-domain does not need to know all the Lagrange multiplier DOFs but only those coming from the Lagrange multipliers that are actually acting on this sub-domain. Thus, a vector $\boldsymbol{y}$ is not communicated completely to each sub-domain, but only the active part $\boldsymbol{y}^{(s)}$ is required, which is extracted using trace operators. For example, taking the sub-domain $\Omega^{(1)}$ from the case depicted at figure 4.11, we understand that only the DOFs coming from the Lagrange multiplier $\boldsymbol{\lambda}_{12}$ is required. It also means that in practice the coupling matrix $\mathbf{B}^{(s)}$ does not have a number of row equals to the total number of Lagrange multiplier DOF, but only equals to those related to the considered sub-domain.

Unlike the dual Schur complement, there is however a step which occurs during the projection that can apparently not be done in parallel (see the algorithm described in table 4.1). Indeed, in the expression of the projector (4.41), there is the so-called *coarse problem* which is the term $(\mathbf{G}^T\mathbf{Q}\mathbf{G})^{-1}$. This coarse problem takes the rigid body DOF as unknowns. It is therefor of low dimension and easy to be factorized before entering into the gradient loop. It can be seen as a coarse correction that plays a particular role in the scalability of the algorithm. Finally, we do not go into more details regarding this issue of parallel computing of the algorithm. Interesting readers can find a large amount of information in the literature since DDMs have been intensely developed during the past

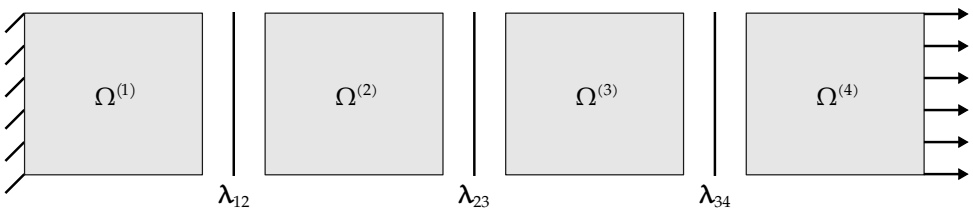
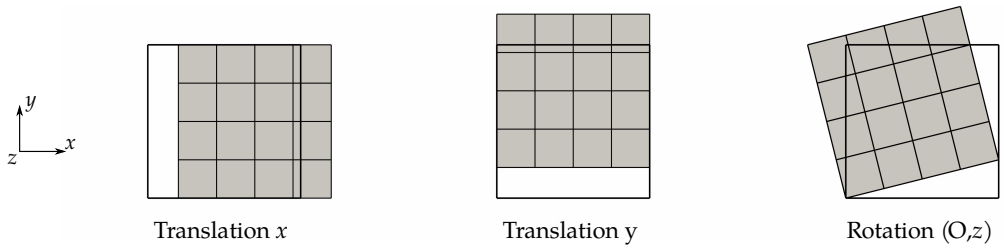


Figure 4.11 – Parallel computing: the communication between the sub-domains is done through the Lagrange multipliers.

(a) Rigid body motions (2D case)



(b) Rigid body motions (3D case)

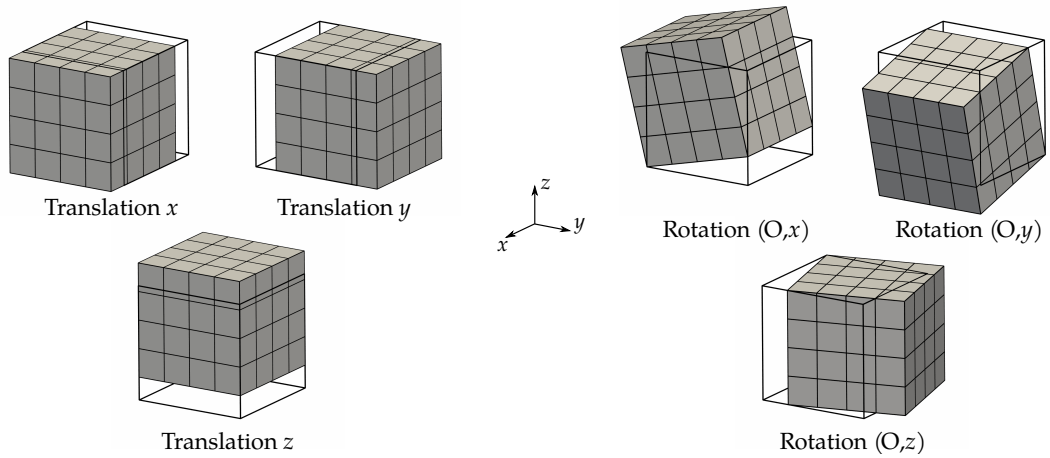


Figure 4.12 – Rigid body motions

decades (see, for example, Jolivet [186] and Kozubek et al. [187]).

4.3.3 Null-space and Pseudo-inverse

Up to now, we did not mention how are formulated the null space $\mathbf{R}^{(s)}$ and the pseudo-inverse $\mathbf{K}^{\dagger(s)}$ that come into play in the one-level FETI solver. There are different possibilities but here we resort to a purely algebraic method because of the embedded Kirchhoff–Love shell formulation which will be introduced in the next chapter 5. This shell formulation has the particularity of being non-isoparametric which involves potential difficulties to build the null space, as discussed in the following.

4.3.3.1 Explicit construction of the kernel

Let us firstly deal with the column matrix $\mathbf{R}^{(s)}$. The null space of a stiffness matrix is commonly associated to the space of rigid-body modes. For 3D elasticity problem and under the condition that no Dirichlet boundary conditions are imposed, we know that there exist 6 rigid-body motions consisting of 3 translations (modes numbered 1, 2, and 3) and 3 rotations (modes numbered 4, 5, and 6) around 3 non-collinear axes. Figure 4.12

presents these rigid-body modes for a simple geometry.

Considering a shell with mid-surface $\boldsymbol{S}(\theta_1, \theta_2)$, an arbitrary kernel element $\mathbf{r}$ is :

$$\mathbf{r}(\theta_1, \theta_2) = \mathbf{a} + \mathbf{b} \times \boldsymbol{S}(\theta_1, \theta_2), \tag{4.43}$$

where $\mathbf{a} \in \mathbb{R}^3$ defines a translation and $\mathbf{b} \in \mathbb{R}^3$ defines a rotation. In order to build a basis of the null space, one can successively describe the translations and the rotations with respect to the global unit vectors $\mathbf{e}_i$. We can re-write equation (4.43) in the discrete form in order to identify the DOF corresponding to these unitary rigid-body modes:

$$\sum_k R_k(\theta_1, \theta_2) \boldsymbol{r}_k^i = \mathbf{e}_i, \qquad \text{and} \qquad \sum_k R_k(\theta_1, \theta_2) \boldsymbol{r}_k^{i+3} = \mathbf{e}_i \times \boldsymbol{S}(\theta_1, \theta_2), \tag{4.44}$$

where R_k are the NURBS basis functions of the discretized displacement field. It is obvious that taking all DOF $\boldsymbol{r}_k^i$ equal to $\mathbf{e}_i$ enables to describe the translation rigid-body modes. For the rotations, it depends on the discretization of the shell mid-surface:

— in the case of an isoparametric shell formulation, the displacement field is approximated using the same discretization as for the mid-surface which gives:

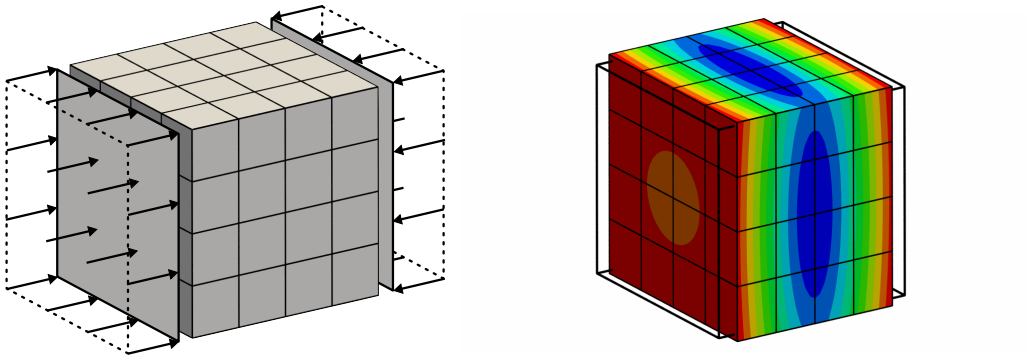
$$\sum_k R_k(\theta_1, \theta_2) \boldsymbol{r}_k^{i+3} = \sum_k R_k(\theta_1, \theta_2)\, \mathbf{e}_i \times \mathbf{P}_k \quad \Rightarrow \quad \boldsymbol{r}_k^{i+3} = \mathbf{e}_i \times \mathbf{P}_k. \tag{4.45}$$

— in the case of a non-isoparametric shell formulation, the displacement field and the mid-surface are discretized differently (R_k for the displacement and M_l for the mid-surface), leading to the linear system of equations:

$$\sum_k R_k(\theta_1, \theta_2) \boldsymbol{r}_k^{i+3} = \sum_l M_l(\theta_1, \theta_2)\, \mathbf{e}_i \times \mathbf{P}_l, \quad \forall\, (\theta_1, \theta_2). \tag{4.46}$$

System (4.46) cannot be solved in general. One can choose a set of parameters $\{(\theta_1, \theta_2)^1, (\theta_1, \theta_2)^2, \ldots\}$ and build an approximation of the null space by interpolation for instance. However, it does not ensure that equation (4.46) is exactly satisfied (*i.e.* true for every couple of parameters (θ_1, θ_2)) which means that this approximation of the null space may not be energy-free. Thus, we cannot explicitly build the kernel of the stiffness matrix in case of a non-isoparametric shell formulation.

In a first attempt, we studied the expression of the null space in the case of a floating sub-domain without any Dirichlet boundary conditions, because we were interested in the approach presented by Dostál et al. [188] called the Total-FETI. This method partly aims at simplifying the implementation of the original FETI method since the same treatment for each sub-domain is performed. Within this approach, the column matrix $\mathbf{R}^{(s)}$ is always taken as the explicit description of the kernel of the all floating sub-domains. Then, they suggest to build the pseudo-inverse by a regularization of the semi-definite stiffness matrix which can be done again through the use of the rigid body modes (see [187, 189] for more details). Because it is not possible to build explicitly the null space for a non-isoparametric shell formulation, the Total-FETI approach does not seem to be applicable in this context. Consequently, we resort to a purely algebraic method to build the null spaces and the pseudo-inverses involved in the algorithm.



Cube in compression

Deformed configuration
(Color: Displacement magn.)

Figure 4.13 – Pure neumann problem: cube in compression with Poisson's ratio $\nu = 0.30$.

4.3.3.2 Algebraic method

For clarity, we voluntarily skip here the sub-domain identification through superscript (s), but the reader should view the following operators as local quantities. The purely algebraic method used in this work has been presented in Farhat et al. [190] and already used in Farhat et al. [152]. The pseudo-inverse and the null space are constructed during the factorization of the stiffness matrix. It is based on the partition of matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{12}{}^T & \mathbf{K}_{22} \end{bmatrix}, \tag{4.47}$$

where $\mathbf{K}_{11} \in \mathbb{R}^{r \times r}$ is a submatrix with full rank r equal to the rank of the initial matrix (*i.e.* $r = \operatorname{rk} \mathbf{K} = \operatorname{rk} \mathbf{K}_{11}$). With such a partition in hand, the following relation holds:

$$\mathbf{K}_{22} - \mathbf{K}_{12}{}^T \mathbf{K}_{11}^{-1} \mathbf{K}_{12} = 0. \tag{4.48}$$

One can define the pseudo-inverse and the null space as:

$$\mathbf{K}^{\dagger} = \begin{bmatrix} \mathbf{K}_{11}^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \qquad \mathbf{R} = \begin{bmatrix} -\mathbf{K}_{11}^{-1} \mathbf{K}_{12} \\ \mathbf{I} \end{bmatrix}, \tag{4.49}$$

where $\mathbf{I}$ is the identity matrix of size $m = n - r$.

In practice, partition (4.47) is not known in advance but it is obtained in an unordered form during a Gaussian-based factorization (as for example a LU decomposition). At each step of the factorization, one can check if a null pivot is encountered. If true, the corresponding equation is redundant and is removed from the system. The reduced column can be recycled to later recover the null space. Furthermore, note that a small tolerance needs to be set to detect the null pivots (*i.e.* the right term in (4.48) is, in practice, not exactly equal to zero). It enables for instance to track the rotation rigid-body modes in the case of the embedded Kirchhoff–Love shell (see next chapter 5 for the

details). More information regarding the algebraic approach can be found in Farhat et al. [190]. Alternative methods, based for example on the Singular Value Decomposition (SVD) are also available [191]. Pure Neumann problems as for instance the cube in uni-axial compression presented in figure 4.13 typically resort to pseudo-inverses of semi-definite system. If required, several works can be found in the literature that deal with this type of singular problems (*e.g.* [189, 192]).

4.3.4 Preconditioning

Last but not least ingredient involved in the FETI algorithm is the preconditioner already introduced in table 4.1 through operator $\mathbf{M}$. Its goal is to reduce the condition number of the linear problem to be solved. Since the rate of convergence of iterative solvers depends on the condition number of the system, they are usually combined with a preconditioner. Moreover, a good preconditioner is crucial for the computational performance of the iterative resolution. The preconditioner can be seen as an approximation of the inverse of the system. The better the approximation, the more efficient is the preconditioning step during the iterative resolution. However, a good preconditioner can be computationally expensive to be built and evaluated, which can override the positive effect of faster iterative convergence.

Dirichlet preconditioner Several preconditioners of the dual Schur complement have been studied in the literature. In the context of FEM matching grids, the Dirichlet preconditioner is said to be optimal regarding the asymptotic bound and thus offers an excellent numerical scalability [193]. The Dirichlet preconditioner is obtained under the approximation that the inverse of the sum of the local contributions is the sum of the inverses of each of these local contributions. More precisely, it is obtained by exactly inverting the local dual Schur complements and summing each local contribution. In case of Mortar interface, we can re-write this Dirichlet preconditioner as:

$$\tilde{\mathbf{F}}_I^{-1} = \sum_{s=1}^{n_s} \left(\mathbf{B}^{(s)}\mathbf{B}^{(s)T}\right)^{-1} \mathbf{B}^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_{bb}^{(s)} \end{bmatrix} \mathbf{B}^{(s)T} \left(\mathbf{B}^{(s)}\mathbf{B}^{(s)T}\right)^{-1}, \tag{4.50}$$

where matrix $\mathbf{S}_{bb}^{(s)}$ denotes the primal Schur complement and is given by:

$$\mathbf{S}_{bb}^{(s)} = \mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi}^{(s)} {\mathbf{K}_{ii}^{(s)}}^{-1} \mathbf{K}_{ib}^{(s)}, \tag{4.51}$$

where indices b correspond to the boundary DOF on the interface and indices i correspond to the internal DOF.

Proof. *Let us prove that expression (4.50) corresponds to the sum of the inverses of the local dual Schur complements. To this purpose, we can re-write the local dual Schur complement* $\mathbf{F}^{(s)}$ *by*

using the splitting of the boundary DOF and the interface DOF. It reads as:

$$\mathbf{F}^{(s)} = \begin{bmatrix} 0 & \mathbf{B}_b^{(s)} \end{bmatrix} \begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{B}_b^{(s)T} \end{bmatrix} \tag{4.52}$$

Next, we introduce the following quantities $\mathbf{C}_i^{(s)}$ *and* $\mathbf{C}_b^{(s)}$ *such that:*

$$\begin{bmatrix} \mathbf{C}_i^{(s)} \\ \mathbf{C}_b^{(s)} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{ii}^{(s)} & \mathbf{K}_{ib}^{(s)} \\ \mathbf{K}_{bi}^{(s)} & \mathbf{K}_{bb}^{(s)} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{B}_b^{(s)T} \end{bmatrix}. \tag{4.53}$$

Thus, we can re-write the equation (4.52) by using the notation from equation (4.53):

$$\mathbf{F}^{(s)} = \begin{bmatrix} 0 & \mathbf{B}_b^{(s)} \end{bmatrix} \begin{bmatrix} \mathbf{C}_i^{(s)} \\ \mathbf{C}_b^{(s)} \end{bmatrix} = \mathbf{B}_b^{(s)} \mathbf{C}_b^{(s)} \quad \Rightarrow \quad \mathbf{C}_b^{(s)} = \mathbf{B}_b^{(s)T} \left(\mathbf{B}_b^{(s)} \mathbf{B}_b^{(s)T} \right)^{-1} \mathbf{F}^{(s)}. \tag{4.54}$$

One can multiply equation (4.53) to the left by the stiffness matrix $\mathbf{K}^{(s)}$*. It results the following equations:*

$$\mathbf{K}_{ii}^{(s)} \mathbf{C}_i^{(s)} + \mathbf{K}_{ib}^{(s)} \mathbf{C}_b^{(s)} = 0, \tag{4.55}$$

$$\mathbf{K}_{bi}^{(s)} \mathbf{C}_i^{(s)} + \mathbf{K}_{bb}^{(s)} \mathbf{C}_b^{(s)} = \mathbf{B}_b^{(s)T}. \tag{4.56}$$

Now, we substitute equation (4.55) into equation (4.56). The local primal Schur complement as previously defined in equation (4.51) appears:

$$\left[\mathbf{K}_{bb}^{(s)} - \mathbf{K}_{bi} \mathbf{K}_{ii}^{(s)^{-1}} \mathbf{K}_{ib}^{(s)} \right] \mathbf{C}_b^{(s)} = \mathbf{B}_b^{(s)T} \quad \Rightarrow \quad \mathbf{S}_{bb}^{(s)} \mathbf{C}_b^{(s)} = \mathbf{B}_b^{(s)T} \tag{4.57}$$

Introducing the result from equation (4.54) into this equation (4.57) gives:

$$\mathbf{S}_{bb}^{(s)} \mathbf{B}_b^{(s)T} \left(\mathbf{B}_b^{(s)} \mathbf{B}_b^{(s)T} \right)^{-1} \mathbf{F}^{(s)} = \mathbf{B}_b^{(s)T} \tag{4.58}$$

Finally, we multiply this last equation by the left by the term $(\mathbf{B}_b{}^{(s)} \mathbf{B}_b{}^{(s)T})^{-1} \mathbf{B}_b{}^{(s)}$*. We get the final result:*

$$\left(\mathbf{B}_b^{(s)} \mathbf{B}_b^{(s)T} \right)^{-1} \mathbf{B}_b^{(s)} \mathbf{S}_{bb}^{(s)} \mathbf{B}_b^{(s)T} \left(\mathbf{B}_b^{(s)} \mathbf{B}_b^{(s)T} \right)^{-1} \mathbf{F}^{(s)} = \mathbf{I}_b \tag{4.59}$$

We find the expression presented in equation (4.50).

Generalized preconditioner More specifically, some works deal with the particular case of Mortar coupling and improvements to the initial version of the Dirichlet preconditioner have been studied in this context [161, 185, 193, 194]. For instance, scaling factors can be added in order to take into account the possible gap between the stiffnesses on both sides of the interface [195, 196]. These heterogeneities are particularly numerous in the case of stiffened structures with the presence of T-junctions, different shell thicknesses and material behaviors [156]. Thus, it is primordial to efficiently deal

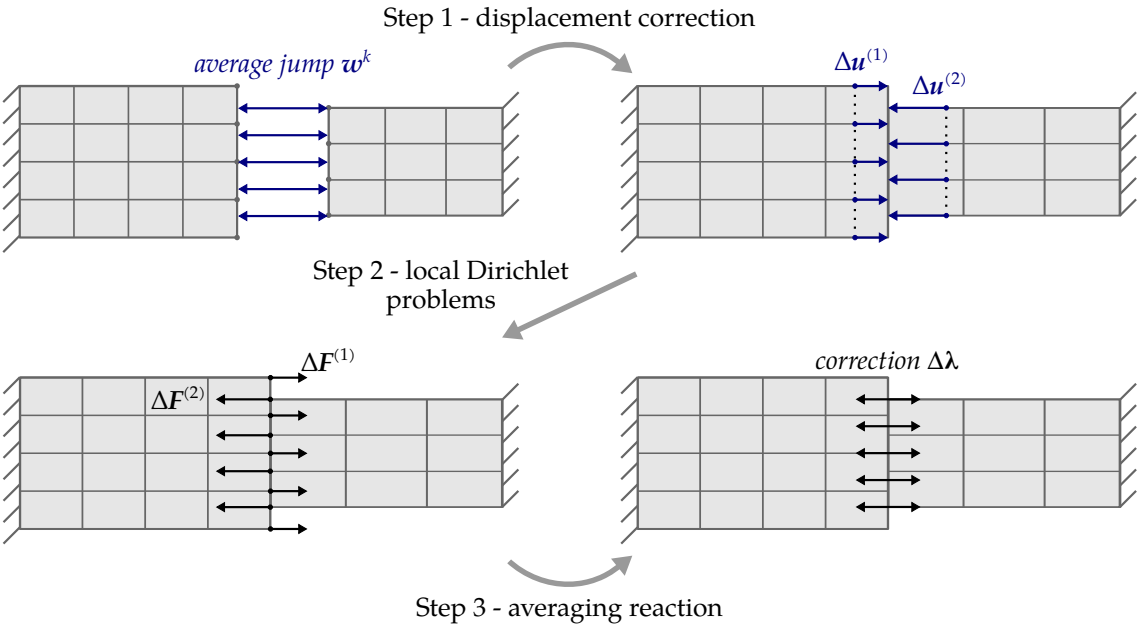


Figure 4.14 – Illustration of the preconditioning step during the iterative resolution of the dual Domain Decomposition solver (adapted from Gosselet et al. [150]).

with those heterogeneities to analyze complex shell structures. Therefore, in the present paper, it is proposed to follow the extended Dirichlet preconditioner suggested by Rixen [193] and mentioned by Stefanica [185] which can be written as:

$$\tilde{\mathbf{F}}_D^{-1} = \sum_{s=1}^{n_s} \left(\mathbf{B}\mathbf{D}\mathbf{B}^T\right)^{-1} \mathbf{B}^{(s)}\mathbf{D}^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_{bb}^{(s)} \end{bmatrix} \mathbf{D}^{(s)}\mathbf{B}^{(s)T} \left(\mathbf{B}\mathbf{D}\mathbf{B}^T\right)^{-1}, \tag{4.60}$$

where matrix $\mathbf{D}^{(s)}$ is chosen here as the superlumped scaling: $\mathbf{D}^{(s)} = \left(\operatorname{diag}\mathbf{K}^{(s)}\right)^{-1}$. The global matrix $\mathbf{D}$ is the diagonal block assembly of all these local scaling matrices, which leads to a diagonal matrix in the present case.

The extended Dirichlet preconditioner has shown better performance in comparison with the original Dirichlet preconditioner in the case of Mortar interfaces [161, 185, 194]. Rixen [193] shows that this preconditioner is mechanically consistent and mentions its mechanical meaning: once applied to the residual of the interface problem (4.32), it computes the correction for the Lagrange multipliers. Figure 4.14 highlights this point. On can split the preconditioner into three steps to understand its mechanical meaning:

1. *Make the interface compatibility* (4.32) *satisfied.* Let $\boldsymbol{u}^k$ and $\boldsymbol{w}^k$ be respectively the displacement and the residual at iteration k (*i.e.* $\mathbf{B}\boldsymbol{u}^k = \boldsymbol{w}^k$). We are looking for the displacement correction $\Delta\boldsymbol{u}$ such that:

$$\mathbf{B}(\boldsymbol{u}^k - \Delta\boldsymbol{u}) = 0, \quad \text{which gives} \quad \mathbf{B}\Delta\boldsymbol{u} = \boldsymbol{w}^k. \tag{4.61}$$

Thus we can build the correction, in a least-square sense, as:

$$\Delta \boldsymbol{u} = \mathbf{D}\mathbf{B}^T \left(\mathbf{B}\mathbf{D}\mathbf{B}^T\right)^{-1} \boldsymbol{w}^k, \tag{4.62}$$

where matrix $\mathbf{D}$ is invertible.

2. *Compute the interface efforts corresponding to the jump correction.* It is done at the sub-domain level through the local primal Schur complements $\mathbf{S}^{(s)}$:

$$\Delta \boldsymbol{F}^{(s)} = \mathbf{S}^{(s)} \Delta \boldsymbol{u}^{(s)}. \tag{4.63}$$

3. *Make the Lagrange multipliers represent these interface efforts.* We are looking for the correction $\Delta\boldsymbol{\lambda}$ such that:

$$\mathbf{B}^T \Delta\boldsymbol{\lambda} = \Delta \boldsymbol{F}. \tag{4.64}$$

As for step 1, this equation can be satisfied, in a least square sense:

$$\Delta\boldsymbol{\lambda} = \left(\mathbf{B}\mathbf{D}\mathbf{B}^T\right)^{-1} \mathbf{B}\mathbf{D}\, \Delta \boldsymbol{F}. \tag{4.65}$$

These three steps are illustrated in figure 4.14.

Stefanica [185] discusses the parallelization properties of this generalized preconditioner. The critical point concerns the two global multiplications by $(\mathbf{B}\mathbf{D}\mathbf{B}^T)^{-1}$ which cannot be done in parallel. One may build local matrices $\tilde{\mathbf{B}}^{(s)} = (\mathbf{B}\mathbf{D}\mathbf{B}^T)^{-1}\mathbf{B}^{(s)}$ once and for all at the beginning of the resolution. In order to preserve the sparsity, biorthogonal Mortars could be helpful in this case [169, 185].

4.4 Numerical investigation of the developed algorithm

In this section, we run several examples to highlight the good behaviour of the proposed algorithm. Firstly, 2D test cases are presented, then we study several shell test problems. For every example, we depict the number of iterations versus the level of refinement of the mesh given a fixed decomposition. In most cases, the convergence study is only given without preconditioner and with the generalized Dirichlet preconditioner (4.60). We do not present, for all examples, the case of the classical Dirichlet preconditioner (4.50). Especially, for homogeneous domain, both presented preconditioners (4.50) and (4.60) improve the convergence rate in an approximately same manner. Thus, for the sake of clarity, we omit to report the convergence study for the classical Dirichlet preconditioner when it leads to identical results than the generalized preconditioner. We highlight the benefits of the scaling brought by the generalized preconditioner on three heterogeneous problems (the heterogeneous beam, the bending plate, and the stiffened panel).

The convergence criteria is defined as a relative tolerance on the projected residual (namely $\boldsymbol{w}^k$ in table 4.1) set as $||\boldsymbol{w}^k|| < 10^{-8}||\boldsymbol{w}^0||$. The results are given in tables

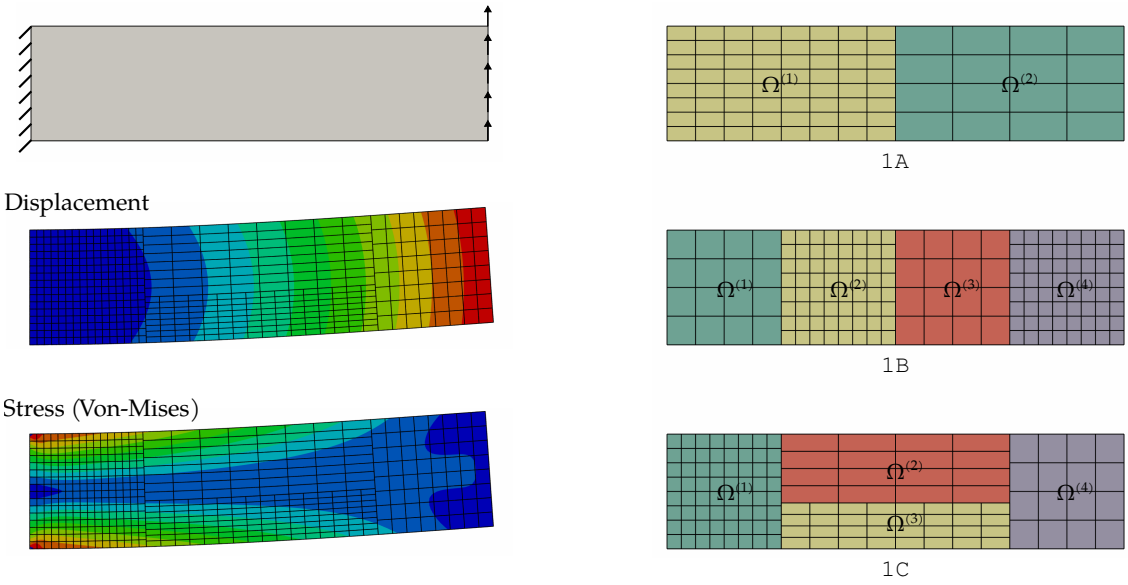


Figure 4.15 – Description and results of the cantilever beam problem. Three decompositions `1A`, `1B`, and `1C` are studied.

	1A				1B				1C			
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$
0	330	12	4	4	702	36	13	8	630	60	48	13
1	946	20	9	7	1966	60	19	9	1686	100	85	14
2	3138	36	21	9	6414	108	36	10	5334	180	149	15
3	11 362	68	45	9	22 990	204	64	10	18 774	340	251	15
4	43 170	132	71	9	86 862	396	95	10	70 230	660	347	16

Table 4.2 – Performance study for the homogeneous cantilever beam problem described in figure 4.15.

where the headers N_{tot} and N_λ denote the numbers of total and interface DOF respectively (*i.e.*, by referring to (4.30), N_λ is the size of $\boldsymbol{\lambda}$, and N_{tot} is the total number of unknowns). Columns with headers **I**, $\tilde{\mathbf{F}}_I^{-1}$ and $\tilde{\mathbf{F}}_D^{-1}$ give the iteration number without a preconditioner, with the classical Dirichlet preconditioner, and with the generalized preconditioner, respectively.

4.4.1 Plane stress problems

The two plane stress problems considered here are tackled using a standard Lagrange multiplier formulation to couple the 2D displacements only. One could perform the same computation using a Kirchhoff–Love formulation with its associated Mortar coupling.

Homogeneous domain The first test case consists in the bending of a 2D cantilever beam with homogeneous material as described in figure 4.15. The geometric and mechanical parameters are chosen as: length $L = 20\,\text{m}$, height $h = 5\,\text{m}$, Young's Modulus $E = 60 \times 10^9\,\text{Pa}$, Poisson's ratio $\nu = 0.30$, and tangential uniform load $P = 1000\,\text{Pa}$. Three different configurations are studied and we denote these decompositions as:

— 1A – two non-matching patches with a similar geometry,

— 1B – four non-matching patches with a similar geometry,

— 1C – four non-matching patches with different geometries.

For every configuration, the patches are discretized with cubic NURBS.

As can be seen in figure 4.15, the displacement and stress fields are smooth and clearly traduce the bending of the beam. Then, table 4.2 summarizes the convergence results for the FETI algorithm for this first test case. One can see that the algorithm requires very few iterations to reach convergence when the preconditioner (4.60) is used. For each decomposition, it converges in a relatively constant number of iterations. It accounts for the numerical scalability of this preconditioner which is in good agreement with the observations of Stefanica [161, 185] and Rixen [193]. For finer refinement levels, the gain is significant in comparison with the un-preconditioned version. It is especially true for the more complex decomposition 1C.

Heterogeneous domain The second test case consists in a heterogeneous beam. The description of the problem is given in figure 4.16. The role of the preconditioner is known to be critical when the sub-domains have large differences of material properties [157, 195]. The beam is made from a sequence of hard and soft regions. One side of the beam is fixed and the other side is subjected to a traction load. Each hard region is defined by a 5 m by 4 m rectangle and has a Young's Modulus equal to $E = 60 \times 10^9\,\text{Pa}$. Each soft region is defined by a 3 m by 4 m rectangle and has a Young's Modulus equal to $E = 6 \times 10^9\,\text{Pa}$. Same Poisson's ratio equal to $\nu = 0.30$ is chosen for both regions. We consider two cases:

— 2A – three non-matching patches; two made of hard material separated by one made of soft material,

— 2B – five non-matching patches; three made of hard material separated by two made of soft material.

Table 4.3 gives the convergence results for the present case. As for the previous example of the homogeneous cantilever beam, the algorithm requires very few iterations to reach convergence when the preconditioner $\tilde{\mathbf{F}}_D^{-1}$ is used, and this number of iterations does not grow with the refinement level. This does not hold true without the preconditioner (columns with header **I**). Therefore, the heterogeneity is correctly tackled by the preconditioner thanks to the superlumped scaling. We also give in table 4.3 the convergence when the classical Dirichlet preconditioner $\tilde{\mathbf{F}}_I^{-1}$ is used. One can see that

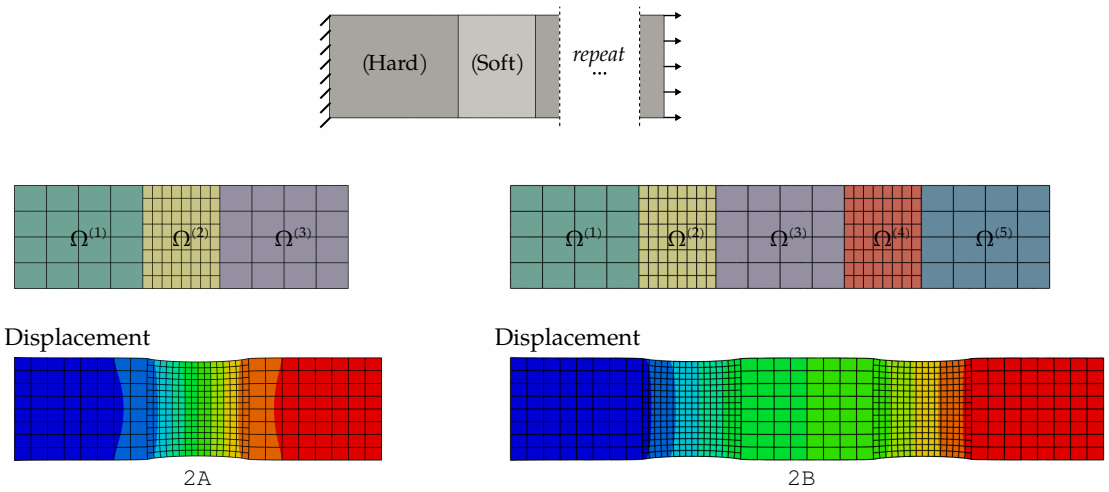


Figure 4.16 – Description and results of the heterogeneous beam problem. The beam is made of a sequence of hard and soft regions. Configuration 2A has two hard regions separated by one soft region. Configuration 2B has three hard regions separated by two soft regions.

	2A					2B				
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	$\tilde{\mathbf{F}}_I^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	$\tilde{\mathbf{F}}_I^{-1}$
0	204	16	6	5	6	368	32	10	6	12
1	448	24	9	7	10	812	48	14	8	16
2	1224	40	15	9	15	2228	80	22	10	21
3	3928	72	29	9	21	7172	144	38	10	26
4	13 944	136	55	10	27	25 508	272	69	11	34

Table 4.3 – Performance study for the heterogeneous beam problem described in figure 4.16.

Figure 4.17 – Convergence of the dual DD solver when applied to the heterogeneous beam problem with five sub-domains 2A (refinement level 4 according to table 4.3). The scaling step of the generalized Dirichlet preconditioner improves drastically the convergence rate.

the benefit is not as good as one can expect from a preconditioner. In fact, for the low levels of refinement, there are no iteration savings when using the classical Dirichlet preconditioner in comparison with the un-preconditioned cases. For finer discretizations, it improves slowly the convergence. The figure 4.17 depicts the convergence histories in case of the configuration 2A with the highest level of refinement (refinement level equals to 4 according to table 4.3). For this particular case, the solver convergence is twice as fast with the preconditioner $\tilde{\mathbf{F}}_I^{-1}$. However, the generalized Dirichlet preconditioner drastically improves the convergence (see figure 4.17). Thus, we highlight here an already known issue: the scaling during the preconditioning is essential to correctly balance the effort between the sub-domains.

4.4.2 Application to non-conforming Kirchhoff–Love shells

Plate bending The third example consists in the bending of a plate as described in figure 4.18. The plate is fully fixed at one side while an uniform pressure is applied over the whole structure. We decompose the plate into two patches and we consider two cases:

— 3A – two patches with the same thickness (homogeneous plate),

— 3B – two patches with different thicknesses such that the ratio is $t^{(1)}/t^{(2)} = 0.1$ (heterogeneous plate).

Cubic NURBS are used and the discretization of the patch with the fixed edge is twice finer than the second one (see figure 4.18). For this particular test case, we observe better results (quality of the stress field) when the Lagrange Multipliers are discretized using the trace space of the finer patch. We opt for this choice here even if the strategy presented in section 4.2.2 leads to acceptable results too. Finally, the parameters of the study are as follows : length $L = 20\,\text{m}$, width $l = 10\,\text{m}$, Young's Modulus $E = 60 \times 10^9\,\text{Pa}$, Poisson's ratio $\nu = 0.20$, and uniform pressure $P = 1.0\,\text{Pa}$. In the homogeneous case, the thickness of both patches is $t = 0.1\,\text{m}$. In the heterogeneous case, the thickness of the fixed patch is $t^{(1)} = 0.1\,\text{m}$ and the thickness of the free patch is $t^{(2)} = 1.0\,\text{m}$.

Table 4.4 presents the performance study of the algorithm for this problem. The results are similar for both the homogeneous and the heterogeneous configurations. More specifically, without the preconditioner the algorithm converges in the same number of iterations in both cases. It can be seen that, with preconditioner $\tilde{\mathbf{F}}_D^{-1}$ the algorithm converges in slightly less iterations for the heterogeneous plate. Furthermore, the number of iterations is kept low even for the finer meshes when this preconditioner is used. Concerning the convergence of the solver with the classical Dirichlet preconditioner $\tilde{\mathbf{F}}_I^{-1}$, good results are obtained in the homogeneous case 3A. Surprisingly, the convergence rate is even better than with the generalized Dirichlet preconditioner for the finest discretization (see refinement level 4 for configuration 3A). However, for the heterogeneous plate with different thickness per patch, this preconditioner performs worse than preconditioner $\tilde{\mathbf{F}}_D^{-1}$ which takes into account the relative stiffness at the interface between

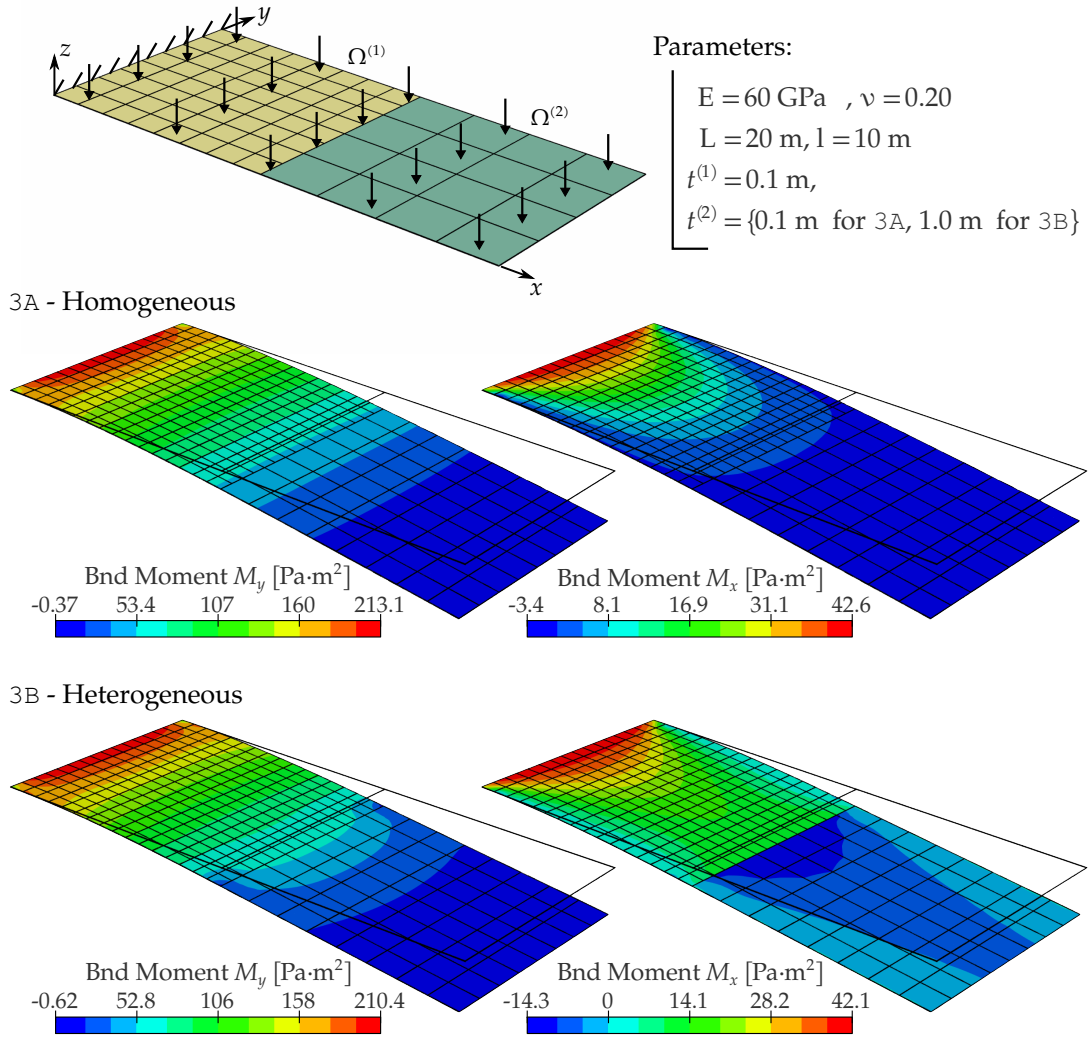


Figure 4.18 – Description and results of the bending plate problem. The plate is decomposed into two non-matching patches. Two cases are studied: 3A the homogeneous case where the patches have the same properties, and 3B the heterogeneous case where the shells have different thicknesses.

	3A					3B				
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	$\tilde{\mathbf{F}}_I^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	$\tilde{\mathbf{F}}_I^{-1}$
0	222	28	9	6	6	222	28	7	4	7
1	510	44	14	7	9	510	44	14	4	11
2	1446	76	29	11	12	1446	76	28	6	18
3	4758	140	60	18	14	4758	140	59	11	28
4	17 142	268	118	26	14	17 142	268	117	16	33

Table 4.4 – Performance study for the bending plate problem described in figure 4.18.

the sub-domains. This point was predictable.

Ideally, the number of iterations should be constant versus the mesh refinement as observed for the plane stress problems 4.4.1. Here, we do not have a perfect numerical scalability but it is not surprising. In fact, the one-level FETI equipped with the Dirichlet preconditioner is known to be numerically scalable for second-order problems. However, it does not hold true for fourth-order problems as shown for example by Farhat et al. [154]. That is why extensions of the classical FETI method, as for instance the two-level FETI and latter on the FETI-DP, have been introduced to optimally deal with plates and shells [154, 155]. Based on these remarks, we understand the increase of the iteration counts versus the mesh refinement and the same behaviour is expected for the other shell problems tackled in this work. Nevertheless, for this test case of a plate, the presented dual Domain Decomposition algorithm performs well and leads to entirely acceptable results.

Scordelis–Lo roof The Scordelis–Lo roof is part of the shell obstacle course which is widely used to study the performance of shell formulations [130]. It consists of a portion of cylinder subjected to a vertical gravity load and fixed at its two end sections using rigid diaphragms. The study parameters and some results for this test case can be found in the abundant shell literature: *e.g.* [94, 128, 130]. We decompose the Scordelis-Lo roof into three non-matching configurations as depicted in figure 4.19:

— `4A` – three patches with a curved interface obtained by using the embedded Kirchhoff–Love formulation (introduced later on in chapter 5),

— `4B` – four quasi-identical patches which intersect at a cross point,

— `4C` – nine patches with different levels of refinement. The coarsest are quartic patches while the finest are cubic patches.

Figure 4.19 shows the vertical displacement field and the distribution of a bending moment for each configurations, which appear smooth as expected. The results investigating the numerical scalability are given in table 4.5. It can be seen that the benefit of the preconditioning step is increasing from the first configuration `4A` to the third configuration `4C`. For the first configuration `4A`, the iteration count is only reduced by a factor of 2 or 3 (except for the coarsest mesh). For the second configuration `4B`, the iteration count is reduced by a factor of 4 for the refinement levels above 1. Finally, for the more complex decomposition `4C`, the iteration count is drastically reduced when using the generalized Dirichlet preconditioner. Looking at the refinement levels 3 and 4 in table 4.5 (and columns referred to the configuration `4C`), we observe that the number of iterations is reduced by a factor of 10, and 15 respectively. Thus, the benefit provided by the preconditioner is all the more remarkable in case of complex Domain Decomposition. Even if numerical scalability is not fully reached, the growth of the iteration count with the refinement level is very slow for each configuration and it leads to satisfactory results.

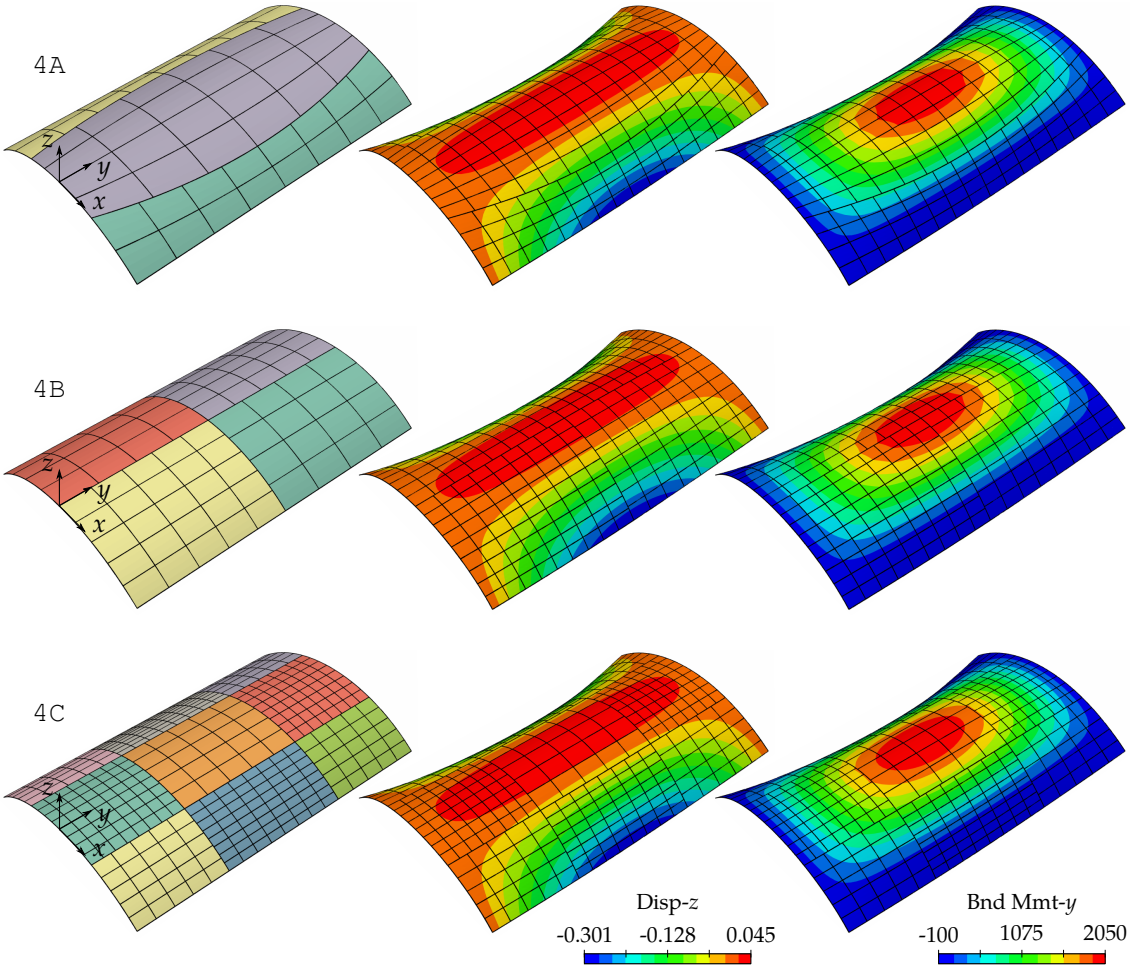


Figure 4.19 – Description and results for the Scordelis–Lo roof test case. Three different non-matching decomposition are studied: `4A` is made of three patches with curved interfaces, `4B` is made of four quasi-identical patches (two patches have one more element per direction), and `4C` is made of nine patches with different levels of refinement and degrees. (Note: The displacement has unit of length and the bending moment has unit of moment per unit length, respectively).

	4A				4B				4C			
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$
0	527	78	47	34	708	92	70	34	1513	276	167	44
1	1399	142	73	35	1652	156	117	38	3017	468	317	52
2	4295	270	124	45	4692	284	188	48	7537	842	558	64
3	14 695	526	158	59	15 380	540	276	64	22 625	1620	949	87
4	53 927	1038	175	81	55 188	1052	405	86	76 993	3156	1783	121

Table 4.5 – Performance study for the Scordelis-Lo shell problem described in figure 4.19.

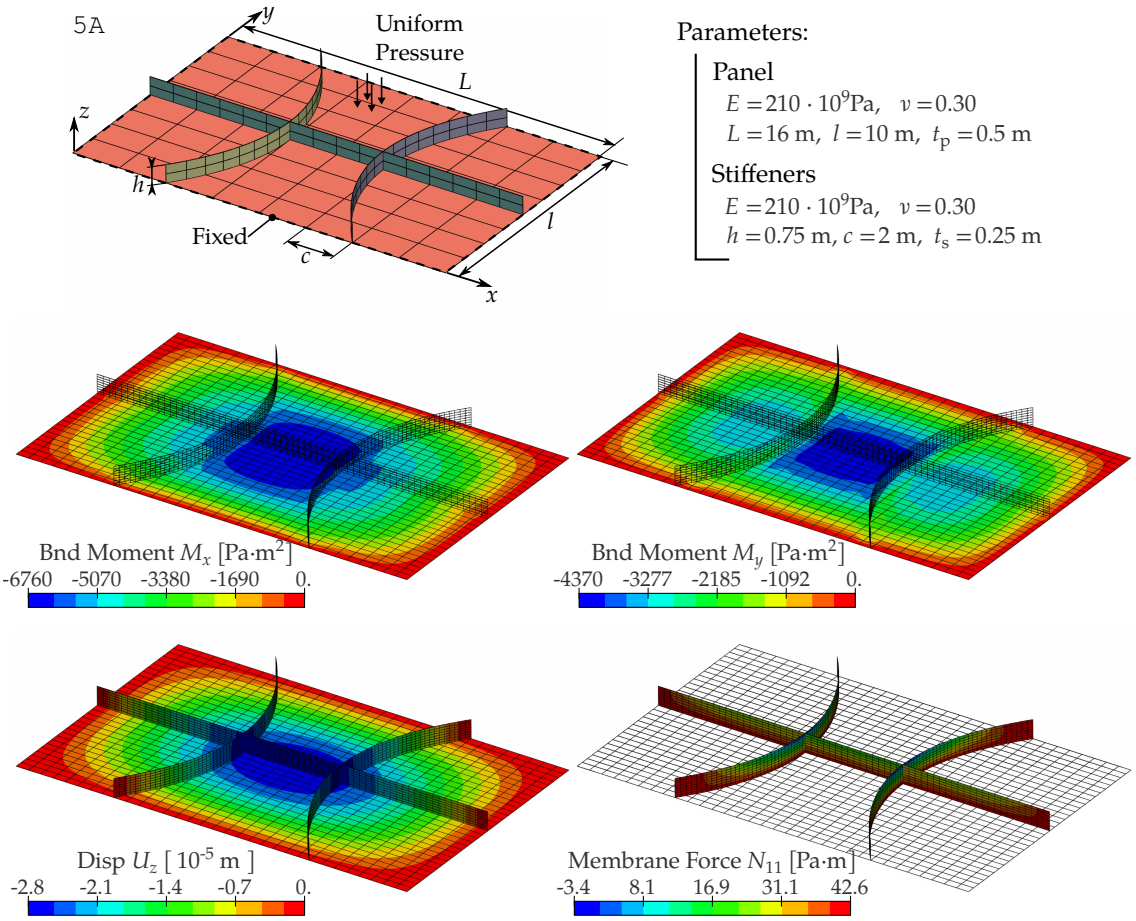


Figure 4.20 – Description and results for the stiffened panel problem. It consists in the assembly of a rectangular plate with three stiffeners; two with a parabolic shape crossed by a straight one. It leads to four non-conforming patches and five coupling interfaces (three stiffener/panel and two stiffener/stiffener interfaces). The main stress resultants for the panel are the bending moments whereas the stiffeners are mainly subjected to membrane forces.

Stiffened panel The problem of the stiffened panel is presented in figure 4.20. It consists in the assembly of a square plate with sub-parts called the stiffeners. We design three stiffeners: two with a parabolic shape and the third one with a straight shape. The straight patch intersects the two curvilinear stiffeners. As a result, five non-conforming interfaces are defined: each interface cuts several isogeometric elements. Every patch is discretized using cubic Kirchhoff–Love shells. The material and geometric parameters of the study are given in figure 4.20. The edges of the panel are fixed (no displacement). The uniform pressure is equal to $P = 1000\,\text{Pa}$. We denote by `5A` this problem of the stiffened panel.

Once again, we run the analysis for several levels of refinement with and without the preconditioning step. The results of the study are given in table 4.6. We also depict the convergence in figure 4.21 for the refinement level equal to 2 (which correspond

	5A				
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$	$\tilde{\mathbf{F}}_I^{-1}$
0	1227	147	109	50	81
1	3313	259	193	68	94
2	10 365	483	341	85	110
3	35 989	931	613	103	137
4	133 317	1827	1106	125	189

Table 4.6 – Performance study (left) for the stiffened panel problem 5A described in figure 4.19.

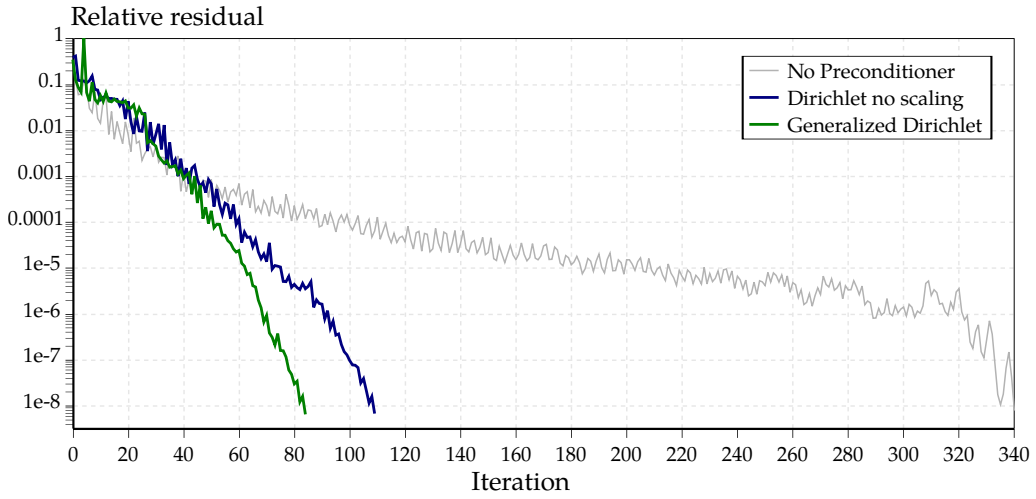


Figure 4.21 – Convergence plot of the dual DD solver when applied to the stiffened panel 5A (refinement level 2 according to table 4.6). Without the preconditioning step, the solver faces difficulties to converge. The scaling of the generalized Dirichlet preconditioner accelerates the convergence in comparison with the classical Dirichlet preconditioner.

to the mesh where the color maps are presented in figure 4.20). We observe the same behaviour than for the previous test cases. Indeed, the iteration count is drastically reduced with the use of the preconditioner and it is all the more true when increasing the refinement level. We give the results for both the classical Dirichlet preconditioner $\tilde{\mathbf{F}}_I^{-1}$ and the generalized preconditioner $\tilde{\mathbf{F}}_D^{-1}$ in order to highlight the positive influence of the scaling step occurring in the second one. In fact, in case of stiffened structures, the T-shape interfaces are source of heterogeneities even if both sub-domains are of the same material properties. For instance, the stiffener is mainly subjected to membrane forces whereas the panel works in bending. This leads to high differences in stiffness at the interface. A shell is much stiffer under membrane loading than under bending loading. Thus, as shown in table 4.6 and in figure 4.21, the generalized preconditioner performs better than classical Dirichlet preconditioner.

As for the previous shell problems, the numerical scalability is not completely achieved but the growth of the number of iterations remains slow while introducing the precon-

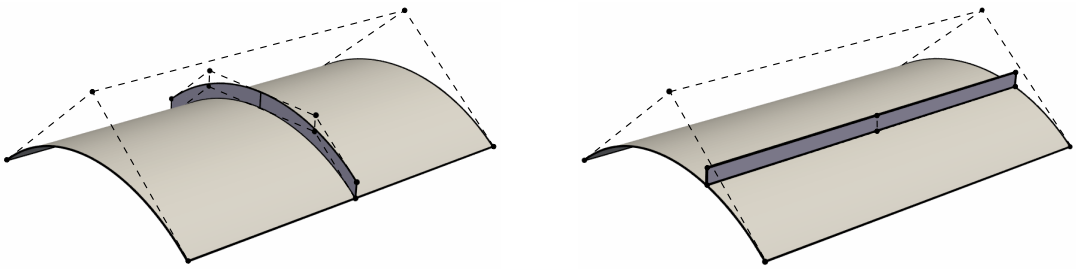
ditioning step. Especially, satisfactory results are obtained even for very fine meshes. For instance, with the refinement level of 4, the dual decomposition domain algorithm converges in 125 iterations even though the total number of DOF is over 100 000. This is a crucial point. Because the algorithm solves an interface problem where the unknowns are the Lagrange Multiplier DOF (see equation (4.39)), we are able to measure, during the resolution, the residual of the coupling condition (4.32). The convergence criteria is formulated on this residual which, in other words, ensure the good fulfillment of the coupling condition. For large problem, the monolithic solution obtained with a basic direct solver applied directly to the linear system (4.30) does not give suitable results. The numerical errors during the resolution significantly affect the correct imposition of the coupling conditions. Thus, we believe that the presented algorithm is viable and therefore attractive to tackle more complex models as encountered in industrial application.

Chapter 5

Multipatch Shape Optimization

Moving a method from a proof-of-concept to a robust industrial process is neither automatic nor evident in practice. Research sometimes lacks of continuity and multiple innovative works unfortunately remain to their initial-stage. Based on what has been yet successfully developed in this work, this chapter aims to apply isogeometric shape optimization for engineering applications, especially for designing Aerostructures. The key point concerns the use of non-conforming multipatch discretizations for shape design. We present a methodology inspired from Free-Form Deformation techniques and we introduce an embedded shell formulation. It enables to impose complex geometric constraints between non-conforming patches in a simple and suitable manner.

Contents



Circumferential stiffener Longitudinal stiffener

Figure 5.1 – Designing simple stiffeners lying on a portion of cylinder.

5.1 Free-Form Deformation

Applying the isogeometric shape optimization strategy for designing large stiffened aerostructures is a challenging task from both geometric and analysis sides. From the analysis point of view, we have now an efficient strategy to couple non-conforming discretizations (see the previous chapter 4). It will be useful here and we will rely on this coupling strategy. Indeed, the use of non-conforming discretizations is convenient during the shape update since it avoids, for example, to call on cumbersome procedures to impose matching meshes between the stiffeners and the panel. However, looking at geometric aspects, we had to develop an adequate methodology in order to impose the shape modification to complex structures with multiple parts. This issue is tackled in this section.

5.1.1 Complex geometric constraints

Let us start with a simple example of a portion of cylinder where the goal is to design a flat stiffener. For instance, one can define a simple flat stiffener in the circumferential or the longitudinal directions as described in figure 5.1. Both cases can be done quite easily.

Now let us try to impose shape modifications for this simple case of the portion of cylinder with one flat stiffener. Based on the isogeometric shape optimization process presented in chapter 3, these shape updates are assigned by moving the control points of the NURBS. In case of the circumferential stiffener, the control points associated to the stiffener can be moved, for example, along the longitudinal direction. This shape modification is depicted in figure 5.2 (top) and does not present any particular problem. However, it becomes more tricky with the longitudinal stiffener. To modify its shape, one can try to move the control points in the circumferential direction as described in figure 5.2 (middle). A first difficulty arises. In fact, we want the bottom of the stiffener to perfectly lie on the cylinder. This geometric constraint is not easily fulfilled. Sliding the control points along the edge of the cylinder disconnects the surfaces. The second difficulty arises when modifying the global shape (here the portion of cylinder)

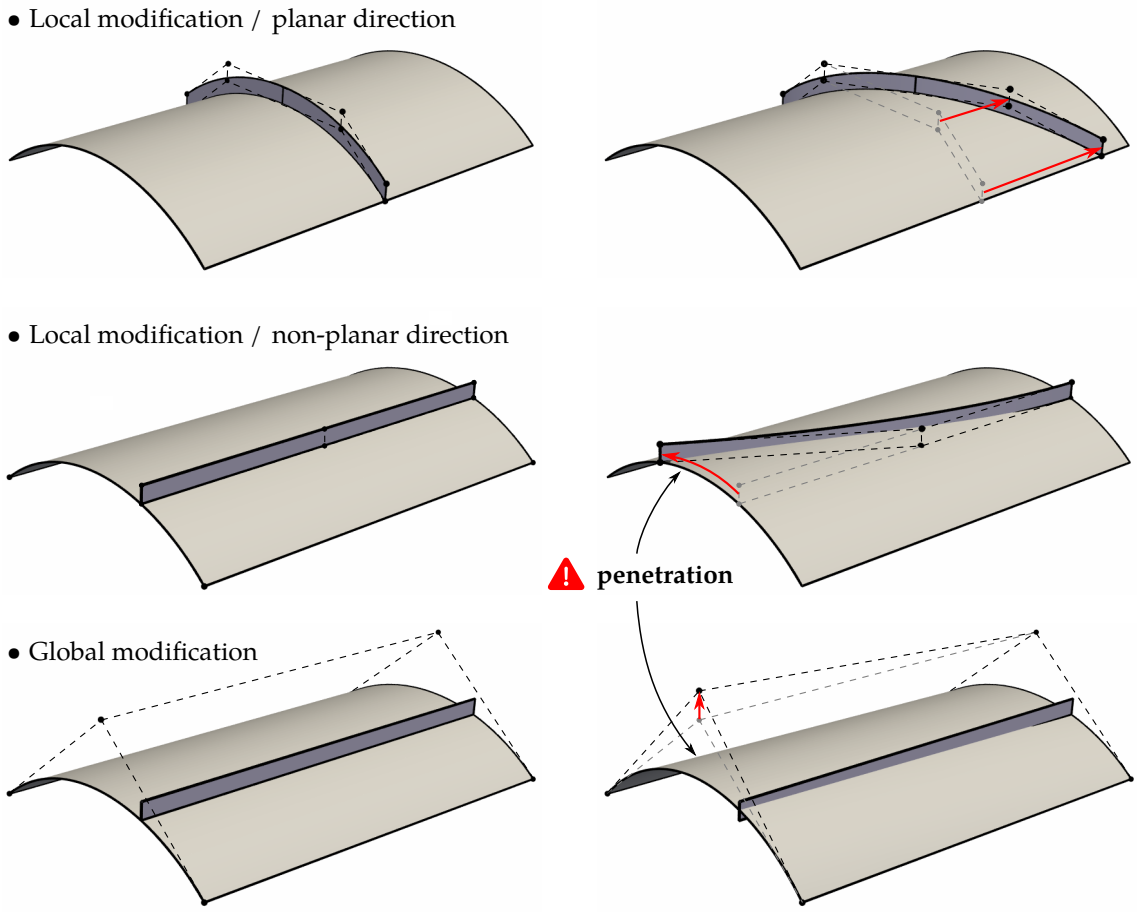


Figure 5.2 – Moving a stiffener lying on a portion of cylinder. Imposing the shape update in non-planar directions is challenging due to the violation of the bonding condition.

while keeping the substructure connected. More precisely, the question here is how to transfer the shape update from the global part to its substructure? An illustration of the problem is shown in figure 5.2 (bottom). If we move one control point of the main surface, the stiffener does not lie anymore on it. In this simple case, one could formulate an appropriate condition in order to glue the stiffener but, in general, it is not a trivial task. In case of curvilinear stiffeners lying on a curved surface, re-parametrizations and approximations appear inevitable if the standard NURBS-based strategy for shape optimization is adopted. Instead, we are looking for a compact approach that does not require cumbersome re-parametrization procedures. This is particularly crucial to get a robust shape optimization strategy.

To summarize, we are looking for a geometric strategy that enables to impose the following shape modifications in a convenient way:

— global shape optimization of a whole structure that contains multiple parts,
— local shape optimization of specific sub-parts (such as stiffeners).

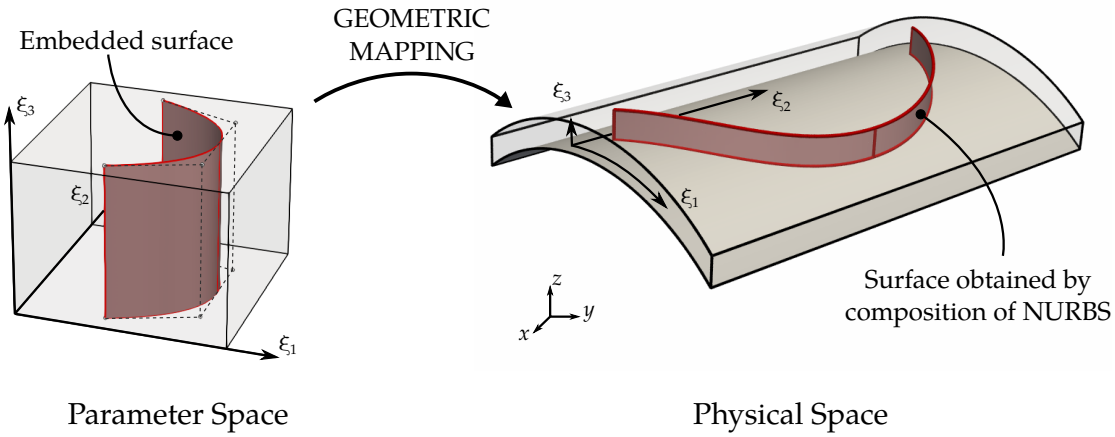


Figure 5.3 – Construction of a stiffened structure using embedded surfaces. The final surface describing the stiffener is obtained by the composition of a NURBS surface and a NURBS volume. This construction ensures that the stiffener is perfectly lying on the portion of cylinder.

5.1.2 Using NURBS compositions for shape updating

Embedded surface We introduce the concept of embedded surface in order to overcome the geometric challenges highlighted in the previous section. The idea comes from Bauer et al. [182] who considered univariate embedded entities. More precisely, they embed a curve into a surface. This technique allows them to model cables in membrane structures or to apply line loads and supports to a shell structure without being limited to the edges of the surface.

We took this idea of embedding entities to define the stiffeners. We will not use classical NURBS surfaces to describe the stiffeners. Instead of a simple surface, we introduce the composition of a NURBS volume and a NURBS surface. In other words, we immerse a surface into the parametric space of a volume. The volume can be seen as a mapping that smartly transforms the embedded surface in order to create a final surface with the desired properties. Figure 5.3 illustrates this process.

This idea offers a novel strategy to generate geometric models of stiffened structures. This new approach is presented in figure 5.4. The construction contains four steps:

1. build the panel (main part to be stiffened) using a standard B-Spline or NURBS surface,
2. extrude the surface representing the panel so as to generate a volume,
3. embed surfaces into the parameter space of the generated volume,
4. transform these embedded surfaces using the volume mapping.

The final surfaces obtained by the composition of the embedded surfaces and the volume mapping give the geometric representation of the stiffener. With this procedure, the stiffeners are perfectly lying on the panel. No approximations are introduced during the geometric modeling.

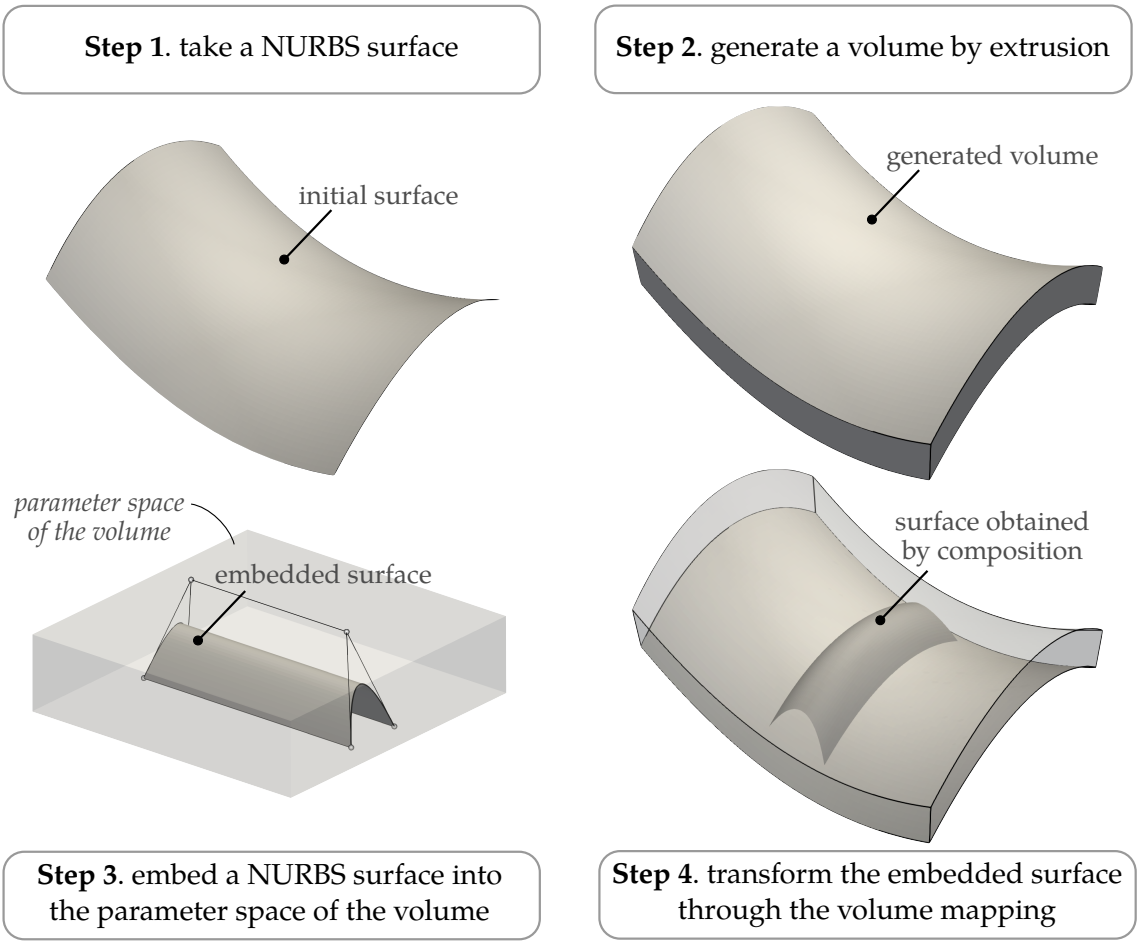


Figure 5.4 – Generating a geometric model of a stiffened structure by using the embedded strategy.

The presented strategy can be related with Free-Form Deformation techniques [197]. These methods are commonly used in graphic design to deform objects and are also applied in engineering fields as, for example, aerostructural optimization [198]. In that sense, the design approach presented in this work is not entirely disconnected from the current industrial methods.

Finally, we will interpret the embedded entities as *slave parts* since their shapes are driven by external geometric properties. Conversely, the *master parts* are those which dictate specific constraints during the shape update. For example, in the case depicted in figure 5.3, the portion of cylinder is the master part and the stiffener constitutes the slave part: modifying the master part imposes to transfer shape modification to the slave part and, in turn, the possible shape modifications of the slave part are restricted to follow the master geometry.

Mathematical description Let us now give the mathematical definitions of the quantities involved in the presented strategy. We define the mapping $\boldsymbol{V}$ as a NURBS volume:

$$\boldsymbol{V}(\xi_1, \xi_2, \xi_3) = \sum_{ijk} R_{ijk}(\xi_1, \xi_2, \xi_3)\boldsymbol{P}_{ijk}, \tag{5.1}$$

where R_{ijk} are trivariate NURBS basis functions and $\boldsymbol{P}_{ijk}$ are the related control points. The parametric coordinates ξ_1, ξ_2 and ξ_3 are defined over the parametric domain $\bar{\Omega}$. The embedded surface $\bar{\boldsymbol{S}}$ is a NURBS surface defined as:

$$\bar{\boldsymbol{S}}(\theta_1, \theta_2) = \sum_{ab} \bar{R}_{ab}(\theta_1, \theta_2)\bar{\boldsymbol{P}}_{ab}, \tag{5.2}$$

with $\bar{R}_{ab}$ some bivariate NURBS basis functions and $\bar{\boldsymbol{P}}_{ab}$ their associated control points. Under the consideration that the codomain of $\bar{\boldsymbol{S}}$ is included in domain $\bar{\Omega}$, we finally compose the two previous mapping (5.1) and (5.2) to get the physical surface:

$$\boldsymbol{S}(\theta_1, \theta_2) = \boldsymbol{V}\big(\ \bar{\boldsymbol{S}}(\theta_1, \theta_2)\ \big). \tag{5.3}$$

The surface $\boldsymbol{S}$ is parametrized by the same set of variables (θ_1, θ_2) as the embedded surface $\bar{\boldsymbol{S}}$ and it returns value in the physical space $\mathbb{R}^3$.

Potential The potential of using embedded entities is multiple from the design point of view. Our first motivation is to design stiffened structures found in aeronautics but, of course, it should not be limited to this purpose.

Figure 5.5 shows the potential of the presented approach from the design point of view. The approach eases the construction of complex stiffened structures. It also offers interesting feature related to shape optimization. For instance, we now have two possibilities to impose shape modifications to a structure with multiple parts. One can impose global shape deformation to the whole structure by acting on the control points associated to the volume mapping. This is finally a Free-Form Deformation process. Using NURBS compositions also enables to impose local shape modifications to specific parts without worrying about the bonding conditions at the interface. Due to the construction (see figure 5.4), the geometric constraints of non-separation between the different parts are automatically fulfilled. The uni-directional translation of the embedded surfaces will slide the stiffener along the panel while automatically respecting its curvature, if one exists. More complex examples are presented in chapter 6 that take advantage of this versatile approach for geometric modeling.

5.2 An Embedded Kirchhoff–Love shell

We have introduced the use of NURBS compositions for geometric requirements. In order to perform the shape optimization using this approach, we know, since chapter 3, that we also need to carry out the structural analysis. In other words, we need a

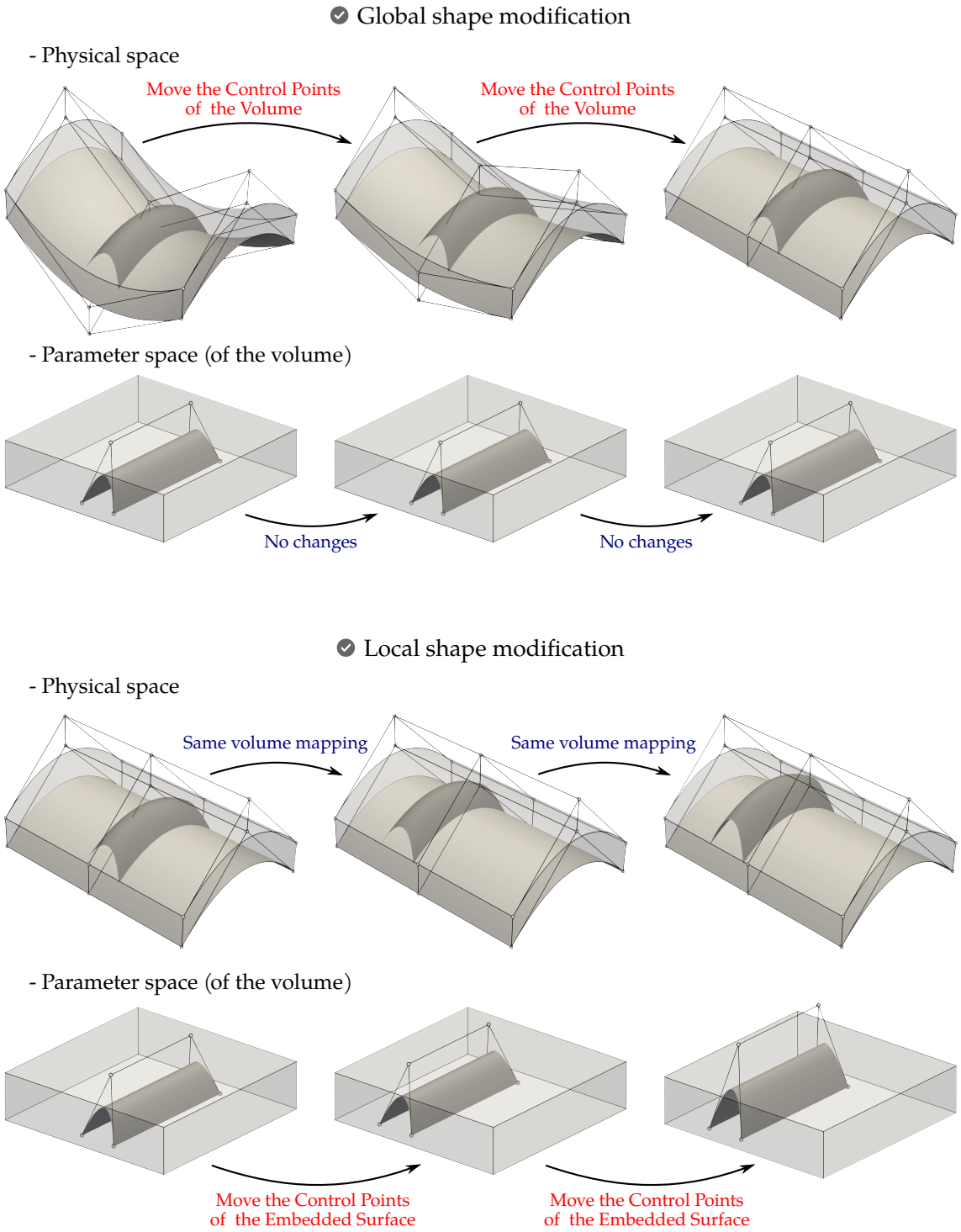


Figure 5.5 – Using NURBS compositions offers two possibilities for imposing shape deformations. By modifying the volume, one can act on the global shape. By acting on the embedded entities, one can modifying specific parts, as for instance the stiffeners. The bonding condition at the interfaces are automatically satisfied.

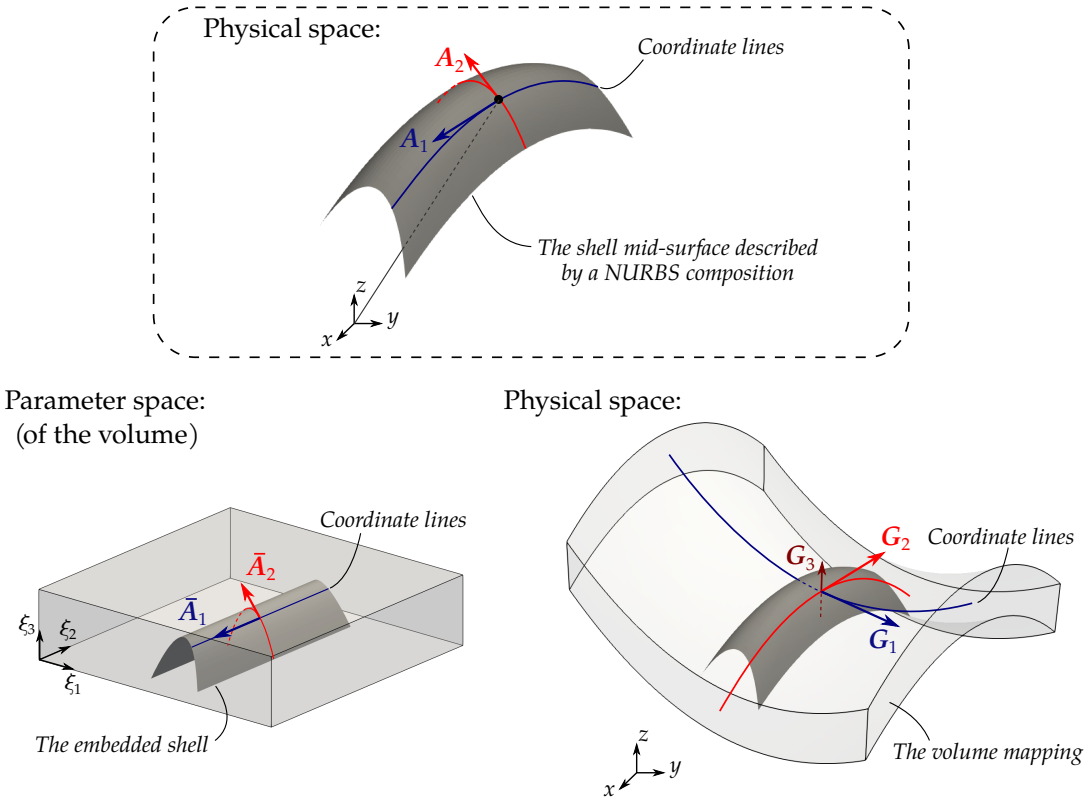


Figure 5.6 – Computing the covariant basis vectors in case of a NURBS composition.

shell element that can be use in case of NURBS compositions. Thus, in this section we present the embedded Kirchhoff–Love shell that fulfills this need. The theoretical results presented in section 3.2.2 for the standard isogeometric Kirchhoff–Love shells serve as prerequisites here.

5.2.1 Discretizations aspects

Mid-surface defined by NURBS composition The continuum formulation of the embedded Kirchhoff–Love is formally identical to the standard Kirchhoff–Love formulation. The same kinematic assumptions are postulated. It means that all the equations from (3.39) to (3.62) remain true. Things become different at the discretization step.

The first difference is that the mid-surface is no longer described by a single NURBS surface but by the NURBS composition of an embedded NURBS surface and a NURBS volume. We already formulate these geometric quantities in equation (5.3). However, the surface obtained by a NURBS composition is still a parametric surface. We can take the geometric parameters of the embedded surface as the curvilinear coordinates (θ_1, θ_2) involved in the shell formulation. Hence, the covariant basis vectors $\boldsymbol{A}_\alpha$ (derivatives of the position vector *w.r.t.* the curvilinear coordinates, see equation (3.26)) are obtained by

applying a chain rule differentiation to equation (5.3). We get:

$$\boldsymbol{A}_\alpha = (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_1)\,\boldsymbol{G}_1 \;+\; (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_2)\,\boldsymbol{G}_2 \;+\; (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_3)\,\boldsymbol{G}_3 \tag{5.4}$$

where $\bar{\boldsymbol{A}}_\alpha$ and $\boldsymbol{G}_i$ denote the covariant vectors associated respectively to the embedded surface and the volume mapping (see figure 5.6). With the same notations as in equations (5.1) and (5.2), these vectors are given by:

$$\boldsymbol{G}_i \;:=\; \boldsymbol{G}_i\big(\bar{\boldsymbol{S}}(\theta_1,\theta_2)\big) \;=\; \sum R_{ijk,i}\big(\bar{\boldsymbol{S}}(\theta_1,\theta_2)\big)\boldsymbol{P}_{ijk}, \tag{5.5}$$

$$\bar{\boldsymbol{A}}_\alpha \;:=\; \bar{\boldsymbol{A}}_\alpha(\theta_1,\theta_2) \;=\; \sum \bar{R}_{ab,\alpha}(\theta_1,\theta_2)\bar{\boldsymbol{P}}_{ab}. \tag{5.6}$$

Then, the calculation of the director vector $\boldsymbol{A}_3$ is done as for the standard Kirchhoff–Love shell. Let us repeat equation (3.40) that defines this vector:

$$\boldsymbol{A}_3 = \frac{1}{A}\boldsymbol{A}_1 \times \boldsymbol{A}_2, \quad \text{with} \quad A = |\boldsymbol{A}_1 \times \boldsymbol{A}_2|.$$

Finally, one can already calculate the geometric quantities involved in the membrane strains (3.52). Indeed, it only requires the covariant vectors:

$$\boldsymbol{e}_{\alpha\beta} = \frac{1}{2}\left(\boldsymbol{v}_{,\alpha}\cdot\boldsymbol{A}_\beta + \boldsymbol{v}_{,\beta}\cdot\boldsymbol{A}_\alpha\right).$$

Let us also repeat the expression of the material tensor (3.54) for sake of clarity:

$$\bar{C}^{\alpha\beta\gamma\delta} = \bar{\lambda}A^{\alpha\beta}A^{\gamma\delta} + \mu\left(A^{\alpha\gamma}A^{\beta\delta} + A^{\alpha\delta}A^{\beta\gamma}\right).$$

Again, the contravariant metrics $A^{\alpha\beta}$ are computed as would be in the case of standard Kirchhoff–Love NURBS elements. Thus, regarding implementation aspects, the part of the code that computes the quantities in question is exactly the same for the standard and the embedded shell.

The only additional adjustments considering the geometric quantities appear in the bending terms. Let us rewrite the expression of the bending strain given by equation (3.53):

$$\begin{aligned}\boldsymbol{\kappa}_{\alpha\beta} = -\boldsymbol{v}_{,\alpha\beta}\cdot\boldsymbol{A}_3 + \frac{1}{A}\Big(\boldsymbol{v}_{,1}\cdot(\boldsymbol{A}_{\alpha,\beta}\times\boldsymbol{A}_2) + \boldsymbol{v}_{,2}\cdot(\boldsymbol{A}_1\times\boldsymbol{A}_{\alpha,\beta})\Big)\\ + \frac{\boldsymbol{A}_3\cdot\boldsymbol{A}_{\alpha,\beta}}{A}\Big(\boldsymbol{v}_{,1}\cdot(\boldsymbol{A}_2\times\boldsymbol{A}_3) + \boldsymbol{v}_{,2}\cdot(\boldsymbol{A}_3\times\boldsymbol{A}_1)\Big).\end{aligned}$$

The derivatives of the covariant vectors *w.r.t.* the curvilinear coordinates are required. Their expressions are somewhat more complicated than for the standard NURBS-based shell. We apply one more time the chain rule to equation (5.4). It reads as:

$$\boldsymbol{A}_{\alpha,\beta} \;=\; \sum_i(\bar{\boldsymbol{A}}_{\alpha,\beta}\cdot\boldsymbol{e}_i)\,\boldsymbol{G}_i \;+\; \sum_i\sum_j(\bar{\boldsymbol{A}}_\alpha\cdot\boldsymbol{e}_i)(\bar{\boldsymbol{A}}_\beta\cdot\boldsymbol{e}_j)\,\boldsymbol{G}_{i,j}\,. \tag{5.7}$$

At this point, we mentioned all the differences regarding the geometric quantities between the standard isogeometric Kirchhoff–Love shells and the embedded Kirchhoff–Love shells. Finally, the only changes concern the covariant vectors (5.4) and their derivatives (5.7). Thus, one has to add very little modifications to an existing code including the standard Kirchhoff–Love NURBS elements such that it offers the embedded version.

Approximation space for the displacement field In addition to the discretization of the geometry, we need to clarify how the displacement is discretized to get the embedded Kirchhoff–Love shell elements.

Applying Galerkin's method raises the question of the choice of the approximation space $\mathcal{V}^h$ for the displacement field. In fact, the isoparametric concept may not be applicable in case of a NURBS composition. We need to define the solution space as all linear combinations of a given set of basis functions. Since second derivatives of the displacement field are involved in the bending term, the approximated displacement field has to be C^1 as the geometry. Two choices emerge:

1. based on the trivariate NURBS discretization of the mapping:

$$u^h(\theta_1,\theta_2) = \sum_{ijk} R_{ijk}\big(\bar{\boldsymbol{S}}(\theta_1,\theta_2)\big)\ \boldsymbol{u}_{ijk}, \tag{5.8}$$

2. based on the bivariate NURBS discretization of the embedded surface:

$$u^h(\theta_1,\theta_2) = \sum_{ab} \bar{R}_{ab}(\theta_1,\theta_2)\,\boldsymbol{u}_{ab}. \tag{5.9}$$

The first choice provides an immersed-like approach where the deformation of the shell is prescribed by the surrounding volume [199–202]. The second choice is close to the standard Kirchhoff–Love isogeometric shell. The mapping only plays a role from the geometric point of view since it modifies the shape of the embedded surface but it is not involved in the solution space. Depending on the application, one can adopt either one or the other approach. The first approach can be interesting if the volume which surrounds the surface is a structural part. In case of void, a lot of degrees of freedom may be inactive leading to an ill-conditioned system, which is a well-known drawback of immersed method. Special treatments, such as penalization techniques [202, 203], have to be performed. It is particularly true for higher continuity discretizations since the supports of the NURBS basis functions becomes larger. The second approach leads to a smaller system because it uses a bivariate parametrization (instead of a trivariate one for the first approach). No regularization is required and higher continuity discretizations (cubic and quartic for example) can be used. We notice that the second approach is also easier from the implementation point of view, especially during the stiffness matrix assembly.

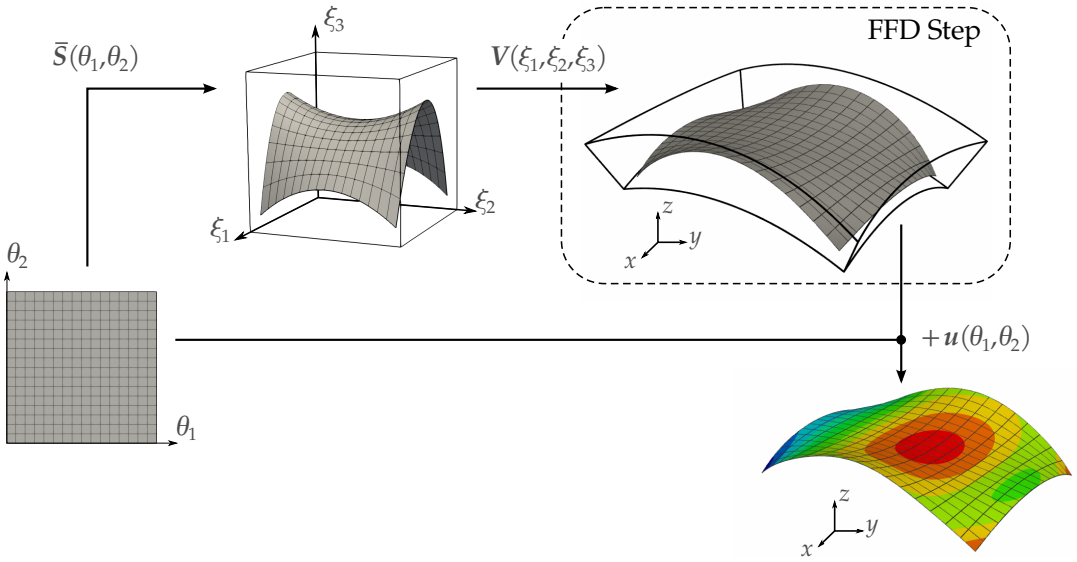


Figure 5.7 – Overview of the embedded isogeometric Kirchhoff–Love shell elements. The displacement field is approximated by the same parametrization as the embedded surface. However, the embedded surface does not describe the geometry of the shell. An additional mapping step further modifies the geometry of the shell.

In this work, we apply the second methodology (A.18). An overview of the approach is depicted in figure A.9. Compared to the classical isogeometric Kirchhoff–Love shell, the difference lies in the additional trivariate mapping step that modifies the shape of the undeformed configuration. We name it the FFD step since it reminds the Free Form Deformation technique [197].

5.2.2 Structural Analysis

With the approximation u^h in hand, one can build the stiffness matrix associated to the embedded Kirchhoff–Love shell. It takes the same form as the standard NURBS-based elements (see equation (3.68)):

$$\mathbf{K}^e_{kl} = \int_{\bar{\Omega}^e_0} \left[t\mathbf{B}^{m\,T}_k \mathbf{H} \mathbf{B}^m_l \; + \; \frac{t^3}{12} \mathbf{B}^{b\,T}_k \mathbf{H} \mathbf{B}^b_l \right] A \mathrm{d}\theta_1 \mathrm{d}\theta_2,$$

where the membrane strain matrices $\mathbf{B}^m$, the bending strain matrices $\mathbf{B}^b$, and the material tensor $\mathbf{H}$ take similar expressions as in equation (3.64), (3.65), and (3.67), respectively. To avoid some confusion, let us note that the basis functions involved in these matrices are those from the embedded surface (denoted previously $\bar{R}_{ab}$). For example, the membrane matrix should be viewed as:

$$\mathbf{B}^m_{ab} = \begin{bmatrix} \bar{R}_{ab,1}\, \boldsymbol{A}^T_1 \\ \bar{R}_{ab,2}\, \boldsymbol{A}^T_2 \\ \bar{R}_{ab,1}\, \boldsymbol{A}^T_1 + \bar{R}_{ab,2}\, \boldsymbol{A}^T_2 \end{bmatrix} \tag{5.10}$$

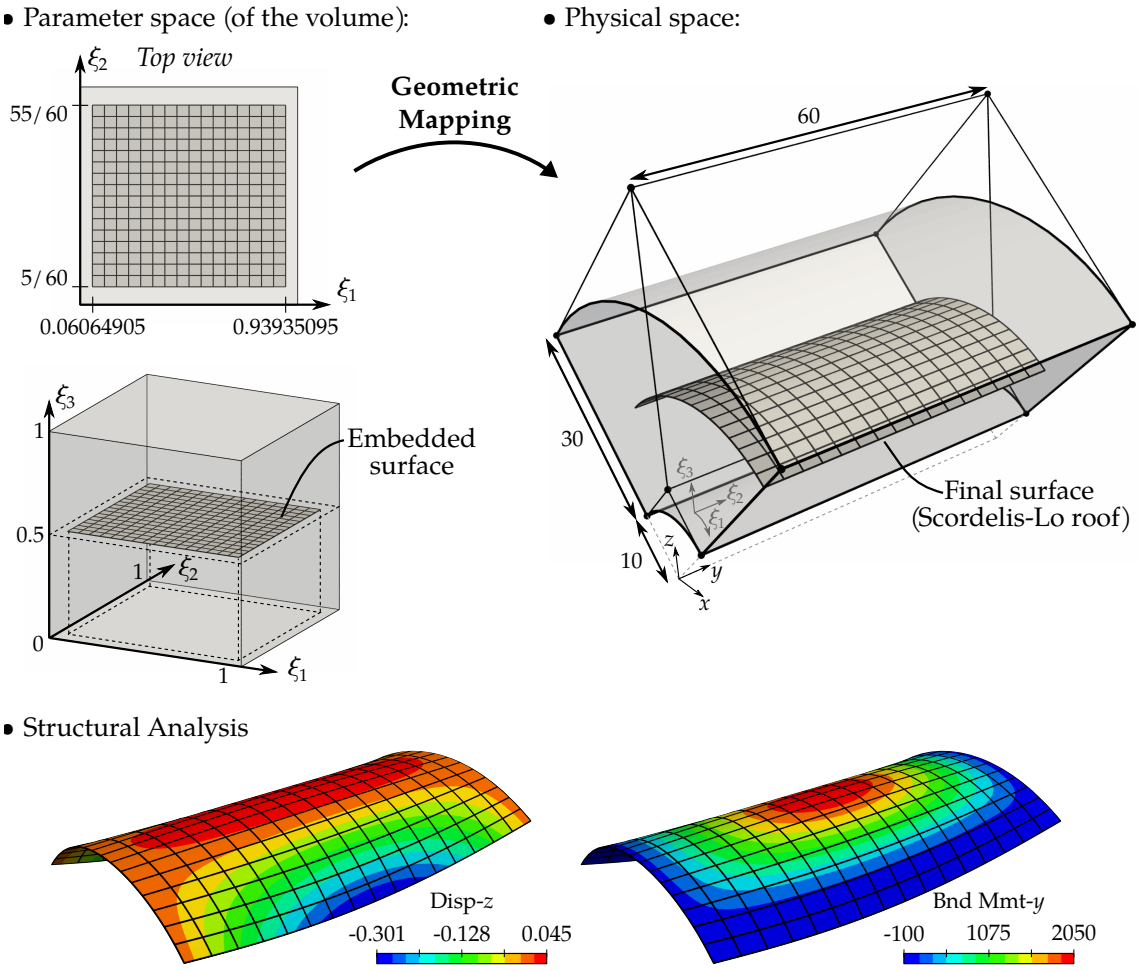


Figure 5.8 – Analyzing the Scordelis-Lo roof using the embedded Kirchhoff–Love shell formulation.

where the covariant vectors $\boldsymbol{A}_\alpha$ are calculated accordingly to equation (5.4).

We now have all the ingredients to perform the structural analysis directly on geometric models built with a NURBS composition. In figure 5.8, we present the analysis of the Scordelis-Lo roof with this new approach. The geometry of the roof is obtained by embedding a square surface into a volume that describes one quarter of a cylindrical annulus. The geometrical parameters of the embedded surface and the volume mapping are defined such that the composition gives a final surface with the correct dimensions (*i.e.* those given in figure 3.5). Once the geometric model is generated, one can perform the analysis. Figure 5.8 shows the vertical displacement field and one component of the bending resultants. The embedded shell is discretized into 16-by-16 quartic B-Spline elements. Since the volume only acts on geometrical quantities and not on the approximation of the displacement field, its refinement level has no influence on the solution. We recover the results obtained with the standard NURBS-based Kirchhoff–Love

formulation (see figure 3.9).

Finally, let us remind that our initial motivation for introducing an embedded shell formulation was to analyze stiffened structures. The presented approach depicted in figure 5.4 leads to non-conforming meshes between the stiffeners and the panel. However, thanks to the Mortar coupling and the Domain Decomposition solver presented in chapter 4, this issue is already tackled. The non-conforming interfaces are tracked in the exact same way than for the standard Kirchhoff–Love shell elements. We have even already presented an example where the embedded Kirchhoff–Love shell were used (see the Scordelis-Lo with curved interfaces in figure 4.19). Thus, we are now able to generate geometric models of stiffened structures and to perform the structural analysis. The examples tackled in the remaining part of this work will demonstrate the good performance of this approach.

5.2.3 Sensitivity Analysis

In order to perform the shape optimization using the presented embedded strategy, we still need to do deal with the sensitivity analysis step. Again, we take for granted all the results presented in chapter 3 that deals with the standard isogeometric approach for shape optimization. The essential points to remember concerning the analytical sensitivity calculation are the following:

1. the shape parametrization links design variables to the control points of the optimization model,
2. the sensitivity propagation between the optimization and the analysis models is done through the refinement matrix,
3. depending on the objective functions and the constraints, adjoint problems need to be solved,
4. analytical derivation of the pseudo-load vector *w.r.t.* to the control point coordinates is achieved by formulating the finite elements using curvilinear kinematics.

Points 2 and 3 remain identical here. For the shape parametrization, we now have two possibilities. We can associate the design variables either to the embedded entities or to the FFD mapping. Consequently, regarding the point 4, the isogeometric operators (stiffness matrix, load vector, etc.) need to be differentiating *w.r.t.* to the control points of either the embedded entities or the mapping transformation.

Shape parametrization Let us take the example of the arch under constant load from figure 3.17 to highlight the two possibilities regarding shape parametrization step. Finally, we already dealt with this issue in figure 5.5 which shows how local and global shape modifications can be imposed.

Firstly, the design variables can be related to the control points of the volume mapping. We call this approach the FFD-like shape update because it recalls Free-Form

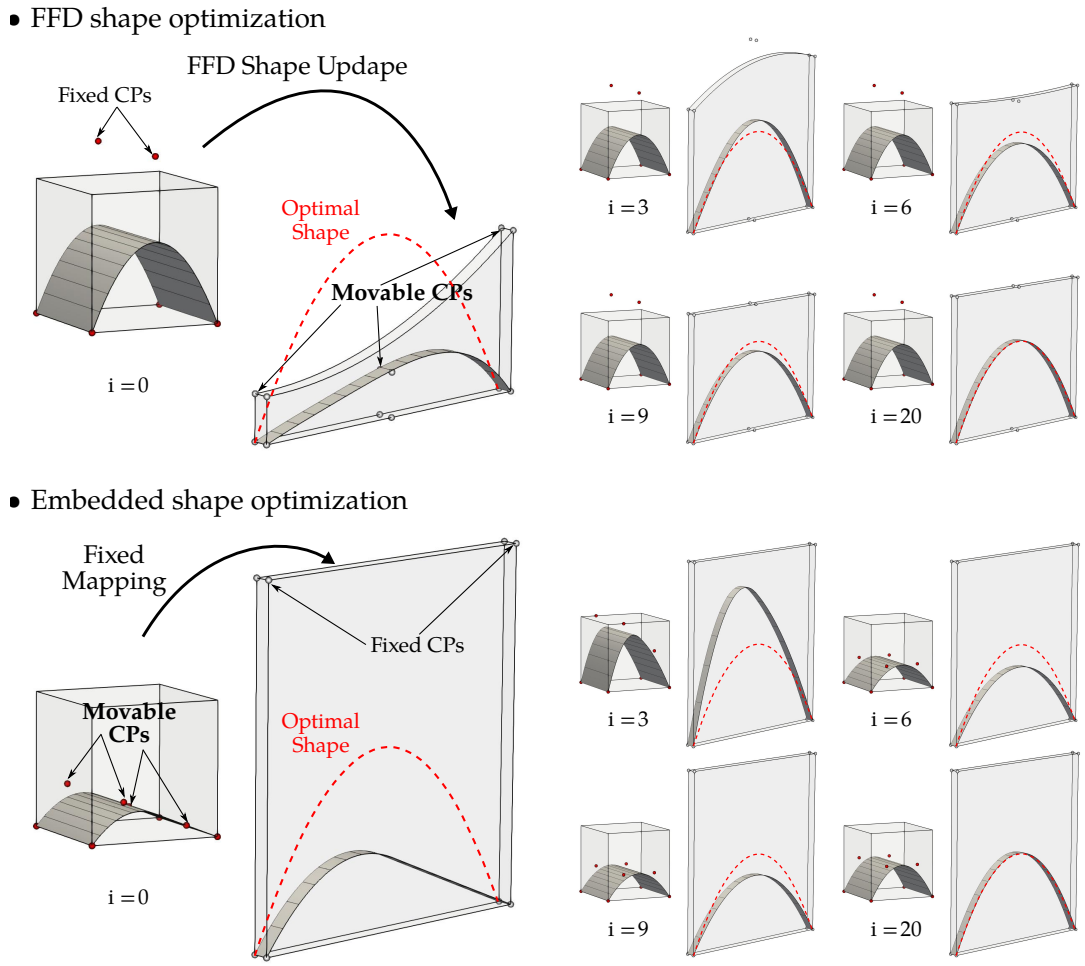


Figure 5.9 – Resolution of the arch optimization problem with embedded Kirchhoff–Love elements in case (top) of FFD-based shape updates, and (bottom) in case of embedded shape updates.

Deformation techniques [197]. Secondly, the design variables can be linked to the control points of the embedded surface in order to modify the shape of the shell. This second approach is referred to the embedded shape update. Figure 5.9 illustrates the resolution of the arch optimization problems with these two techniques. Obviously, the parametrizations used here are not the unique way to solve the problem. For the FFD shape update, we define the embedded surface as a parabola with an arbitrary length to height ratio, because we know, in this particular case, that the final volume we are looking for is a linear mapping that corrects the ratio in order to obtain the expected parabola. Figure 5.9 shows the optimized solution. The final volume has a rectangular shape as predicted. It gives a parabola with the correct height. Note that another initialization than a parabola could have been considered and also lead to the correct optimal design. In case of embedded shape updates, the volume is fixed. We define it suffi-

ciently large to ensure that the design space contains the optimal solution. Figure 5.9 shows that the optimized shape corresponds once again to the analytical solution.

Analytical sensitivities The results depicted in figure 5.9 have been obtained using analytical sensitivities. We need to clarify how this is achievable. As said before, the main theoretical results from chapter 3 are postulated here, and we voluntary skip redundant steps. In fact, the calculation follows the exact same procedures.

Let us only remind few key elements. The goal is to calculate the gradients of a function denoted f *w.r.t.* design variables. After introducing the shape parametrization, the adjoint solution, and the sensitivity propagation, we are at the point where we need to compute the following quantity:

$$\frac{df}{d\boldsymbol{Q}} = \frac{\partial f}{\partial \boldsymbol{Q}} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} - \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u} \right),$$

where $\boldsymbol{Q}$ denotes the control point coordinates of the analysis model (while $\boldsymbol{P}$ denotes those of the optimization model). Thus, in case of NURBS composition, we have both the control points associated to the volume mapping and those associated to the embedded surface. Obviously, the derivatives *w.r.t.* the coordinates of the control points of the mapping are involved in the FFD shape optimization. Conversely, the derivatives *w.r.t.* the coordinates of the control points of the embedded surface enable to perform embedded shape optimization.

Then, let us recall another main result, *i.e.* equation (3.107), that we repeat here for the sake of clarity:

$$\begin{aligned} \boldsymbol{u}^* \cdot \frac{\partial \mathbf{K}}{\partial \boldsymbol{Q}} \boldsymbol{u} = &\int_{\bar{\Omega}_0} \left\{ \left(\frac{\partial \boldsymbol{e}^*}{\partial \boldsymbol{Q}} : \boldsymbol{n} + \boldsymbol{e}^* : \frac{\partial \boldsymbol{n}}{\partial \boldsymbol{Q}} \right) A \; + \; (\boldsymbol{e}^* : \boldsymbol{n}) \frac{\partial A}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega}_0 \\ &+ \int_{\bar{\Omega}_0} \left\{ \left(\frac{\partial \boldsymbol{\kappa}^*}{\partial \boldsymbol{Q}} : \boldsymbol{m} + \boldsymbol{\kappa}^* : \frac{\partial \boldsymbol{m}}{\partial \boldsymbol{Q}} \right) A \; + \; (\boldsymbol{\kappa}^* : \boldsymbol{m}) \frac{\partial A}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\bar{\Omega}_0 \,. \end{aligned}$$

As for the standard Kirchhoff–Love shell, we need to compute the derivatives of the membrane strains and stresses, the bending strains and stresses, and the Jacobian. Once again, this is done similarly for the embedded Kirchhoff–Love shell formulation.

The initial point for computing these derivatives, leads to the derivation of the covariant vectors. In the case of NURBS composition, the two covariant vectors $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are given by equation (5.4). Thus, by differentiating their expression (5.4) *w.r.t.* to the control points, we get:

$$\frac{\partial \boldsymbol{A}_\alpha}{\partial \boldsymbol{Q}} = \sum_{i=1,3} \left(\left(\frac{\partial \bar{\boldsymbol{A}}_\alpha}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_i \right) \boldsymbol{G}_i \; + \; \left(\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_i \right) \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{Q}} \right) \tag{5.11}$$

This is a generic formula. In order to distinguish the derivatives, let also us identify with a bar the control points of the embedded surface of the analysis model. Having the expressions of the embedded covariant vectors $\bar{\boldsymbol{A}}_\alpha$ (given in equation (5.5)), and the

mapping covariant vectors $\boldsymbol{G}_i$ (given in equation (5.6)), we can detail the derivatives involved in equation (5.11). Let us start with the derivatives *w.r.t.* the mapping control points (denoted without the bar superscript). We obtain:

$$\frac{\partial \bar{\boldsymbol{A}}_\alpha}{\partial Q_{ja}} = 0, \qquad \text{and} \qquad \frac{\partial \boldsymbol{G}_i}{\partial Q_{ja}} = R_{a,i}\boldsymbol{e}_j, \tag{5.12}$$

where Q_{ja} denotes the j^{th}-component of the mapping control point a (*i.e.* $Q_{ja} = \boldsymbol{Q}_a \cdot \boldsymbol{e}_j$). Identically, one can calculate those derivatives *w.r.t.* the embedded control points. One should get:

$$\frac{\partial \bar{\boldsymbol{A}}_\alpha}{\partial \bar{Q}_{ja}} = \bar{R}_{a,\alpha}\boldsymbol{e}_j, \qquad \text{and} \qquad \frac{\partial \boldsymbol{G}_i}{\partial \bar{Q}_{ja}} = \bar{R}_a \boldsymbol{G}_{i,j}\,. \tag{5.13}$$

Finally, by introducing equations (5.12) and (5.13) into the general formula (5.11), one can calculate the derivatives *w.r.t.* the control points associated to the volume mapping and the embedded surface, respectively. For the volume mapping, the final expression of these derivatives reads as:

$$\frac{\partial \boldsymbol{A}_\alpha}{\partial Q_{ja}} = \sum_{i=1,3} (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_i)\, R_{a,i}\, \boldsymbol{e}_j. \tag{5.14}$$

For the embedded control points, we obtain:

$$\frac{\partial \boldsymbol{A}_\alpha}{\partial \bar{Q}_{ja}} = \bar{R}_{a,\alpha}\boldsymbol{G}_j \; + \; \sum_{i=1,3} (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_i)\, \bar{R}_a\, \boldsymbol{G}_{i,j}\,. \tag{5.15}$$

We can almost stop here our developments. In fact, at this point, one should be able to compute the derivatives of the membrane strains and stresses, and the derivatives of the Jacobian involved in equation (3.107). As for the structural analysis, implementing the sensitivity analysis for the presented embedded Kirchhoff–Love formulation requires only few modifications when having in hand its implementation for the standard Kirchhoff–Love formulation. One simply has to put the derivatives (5.14) or (5.15) instead of the basis derivatives (3.93) used in the standard case. Only a last term in the bending component still needs to be investigated to get the complete sensitivity. In the differentiation of the bending strains, the derivatives $\partial \boldsymbol{A}_{\alpha,\beta}/\partial \boldsymbol{Q}$ are required. To compute these derivatives, we differentiate equation (5.7). Firstly, let us give a general expression without distinguishing the mapping and embedded control points. We obtain:

$$\begin{aligned}
\frac{\partial \boldsymbol{A}_{\alpha,\beta}}{\partial \boldsymbol{Q}} = & \sum_{i=1,3} \left(\left(\frac{\partial \bar{\boldsymbol{A}}_{\alpha,\beta}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_i \right) \boldsymbol{G}_i \; + \; \left(\bar{\boldsymbol{A}}_{\alpha,\beta} \cdot \boldsymbol{e}_i \right) \frac{\partial \boldsymbol{G}_i}{\partial \boldsymbol{Q}} \right) \\
& + \sum_{i=1,3} \sum_{j=1,3} \left(\left(\frac{\partial \bar{\boldsymbol{A}}_\alpha}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_i \right) (\bar{\boldsymbol{A}}_\beta \cdot \boldsymbol{e}_j) + (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_i) \left(\frac{\partial \bar{\boldsymbol{A}}_\beta}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_j \right) \right) \boldsymbol{G}_{i,j} \\
& + \sum_{i=1,3} \sum_{j=1,3} (\bar{\boldsymbol{A}}_\alpha \cdot \boldsymbol{e}_i)(\bar{\boldsymbol{A}}_\beta \cdot \boldsymbol{e}_j) \frac{\partial \boldsymbol{G}_{i,j}}{\partial \boldsymbol{Q}}.
\end{aligned} \tag{5.16}$$

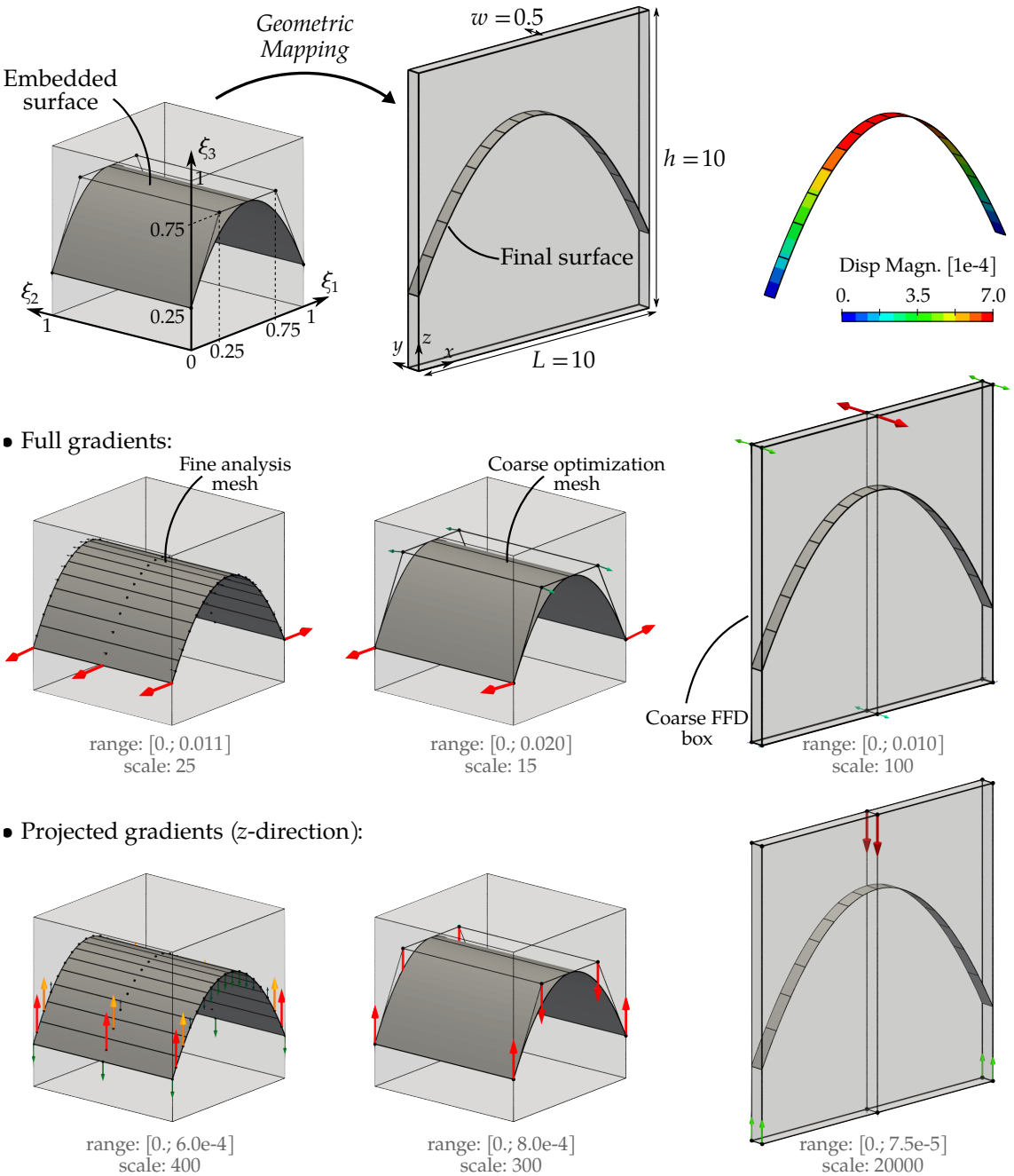


Figure 5.10 – Sensitivity analysis of a parabolic beam using the embedded Kirchhoff–Love shell formulation.

The derivatives of the covariant basis vectors associated to the embedded surface and the volume mapping have already been expressed in equations (5.12) and (5.13). Similarly, let us compute the missing derivatives. For the mapping control points, we have:

$$\frac{\partial \bar{\boldsymbol{A}}_{\alpha,\beta}}{\partial Q_{ka}} = 0, \qquad \text{and} \qquad \frac{\partial \boldsymbol{G}_{i,j}}{\partial Q_{ka}} = R_{a,ij}\boldsymbol{e}_k, \tag{5.17}$$

and the derivatives *w.r.t.* the embedded control points are:

$$\frac{\partial \bar{\boldsymbol{A}}_{\alpha,\beta}}{\partial \bar{Q}_{ka}} = \bar{R}_{a,\alpha}\boldsymbol{e}_k, \qquad \text{and} \qquad \frac{\partial \boldsymbol{G}_{i,j}}{\partial \bar{Q}_{ka}} = \bar{R}_a \boldsymbol{G}_{i,j,k}\,. \tag{5.18}$$

The vector $\boldsymbol{G}_{i,j,k}$ requires third order derivatives of the NURBS basis functions associated to the volume mapping:

$$\boldsymbol{G}_{i,j,k} = \sum_b R_{b,ijk}\boldsymbol{Q}_b\,. \tag{5.19}$$

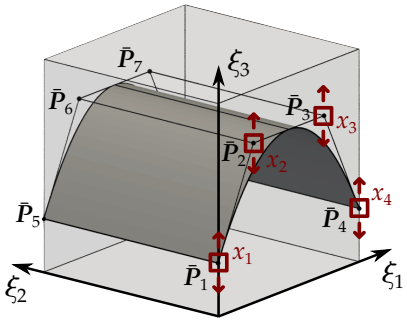
This vector is null in case of a quadratic NURBS volume.

We have now tackled all the differences involved by the embedded approach in comparison with the standard approach. We will not repeat all the equations here, and the interested reader can refer to section 3.3.2.2 to get the overall expression of the analytical sensitivities in case of the embedded Kirchhoff–Love shell.

Validation In order to validate the correct implementation of the gradients, one can perform at first all the examples tackled for the standard IGA-based shape optimization 3.3. For instance, one can define the volume mapping as the identity transformation. Computing the gradients *w.r.t.* the embedded control points should give the exact same results than for the standard approach (*i.e.* with and without the volume mapping).

In order to further validate the sensitivities, we build a parabolic arch using a NURBS composition as shown in figure 5.10. The mechanical settings are those from the arch presented in figure 3.17. The geometric settings are directly given in figure 5.10. The final surface obtained by the NURBS composition describes a parabola of length $L = 10$ and of height $h = 5$. The volume mapping is defined with one single B-Spline element. Degree 2 is set in direction x while degree 1 is set in the two others. The coarse discretization of the embedded surface (optimization model) is made of two B-Spline elements with degree 2 in the main direction and degree 1 along the width. The analysis model is obtained through k-refinement such that the mesh counts 16 quadratic elements. We display both the gradients of the compliance *w.r.t.* the mapping and the embedded control points (given by equation (3.121)).

Again, we can compare these analytical sensitivities with full FD or semi-AN approximations. For this purpose, we study two shape parametrizations: an embedded shape parametrization as presented in figure 5.11, and an FFD shape parametrization as presented in figure 5.12. The results of the sensitivity analysis are given in tables 5.1 and 5.2. For both the embedded and the mapping shape parametrizations, the analytical sensitivities are close to the approximations when a central scheme is used. The



— Shape parametrization:

$$\begin{aligned} \bar{P}_1 &= \bar{P}_1^0 + x_1 e_3 & \bar{P}_5 &= \bar{P}_5^0 + x_1 e_3 \\ \bar{P}_2 &= \bar{P}_2^0 + x_2 e_3 & \bar{P}_6 &= \bar{P}_6^0 + x_2 e_3 \\ \bar{P}_3 &= \bar{P}_3^0 + x_3 e_3 & \bar{P}_7 &= \bar{P}_7^0 + x_3 e_3 \\ \bar{P}_4 &= \bar{P}_4^0 + x_4 e_3 & \bar{P}_8 &= \bar{P}_8^0 + x_4 e_3 \end{aligned}$$

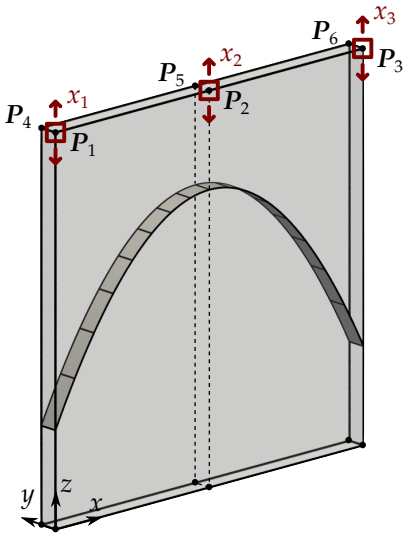
— Sensitivity analysis:

$$\frac{df_{\text{cpl}}}{dx_i} = \frac{df_{\text{cpl}}}{d\bar{P}_i} \cdot e_3 + \frac{df_{\text{cpl}}}{d\bar{P}_{i+4}} \cdot e_3, \quad i = 1, \dots, 4$$

Figure 5.11 – Shape parametrization of the arch through the embedded surface.

Type	df_{cpl}/dx_1	df_{cpl}/dx_2	df_{cpl}/dx_3	df_{cpl}/dx_4
FD (1e-6)	1.60593949	-1.60502817	-1.60502803	1.60593957
central FD (1e-6)	1.60574315	-1.60574318	-1.60574318	1.60574321
sAN (1e-6)	1.60576036	-1.60572716	-1.60572717	1.60576047
central sAN (1e-6)	1.60574314	-1.60574318	-1.60574321	1.60574319
full AN (-)	1.60574316	-1.60574316	-1.60574316	1.60574316

Table 5.1 – Different types of gradients for the arch with the settings from figure 5.10, and with the embedded shape parametrization as in figure 5.11 (factor $\times 1e$-03).



— Shape parametrization:

$$\begin{aligned} P_1 &= P_1^0 + 10 x_1 e_3 & P_4 &= P_4^0 + 10 x_1 e_3 \\ P_2 &= P_2^0 + 10 x_2 e_3 & P_5 &= P_5^0 + 10 x_2 e_3 \\ P_3 &= P_3^0 + 10 x_3 e_3 & P_6 &= P_6^0 + 10 x_3 e_3 \end{aligned}$$

— Sensitivity analysis:

$$\frac{df_{\text{cpl}}}{dx_i} = 10 \frac{df_{\text{cpl}}}{dP_i} \cdot e_3 + 10 \frac{df_{\text{cpl}}}{dP_{i+3}} \cdot e_3, \quad i = 1, \dots, 3$$

Figure 5.12 – Shape parametrization of the arch through the volume mapping.

Type	df_{cpl}/dx_1	df_{cpl}/dx_2	df_{cpl}/dx_3
FD (1e-7)	-0.07765254	-1.45041246	-0.07765202
central FD (1e-5)	-0.07766358	-1.45041602	-0.07766357
sAN (1e-6)	-0.07766173	-1.45041168	-0.07766161
central sAN (1e-5)	-0.07766357	-1.45041602	-0.07766357
full AN (-)	-0.07766357	-1.45041602	-0.07766357

Table 5.2 – Different types of gradients for the arch with the settings from figure 5.10, and with the mapping shape parametrization as in figure 5.12 (factor $\times 1e$-03).

results are exactly the same up to 7^{th} decimal. With a forward Finite Difference scheme, both the full DF or the semi-AN sensitivities are less accurate. For example, in case of the FFD shape parametrization 5.12, the relative error for the first design variable (*i.e.* df_{cpl}/dx_1, line corresponding to DF) is equal to 1.4e-4. Even by changing the step length in the approximation, we do not achieve to improve the accuracy. We observe the same limit for the semi-AN sensitivities. Perhaps the precision suffers from the inconsistency regarding the rigid-body test as discussed in section 3.1.3 (see more specifically the paragraph about semi-AN sensitivities page 57).

Thus, this study regarding the sensitivities enables to validate the computation of analytical gradients for both the embedded and the mapping shape parametrizations. It also show that the semi-Analytical with forward FD approximations (which is the cheapest approximated gradient) is only correct up to a certain precision. The inaccuracy of the gradient can have bad influence on the convergence during the optimization. For example, unwanted dissymmetries may appear during the iteration of the algorithm because of the accumulation of small (but not insignificant) errors. Finally, let us mention that, within our *code*, the analytical sensitivity requires the lowest computational time in comparison with the other types of gradients. For this simple example of the arch, the difference is not important. However, for larger problem we observe major gains.

5.3 Potential for designing multipatch shell structures

Now, we want to highlight the potential of the presented embedded approach for shape optimization. We show here how it allows to optimize multipatch structures that contain non-conforming meshes. Specifically, we focus on stiffened structures.

5.3.1 Sensitivity analysis of non-conforming multipatch structures

A last ingredient is required to be able to optimize complex multipatch structures. So far, we have not integrated the Mortar coupling in our optimization process. Especially,

we need to take a look at the computation of the sensitivities when several patches are coupled through the Mortar method (as introduced earlier, see section 4.2.2).

5.3.1.1 Using Mortar coupling in optimization

Let us take the simple case of two patches coupled with the Mortar method. We remind the coupled linear system to be solved (see equation (4.16)). It takes the following form:

$$\begin{bmatrix} \mathbf{K}^{(1)} & 0 & \mathbf{B}^{(1)T} \\ 0 & \mathbf{K}^{(2)} & \mathbf{B}^{(2)T} \\ \mathbf{B}^{(1)} & \mathbf{B}^{(2)} & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}^{(1)} \\ \boldsymbol{F}^{(2)} \\ 0 \end{pmatrix}, \tag{5.20}$$

where the coupling operators $\mathbf{B}^{(s)}$ can be computed by equation (4.29) in case of Kirchhoff–Love shells.

In order to perform the sensitivity analysis, let us also remind, once again, the generic formula for adjoint-based discrete sensitivities (see equation (3.15)):

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \boldsymbol{u}^* \cdot \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i} \boldsymbol{u} \right).$$

In case of two patches coupled with the Mortar method, the state variables, the adjoint solution, the global load vector, and the global stiffness matrix are respectively given by:

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \boldsymbol{\lambda} \end{pmatrix}, \quad \boldsymbol{u}^* = \begin{pmatrix} \boldsymbol{u}^{*(1)} \\ \boldsymbol{u}^{*(2)} \\ \boldsymbol{\lambda}^* \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \boldsymbol{F}^{(1)} \\ \boldsymbol{F}^{(2)} \\ 0 \end{pmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}^{(1)} & 0 & \mathbf{B}^{(1)T} \\ 0 & \mathbf{K}^{(2)} & \mathbf{B}^{(2)T} \\ \mathbf{B}^{(1)} & \mathbf{B}^{(2)} & 0 \end{bmatrix}. \tag{5.21}$$

Then, one can get the expression of the sensitivity by substituting these expressions into the general formulation of the adjoint-based discrete sensitivities (3.15) that we just repeated. For two patches coupled with the Mortar method, we have:

$$\begin{aligned} \frac{df}{dx_i} = \frac{\partial f}{\partial x_i} &+ \boldsymbol{u}^{*(1)} \cdot \left(\frac{\partial \boldsymbol{F}^{(1)}}{\partial x_i} - \frac{\partial \mathbf{K}^{(1)}}{\partial x_i} \boldsymbol{u}^{(1)} \right) + \boldsymbol{u}^{*(2)} \cdot \left(\frac{\partial \boldsymbol{F}^{(2)}}{\partial x_i} - \frac{\partial \mathbf{K}^{(2)}}{\partial x_i} \boldsymbol{u}^{(2)} \right) \\ &- \boldsymbol{\lambda}^* \cdot \left(\frac{\partial \mathbf{B}^{(1)}}{\partial x_i} \boldsymbol{u}^{(1)} + \frac{\partial \mathbf{B}^{(2)}}{\partial x_i} \boldsymbol{u}^{(2)} \right) - \boldsymbol{\lambda} \cdot \left(\frac{\partial \mathbf{B}^{(1)}}{\partial x_i} \boldsymbol{u}^{*(1)} + \frac{\partial \mathbf{B}^{(2)}}{\partial x_i} \boldsymbol{u}^{*(2)} \right). \end{aligned} \tag{5.22}$$

The coupling between the patches introduces additional terms where the derivatives of the coupling operator *w.r.t.* the design variables are involved. We will show that these terms are computed analogously to the derivatives of the stiffness matrices (already tackled). Depending on the shape parametrization, these coupling terms in the derivative (5.22) vanish.

However, prior to that, let us tackle the specific case of the compliance. Its initial expression (3.9) reads in the case of two weakly coupled sub-domains as:

$$f_{\text{cpl}} \;=\; f_{\text{cpl}}^{(1)} \;+\; f_{\text{cpl}}^{(2)} \;=\; \frac{1}{2}\boldsymbol{F}^{(1)}\cdot\boldsymbol{u}^{(1)} \;+\; \frac{1}{2}\boldsymbol{F}^{(2)}\cdot\boldsymbol{u}^{(2)} \tag{5.23}$$

Equivalently to equations (3.17) and (3.18), one can calculate the partial derivatives $\partial f_{\text{cpl}}/\partial x_i$ and the adjoint solution. By putting the obtain quantities into the expression (5.22), one should get the following formula of the sensitivity of the compliance:

$$\frac{df_{\text{cpl}}}{dx_i} = \frac{\partial f_{\text{cpl}}^{(1)}}{\partial x_i} + \frac{\partial f_{\text{cpl}}^{(2)}}{\partial x_i} - \boldsymbol{\lambda}\cdot\frac{\partial \mathbf{B}^{(1)}}{\partial x_i}\boldsymbol{u}^{(1)} - \boldsymbol{\lambda}\cdot\frac{\partial \mathbf{B}^{(2)}}{\partial x_i}\boldsymbol{u}^{(2)}, \tag{5.24}$$

with:

$$\frac{\partial f_{\text{cpl}}^{(s)}}{\partial x_i} = \boldsymbol{u}^{(s)}\cdot\left(\frac{\partial \boldsymbol{F}^{(s)}}{\partial x_i} - \frac{1}{2}\frac{\partial \mathbf{K}^{(s)}}{\partial x_i}\boldsymbol{u}^{(s)}\right). \tag{5.25}$$

Up to now, we are able to compute analytically the term in equation (5.25). However, to get the complete gradient, we need to deal with the Mortar operators.

5.3.1.2 Shape parametrization

We need to look at the shape parametrization that requires some attention to be able to perform the shape optimization with multiple non-conforming patches. The principle does not change. The design variables move the control points of the different patches as defined by equation (3.73):

$$\boldsymbol{P}_k = \boldsymbol{P}_k^0 + \sum_{i\in\mathcal{D}_k} x_i \boldsymbol{n}_i^k.$$

However, this shape parametrization should also be aware of updating the interface curves. In fact, while modifying the shape of the different patches, the non-conforming interfaces may change. The problem is that the link between the design variables and the control points $\check{\boldsymbol{P}}$ of the interface curves may not be easy to get. In general, this link cannot be explicitly formulated. Thus, during the sensitivity analysis, one option is to approximate the derivatives that cannot be explicitly calculated. For instance:

$$\frac{\partial \check{\boldsymbol{P}}}{\partial x_i} \approx \frac{\check{\boldsymbol{P}}(x_i+\Delta x_i) - \check{\boldsymbol{P}}(x_i)}{\Delta x_i}, \tag{5.26}$$

where $\check{\boldsymbol{P}}(x_i+\Delta x_i)$ are the control points of the interface curves after perturbing the overall shape by applying a small change to the design variable x_i. Those control points may be obtained from a surface-to-surface intersection problem (CAD task).

Fortunately, with the presented embedded approach we have a suitable way to circumvent this issue. It is possible to define a shape parametrization that explicitly takes

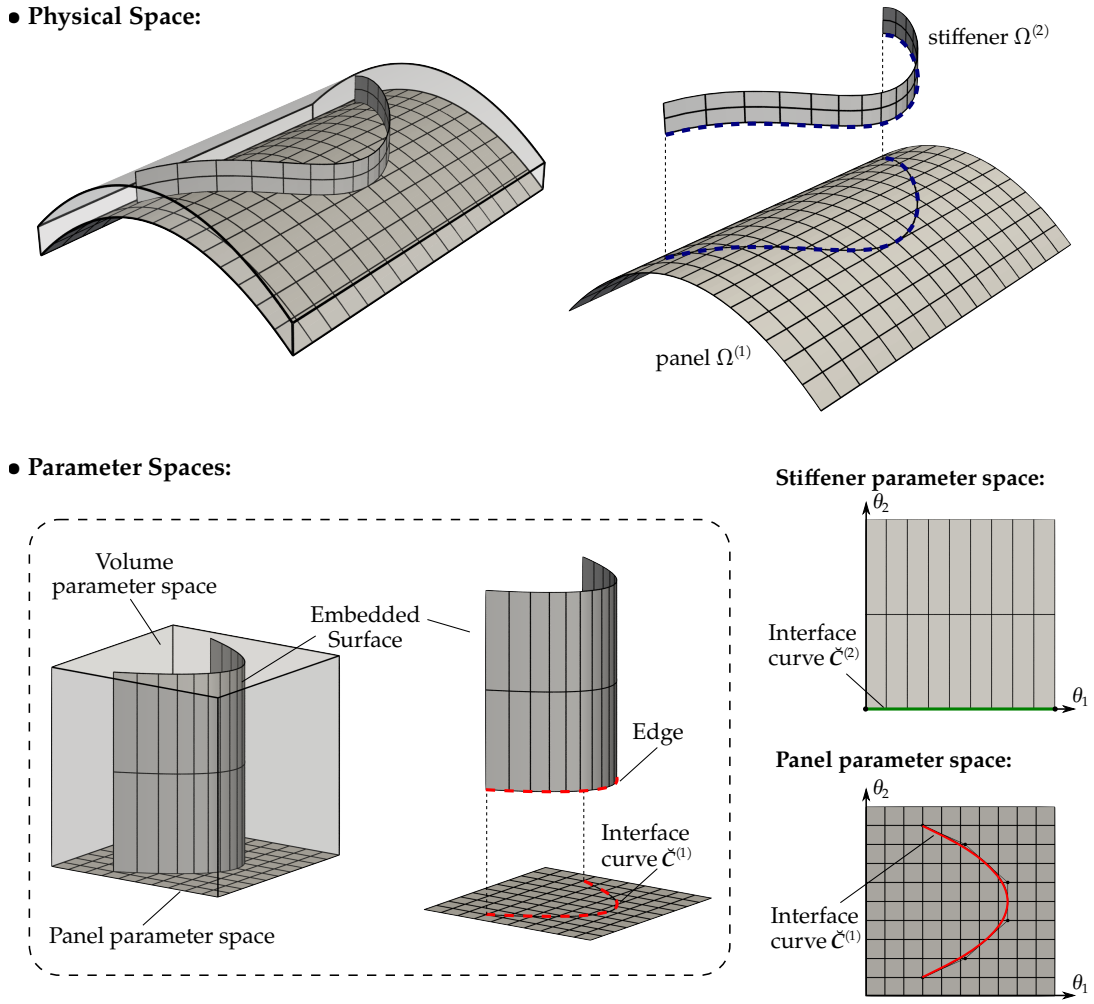


Figure 5.13 – Definition of the interface curves with the embedded approach. For the stiffener, the interface curve $\check{C}^{(2)}$ lies on one edge of the embedded surface. For the panel, the interface curve $\check{C}^{(1)}$ can be extracted from the embedded surface.

into account the update of the interface curves. This is made possible within the geometric modeling process introduced previously in figure 5.4. To highlight this point, we refer to figure 5.13. In our construction of stiffened structures, one face of the volume mapping (in which the stiffener is embedded) is matching with the surface describing the panel. Therefore, we can locate the parameter space of the panel in the parameter space of this volume. Then, one can notice in figure 5.13 that the embedded surface is lying on the parameter space of the panel. Hence, the interface curve on the panel side can be extracted from the embedded surface. We can build a B-Spline curve that is fully matching with the edge embedded surface:

$$\check{C}^{(1)}(\xi) = \bar{S}^{(2)}(\xi, 0), \quad \forall \xi. \tag{5.27}$$

For the stiffener, the construction of the interface curve is straightforward:

$$\check{\boldsymbol{C}}^{(2)}(\xi) = (\xi \;\; 0), \quad \forall \xi. \tag{5.28}$$

It can be defined using a linear NURBS curve with two control points. For the numerical integration, both interface curves are further refined.

Thus, the control points of the panel interface curve can be explicitly connected to the movement of the stiffener. The design variables move the panel interface curve and the edge of the embedded surface in the same way. On the other side, the stiffener interface curve usually does not need to be updated. For instance, the same edge of the embedded surface may represent the interface during the whole optimization process. This point could be foreseen in figure 5.5. Of course, this issue cannot be generalized and is specific to the stiffened structures generated through the presented strategy.

5.3.1.3 Differentiating the Mortar operators

The differentiation of the Mortar coupling matrices is done in the same manner as for the stiffness matrices. For instance, by invoking the chain rule, we express the derivatives *w.r.t.* the design variables using the derivatives *w.r.t.* the control point coordinates of the optimization model. Then, the refinement matrix is introduced in order to propagate the sensitivity between the optimization model and the analysis model (see the procedure from section 3.3.1). These chain rules bring us to calculate quantities of the type:

$$\boldsymbol{\lambda} \cdot \frac{\partial \mathbf{B}^{(s)}}{\partial \boldsymbol{Q}} \boldsymbol{u}^{(s)}.$$

We do not differentiate direct and adjoint quantities in what follows, but one can consider the vectors of DOF, $\boldsymbol{\lambda}$ or $\boldsymbol{u}$, as being either one or the other. We also voluntarily skip the sub-domain identification through superscript (s) because the expression is the same independently of the sub-domain.

The derivatives of the Mortar operators *w.r.t.* the control points of the analysis model is performed component-by-component:

$$\boldsymbol{\lambda} \cdot \frac{\partial \mathbf{B}}{\partial \boldsymbol{Q}} \boldsymbol{u} = \sum_k \sum_l \lambda_k \frac{\partial \mathbf{B}_{kl}}{\partial \boldsymbol{Q}} u_l, \tag{5.29}$$

where λ_k and u_l are DOFs of the Lagrange multipliers and the sub-domain displacement field, respectively. $\mathbf{B}_{kl}$ is the component of matrix $\mathbf{B}$ related to these DOFs.

Displacement continuity To go further, let us firstly take the case of the Mortar matrix $\mathbf{C}$ that imposes the continuity of the displacement field at the interface (see equation (4.20)). We repeat its expression for the sake of clarity:

$$\mathbf{C}_{kl} = \pm \int_{[0,1]} \tilde{R}_a(\xi)\, R_b\big(\bar{\boldsymbol{C}}(\xi)\big)\, \delta_{ij}\, |\boldsymbol{T}(\xi)|\, \mathrm{d}\xi.$$

We refer to equation (4.20) and its associated relations to fully understand the notation if needed (especially equation (4.19)). The derivatives of the coupling matrix $\mathbf{C}$ *w.r.t.* the control points is given by:

$$\frac{\partial \mathbf{C}_{kl}}{\partial \boldsymbol{Q}} = \pm \int_{[0,1]} \left\{ \tilde{R}_a \frac{\partial R_b}{\partial \boldsymbol{Q}} |\boldsymbol{T}| + \tilde{R}_a R_b \frac{\partial |\boldsymbol{T}|}{\partial \boldsymbol{Q}} \right\} \delta_{ij} \, \mathrm{d}\xi \,. \tag{5.30}$$

Then, we can put this last equation into equation (5.29). By commuting the double sum and the integral, we get:

$$\tilde{\boldsymbol{\lambda}} \cdot \frac{\partial \mathbf{C}}{\partial \boldsymbol{Q}} \boldsymbol{u} = \pm \int_{[0,1]} \left\{ \left(\tilde{\lambda}^h \cdot \frac{\partial u^h}{\partial \boldsymbol{Q}} \right) |\boldsymbol{T}| \; + \; \left(\tilde{\lambda}^h \cdot u^h \right) \frac{\partial |\boldsymbol{T}|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\xi \,. \tag{5.31}$$

where:

— $\tilde{\lambda}^h$ is the discretized Lagrange multiplier:

$$\tilde{\lambda}^h(\xi) = \sum_a \tilde{R}_a(\xi) \, \tilde{\boldsymbol{\lambda}}_a, \tag{5.32}$$

— u^h is the discretized displacement field:

$$u^h(\theta_1, \theta_2) = \sum_b R_b(\theta_1, \theta_2) \, \boldsymbol{u}_b \tag{5.33}$$

The derivative of the displacement field *w.r.t.* to the control point coordinates is involved in equation (5.31) because of the change of variables that has been made to perform the numerical integration. In fact, we have introduced embedded curves to represent the interfaces on the parameter space of the patch. Let us give the formal expression of the derivative for clarification:

$$\frac{\partial u^h}{\partial \boldsymbol{Q}} = \sum_b \frac{\partial R_b}{\partial \boldsymbol{Q}} (\bar{\boldsymbol{C}}(\xi)) \boldsymbol{u}_b \tag{5.34}$$

Clearly, the derivative is non-zero only for the control points of the interface curve. Similar derivatives for the Lagrange multiplier (*i.e.* $\partial \tilde{\lambda}^h / \partial \boldsymbol{Q}$) might be included in equation (5.31) if it was not express as a function of the interface curvilinear parameter ξ.

— $|\boldsymbol{T}|$ is the Jacobian of the change of variables made in the integral. It is interesting to look at the term that multiplies the derivatives $\partial |\boldsymbol{T}| / \partial \boldsymbol{Q}$. Especially, in case of two sub-domains where the Mortar method is applied to enforce the displacement continuity at the interface, the following condition is weakly imposed:

$$u^{h(1)} - u^{h(2)} = 0.$$

As a consequence, if equivalent derivatives of the Jacobian $\partial |\boldsymbol{T}| / \partial \boldsymbol{Q}$ are involved for each sub-domains, summing the mentioned terms leads to zero. Depending on the shape parametrization, the overall sensitivity coming from the Mortar coupling vanishes. We will return to this point with the help of illustrative examples.

In order to finalize the expression of the sensitivity, let us give the resulting derivatives for the displacement field. Using the preliminary result from equation (5.34), we get:

$$\frac{\partial u^h}{\partial \boldsymbol{Q}} = \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_1\right) u^h_{,1} \; + \; \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_2\right) u^h_{,2} \tag{5.35}$$

with

$$u^h_{,\alpha} = \frac{\partial u^h}{\partial \theta_\alpha} = \sum R_{b,\alpha} \boldsymbol{u}_b \,. \tag{5.36}$$

Let us distinguish the nature of the control points of the analysis model as follows. In case of a standard IGA element (solid or Kirchhoff–Love shell), the control points number a of the patch is denoted $\boldsymbol{Q}_a$. For the embedded Kirchhoff–Love shell, we add a bar in order to identify the control points of the embedded surface (*i.e.* $\boldsymbol{Q}_a$ is associated to the volume mapping, and $\bar{\boldsymbol{Q}}_a$ to the embedded surface). We denote the control points of the interface curves as $\check{\boldsymbol{Q}}_a$. With these notations, we can give the derivative of the interface curve *w.r.t.* the several types of control points. We have for standard elements:

$$\frac{\partial \check{\boldsymbol{C}}}{\partial \check{Q}_{\alpha a}} = \check{R}_a \boldsymbol{e}_\alpha, \qquad \frac{\partial \check{\boldsymbol{C}}}{\partial \boldsymbol{Q}_{ia}} = 0, \tag{5.37}$$

and for the embedded shell:

$$\frac{\partial \check{\boldsymbol{C}}}{\partial \check{Q}_{\alpha a}} = \check{R}_a \boldsymbol{e}_\alpha, \qquad \frac{\partial \check{\boldsymbol{C}}}{\partial \bar{Q}_{ia}} = 0, \qquad \frac{\partial \check{\boldsymbol{C}}}{\partial Q_{ia}} = 0. \tag{5.38}$$

We deduce the following relation:

$$\tilde{\lambda}^h \cdot \frac{\partial u^h}{\partial \check{Q}_{\alpha a}} = \check{R}_a \left(u^h_{,\alpha} \cdot \tilde{\lambda}^h\right) \tag{5.39}$$

which can be then put in equation (5.31).

Finally, we need to describe the derivatives of the Jacobian $\partial|\boldsymbol{T}|/\partial\boldsymbol{Q}$. Let us remind equation (4.24) where the expression of the vector $\boldsymbol{T}$ is given:

$$\boldsymbol{T} = \left(\check{\boldsymbol{C}}_{,\xi} \cdot \boldsymbol{e}_1\right) \boldsymbol{A}_1 + \left(\check{\boldsymbol{C}}_{,\xi} \cdot \boldsymbol{e}_2\right) \boldsymbol{A}_2.$$

By using the relation $|\boldsymbol{T}| = (\boldsymbol{T} \cdot \boldsymbol{T})^{1/2}$, one can derive the Jacobian and get the following result:

$$\frac{\partial |\boldsymbol{T}|}{\partial \boldsymbol{Q}} = \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{t}, \tag{5.40}$$

where $\boldsymbol{t} = \boldsymbol{T}/|\boldsymbol{T}|$ is the unit vector tangent to the interface curve that has been already introduced to impose the continuity of the rotation for the coupling of shells. The derivation of the vector $\boldsymbol{T}$ reads as:

$$\frac{\partial \boldsymbol{T}}{\partial \boldsymbol{Q}} = \left(\frac{\partial \check{\boldsymbol{C}}_{,\xi}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_1\right) \boldsymbol{A}_1 + \left(\frac{\partial \check{\boldsymbol{C}}_{,\xi}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_2\right) \boldsymbol{A}_2 \; + \; \left(\check{\boldsymbol{C}}_{,\xi} \cdot \boldsymbol{e}_1\right) \frac{\partial \boldsymbol{A}_1}{\partial \boldsymbol{Q}} + \left(\check{\boldsymbol{C}}_{,\xi} \cdot \boldsymbol{e}_2\right) \frac{\partial \boldsymbol{A}_2}{\partial \boldsymbol{Q}}. \tag{5.41}$$

At this point, we need to gather all the previous results. In fact, we already know how to compute several partial derivatives involved in the derivation of the Jacobian. For instance, we know how to compute the derivatives of the covariant vectors *w.r.t.* the control points related to the shell mid-surface (see equation (3.93) for the standard elements, and equations (5.12) and (5.13) for the embedded approach). We require to give the derivatives of the covariant vectors *w.r.t.* to the control points of the interface curve. We have:

$$\frac{\partial \boldsymbol{A}_\alpha}{\partial \check{\boldsymbol{Q}}_a} = \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \check{\boldsymbol{Q}}_a} \cdot \boldsymbol{e}_1 \right) \boldsymbol{A}_{\alpha,1} \; + \; \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \check{\boldsymbol{Q}}_a} \cdot \boldsymbol{e}_2 \right) \boldsymbol{A}_{\alpha,2}. \tag{5.42}$$

Then, recalling equation (5.37) (or (5.38) for the embedded shell), we get the desired result:

$$\frac{\partial \boldsymbol{A}_\alpha}{\partial \check{Q}_{\beta a}} = \check{R}_a \boldsymbol{A}_{\alpha,\beta}. \tag{5.43}$$

To get the full expression of the derivative of the Jacobian, we need to express the last intermediary formulas:

$$\frac{\partial \check{\boldsymbol{C}}_{,\xi}}{\partial \check{Q}_{\alpha a}} = \check{R}_{a,\xi} \boldsymbol{e}_\alpha \,, \qquad \frac{\partial \check{\boldsymbol{C}}_{,\xi}}{\partial \bar{Q}_{ia}} = 0 \,, \qquad \frac{\partial \check{\boldsymbol{C}}_{,\xi}}{\partial Q_{ia}} = 0. \tag{5.44}$$

We now have all the ingredients to compute the analytical sensitivity as formulated in equation (5.31).

Rotation continuity What has been done for the Mortar operator that enforces the continuity of the displacement can be generalized. For instance, in the case of the Mortar operator that enforces the continuity of the rotation, the sensitivity reads as:

$$\hat{\boldsymbol{\lambda}} \cdot \frac{\partial \mathbf{Z}}{\partial \boldsymbol{Q}} \boldsymbol{u} = \pm \int_{[0,1]} \left\{ \hat{\lambda}^h \frac{\partial}{\partial \boldsymbol{Q}} \left(\Phi^h \cdot \boldsymbol{t} \right) |\boldsymbol{T}| \; + \; \hat{\lambda}^h (\Phi^h \cdot \boldsymbol{t}) \frac{\partial |\boldsymbol{T}|}{\partial \boldsymbol{Q}} \right\} \mathrm{d}\xi \,. \tag{5.45}$$

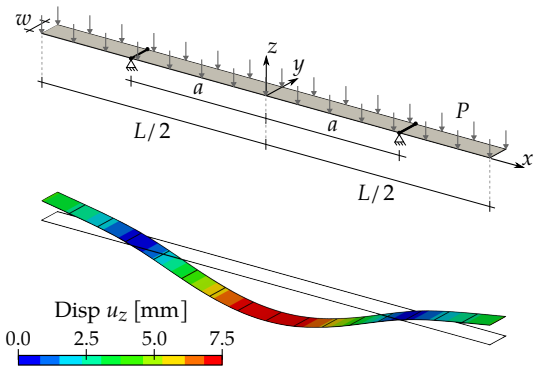
where Φ^h is the rotation vector as given by equation (3.47) applied to the discretized displacement field:

$$\Phi^h = \frac{1}{A} \left(u^h_{,2} \cdot \boldsymbol{A}_3 \right) \boldsymbol{A}_1 - \frac{1}{A} \left(u^h_{,1} \cdot \boldsymbol{A}_3 \right) \boldsymbol{A}_2. \tag{5.46}$$

Without giving the details, let us mention that:

$$\frac{\partial}{\partial \boldsymbol{Q}} \left(\Phi^h \cdot \boldsymbol{t} \right) |\boldsymbol{T}| + (\Phi^h \cdot \boldsymbol{t}) \frac{\partial |\boldsymbol{T}|}{\partial \boldsymbol{Q}} \;\; = \;\; \frac{\partial \Phi^h}{\partial \boldsymbol{Q}} \cdot \boldsymbol{T} + \Phi^h \cdot \frac{\partial \boldsymbol{T}}{\partial \boldsymbol{Q}}, \tag{5.47}$$

which enables to simplify a bit the calculation of the sensitivity. At this point, we only need to mix all the presented results to get the full expression of the sensitivity. In fact, we just formulate the derivatives of the vector $\boldsymbol{T}$ in the foregoing (see equation (5.41) and the related ones). For the derivatives of the discretized rotation vector, one has to apply a chain rule to equation (5.46). We know how to compute the derivatives of the covariant



Parameters:

Mechanical

E = 1 GPa, ν = 0.00 , P = 0.1 MPa

L = 1 m, w = 0.05 m, t = 0.05 m

Optimization problem

minimize compliance

a_{opt} = 0.27526 m

- Sensitivity Analysis

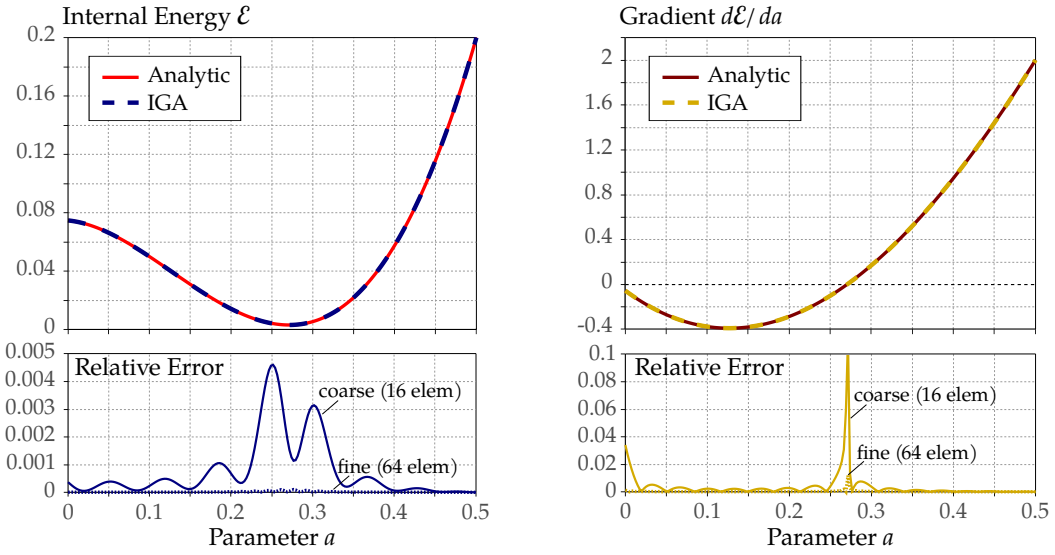


Figure 5.14 – Definition of the interface curves with the embedded approach.

vectors. In addition, we have seen that the derivatives of the area A and the director vector $\boldsymbol{A}_3$ *w.r.t.* the control points are also expressed in terms of the derivatives of the covariant vectors. Finally, the only intermediate result that is missing is the following:

$$\frac{\partial u^h_{,\alpha}}{\partial \boldsymbol{Q}} = \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_1\right) u^h_{,\alpha 1} + \left(\frac{\partial \check{\boldsymbol{C}}}{\partial \boldsymbol{Q}} \cdot \boldsymbol{e}_2\right) u^h_{,\alpha 2}\,, \tag{5.48}$$

which is similar to the result from equation (5.35). Hence, by merging all the mentioned results, one should be able to calculate the derivative of the rotation vector *w.r.t.* every type of control points. Thus, as for the Mortar operator that enforces the continuity of the displacement, it is also possible to compute the analytical sensitivity for the Mortar operator that enforces the continuity of the rotation.

Validation The correct computation of the sensitivities with the Mortar operator can be verified on simple examples. One can for instance take a simple planar beam subjected to an uniform pressure and supported at two locations as depicted in figure 5.14. Here, the beam is described using a single patch and the displacement boundary conditions are imposed through the Mortar operators. The optimization problem consists in finding the best position of the supports that minimizes the compliance. Hence, the shape parametrization imposes the translation of the interface curve along the beam. One single design variable, which correspond to parameter a in figure 5.14, is defined.

Analytical formulation of the internal energy can be obtained. The beam is in pure bending deformation, and thus, the strain energy is reduced to the integration of one single term:

$$\mathcal{E}_{an} = \frac{1}{2}\int_{\bar{\Omega}} m_{xx}\kappa_{xx}\, dxdy. \tag{5.49}$$

Let us give the expression of the non-zero component of the bending moments. Its expression depends on the position x along the beam. We have:

$$m_{xx} = \frac{1}{2}\left(L/2 - x\right)^2 P \; - \; \frac{1}{2}L(a-x)P, \qquad 0 \leq x \leq a, \tag{5.50}$$

$$m_{xx} = \frac{1}{2}\left(L/2 - x\right)^2 P, \qquad a \leq x \leq L/2. \tag{5.51}$$

The bending strain κ_{xx} is related to the bending moment as follows:

$$\kappa_{xx} = \frac{1}{EI} m_{xx}, \quad \text{with} \quad I = t^3/12. \tag{5.52}$$

With the expression of the bending moment in hand, we are now able to calculate analytically the strain energy:

$$\mathcal{E}_{an} = \frac{w}{EI}\int_0^a {m_{xx}}^2 dx + \frac{w}{EI}\int_a^{L/2} {m_{xx}}^2 dx, \tag{5.53}$$

which gives:

$$\mathcal{E}_{an} = \frac{wP^2}{EI}\left(\frac{1}{20}\left(\frac{L}{2}-a\right)^5 + \frac{1}{4}\left(\frac{a^5}{5} + \frac{2}{3}a^3c + c^2a\right)\right), \quad c = L^2/4 - aL. \tag{5.54}$$

It is possible to get the analytic expression of the derivative of the internal energy *w.r.t.* the parameter a. One has to derive equation (5.54), and should get the following:

$$\frac{d\mathcal{E}_{an}}{da} = \frac{wP^2}{4EI}\left(-\left(\frac{L}{2}-a\right)^4 + \; a^4 + 2a^2c - \frac{2}{3}a^3L + c^2 - 2caL\right), \tag{5.55}$$

which can possibly be simplified.

Figure 5.14 gives the comparison between the analytic solution and the numerical results. We show the good agreement of the IGA results for the internal energy and

for the gradient. We display the relative errors when varying the parameter a for two discretizations: a coarse mesh with 16 quartic elements and a fine mesh with 64 quartic elements. The relative error for the gradient presents a pic near the optimal solution. This is normal since at this location, the gradient is zero. However, the numerical gradient does not give exactly zero, and thus the relative error is high at this specific location (infinite because dividing by zero). That being said, this study allows to validate the computation of the sensitivities of the Mortar approach as presented in equation (5.31). Since the shape of the beam does not change in this problem, only the derivatives of the Mortar operator is involved in the sensitivity of the compliance. Looking at the results presented in figure 5.14, one can notice that we fully recover the analytic results. The errors are small and they decrease with the mesh refinement. The computation of the sensitivities can be done with the standard Kirchhoff–Love formulation or with the embedded formulation. One can perform similar studies where the rotations are also involved in order to validate the expression and implementation of the sensitivities of the Mortar operator $\mathbf{Z}$. Such a study is not presented here, but the remaining examples will allow us to validate the computation of the sensitivities with the rotation continuity.

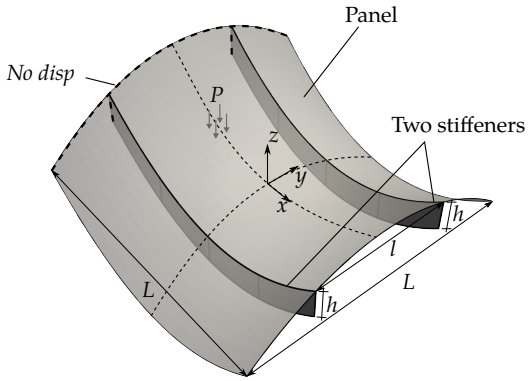
5.3.2 Local optimization of specific parts of a complete structure

It is time to use the developed strategy for shape optimization. We tackle here the local optimization strategy based on what has been presented in figure 5.5. To this end, we deal with the optimization of a hyperbolic paraboloid shell with stiffeners. This example is presented in figure 5.15. A similar test case can be found in Dornisch et al. [204] without shape optimization.

Settings The goal is to design the stiffeners in order to increase the overall stiffness of the structure. To do this, we apply the embedded strategy that enables to parametrize the shape of the stiffeners in an effective manner. The construction of the model is highlighted in figure 5.15. The principle remains the same: starting with the NURBS surface representing the panel, a NURBS volume is built by extrusion. Then, NURBS surfaces are embedded into the parameter space of the extruded volume. The final surfaces that represents the stiffeners are obtained by the composition of these embedded surfaces and the volume mapping. We give in figure 5.15, the control point coordinates of the volume mapping as used here. The patch contains one single element with quadratic degree in the two principal direction (those coming from the panel), whereas linear degree is set in the extrusion direction. The embedded surfaces are modeled using patches with four elements each. Degree two and one are set in the direction 1 and 2, respectively (accordingly to figure 5.15). Uniform knot vectors are used. The data given in figure 5.15 should be sufficient for anyone interested in carrying out the same study.

In order to impose shape variations to the stiffeners, we link several design variables to the control points of the embedded surfaces. Here, we choose to impose identical

• Problem defintion



Parameters:

Mechanical

E = 210 GPa, ν = 0.30 , P = 0.1 MPa

L = 1 m, l = 0.5 m, h = 0.125 m

t_p = 0.04 m, t_s = 0.02 m

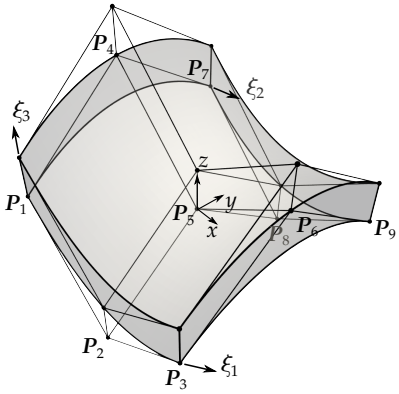
Optimization problem

minimize compliance

$A \leq 1.05 A_0$

$0. \leq x_i \leq 0.40$

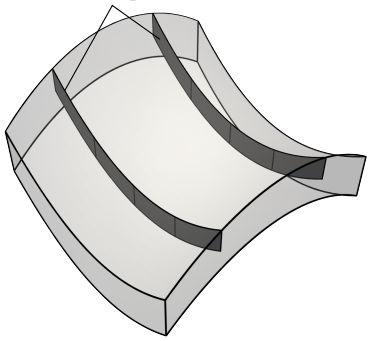
• Geometric modeling



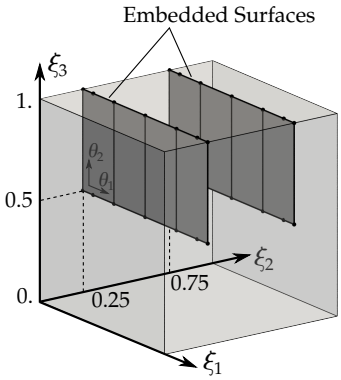
	x	y	z
P_1	-0.5	-0.5	0.
P_2	0.	-0.5	-0.5
P_3	0.	-0.5	0.
P_4	-0.5	0.	0.5
P_5	0.	0.	0.
P_6	0.5	0.	0.5
P_7	-0.5	0.5	0.
P_8	0.	0.5	-0.5
P_9	0.5	0.5	0.

$P_{i+9} = P_9 + 0.25 e_z$

Stiffeners obtained by NURBS composition



• Shape parametrization



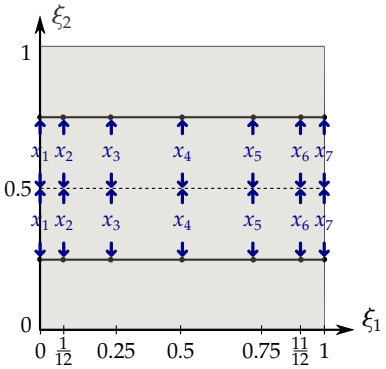


Figure 5.15 – Settings of the shape optimization problem of the partly clamped hyperbolic paraboloid with stiffeners. Seven design variables x_i are associated to the embedded surfaces in order to impose shape changes to the stiffeners.

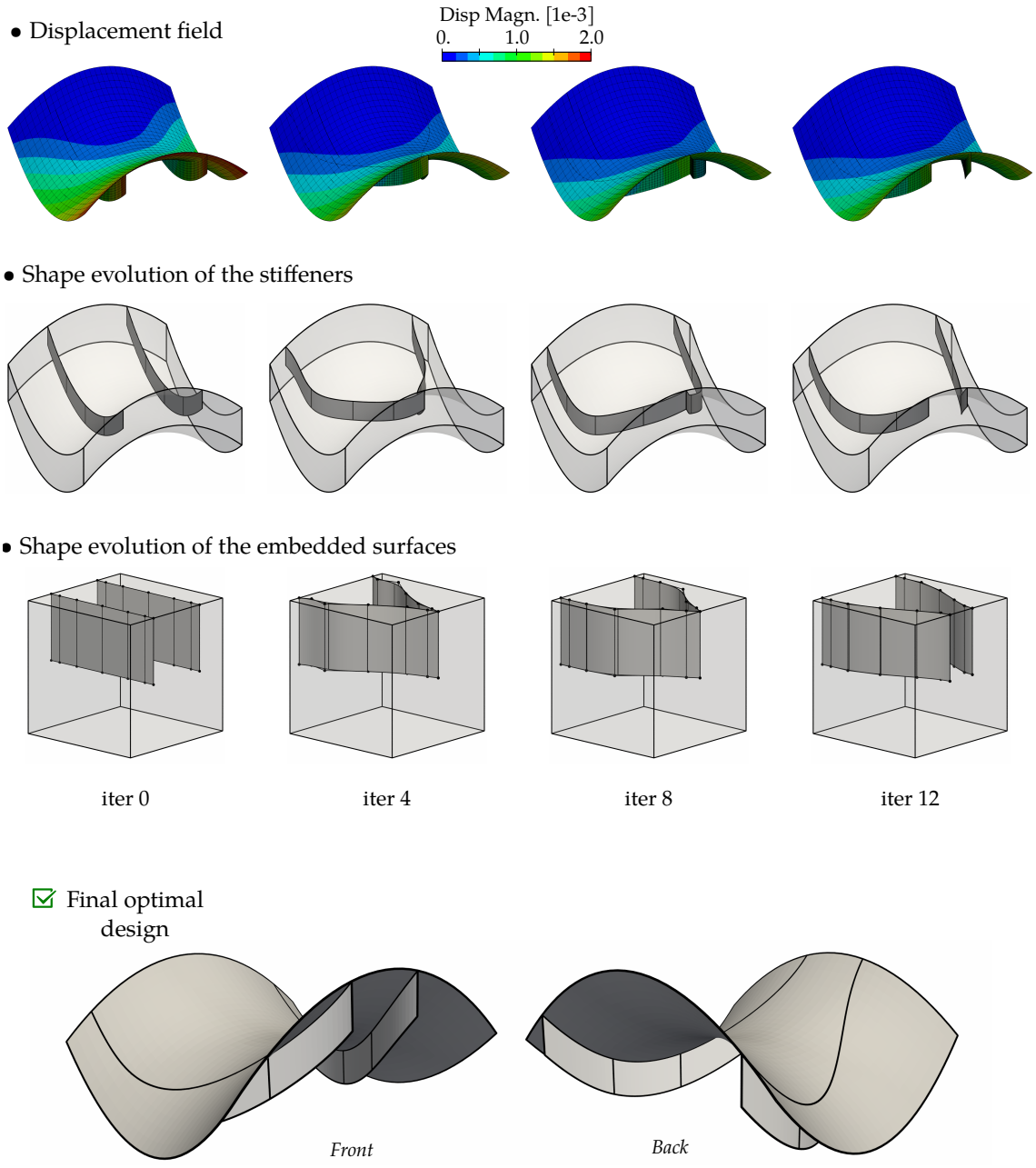


Figure 5.16 – Optimization results for the hyperbolic paraboloid with stiffeners.

shape updates to the stiffeners. Thus, seven design variables are introduced as given in figure 5.15. Bound constraints are set on the design variables to avoid the intersection of the stiffeners and to maintain the embedded surfaces into the parameter space of the volume. Finally, the design space is constrained such that the area (or indirectly the mass) of the stiffeners cannot exceed 105% of the initial one which is equal to $A_0 = 0.25\,\mathrm{m}^2$.

The analysis model is defined thanks to the refinement procedures. The panel is discretized with 32-by-32 quartic standard Kirchhoff–Love shell elements. Each stiffener counts 64-by-8 quartic embedded Kirchhoff–Love shell elements. The Lagrange multipliers are discretized accordingly to the strategy presented in section 4.2.2. For the displacement coupling, the Lagrange multiplier is discretized with 32 cubic elements. For the rotation coupling, the Lagrange multiplier is discretized with 32 quadratic elements.

Optimization results The optimization results are given in figure 5.16. The optimization algorithm converges quickly and the final design is achieved after less than 20 iterations. We depict in figure 5.16 several shape updates obtained during the resolution. We present the evolution of the displacement field, and the evolution of the shape of the stiffeners. This example shows the great benefit of the developed embedded strategy. Indeed, the stiffeners automatically follow the curvature of the panel. One should notice that we do not apply any specific treatment to impose the bounding (geometric) condition at the interface between the panel and the stiffeners. Due to the construction, when moving the control points of the embedded surfaces, the stiffeners smoothly slide along the panel. No geometric approximation is involved in the entirely process. Looking at the shape evolution of the embedded surfaces (see again figure 5.16), one can see that the shape updates are relatively simple. However, despite its apparent simplicity, it integrates the complex task of taking into account the curvature of the panel.

Finally, let us give some quantitative results regarding the optimal design. The compliance is decreased by more than 30%. More specifically, the initial compliance in the case of straight stiffeners is equal to $c_0 = 38.53$. The final compliance for the optimal shape is equal to $c_{\mathrm{opt}} = 26.56$, which gives the ratio $c_{\mathrm{opt}}/c_0 = 0.6894$. It highlights that a true benefit can be brought by designing curvilinear stiffeners instead of simple straight stiffeners. The final design does not differ that much from the initial one but it behaves much better (in terms of stiffness).

Because the number of design variables is low, we can express their final values:

$$x_{\mathrm{opt}} = \begin{pmatrix} 0.4 & 0.4 & 0.4 & 0.328 & 0.204 & 0.128 & 0.124 \end{pmatrix}.$$

Several views of this final design are depicted in figure 5.16. This example may constitute a good test case if one tries to apply our methodology, or if one want to confront their results provided by an alternative approach with ours. In fact, to the authors'

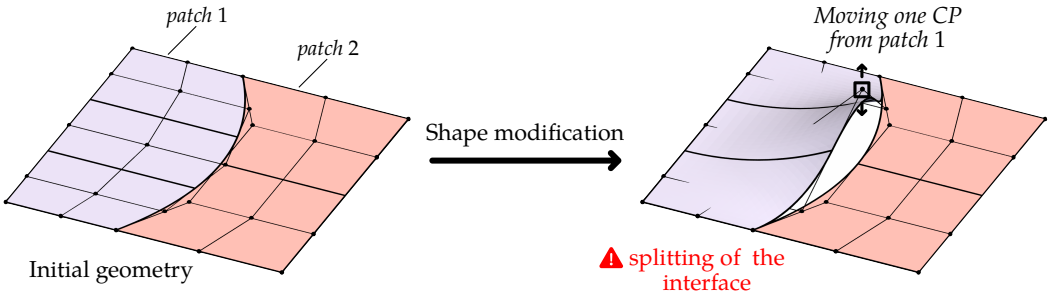


Figure 5.17 – The shape update in case of non-conforming multipatch models should be aware of preserving the geometric continuity at the patch interfaces.

knowledge, there do not exist benchmark examples for the shape optimization of curvilinear stiffened structures.

5.3.3 Optimizing the global shape of a multipatch structures

The presented embedded strategy also enables to optimize the global shape of a multipatch structure. Once again, the benefit lies in the fact that the connecting constraints at the patch interfaces are directly taken into account without any specific treatment. For that purpose, we apply the global optimization approach based on what as been presented in figure 5.5. We will impose the shape updates through the volume mapping.

To highlight the capability of our approach, we will reuse the optimization problem of the square roof already tackled with a monopatch model (see figure 3.19 and its associated discussion). The difference is that the initial square plate is modeled here with a non-conforming multipatch model. The goal of this study is to underline that the embedded Kirchhoff–Love shell brings general insights for the shape optimization of non-conforming multipatch structures and should not be restricted to stiffened structures. We reserve the case of stiffened structures to chapter 6 which is dedicated to these specific type of application.

Geometric constraints Let us consider having a given model in hand that contains, for instance, several non-conforming patches. As an example, take the square plate made of two non-matching patches as described in figure 5.17. The question is how shape optimization can be performed with this multipatch model ? It is straightforward that the control points of each patch cannot freely move at the interface (as done in figure 5.17), but on the contrary, some constraints need to be imposed such that a certain level of continuity is preserved at the interface during the shape updates.

This question is not new and solutions have already been studied. In particular, Schmidt [205] deals with the enforcement of geometric continuity (G^1-continuity) in the case of a design model that has non-matching patches. They introduce variational

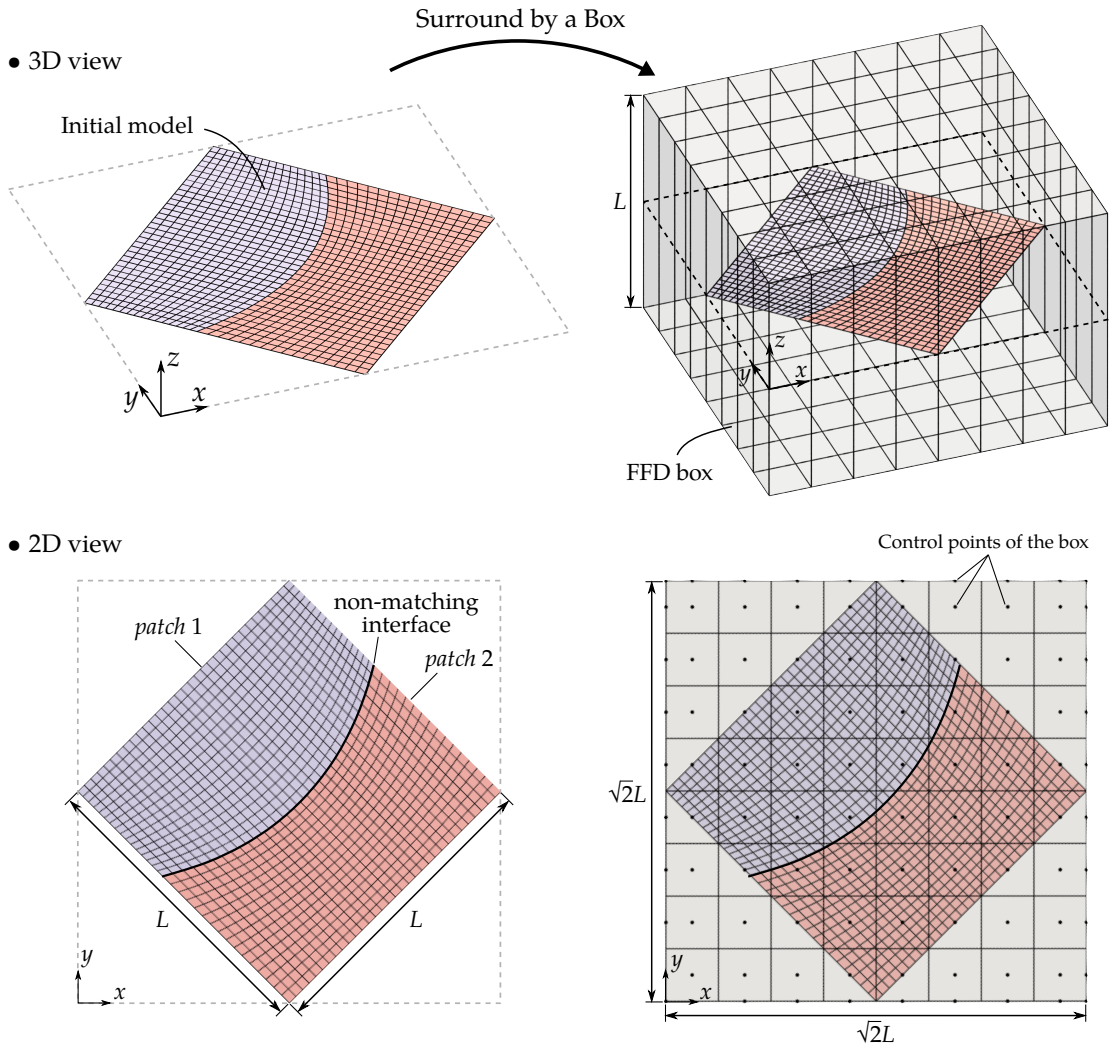


Figure 5.18 – Setting up a FFD shape parametrization in order to impose shape modifications that preserve the geometric continuity between the non-conforming patches.

equality constraints in the optimization problem which enforce the geometric continuity in a weak sense.

Here, we show how the embedded Kirchhoff–Love shell constitutes an alternative approach that has the advantage of explicitly taking into account the geometric constraints of preserving a certain level of continuity at the interface between non-conforming patches of an optimization model. The approach is inspired by the Free-Form Deformation technique widely used in aerodynamic shape optimization [50, 206–208]. The idea consists in diving the initial model into a morphing box. This FFD box enables to deform the shape of the immersed object, for example airfoils and wings.

FFD of non-conforming models To illustrate the approach, let us take the case of the non-conforming multipatch model of the square plate as shown in figure 5.18. We can define a box that surrounds the plate. The box is built using a B-Spline volume that represents the identity transformation. Its parameter space and its physical space are identical. Now the initial surfaces defining the geometry of the plate can be seen as embedded surfaces of the volume. The geometry of the shell is henceforth defined through NURBS compositions. This is a major difference with the traditional FFD approach since the box does not act on the node (or control point) coordinates of the immersed entity but the deformed shapes are obtained through this composition of the initial model and the volume mapping (NURBS composition in our case):

$$\boldsymbol{S}(\theta_1, \theta_2) = \boldsymbol{V}\left(\boldsymbol{S}^{\text{init}}(\theta_1, \theta_2)\right) \tag{5.56}$$

We can impose shape modifications to the plate by modifying the volume mapping. Taking the volume as a quadratic B-Spline ensures that the geometric continuity at the non-matching patches is preserved. This requirement is therefore automatically fulfilled. Thanks to the embedded Kirchhoff–Love shell formulation, we are finally able to perform the structural analysis and the sensitivity analysis directly on this model based on NURBS compositions.

Settings In order to illustrate the capability of the presented approach, we recall the square roof problem tackled with a monopatch model in chapter 3 (see figure 3.19). It consists in a square roof which is fixed at its corners and subjected to a constant vertical load. We consider here the model of the plate with two non-matching patches and the FFD box as already described in figure 5.18. Of course, the definition of the volume is not unique and other choices could have been done. The presented configuration has the advantage that the four corners of the plate are located on the faces of the box. If one does not want to modify the location of these corners, one can simply keep these faces untouched. We make this choice here. We decide to associate the design variables only to the control points from the upper face of the box (besides those in common with the other faces).

More specifically, the box is built starting from a linear patch with one single element. The knot vectors are defined as follows:

$$\Xi = \mathcal{H} = \{0,\ 0,\ \sqrt{2}L,\ \sqrt{2}L\}, \qquad \mathcal{Z} = \{-L/2, -L/2,\ L/2,\ L/2\}.$$

The degree is then elevated by one in the first two directions (corresponding to the knot vectors Ξ and $\mathcal{H}$, and to the directions x and y in figure 5.18). Uniform knot insertions are also performed such that the refined box counts 8-by-8-by-1 elements. It leads to 64 design variables. They move the control points in the z direction only.

The two embedded surfaces (patch 1 and 2 in figure 5.18) are discretized with 33-by-16 and 32-by-16 cubic elements. Their common interface describe a parabola of height $h = 2\,\text{m}$ and of length $L = 10\,\text{m}$. The mechanical coupling is applied using the Mortar approach as presented in section 4.2.

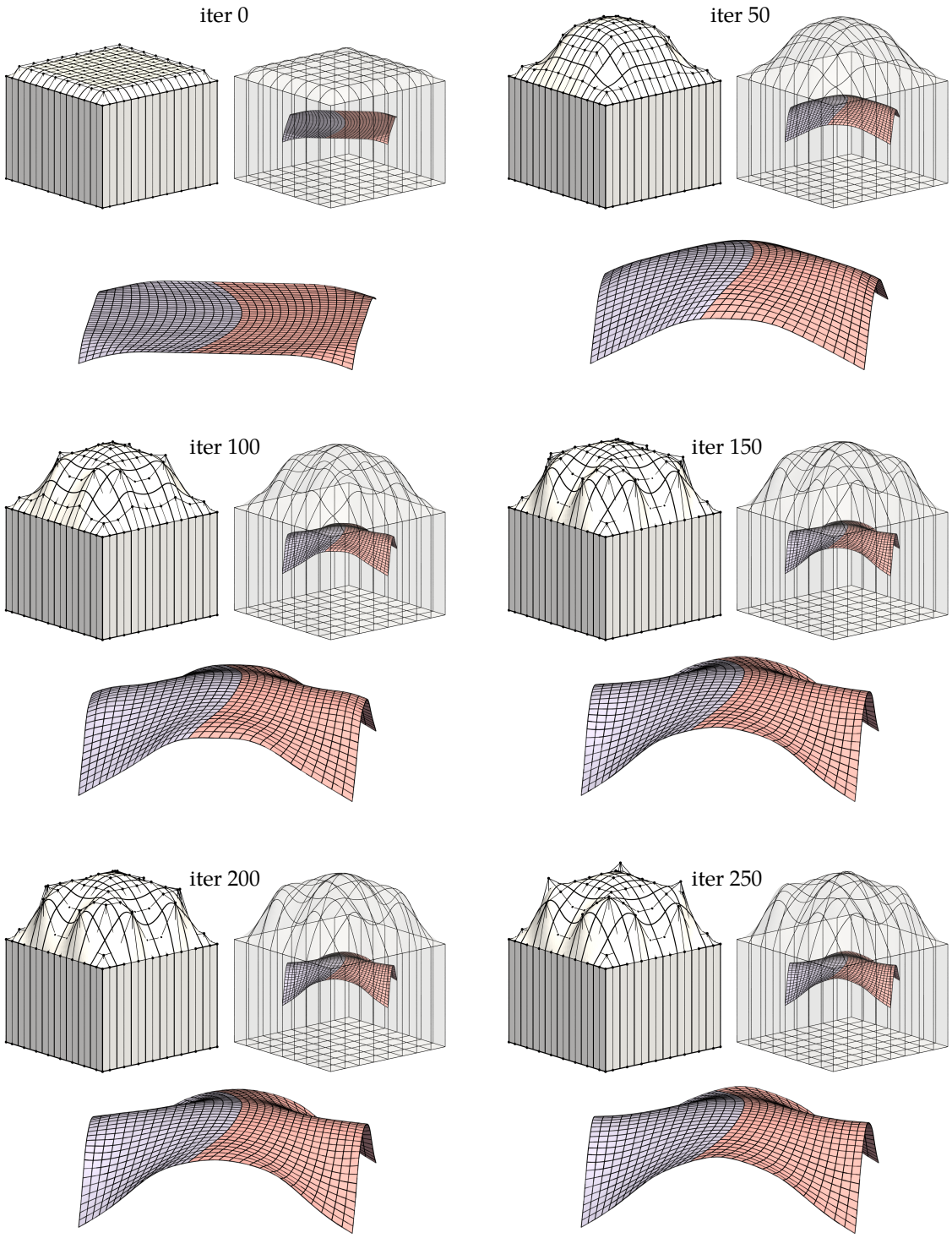
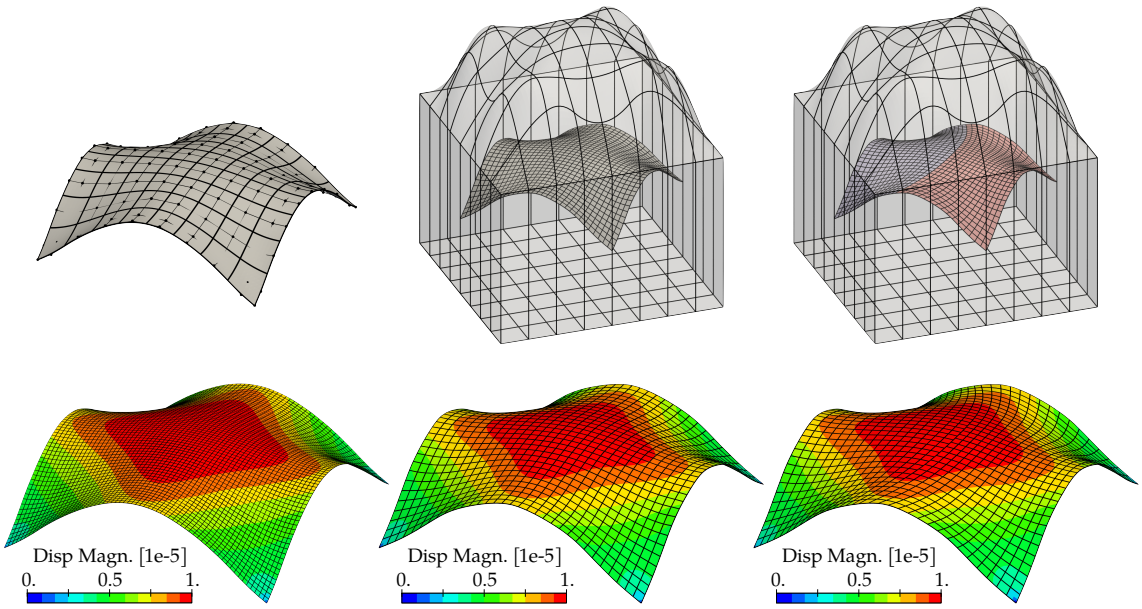


Figure 5.19 – Optimization history of the Free-From Deformation approach applied to the shape optimization problem of the square roof with a non-conforming multipatch model.



Standard Approach FFD monopatch FFD multipatch

Figure 5.20 – Comparison of the results obtained for the shape optimization of the square roof. The FFD approach applied to the multipatch model gives very similar results than both monopatch resolutions (standard and FFD).

Optimization Results We give in figure 5.19 several shapes obtained during the resolution of the square roof problem. The starting point is not the square plate because it is a saddle point of the problem, but a slight curvature is set (design variables equal to one instead of zero). One can see how the volume is modified, and what are the consequences for the shape of the shell. Looking at the optimal design, we recover a similar shape than the one obtained with a monopatch model and the standard Kirchhoff–Love shell. To highlight this point, we present several results in figure 5.20. We present the result obtained with the standard IGA-based approach where the optimization model counts 96 design variables. We consider this optimal geometry as the reference solution here. The differences between this reference optimal design (left in figure 5.20) and the non-matching multipatch result (right in figure 5.20) are almost undetectable. Looking at the displacement fields of both these configurations also confirms the correctness of the multipatch resolution.

In order to study the influence of having a multipatch model, we also solve the optimization problem with the FFD approach applied to a monopatch model. Instead of considering the multipatch model in figure 5.18, we take a simple square surface with 32-by-32 cubic elements. The final result is presented in figure 5.20 (middle). Again, a similar optimal shape is achieved. More specifically, we can quantify the differences between both FFD results by looking at the final values of the design variables. Let us

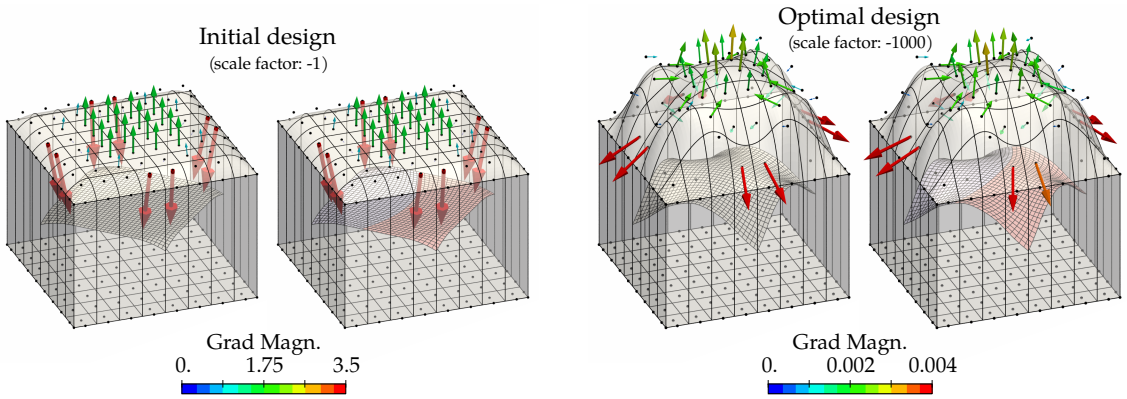


Figure 5.21 – Comparison of the sensitivity analysis results of the FFD approach when applied to the monopatch or the multipatch model.

denote by x^{mono} and x^{multi} the final design variables for the FFD monopatch and FFD multipatch resolution, respectively. We can evaluate the gap between these two results as follows:

$$\mathcal{E}^{\text{multi}} = \frac{||x^{\text{mono}} - x^{\text{multi}}||}{||x^{\text{mono}} + x^{\text{multi}}||} = 0.137\%.$$

It proves that it leads to the same result. A slight asymmetry exists for the multipatch case. It originates from the initial discretization of the square plate that is not symmetric due to the curved interface. Lastly, we give in figure 5.21, the results of the sensitivity analysis for the initial and the final designs for both monopatch and multipatch cases. It enables to show that the influence of having a non-matching multipatch model is very low. It leads to the same sensitivities as with the monopatch model. Thus, similar shape updates and final optimal shapes are expected. We observe this behavior in this example. Finally, let us briefly notice that, in the sensitivity analysis of the multipatch case, there is no need to compute the terms coming from the Mortar coupling. It vanishes here since the interface is moving identically from the perspective of each patch.

To conclude, this example demonstrates the capability of the presented FFD approach to handle complex multipatch for shape optimization. The use of NURBS compositions and of the dedicated embedded shell formulation has a high potential for this purpose. The approach seems applicable for the shape optimization of more complex models that contains, for example trimming patches. The difficulty of preserving the geometric continuity is automatically removed, even for these types of configurations.

Chapter 6

A robust framework for designing innovative Aerostructures

New methodologies are considered as attractive only if they are able to solve problems of real interest. The goal of this last chapter is to highlight the potential of the presented strategies for designing innovative aerostructures. To this purpose, we firstly show how all the developed work can be concatenated into one single framework. Several examples of shape optimization of stiffened structures are then presented. It brings to light the capabilities of our framework to design futuristic stiffened structures.

Contents

6.1 Unify Domain Decomposition and Multipatch Optimization

Structural optimization is computationally expressive because it requires a large successive number of structural analysis (and sensitivity analysis). If the goal is to design complex *real-world* structures, then it is crucial that each step of the resolution is adequately developed. Since chapter 4, we know of to handle non-conforming discretizations and how to analyze such multipatch models in an efficient manner thanks to the dual Domain Decomposition solver. In chapter 5, we developed an embedded strategy which allows to perform shape optimization on non-conforming multipatch models in a smooth manner. However, we did not mention yet how both approach can be merged. In this section we discuss this issue.

6.1.1 Shape update in Domain Decomposition

The domain decomposition adopted in this work consists in taking each patch as a sub-domain. During the resolution with the dual Domain Decomposition, we mention that a sub-domain can be seen as a black-box (see discussion on parallel computing on page 122). It takes the interface effort as an input, and it returns the displacement (and the rotation for shells) at the interface. We wonder if we can adopt the same vision to perform shape optimization on multipatch models: viewing each patch as a sub-domain (black-box), what would be the inputs and the outputs ?

Simple example Let us take a simple example of a panel with one stiffener depicted in figure 6.1. The structure is built using the construction method presented in section 5.1.2. Thus, the stiffener is defined through the composition of a volume mapping and an embedded surface. The design variables can be associated to the surface representing the panel, to the volume mapping, or to the embedded surface. The key point is that the shape parametrization cannot be independent for each patch. The shape update of the panel is transferred to the stiffener through the shape modification of the volume. Conversely, the shape modification of the stiffener may lead to motion of the interface over the panel. Thus, the panel should be aware of this motion. It means that information about the shape modifications have to be communicated between the sub-domains.

Figure 6.1 illustrates this dependency of the design variables related to each sub-domain. Let $\mathcal{X}^{\text{panel}}$ and $\mathcal{X}^{\text{stiffener}}$ denote two sets of design variables acting on the panel and on the stiffener, respectively. For the panel, the design variables can either be applied to the control points of the surface or on the control points of the interface curve. Mathematically, we can decompose the set $\mathcal{X}^{\text{panel}}$ as follows:

$$\mathcal{X}^{\text{panel}} = \mathcal{X}^{\text{surface}} \cup \mathcal{X}^{\text{curve}}.$$

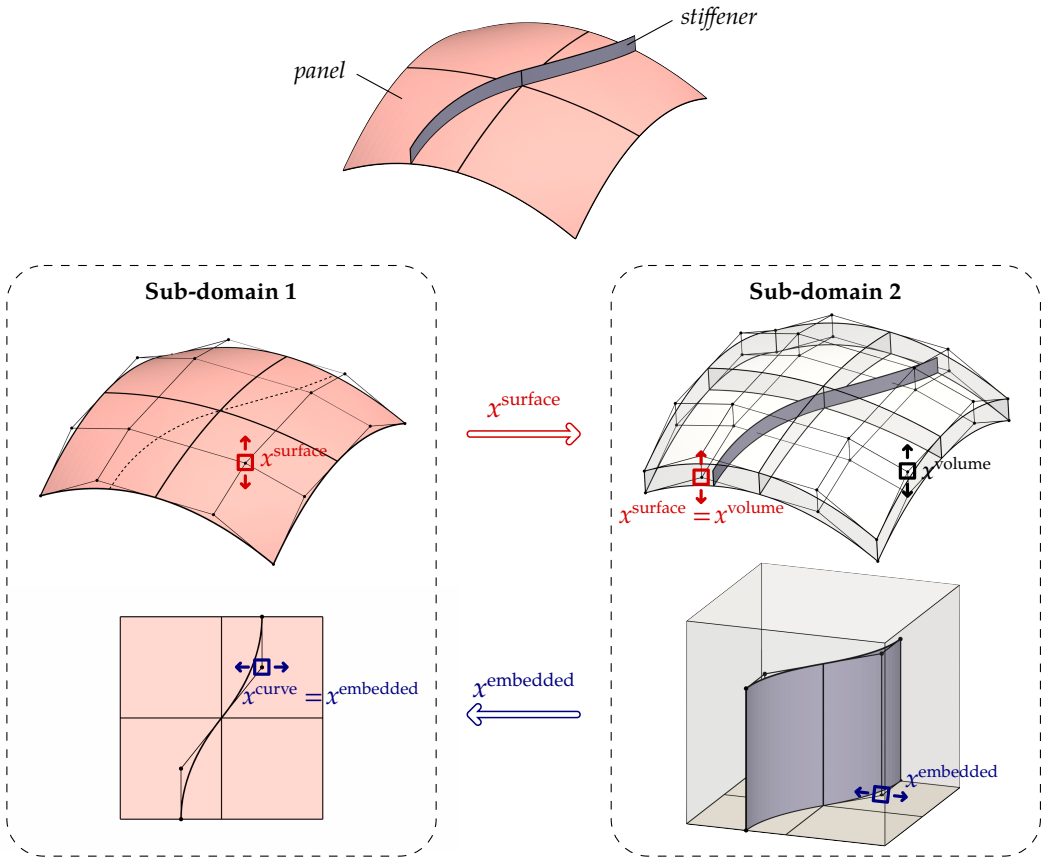


Figure 6.1 – Dependency between the sub-domains regarding the shape parametrization.

For the stiffener, the design variables can either act on the volume mapping or on the embedded surface. Thus, we can view $\mathcal{X}^{\text{stiffener}}$ as:

$$\mathcal{X}^{\text{stiffener}} = \mathcal{X}^{\text{volume}} \cup \mathcal{X}^{\text{embedded}}.$$

The intersection between these two sets, $\mathcal{X}^{\text{panel}}$ and $\mathcal{X}^{\text{stiffener}}$, is not null. As shown in figure 6.1 and as already mentioned, every design variables from the panel side should be transferred to the stiffener in order to update the volume. In the other side, some of the design variables acting on the embedded surface should be transferred to the panel in order to update the interface curve. In other words, the set $\mathcal{X}^{\text{comm}}$ of design variables that needs to be communicated between the two sub-domains is given by:

$$\mathcal{X}^{\text{comm}} = \mathcal{X}^{\text{panel}} \cap \mathcal{X}^{\text{stiffener}} = \mathcal{X}^{\text{surface}} \cup \mathcal{X}^{\text{embedded}}.$$

Finally, the panel and the stiffener share quite a lot of design variables. Only the stiffener may have independent design variables (associated to the volume excepted the face matching with the stiffener, and to the embedded surface excepted the edge representing the interface).

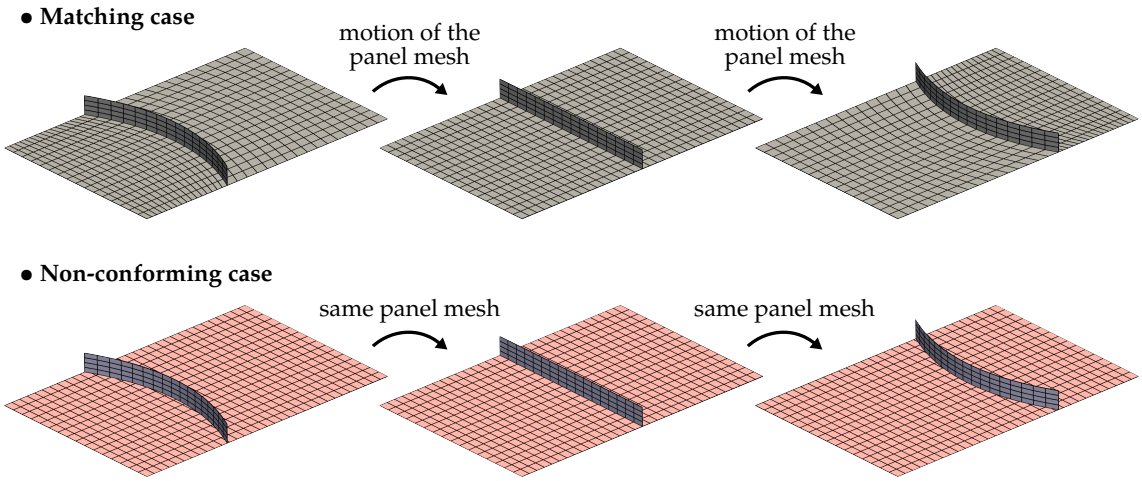


Figure 6.2 – The discretization of the panel need to be update at each shape modification of the stiffener when a matching multipatch model is used. Using model with non-conforming discretization enables to keep the mesh of the panel untouched.

Generalization Based on what has been observed on the simple example, we can briefly generalize to models with multiple panels/stiffeners. When a stiffener and a panel are coupled,

— the panel sends the shape update of its surface to the stiffener,
— the stiffener receives this information in order to update the volume mapping,
— the stiffener sends the shape update of the embedded surface to the panel,
— the panel receives this information in order to update the interface curve.

Hence, we understand, as of now, that the communication between the sub-domain does not only involve quantities coming from the structural analysis. Implementing the resolution in parallel is more challenging than that.

6.1.2 Parallel computing and Numerical efficiency

We have not implement the method in parallel within the scope of this work. We discuss this issue here in order to motivate researches in this direction.

Non-design parts A first interesting point to notice concerns the structural analysis. If the mesh of a patch is not modified during the shape optimization, then several savings in term of computational time can be made. In fact, the sub-domain stiffness matrix $\mathbf{K}^{(s)}$ will not change during the whole process. More importantly, its pseudo-inverse $\mathbf{K}^{\dagger(s)}$ needs to be built once and for all at the beginning of the resolution. This can represent a great numerical saving. This is true when only the shape of the stiffeners is optimized and the panel is kept fixed. The pseudo-inverses associated to the panel are built only once, and it can be reused during the whole resolution. It is also interesting to point out

that the panel remains unchanged because of the use of non-conforming meshes. As shown in figure 6.2, in the case of a model with matching meshes between the stiffener and the panel, the mesh of panel is modified during the shape update of the stiffener even if the overall shape remains the same (*i.e.* a rectangle). This is an interesting feature provided by the use of non-conforming discretizations.

Regarding the PCPG algorithm 4.1 involved in the dual Domain Decomposition solver, the choice of the starting point $\boldsymbol{\lambda}^0$ may be improved by taking the solution of the previous design. The algorithm may converge faster with such a choice, notably when the design changes between two successive iterations are not important. The solution of the FETI problem should not differ a lot when the design changes are low. Thus, the use of non-conforming meshes and the dual Domain Decomposition solver can significantly accelerate the structural analysis step.

Sensitivity analysis / Adjoint problem As for the structural analysis, the sensitivity analysis may also be performed in parallel. Let us remind the generic expression of the discrete sensitivity as given by equation (3.14):

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial \boldsymbol{u}} \cdot \mathbf{K}^{-1} \left(\frac{\partial \boldsymbol{F}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i} \boldsymbol{u} \right).$$

Either for the direct method (3.16) or the adjoint method (3.15), the calculation of the sensitivities often requires the resolution of a system (or even multiple ones). But thanks to the dual Domain Decomposition solver, this resolution step in the sensitivity analysis can be done in parallel. Furthermore, since the pseudo-inverses $\mathbf{K}^{+(s)}$ have already been built during the structural analysis, the numerical cost for solving the linear systems could be largely reduced. The same remark about the improvement of the starting point for the PCPG algorithm is also true here.

Regarding the communications between the sub-domain (in other words between the processors), we have seen that they share common design variables. For instance, the derivative of the respond function *w.r.t.* a design variable x^{srf} associated to the surface of the panel has two terms: one coming from the panel and on coming from the stiffener:

$$\frac{df}{dx^{\text{srf}}} = \left(\frac{df}{dx^{\text{srf}}} \right)^{\text{panel}} + \left(\frac{df}{dx^{\text{srf}}} \right)^{\text{stiffener}} \tag{6.1}$$

Therefore, it raises the question how to store the components of the gradient. It may no be necessary that each sub-domain knows the whole gradient. In the example given by equation (6.1), a solution could be that the stiffener sends its contribution to the panel only. There is *a priori* no need that the panel sends its contribution to the stiffener. Thus, only the panel knows the complete derivative. A closer look at this point should be of great interest.

Optimization algorithm Lastly, there is the question of the optimization algorithm. If the design model is really large with a lot of design variables, then the optimization

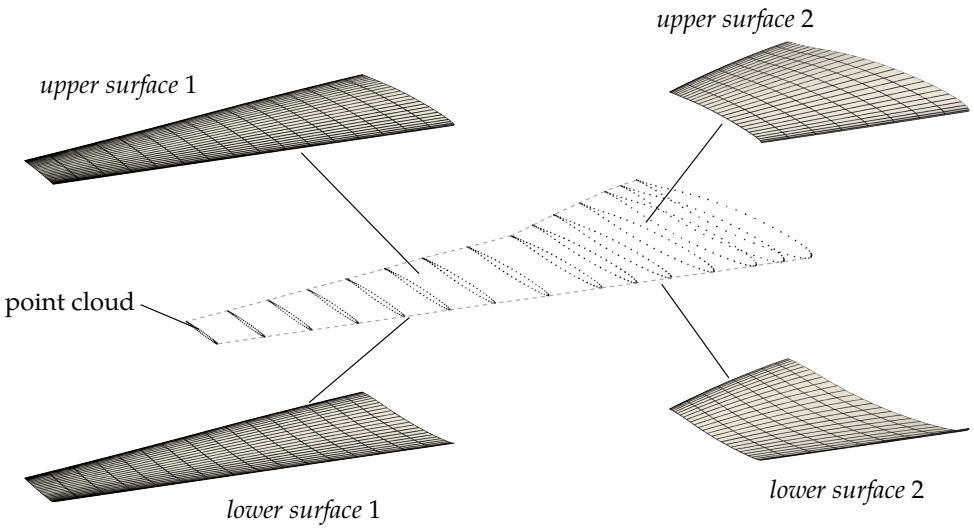


Figure 6.3 – Defining the outer skins of the wing from a point cloud. Four surfaces are interpolated.

algorithm should be adequately chosen. It seems, in this case, not efficient to associate one single processor to handle the optimization algorithm. Indeed, the number of communication with the sub-domain would be high. For instance, it would, in this case, receive the contribution of each sub-domain for the gradient etc. Topology optimization requires all a lot of computational resources and works have been done in order to deal with very large model with millions of design variables [38, 209–213]. An impressive result can be found in [38] where the design of the internal structure of a full-scale aeroplane wing is presented. Other interesting ideas could surely be found by looking at the methods developed in multidisciplinary optimization [142, 143, 214]. Instead of solving one large optimization problem, multiple local optimization problems could be solved at the sub-domain level. Thus, interesting ideas taken from these cited works could surely be applied in our framework in order to get a full parallel implementation of the *code*.

6.2 Innovative Design of Stiffened Structures

We present in this section several optimization of stiffened structures that have been done with the methods developed in this work. It highlights that our approach is quite general and can be applied to a large variety of problems.

No software has been used (except paraview for the post-processing). The core part of the *code* is written in `Fortran` programming language (stiffness matrices, coupling operators, sensitivities, post-processing outputs, etc). These main routines are latter wrapped using `f2py`. It enables to interface the `Fortran` routines into `Python` scripts. `Python` programming language has the advantage of supporting object-oriented pro-

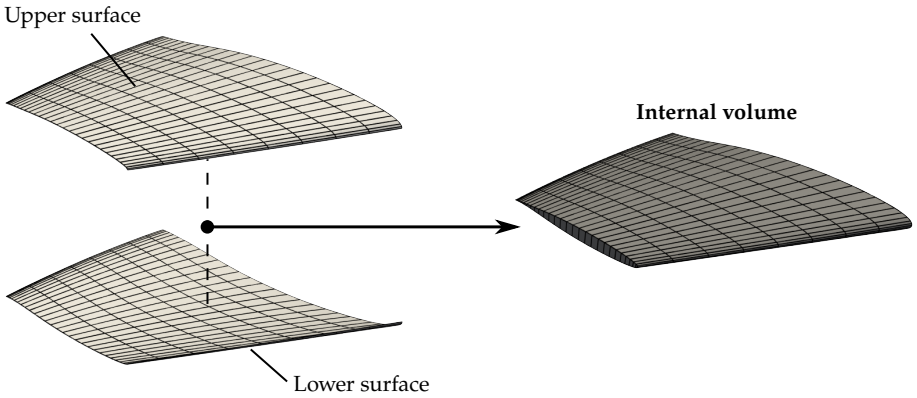


Figure 6.4 – The upper surface and the lower surface are used to generate a volume that fills the domain delimited by the skin of the wing.

gramming. Several `classes` have been created in order to get a convenient way to set up an optimization problem.

6.2.1 Versatile construction of an Aircraft Wing

We have presented and built our embedded approach in order to generated stiffened structures (see the process in figure 5.4). Since now, we have mainly showed how it enables to add stiffeners to curved panel (see for example the stiffened paraboloid from figure 5.15). Nevertheless, the method is capable the generate more complex geometries. In order to prove this point, let us try to build a model an aircraft wing with its internal sub-structure.

Outer surfaces The starting point is the representation of the external skins that define the outer geometry of the wing. This can be done either by using a CAD software as Rhinoceros 3D [215] or by directly interpolating a point cloud. The NASA common research model [216, 217] is used in this work. The geometric data for the CRM wing [216] enables to generate a point cloud that represents the outer geometry of the wing. Then, four B-Spline surfaces are generated by interpolation/least-square approximation of these points. Information about interpolation techniques with Splines can be found in Piegl et al. [89]. We end up with the surfaces depicted in figure 6.3.

Internal volume The next step consists in the construction of volumes that fill the domain delimited by the skins. To generate these volumes, we simply rely on the surfaces. We define two B-Spline volumes. These volumes are defined such that two of their faces match with the upper and the lower skins. Linear degree is chosen in the additional direction of the volume (from lower surface to the upper one). The knot vector is simply taken as $\mathcal{Z} = \{0, 0, 1, 1\}$. Thus, no additional control points are generated during the definition of the volumes. An example of a volume is given in figure 6.4. The two

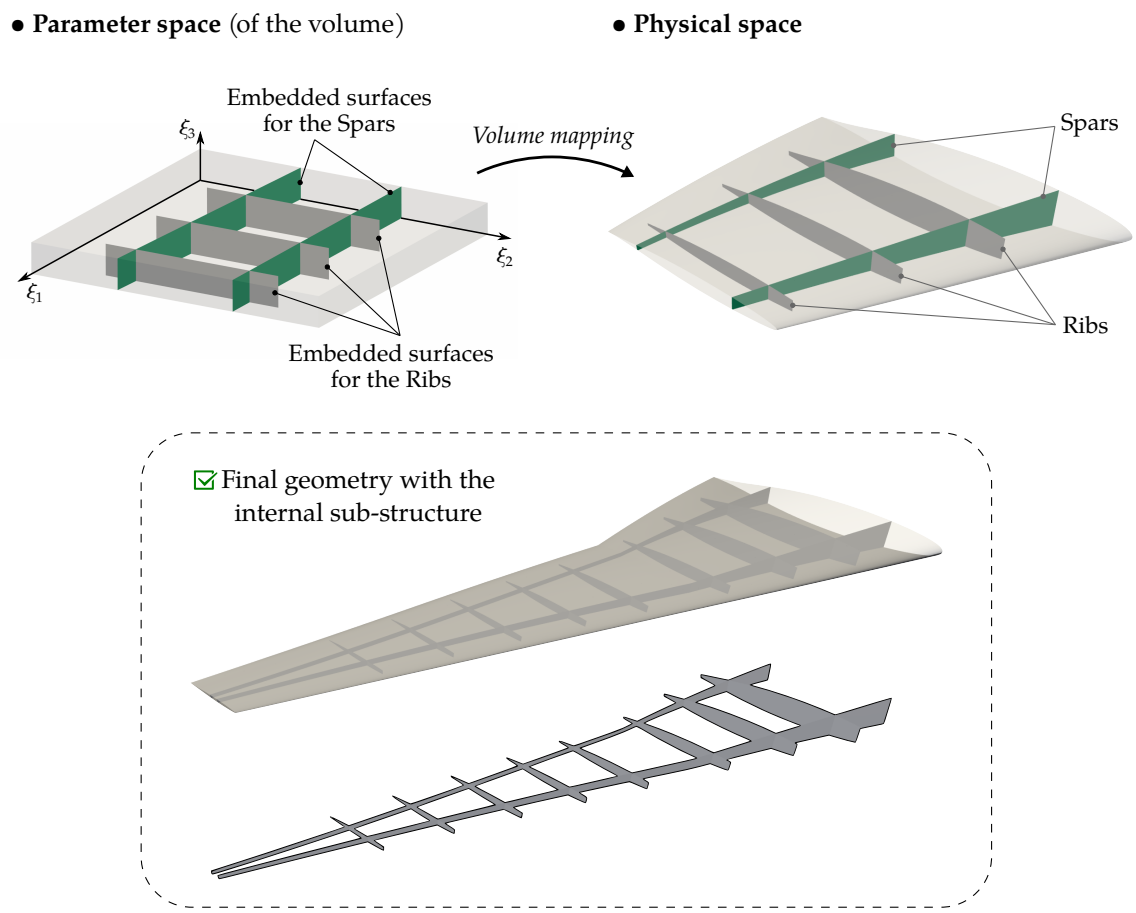
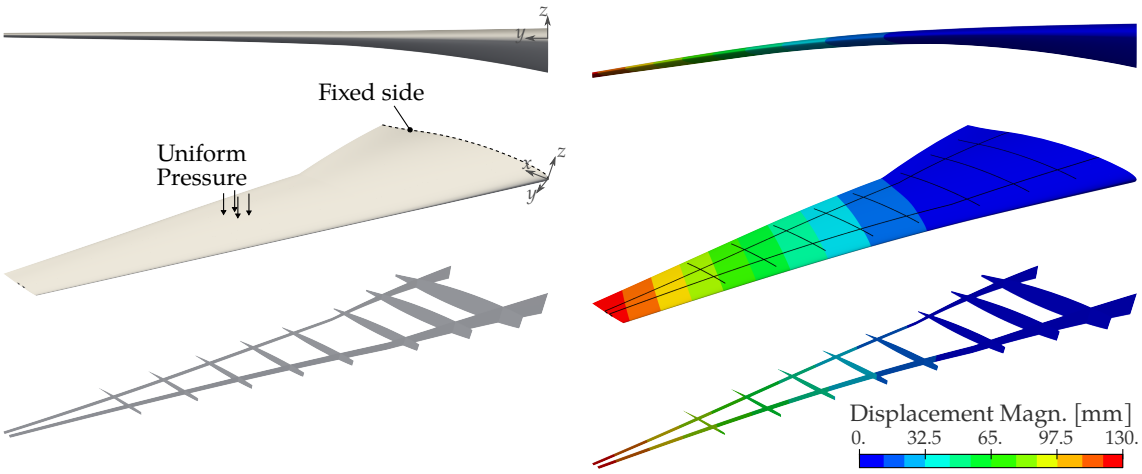


Figure 6.5 – Geometric modeling of the internal substructure: the spars and the ribs are defined by embedding NURBS surfaces into the parameter space of the previously generated internal volumes (see figure 6.4).

generated volumes do not represent a structural part of the wing. They are built only for geometrical purposes.

Internal sub-structure In fact, once the internal volume is defined, we apply the embedded strategy in order to describe the ribs and the spars that compose the internal sub-structure of the wing (the wing box). We embed surfaces into the parameter spaces of the volumes. Then, the compositions of these embedded surfaces and the internal volumes give the final surfaces of the internal sub-structure. An example of the modeling process is shown in figure 6.5. This construction guarantees that the sub-parts are perfectly inside the domain delimited by the outer surfaces. The ribs and the sparse can easily be defined such that they perfectly lie on the skins.

Structural Analysis Interestingly, the model is directly usable for structural analysis by using the tools developed in this work. Firstly, the geometric modeling of the spars



Initial Configuration Deformed Configuration (factor 10)

Figure 6.6 – Structural analysis of the CRM wing.

	Wing			
ref	N_{tot}	N_λ	**I**	$\tilde{\mathbf{F}}_D^{-1}$
0	9246	2272	1874	258
1	28 554	4096	-	299
2	102 444	7744	-	378

Table 6.1 – Performance study for the wing problem.

and the ribs is done through NURBS composition. However, thanks to the embedded Kirchhoff–Love shell formulation, we are able to analyze these parts directly. Secondly, the Mortar coupling enables to deal with the non-conforming junctions between the different parts. The model contains 16 patches: 4 quadratic standard Kirchhoff–Love shells for the skin, 8 and 4 cubic embedded Kirchhoff–Love shells for the ribs and the spars, respectively. Finally, the dual Domain Decomposition solver gives an efficient way to perform the structural analysis. Each patch is taken as a sub-domain for our solver.

Figure 6.6 presents an example of results that have been obtained on the wing model. The mechanical properties are the following: every part (the skin, the ribs, and the spars) has a Young's Modulus $E = 210\,\text{GPa}$, and a Poisson's ratio $\nu = 0.30$. The thickness of the skin is taken as $t^{\text{skin}} = 30\,\text{mm}$, the thickness of the ribs is taken as $t^{\text{ribs}} = 40\,\text{mm}$, and the thickness of the spars is taken as $t^{\text{spars}} = 50\,\text{mm}$. The root side of the wing is fixed and a uniform pressure $P = 10\,\text{kPa}$ is applied on the upper skins (made of two surfaces, see figure 6.3). For the other geometric settings, we refer once again to Vassberg et al. [216]. A scale factor is applied such that wing has the following dimensions: the span is 15 m, the sweep angle is 35°, the root and tip chords are equal to 7 m and 1 m respectively.

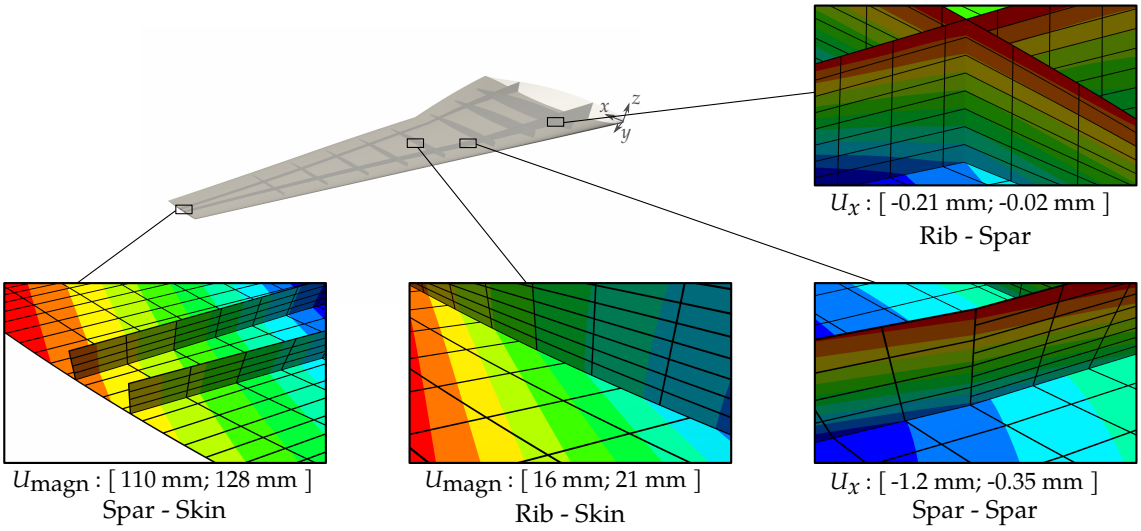


Figure 6.7 – Structural analysis of the CRM wing.

In order to highlight the correct imposition of the coupling conditions, we depict detail views of the different non-conforming interfaces in figure 6.7. One can see the continuity of the displacement over these interfaces. Concerning the convergence of the dual Domain Decomposition algorithm, we were only able to run the resolution without the preconditioning step for the coarsest discretization. In this case, it already takes almost 2000 iterations to converge (see table 6.1). For the highest level of refinement, it does not converge before the stopping criteria of maximal iteration number set to 2000. However, the generalized Dirichlet preconditioner drastically accelerates the convergence as shown in table 6.1. It enables to analyze refined versions of the wing model in an acceptable number of iterations. We get a slow growth of the iteration count versus the refinement level. It demonstrates that the generalized Dirichlet preconditioner introduced in section 4.3.4 is robust even for more complex cases as the aircraft wing.

6.2.2 Global optimization: the stiffened roof

The stiffened roof problem highlights the potential of our approach to optimize the global shape of stiffened structures. Here we apply the Free-Form Deformation shape parametrization as introduced in chapter 5 (see more specifically figure 5.5, and section 5.3.3). The shape modification is done through volume that contains embedded surfaces.

Settings The problem of the stiffened roof is derived from the initial shape optimization problem of a roof under vertical uniform load [144, 145]. We also dealt with the square roof problem multiple times in this work: see the standard monopatch resolution in figure 3.19, and the multipatch resolution in section 5.3.3.

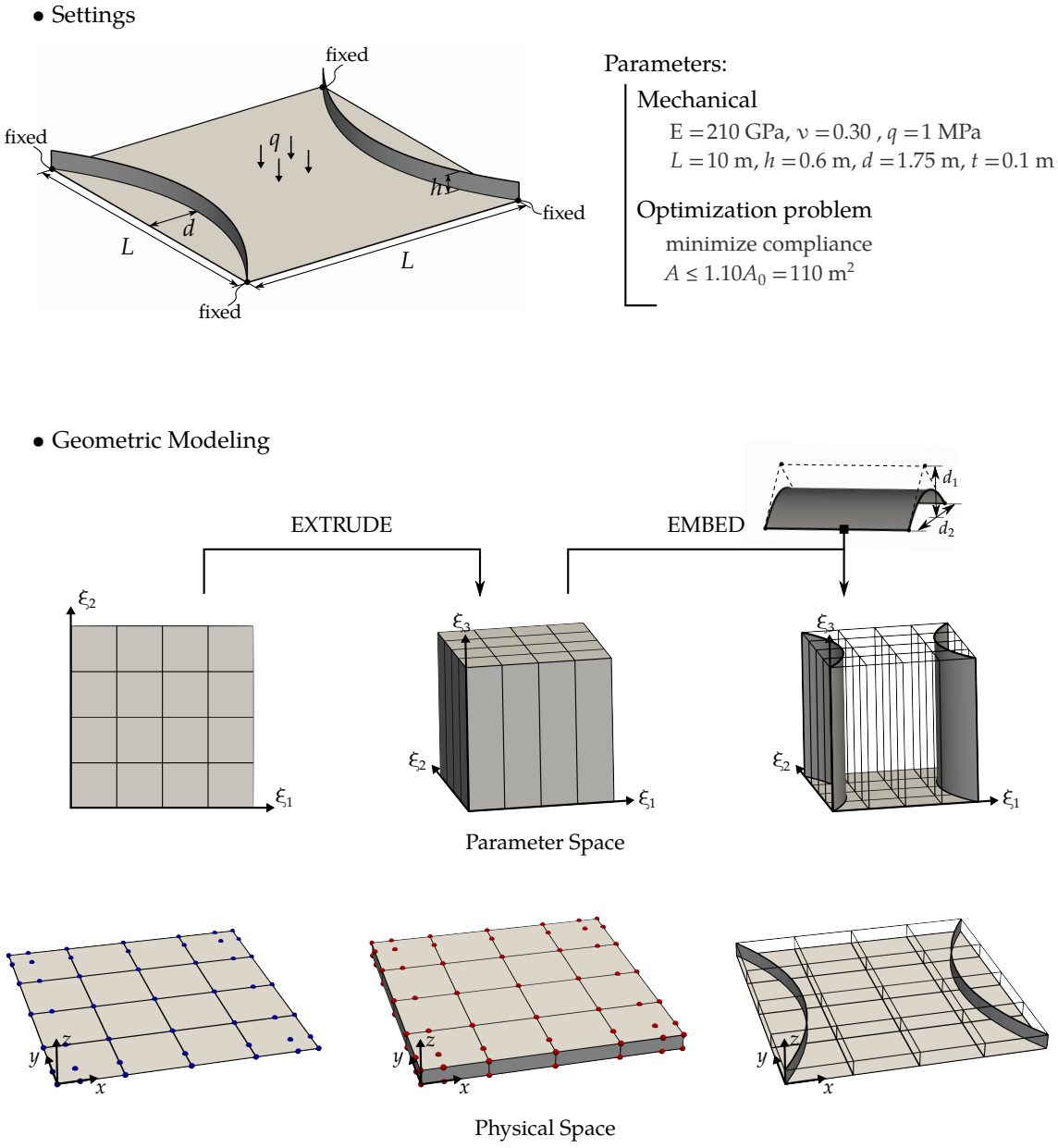


Figure 6.8 – Settings of the stiffened roof problem. The geometric modeling employs the embedded approach: a volume is extruded from a square plate, and then, the stiffener are generated by embedding surfaces into the parameter space of this volume.

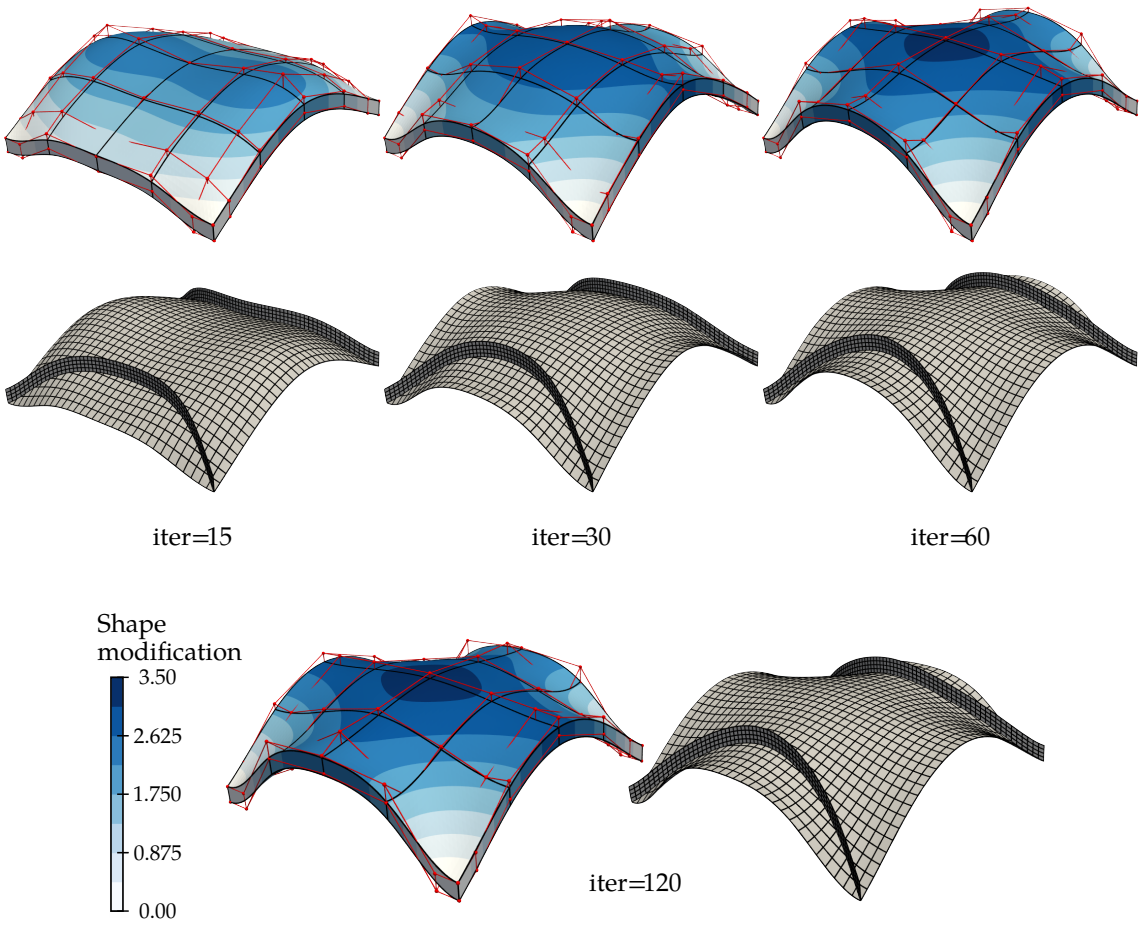


Figure 6.9 – Several intermediary shapes obtained during the optimization of the stiffened roof.

Two stiffeners are added to the initial square plate. We introduce two embedded surfaces to define these stiffeners as shown in figure 6.8. The construction can be divided into two steps. First, the volume mapping is generated by the extrusion of the square plate. We set the degree to one in the direction of extrusion (*i.e.* the knot vector in direction ξ_3 is defined as $W = \{0\,0\,1\,1\}$). Thus, the generation of the volume is straightforward. The volume shares half of its control points with those associated to the square plate. The additional control points are a copy of the previous ones where the z-component is set to an offset value. The second step consists in embedding surfaces into the parameter space of the created volume. These embedded surfaces, once composed with the volume mapping, give the final surfaces within the physical space that define the stiffeners. Here, the embedded surfaces are B-Splines of one single element with degree 2-by-1. Thus, each surface has six control points as depicted in figure 6.8. The coordinates of their control points can directly be interpreted from this figure considering the ratio $d_1/d_2 = 0.35$. The other geometric and mechanical parameters of the problem are given in figure 6.8.

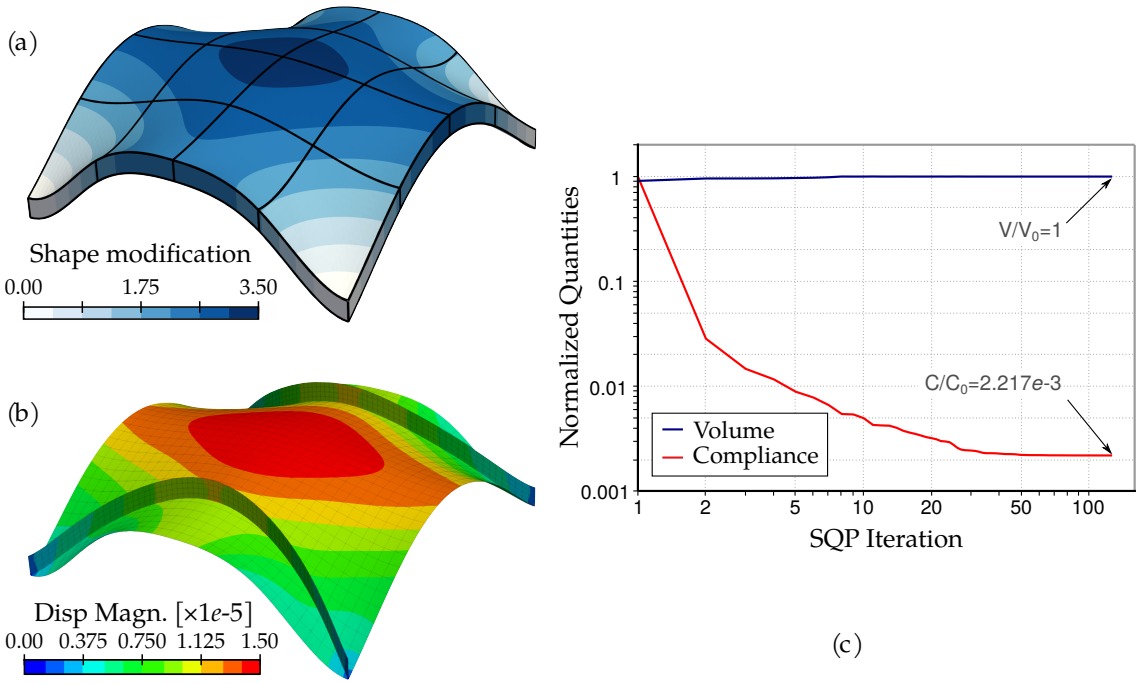


Figure 6.10 – Final design and result for the stiffened roof problem once the algorithm has converged. The optimal design is much more stiffer than the initial one.

In this problem, the embedded surfaces remain unchanged during the optimization. We apply the FFD design technique. The shape modifications of the stiffeners are only imposed by the design updates of the surface representing the roof. To this purpose, the design variables are associated to the control points of the mapping. The volume mapping is discretized using 4-by-4-by-1 elements. Degree 3 is taken in directions ξ_1 and ξ_2 while degree 1 is kept in direction ξ_3 (red points in figure 6.8). Thus, the optimization model has 45 design variables that move the control points of the mapping in the vertical direction z. The analysis model is defined such that the roof and both stiffeners are discretized into 1024 and 256 bi-cubic elements respectively. The area of the roof without the stiffeners is constrained to be lower than 110% of the initial area of the square plate (.i.e. lower than $V_0 = 110$).

Results Figure 6.9 shows the shape evolution of the optimization model and the analysis model. Due to the construction, the stiffeners follow exactly the shape modifications of the roof (the panel). No additional procedures are needed to enforce this constraint.

The final shape is obtained after 125 iterations of the optimization algorithm when setting the stopping criteria regarding the relative objective function to $1e$-8. Figure 6.10(c) depicts the evolution of the relative compliance during the resolution of the optimization problem. The compliance is drastically reduced: final relative compliance is equal to $c_{\text{opt}}/c_0 = 2.217e$-3. The optimized structure deforms much less than the initial stiff-

ened plate. The final displacement field is presented in figure 6.10(b). Looking at the continuity of the displacement field at the interface further confirms the suitability of the presented approach. It is interesting to note that the final shape has two planes of symmetry (see figure 6.10(a)). This is expected since the problem presents these symmetries. Therefore, some design variables have identical optimal values. The fact that we obtain the symmetries without setting groups of design variables is a meaningful indication to validate the result.

From an engineering point of view, this problem of the stiffened roof highlights an interesting issue. We manage to solve the optimization problem with the substructure. One could also simplify the problem by ignoring in the optimization process the stiffeners assuming that they only have small influence on the behaviour of the global structure. We tried to solve in a first step the optimization problem without the stiffeners, and we built the substructure afterwords. We computed the compliance on this new global structure and we got a compliance approximately 15% higher than the one obtained when optimizing the structure with the stiffeners. It highlights that tacking into account, during the optimization, sub-parts as stiffeners, holes, and other geometric details can be essential to design even better structures.

6.2.3 Local optimization: the curved wall

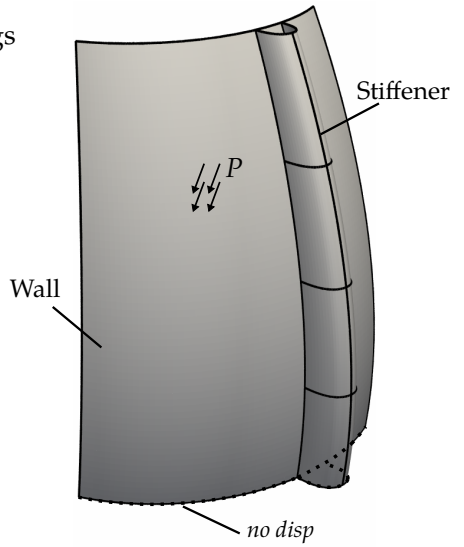
The curved wall problem highlights the potential of the presented approach to optimize the local shape of stiffened structures. Here we apply the embedded shape parametrization as introduced in chapter 5 (see more specifically figure 5.5, and section 5.3.2). The shape modification is done through the embedded surfaces while the mapping volumes are kept fixed. With this example of the curved wall we want to highlight that the approach enables to design stiffeners with complex shape. It is not restricted to flat stiffeners (as for the hyperbolic paraboloid with stiffeners from section 5.3.2).

Settings The construction principle of the stiffened curved wall is the same than the already presented examples. Starting from the surface defining the wall, we create a volume by extrusion. Once the volume is obtained, a surface is embedded in the parameter space of the volume. The composition of the embedded surface with the volume mapping gives the surface defining the stiffener in the physical space.

More specifically, the wall is defined by a quadratic NURBS surface. This surface originally represents a quarter cylinder. We move some control points to generate the final curved wall depicted in figure 6.11. We also give the coordinates of the control points and their weights in this figure. The embedded surface is defined by a quadratic NURBS with 2-by-4 elements. Its initial shape is described in figure 6.11.

In order to perform the shape optimization of the stiffener, a set of design variables is associated to the control points of the embedded surface. These design variables mod-

• Settings



Parameters:

Mechanical

E = 100 GPa, $\nu = 0.30$, $P = 0.1$ MPa

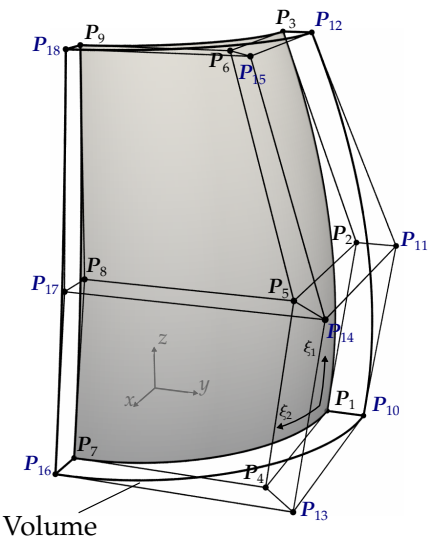
$t_w = 0.25$ m, $t_s = 0.1$ m

Optimization problem

minimize compliance

$A \leq A_0$

• Geometric Modeling / Shape Parametrization



x	y	z	w
0.	5.	0.	1.
1.	6.	5.	1.
0.	3.75	10.	1.
5.	5.	0.	$1./\sqrt{2}$
6.	6.	5.	$1./\sqrt{2}$
3.75	3.75	10.	$1./\sqrt{2}$
5.	0.	0.	1.
6.	1.	5.	1.
3.75	0.	10.	1.
0.	6.	0.	1.
1.	7.	5.	1.
0.	4.5	10.	1.
6.	6.	0.	$1./\sqrt{2}$
7.	7.	5.	$1./\sqrt{2}$
4.5	4.5	10.	$1./\sqrt{2}$
6.	0.	0.	1.
7.	1.	5.	1.
4.5	0.	10.	1.

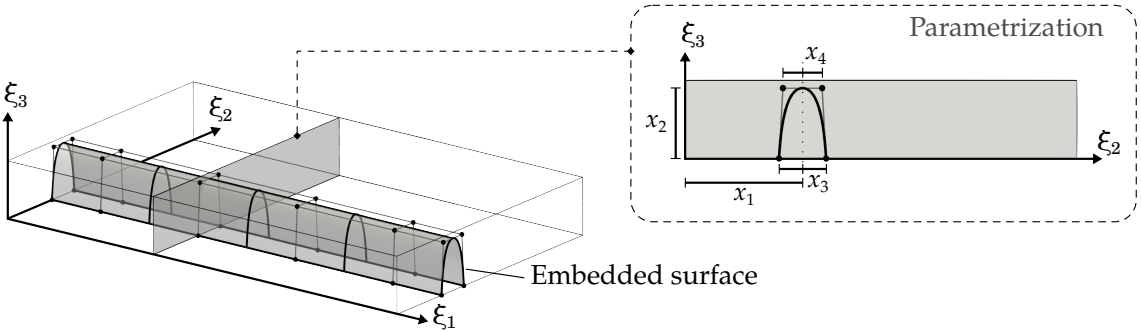


Figure 6.11 – Settings of the curved wall problem and construction of the model.

ify the cross-section and the position of the stiffener along the wall. With the chosen parametrization, six groups of four control points are spread out along the main direction of the stiffener. For each group of control points, we define four design variables $\mathbf{x}^i = [x_1^i, x_2^i, x_3^i, x_4^i]$ as explain in figure 6.11 (i identifies the group number). Thus, a total of $6 \times 4 = 24$ design variables is used for this example. The initial shape is obtained by setting the design variables of each group equal to $\mathbf{x}_0 = [0.3, 0.90, 0.06, 0.05]$. The area of the stiffener is constrained to be lower than the initial one which is $A_0 = 21.579\,\mathrm{m}^2$. Additional inequality constraints are included to the optimization problem in order to prevent undesired shapes during the resolution:

$$x_3^i - x_1^i \leq 0, \quad x_1^i + x_3^i - 1 \leq 0, \quad x_4^i - x_3^i \leq 0. \tag{6.2}$$

The first two constraints ensure that the embedded surface remains in the parameter space of the volume mapping. The third constraint limits the curvature of the cross-section of the stiffener. Finally, bound constraints set the range of variation of the design variables. The lower bounds are $\mathbf{x}_{\text{lo}} = [0., 0.1, 0.05, 0.01]$ and the upper bounds are $\mathbf{x}_{\text{up}} = [1., 1., 0.15, 0.15]$.

The analysis model is defined trough k-refinement of the optimization model. More specifically, the wall is discretized with 32-by-32 bi-cubic NURBS elements and the classical Kirchhoff–Love formulation is used. The embedded surface is discretized with 16-by-64 cubic NURBS elements and the embedded Kirchhoff–Love formulation is applied. The mechanical parameters of the problem are defined in figure 6.11. The pressure is applied over the mid-surface of the wall. The bottom of the wall and the bottom of the stiffener are both fixed. With this analysis model in hand, the compliance for the initial configuration of the multipatch structure is found to be equal to $c_0 = 38.79\,\mathrm{N\,m}$.

Results The optimization results are given in figures 6.12 and 6.13. During the resolution, the stiffener moves along the wall until it is located at the middle. The stiffener follows perfectly the curvature of the wall. With the embedded approach, no geometrical approximations at the interface are introduced. Moreover, it avoids calling on cumbersome procedures as mesh projection etc. The shape parametrization is finally quite simple (already presented in figure 6.11): the uni-directional translation of the embedded surface leads to the motion of the whole stiffener along the wall. This is true no matter the shape complexity of the wall and of the embedded surface.

Due to the symmetry of the problem, it was predictable that the final stiffener is located at the middle of the wall. The optimal shape has this symmetry which demonstrates the good quality of the result. Especially, the weak coupling between the non-conforming patches is well defined and is not source of inaccuracy during the shape updates. The cross-section of the stiffener is larger at the bottom than at the top of the wall. A large cross-section at the bottom improves the fixation of the overall structure. The cross-section also becomes larger where the deformations are critical. Reducing the deformation at the bottom decreases the global deformation of the structure. The

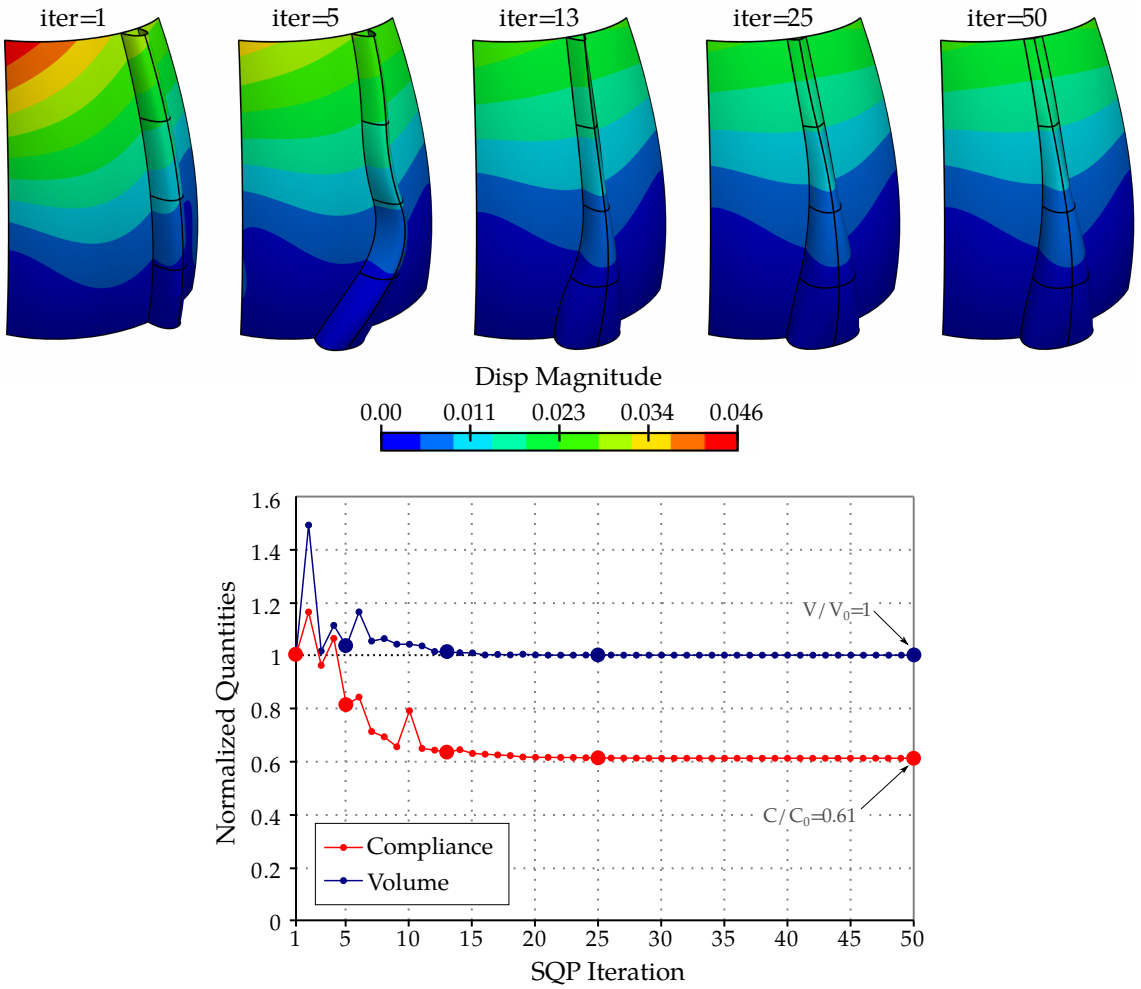


Figure 6.12 – Optimization histories for the resolution of the curved wall problem: evolution of the geometry and the displacement field, and evolution of the compliance and the volume.

compliance of the final multipatch shape is equal to $c_{\text{opt}} = 24.04\,\text{N m}$, which is 38% lower that for the initial configuration. More details on the optimal shape are given in figure 6.13. Information about the final coordinates of the control points associated to the embedded surface can be found in this figure.

We believe that our framework enables to solve this optimization problem of the curved wall in a suitable way. We are not sure whether commercial software as Optistruct (see www.altair.com/optistruct) could handle this type of shape optimization problem, and more specifically the junction between the stiffener and the wall during the shape modification of the stiffener. Thus, we believe that our framework offers new possibilities from the design point of view. It can be used to design new innovative stiffened structures. Indeed, it offers alternatives to straight and flat stiffeners which could further increase the performance of stiffened structures.

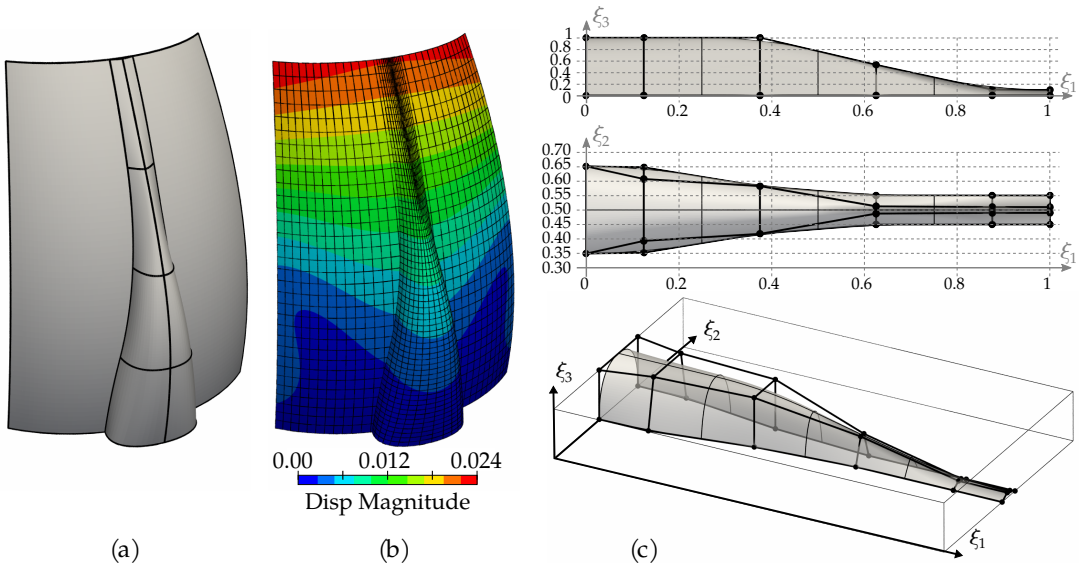


Figure 6.13 – Optimal configuration of the stiffened curved wall: (a) final geometry and (b) the magnitude of the displacement. (c) The final embedded surface: side, top views and 3D views. The 2D views give information on the coordinates of the control points.

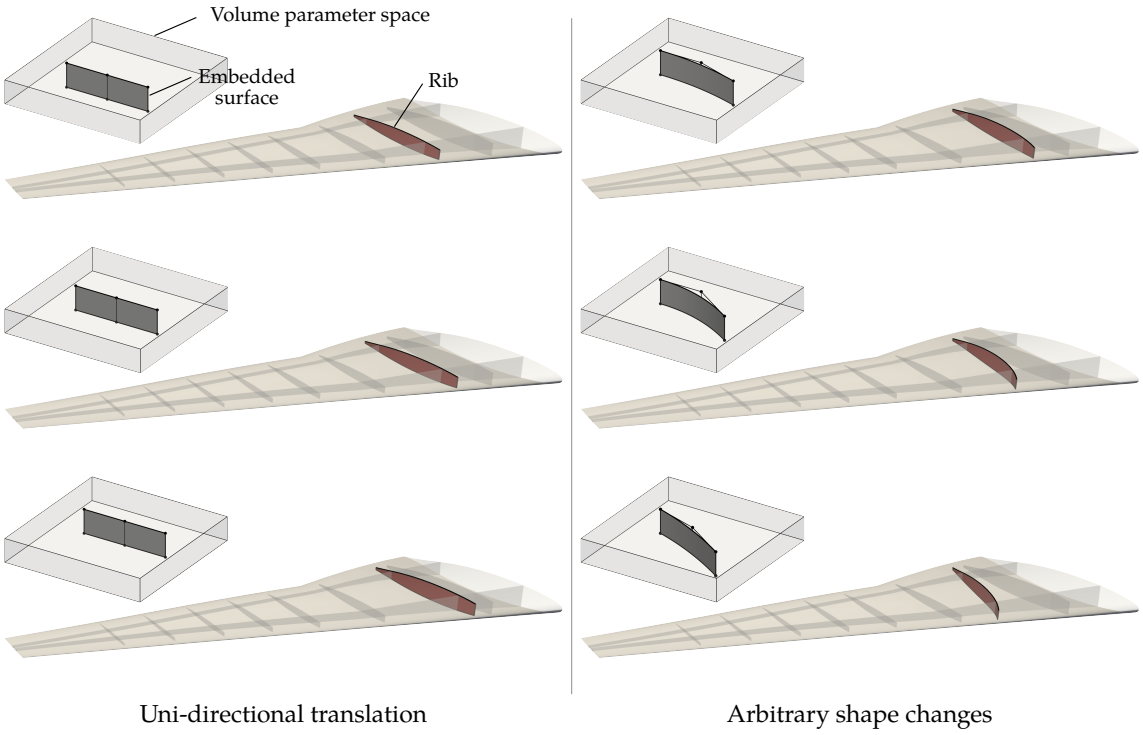
6.2.4 Designing an aircraft wing box

We show in section 6.2.1 how it is possible to generate a model of an aircraft wing with our framework. We saw that structural analysis can directly be applied on this model thanks to the developed tools. It is possible to go further by exploiting the ability of the model to impose shape changes, and thus performing shape optimization.

Shape parametrization Several parametrization-based methods have been recently suggested in order to obtain alternative designs to classical wing box designs characterized by straight and parallel ribs and spars [4, 8–10, 13]. The inspiration for curved structural members comes from dragonfly wings [218]. A good parametrization method should provide a large design space described by a relative small number of design variables [8]. Dubois et al. [13] underlines the importance of having a parametrization scheme that can be equipped with a full analytical sensitivity analysis. For complex geometries, obtaining gradient information is a challenging task [219]. The main issue comes from the numerous intermediary steps that separate the definition of the design parameters to the numerical results. Analytical sensitivities can be computed only if a close link between the design variables and the analysis model exists.

The adopted strategy has already been introduced in this work for others optimization problem (as for example the stiffened curved wall that we tackled just before). We use the embedded shape parametrization in order to prescribe shape changes to the structural members of the wings (spars and ribs). In fact, let us remind that we defined the geometry of these structural members through the use of NURBS composition of

- **Shape modification of the Ribs**



- **Shape modification of the Spars**

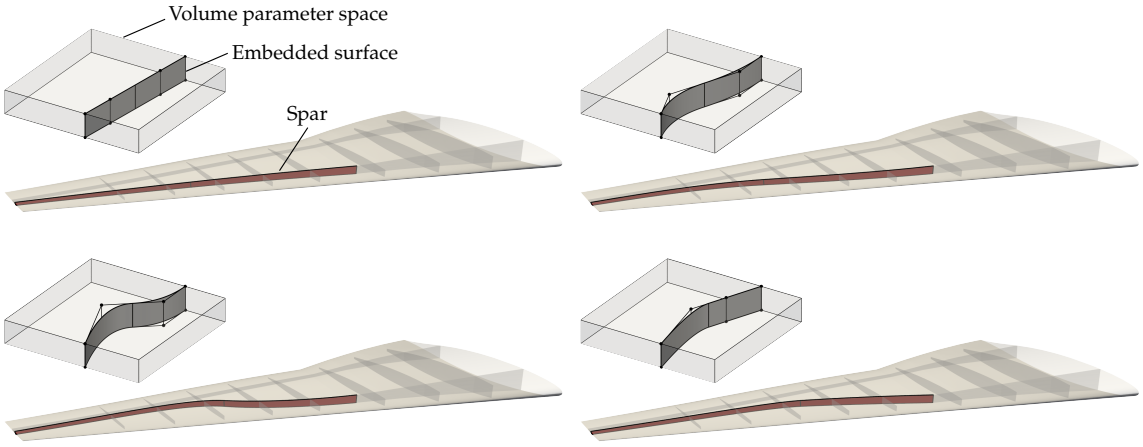


Figure 6.14 – Shape parametrization of the internal sub-structure of the wing. The shape modifications of the ribs and spars are done by moving the control points of the embedded surfaces.

embedded surfaces and mapping volumes (see figure 6.5). By moving the control points associated to the embedded surfaces, we have the possibility to impose shape modifications to the internal sub-structure of the wing. This process is highlighted in figure 6.14. Due to the construction, the curvature and the cross-section variation of the wing are automatically taken into account. The volume mapping serves the function of keeping the structural members inside the volume delimited by the skins. It ensures that the internal sub-structure is perfectly coincident with the skins independently of the complexity of the shape update. Thus, the designer does not need to worry about these geometrical constraints. No additional procedures are required that could have broken the link between the shape parameters and the final geometric model of the wing. It finally offers a convenient way to define the shape of the spars and ribs. In fact, it is based on NURBS which is recognized as a powerful tool for building parametrization-based approaches. In figure 6.14 we show how a uni-directional translation of the embedded surface in the parameter space of the mapping leads to a smooth translation of the rib which follows the skins. More arbitrary and complex shape changes can also be imposed in simple manner.

Finally let us mention that our method shares some similarities with existing frameworks. For instance, the *SpaRibs* framework [8–10] uses spline curves defined in a normalized one-by-one 2D square. The design variables are linked to the control points of these curves. Then, several steps including mapping, projection, extrusion, meshing, etc. lead to the final analysis model. The shape parametrization scheme presented by Dubois et al. [13] (which seems to be the only one that provides analytical sensitivity analysis) also starts with a 2D representation of the structural layout where each spar/rib is defined as a second-order polynomial. Then, each part is discretized and then extended vertically to generate a 3D representation of the wing structure. The extrusion should be aware of the geometry of the upper and lower skins of the wing. This can be complex in case of large curvature changes. It also forces the shape of the structural members to have a direction of extrusion. For instance, they cannot have holes or out-of-plane curvatures. With the presented embedded strategy, we can imagine to define the spars as cylinders instead of straight rectangles. This is the advantage of using a 3D representation of the ribs/spars at each step of the geometric modeling. Finally, the presented strategy can be applied to other purposes and is not limited to the parametrization of wing-boxes. Shape parametrized models of a fuselage can be done by following the exact same methodology.

Shape Optimization It is time to exploit the generated model of the aircraft wing and to perform a shape optimization of the internal sub-structure. The problem is presented in figure 6.15. The geometric and mechanical settings are the same than for the structural analysis of the wing presented previously in section 6.2.1. The definition of the design variables is depicted in figure 6.15. Each rib is described with a B-Spline of one single element with degree two in the main direction, and degree one in the vertical direction.

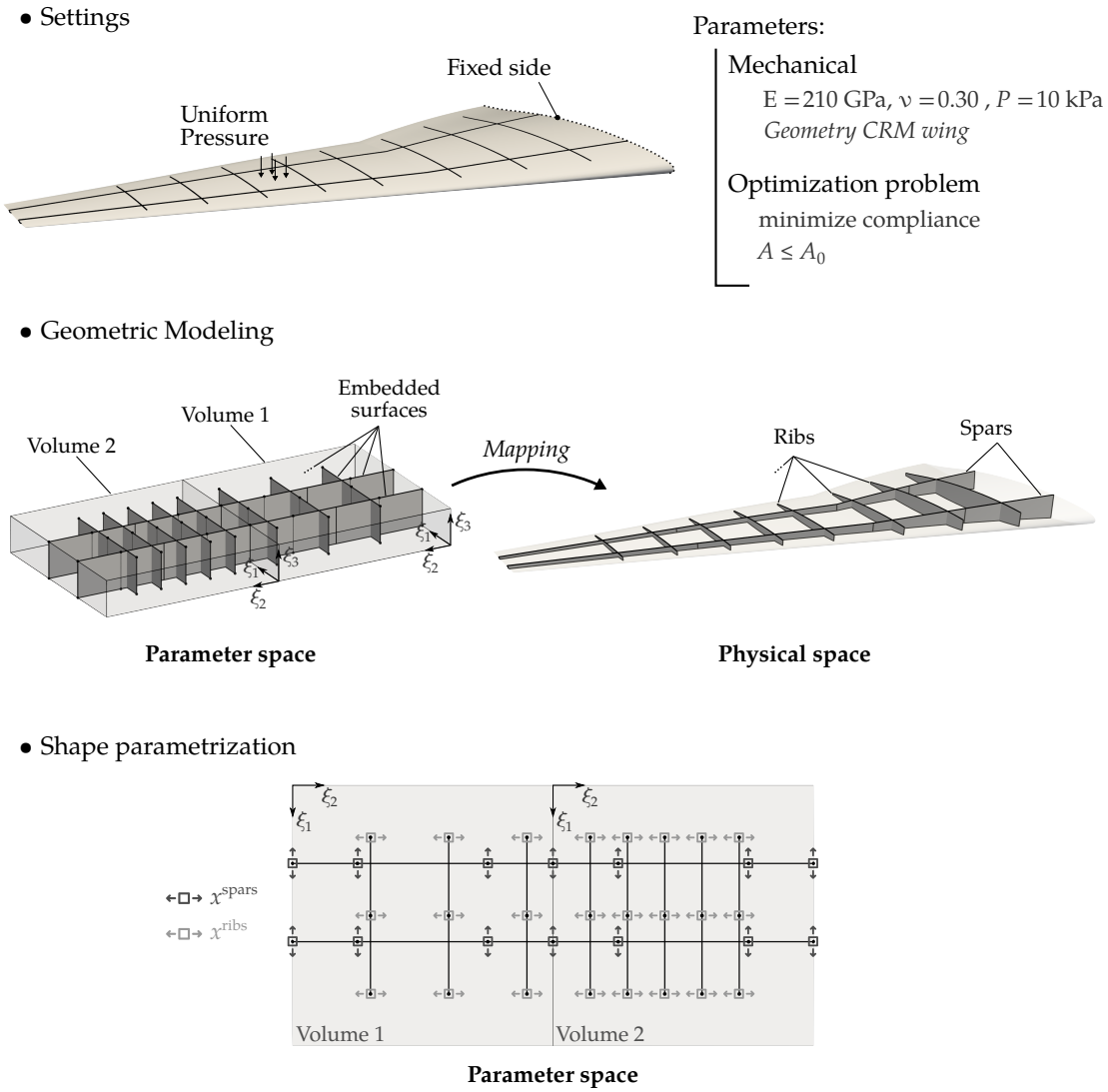


Figure 6.15 – Settings for the shape optimization problem of the internal sub-structure of a wing.

The same goes for the spars except that they are discretized with two elements which allows to set more design parameters. The control points of the spars are movable in the ξ_1-direction of the volume parameter space. The same motion is prescribed to the edges joining the upper and the lower skin. Thus, four design variables per spar are set. One common design variable is defined at the junction between the spars embedded in the volume 1 and 2 in order to maintain there connection. Additionally, three design variables per rib are defined, and move the control points in the ξ_2-direction of the volume parameter space (see figure 6.15). We end up with a total of 38 design variables. Here we choose to prevent the crossing between the ribs during the shape evolution. We add geometric constraints into the formulation of the optimization problem that guarantees a minimal distance between the ribs. It is simply done in this work by

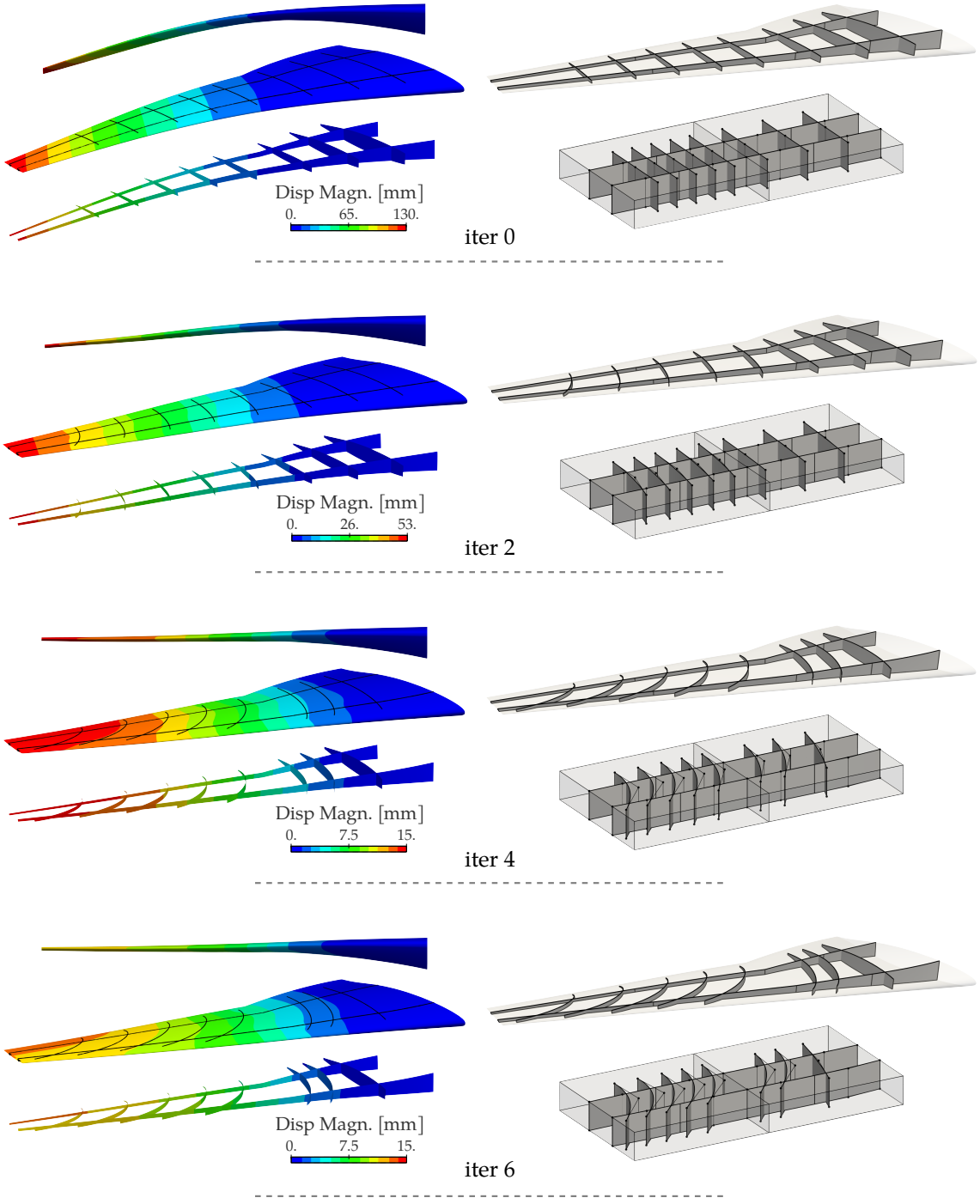


Figure 6.16 – Shape evolution during the optimization of the internal sub-structure of the wing.

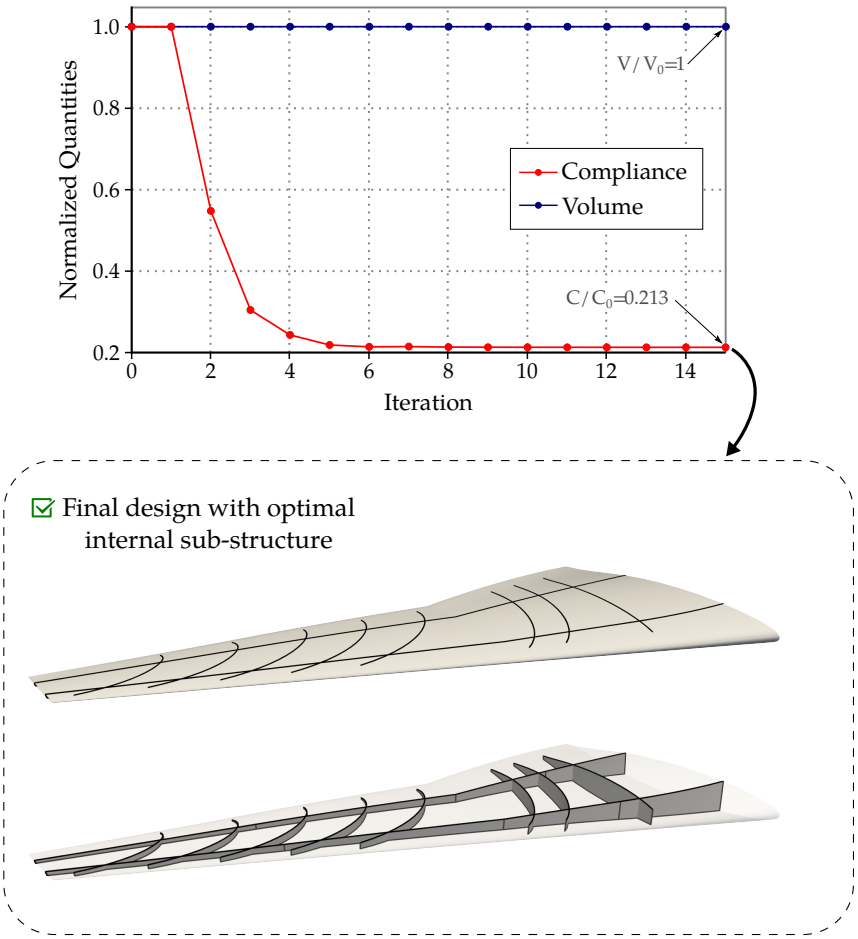


Figure 6.17 – Optimization histories for the resolution of the wing problem, and final design of the ribs and spars.

settings a minimal difference between two design variables of two neighboring ribs. We also voluntarily skip to enforce the coupling between the ribs and the spars for the sake of simplicity. In fact it avoids to locate their intersections during the optimization process (see the discussion from section 5.3.1.2).

The optimization results are given in figures 6.16 and 6.17. The evolution of the shape of the internal sub-structure is shown in figure 6.16. One can observe the shape modifications of the embedded surfaces and their consequences on the shape of the ribs and the spars. One could also observe the evolution of the deformation of the wing. The overall displacement of the wing is drastically reduced during the optimization. The maximal displacement of the final design is 10 times lower than the one of the initial design (129 mm for the initial versus 12.8 mm for the optimal). The compliance is reduced to a great extent, $c^{\text{opt}}/c_0 = 0.213$, whereas the overall area (in other words the mass) is kept constant. For the resolution, we use the MMA algorithm [220] due to the quite large number of constraints we set in the optimization problem. The convergence

is fast for the example as shown in figure 6.17.

The final design has ribs and spars with curved shapes. It proves that curved structural members can largely improve the mechanical behavior of the wing. It coincides with the observations from other works [4–14]. Such innovative designs can surely improve further aircraft structures, and more generally, stiffened structures. It also motivates the construction of robust and efficient frameworks that enable to design these new future designs. We hope that this example confirms the potential of the methodology developed in this work.

Chapter 7

Conclusion

7.1 Final Overview

The core objective of this thesis was to study the benefits of IsoGeometric Analysis for the shape optimization of structures, and especially, for the shape optimization of complex stiffened structures as found in aeronautics. The benefits of IGA for shape optimization turns out to be true in view of existing works. However, several issues needed to be tackled to broaden the scope of application of isogeometric shape optimization.

More specifically, we started in chapter 3 by looking at the specific case of shells. In the context of isogeometric shape optimization of shells, very little work has tackled the influence of the chosen shell formulation. Thus, we explored the use of a solid-shell formulation for the shape optimization of thin-structures. We carried out a range of numerical experiments to compare the results of the proposed solid-shell approach with the commonly adopted technique based on the Kirchhoff–Love formulation. Similar results have been observed for the two shell formulations in terms of final optimized design and in terms of global behavior during the optimization process (convergence speed, intermediary shapes). Unlike pure shell elements, we emphasize that the solid-shell element used in this work is based on a 3D solid continuum element. With any simple IGA code in hand, one can perform shape optimization of shells without almost any additional developments by imposing identical shape updates on both inner and outer surfaces. Of course, this is restricted to preliminary studies, and more accurate solid-shell formulations that treat the numerical locking should be used to build a robust framework.

We also developed analytical sensitivities for both Kirchhoff–Love and solid-shell formulations in the context of linear elasticity. The presented sensitivities are adjoint-based discrete sensitivities. Their expression is based on the use of curvilinear coordinates in the element formulation. This is very convenient with IGA since splines represent the geometries in the parametric form. We mainly considered the case of the compliance

but the developed analytical sensitivities enable to tackle other structural responses as a specific displacement or stress components, etc. Because the considered solid-shell formulation is based on a 3D solid continuum element, the analytical sensitivities are also available for standard solid elements (we deal with both 2D and 3D examples). The numerical cost of these analytical sensitivities turned out to be low which is essential for complex structures.

Finally, the major interest of solid-shell elements concerns the geometric representation of the thickness. Continuous and smooth thickness variations can be imposed by modifying the control point coordinates of one outer surface. Thanks to NURBS functions, very few design variables were required to describe quite complex geometries. The direct link with the CAD format provided by the NURBS functions is also highly attractive in this context. The volume representation of the structure makes the optimal shape directly available for CAD drawings, and thus the transfer to the manufacturing step is simplified. The presented multi-model optimization process combined the benefits of both Kirchhoff–Love shell and solid-shell. It consisted in using the Kirchhoff–Love shell in a first stage to optimize the global shape of a structure. In a second stage, thickness optimization was operated on a high-fidelity volume model using the solid-shell approach.

In order to deal with the shape optimization of stiffened structures, we required to face the difficulty of handling multipatch models with non-conforming parametrizations. This is the cornerstone of the problem and developments have been done from both the geometric and analysis points of view. From the analysis side, we built a dual Domain Decomposition solver for the analysis of non-conforming multipatch isogeometric shell models (see chapter 4). The starting point was the definition of a Mortar coupling which weakly imposes the coupling conditions through Lagrange Multipliers. Thus, we developed a new Mortar method for the coupling of Kirchhoff–Love shells. In the literature, one argument against Mortar methods is commonly claimed; it concerns the additional unknowns introduced by the Lagrange multiplier field that increase the overall size of the system to be solved. We believe that this argument may be not relevant and the present work fully takes advantage of those additional DOF. In fact, it allowed us to formulate an interface problem, namely the one-level FETI problem, where the unknowns of the system were these interface DOF. We highlighted for different examples with increasing levels of complexity that it enabled to accurately analyze non-conforming multipatch structures, even for very fine level of refinement, since we tracked the interface residual during the resolution.

From the geometric side, we introduced in chapter 5 an embedded strategy in order to properly impose shape changes to stiffened structures. It consisted in geometrically defining the shell mid-surface by a NURBS composition of an embedded surface with a volume mapping. We illustrated this construction principle on multiple examples and

we applied it for stiffened structures. We managed to directly use the NURBS composition for the analysis by introducing the embedded Kirchhoff–Love shell formulation. This means that the original idea behind isogeometric analysis was maintained since both the design and the analysis are performed on an common model. It is particularly attractive when it comes to structural optimization. This way, analytical sensitivities were achieved in this work. Moreover, the presented Free-Form Deformation technique offered a smooth and an effective way to impose complex shape variations. It eliminated sophisticated geometric rules, reparametrizations and other approximations one could introduce when designing large and complex multipatch structures.

Thus, we were able, for example, to create a versatile model of a wing with its substructure. Not only the position and the shape of the stiffeners were easily modified but also the shape of the whole design of the wing could have been parametrized. One could have changed the shape of the outer skins by acting on the control points associated to the volume mapping. Due to the use of embedded entities, the internal substructure of the wing automatically would have followed the update of the outer geometry.

Finally, we ended up with a consistent framework to perform the shape optimization of complex structures. In chapter 6, we highlighted how all the developed methods of this thesis could be merged into one single framework in order to tackle industrial problems. It led to innovative designs of stiffened structures. It brougt the possibility to explore new types of design, and finally further improve the mechanical behavior of aerostructures. In the usual aero-structural optimization framework, it seems that mainly sizing variables such as the thickness are introduced into the design process. The use of an isogeometric model with embedded entities can be very attractive to enlarge the design space by adding shape variables as the position and the curvature of the stiffeners.

7.2 Prospects

This works opens up new promising perspectives and motivations for further improvements. Let us mention several issues that come to mind.

Firstly, we hope that this thesis motivates researches on IGA-based shape optimization. We truly believe that shape optimization is a great application of IGA, and that, consequently, IGA could democratize the use of shape optimization methods in design offices. In order to do so, there still exist points to look at. For instance, we did not study the influence of the optimization algorithm. Instead, we mainly used black-box solvers in this work. Designing specific optimization algorithms for IGA-based shape optimization appears essential to get a general method that works for any given shape parametrization. The question concerns the update of the design variables. The shape updates should be managed such that mesh overlapping and self-intersection are pre-

vented. The handling of such free-form shape parametrization is required to automatize the setting of new optimization problems.

The analytical sensitivities could also be enlarged to other structural behaviors as non-linearity (geometric or material). It seems also crucial to take into account other phenomena such as buckling. This is particularly true for stiffened structures. Due to their simplicity, those mechanical behavior should be more easily tracked with solid-shell formulations than with pure shell formulations. Indeed, differentiating the geometric stiffness matrix of the Kirchhoff–Love shell formulation *w.r.t.* the control point coordinates involves much more terms than in case of the solid-shell formulation.

Regarding the multipatch modeling, the developed solver is naturally parallelizable. In fact, the presented algorithm belongs to the FETI family which has been intensively developed over the past decades. Those Domain Decomposition algorithms have shown great performances to analyze very large problems with millions of DOF. Implementing the overall developed framework in parallel is necessary to handle industrial structures. Furthermore, alternatives to the initial one-level FETI problem, regarding for example the formulation of the interface problem or the preconditioning step, have been suggested and studied in the literature. All those works can be applied to develop efficient analysis solvers for complex isogeometric models. The present work aims at motivating researches in that direction. Finally, the approximation space of the Lagrange multipliers in the Mortar approach is of high interest. Optimal discretization spaces still need to be formulated in case of non-conforming Kirchhoff–Love shells. This would improve the quality of the multipatch solutions. Thus, the direct use of geometric models with non-conforming interfaces and trimming procedures for the structural analysis could be pushed forward with those new Domain Decomposition solvers and optimal Mortar coupling approaches.

The application of the developed strategies is surely not limited to stiffened aerostructures. The presented approach is versatile and can be applied to other purposes. Note that, in order to do so, the availability of pre-processing tools is necessary. Developing these tools is essential to democratize the use of IGA-based shape optimization and even the use of Isogeometric Analysis.

Appendix A

Résumé des travaux (en Français)

Ce chapitre reprend en français les éléments principaux présentés dans cette thèse. La structure de ce résumé est semblable à l'organisation du manuscrit. Il s'agit dans un premier temps de motiver l'utilisation de l'Analyse IsoGéométrique pour l'optimisation paramétrique des structures coques. Nous rappelons certains éléments théoriques centraux et montrons l'intérêt des formulations massifs coques et Kirchhoff–Love dans ce contexte. Afin de traiter le cas des structures raidies, nous introduisons un algorithme basé sur les méthodes de décomposition de domaine pour analyser efficacement des modèles aux discrétisations non-conformes. Enfin, une approche immergée issue de l'infographie 3D nous permet de paramétrer tant la forme globale de la structure que la forme des raidisseurs. Des exemples permettent de justifier le potentiel de l'approche développée tout au long de ce travail.

Contents

A.1 Introduction

L'optimisation de forme des structures requiert une combinaison harmonieuse entre une description géométrique précise et un modèle d'analyse performant d'un même objet. Plus encore, un lien étroit entre les modèles géométrique et d'analyse est nécessaire puisqu'ils communiquent simultanément durant la résolution du problème d'optimisation. Ainsi, l'optimisation de forme est une application intéressante de l'Analyse Isogéométrique (IGA) [17] car cette méthode s'appuie sur les fonctions NURBS pour représenter d'une part la géométrie et, d'autre part pour effectuer l'analyse. Cet atout a été démontré et l'optimisation isogéométrique de forme suscite un intérêt grandissant depuis 10 ans, et cela pour différents champs d'application [18, 67, 72].

L'optimisation isogéométrique de forme reste cependant restreint à des cas d'école et la viabilité de l'approche n'a pas encore été démontré pour des cas plus complexes semblables à des cas industriels. Dans ce travail, nous cherchons à développer des outils numériques d'aide à la conception des structures raidies comme on peut les trouver en aéronautique. Ces structures sont obtenues par l'assemblage d'une pièce principale (appelée la panneau ou la peau) et de sous-structures (appelés les raidisseurs). C'est l'assemblage des raidisseurs avec le panneau qui donne une structure finale avec un comportement mécanique intéressant pour une masse généralement faible. On retrouve ce principe de construction pour le fuselage ou encore les ailes (voir figure 1.1). Des travaux récents ont montré qu'une voie d'amélioration dans la conception de ces structures raidies consiste à proposer des formes innovantes. En effet, au lieu de concevoir des raidisseurs droits et de les disposer de manières uniformes tout le long de la structure, il peut être bénéfique de concevoir des raidisseurs présentant des courbures et de les disposer plus spécifiquement (cf. figure 1.2). On propose d'étendre les outils d'optimisation isogéométrique de forme pour construire une stratégie de conception des structures raidies innovantes. Bien que l'objectif semble restreint à une application précise, les outils et les méthodologies développés sont finalement généralisables et ont pour but d'être appliqués à bien d'autres applications.

Nous reprenons ici les points principaux présentés dans ce manuscrit de thèse. La section A.2 présente les bases sur l'optimisation isogéométrique de forme pour les structures coques. On présente une approche multi-modèle qui combine les avantages des formulations coques de type Kirchhoff–Love et à ceux apportés par des éléments massifs coques. La section A.3 introduit les outils développés pour effectuer l'analyse sur des modèles aux discrétisations non-conformes. Finalement, la section A.4 présente l'approche immergée qui permet d'imposer des variations de formes sur des structures complexes, et plus spécifiquement les structures raidies. Cela nous amènera à formuler des conclusions sur ce travail et de présenter les perspectives qu'il nous semble intéressantes.

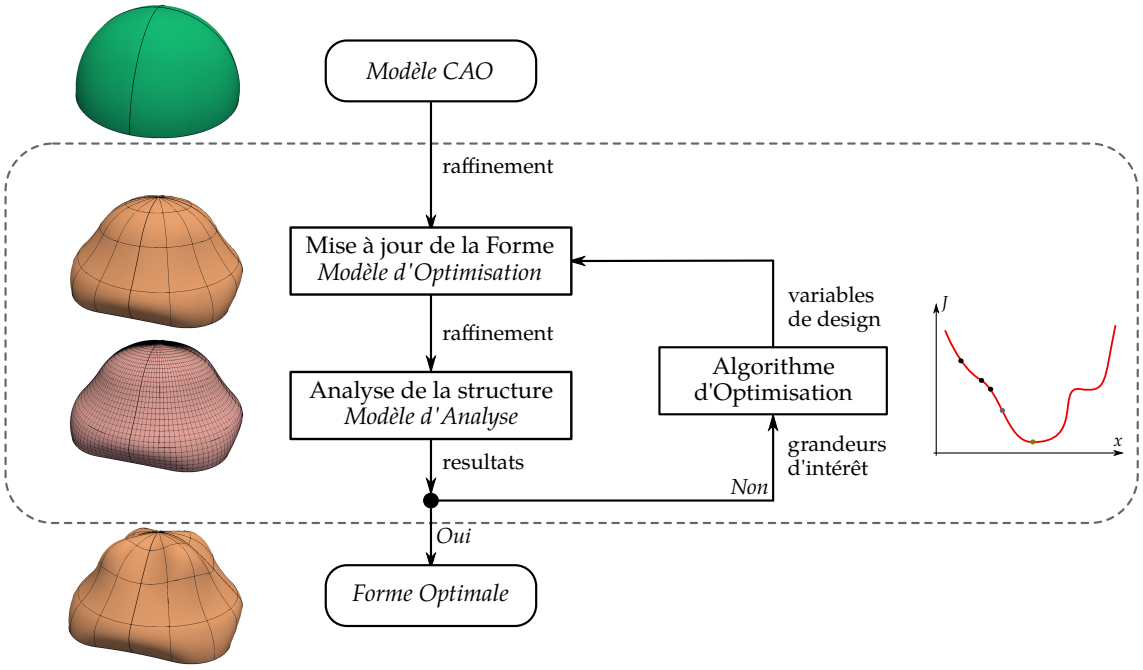


Figure A.1 – Fonctionnement de l'optimisation de forme dans le cadre de l'IGA ; visualisation des étapes principales.

A.2 Optimisation Isogéométrique des Structures coques

A.2.1 Optimisation paramétrique

La méthode d'optimisation adoptée dans ce travail s'appuie sur des travaux déjà existant quant à l'utilisation de l'analyse isogéométrique pour les problèmes d'optimisation paramétrique de forme [87, 88]. Le fonctionnement général de l'approche est présenté en figure A.1. A partir d'un modèle CAO initial (modèle NURBS), la forme optimale est obtenue itérativement. Chaque itération contient trois étapes principales :

— mise à jour de la géométrie en déplaçant les points de contrôle et, dans certains cas, en modifiant leurs poids,
— effectuer l'analyse de structure pour cette nouvelle géométrie et calculer des grandeurs d'intérêts,
— traiter les données issues de l'analyse pour déterminer la prochaine modification de forme.

Un point intéressant apporté par l'utilisation de l'analyse isogéométrique est la possibilité de choisir proprement les espaces d'optimisation et d'analyse [68-72, 75, 84]. En effet, une discrétisation fine est choisie afin de garantir une bonne qualité de résolution. D'un autre côté, le modèle d'optimisation est défini de manière à contrôler les variations de forme.

Le niveau de raffinement du modèle d'optimisation est généralement moins élevé

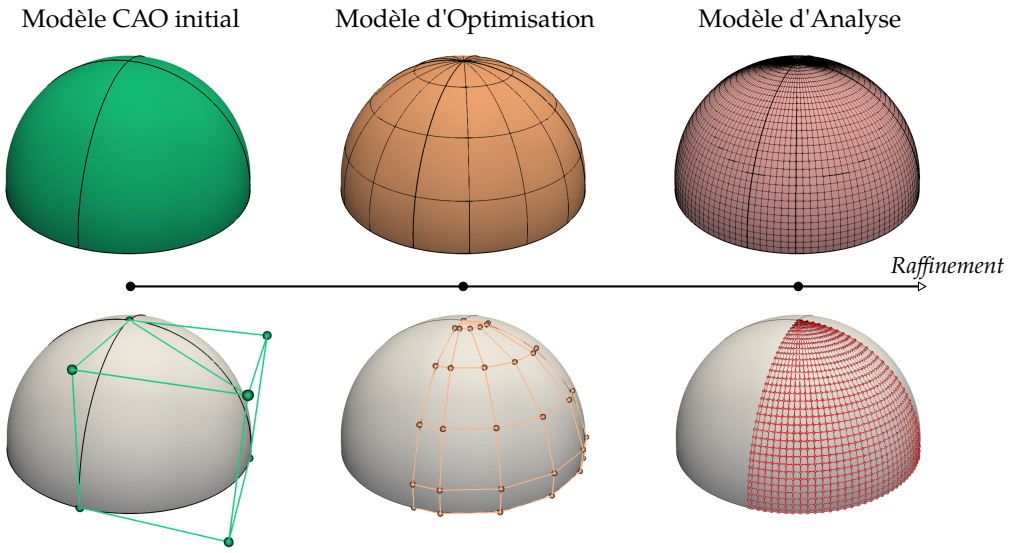


Figure A.2 – Optimisation de forme dans le cadre de l'IGA ; l'optimisation peut s'effectuer sur le modèle CAO initial, sur le modèle fin d'analyse, ou pour des niveaux de raffinement intermédiaires.

que celui du modèle d'analyse (voir figure A.2). En effet, un modèle d'optimisation fin peut engendrer une forme finale très complexe. Réduire son niveau de raffinement garantit que la forme obtenue ait un niveau de complexité contrôlé (courbure maximale par exemple). Ce qui différencient les modèles d'analyse et d'optimisation est la densité de leur polygone de contrôle. Cependant, tous deux représentent exactement la même géométrie. Aussi, le passage de l'un à l'autre s'effectue facilement grâce aux procédures de raffinement propres aux NURBS (et splines en général).

A.2.2 Calculs de coques

Deux formulations coques différentes sont envisagées dans ce travail : une formulation massif-coque et une formulation Kirchhoff–Love (voir figure A.3). De nombreux travaux envisagent l'utilisation d'éléments de type Kirchhoff–Love [71, 72, 122]. Ces éléments coques ont l'avantage d'être peu gourmand en ressources numériques. Cependant nous montrons que les éléments massifs coques peuvent aussi être intéressants dans le contexte de l'optimisation de forme : la représentation explicite de l'épaisseur permet de représenter des variations continues de celle-ci.

Formulation massif coque La formulation massif-coque employée dans ce travail s'appuie sur les observations faites par Bouclier et al. [128]. Elle fait appel à une cinématique 3D classique. L'idée est d'utiliser qu'un seul élément solide dans l'épaisseur de la coque. De plus, un degré deux est fixé dans l'épaisseur peu importe le degré choisi dans les directions principales. Par commodité, on utilise ici les coordonnées curvilignes

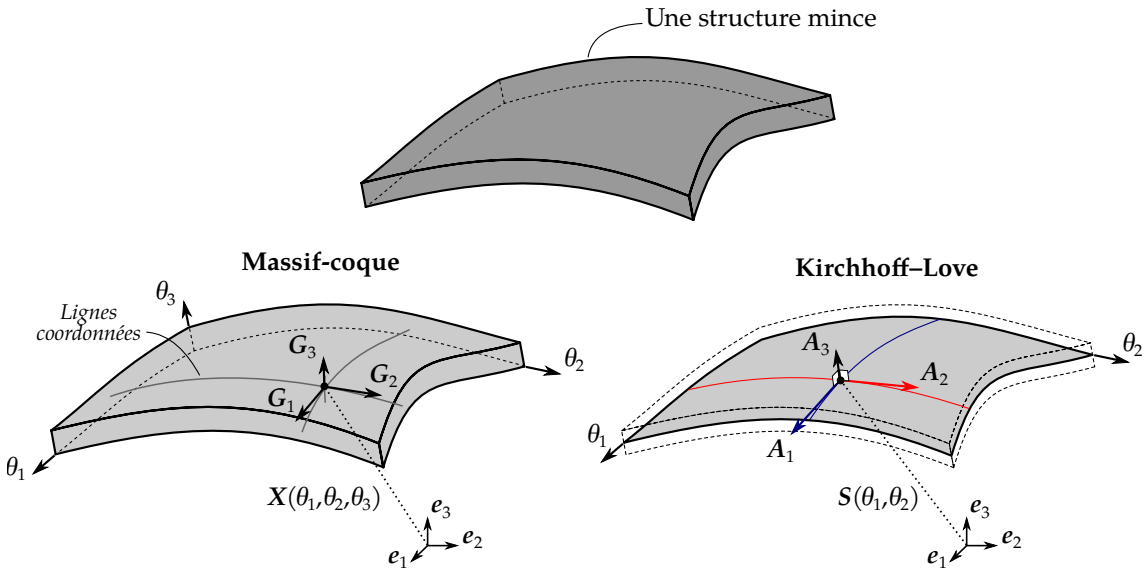


Figure A.3 – Deux approches pour le calcul de coques : l'approche massif-coque où la structure est entièrement représenté et l'approche Kirchhoff–Love où seulement structure est décrite à partir de la surface moyenne uniquement.

pour formuler l'élément : le vecteur position prend la forme d'un volume paramétrique (*i.e.* de la forme $\boldsymbol{X} := \boldsymbol{X}(\theta_1, \theta_2, \theta_3)$). Dans le cadre de l'élasticité linéaire, le tenseur de Green-Lagrange généralisé s'écrit (dans un système de coordonnées curvilignes) :

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left(\boldsymbol{u}_{,i} \cdot \boldsymbol{G}_j + \boldsymbol{u}_{,j} \cdot \boldsymbol{G}_i\right) \boldsymbol{G}^i \otimes \boldsymbol{G}^j. \tag{A.1}$$

où $\boldsymbol{G}_i$ et $\boldsymbol{G}^i$ sont respectivement les vecteurs covariants et contravariants. Tous les détails de la formulation sont présentés en section 3.2.1.

Formulation Kirchhoff–Love Une formulation isogéométrique de type Kirchhoff–Love est également utilisée dans ce travail. L'idée est de décrire le comportement de la structure globale uniquement à l'aide de quantités définies sur la surface moyenne. Ainsi, la géométrie globale (le volume) est décrit ainsi :

$$\boldsymbol{X}(\theta_1, \theta_2, \theta_3) = \boldsymbol{S}(\theta_1, \theta_2) + \theta_3 \boldsymbol{A}_3(\theta_1, \theta_2), \quad \theta_3 \in [-\tfrac{t}{2}, \tfrac{t}{2}], \tag{A.2}$$

où le vecteur directeur est défini comme le vecteur unitaire normal à la surface tout au long de la déformation. Par conséquent, les déformations de cisaillement transverse sont négligées limitant cette formulation au cas de structures minces. A nouveau, les détails quant à cette formulation sont donnés en section 3.2.2.

A.2.3 Une approche multi-modèle

Dans ce travail, plusieurs exemples ont été présentés pour montrer la viabilité de l'approche massif coque pour les problèmes d'optimisation des structures minces. Pour

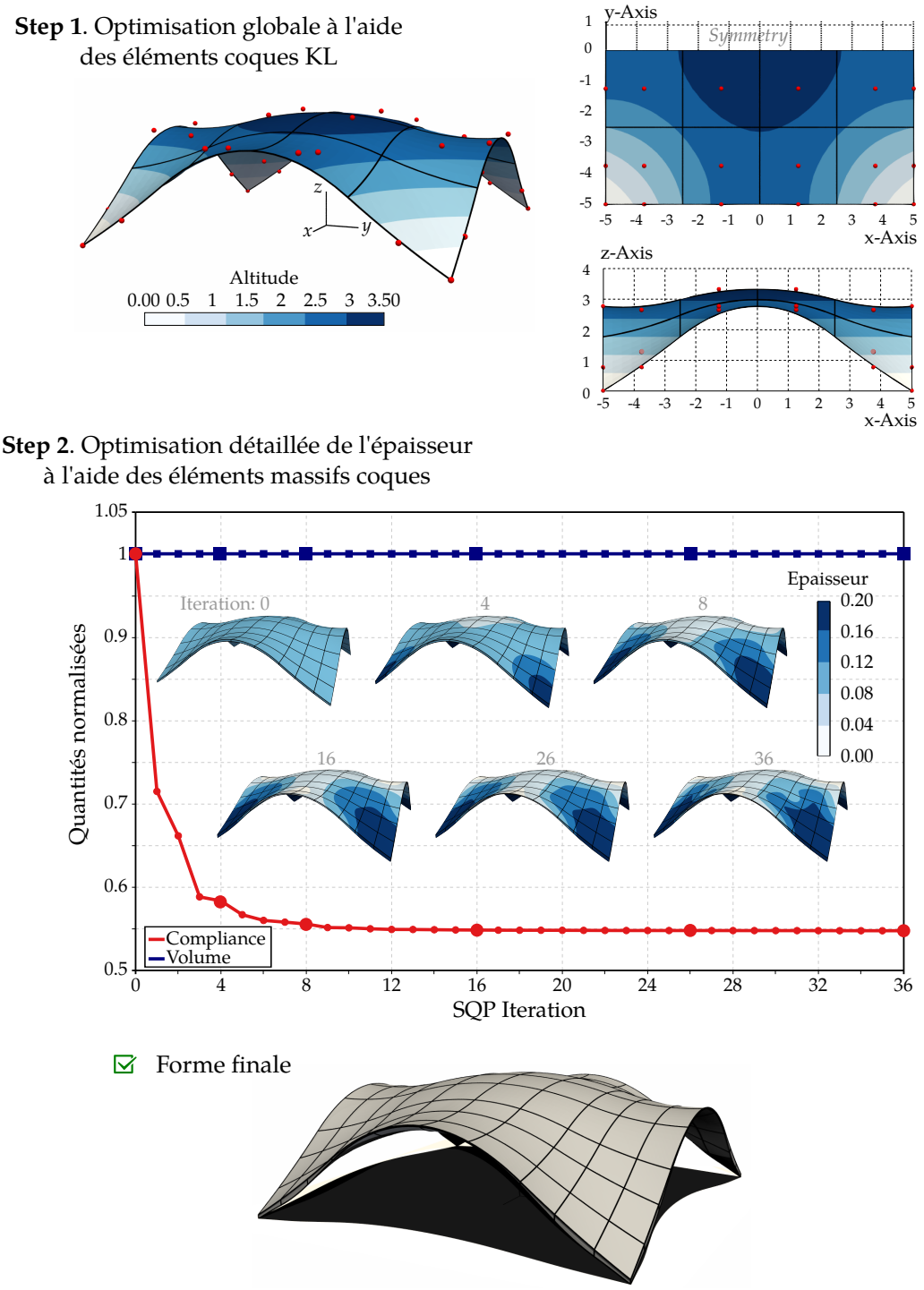


Figure A.4 – Approche mutli-modèle : dans une première phase de conception, une formulation coque type Kirchhoff–Love permet d'optimiser la forme globale de la structure. De cette première étape, un modèle volumique est généré afin de représenter explicitement toute la structure. Ensuite, l'approche basée sur les éléments massifs coques permet d'optimiser la carte d'épaisseur jusqu'à obtenir la géométrie finale.

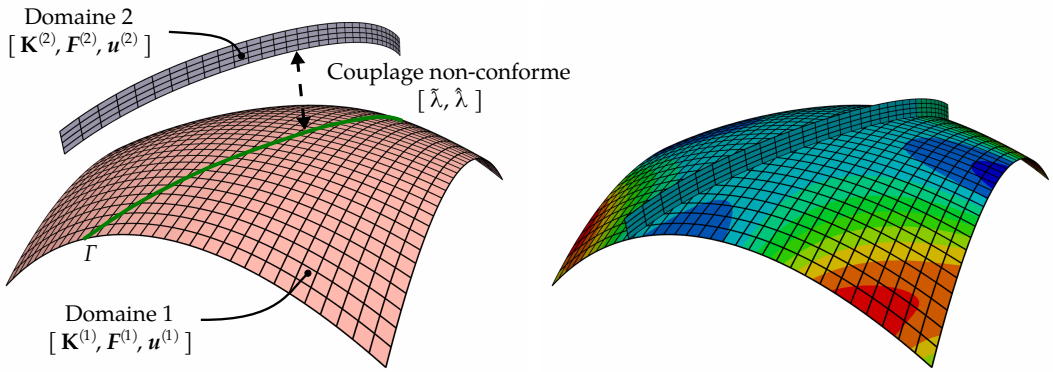


Figure A.5 – Couplage par multiplicateurs de Lagrange de deux domaines non-conformes à l'interface.

cela nous avons comparé sur des exemples simples les résultats obtenus à l'aide des deux approches massif coques et Kirchhoff–Love. Cette étude est donnée en section 3.4.1 et montre qu'elles conduisent à des géométries optimales similaires.

L'intérêt d'utiliser des éléments peu coûteux en terme de temps de calcul comme le sont les éléments Kirchhoff–Love, est de pouvoir explorer une large variété de forme rapidement. Ainsi, cette approche est très intéressante en phase de prototypage où plusieurs configurations peuvent être traitées. Une fois qu'une idée globale de la forme optimale est obtenue, l'approche massif coque permet de faire une étude plus détaillée et d'introduire des variations d'épaisseur afin d'apporter des améliorations supplémentaires quant au comportement de la structure. C'est l'approche multi-modèle appliquée pour le cas d'un toit en figure A.4. Finalement le résultat final est directement exploitable pour la phase production. Le modèle obtenu s'appuie sur une description volumique de la géométrie et utilise des outils issus de la CAO (B-Spline/NURBS etc.).

A.3 Calculs multipatch pour les structures complexes

A.3.1 Couplage Mortar

Les structures raidies sont obligatoirement modélisées par plusieurs patchs : plusieurs surfaces NURBS et/ou compositions de NURBS. Pour pouvoir analyser la structure complète il faut donc coupler ces patchs. Les discrétisations sont généralement non-conformes aux interfaces (voir figure A.5), et les conditions de raccords sont alors nécessairement imposées faiblement. Considérons le cas de deux domaines Ω_1 et Ω_2 modélisés par des coques dont l'intersection Γ décrit une courbe (cas sans recouvrement). Les conditions cinématiques de raccord s'écrivent ainsi :

$$u^{(1)} = u^{(2)} \quad \text{on } \Gamma, \tag{A.3}$$

$$\Phi^{(1)} \cdot \boldsymbol{t} = \Phi^{(2)} \cdot \boldsymbol{t} \quad \text{on } \Gamma, \tag{A.4}$$

A l'interface, on impose la continuité des déplacements. De plus, on impose la continuité de la rotation de la surface moyenne par rapport à la tangente à l'interface $\boldsymbol{t}$. Il existe différentes méthodes pour imposer faiblement ces conditions cinématiques. On peut citer les approches de couplage par pénalisation, par multiplicateur de Lagrange ou encore les méthodes de Nitsche. Dans ces travaux, nous adoptons une méthode de couplage par multiplicateurs de Lagrange. On introduit deux multiplicateurs de Lagrange $\tilde{\lambda} \in \mathcal{L}_d$ et $\hat{\lambda} \in \mathcal{L}_r$ ($\mathcal{L}_d$ et $\mathcal{L}_r$ étant des espaces *Ad hoc* pour les déplacements et la rotation respectivement). Le principe variationnel écrit sous forme discrète donne le système linéaire à résoudre :

$$\begin{bmatrix} \mathbf{K}^{(1)} & 0 & \mathbf{C}^{(1)T} & \mathbf{Z}^{(1)T} \\ 0 & \mathbf{K}^{(2)} & \mathbf{C}^{(2)T} & \mathbf{Z}^{(2)T} \\ \mathbf{C}^{(1)} & \mathbf{C}^{(2)} & 0 & 0 \\ \mathbf{Z}^{(1)} & \mathbf{Z}^{(2)} & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \\ \tilde{\boldsymbol{\lambda}} \\ \hat{\boldsymbol{\lambda}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{F}^{(1)} \\ \boldsymbol{F}^{(2)} \\ 0 \\ 0 \end{pmatrix}, \tag{A.5}$$

où les vecteurs $\boldsymbol{u}^{(1)}$, $\boldsymbol{u}^{(2)}$, $\tilde{\boldsymbol{\lambda}}$ et $\hat{\boldsymbol{\lambda}}$ collectent les degrés de liberté des discrétisations de $u^{(1)}$, $u^{(2)}$, $\tilde{\lambda}$ et $\hat{\lambda}$ respectivement. Les opérateurs de couplage pour les déplacements $\mathbf{C}^{(s)}$ et les rotations $\mathbf{Z}^{(s)}$ sont des matrices rectangulaires creuses. Comme les matrices de rigidité $\mathbf{K}^{(s)}$, elles sont obtenues par assemblage de matrices locales et traduisent les conditions données dans les équations (A.3) et (A.4).

A.3.2 Décomposition de domaine

Le système couplé A.5 peut être utilisé comme tel pour analyser des modèles non-conformes. Cependant, il est possible de faire appel à des stratégies plus performantes basées sur les méthodes de décomposition de domaine [150]. L'idée générale est de ne pas résoudre un unique système de grande taille mais de résoudre plusieurs systèmes à l'échelle des patches et d'itérer jusqu'à satisfaire les conditions de couplage aux interfaces des patches.

Construction de l'algorithme Nous commençons par écrire l'équilibre de chaque sous-domain (ou chaque patch) soumis à des efforts extérieurs $\boldsymbol{F}^{(s)}$ et d'interface $\mathbf{B}^{(s)T}\boldsymbol{\lambda}$:

$$\boldsymbol{u}^{(s)} = \mathbf{K}^{\dagger(s)} \left(\boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda} \right) + \mathbf{R}^{(s)}\boldsymbol{\alpha}^{(s)}, \tag{A.6}$$

où $\mathbf{K}^{\dagger(s)}$ est un pseudo-inverse associé à la matrice de rigidité du patch, et $\mathbf{R}^{(s)}$ et $\boldsymbol{\alpha}^{(s)}$ correspondent respectivement à une base et à l'amplitude des mouvements de corps rigide du sous-domaine considéré. La matrice $\mathbf{B}^{(s)}$ concatène les deux matrices de couplage $\mathbf{C}^{(s)}$ et $\mathbf{Z}^{(s)}$. De même, les degrés de liberté des efforts d'interfaces, $\tilde{\boldsymbol{\lambda}}$ et $\hat{\boldsymbol{\lambda}}$, sont fusionnés dans le vecteur $\boldsymbol{\lambda}$.

Ensuite, nous introduisons deux conditions supplémentaires. Pour que le sous-domaine soit à l'équilibre, c'est-à-dire, pour que l'équation (A.6) ait du sens, il faut que l'ensemble

des efforts qui s'appliquent sur le patch le permette. C'est la condition d'admissibilité du chargement qui s'exprime ainsi :

$$\mathbf{R}^{(s)T}\left(\boldsymbol{F}^{(s)} - \mathbf{B}^{(s)T}\boldsymbol{\lambda}\right) = 0. \tag{A.7}$$

Finalement, le saut à l'interface des déplacements et des rotations doit être nul : c'est la condition de compatibilité :

$$\sum_{s=1}^{n_s} \mathbf{B}^{(s)}\boldsymbol{u}^{(s)} = 0. \tag{A.8}$$

L'algorithme de décomposition de domaine résout itérativement l'équation (A.8). Les inconnues du problème sont les amplitudes des mouvements de corps rigides des sous-domaines ainsi que les efforts d'interface. Plus précisément, les solutions locales (A.6) sont substituées dans l'équation (A.8) ce qui, combinée avec les équations d'admissibilité (A.7), donne un système où les inconnues sont les d.d.l. des multiplicateurs de Lagrange et les amplitudes des mouvements de corps rigide. Ce système est connu sous le nom *one-level FETI problem* et prend la forme :

$$\begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d} \\ \boldsymbol{e} \end{pmatrix}, \tag{A.9}$$

où l'opérateur $\mathbf{F}$ est le complément de Schur dual :

$$\mathbf{F} = \sum_{s=1}^{n_s} \mathbf{B}^{(s)}\mathbf{K}^{\dagger(s)}\mathbf{B}^{(s)T}. \tag{A.10}$$

Les expressions des autres quantités intervenant dans le système (A.9) ne sont pas rappelés ici et le lecteur intéressé peut se référer aux équations données en section 4.3.1. Comme évoqué, la résolution se fait à l'aide d'un algorithme itératif de sorte à effectuer uniquement des produits matrice-vecteur. Ainsi, le complément de Schur n'a pas besoin d'être assemblé ni inversé (ce qui serait très coûteux). Encore plus important, la multiplication du complément de Schur par un vecteur (correspondant aux efforts d'interface) prend la forme d'une somme de contribution calculée par sous-domaine. Cela en fait une approche naturellement parallélisable. L'utilisation d'un algorithme itératif requiert néanmoins une étape de préconditionnement afin de diminuer le nombre d'itérations et de garantir la robustesse de l'algorithme.

Exemples Différents exemples ont été présentés en section 4.4. Le but était d'étudier l'efficacité du solveur développé. Les résultats obtenus ont montré que le préconditionneur utilisé (voir section 4.4) accélère convenablement la convergence de l'algorithme. Il permet également de bien distribuer les efforts entre les patch en prenant en compte les différences de rigidité. Nous avons étudié l'influence du raffinement du maillage sur la convergence de l'algorithme. Idéalement, peu importe le niveau de raffinement,

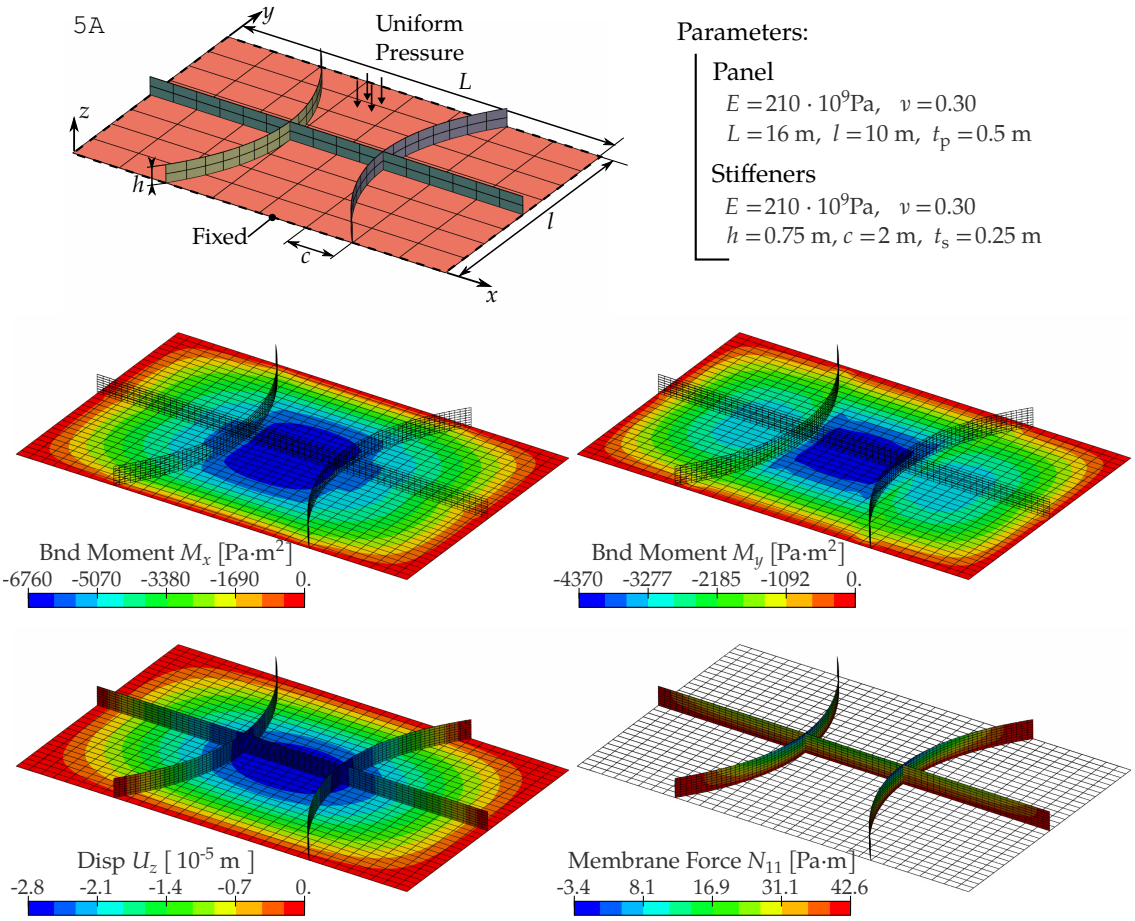


Figure A.6 – Description et résultats pour l'analyse par décomposition de domaine d'un panneau raidi.

l'algorithme converge en un même nombre d'itération. Pour le cas des coques, la scalabilité n'est pas totalement observée mais l'influence du niveau de raffinement reste faible. Aussi, des exemples relativement compliqués comme le panneau raidi avec plusieurs raidisseurs repris en figure A.6, ont pu être traité avec le solveur développé. Un point intéressant est que le critère d'arrêt de l'algorithme se fait sur le saut à l'interface. Il s'arrête uniquement lorsque l'équation de compatibilité est vérifiée (à une tolérance près). Ainsi, même pour des discrétisations très fines et des interfaces compliqués, nous avons un moyen de mesurer le saut à l'interface et donc de vérifier que le couplage est bien pris en compte.

A.4 Approche immergée pour modéliser les structures raidies

On cherche ici à construire un modèle paramétrique de structures complexes généralement obtenues par assemblage des plusieurs pièces. Afin de s'inscrire dans un

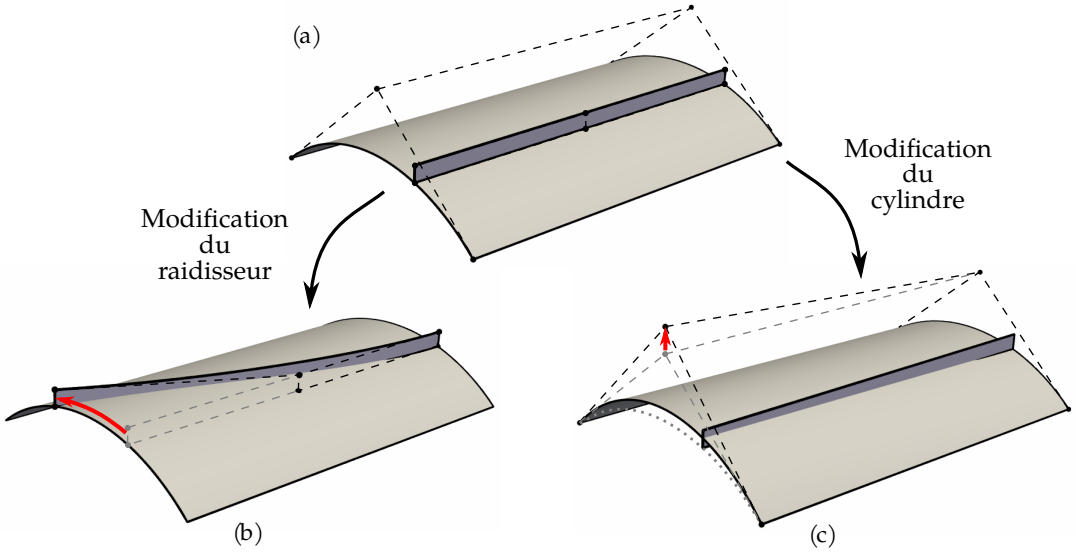


Figure A.7 – Portion de cylindre avec un raidisseur : (a) deux surfaces NURBS permettent de décrire cette géométrie. Cependant le paramétrage de la forme est délicat : (b) modifier la position des points de contrôle du raidisseur ou (c) ceux du cylindre déconnecte les deux pièces.

processus d'optimisation de forme, il est nécessaire d'éviter toutes étapes d'approximation géométrique, de reparamétrisation, et autres règles géométriques sophistiquées. L'objectif est de pouvoir piloter simplement la forme de la structure.

A.4.1 Modèle géométrique versatile

A.4.1.1 Forme paramétrique d'objets complexes

Afin d'observer la difficlté de créer un modèle parametrique d'une structure raidie, considérons un exemple simple d'une portion de cylindre avec un raidisseur comme représenté en figure A.7. Chacun de ces deux objets, la portion de cylindre et le raidisseur, peuvent être construit à l'aide d'une surface NURBS. Maintenant, imaginons que nous voulons modifier la forme de ces deux objets. Pour cela, il est communément adopté de modifier la position des points de contrôle. La difficulté ici est illustrée en figure A.7. Lorsqu'on modifie le raidisseur, le lien à l'interface avec la portion de cylindre est perdu. Aussi, lorsqu'on modifie la portion de cylindre, le raidisseur ne suit pas les changements de géométrie. La raison est qu'il n'existe pas de lien explicite entre la géométrie du raidisseur et celle du cylindre. Dans des cas simples, on peut imaginer des relations entre les points de contrôle des différentes pièces, mais dans le cas général cela n'est pas envisageable.

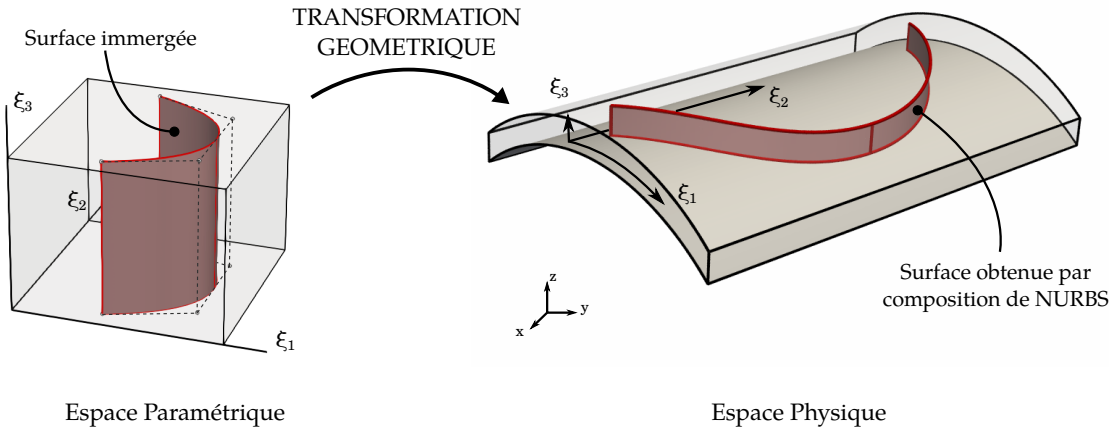


Figure A.8 – Construction d'une structure raidie en utilisant les surfaces immergées. La surface finale décrivant le raidisseur est obtenu pour la composition d'une surface NURBS et d'un volume NURBS. Cette construction garantit que le raidisseur repose parfaitement sur la portion de cylindre.

A.4.1.2 Techniques avancées issues de l'infographie 3D

Pour remédier à la difficulté de paramétrage des structures complexes, nous introduisons le principe de surface immergée. Dans le cas précédent de la portion de cylindre raidie (voir figure A.7), l'idée est de construire le raidisseur par composition d'une surface et d'un volume. Pour cela, une surface NURBS est immergée dans l'espace paramétrique d'un volume NURBS. C'est la composition de ces deux transformations qui va décrire la surface finale du raidisseur. Le volume NURBS est définit de manière à garantir que la surface finale repose sur le cylindre. Pour cela, le volume peut être généré par extrusion de la portion de cylindre. Cette idée est illustrée en figure A.8. En infographie 3D, ces techniques de paramétrisations de forme emboîtées sont connues sous le terme Free-Form Deformation (FFD).

Maintenant, la forme est paramétrée et l'on peut la piloter simplement. Bouger les points de contrôle de la surface immergée va modifier le raidisseur tout en garantissant la contrainte géométrique de contact à l'interface. Aussi, il est possible de modifier la position des points de contrôle associés au volume afin d'optimiser la forme globale de la structure. Les exemples de la section A.4.3 illustrent ces possibilités et montrent la généralisation du principe de construction présenté ici pour le cylindre avec un raidisseur.

A.4.2 Formulation Kirchhoff–Love immergée

Pour répondre à des exigences du point de vue géométrique, nous avons introduit le principe de surfaces immergées et de composition de fonctions NURBS. Désormais, il s'agit d'effectuer l'analyse des structures ainsi construites. Le but est d'appliquer le concept de l'analyse isogéométrique, c'est à dire d'utiliser directement le modèle géo-

métrique pour effectuer l'analyse. Pour cela deux ingrédients sont nécessaires : une formulation coque adaptée au composition de NURBS, et une méthode de couplage pour des paramétrisations non-conformes.

Composition de NURBS L'introduction de la composition de fonctions NURBS ne permet pas d'utiliser directement les formulations coques connues. Une surface $\mathbf{S}$ obtenue par composition de NURBS n'est pas une surface NURBS. Son expression est un peu plus compliquée, et est donnée par :

$$\mathbf{S}(\theta_1, \theta_2) = \mathbf{V}\big(\, \bar{\mathbf{S}}(\theta_1, \theta_2) \,\big) \tag{A.11}$$

où

$$\mathbf{V}(\xi_1, \xi_2, \xi_3) = \sum_{ijk} R_{ijk}(\xi_1, \xi_2, \xi_3)\mathbf{P}_{ijk} \qquad \text{[volume NURBS]} \tag{A.12}$$

$$\bar{\mathbf{S}}(\theta_1, \theta_2) = \sum_{ab} \bar{R}_{ab}(\theta_1, \theta_2)\mathbf{Q}_{ab}. \qquad \text{[surface immergée]} \tag{A.13}$$

Le volume NURBS $\mathbf{V}$ est construit à l'aide de fonctions de bases NURBS à 3 variables R_{ijk} et des points de contrôle $\mathbf{P}_{ijk}$. De manière similaire, la surface immergée $\bar{\mathbf{S}}$ est définit par de fonctions de bases NURBS à 2 variables $\bar{R}_{ab}$ et des points de contrôle $\mathbf{Q}_{ab}$.

Formulation isogéométrique Nous proposons une formulation de type Kirchhoff–Love [94], particulièrement intéressante pour l'optimisation de forme de structures minces (cf. section A.2). Les hypothèses cinématiques conduisent à décrire le champ de déplacement $\boldsymbol{u}^{3D}$ de la coque uniquement à partir du champ de déplacement $\boldsymbol{u}$ de sa surface moyenne. Le champ des déplacements est donné par :

$$\boldsymbol{u}^{3D}(\theta_1, \theta_2, \theta_3) = \boldsymbol{u}(\theta_1, \theta_2) + \theta_3 \left[\Phi \times \boldsymbol{A}_3\right](\theta_1, \theta_2), \tag{A.14}$$

où $\theta_3 \in [\text{-}\frac{t}{2}; \frac{t}{2}]$, t étant l'épaisseur de la coque. Le vecteur unitaire normal à la surface moyenne $\boldsymbol{A}_3$ est définie à l'aide des vecteurs covariants $\boldsymbol{A}_1$ et $\boldsymbol{A}_2$ associés à la surface paramétrée $\mathbf{S}$. Ces vecteurs sont déterminés par les relations suivantes :

$$\boldsymbol{A}_1 = \sum_{k=1,3} \left[\bar{\mathbf{S}}_{,\theta_1} \cdot \mathbf{e}_k\right] \mathbf{V}_{,\xi_k}(\bar{\mathbf{S}}), \quad \boldsymbol{A}_2 = \sum_{k=1,3} \left[\bar{\mathbf{S}}_{,\theta_2} \cdot \mathbf{e}_k\right] \mathbf{V}_{,\xi_k}(\bar{\mathbf{S}}), \quad \boldsymbol{A}_3 = \frac{1}{J}\boldsymbol{A}_1 \times \boldsymbol{A}_2\,, \tag{A.15}$$

où $J = |\boldsymbol{A}_1 \times \boldsymbol{A}_2|$. Le taux de rotation Φ de la normale à la surface est donné par :

$$\Phi = \varphi_1 \boldsymbol{A}_1 + \varphi_2 \boldsymbol{A}_2, \qquad \text{avec} \qquad \varphi_1 = \frac{1}{J}\boldsymbol{u}_{,\theta_2} \cdot \boldsymbol{A}_3 \quad \text{et} \quad \varphi_2 = -\frac{1}{J}\boldsymbol{u}_{,\theta_1} \cdot \boldsymbol{A}_3. \tag{A.16}$$

Il reste à décrire le champ de déplacement de la surface moyenne. L'application de la méthode de Galerkine pose la question de l'espace d'approximation $\mathcal{V}^h$ du déplacement. Le principe isoparamétrique n'est pas applicable pour les compositions de NURBS. Pour construire cet espace d'approximation, deux choix apparaissent envisageables :

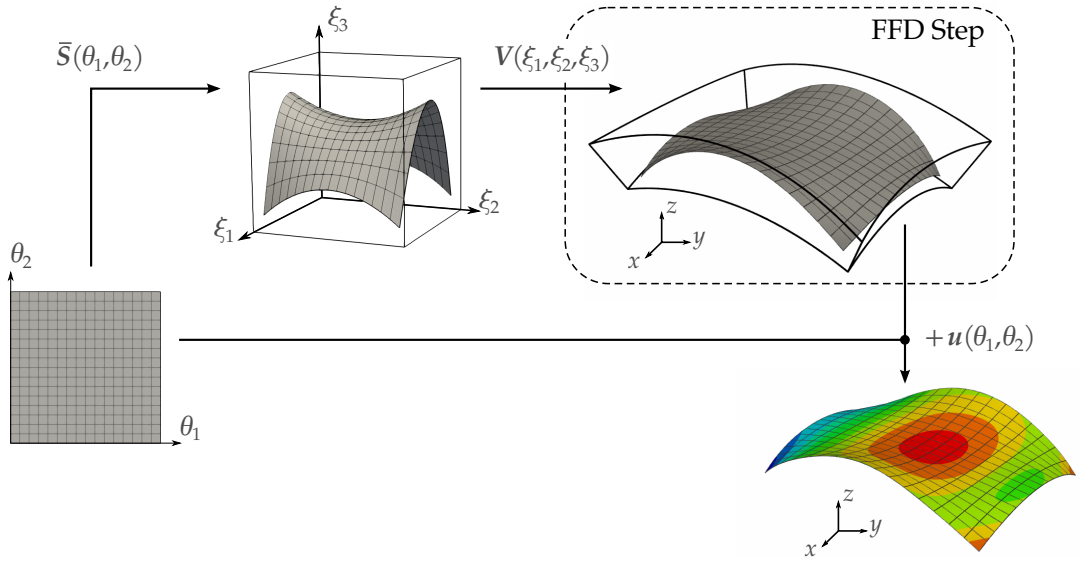


Figure A.9 – Formulation Kirchhoff–Love immergée. Le champ des déplacements est approché par la même paramétrisation que la surface immergée $\bar{\mathbf{S}}$. Cependant cette surface ne décrit pas la géométrie de la coque mais une transformation volumique supplémentaire est appliquée. La configuration déformée est obtenue en additionnant le champ des déplacements à la composition de NURBS.

1. utiliser les fonctions de base associées à la transformation volumique :

$$u^h(\theta_1,\theta_2) = \sum_{ijk} R_{ijk}\big(\bar{\mathbf{S}}(\theta_1,\theta_2)\big)\ \mathbf{U}_{ijk}, \tag{A.17}$$

2. utiliser les fonctions de base associées à la surface immergée :

$$u^h(\theta_1,\theta_2) = \sum_{ab} \bar{R}_{ab}(\theta_1,\theta_2)\,\mathbf{U}_{ab}. \tag{A.18}$$

Le premier choix constitue une approche immergée. Les degrés de liberté sont imposés sur les points de contrôle du volume dans lequel est immergée la surface. Ces approches nécessitent une attention particulière pour ne pas obtenir un système mal-conditionné. Cela est dû aux supports compacts des fonctions de base. Dans l'équation (A.17), il se peut que l'image de la surface immergée $\bar{\mathbf{S}}$ n'intersecte pas le support de certaines fonctions de base R_{ijk}. Ainsi, les degrés de liberté associés à ces fonctions sont inactifs et doivent être retirés du système d'équations. Ici nous adoptons la seconde approche où le déplacement est approché à l'aide de la paramétrisation de la surface immergée. Cette approche a l'avantage de conduire à des systèmes de petites tailles car les fonctions de base sont celles d'une surface (et non d'un volume comme dans le premier cas). Avec ce choix d'approximation des déplacements (A.18) et les quantités définies dans les équations (A.11) à (A.16), il est désormais possible de construire un élément coque de type Kirchhoff–Love où la surface moyenne est décrite par composition de NURBS.

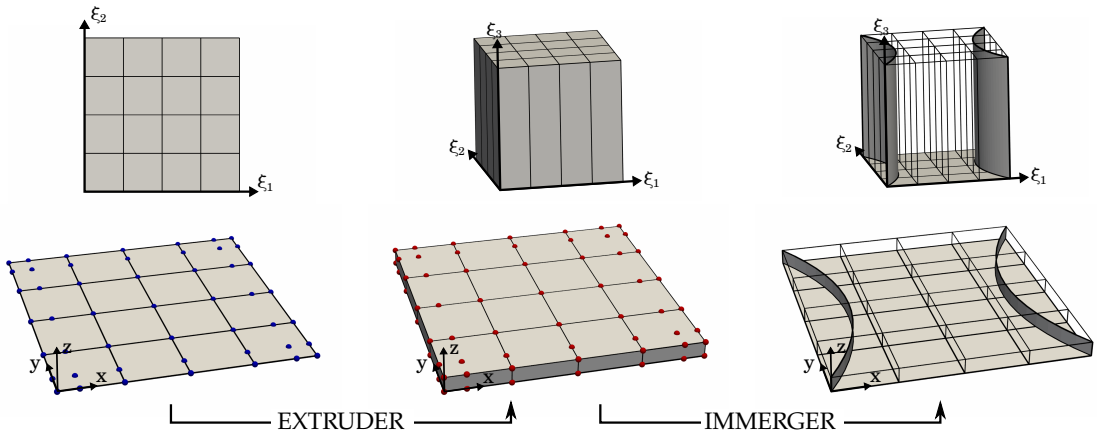


Figure A.10 – Construction de la géométrie du toit raidi : à partir d'une plaque plane, un volume est généré par extrusion puis deux surfaces sont immergées dans l'espace paramétrique (en haut) de ce volume. Les deux raidisseurs dans l'espace physique (en bas) sont alors obtenus par composition.

Les expressions du champ des déformations, des contraintes et la matrice élémentaire sont données en section 5.2. Le lecteur intéressé pourra également se référer à la formulation coque Kirchhoff–Love initiale [94] et remplacer les quantités évoquées par les expressions données ici. Un aperçu global de l'approche est décrit en figure A.9.

A.4.3 Application pour l'optimisation de forme

En combinant l'approche de modélisation géométrique par composition de NURBS, la formulation coque immergée, l'approche Mortar pour le couplage, ainsi que l'algorithme de résolution par décomposition de domaine, nous obtenons un outil numérique d'aide à la conception de structures raidies. Nous reprenons dans cette section deux exemples qui illustrent les capacités de l'approche développée. D'autres exemples peuvent être trouvés dans le coeur de cette thèse et plus spécifiquement dans les chapitres 5 et 6.

A.4.3.1 Optimisation globale d'une structure raidie

Le premier exemple est un toit auquel est fixé deux raidisseurs curvilignes. Initialement, le toit est représenté par une plaque carrée. Ses quatre coins sont fixés et la structure est soumise à une pression uniforme verticale. La construction du modèle est expliquée en figure A.10. Les raidisseurs sont construits par composition de NURBS où la transformation volumique est générée par extrusion de la surface NURBS définissant la plaque carrée.

L'intérêt de cette construction est la possibilité de modifier la forme du toit tout en garantissant que les raidisseurs suivent ces modifications de forme. Durant l'optimisation, les raidisseurs reposent constamment et parfaitement sur la surface représentant le

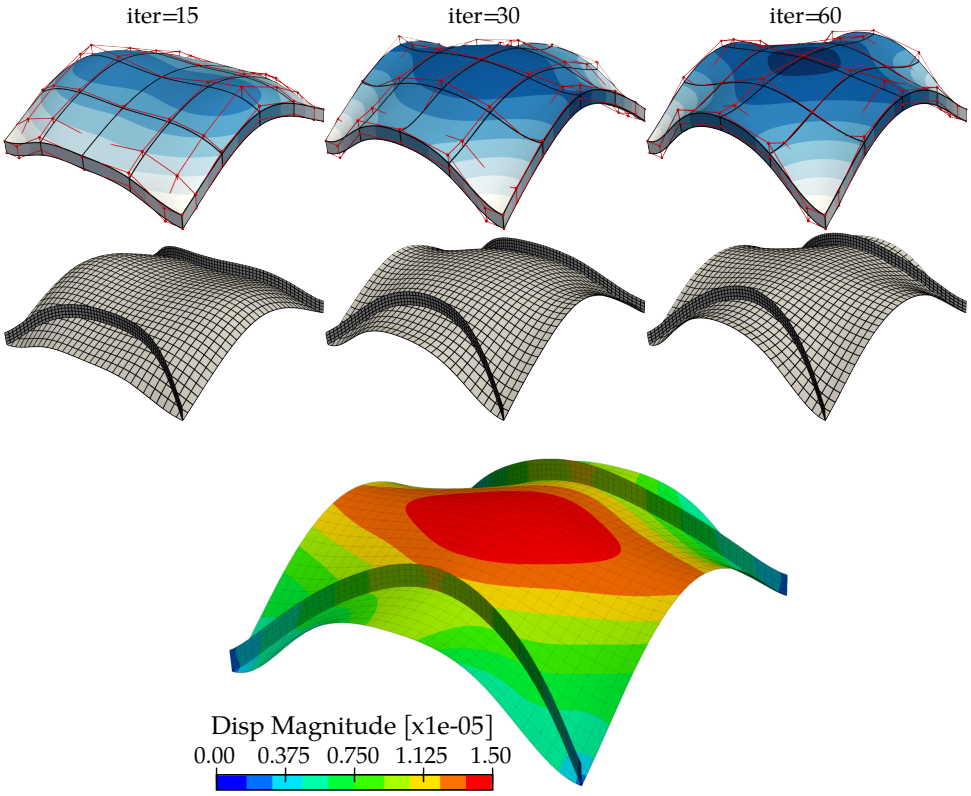


Figure A.11 – Résultats pour le problème du toit muni de deux raidisseurs. A gauche, modification de la forme imposée par le mouvement vertical des points de contrôle du volume (points rouges). Le champ de couleurs quantifie cette variation de forme. A droite, le modèle d'analyse de la géométrie finale.

toit. Pour cela, les variations de forme sont imposées au travers des points de contrôle de la transformation volumique. Les surfaces immergées sont contenues dans le volume et subissent automatiquement la variation de forme globale (voir figure A.11). C'est le concept de *Free-From Deformation* largement utilisé en optimisation aérodynamique par exemple.

Des résultats du problème d'optimisation sont présentés en figure A.11. L'évolution de la forme du toit est donnée pour différentes itérations de la résolution. On note que le modèle d'analyse (maillage fin représenté en figure A.11) est obtenu par raffinement du modèle d'optimisation sur lequel sont définies les variables de *design*. Le niveau de raffinement est choisi de manière à garantir une bonne qualité de résolution. Les fonctions NURBS jouissent de procédures de raffinement simples à mettre en place et peu coûteuses (applications linéaires) [62]. Finalement, le comportement de la géométrie optimale est bien meilleur au regard du problème considéré. La compliance relative de la géométrie finale est de $C_{\text{opt}}/C_0 = 2.217e\text{-}3$ pour les paramètres utilisés ici.

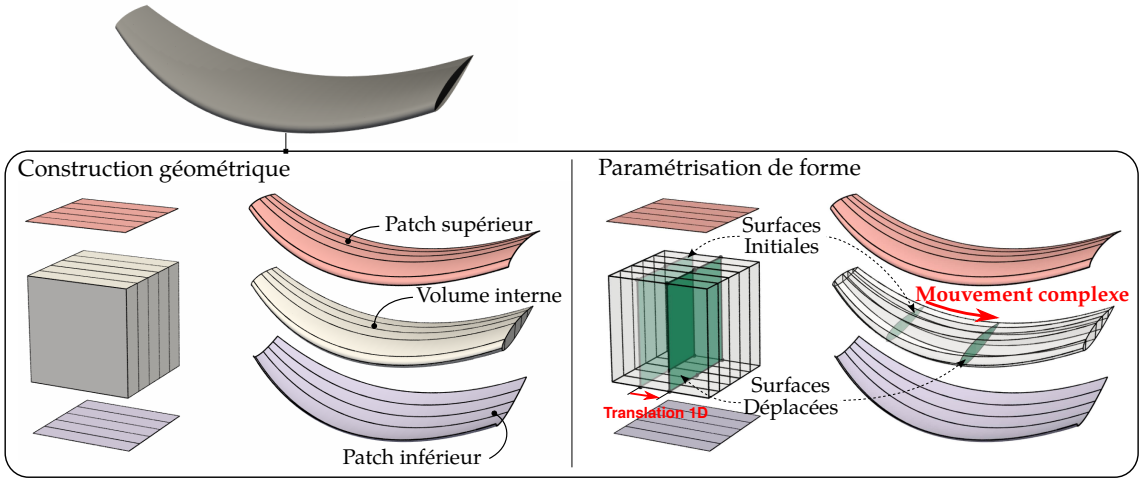


Figure A.12 – Construction et paramétrisation de forme de la structure interne d'une aile. Un volume est généré à partir de deux surfaces représentant la géométrie extérieure de l'aile. Il est alors possible d'immerger et de déplacer des surfaces dans l'espace paramétrique du volume pour construire la structure interne.

A.4.3.2 Optimisation locale de la structure interne d'une aile

Finalement, on présente ici un exemple d'une aile simplifiée où le but est de concevoir sa structure interne (note : le cas d'une aile plus réaliste est présentée dans le chapitre 6). La géométrie globale est fixée par les peaux extérieures. Du point de vue de la géométrie, la surface extérieure de l'aile impose une contrainte géométrique à la structure interne : celle-ci doit être contenue à l'intérieur du domaine délimité par la surface extérieure. La construction illustrée en figure A.12 permet de naturellement prendre en compte cette contrainte. L'idée est de définir un volume remplissant l'intérieur de l'aile. Ce volume est généré par interpolation à partir de deux surfaces qui représentent les peaux supérieure et inférieure. Finalement, les éléments composants la structure interne sont créés en immergeant des surfaces dans l'espace paramétrique de ce volume. La forme de la structure interne peut facilement être modifiée en déplaçant les points de contrôle associés aux surfaces immergées. Par exemple, sur la figure A.12, une translation unidirectionnelle de la surface immergée va déplacer la surface finale le long de l'aile. Les variations de section de l'aile ainsi que sa courbure sont automatiquement prises en compte.

La figure A.13 illustre des résultats d'optimisation pour le problème de conception de la structure interne de l'aile. Les deux cas considérés consistent à disposer 4 et 8 raidisseurs le long de l'aile. Initialement ces raidisseurs sont espacés de manière uniforme. En suivant la procédure décrite en figure A.12, une variable de *design* est associée à chacun des raidisseurs afin de paramétrer leur position le long de l'aile. L'aile est encastrée en son bord et est soumise à une pression uniforme sur sa surface du haut. Les résultats du problème d'optimisation donnent les positions optimales des raidisseurs garantis-

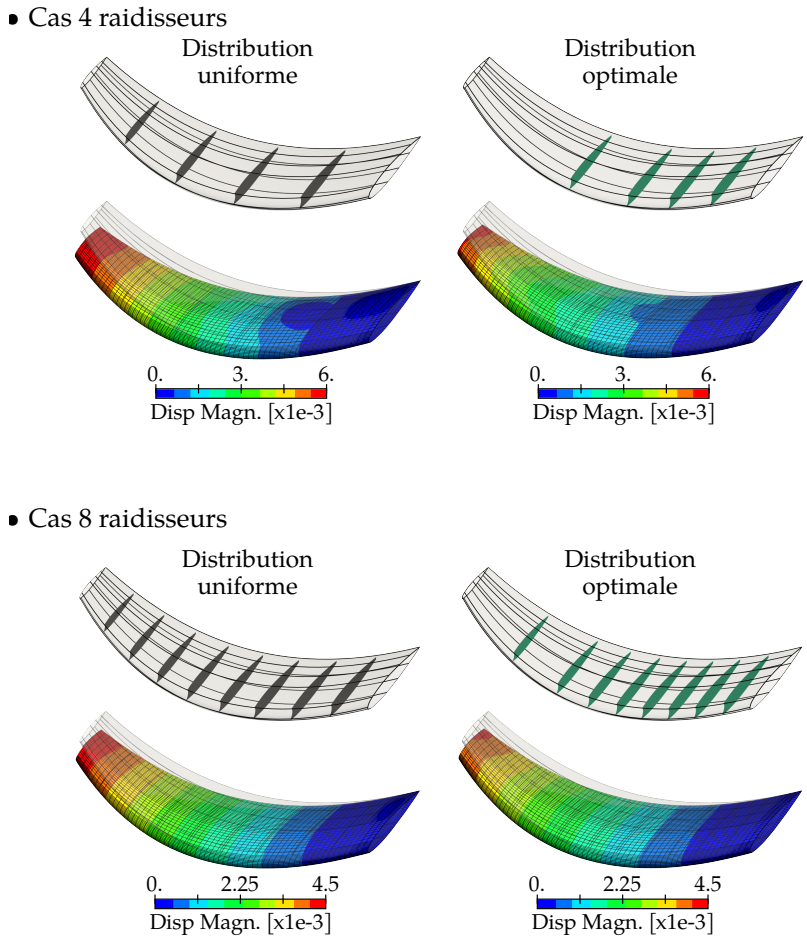


Figure A.13 – Optimisation de la structure interne d'une aile : distribution optimale de 4 et de 8 raidisseurs la long de l'aile. Un meilleur positionnement des raidisseurs rigidifie l'ensemble de la structure.

sant une plus grande rigidité de la structure globale. La compliance finale a diminué de 15.1% pour le cas à 4 raidisseurs, et de 7.69% pour le cas à 8 raidisseurs.

A.5 Conclusion

Ce travail permet de justifier de l'intérêt de l'analyse isogéométrique pour l'optimisation paramétrique de forme. Le gain apporté par l'utilisation d'un unique modèle pour représenter fidèlement la géométrie et effectuer l'analyse avec précision est réellement bénéfique. La procédure d'optimisation se trouve allégée car la communication entre le modèle d'optimisation sur lequel sont imposées les modifications de forme et le modèle d'analyse est naturelle. L'intérêt pour le concepteur est réel car ce lien étroit permet indéniablement d'accélérer la phase de prototypage par exemple. Cela peut être un outil intéressant dans un contexte d'optimisation multidisciplinaire. La communication

entre les différentes disciplines est facilité lorsque le modèle utilisé par chacune d'elles est identique.

L'approche principale développée dans ce travail est la stratégie immergée mise en place pour pouvoir traiter le cas des structures raidies. Cette approche permet d'étendre le cadre de l'optimisation isogéométrique à des cas plus compliqués où plusieurs patches sont utilisés. La composition de NURBS donne accès à des paramètres globaux associés à la forme globale de la structure, mais également des paramètres locaux permettant de modifier des pièces spécifiques tout en garantissant automatiquement le respect de contraintes géométriques imposées par les pièces environnantes. L'utilisation de méthodes avancées pour traiter des discrétisations non-conformes ont pu conduire à une approche finale robuste et générale. Le champ d'application est large et n'est pas cantonné à la conception des structures raidies.

Bibliography

[1] H. Goussé. *A350-XWB interior view*. 2012. URL: https://cdt31.media.tourinsoft.eu/upload/A350-XWB-INTERIOR-VIEW.jpg.

[2] D. Parker Brown. *Airbus Beluga XL Becoming a Reality*. 2017. URL: https://www.airlinereporter.com/2017/01/photos-airbus-beluga-xl-becoming-reality/.

[3] D. Wallsworth. *Airbus A350 wing box section*. 2018. URL: https://pbs.twimg.com/media/DmaosJOX0AEYnRu.jpg:large.

[4] H.-T. Pedro, G. Reich, M. Kobayashi, and R. Kolonay. "On a Cellular Division Method for Topology Optimization". In: *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. May. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2009.

[5] T. D. Dang, R. K. Kapania, W. C. H. Slemp, M. Bhatia, and S. P. Gurav. "Optimization and Postbuckling Analysis of Curvilinear-Stiffened Panels Under Multiple-Load Cases". In: *Journal of Aircraft* 47.5 (2010), pp. 1656–1671.

[6] M. Bhatia, R. K. Kapania, and D. Evans. "Comparative Study on Optimal Stiffener Placement for Curvilinearly Stiffened Panels". In: *Journal of Aircraft* 48.1 (2011), pp. 77–91.

[7] S. B. Mulani, W. C. H. Slemp, and R. K. Kapania. "EBF3PanelOpt: An optimization framework for curvilinear blade-stiffened panels". In: *Thin-Walled Structures* 63 (2013), pp. 13–26.

[8] D. Locatelli, A. Yeilaghi Tamijani, S. B. Mulani, Q. Liu, and R. K. Kapania. "Multidisciplinary Optimization of Supersonic Wing Structures Using Curvilinear Spars and Ribs (SpaRibs)". In: *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2013, pp. 1–22.

[9] C. V. Jutte, B. Stanford, C. D. Wieseman, and J. B. Moore. "Aeroelastic Tailoring of the NASA Common Research Model via Novel Material and Structural Configurations". In: *52nd Aerospace Sciences Meeting*. January. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2014, pp. 1–20.

[10] D. Locatelli, S. B. Mulani, and R. K. Kapania. "Parameterization of Curvilinear Spars and Ribs for Optimum Wing Structural Design". In: *Journal of Aircraft* 51.2 (2014), pp. 532–546.

[11] P. Hao, B. Wang, K. Tian, G. Li, and X. Zhang. "Optimization of Curvilinearly Stiffened Panels with Single Cutout Concerning the Collapse Load". In: *International Journal of Structural Stability and Dynamics* 16.07 (2016), p. 1550036.

[12] W. Zhao and R. K. Kapania. "Buckling analysis of unitized curvilinearly stiffened composite panels". In: *Composite Structures* 135 (2016), pp. 365–382.

[13] A. Dubois, C. Farhat, A. H. Abukhwejah, and H. M. Shageer. "Parameterization Framework for the MDAO of Wing Structural Layouts". In: *AIAA Journal* 56.4 (2018), pp. 1627–1638.

[14] D. Wang, M. M. Abdalla, Z.-P. Wang, and Z. Su. "Streamline stiffener path optimization (SSPO) for embedded stiffener layout design of non-uniform curved grid-stiffened composite (NCGC) structures". In: *Computer Methods in Applied Mechanics and Engineering* 344 (2019), pp. 1021–1050.

[15] M. Renard. *STELIA Aerospace imprime en 3D un panneau de fuselage*. 2018. URL: https://www.3dnatives.com/stelia-aerospace-21022018/.

[16] D. Havens, S. Shiyekar, A. Norris, R. K. Bird, R. K. Kapania, and O. Robert. *Design, Optimization, and Evaluation of Integrally-Stiffened Al-2139 Panel with Curved Stiffeners*. Tech. rep. 217308. National Aeronautics and Space Administration, 2011.

[17] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer Methods in Applied Mechanics and Engineering* 194.39-41 (2005), pp. 4135–4195.

[18] T. Hirschler, R. Bouclier, A. Duval, T. Elguedj, and J. Morlier. "Isogeometric sizing and shape optimization of thin structures with a solid-shell approach". In: *Structural and Multidisciplinary Optimization* 59.3 (2019), pp. 767–785.

[19] 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". In: *Computer Methods in Applied Mechanics and Engineering* 349 (2019), pp. 774–797.

[20] 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". In: *Computer Methods in Applied Mechanics and Engineering* 357 (2019), p. 112578.

[31] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The Finite Element Method: Its Basis and Fundamentals, Sixth Edition*. 6th ed. Butterworth-Heinemann, 2005.

[32] T. J. R. Hughes. *The finite element method : linear static and dynamic finite elment analysis*. English. Mineola, NY : Dover Publications, 2000.

[33] M. A. Crisfield. *Non-Linear Finite Element Analysis of Solids and Structures: Advanced Topics*. 1st. New York, NY, USA: John Wiley & Sons, Inc., 1997.

[34] T. Belytschko, W. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. English. John Wiley & Sons, Ltd, 2000.

[35] SIMULIA. *SIMULIA Tosca Structure*. Brochure. 2017. URL: https://www.3ds.com/fileadmin/PRODUCTS/SIMULIA/PDF/brochures/simulia-tosca-structure-brochure.pdf.

[36] K.-U. Bletzinger. "Shape Optimization". In: *Encyclopedia of Computational Mechanics Second Edition*. American Cancer Society, 2017, pp. 1–42.

[37] J.-H. Zhu, W.-H. Zhang, and L. Xia. "Topology Optimization in Aircraft and Aerospace Structures Design". In: *Archives of Computational Methods in Engineering* 23.4 (2016), pp. 595–622.

[38] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund. "Giga-voxel computational morphogenesis for structural design". In: *Nature* 550.7674 (2017), pp. 84–86.

[39] C. D. Nha, Y. Xie, and G. Steven. "An evolutionary structural optimization method for sizing problems with discrete design variables". In: *Computers & Structures* 68.4 (1998), pp. 419–431.

[40] L. Simões and J. Negrão. "Sizing and geometry optimization of cable-stayed bridges". In: *Computers & Structures* 52.2 (1994), pp. 309–321.

[41] P. Vannucci. "Designing the elastic properties of laminates as an optimisation problem: a unified approach based on polar tensor invariants". In: *Structural and Multidisciplinary Optimization* 31.5 (2006), pp. 378–387.

[42] L. Krog, A. Tucker, G. Rollema, et al. "Application of topology, sizing and shape optimization methods to optimal design of aircraft components". In: *Proc. 3rd Altair UK HyperWorks Users Conference*. 2002.

[43] G. Schuhmacher, M. Stettner, R. Zotemantel, O. O'Leary, and M. Wagner. "Optimization Assisted Structural Design of a New Military Transport Aircraft". In: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. September. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2004, pp. 1–9.

[44] L. Krog, A. Tucker, M. Kemp, and R. Boyd. "Topology Optimisation of Aircraft Wing Box Ribs". In: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Vol. 3. September. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2004, pp. 2020–2030.

[45] P. Cervellera, M. Zhou, and U. Schramm. "Optimization driven design of shell structures under stiffness, strength and stability requirements". In: *6th World Congresses of Structural and Multidisciplinarity Optimization*. 2005.

[46] A. Hall, T. Mayer, I. Wuggetzer, and P. Childs. "Future aircraft cabins and design thinking: optimisation vs. win-win scenarios". In: *Propulsion and Power Research* 2.2 (2013), pp. 85–95.

[47] L. Zhu, N. Li, and P. Childs. "Light-weighting in aerospace component and system design". In: *Propulsion and Power Research* 7.2 (2018), pp. 103–119.

[48] S. Coniglio, C. Gogu, R. Amargier, and J. Morlier. "Engine Pylon Topology Optimization Framework Based on Performance and Stress Criteria". In: *AIAA Journal* (2019), pp. 1–13.

[49] P. A. Suárez Espinoza. "Design Process for the Shape Optimization of Pressurized Bulkheads as Components of Aircraft Structures". PhD thesis. Technische Universität München, 2015.

[50] J. A. Samareh. "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization". In: *AIAA Journal* 39.5 (2001), pp. 877–884.

[51] M. H. Imam. "Three-dimensional shape optimization". In: *International Journal for Numerical Methods in Engineering* 18.5 (1982), pp. 661–673.

[52] V. Braibant and C. Fleury. "Shape optimal design using B-splines". In: *Computer Methods in Applied Mechanics and Engineering* 44.3 (1984), pp. 247–267.

[53] R. T. Haftka and R. V. Grandhi. "Structural shape optimization–A survey". In: *Computer Methods in Applied Mechanics and Engineering* 57.1 (1986), pp. 91–106.

[54] K.-U. Bletzinger, S. Kimmich, and E. Ramm. "Efficient modeling in shape optimal design". In: *Computing Systems in Engineering* 2.5-6 (1991), pp. 483–495.

[55] N. Olhoff, J. Rasmussen, and E. Lund. "A Method of "Exact" Numerical Differentiation for Error Elimination in Finite-Element-Based Semi-Analytical Shape Sensitivity Analyses*". In: *Mechanics of Structures and Machines* 21.1 (1993), pp. 1–66.

[56] Y.-L. Hsu. "A review of structural shape optimization". In: *Computers in Industry* 25.1 (1994), pp. 3–13.

[57] K.-U. Bletzinger, R. Reitinger, S. Kimmich, and E. Ramm. "Shape Optimization with Program CARAT". In: *Software Systems for Structural Optimization*. Vol. 110. Basel: Birkhäuser Basel, 1993, pp. 97–124.

[58] C. Le, T. Bruns, and D. Tortorelli. "A gradient-based, parameter-free approach to shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 200.9-12 (2011), pp. 985–996.

[59] M. Hojjat, E. Stavropoulou, and K.-U. Bletzinger. "The Vertex Morphing method for node-based shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 268 (2014), pp. 494–513.

[60] M. Shimoda and Y. Liu. "A non-parametric free-form optimization method for shell structures". In: *Structural and Multidisciplinary Optimization* 50.3 (2014), pp. 409–423.

[61] F.-J. Ertl, G. Dhondt, and K.-U. Bletzinger. "Vertex assigned morphing for parameter free shape optimization of 3-dimensional solid structures". In: *Computer Methods in Applied Mechanics and Engineering* 353 (2019), pp. 86–106.

[62] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. 1st. Wiley Publishing, 2009.

[63] S. Hartmann, D. Benson, and A. Nagy. "Isogeometric Analysis with LS-DYNA". In: *Journal of Physics: Conference Series* 734 (2016), p. 032125.

[64] M. Occelli, T. Elguedj, S. Bouabdallah, and L. Morançay. "LR B-Splines implementation in the Altair RadiossTM solver for explicit dynamics IsoGeometric Analysis". In: *Advances in Engineering Software* 131.October 2018 (2019), pp. 166–185.

[65] B. Marussig and T. J. R. Hughes. "A Review of Trimming in Isogeometric Analysis: Challenges, Data Exchange and Simulation Aspects". In: *Archives of Computational Methods in Engineering* 25.4 (2018), pp. 1059–1127.

[66] K.-U. Bletzinger and E. Ramm. "Computational form finding and optimization". In: *Shell Structures for Architecture*. Ed. by S. Adriaenssens, P. Block, D. Veenendaal, and C. Williams. London: Routledge, 2014. Chap. 5, pp. 45–55.

[67] W. A. Wall, M. A. Frenzel, and C. Cyron. "Isogeometric structural shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 197.33-40 (2008), pp. 2976–2988.

[68] X. Qian. "Full analytical sensitivities in NURBS based isogeometric shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 199.29-32 (2010), pp. 2059–2071.

[69] A. P. Nagy, M. M. Abdalla, and Z. Gürdal. "Isogeometric sizing and shape optimisation of beam structures". In: *Computer Methods in Applied Mechanics and Engineering* 199.17-20 (2010), pp. 1216–1230.

[70] A. P. Nagy, M. M. Abdalla, and Z. Gürdal. "Isogeometric design of elastic arches for maximum fundamental frequency". In: *Structural and Multidisciplinary Optimization* 43.1 (2011), pp. 135–149.

[71] A. P. Nagy, S. T. IJsselmuiden, and M. M. Abdalla. "Isogeometric design of anisotropic shells: Optimal form and material distribution". In: *Computer Methods in Applied Mechanics and Engineering* 264 (2013), pp. 145–162.

[72] J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. "Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting". In: *Computer Methods in Applied Mechanics and Engineering* 274 (2014), pp. 148–167.

[73] A. Taheri and B. Hassani. "Simultaneous isogeometrical shape and material design of functionally graded structures for optimal eigenfrequencies". In: *Computer Methods in Applied Mechanics and Engineering* 277 (2014), pp. 46–80.

[74] W. Wang, P. M. Clausen, and K.-U. Bletzinger. "Improved semi-analytical sensitivity analysis using a secant stiffness matrix for geometric nonlinear shape optimization". In: *Computers & Structures* 146 (2015), pp. 143–151.

[75] D. Fußeder, B. Simeon, and A.-V. Vuong. "Fundamental aspects of shape optimization in the context of isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 286 (2015), pp. 313–331.

[76] P. Kang and S. K. Youn. "Isogeometric shape optimization of trimmed shell structures". In: *Structural and Multidisciplinary Optimization* 53.4 (2016), pp. 825–845.

[77] H. Lian, P. Kerfriden, and S. Bordas. "Shape optimization directly from CAD: An isogeometric boundary element approach using T-splines". In: *Computer Methods in Applied Mechanics and Engineering* 317 (2017), pp. 1–41.

[78] Z.-P. Wang, M. Abdalla, and S. Turteltaub. “Normalization approaches for the descent search direction in isogeometric shape optimization”. In: *Computer-Aided Design* 82.June (2017), pp. 68–78.

[79] M.-J. Choi and S. Cho. “Constrained isogeometric design optimization of lattice structures on curved surfaces: computation of design velocity field”. In: *Structural and Multidisciplinary Optimization* 58.1 (2018), pp. 17–34.

[80] Z. Lei, F. Gillot, and L. Jezequel. “Shape Optimization for Natural Frequency with Isogeometric Kirchhoff-Love Shell and Sensitivity Mapping”. In: *Mathematical Problems in Engineering* 2018 (2018), pp. 1–11.

[81] C. Ding, X. Cui, G. Huang, G. Li, K. Tamma, and Y. Cai. “A gradient-based shape optimization scheme via isogeometric exact reanalysis”. In: *Engineering Computations* (2018), EC–08–2017–0292.

[82] O. Weeger, B. Narayanan, and M. L. Dunn. “Isogeometric shape optimization of nonlinear, curved 3D beams and beam structures”. In: *Computer Methods in Applied Mechanics and Engineering* (2018).

[83] Z.-P. Wang, S. Turteltaub, and M. Abdalla. “Shape optimization and optimal control for transient heat conduction problems using an isogeometric approach”. In: *Computers & Structures* 185 (2017), pp. 59–74.

[84] D. M. Nguyen, A. Evgrafov, and J. Gravesen. “ISOGEOMETRIC SHAPE OPTIMIZATION FOR ELECTROMAGNETIC SCATTERING PROBLEMS”. In: *Progress In Electromagnetics Research B* 45.January 2016 (2012), pp. 117–146.

[85] N. Dang Manh, A. Evgrafov, J. Gravesen, and D. Lahaye. “Iso-geometric shape optimization of magnetic density separators”. In: *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering* 33.4 (2014), pp. 1416–1433.

[86] B.-U. Park, Y.-D. Seo, O. Sigmund, and S.-K. Youn. “Shape optimization of the stokes flow problem based on isogeometric analysis”. In: *Structural and Multidisciplinary Optimization* 48.5 (2013), pp. 965–977.

[87] S. D. Daxini and J. M. Prajapati. “Parametric shape optimization techniques based on Meshless methods: A review”. In: *Structural and Multidisciplinary Optimization* 56.5 (2017), pp. 1197–1214.

[88] Y. Wang, Z. Wang, Z. Xia, and L. Hien Poh. “Structural Design Optimization Using Isogeometric Analysis: A Comprehensive Review”. In: *Computer Modeling in Engineering & Sciences* 117.3 (2018), pp. 455–507.

[89] L. Piegl and W. Tiller. *The NURBS Book*. 2nd ed. Berlin, Heidelberg: Springer-Verlag, 1997.

[90] W. J. Gordon and R. F. Riesenfeld. “B-SPLINE CURVES AND SURFACES”. In: *Computer Aided Geometric Design*. Elsevier, 1974, pp. 95–126.

[91] E. Cohen, R. F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. Natick, MA, USA: A. K. Peters, Ltd., 2001.

[92] G. Farin. *Curves and Surfaces for CAGD: A Practical Guide*. 5th. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[93] B.-G. Lee and Y. Park. “Degree elevation of B-spine curves and its matrix form”. In: *Journal of the Korean Society for Industrial and Applied Mathematics* 4.2 (2000), pp. 1–9.

[94] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. “Isogeometric shell analysis with Kirchhoff–Love elements”. In: *Computer Methods in Applied Mechanics and Engineering* 198.49-52 (2009), pp. 3902–3914.

[95] R. Echter, B. Oesterle, and M. Bischoff. “A hierarchic family of isogeometric shell finite elements”. In: *Computer Methods in Applied Mechanics and Engineering* 254 (2013), pp. 170–180.

[96] R. Bouclier, T. Elguedj, and A. Combescure. "An isogeometric locking-free NURBS-based solid-shell element for geometrically nonlinear analysis". In: *International Journal for Numerical Methods in Engineering* 101.10 (2015), pp. 774–808.

[97] A. Seitz, P. Farah, J. Kremheller, B. I. Wohlmuth, W. A. Wall, and A. Popp. "Isogeometric dual mortar methods for computational contact mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 301 (2016), pp. 259–280.

[98] D. Kamensky, M.-C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. "Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines". In: *Computer Methods in Applied Mechanics and Engineering* 314 (2017), pp. 408–472.

[99] A. Apostolatos, G. De Nayer, K.-U. Bletzinger, M. Breuer, and R. Wüchner. "Systematic evaluation of the interface description for fluid–structure interaction simulations using the isogeometric mortar-based mapping". In: *Journal of Fluids and Structures* 86 (2019), pp. 368–399.

[100] T. Elguedj, Y. Bazilevs, V. M. Calo, and T. J. R. Hughes. "B-bar and F-bar projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements". In: *Computer Methods in Applied Mechanics and Engineering* 197.33-40 (2008), pp. 2732–2762.

[101] R. Bouclier, T. Elguedj, and A. Combescure. "Development of a mixed displacement-stress formulation for the analysis of elastoplastic structures under small strains: Application to a locking-free, NURBS-based solid-shell element". In: *Computer Methods in Applied Mechanics and Engineering* 295 (2015), pp. 543–561.

[102] M. Ambati, J. Kiendl, and L. De Lorenzis. "Isogeometric Kirchhoff–Love shell formulation for elasto-plasticity". In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 320–339.

[103] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, and G. Scovazzi. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows". In: *Computer Methods in Applied Mechanics and Engineering* 197.1-4 (2007), pp. 173–201.

[104] I. Akkerman, Y. Bazilevs, C. Kees, and M. Farthing. "Isogeometric analysis of free-surface flow". In: *Journal of Computational Physics* 230.11 (2011), pp. 4137–4152.

[105] A. Reali. "An Isogeometric Analysis approach for the study of structural vibrations". In: *Journal of Earthquake Engineering* 10.spec01 (2006), pp. 1–30.

[106] S. Shojaee, E. Izadpanah, N. Valizadeh, and J. Kiendl. "Free vibration analysis of thin plates by using a NURBS-based isogeometric approach". In: *Finite Elements in Analysis and Design* 61 (2012), pp. 23–34.

[107] S. Morganti, F. Auricchio, D. J. Benson, F. I. Gambarin, S. Hartmann, T. J. R. Hughes, and A. Reali. "Patient-specific isogeometric structural analysis of aortic valve closure". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 508–520.

[108] S. Lipton, J. Evans, Y. Bazilevs, T. Elguedj, and T. Hughes. "Robustness of isogeometric structural discretizations under severe mesh distortion". In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 357–373.

[109] Y. Bazilevs, L. Beirão da Veiga, J. Cottrell, T. Hughes, and G. Sangalli. "Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes". In: *Mathematical Models and Methods in Applied Sciences* 16.07 (2006), pp. 1031–1090.

[110] M. Firl. "Optimal shape design of shell structures". PhD thesis. Technische Universität München, 2010.

[111] R. T. Haftka and Z. Gürdal. *Elements of Structural Optimization*. Springer Netherlands, 1992.

[112] Z. Gürdal, R. T. Haftka, and P. Hajela. *Design and optimization of laminated composite materials*. John Wiley & Sons, 1999.

[113] G. Allaire. *Shape Optimization by the Homogenization Method*. Springer New York, 2002.

[114] M. P. Bendsøe and O. Sigmund. *Topology Optimization*. Springer Berlin Heidelberg, 2004.

[115] J. Nocedal and S. J. Wright. *Numerical Optimization*. second. New York, NY, USA: Springer, 2006.

[116] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 2001–. URL: `http://www.scipy.org/`.

[117] A. P. Nagy. "Isogeometric design optimisation". PhD thesis. Delft University of Technology, 2011.

[118] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.

[119] S. G. Johnson. *The NLopt nonlinear-optimization package*. 2014–. URL: `http://github.com/stevengj/nlopt`.

[120] F. van Keulen, R. Haftka, and N. Kim. "Review of options for structural design sensitivity analysis. Part 1: Linear systems". In: *Computer Methods in Applied Mechanics and Engineering* 194.30-33 (2005), pp. 3213–3243.

[121] K.-U. Bletzinger, M. Firl, and F. Daoud. "Approximation of derivatives in semi-analytical structural optimization". In: *Computers & Structures* 86.13-14 (2008), pp. 1404–1416.

[122] K. Bandara, T. Rüberg, and F. Cirak. "Shape optimisation with multiresolution subdivision surfaces and immersed finite elements". In: *Computer Methods in Applied Mechanics and Engineering* 300 (2016), pp. 510–539.

[123] K. Bandara and F. Cirak. "Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces". In: *Journal of Physics: Conference Series* 734.3 (2016), p. 032142. arXiv: `1605.06288`.

[124] J. N. Lyness and C. B. Moler. "Numerical Differentiation of Analytic Functions". In: *SIAM Journal on Numerical Analysis* 4.2 (1967), pp. 202–210.

[125] L. F. R. Espath, R. V. Linn, and A. M. Awruch. "Shape optimization of shell structures based on NURBS description using automatic differentiation". In: *International Journal for Numerical Methods in Engineering* 88.7 (2011), pp. 613–636.

[126] D. L. Whitfield, J. C. Newman, and W. K. Anderson. "Step-Size Independent Approach for Multidisciplinary Sensitivity Analysis". In: *Journal of Aircraft* 40.3 (2003), pp. 566–573.

[127] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. "The complex-step derivative approximation". In: *ACM Transactions on Mathematical Software* 29.3 (2003), pp. 245–262.

[128] R. Bouclier, T. Elguedj, and A. Combescure. "Efficient isogeometric NURBS-based solid-shell elements: Mixed formulation and B-Method". In: *Computer Methods in Applied Mechanics and Engineering* 267 (2013), pp. 86–110.

[129] D. J. Benson, Y. Bazilevs, M. C. Hsu, and T. J. R. Hughes. "Isogeometric shell analysis: The Reissner-Mindlin shell". In: *Computer Methods in Applied Mechanics and Engineering* 199.5-8 (2010), pp. 276–289.

[130] T. Belytschko, H. Stolarski, W. K. Liu, N. Carpenter, and J. S. Ong. "Stress projection for membrane and shear locking in shell finite elements". In: *Computer Methods in Applied Mechanics and Engineering* 51.1-3 (1985), pp. 221–258.

[131] J. F. Caseiro, R. A. F. Valente, A. Reali, J. Kiendl, F. Auricchio, and R. J. Alves De Sousa. "On the Assumed Natural Strain method to alleviate locking in solid-shell NURBS-based finite elements". In: *Computational Mechanics* 53.6 (2014), pp. 1341–1353.

[132] R. Bouclier, T. Elguedj, and A. Combescure. "An isogeometric locking-free NURBS-based solid-shell element for geometrically nonlinear analysis". In: *International Journal for Numerical Methods in Engineering* 101.10 (2015), pp. 774–808.

[133] R. P. R. Cardoso and J. M. A. Cesar De Sa. "Blending moving least squares techniques with NURBS basis functions for nonlinear isogeometric analysis". In: *Computational Mechanics* 53.6 (2014), pp. 1327–1340.

[134] J. F. Caseiro, R. A. F. Valente, A. Reali, J. Kiendl, F. Auricchio, and R. J. Alves de Sousa. "Assumed natural strain NURBS-based solid-shell element for the analysis of large deformation elasto-plastic thin-shell structures". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 861–880.

[135] J. Kiendl. "Isogeometric Analysis and Shape Optimal Design of Shell Structures". PhD thesis. Technische Universität München, 2011.

[136] R. Echter. "Isogeometric Analysis of Shells". PhD thesis. Institut für Baustatik und Baudynamik, Universität Stuttgart, 2013.

[137] C. S. Ding, X. Y. Cui, and G. Y. Li. "Accurate analysis and thickness optimization of tailor rolled blanks based on isogeometric analysis". In: *Structural and Multidisciplinary Optimization* 54.4 (2016), pp. 871–887.

[138] W. Dornisch, S. Klinkel, and B. Simeon. "Isogeometric Reissner–Mindlin shell analysis with exactly calculated director vectors". In: *Computer Methods in Applied Mechanics and Engineering* 253 (2013), pp. 491–504.

[139] G. Kirchhoff. "Über das Gleichgewicht und die Bewegung einer elastischen Scheibe." ger. In: *Journal für die reine und angewandte Mathematik* 40 (1850), pp. 51–88. URL: http://eudml.org/doc/147439.

[140] A. E. H. Love. "XVI. The small free vibrations and deformation of a thin elastic shell". In: *Philosophical Transactions of the Royal Society of London.(A.)* 179 (1888), pp. 491–546.

[141] M. Bischoff, E. Ramm, and J. Irslinger. "Models and Finite Elements for Thin-Walled Structures". In: *Encyclopedia of Computational Mechanics Second Edition*. 1859. Chichester, UK: John Wiley & Sons, Ltd, 2017, pp. 1–86. URL: http://doi.wiley.com/10.1002/9781119176817.ecm2026.

[142] M. Balesdent, N. Bérend, P. Dépincé, and A. Chriette. "A survey of multidisciplinary design optimization methods in launch vehicle design". In: *Structural and Multidisciplinary Optimization* 45.5 (2012), pp. 619–642.

[143] J. R. R. A. Martins and A. B. Lambe. "Multidisciplinary Design Optimization: A Survey of Architectures". In: *AIAA Journal* 51.9 (2013), pp. 2049–2075.

[144] M. Kegl and B. Brank. "Shape optimization of truss-stiffened shell structures with variable thickness". In: *Computer Methods in Applied Mechanics and Engineering* 195.19-22 (2006), pp. 2611–2634.

[145] K.-U. Bletzinger, R. Wüchner, F. Daoud, and N. Camprubí. "Computational methods for form finding and optimization of shells and membranes". In: *Computer Methods in Applied Mechanics and Engineering* 194.30-33 (2005), pp. 3438–3452.

[146] K. Ikeya, M. Shimoda, and J.-X. Shi. "Multi-objective free-form optimization for shape and thickness of shell structures with composite materials". In: *Composite Structures* 135 (2016), pp. 262–275.

[147] 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". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 123–150.

[148] T. Maquart. “Trivariate Models Generation From Unstructured Surface Manifolds For Isogeometric Analysis - Application To Reduced Order Modeling With Geometric Parameters”. Theses. Université de Lyon ; INSA-Lyon, 2019.

[149] H. Al Akhras, T. Elguedj, A. Gravouil, and M. Rochette. “Towards an automatic isogeometric analysis suitable trivariate models generation—Application to geometric parametric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 623–645.

[150] P. Gosselet and C. Rey. “Non-overlapping domain decomposition methods in structural mechanics”. In: *Archives of Computational Methods in Engineering* 13.4 (2006), pp. 515–572.

[151] S. K. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. “IETI – Isogeometric Tearing and Interconnecting”. In: *Computer Methods in Applied Mechanics and Engineering* 247-248 (2012), pp. 201–215.

[152] C. Farhat and F.-X. Roux. “A method of finite element tearing and interconnecting and its parallel solution algorithm”. In: *International Journal for Numerical Methods in Engineering* 32.6 (1991), pp. 1205–1227.

[153] P. Tallec, Y. Roeck, and M. Vidrascu. “Domain decomposition methods for large linearly elliptic three-dimensional problems”. In: *Journal of Computational and Applied Mathematics* 34.1 (1991), pp. 93–117.

[154] C. Farhat, P.-S. Chen, J. Mandel, and F. X. Roux. “The two-level FETI method Part II: Extension to shell problems, parallel implementation and performance results”. In: *Computer Methods in Applied Mechanics and Engineering* 155.1-2 (1998), pp. 153–179.

[155] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. “FETI-DP: a dual-primal unified FETI method–part I: A faster alternative to the two-level FETI method”. In: *International Journal for Numerical Methods in Engineering* 50.7 (2001), pp. 1523–1544.

[156] A. Mobasher Amini, D. Dureisseix, P. Cartraud, and N. Buannic. “A domain decomposition method for problems with structural heterogeneities on the interface: Application to a passenger ship”. In: *Computer Methods in Applied Mechanics and Engineering* 198.41-44 (2009), pp. 3452–3463.

[157] P. Gosselet, D. Rixen, F.-X. Roux, and N. Spillane. “Simultaneous FETI and block FETI: Robust domain decomposition with multiple search directions”. In: *International Journal for Numerical Methods in Engineering* 104.10 (2015), pp. 905–927.

[158] C. Bovet, A. Parret-Fréaud, N. Spillane, and P. Gosselet. “Adaptive multipreconditioned FETI: Scalability results and robustness assessment”. In: *Computers & Structures* 193 (2017), pp. 1–20.

[159] D. Dureisseix and C. Farhat. “A numerically scalable domain decomposition method for the solution of frictionless contact problems”. In: *International Journal for Numerical Methods in Engineering* 50.12 (2001), pp. 2643–2666.

[160] C. Bernardi, Y. Maday, and A. T. Patera. “Domain Decomposition by the Mortar Element Method”. In: *Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters*. Dordrecht: Springer Netherlands, 1993, pp. 269–286.

[161] D. Stefanica. “A Numerical Study of FETI Algorithms for Mortar Finite Element Methods”. In: *SIAM Journal on Scientific Computing* 23.4 (2001), pp. 1135–1160.

[162] A. Apostolatos, M. Breitenberger, R. Wüchner, and K.-U. Bletzinger. “Domain Decomposition Methods and Kirchhoff-Love Shell Multipatch Coupling in Isogeometric Analysis”. In: *Isogeometric Analysis and Applications 2014*. Ed. by B. Jüttler and B. Simeon. Cham: Springer International Publishing, 2015, pp. 73–101.

[163] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, and K.-U. Bletzinger. “Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures”. In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 401–457.

[164] A. J. Herrema, E. L. Johnson, D. Proserpio, M. C. Wu, J. Kiendl, and M.-C. Hsu. "Penalty coupling of non-matching isogeometric Kirchhoff–Love shell patches with application to composite wind turbine blades". In: *Computer Methods in Applied Mechanics and Engineering* (2018).

[165] E. Brivadis, A. Buffa, B. Wohlmuth, and L. Wunderlich. "Isogeometric mortar methods". In: *Computer Methods in Applied Mechanics and Engineering* 284 (2015), pp. 292–319. arXiv: 1407.8313.

[166] R. Bouclier, J.-C. Passieux, and M. Salaün. "Local enrichment of NURBS patches using a non-intrusive coupling strategy: Geometric details, local refinement, inclusion, fracture". In: *Computer Methods in Applied Mechanics and Engineering* 300 (2016), pp. 1–26.

[167] 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". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 449–496.

[168] Z. Zou, M. Scott, M. Borden, D. Thomas, W. Dornisch, and E. Brivadis. "Isogeometric Bézier dual mortaring: Refineable higher-order spline dual bases and weakly continuous geometry". In: *Computer Methods in Applied Mechanics and Engineering* 333 (2018), pp. 497–534.

[169] L. Wunderlich, A. Seitz, M. D. Alaydın, B. Wohlmuth, and A. Popp. "Biorthogonal splines for optimal weak patch-coupling in isogeometric analysis with applications to finite deformation elasticity". In: *Computer Methods in Applied Mechanics and Engineering* 346 (2019), pp. 197–215.

[170] S. Schuß, M. Dittmann, B. Wohlmuth, S. Klinkel, and C. Hesch. "Multi-patch isogeometric analysis for Kirchhoff–Love shell elements". In: *Computer Methods in Applied Mechanics and Engineering* 349 (2019), pp. 91–116.

[171] 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". In: *International Journal for Numerical Methods in Engineering* 97.7 (2014), pp. 473–504.

[172] 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". In: *Computational Mechanics* 53.6 (2014), pp. 1163–1182.

[173] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, S. K. 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". In: *Computer Methods in Applied Mechanics and Engineering* 309 (2016), pp. 625–652.

[174] Y. Guo, M. Ruess, and D. Schillinger. "A parameter-free variational coupling approach for trimmed isogeometric thin shells". In: *Computational Mechanics* 59.4 (2017), pp. 693–715.

[175] N. Nguyen-Thanh, K. Zhou, X. Zhuang, P. Areias, H. Nguyen-Xuan, Y. Bazilevs, and T. Rabczuk. "Isogeometric analysis of large-deformation thin shells using RHT-splines for multiple-patch coupling". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 1157–1178.

[176] R. Bouclier and J.-C. Passieux. "A Nitsche-based non-intrusive coupling strategy for global/local isogeometric structural analysis". In: *Computer Methods in Applied Mechanics and Engineering* 340 (2018), pp. 253–277.

[177] 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". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 123–150.

[178] E. Burman. "A Penalty-Free Nonsymmetric Nitsche-Type Method for the Weak Imposition of Boundary Conditions". In: *SIAM Journal on Numerical Analysis* 50.4 (2012), pp. 1959–1981.

[179] Y. Guo, J. Heller, T. J. R. Hughes, M. Ruess, and D. Schillinger. "Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format". In: *Computer Methods in Applied Mechanics and Engineering* 336.November (2018), pp. 39–79.

[180] T. X. Duong, F. Roohbakhshan, and R. A. Sauer. "A new rotation-free isogeometric thin shell formulation and a corresponding continuity constraint for patch boundaries". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 43–83.

[181] M. Bernadou and A. Cubier. "Numerical analysis of junctions between thin shells Part 1: Continuous problems". In: *Computer Methods in Applied Mechanics and Engineering* 161.3-4 (1998), pp. 349–363.

[182] A. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, and K.-U. Bletzinger. "Embedded structural entities in NURBS-based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 325.July (2017), pp. 198–218.

[183] D. Schöllhammer and T. P. Fries. "Kirchhoff–Love shell theory based on tangential differential calculus". In: *Computational Mechanics* (2018).

[184] D. J. Rixen, C. Farhat, R. Tezaur, and J. Mandel. "Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver". In: *International Journal for Numerical Methods in Engineering* 46.4 (1999), pp. 501–533.

[185] D. Stefanica. "Parallel FETI algorithms for mortars". In: *Applied Numerical Mathematics* 54.2 (2005), pp. 266–279.

[186] P. Jolivet. "Domain decomposition methods. Application to high-performance computing". Theses. Université Grenoble Alpes, 2014.

[187] T. Kozubek, V. Vondrák, M. Menšík, D. Horák, Z. Dostál, V. Hapla, P. Kabelíková, and M. Čermák. "Total FETI domain decomposition method and its massively parallel implementation". In: *Advances in Engineering Software* 60-61 (2013), pp. 14–22.

[188] Z. Dostál, D. Horák, and R. Kučera. "Total FETI-an easier implementable variant of the FETI method for numerical solution of elliptic PDE". In: *Communications in Numerical Methods in Engineering* 22.12 (2006), pp. 1155–1162.

[189] E. B. Savenkov, H. Andrae, O. P. Iliev, and H. Iliev. *An analysis of one regularization approach for solution of pure Neumann problem*. Fraunhofer-Institut für Techno- und Wirtschaftsmathematik, 2008.

[190] C. Farhat and M. Géradin. "On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures". In: *International Journal for Numerical Methods in Engineering* 41.4 (1998), pp. 675–696.

[191] G. Golub and W. Kahan. "Calculating the Singular Values and Pseudo-Inverse of a Matrix". In: *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis* 2.2 (1965), pp. 205–224.

[192] M. Kuchta, K.-A. Mardal, and M. Mortensen. "On the singular Neumann problem in linear elasticity". In: *Numerical Linear Algebra with Applications* 26.1 (2019), e2212.

[193] D. J. Rixen. "Extended preconditioners for the FETI method applied to constrained problems". In: *International Journal for Numerical Methods in Engineering* 54.1 (2002), pp. 1–26.

[194] C. Lacour. "Analyse et résolution numérique de méthodes de sous-domaines non conformes pour des problèmes de plaques." PhD thesis. Université Pierre et Marie Curie - Paris VI, 1997.

[195] D. J. Rixen and C. Farhat. "A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems". In: *International Journal for Numerical Methods in Engineering* 44.4 (1999), pp. 489–516.

[196] A. Klawonn and O. Widlund. "FETI and Neumann-Neumann iterative substructuring methods: Connections and new results". In: *Communications on Pure and Applied Mathematics* 54.1 (2001), pp. 57–90.

[197] T. W. Sederberg and S. R. Parry. "Free-form Deformation of Solid Geometric Models". In: *SIGGRAPH Comput. Graph.* 20.4 (1986), pp. 151–160.

[198] G. K. W. Kenway and J. R. R. A. Martins. "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration". In: *Journal of Aircraft* 51.1 (2014), pp. 144–160.

[199] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. "Geometric modeling, isogeometric analysis and the finite cell method". In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (2012), pp. 104–115.

[200] G. Legrain. "A NURBS enhanced extended finite element approach for unfitted CAD analysis". In: *Computational Mechanics* 52.4 (2013), pp. 913–929.

[201] D. Schillinger and M. Ruess. "The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models". In: *Archives of Computational Methods in Engineering* 22.3 (2015), pp. 391–455.

[202] P. Hansbo, M. G. Larson, and K. Larsson. "Cut Finite Element Methods for Linear Elasticity Problems". In: *Lecture Notes in Computational Science and Engineering*. Vol. 121. 2017, pp. 25–63. arXiv: 1703.04377.

[203] E. Burman and P. Hansbo. "Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method". In: *Applied Numerical Mathematics* 62.4 (2012), pp. 328–341.

[204] W. Dornisch and S. Klinkel. "Treatment of Reissner–Mindlin shells with kinks without the need for drilling rotation stabilization in an isogeometric framework". In: *Computer Methods in Applied Mechanics and Engineering* 276 (2014), pp. 35–66.

[205] R. Schmidt. "Trimming, Mapping, and Optimization in Isogeometric Analysis of Shell Structures". PhD thesis. Technische Universität München, 2013.

[206] M. Andreoli, J. Ales, and J.-A. Desideri. *Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics*. Research Report RR-5019. INRIA, 2003. URL: https://hal.inria.fr/inria-00071565.

[207] R. Duvigneau. *Adaptive Parameterization using Free-Form Deformation for Aerodynamic Shape Optimization*. Research Report. 2006, p. 40. URL: https://hal.inria.fr/inria-00085058.

[208] T. Lassila and G. Rozza. "Parametric free-form shape design with PDE models and reduced basis method". In: *Computer Methods in Applied Mechanics and Engineering* 199.23-24 (2010), pp. 1583–1592.

[209] T. Borrvall and J. Petersson. "Large-scale topology optimization in 3D using parallel computing". In: *Computer Methods in Applied Mechanics and Engineering* 190.46-47 (2001), pp. 6201–6229.

[210] T. S. Kim, J. E. Kim, and Y. Y. Kim. "Parallelized structural topology optimization for eigenvalue problems". In: *International Journal of Solids and Structures* 41.9-10 (2004), pp. 2623–2641.

[211] N. Aage and B. S. Lazarov. "Parallel framework for topology optimization using the method of moving asymptotes". In: *Structural and Multidisciplinary Optimization* 47.4 (2013), pp. 493–505.

[212] J. D. Deaton and R. V. Grandhi. "A survey of structural and multidisciplinary continuum topology optimization: post 2000". In: *Structural and Multidisciplinary Optimization* 49.1 (2014), pp. 1–38.

[213] N. Aage, E. Andreassen, and B. S. Lazarov. "Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework". In: *Structural and Multidisciplinary Optimization* 51.3 (2015), pp. 565–572.

[214] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. "Problem Formulation for Multidisciplinary Optimization". In: *SIAM Journal on Optimization* 4.4 (1994), pp. 754–776.

[215] Robert McNeel & Associates. *Design, model, present, analyze, realize...* 2019. URL: `https://www.rhino3d.com/`.

[216] J. Vassberg, M. Dehaan, M. Rivers, and R. Wahls. "Development of a Common Research Model for Applied CFD Validation Studies". In: *26th AIAA Applied Aerodynamics Conference*. August. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2008, pp. 1–22.

[217] S. Keye, T. Klimmek, M. Abu-Zurayk, M. Schulze, and C. Ilic. "Aero-Structural Optimization of the NASA Common Research Model". In: *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. June. Reston, Virginia: American Institute of Aeronautics and Astronautics, 2017, pp. 1–9.

[218] S. R. Jongerius and D. Lentink. "Structural Analysis of a Dragonfly Wing". In: *Experimental Mechanics* 50.9 (2010), pp. 1323–1334.

[219] D. Lazzara, J. Parham, and R. Haimes. "On Structural Layout Using Multifidelity Geometry in Aircraft Conceptual Design". In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. January. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2010, pp. 1–20.

[220] K. Svanberg. "The method of moving asymptotes—a new method for structural optimization". In: *International Journal for Numerical Methods in Engineering* 24.2 (1987), pp. 359–373.

FOLIO ADMINISTRATIF

THÈSE DE L'UNIVERSITÉ DE LYON OPERÉE AU SEIN DE L'INSA LYON

NOM : HIRSCHLER
Prénom : Thibaut

DATE de SOUTENANCE : 14/11/2019

TITRE : **IsoGeometric Modeling for the Optimal Design of Aerostructures**

NATURE : Doctorat

Numéro d'ordre : 2019LYSEI092

École doctorale : MEGA (ED 162) – Mécanique, Énergétique, Génie Civil, Acoustique
Spécialité : Mécanique

RÉSUMÉ

Concevoir des structures au regard de leur comportement mécanique est une tâche importante en ingénierie. Cependant, cette phase de conception peut s'avérer longue et fastidieuse lorsque les relations de cause à effet ne sont pas correctement identifiées. Plus spécifiquement, il est généralement difficile d'obtenir la meilleure forme d'une structure (la forme optimale), car cela fait appel à de nombreuses compétences. En effet, des outils avancés pour la modélisation géométrique sont nécessaires pour représenter fidèlement la structure et pour explorer une large variété de formes. A cela s'ajoute le besoin d'une méthode de calcul de structures compétente et rapide afin de réduire la durée du processus d'optimisation. Un lien étroit entre le modèle géométrique et celui d'analyse est souhaité car tous deux sont amenés à communiquer successivement. Ainsi, l'Analyse IsoGéométrique apparaît comme un outil adéquat pour les problèmes d'optimisation paramétrique de formes. En effet, cette méthode s'appuie sur un modèle unique pour décrire fidèlement la géométrie et pour effectuer l'analyse. Entièrement basée sur le concept de l'AIG, nous construisons une stratégie de conception optimale de la forme pour les structures raidies, omniprésentes en aéronautique. Nous présentons une formulation massif coque permettant d'imposer des variations continues d'épaisseur à une structure mince. Nous formulons des sensibilités analytiques pour les éléments isogéométriques standards et coques. Aussi, nous présentons une approche immergée pour appliquer des changements de forme aux structures raidies. Du point de vue de l'analyse, nous introduisons une formulation coque adaptée aux compositions géométriques et nous construisons un algorithme dédié aux cas de discrétisations non-conformes en s'appuyant sur les méthodes de couplage Mortar et de décomposition de domaine. *In fine*, tous ces développements sont fusionnés et permettent de traiter divers exemples avec des niveaux de complexité croissants.

Mots clés : Optimisation de forme, Analyse Isogéométrique, Coques, Structures raidies, Couplage, Décomposition de domaine.

ABSTRACT

Designing structural parts against the material limits, the impact of loads, and many other constraints, is of standard interest in engineering. However, improving the design of a structure can be long and drawn out, especially when a clear understanding of cause-effect relationships is missing. Finding the best possible design, namely the optimal design, is a complex task because it requires several competences. Usually, efficient geometric modeling is needed to accurately represent the structure. Conjointly, the geometric model should provide high flexibility during the design exploration. In addition, structural analysis must be fast enough to shorten the overall process. Besides, for the sake of compactness, a close connection between the geometric model and the structural analysis seems essential. Finally, all modeling choices are deeply related, and thus, they should be thought and built accordingly to the others. Therefore, IsoGeometric Analysis appears as a powerful tool for structural optimization since it uses a unique model with both high quality geometric and analysis properties. Here, we present a compact framework built on the core idea of IGA. We strive to construct unified models with new opportunities for structural design with a direct application to stiffened Aerostructures. More specifically, we present a solid-shell approach to impose continuous thickness variations. We formulate analytical sensitivities for standard and shell formulations. Then, we introduce an embedded technique that enables to impose complex shape updates. From the analysis point of view, we build a shell formulation dedicated to geometric compositions and we design a specific solver based on Domain Decomposition methods and Mortar approach for the coupling of non-conforming discretizations. Different examples with increasing level of complexity show the performances of the adopted methodologies.

Keywords: Shape optimization, Isogeometric analysis, Shells, Stiffened structures, Coupling, Domain decomposition.

Laboratoire de recherche : Laboratoire de Mécanique des Contacts et des Structures (LaMCoS)

Directeur de thèse : ELGUEDJ, Thomas

Président du jury : RIXEN, Daniel J.

Composition du jury : BOUCARD, Pierre-Alain, BOUCLIER, Robin, COLLIN, Annabelle, DUVAL, Arnaud, ELGUEDJ, Thomas, EYHERAMENDY Dominique, KIENDL, Josef, MORLIER, Joseph, RIXEN, Daniel J., TOUZEAU, Josselyn