

**A framework for isogeometric-analysis-based design and
optimization of wind turbine blades**

by

Austin J. Herrema

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Co-majors: Mechanical Engineering,
Wind Energy Science, Engineering, and Policy

Program of Study Committee:
Ming-Chen Hsu, Major Professor
Baskar Ganapathysubramanian
Adarsh Krishnamurthy
James McCalley
Frank Peters

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

Copyright © Austin J. Herrema, 2018. All rights reserved.

DEDICATION

To my wife, Marisa. Your love, support, and joy during these turbulent, adventure-filled years have been utterly inspiring, and I cannot imagine having completed this degree without you by my side.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
ACKNOWLEDGEMENTS	xiv
ABSTRACT	xvii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. BACKGROUND	5
2.1 Optimal wind turbine blade design	5
2.1.1 Engineering optimization	5
2.1.2 Optimization of wind turbine blades	7
2.2 NURBS-based isogeometric analysis	9
2.3 Composite isogeometric Kirchhoff–Love shells	12
2.4 Acknowledgements	17
CHAPTER 3. PARAMETRIC DESIGN OPTIMIZATION USING IGA	18
3.1 Chapter overview	18
3.2 IGA and design-through-analysis	19
3.3 Isogeometric design optimization framework	22
3.3.1 Overview	22
3.3.2 Design	24
3.3.3 Analysis	29

3.3.4	Optimization	31
3.3.5	Visualization	33
3.4	Tube profile optimization	35
3.4.1	Objective function	35
3.4.2	Design variables	37
3.4.3	Simulation setup and solution strategies	38
3.4.4	Results and discussion	39
3.5	Wind turbine blade optimization	41
3.5.1	Objective function	41
3.5.2	Design variables and constraints	45
3.5.3	Simulation setup and solution strategies	47
3.5.4	Results and discussion	49
3.6	Chapter conclusion	52
3.7	Acknowledgements	54
CHAPTER 4. A PENALTY FORMULATION FOR PATCH COUPLING		55
4.1	Chapter overview	55
4.2	Coupling of isogeometric Kirchhoff–Love patches	56
4.3	Penalty-based patch coupling formulation	58
4.3.1	Coupling methodology	58
4.3.2	Implementation	62
4.3.3	Penalty parameter formulations	63
4.4	Benchmark examples	65
4.4.1	Scordelis-Lo roof	66
4.4.2	T-beam	69
4.4.3	Plate buckling	72
4.4.4	Nonlinear slit annular plate	75

4.5	Chapter conclusion	78
4.6	Acknowledgements	79
CHAPTER 5. IGA-BASED BLADE DESIGN		80
5.1	Chapter overview	80
5.2	Framework overview	81
5.2.1	Parametric generation of blade geometry	81
5.2.2	Isogeometric shell analysis	81
5.2.3	IGA solution strategies	82
5.2.4	Integration with aeroelastic simulation	83
5.3	Wind turbine blade analysis	83
5.3.1	NREL/SNL 5 MW blade model	84
5.3.2	Model verification through vibration analysis	88
5.3.3	Buckling analysis	90
5.3.4	Comparative mesh refinement study	91
5.3.5	Alternative penalty parameter formulations	93
5.3.6	Nonlinear deflection analysis	96
5.4	Wind turbine blade optimization	98
5.4.1	Aeroelastic simulation	98
5.4.2	Objective function	99
5.4.3	Constraints	100
5.4.4	Design variables	102
5.4.5	Implementation and solution strategies	104
5.4.6	Results and discussion	104
5.5	Chapter conclusion	110
5.6	Acknowledgements	111
CHAPTER 6. CONCLUSION AND OUTLOOK		112

BIBLIOGRAPHY	115
APPENDIX. NREL/SNL 5 MW BLADE DEFINITION TABLE	132

LIST OF TABLES

Table 3.1	Summary of results of interest for original and optimized designs. Only tip deflection is constrained for the first case, whereas both tip deflection and maximum strain are constrained for the second case. Additional profit is defined over the entire lifetime of a large-scale offshore wind farm featuring an given blade design.	50
Table 5.1	Orthotropic material properties used in the NREL/SNL 5 MW blade design.	87
Table 5.2	Frequencies of vibration of the reference model and the IGA-based model for three different levels of discretization. Mesh 1 indicates the coarsest discretization and Mesh 3 indicates the finest. Blade modes, v_i , depicted below for Mesh 1.	89
Table 5.4	The price of the various materials used in the NREL/SNL 5 MW blade design. Fiberglass costs (E-LT-5500, SNL Triax, and Saertex) are based on dry fiber material whereas UD Carbon material is pre-impregnated.	101
Table 5.5	Performance details of two of the best-performing optimized designs.	106
Table 5.6	Various performance metrics of the reference and optimized 5 MW blade designs.	108
Table A.1	Definition of cross section parameters used to construct the NREL/SNL 5 MW blade model. “DP” stands for “division point.”	132

LIST OF FIGURES

Figure 2.1	Composite layup with non-uniform and nonsymmetric ply distribution [51, Figure 3].	16
Figure 3.1	Overall structure of the isogeometric design optimization framework laid out in Grasshopper, an algorithmic modeling interface for Rhino. Optimization procedures can be performed either inside or outside of the Grasshopper environment using various optimization toolboxes, such as MATLAB [91], Dakota [92], Galapagos [60], etc.	23
Figure 3.2	Illustration of the different designs achieved when design intent (geometric connection of surfaces 1 and 2) is maintained (bottom case) versus when design intent is not respected (top case) for a given model change.	26
Figure 3.3	Family of tube designs (bottom) and associated Grasshopper generative algorithm $\Theta^t(\chi^t)$ (top).	28
Figure 3.4	Family of blade designs (bottom) and associated Grasshopper generative algorithm $\Theta^b(\chi^b)$ (top).	29
Figure 3.5	Grasshopper implementation of visualization methodology.	36
Figure 3.6	Definition of the design variable r for the internally pressurized tube problem.	38
Figure 3.7	Design variable r and current best objective function value versus number of pattern search iterations.	40

Figure 3.8	Undeformed and deformed shapes of current best tube design at selected optimization iterations. Color contour denotes value of maximum component of curvature change.	40
Figure 3.9	Demonstration of strategy for variation of chord profile using reduced number of parameters. Internal control points of a quadratic B-spline are moved and the variation profile is superposed upon the original profile. Original blade geometry is shown in black (second from bottom) and new blade geometry is shown in blue (bottom). .	46
Figure 3.10	Simplified composite layup used for wind turbine blade optimization. Green color (top) indicates base E-LT-5500 fiberglass over entire blade, blue color (middle) indicates root buildup of SNL Triax, and purple color (bottom) indicates spar cap region made up of uni-directional carbon.	48
Figure 3.11	Iterative history of best objective function value for each constraint case (left) and chord profiles of original and optimized designs (right).	50
Figure 3.12	Comparison of original and optimized blade shapes viewed from the flapwise direction. Station sizes and locations are indicated by black lines.	50
Figure 3.13	Comparison of strain distributions on original and optimized blade designs (shown in their most deformed states). The first principal in-plane strain on the outer surface of the shells is plotted. Blades are pitched 35 degrees from angle used in Figure 3.12 to show region of strain concentration.	51

Figure 4.2	Various possible patch connection configurations as indicated by the relative thickness of the coupled patches. The general composite configuration with a material stiffness discontinuity at the coupling location (a); composite configuration with the same stiffness on both patches at the coupling location (b); and isotropic configuration with the same stiffness on both patches at the coupling location (c) are depicted.	64
Figure 4.4	Scordelis-Lo roof problem description and deformation (scaled). . .	66
Figure 4.6	Meshes for the matching (a) and non-matching (b) configurations of the Scordelis-Lo roof. Black lines indicate patch boundaries. . . .	67
Figure 4.8	Scordelis-Lo roof displacement at midpoint, normalized with respect to reference value, with increasing penalty value α using the proposed penalty approach and the α -only approach for both matching and non-matching discretizations and different combinations of setup variables.	68
Figure 4.9	Convergence of the roof midpoint displacement under h -refinement for the proposed penalty approach with $\alpha = 10^3$	70
Figure 4.11	T-beam problem description and deformation (scaled).	70
Figure 4.12	Meshes for the matching (left) and non-matching (right) configurations of the T-beam problem. Black lines indicate patch boundaries.	71
Figure 4.14	Top view of meshes for the matching and non-matching configurations of the T-beam problem. Circular markers indicate discretization of perpendicular patch.	71
Figure 4.16	Angle between patches of the T-beam (left) and total twist at the end of the vertical patch (right) with increasing penalty value α for both matching and non-matching configurations.	72

Figure 4.18	Plate buckling problem description and the deformation for the first buckling mode (color contour indicates relative displacement magnitude).	73
Figure 4.20	Meshes for the (a) matching and (b) non-matching configurations of the plate buckling problem. Black lines indicate patch boundaries.	74
Figure 4.21	Buckling load factor with increasing α for both matching and non-matching configurations of the plate buckling problem.	74
Figure 4.23	Nonlinear slit annular plate problem description and deformation.	75
Figure 4.25	Meshes for the (a) matching and (b) non-matching cases of the nonlinear slit annular plate. Black lines indicate patch boundaries.	76
Figure 4.26	Displacement of point B under h -refinement for both matching and non-matching configurations using $\alpha = 10^3$	77
Figure 4.28	Displacement at points A and B of the slit annular plate due to the maximum applied P for varying values of α	77
Figure 4.29	Vertical displacement of points A and B versus applied distributed load for the nonlinear slit annular plate with $\alpha = 10^3$	78
Figure 5.1	Two NURBS surfaces, \mathcal{S}^A and \mathcal{S}^B , representing the shear web and spar cap of a wind turbine blade, which must be coupled along the curve \mathcal{L} . Control point locations indicated by spheres.	82
Figure 5.2	Illustration of parameters defined at each station along the blade span in order to define blade geometry and material regions. Parameter values for the NREL/SNL 5 MW blade are given in Table A.1.	85
Figure 5.3	A NURBS-based model of the NREL/SNL 5 MW blade. Colors indicate zones with distinct material stacking sequences. Grey lines indicate element edges and black lines indicate surface patch edges. Stacking sequence for each material zone provided; stack definitions can be found in Figure 5.4.	86

Figure 5.4	Definition of material stack thickness distribution and other relevant information.	87
Figure 5.5	Mesh densities for IGA Meshes 1, 2, and 3.	89
Figure 5.6	NREL/SNL 5 MW blade frequencies of vibration using a range of values for α	90
Figure 5.7	Convergence of lowest predicted buckling load factor under mesh refinement relative to the most converged solution for both IGA computations (blue) and computations performed by Resor [118] (red).	92
Figure 5.8	Lowest buckling load factor as a function of α for the various possible methods of formulating the penalty parameters.	95
Figure 5.9	First (top), second (middle), and third (bottom) buckling modes of the NREL/SNL 5 MW blade using $\alpha = 10^3$. Color contour indicates relative magnitude of deflection in buckling.	96
Figure 5.10	Maximum flapwise tip deflection due to nonlinear analysis for a range of values of α	97
Figure 5.11	Deformation of the NREL/SNL 5 MW blade due to nonlinear analysis using $\alpha = 10^3$	97
Figure 5.12	Material stack thickness distributions with design variables identified by numerical markers. For variables with markers on horizontal line segments, thickness variation is applied to the line segment end points. For variables with markers directly on points, thickness variation is applied directly to the point. Indices of the variables Δt_i and corresponding names given at right.	103

Figure 5.13	At left, blade mass and material cost of each starting point (diamonds) and blade mass and material cost of corresponding optimized designs (circles). Unfilled points signify invalid designs as indicated by an objective function penalization of greater than 0.05. At right, blade mass and material cost of only valid optimized points and reference design [118].	105
Figure 5.14	Thickness design variable values for the reference, balanced, and lightweight 5 MW blade designs. Bars indicate the allowable range for each design variable.	106
Figure 5.15	The total mass and material cost contributions of three primary types of materials—fiberglass, which includes E-LT-5500, SNL Triax, Saertex, and resin; foam; and carbon—to the baseline, balanced, and lightweight 5 MW blade designs.	107
Figure 5.16	The first buckling mode of the reference (left), balanced (middle), and lightweight (right) 5 MW blade designs. Modes are shown in the internal structure of the blade (bottom) and on the suction side of the blade (top).	109

ACKNOWLEDGEMENTS

I would first like to express deep gratitude and appreciation to my advisor, Dr. Ming-Chen Hsu. In addition to effectively and graciously sharing much of his knowledge, Dr. Hsu has continually encouraged and challenged me throughout my work. His guidance has been instrumental in developing me as a researcher, for which I am immensely grateful, and his passion for computational science and engineering has been inspirational. This dissertation would never have been written were it not for his unyielding support.

I also wish to thank my other committee members—Dr. Baskar Ganapathysubramanian, Dr. Adarsh Krishnamurthy, Dr. James McCalley, and Dr. Frank Peters—for their time and guidance throughout my degree. I am especially grateful to Dr. McCalley for his tireless and devoted leadership of the Wind Energy Science, Engineering, and Policy (WESEP) program at Iowa State University. Being a part of this unique, interdisciplinary program has been thoroughly enjoyable, and Dr. McCalley has been relentlessly dedicated to educating and bringing together a diverse group of students. I have felt profoundly well-supported by the WESEP program throughout my time at Iowa State.

Special thanks are in order to Dr. Josef Kiendl. Dr. Kiendl has been a key collaborator throughout this work, and he has always been willing to patiently and attentively lend his expertise. Further, during the four weeks that my wife and I spent in Germany while I performed research at Braunschweig University of Technology, Dr. Kiendl was a kind and hospitable host.

My wife and I were also able to spend seven weeks in Roskilde, Denmark while I performed research at Risø National Laboratory for Sustainable Energy. I am indebted to Flemming Rasmussen for agreeing to host me and to Dr. Frederik Zahle and Dr. Katherine Dykes for guiding my work during that time. I also greatly valued the fruitful discussions

that I was able to have with Dr. Frederik Zahle, Dr. Michael McWilliam, Dr. Anders Smærup Olson, and many others at DTU. My time at DTU Risø and in Roskilde was both enjoyable and enlightening, and I will continue to treasure that experience.

The wind energy community as a whole has been quite receptive and, correspondingly, has been instrumental in shaping my research. I am grateful for even the shortest discussions that I was able to have with experts within the wind energy industry, national laboratories, and academia. The support and insight provided by Envision Energy's Global Blade Innovation Center has been particularly valuable. I would like to thank Kevin Standish for making my relationship with Envision possible and Sunil Jonnalagadda, Justin Mullings, Amaury Vuillaume, and many others throughout the organization for playing a key role in helping me to develop more detailed and accurate industrial perspectives.

Of course, I am forever indebted to my friends and family for supporting me throughout my PhD career and long before it. I cannot possibly list everyone here by name, but I am grateful for the camaraderie and collaboration of my labmates and friends at Iowa State, a supportive church community in Ankeny, the continued love of treasured friends throughout the country, the outstanding preparation I received at Dordt College, the dedication of educators throughout my life, and the generations before me who have shaped my life tremendously. Perhaps most importantly, words cannot express the strongest love and thanks owed to my immediate family. My wife, Marisa, has been a constant and crucial source of abounding joy and support, and my parents, Scott and Beverly, have sacrificed for me, loved me, and shaped me in truly foundational ways, many of which I am sure I do not even now grasp. Success is communal and I am keenly aware that, throughout my life, I have had the undeserved and unequivocal privilege of a strong and loving community around me.

Lastly, this work would be impossible if not for a faithfully upheld universe and meaningless if not for the deep and beautiful mystery of humanity. Thus, in the wise words of

my older brother, I must thank my Savior and my Lord, Jesus Christ, by whom I exist, through whom I am saved, and for whom I live. Soli Deo gloria.

ABSTRACT

Typical wind turbine blade design procedures employ reduced-order models almost exclusively for early-stage design; high-fidelity, finite-element-based procedures are reserved for later design stages because they entail complex workflows, large volumes of data, and significant computational expense. Yet, high-fidelity structural analyses often provide design-governing feedback such as buckling load factors. Mitigation of the issues of workflow complexity, data volume, and computational expense would allow designers to utilize high-fidelity structural analysis feedback earlier, more easily, and more often in the design process. Thus, this work presents a blade analysis framework which employs isogeometric analysis (IGA), a simulation method that overcomes many of the aforementioned drawbacks associated with traditional finite element analysis (FEA). IGA directly utilizes the mathematical models generated by computer-aided design (CAD) software, requires less user interaction and no conversion of CAD geometries to finite element meshes, and tends to have superior per-degree-of-freedom accuracy compared to traditional FEA.

The presented framework employs the parametric capabilities of the Grasshopper algorithmic modeling interface developed for the CAD software Rhinoceros 3D. This Grasshopper-based framework enables seamless, iterative design and IGA of CAD-based geometries and is demonstrated through the optimization of both a pressurized tube and a simplified wind turbine blade design. Further, because engineering models, such as wind turbine blades, are typically composed of numerous surface patches, a novel patch coupling technique is presented. For the sake of straightforward implementation and flexibility, the coupling technique is based on a penalty energy approach. Formulations for the penalty parameters are proposed to eliminate the problem-dependent nature of the penalty method. This cou-

pling methodology is successfully demonstrated using a number of multi-patch benchmark examples with both matching and non-matching interface discretizations.

Together, these technologies enable practical and efficient design and analysis of wind turbine blade shell structures. The presented IGA approach is employed to perform vibration, buckling, and nonlinear deformation analysis of the NREL/SNL 5 MW wind turbine blade, validating the effectiveness of the proposed approach for realistic, composite wind turbine blade designs. Further, a blade design framework that combines reduced-order aeroelastic analysis with the presented IGA methodologies is outlined. Aeroelastic analysis is used to efficiently provide dynamic kinematic data for a wide range of wind load cases, while IGA is used to perform high-fidelity buckling analysis. Finally, the value and feasibility of incorporating high-fidelity IGA feedback into optimization is demonstrated through optimization of the NREL/SNL 5 MW wind turbine blade. Alternative structural designs that have improved blade mass and material cost characteristics are identified, and IGA-based buckling analysis is shown to provide design-governing constraint information.

CHAPTER 1. INTRODUCTION

Modern wind energy technologies provide clean, secure, and economical [1] electricity generation. Still, some of the challenges associated with wind energy, such as intermittent power generation due to variable wind speeds, continue to motivate technological innovation. For example, one of the foremost trends in recent years has been the use of continually larger rotor diameters; in fact, this growth has outpaced growth both in nameplate capacity and hub height [2]. Large-diameter rotors are commonly designed specifically for sites with low wind speeds and also tend to feature higher capacity factors. This is evident in that, while the average capacity factor among projects built from 2004 to 2011 was 32.1%, the average capacity factor among projects built in 2014 and 2015 was 42.5%. Even more recently, GE Renewable Energy announced a 12 MW wind turbine having a rotor diameter of 220 m and a capacity factor of 63% [3]. These shifts demonstrate that innovative large-rotor design has the potential to drive down energy costs and to improve the overall reliability of wind energy.

Although larger rotors clearly have benefits with respect to power generation, they also present some challenges. Under some simplistic assumptions, it can be shown that, while rotor power increase is related to rotor size through a quadratic relationship, rotor weight increase is related to rotor size through a cubic relationship [4]. This is known as the cube-square law and it suggests significant challenges in the design of larger blades, especially because cost-effectiveness is a priority. Although the cube-square law appears to have been overcome in practice through a variety of design methods [5], increasing blade length still entails considerable structural design challenges as a result of shifts in design-governing

phenomena. In particular, it has been suggested that blade buckling, especially with regard to nonlinear effects, is of concern [6–10].

Wind turbine blade design typically relies heavily on computationally efficient, reduced-order aerodynamic and structural models, especially in the early stages of blade design. This is a sensible approach given the vast design space, the large number of design iterations typically performed, and the ample accuracy of many reduced-order models. The value of such an approach can be especially realized in the context of optimization [11–17]. Some types of analyses, however, are more difficult to perform reliably using reduced-order models. Thus, higher-fidelity finite element analysis (FEA), including both shell and solid FEA, is typically employed in the later design stages to evaluate, for example, buckling load factors as required by design standards such as the IEC 61400 [18]. Because blades are often initially optimized without taking this accurate buckling feedback into account, late-stage FEA can reveal design flaws that trigger additional, laborious design iteration. Notably, the focus during these iterations usually shifts away from rigorous blade optimality and towards mere satisfaction of design standards. While an ideal design procedure would incorporate accurate buckling feedback throughout optimization, the computational cost and labor associated with traditional high-fidelity analyses are prohibitive. Thus, simplification of high-fidelity structural analysis workflows and methodologies could improve not only the design iteration that occurs in later blade design stages, but also the feasibility of incorporating accurate high-fidelity analyses into optimization.

Isogeometric analysis (IGA) was first introduced by Hughes et al. [19] and is based on the idea that the smooth, parametric functions used for geometry modeling in computer-aided design (CAD) can also be used as finite element basis functions, eliminating the need for extra mesh generation. IGA is therefore capable of effectively uniting design and analysis paradigms, employing a single model for both design and analysis, and ultimately reducing requisite user interaction. IGA has also been shown to feature significantly improved per-degree-of-freedom accuracy relative to traditional FEA in many cases [20, 21].

Despite the potential benefits provided by IGA, a few key barriers prevent it from being readily adopted in wind turbine blade design. One of these barriers is a result of the geometry modeling procedure itself. Although modern CAD software can employ sophisticated algorithms to generate a wide range of surface geometries, such as wind turbine blades, not all resultant geometries are directly suitable for IGA. This and other factors have led to generally poor integration of IGA into the interactive CAD environment. Additionally, even if care is taken to ensure analysis-suitability of the individual surface patches in a multi-patch design, the isogeometric Kirchhoff–Love formulation, which is widely utilized for the sake of efficiency, dictates that surface interfaces, especially with non-matching discretizations, cannot be accurately simulated without special treatment. Lastly, although high-fidelity shell analysis is indispensable at various points throughout the blade design procedure, efficient, reduced-order blade models are sufficiently accurate for many tasks. Thus, the benefits of IGA in blade design and optimization are best realized when IGA is used in conjunction with such reduced-order models. Past wind turbine and wind turbine blade simulations have employed IGA [22–33], but none of these efforts have simultaneously integrated IGA into a parametric design environment, employed practical, discretization-agnostic methodologies for coupling surfaces in analysis, and engaged the various types of structural analysis, such as buckling analysis, that are particularly relevant in blade design.

The present work therefore outlines a framework for IGA-based design and optimization of wind turbine blade structures. In Chapter 2, some of the relevant technical background is summarized. In Chapter 3, the generation and analysis of IGA-suitable, parametric geometries in an interactive CAD environment is enabled through the construction of an isogeometric design optimization framework. The framework is demonstrated through the parametric optimization of both a pressurized tube and a simplified wind turbine blade design. In Chapter 4, a unique solution to the problem of isogeometric Kirchhoff–Love shell patch coupling is presented. The approach employs a penalty coupling methodology and novel formulations for the penalty parameters are presented to eliminate the parameters’

problem-dependent nature. The approach is thoroughly explored and tested using benchmark problems. In Chapter 5, these novel approaches are adopted in the development of an IGA-based wind turbine blade design and optimization framework. The framework employs both IGA and FAST, an aeroelastic modeling tool developed by the National Renewable Energy Laboratory (NREL). The framework is demonstrated through optimization of the composite layup of a 5 MW blade design; a number of design alternatives with potentially desirable mass and material cost characteristics are identified. Chapter 6 presents conclusions and possible directions for future research.

CHAPTER 2. BACKGROUND

2.1 Optimal wind turbine blade design

2.1.1 Engineering optimization

Optimization is an immense field of scientific and mathematical inquiry, and it is not the goal of this work to give an overview of this field in its entirety; other works, such as [34–37], provide thorough overviews of optimization and the use of optimization within engineering. Still, because an optimization-oriented view of engineering design, in general, and of wind turbine blade design, in particular, can be helpful, a relatively non-technical, introductory overview of optimization in the broader context of engineering design is given here.

The discipline of engineering design is vast and diverse. Distinct design methodologies can be identified throughout different industries and across various design scales and levels of complexity. Despite this, many engineering design endeavours can be understood, at various levels of abstraction, as optimization problems. Generally, an optimization problem can be defined

$$\begin{aligned}
 &\text{minimize} && \mathcal{J}(\boldsymbol{\chi}) \\
 &\text{subject to} && \boldsymbol{\chi} \in \boldsymbol{\Omega} , \\
 &&& \mathcal{B}_i(\boldsymbol{\chi}) = 0 , \quad i = 1, \dots, n_b , \\
 &&& \mathcal{C}_j(\boldsymbol{\chi}) \leq 0 , \quad j = 1, \dots, n_c .
 \end{aligned} \tag{2.1}$$

$\mathcal{J}(\boldsymbol{\chi})$ represents the “objective function,” “cost function,” or “fitness function,” and it indicates how well a particular design performs according to designer-defined criteria. $\mathcal{J}(\boldsymbol{\chi})$ could be defined, in some relatively straightforward and quantitative cases, as the total mass

or total material cost of an engineering design. $\mathcal{J}(\boldsymbol{\chi})$ may also represent more qualitative objectives related to, for example, aesthetics. $\mathcal{J}(\boldsymbol{\chi})$ might be defined at the component level, referring to the performance of a single instance of a device, or at any other level, such as component performance in a broader system or overall product performance in a market. Using a set of weighting variables, $\{w_1, w_2, \dots, w_n\}$, the relative importance of multiple objectives, $\{\mathcal{J}(\boldsymbol{\chi})_1, \mathcal{J}(\boldsymbol{\chi})_2, \dots, \mathcal{J}(\boldsymbol{\chi})_n\}$, in a single optimization problem can be indicated. For example, if linear combination is employed:

$$\mathcal{J}(\boldsymbol{\chi}) = w_1\mathcal{J}(\boldsymbol{\chi})_1 + w_2\mathcal{J}(\boldsymbol{\chi})_2 + \dots + w_n\mathcal{J}(\boldsymbol{\chi})_n . \quad (2.2)$$

The vector $\boldsymbol{\chi}$ indicates the “variables” or “design variables” of the optimization problem being considered and $\boldsymbol{\Omega}$ indicates their allowable ranges. The types of variables that $\boldsymbol{\chi}$ refers to also depends heavily on the optimization problem. $\boldsymbol{\chi}$ could contain variables related to material parameters, fundamental or abstract geometric properties, or even geographical locations. The n_b equality constraints, $\mathcal{B}_i(\boldsymbol{\chi})$, and n_c inequality constraints, $\mathcal{C}_j(\boldsymbol{\chi})$, indicate various criteria that must be met by the design. For example, an engineer may require that a design occupies a specific volume or outputs a particular electrical frequency, both of which would be equality constraints. Similarly, an engineer may require that a design does not exceed a particular operational temperature or that it is capable of supporting a predefined load, both of which are examples of inequality constraints.

The steps required to evaluate an objective function and its constraints or to find an optimal solution vary widely for different types of problems. In the simplest case of unconstrained optimization wherein $\mathcal{J}(\boldsymbol{\chi})$ is defined by a simple algebraic expression, evaluating $\mathcal{J}(\boldsymbol{\chi})$ is trivial and an optimal solution may be obtained using rudimentary calculus. In contrast, evaluation of $\mathcal{J}(\boldsymbol{\chi})$, $\mathcal{B}_i(\boldsymbol{\chi})$, and $\mathcal{C}_j(\boldsymbol{\chi})$ may involve complex mathematical or computational analyses spanning disciplines from structural mechanics to finance. In an even more abstract sense, $\mathcal{J}(\boldsymbol{\chi})$ may represent a comprehensive design process involving entire organizations, disparate analysis and design procedures, and varying degrees of human interaction. In some cases, engineering design is driven almost entirely by human intuition

and experience. However, due to growing computational resources and advanced analysis methodologies, it is increasingly feasible to rigorously optimize engineering designs using, for example, heuristic and metaheuristic techniques which repeatedly and intelligently evaluate $\mathcal{J}(\boldsymbol{\chi})$. While such techniques often cannot guarantee that a global optimum is found, they have, in many cases, been proven effective for addressing design problems of unprecedented size and complexity. Of course, the expense associated with evaluating $\mathcal{J}(\boldsymbol{\chi})$ strongly influences the degree to which an optimization problem can be solved.

2.1.2 Optimization of wind turbine blades

Wind turbine blades, as the part of a wind turbine responsible for converting the kinetic energy in the wind into rotor rotation, are the focus of much optimization effort in the wind energy industry. For the sake of simplicity and efficiency, these efforts typically utilize reduced-order analysis methods like blade element momentum (BEM) aerodynamic theory [4, 38] and a variety of structural beam models [39, 40] to predict overall performance. For example, Xudong et al. [12] use BEM theory and general structural dynamics to minimize the cost of energy associated with three wind turbine rotors ranging in capacity from 25 kW to 5 MW. Similarly, Fuglsang and Madsen [13] utilize a multidisciplinary approach in conjunction with reduced-order aeroelastic simulation tools to minimize cost of energy subject to fatigue and extreme load constraints. Lanzafame and Messina [14] focus on optimizing aerodynamic performance at low wind velocities, implementing a modified version of BEM theory and comparing analysis results with experimental data. Benini and Toffolo [15] use evolutionary algorithms to investigate the tradeoff between maximized annual energy production and minimized cost of energy, attending especially to the design implications for wind turbine blades. Additional factors may also be considered in such optimization problems, including geometric uncertainty [16], site-specific characteristics, and the design of other wind turbine components [17].

Despite the sophistication of many reduced-order-analysis-based optimization methods, the extent to which a wind turbine blade’s performance can be optimized depends, in part, on the accuracy and fidelity of the analysis methods used to evaluate its theoretical performance. Additionally, high-fidelity analysis, such as finite element analysis and computational fluid dynamics (CFD), is becoming increasingly accessible as computational resources continue to grow. These realities are fundamental drivers of research that seeks to employ high-fidelity-analysis-based design optimization to improve wind turbine rotor performance. For example, CFD can be used to optimize an entire blade geometry [41] or to parametrically optimize new design features such as winglets [42]. High-fidelity structural analysis can also be utilized to optimize the structural design of a blade. For example, iterative finite element analysis is useful for exploring different material stacking arrangements or for characterizing a limited parametric design space [8, 43]. Buckney et al. [44] employ more advanced methods, such as topology optimization, to identify alternative internal structural layouts. As with reduced-order approaches, it can be advantageous to simultaneously consider aerodynamic and structural properties when using high-fidelity approaches. Kenway et al. [45] use free-form volume deformation and analytic derivatives to perform gradient-based aerostructural optimization of a subsonic wing design, techniques that might be useful in the context of wind turbine blade design.

High-fidelity-analysis-based design of wind turbine blades is attractive due to the superior level of detail and accuracy that high-fidelity analysis provides. However, high-fidelity approaches are also relatively computationally expensive, a problem that is exacerbated when many hundreds of load cases or a large design space must be considered. Additionally, setting up high-fidelity analyses can consume significant resources due to mesh generation, large file sizes, multiple software platforms, and other practical considerations [46]. It is therefore most common for wind turbine blade designers to use reduced-order simulations for the majority of the blade design and optimization procedures, and to use a smaller number of high-fidelity analyses for final design validation. While this approach is

reasonable, it has the disadvantage that detailed, potentially design-driving analysis information is not considered until the later design stages. This is not only a potential hassle for blade designers, but it could also lead to sub-optimal designs. Thus, developments that improve the accuracy, efficiency, and practicality of high-fidelity analysis methods can have an important influence on wind turbine blade design.

2.2 NURBS-based isogeometric analysis

Traditional finite element methods rely on networks of interpolants to represent the solution fields of partial differential equations. The network, or mesh, and hence the analysis space, generally consists of nodes, at which the solution of interest is directly represented, and elements, over which the solution field is represented by specified functions. Thus, geometries in FEA are represented by finite element meshes comprised of, for example, linear triangular elements. Interestingly, as reviewed by Hughes et al. [19], the use of finite element analysis in engineering originated in the aerospace industry in the 1950s and 1960s, decades before the advent of CAD as it is known today. This perhaps explains the simple approach used for geometry modeling even in today’s finite-element-based computer-aided engineering (CAE) software.

Conversely, modern CAD software, which has its origins in the 1970s and 1980s, typically employs a variety of analytic or parametric functions, such as splines, for advanced geometry representation, visualization, and manipulation. This means that, in modern engineering workflows, geometries are originally constructed using CAD-based functions and are then converted into finite element meshes, which are less efficient geometrically, for use in CAE. Thus, for the engineering designer utilizing both CAD and CAE, unique geometric paradigms entail the management of separate, but necessarily related, geometry descriptions. This can lead to a significant amount of manual labor for the design engineer [46].

The core recognition of Hughes et al. [19] was that the parametric functions used by CAD systems to represent geometry can also be used to represent the solution fields of partial differential equations. For example, if an object is modeled geometrically using non-uniform rational B-splines (NURBS), which are employed in this work, the solution field can be readily represented using the exact same set of NURBS functions. In addition to the practical advantages suggested by this approach, such as only needing to maintain a single geometry description throughout design and analysis, NURBS basis functions have a number of properties making them useful for analysis, including partition of unity and non-negativity [19].

Because they are central to geometry modeling and to NURBS-based IGA in general, the mathematical background of NURBS functions is given here. NURBS are a generalization of B-splines, so B-splines are described first. Given a parametric variable u , a B-spline curve is defined by a set of non-interpolated control points \mathbf{P}_i where $i = 1, \dots, n$; degree, p ; and a knot vector, $U = [u_1, u_2, \dots, u_{n+p+1}]$, which is a set of parametric coordinates dividing the curve into segments over which distinct sets of basis functions have influence. The basis functions are defined by the Cox-deBoor recursion formula [47]. For degree $p = 0$,

$$N_{i,0}(u) = \begin{cases} 0 & \text{if } u_i \leq u < u_{i+1} , \\ 1 & \text{otherwise,} \end{cases} \quad (2.3)$$

and for degree $p \geq 1$,

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) . \quad (2.4)$$

A B-spline curve can be represented via linear combination of the n basis functions, $N_{i,p}$, and n control points, \mathbf{P}_i :

$$\mathbf{C}(u) = \sum_{i=1}^n N_{i,p}(u) \mathbf{P}_i . \quad (2.5)$$

For a net of $m \times n$ control points and with the introduction of a second parametric direction, v ; associated knot vector, V ; degree, q ; and set of basis functions, $N_{j,q}(v)$; a B-spline surface

can be constructed using the tensor product of the B-spline basis functions in the two parametric directions:

$$\mathbf{S}(u, v) = \sum_{i=1}^n \sum_{j=1}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} . \quad (2.6)$$

These B-spline basis functions can be further generalized to define NURBS entities. For each of the n control points, \mathbf{P}_i , in a curve, a weight, w_i , is assigned. These weights are used to describe the relative strength of influence of a basis function, in effect defining new, rational basis functions:

$$R_i^p(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^n N_{j,p}(u)w_j} . \quad (2.7)$$

A NURBS curve can therefore be defined as a linear combination of the n basis functions and n control points:

$$\mathbf{C}(u) = \sum_{i=0}^n R_i^p(u) \mathbf{P}_i . \quad (2.8)$$

A similar approach can be used to define bi-directional rational basis functions and NURBS surfaces:

$$R_{i,j}^{p,q}(u, v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u)N_{l,q}(v)w_{k,l}} , \quad (2.9)$$

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}^{p,q}(u, v) \mathbf{P}_{i,j} . \quad (2.10)$$

If $w_{i,j} = 1$ for all i and j , then $\mathbf{S}(u, v)$ becomes a B-spline surface.

Much of the utility of both surface modeling and IGA is a result of the advantageous characteristics of these functions; more of the mathematical details of NURBS are given by Piegl and Tiller [47] and Hughes et al. [19]. While NURBS are the primary geometrical framework employed in this work, numerous other options, each with distinct properties and intended use cases, exist.

IGA relies on the utilization of CAD-based functions to derive appropriate variational formulations for partial differential equations. For NURBS-based structural analyses, the degrees of freedom (DOF) are usually the displacement of the control points. As is the case in traditional FEA, the concept of “elements” is employed in IGA. In NURBS-based IGA, the spaces in between the knots in a valid knot vector are considered elements. For example, given a knot vector of $U = [0, 0, 0, 1, 2, 3, 3, 3]$, the parametric space $u = [0, 1]$ is considered an element. This definition is employed despite the fact that some of basis functions that are active in the space of this element remain active beyond $u = 1$, a key distinction from traditional finite elements. Also as in traditional FEA, mesh refinement is an important procedure in IGA, with both h -refinement, which corresponds to *knot insertion* of NURBS functions, and p -refinement, which corresponds to *degree elevation*, possible. k -refinement, which consists of p -refinement followed by h -refinement, is also possible in IGA and has some beneficial qualities. For more information, details about both IGA and the use of NURBS in the context of IGA are described by Hughes et al. [19].

2.3 Composite isogeometric Kirchhoff–Love shells

Since the inception of IGA, shell-based IGA has been an incredibly active research area. Bischoff et al. [48] provide a general review of shell theories and their formulations; the two most widely recognized and used shell theories are the Reissner–Mindlin and Kirchhoff–Love theories. The Reissner–Mindlin theory is perhaps the most widely-used formulation in traditional finite element settings, and IGA has been successfully used in Reissner–Mindlin analysis [49]. However, IGA has been particularly impactful in the realm of Kirchhoff–Love thin-shell analysis. Kirchhoff–Love theory is typically applied to thin-shell structures—indicated by $R/t \geq 20$, where R is the shell’s radius of curvature and t is its thickness [48]—and assumes that transverse shear strains are negligible, a reasonable assumption for many shell structures of interest. This assumption also indicates that no rotational DOF must be considered, improving analysis efficiency. Importantly, second-order derivatives

appear in the governing variational equations of the Kirchhoff–Love theory, implying the necessity of C^1 -continuous approximation functions. This condition has always been a major obstacle for the development of efficient finite element thin-shell formulations. However, in Kirchhoff–Love IGA that uses, for example, NURBS, the C^1 -smoothness requirement is naturally satisfied.

An isogeometric Kirchhoff–Love thin-shell formulation was first proposed by Kiendl et al. [50] and was reformulated for composite shells by Bazilevs et al. [23]. Additionally, the formulation was shown to accurately capture the kinematic behavior of wind turbine blades in [51] and [28]. The relevant details of the formulation in the context of the presented framework for IGA-based blade design and optimization are given here. The variational formulation is based on the principle of virtual work:

$$\delta W = \delta W_{\text{int}} + \delta W_{\text{ext}} = 0 , \quad (2.11)$$

where W is the total work, W_{int} is the internal work, W_{ext} is the external work, and δ denotes a variation with respect to the virtual displacement variables $\delta \mathbf{u}$:

$$\delta W = \frac{\partial W}{\partial \mathbf{u}} \delta \mathbf{u} . \quad (2.12)$$

The internal virtual work is given by

$$\delta W_{\text{int}} = \int_{\Omega} (\mathbf{S} : \delta \mathbf{E}) \, d\Omega , \quad (2.13)$$

where Ω is the shell volume in the undeformed configuration, \mathbf{S} is the second Piola–Kirchhoff stress tensor, and \mathbf{E} is the Green–Lagrange strain tensor.

The core tenet of shell theory is that a continuum, such as the volume Ω , can be sufficiently described using only the shell midsurface, hence neglecting transverse normal stress. Further, for Kirchhoff–Love theory, it is assumed that the shell director remains normal to its midsurface in all configurations, which implies that transverse shear strains are neglected. Thus, only in-plane stress and strain tensors are considered; here, Greek

indices $\alpha = 1, 2$ and $\beta = 1, 2$ denote their components. In this model, the Green–Lagrange strain tensor is separated into membrane strain, $\varepsilon_{\alpha\beta}$, and curvature change, $\kappa_{\alpha\beta}$:

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi_3 \kappa_{\alpha\beta} , \quad (2.14)$$

where $\xi_3 \in [-0.5h_{\text{th}}, 0.5h_{\text{th}}]$ is the through-thickness coordinate and h_{th} is the thickness of the shell. Let geometric variables indicated by $(\dot{\cdot})$ refer to the undeformed configuration. Membrane strain and curvature change are defined as

$$\varepsilon_{\alpha\beta} = \frac{1}{2} (a_{\alpha\beta} - \dot{a}_{\alpha\beta}) , \quad (2.15)$$

$$\kappa_{\alpha\beta} = \dot{b}_{\alpha\beta} - b_{\alpha\beta} , \quad (2.16)$$

where

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta = \mathbf{x}_{,\alpha} \cdot \mathbf{x}_{,\beta} , \quad (2.17)$$

$$b_{\alpha\beta} = \mathbf{a}_{\alpha,\beta} \cdot \mathbf{a}_3 . \quad (2.18)$$

In the above, \mathbf{a}_α are the covariant surface basis vectors in the deformed configurations, $\mathbf{x}(\xi_1, \xi_2)$ is the spatial coordinate of the midsurface in the deformed configuration with ξ_1 and ξ_2 being the parametric coordinates used in defining the midsurface, $(\cdot)_{,\alpha} = \partial(\cdot)/\partial\xi_\alpha$ indicates the partial derivatives with respect to ξ_α , and \mathbf{a}_3 is the unit vector normal to the shell midsurface in the deformed configuration, given by

$$\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\|\mathbf{a}_1 \times \mathbf{a}_2\|} . \quad (2.19)$$

Additionally, a local Cartesian basis can be defined:

$$\mathbf{e}_1 = \frac{\dot{\mathbf{a}}_1}{\|\dot{\mathbf{a}}_1\|} , \quad (2.20)$$

$$\mathbf{e}_2 = \frac{\dot{\mathbf{a}}_2 - (\dot{\mathbf{a}}_2 \cdot \mathbf{e}_1) \mathbf{e}_1}{\|\dot{\mathbf{a}}_2 - (\dot{\mathbf{a}}_2 \cdot \mathbf{e}_1) \mathbf{e}_1\|} . \quad (2.21)$$

With the strain tensor fully defined, the internal virtual work, Eq. (2.13), can again be considered. Assuming the St. Venant–Kirchhoff material law, the following stress–strain relationship is defined in the local Cartesian coordinate system:

$$\mathbf{S} = \mathbf{C}\mathbf{E} , \quad (2.22)$$

where \mathbf{C} is a constitutive material matrix. Introducing Eq. (2.22) into Eq. (2.13), separating out the through-thickness integration, and utilizing the definition of the strain tensor \mathbf{E} given in Eq. (2.14), one obtains

$$\begin{aligned} \delta W_{\text{int}} = & \int_{\mathcal{S}} \delta \boldsymbol{\varepsilon} \cdot \left(\left(\int_{h_{\text{th}}} \mathbf{C} d\xi_3 \right) \boldsymbol{\varepsilon} + \left(\int_{h_{\text{th}}} \xi_3 \mathbf{C} d\xi_3 \right) \boldsymbol{\kappa} \right) d\mathcal{S} \\ & + \int_{\mathcal{S}} \delta \boldsymbol{\kappa} \cdot \left(\left(\int_{h_{\text{th}}} \xi_3 \mathbf{C} d\xi_3 \right) \boldsymbol{\varepsilon} + \left(\int_{h_{\text{th}}} \xi_3^2 \mathbf{C} d\xi_3 \right) \boldsymbol{\kappa} \right) d\mathcal{S} , \end{aligned} \quad (2.23)$$

where \mathcal{S} is the shell surface domain and h_{th} is the shell thickness domain. Eq. (2.23) is expressed in the local Cartesian coordinate system.

For a general orthotropic material, the constitutive material matrix is defined as

$$\mathbf{C}_{\text{ort}} = \begin{bmatrix} \frac{E_1}{1 - \nu_{12}\nu_{21}} & \frac{\nu_{21}E_1}{1 - \nu_{12}\nu_{21}} & 0 \\ \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}} & \frac{E_2}{1 - \nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} , \quad (2.24)$$

where E_1 and E_2 are the Young’s moduli in the directions defined by the local material coordinate system, ν_{12} and ν_{21} are the Poisson’s ratios, $\nu_{21}E_1 = \nu_{12}E_2$ to ensure the symmetry of the constitutive matrix, and G_{12} is the shear modulus. In the case of composite materials, which are especially important to consider for wind turbine blade modeling, the structure is assumed to be composed of a number of orthotropic plies. Each ply, denoted by the index k , can be oriented at an angle, ϕ_k , with respect to the local Cartesian coordi-

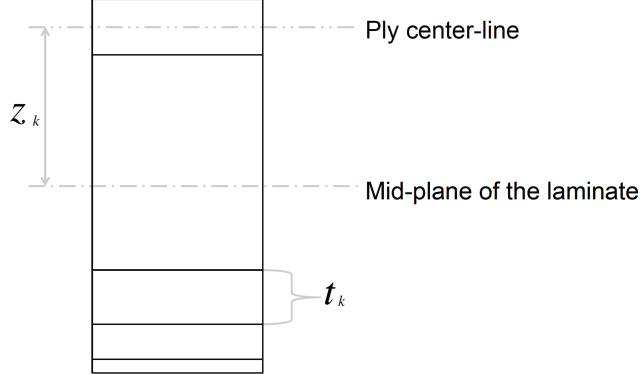


Figure 2.1: Composite layup with non-uniform and nonsymmetric ply distribution [51, Figure 3].

nates. Transformation of Eq. (2.24) from the material coordinates of each ply to the local Cartesian coordinates is reflected in the expressions

$$\mathbf{C}_k = \mathbf{T}^T(\phi_k) \mathbf{C}_{\text{ort}} \mathbf{T}(\phi_k) , \quad (2.25)$$

$$\mathbf{T}(\phi) = \begin{bmatrix} \cos^2 \phi & \sin^2 \phi & \sin \phi \cos \phi \\ \sin^2 \phi & \cos^2 \phi & -\sin \phi \cos \phi \\ -2 \sin \phi \cos \phi & 2 \sin \phi \cos \phi & \cos^2 \phi - \sin^2 \phi \end{bmatrix} . \quad (2.26)$$

If the thickness of the k th ply is denoted by t_k and its centroid with respect to the mid-plane of the laminate is given by z_k , as illustrated in Figure 2.1, then, according to classical lamination theory [52], the homogenized extensional (membrane) stiffness, \mathbf{K}^{exte} ; coupling stiffness, \mathbf{K}^{coup} ; and bending stiffness, \mathbf{K}^{bend} , are given by

$$\mathbf{K}^{\text{exte}} = \int_{h_{\text{th}}} \mathbf{C} d\xi_3 = \sum_{k=1}^n \mathbf{C}_k t_k , \quad (2.27)$$

$$\mathbf{K}^{\text{coup}} = \int_{h_{\text{th}}} \xi_3 \mathbf{C} d\xi_3 = \sum_{k=1}^n \mathbf{C}_k t_k z_k , \quad (2.28)$$

$$\mathbf{K}^{\text{bend}} = \int_{h_{\text{th}}} \xi_3^2 \mathbf{C} d\xi_3 = \sum_{k=1}^n \mathbf{C}_k \left(t_k z_k^2 + \frac{t_k^3}{12} \right) . \quad (2.29)$$

Note the correspondence of Eqs. (2.27)–(2.29) to the various terms in Eq. (2.23). The internal virtual work for a composite shell can therefore be defined:

$$\delta W_{\text{int}} = \int_S \delta \boldsymbol{\varepsilon} \cdot \left(\mathbf{K}^{\text{exte}} \boldsymbol{\varepsilon} + \mathbf{K}^{\text{coup}} \boldsymbol{\kappa} \right) dS + \int_S \delta \boldsymbol{\kappa} \cdot \left(\mathbf{K}^{\text{coup}} \boldsymbol{\varepsilon} + \mathbf{K}^{\text{bend}} \boldsymbol{\kappa} \right) dS . \quad (2.30)$$

More details about this formulation, including linearization of the nonlinear equation system, can be found in Kiendl et al. [50].

2.4 Acknowledgements

Chapter 2 is, in part, a reprint of material as it appears in: “A framework for isogeometric-analysis-based design and optimization of wind turbine blade structures,” (with J. Kiendl and M.-C. Hsu), *Wind Energy*, in review. Chapter 2 is also, in part, a reprint of material as it appears in: “A patch coupling technique for non-matching isogeometric shells with application to composite wind turbine blades,” (with E.L. Johnson, D. Proserpio, J. Kiendl, and M.-C. Hsu), *Computational Mechanics*, in preparation. The dissertation author was the primary investigator of both of these papers.

CHAPTER 3. PARAMETRIC DESIGN OPTIMIZATION USING IGA

3.1 Chapter overview

Although NURBS curves and surfaces are technically defined by fundamental properties such as control points, as described in Section 2.2, sophisticated engineering designs, such as wind turbine blades, are more commonly defined using abstract parametric geometry descriptions. For example, in the case of a wind turbine blade, cross-section-specific input parameters such as chord, twist, and pitch axis define the blade’s planform. However, few frameworks exist that feature a seamless connection between CAD and IGA, limiting the extent to which CAD can be employed in IGA-based design and optimization. Chapter 3 therefore outlines a framework which enables seamless IGA-based optimization of parametrically-defined structures within a CAD software platform.

Chapter 3 is outlined as follows. In Section 3.2, the background of the use of IGA in the context of design-through-analysis and design optimization is given. In Section 3.3, the structure of the presented isogeometric design optimization framework is described and the salient features of the parametric design, analysis, visualization, and optimization procedures are highlighted. Specifically, the role that each procedure plays in enabling the overall, novel approach to design optimization is emphasized. In Section 3.4, the capabilities of the framework are demonstrated and validated by first optimizing a simple tube structure. A wind turbine blade design is then optimized in Section 3.5 to demonstrate the framework’s ability to promote analysis-driven design of realistic, industrial-scale engineering solutions. In Section 3.6, concluding remarks are given.

3.2 IGA and design-through-analysis

One of the originally identified advantages of IGA [19] is that it enables tight integration of high-fidelity FEA into the engineering design workflow. In traditional design-and-analysis workflows, approximately 80% of the overall design lifecycle is devoted to finite element mesh generation and creation of analysis-suitable models; only 20% of the remaining lifecycle is spent on actually performing analysis [46]. In the context of iterative design optimization, such inefficiency is amplified.

Because it allows a consistent geometry description to be used for both design and analysis, IGA theoretically allows designers to analyze, redesign, and optimize mechanical components easily and quickly, especially when used in lieu of traditional analysis methods. However, the efficient creation of analysis-suitable models of realistic engineering designs remains a barrier even to design lifecycles that make use of IGA. This is mainly due to the significant restrictions many isogeometric methodologies place on the construction of analysis-suitable CAD models. Thus, while the intent of IGA is to improve design lifecycle efficiency, the tedious process of mesh generation is often merely replaced with another, similarly tedious process of generating analysis-suitable models. Hence, further exploration of the design paradigms surrounding model generation is required to truly translate the benefits of IGA to the engineering world.

Previous works have sought to improve design-and-analysis frameworks by incorporating IGA. An isogeometric design-through-analysis concept was previously explored in Schillinger et al. [53] based on hierarchical refinement of NURBS and T-splines [46, 54] using the finite cell method [55, 56]. Breitenberger et al. [57] presented an Analysis in Computer-Aided Design (AiCAD) concept using NURBS-based boundary representation (B-rep) models for nonlinear isogeometric shell analysis, which included enhancements such as the ability to perform analysis on trimmed surfaces and the use of the penalty method. The AiCAD concept was implemented in CAD software packages such as Rhino [58] and Siemens NX [59]. Additionally, Hsu et al. [30] developed a user-interface-based parametric design platform for

IGA directly within the Rhino CAD environment, utilizing Rhino’s algorithmic modeling interface, Grasshopper [60], for parametric geometry generation.

The tight coupling between geometry and analysis also allows IGA to be naturally integrated with shape optimization. Wall et al. [61] and Fußeder et al. [62] presented frameworks for structural shape optimization of basic two-dimensional geometries using isogeometric structural analysis and gradient-driven optimization methods. Moysidis and Koumoussis [63] performed shape optimization of plane stress structures in the context of a hysteric formulation for IGA. Julisson et al. [64] used IGA and Powell’s derivative-free optimization algorithm to perform structural shape optimization of three-dimensional thin-shell structures. Cho and Ha [65], Qian [66], and Kiendl et al. [67] used shape sensitivity analysis to recover optimal shapes through structural analysis, the latter doing so in three dimensions. In each of these cases, the locations of designated control points—and control point weights, in some cases—of interest were used as the design variables. Additionally, isogeometric shape optimization has been used to address problems of electromagnetic scattering [68], vibrating membranes [69], heat conduction [70], fluid mechanics [71], and the design of magnetic actuators [72]. A notable departure from the optimization of control point weights and locations is found in Kostas et al. [73] in which geometry parameterization is a primary focus and in which an IGA-based boundary element method provides the basis for optimizing the wave resistance of a T-spline ship hull.

While important work has been done in both uniting IGA with CAD software platforms and recognizing the natural ability of IGA to facilitate shape optimization, additional work must be done to demonstrate that IGA-based optimization is relevant and beneficial in the context of modern CAD paradigms. For example, modern CAD systems, such as SolidWorks [74], Pro-Engineer [75], etc., use feature-based modeling¹ [78] to capture design intent [76]. In addition, they support parametric design modifications

¹Please refer to the book by Shah and Mäntylä [76] and the review by Salomons et al. [77] for a detailed background on feature-based CAD. A feature tree is used to keep track of the different features that are used to model the geometry. A feature can give rise to new geometric entities, such as faces, which can then be used by a child feature in subsequent modeling steps.

using a constraint-based solver [79–82] and solid modeling [83]. On the other hand, many isogeometric design-through-analysis frameworks are based on Rhino, a freeform surface modeling system, mainly because Rhino uses a NURBS-based geometry kernel. Rhino does not natively facilitate parametric constraint- or feature-based modeling. This makes editing engineering models in Rhino difficult since model parameters cannot be simply changed to the desired value. The lack of feature-based modeling also implies that any change to a particular surface can lead to a geometry configuration that is inconsistent with the original design intent (e.g., introducing non-manifold geometry in a solid model, self-intersecting geometry, or new gaps). Hence, one may have to perform changes to multiple surfaces, even when the desire is to change a single feature, in order to generate an analysis-suitable design. Finally, modern CAD platforms prioritize interactivity throughout the design process [84]. Thus, a parametric optimization workflow that uses design-specific, syntax-heavy, compilation-dependent code, while technically parametric, is not sufficiently interactive for extensive use in many engineering design contexts.

Another reality of large-scale engineering design contexts is that analyses from multiple disciplines must often be performed in order to quantify the effectiveness of a particular design, and the parametric inputs for these complex designs may be more abstract than fundamental geometric parameters such as control point locations (e.g., constraint-based dimensions or material parameters). Martins and Lambe [36] surveyed various methods encountered within the field of multidisciplinary design optimization, a field of research that studies the application of numerical optimization techniques to the design of engineering systems. Multidisciplinary design optimization is commonly used to address engineering design problems (see, e.g., [85–89]); some such problems use traditional FEA and are hence forced to script mesh generation procedures and to manage separate geometry descriptions.

The central goal of this chapter, therefore, is to develop a natural computational framework that is capable of IGA-based parametric design optimization in an interactive, multidisciplinary design context. In addition to facilitating high-level parameterization and

interactivity, the framework should be designed around the notion of consistently creating families of analysis-suitable models. The methodology suggested in this work is demonstrated using the Grasshopper algorithmic modeling interface to promote workflow consistency, efficiency, interactivity, and cost-effectiveness—in terms of both time and money—of the engineering design process. The Rhino-based Grasshopper can access Rhino’s NURBS-based geometry kernel to construct analysis-suitable models. The framework is also designed to enable truly seamless, heuristic design optimization based on IGA results. To the best of the author’s knowledge, it is the first computational framework capable of performing IGA-based parametric design optimization of engineering structures that are practically modeled through the use of complex, CAD-based geometric operations.

3.3 Isogeometric design optimization framework

3.3.1 Overview

An interactive parametric design and geometry modeling platform was proposed by Hsu et al. [30] to directly employ IGA within the Rhino CAD environment. Hsu et al. [30] used a traditional approach to the model generation and analysis workflow in that the platform was constructed with the intent of a user interacting with the model within the Rhino viewport and invoking design, analysis, and post-processing procedures via a user interface. This methodology is not suitable for rigorous design optimization, in part because it was formulated with the intent of strong user interaction.

Alternatively, this work demonstrates the development of a computational framework capable of performing IGA-based parametric design optimization of realistic engineering structures that are practically constructed through the use of complex, CAD-based geometric operations. The general structure of a simple design optimization procedure can be seen in Figure 3.1, which depicts the cycle of parametric model construction and analysis. Such cycles are commonplace in the engineering design world, demonstrated by the popularity of software platforms like ANSYS Workbench [90]. However, the use of specific

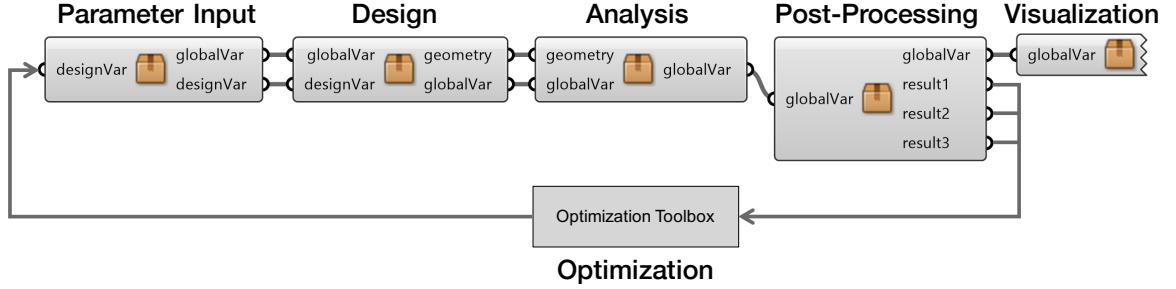


Figure 3.1: Overall structure of the isogeometric design optimization framework laid out in Grasshopper, an algorithmic modeling interface for Rhino. Optimization procedures can be performed either inside or outside of the Grasshopper environment using various optimization toolboxes, such as MATLAB [91], Dakota [92], Galapagos [60], etc.

strategies within the modeling and analysis stages of the design cycle, as proposed in this work and demonstrated below, allows seamless, IGA-based design optimization. Not only is this seamless approach practical and efficient, but it also inherits the characteristics of IGA that are particularly advantageous in optimization settings (see Section 3.3.4.1 for more details).

The framework as implemented in this work and as shown in Figure 3.1 exists primarily within Grasshopper [60], an algorithmic modeling interface which makes use of and controls the CAD software called Rhino [58]. Rhino uses NURBS-based surface geometry descriptions and features a variety of numerically robust and efficient algorithms for creating and modifying NURBS geometry. Having access to this advanced geometric functionality is invaluable to engineering designers who rely heavily on complex, pre-defined algorithms. In the overarching context of the isogeometric design optimization framework, Grasshopper is used to create or integrate parametric design algorithms, analysis codes, post-processing operations, optimization toolboxes, and result visualization. Because Rhino and Grasshopper have distinct modeling approaches and interfaces, they are, in theory, different software platforms. However, Grasshopper is, in fact, a plugin for Rhino that uses Rhino’s geometry kernel and visualization capabilities. Hence, Rhino and Grasshopper are tightly integrated

and the connection between them is automatically established through the installation of Grasshopper.

Grasshopper features many “components,” which are visualized in the two-dimensional Grasshopper workspace as small rectangles, each with unique geometric or programmatic functions. The user inserts the desired components and links the functions’ inputs and outputs together via graphical “wires.” If no Grasshopper component contains the exact functionality desired by the user, custom scripting components, available in a variety of programming languages and capable of accessing Rhino’s core functionality, can be created. Groups of functions can be also packaged into “clusters” which then appear as a single component in the Grasshopper workspace. The clusters responsible for the design, analysis, post-processing, and other operations are shown within the Grasshopper interface in Figure 3.1.

Sections 3.3.2, 3.3.3, and 3.3.4 will detail the contents of the design, analysis, and optimization clusters, respectively, shown in Figure 3.1. Visualization of analysis results may not be required within the design optimization loop. However, it may be beneficial to visualize the results during or after the optimization process. The visualization methodology, which is unique for IGA and is implemented within the “Visualization” cluster in Figure 3.1, is therefore outlined in Section 3.3.5.

3.3.2 Design

The contents of the “Design” cluster in Figure 3.1 are necessarily distinct for unique design optimization applications. The elements of engineering model design that are important to consider when constructing an IGA framework intended to optimize realistic engineering designs are discussed. A few characteristics, such as parametric design, maintenance of valid analysis-suitable geometries, and interactivity, are considered indispensable for efficient model development.

3.3.2.1 Parametric model construction

The ability to establish direct parametric control of geometry is a necessity for engineering designers. Most CAD software packages, such as SolidWorks [74], employ constraint-based systems [93–95] that allow designers to directly alter model-defining dimensions, such as line length or arc radius, and geometric constraints like straightness or tangency. Changing any of these values causes the position and size of the relevant geometric entities to be automatically recalculated such that all user-defined constraints and dimensions are satisfied. This simplifies model construction and makes it easier to build design intent into a model, especially when the model is based on engineering drawings which use relative dimensioning almost exclusively.

Rhino is often used to generate models for IGA because it is built upon a NURBS-based geometry kernel and has been used within the IGA community in the past. However, Rhino does not natively feature constraint-based design capabilities, forcing the designer to painstakingly calculate the absolute position or size of geometric entities. Subsequent model adjustment must be performed in a similar fashion. This hinders the extent to which a designer can change an engineering design based on analysis results, a fundamental goal of IGA.

Another problem with using a freeform modeling system like Rhino in an iterative IGA context is that, due to the lack of constraint-based modeling capabilities, editing individual surfaces may lead to a model that is inconsistent with the original design intent. This behavior is demonstrated in Figure 3.2, where an initial model, on the left, is comprised of surfaces 1 and 2, where surface 2 is generated based on the location of the lower edge of surface 1. Using a freeform modeling system, if the designer were to change the radius of surface 1, then surface 2 would not be inherently regenerated accordingly. The result would be the top configuration on the right side of Figure 3.2, where the edges of the surfaces are no longer coincident.

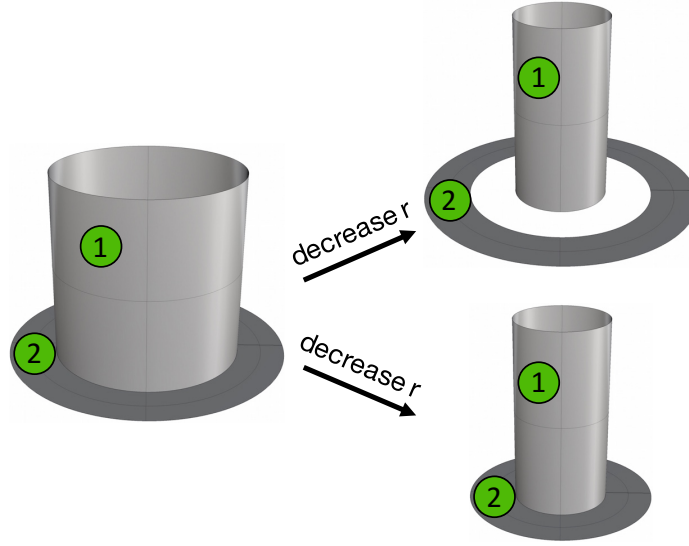


Figure 3.2: Illustration of the different designs achieved when design intent (geometric connection of surfaces 1 and 2) is maintained (bottom case) versus when design intent is not respected (top case) for a given model change.

A modeling platform is therefore required which is capable of both parametric modeling and consistent generation of geometry that aligns with the original design intent. In the present framework, the Grasshopper interface for Rhino is utilized to achieve this goal. Grasshopper allows the designer to graphically develop a procedural algorithm [96] to create a model using interrelated geometric functions. The algorithms in the presented framework are developed such that the desired inputs are the engineering parameters of interest, effectively establishing direct parametric control of the NURBS objects within Rhino. Thus, procedural model generation gives the designer parametric control over the model without explicitly developing a fully developed, constraint-based modeling system. The use of procedural generation also entails the capability to more reliably generate multiple geometries that maintain design intent. For the example in Figure 3.2, one can construct an algorithm wherein the radius of surface 1 is a parametric input and surface 2 is generated based on the lower edge of surface 1. Then, if the designer were to change the radius of surface 1, the entire algorithm would recompute, producing the configuration on the bottom-right side of Figure 3.2. Thus, the design remains valid for a wide range of parametric input.

The notion of parametric procedural design can be abstracted using an expression $\Theta(\boldsymbol{\chi})$ in which $\boldsymbol{\chi}$ is a vector of parametric design variables and $\Theta(\boldsymbol{\chi})$ is an algorithm² that generates the design model based on given design variables. In this sense, the general expression $\Theta(\boldsymbol{\chi})$ might be thought of as a means of generating a “family of designs.” Some common design variables, especially in the context of IGA, are the NURBS control point locations and weights; an example design vector $\boldsymbol{\chi}^e$ can thus be defined as

$$\boldsymbol{\chi}^e = \{\mathbf{P}_k, W_k\}, \quad k = 1, 2, \dots, n, \quad (3.1)$$

where \mathbf{P} and W denote control point location and weight, respectively, boldface indicates a spatially dimensioned vector, and n is the total number of control points. One can then establish an algorithm, $\Theta^e(\boldsymbol{\chi}^e)$, for a particular design application. Of course, for the purposes of communication and efficiency, most realistic engineering designs require the establishment of high-level design parameters, such as component width or relative feature location.

Different sets of design parameters and generative algorithms correspond to different design scenarios. For example, the family of tube designs investigated in Section 3.4 and the Grasshopper algorithm $\Theta^t(\boldsymbol{\chi}^t)$ for generating these designs are shown in Figure 3.3. Additionally, the family of wind turbine blade designs investigated in Section 3.5 and the Grasshopper algorithm $\Theta^b(\boldsymbol{\chi}^b)$ for generating those designs are shown in Figure 3.4. More information on the design variables used in these cases can be found in the corresponding example sections.

3.3.2.2 Interactivity

Interactivity is important in the engineering design process not only because it improves the designer’s aesthetic experience, but also because immediate visual feedback and intuitive interfaces improve the efficiency of the design process. This is yet another reason why

²In this context, the word “algorithm” is used to refer to a set of generative or manipulative geometric or otherwise programmatic functions which, when executed in sequence, procedurally generate a particular engineering design model.

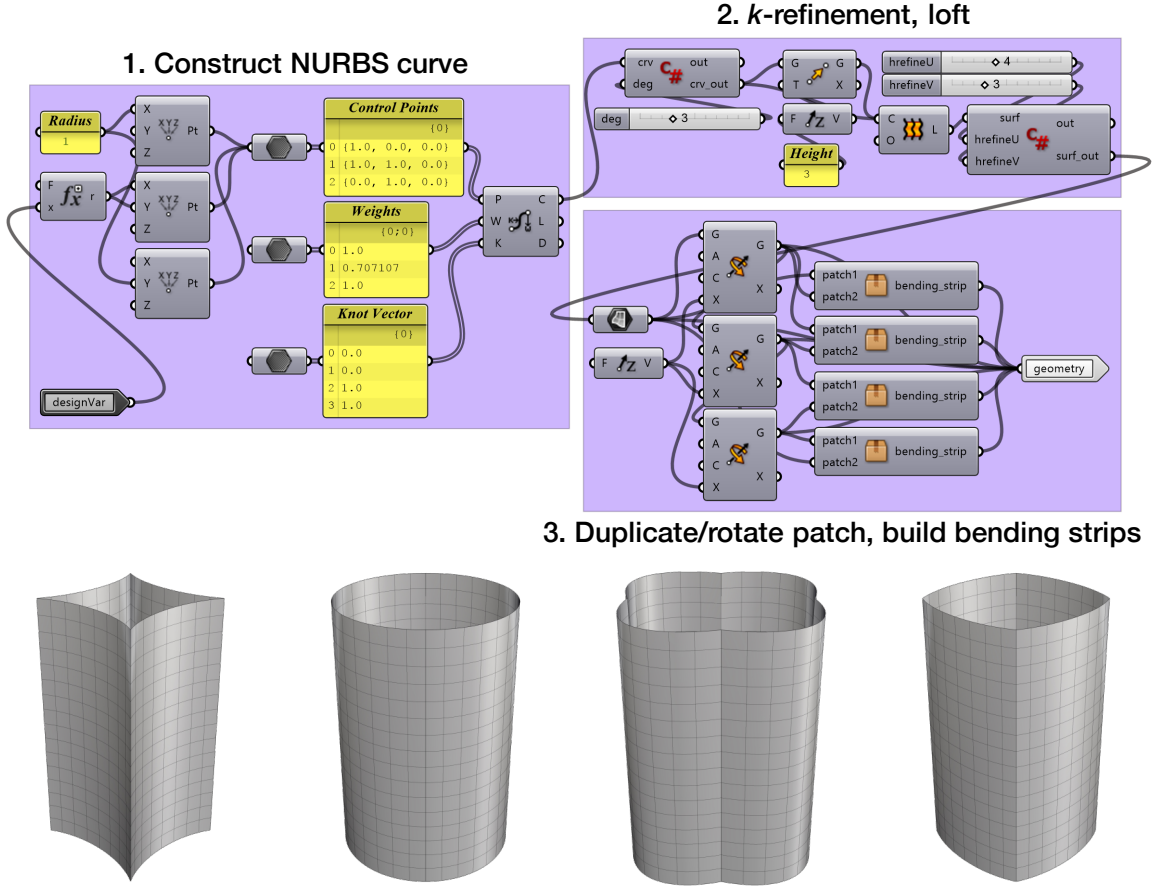


Figure 3.3: Family of tube designs (bottom) and associated Grasshopper generative algorithm $\Theta^t(\chi^t)$ (top).

the computational framework is demonstrated in Grasshopper. In contrast to the code-based, syntax-heavy parametric design approach used in some other works, the visually programmed generative algorithms in Grasshopper are simple to edit, provide immediate visual feedback in three-dimensional space, and do not require compilation.

This point may seem trivial, but it is important to consider if one intends to abide by the original spirit of IGA. In essence, IGA and the notion of interactivity in the design context serve the same purpose: to improve the quality of design feedback and to deliver such feedback efficiently and elegantly. Thus, focusing on IGA without considering the interactive design context may not result in a net improvement in overall design-and-analysis workflow.

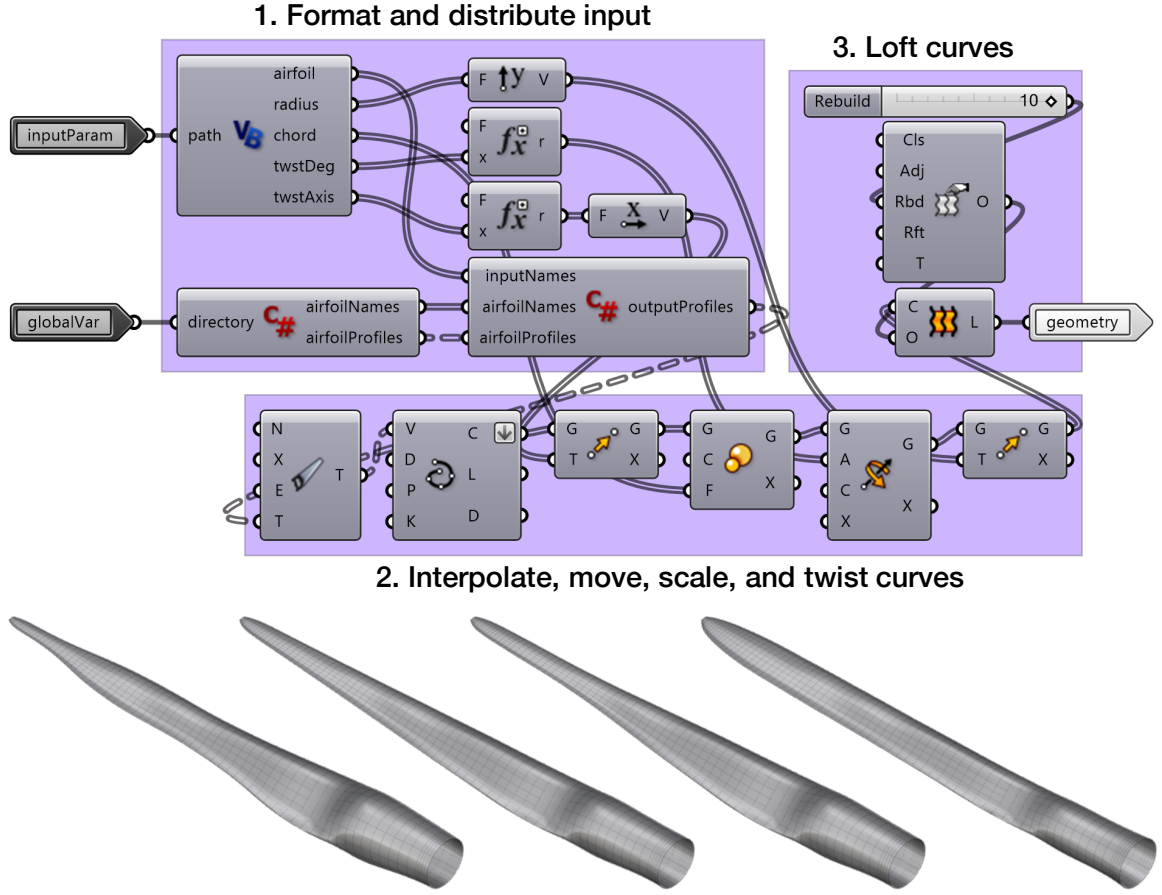


Figure 3.4: Family of blade designs (bottom) and associated Grasshopper generative algorithm $\Theta^b(\chi^b)$ (top).

3.3.3 Analysis

Having established the utilization of a platform that is both NURBS-based and facilitates efficient, parametric model design, IGA is incorporated into the overall workflow as indicated by the “Analysis” cluster in Figure 3.1. Assuming a model’s NURBS information, such as control point locations, degree, and knot vectors, is immediately available at the analysis stage, as is the case in Grasshopper, and assuming that the geometry is analysis-suitable, IGA can be readily performed. The overall procedure in Grasshopper recognizes new parametric input, constructs the model according to a parametric algo-

rithm, outputs the relevant NURBS information, and automatically calls an analysis code through a customizable C# scripting component. Compared to the overall analysis time, the computational cost of these input and output procedures is insignificant, as illustrated in Sections 3.4 and 3.5.

The rotation-free Kirchhoff–Love thin-shell variational formulation is utilized for both the pressurized tube and the wind turbine blade applications in Sections 3.4 and 3.5. Thus, the internal work formulation presented in Eq. (2.30) is employed in the dynamic setting (both static and dynamic forms are employed in this chapter). The formulation is further augmented to handle regions where the mapping reduces to the C^0 level using the bending strip approach [97]. The formulation may be stated as: find the displacement of the shell midsurface $\mathbf{u} \in \mathcal{V}_u$ such that for all test functions $\mathbf{w} \in \mathcal{W}_u$,

$$\begin{aligned} & \int_{\mathcal{S}} \mathbf{w} \cdot h_{\text{th}} \rho_0 \left(\frac{d^2 \mathbf{u}}{dt^2} - \mathbf{f} \right) d\mathcal{S} + \int_{\mathcal{S}} \delta \boldsymbol{\varepsilon} \cdot \left(\mathbf{K}^{\text{exte}} \boldsymbol{\varepsilon} + \mathbf{K}^{\text{coup}} \boldsymbol{\kappa} \right) d\mathcal{S} \\ & + \int_{\mathcal{S}} \delta \boldsymbol{\kappa} \cdot \left(\mathbf{K}^{\text{coup}} \boldsymbol{\varepsilon} + \mathbf{K}^{\text{bend}} \boldsymbol{\kappa} \right) d\mathcal{S} + \int_{\mathcal{S}^b} \delta \boldsymbol{\kappa} \cdot \mathbf{K}^{\text{best}} \boldsymbol{\kappa} d\mathcal{S} - \int_{\mathcal{S}^h} \mathbf{w} \cdot \mathbf{h} d\mathcal{S} = 0, \end{aligned} \quad (3.2)$$

where \mathcal{V}_u and \mathcal{W}_u denote the trial and test function spaces, respectively, for the structural mechanics problem; \mathcal{S} and \mathcal{S}^b denote the shell midsurface and bending strip domains in the reference configuration, respectively; ρ_0 is the through-thickness-averaged shell density; \mathbf{h} is the prescribed traction on \mathcal{S}^h ; \mathbf{f} denotes body forces per unit mass; and \mathbf{K}^{best} is the bending stiffness of the bending strips. The bending stiffness of the bending strips is defined

$$\mathbf{K}^{\text{best}} = \frac{h_{\text{th}}^3}{12} \begin{bmatrix} E_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.3)$$

where E_s is sufficiently high to penalize the change in angle between triples of control points at the patch interface during deformation. For more details, please see [97, 98]. Eq. (3.2), the weak form of the system of partial differential equations (PDEs) that describes the physics, is denoted as $\mathcal{B}(\mathbf{u}) = 0$.

Importantly, it would be entirely possible to replace the Kirchhoff–Love shell variational formulation used here with many other isogeometric methods, such as other shell formulations [49, 99], boundary element methods [100, 101], and finite cell [53, 55] or immersogeometric techniques [102, 103]. Much of the work in achieving such implementations would consist merely of ensuring that the IGA solver can recognize new geometries and communicate analysis results to the Grasshopper environment. It is also feasible to perform isogeometric analysis on solid volumetric geometries as was done with a gas turbine modeled using trivariate NURBS in [30]. Because Grasshopper and Rhino do not support trivariate splines natively, this is achieved through unique surface construction techniques and pre-processing that builds volumetric models based on a network of surfaces. This represents a rich and fruitful avenue of future research and development.

3.3.4 Optimization

The “Optimization” cluster in Figure 3.1 indicates the use of an optimization toolbox to drive the iterative design-and-analysis process. It was previously discussed that, although the fundamental integration of CAD and CAE paradigms through the use of IGA theoretically enables a more iterative approach to engineering design and analysis, practical limitations have hindered the establishment of IGA-based design-and-analysis workflows. These workflow limitations have led to limitations in parametric design optimization. One of the key goals of creating a parameterized model within a design-through-analysis framework is to allow the designer to understand the influence of relevant design parameters on values of interest that are obtained through computational analysis. This can be done, and is often still done, manually, especially in the context of high-fidelity structural analysis; the designer performs analysis, views the result, adjusts the design, and repeats as necessary until the desired result is achieved. However, if the design-and-analysis workflow is made completely seamless using IGA and parametric design techniques, as is the case with

the presented framework, computational power can be further leveraged using automated optimization methods.

In this work, MATLAB is integrated into the design-and-analysis framework, allowing the use of the many optimization techniques incorporated into MATLAB’s optimization toolbox [91]. Externally routing the design pipeline through MATLAB is acceptable in this implementation because relatively little information (in the presented examples, only design variables and objective function values) must be transferred. However, depending on the volume of transferred information and other performance requirements, alternative optimization techniques and packages, including techniques native to Grasshopper, could also be used. In the present implementation, MATLAB provides input parameters, allows the design-through-analysis framework to build a model and perform analysis, and then retrieves relevant output values from Grasshopper to inform future iterations. This process is entirely automated and enables the optimization algorithm to search within a parameterized family of designs, $\Theta(\chi)$, freely. Importantly, both local (gradient-based and gradient-free) and global (meta-heuristic and multi-start) optimization methods can be seamlessly applied within this paradigm.

3.3.4.1 Advantages of IGA in optimization setting

As discussed previously, one traditional barrier to analysis automation in the finite element context is the difficulty associated with generating finite element meshes for complex geometries. Ensuring that quality meshes can be generated automatically from CAD models remains a challenging task, often requiring manual intervention and thus reducing the overall efficiency of the optimization framework. A key benefit of a design optimization framework that makes use of IGA is that such mesh generation can be avoided, assuming the generative algorithm is designed carefully such that analysis suitability is ensured. The geometry can then be directly referenced for analysis, reducing the number of required pre-analysis tasks and easing setup of the overall optimization problem. Additionally, CAD-based geometry

models tend to be composed of considerably lower volumes of data relative to their CAE mesh counterparts.

Isogeometric analysis may also reduce the computational time required for accurate analysis of a given design. This benefit is especially important in the context of design optimization, where the reduction of a single function evaluation by a minute can translate to hours of saved optimization time. Compared to traditional finite element methods, isogeometric analysis is capable of more quickly producing results of equivalent accuracy. Benson et al. [20] demonstrated that structural analysis of a roof using 450 quadratic NURBS elements could produce results in 2.90 CPU seconds that are approximately the same as those produced by an analysis using 4,512 linear Belytschko-Tsay elements, requiring 10.5 CPU seconds. It is therefore apparent that IGA is a uniquely apt tool in the context of design optimization, where limiting analysis time—without unnecessarily sacrificing analysis accuracy—is critical.

3.3.5 Visualization

Visualizing analysis results throughout optimization can provide valuable feedback and can help the designer to understand the progression of the optimization procedure. This is especially relevant if a solution field, such as maximum in-plane strain, is used as an optimization constraint or objective. After performing the IGA simulation, the control variables (or degrees of freedom) for the solution fields (e.g., displacement, velocity, temperature, etc.) are defined on the control points, which are typically not located on the physical geometry. These solution variables need to be coupled with basis functions to generate continuous solution fields that can be mapped to the physical geometry. In this work, a Grasshopper-generated visualization mesh is used to map the solution fields. More sophisticated visualization techniques, such as direct volume rendering [104, 105], isosurface mesh extraction [106, 107], and direct rendering of isosurfaces [108–110] have been developed for visualizing volumetric IGA results.

An approximate, mesh-based methodology for visualizing IGA results within the Rhino viewport was proposed in Hsu et al. [30]. A visualization mesh is constructed and the coordinates of the mesh points are fed to a Grasshopper component that finds the closest corresponding points on the NURBS surface and returns the parametric coordinates. Along with the control variables and basis function information, the solution values are evaluated on the NURBS surfaces using an in-house code and are then transferred back to the visualization mesh points.

Here, an entirely Grasshopper-based implementation of this approach is proposed in which a “solution surface,” defined as a NURBS surface, is constructed within Rhino’s geometry kernel and is then evaluated at the mesh points. This is possible because, in this work, analysis is performed only on shell structures. Grasshopper natively features visualization meshes for the display of color contours on geometries. A relatively dense visualization mesh can easily be generated for virtually any geometry with mesh points defined by parametric (u, v) coordinates. A color can then be assigned to each mesh point. Thus, a particular solution parameter, such as maximum in-plane strain, must be defined at each mesh point location defined by parametric (u, v) coordinates. Note that, here, the solution coefficients which are defined on each control point are denoted $\mathbf{Q}_{i,j}$, where i and j correspond to the index of the control point in the u and v directions, respectively.

The definition of a NURBS surface of degree p in the u direction and degree q in the v direction is given by Eq. (2.10). Rhino’s C# programming library [111], which can be referenced by the C# scripting components in Grasshopper, contains a function for the construction of a NURBS surface object, $\mathbf{S}(u, v)$, given the constituents of the basis function $R_{i,j}^{p,q}(u, v)$ and the control points $\mathbf{P}_{i,j}$. In the context of solution visualization, the solution, rather than the geometry’s physical location, must be evaluated at the given mesh points.

Thus, these same C# functions can be used to construct *not* a physical surface, $\mathbf{S}(u, v)$, but rather a mapping of the solution variable:

$$\mathbf{U}(u, v) = \sum_{i=1}^{n_c} \sum_{j=1}^{m_c} R_{i,j}^{p,q}(u, v) \mathbf{Q}_{i,j} , \quad (3.4)$$

where $R_{i,j}^{p,q}(u, v)$ is exactly the same as in (2.10), but the solution coefficients, $\mathbf{Q}_{i,j}$, are used in place of the control points, $\mathbf{P}_{i,j}$. This process constructs a solution “surface,” $\mathbf{U}(u, v)$, which may be evaluated for each mesh point. Having obtained a result value for each mesh point, the values are assigned a color according to a relative color scale and then visualized in the Rhino viewport via the visualization mesh.

The Grasshopper implementation of this process is shown in Figure 3.5. The solution mapping, $\mathbf{U}(u, v)$, is constructed within the C# scripting component in the top left, whereas it is evaluated at the mesh point coordinates in the components in the upper right. The solution mesh is constructed in the components in the bottom left of the figure and is then colorized according to the evaluated solution values using the bottom right components. This colorized solution mesh is then automatically displayed in the Rhino viewport. The density of the visualization mesh can also be varied within the Grasshopper definition since $\mathbf{U}(u, v)$ itself is interactively evaluated within the definition.

3.4 Tube profile optimization

In order to demonstrate the effectiveness of the IGA-based parametric design optimization framework, a design with a known solution is optimized: the cross-sectional geometry of an internally pressurized tube.

3.4.1 Objective function

The general optimization problem is encoded in an objective functional $\mathcal{J}^t(\mathbf{u}; \boldsymbol{\chi}^t)$. The objective functional depends explicitly on the displacement field variables, \mathbf{u} , which are evaluated via solving the PDE $\mathcal{B}(\mathbf{u}) = 0$. Additionally, the objective functional depends

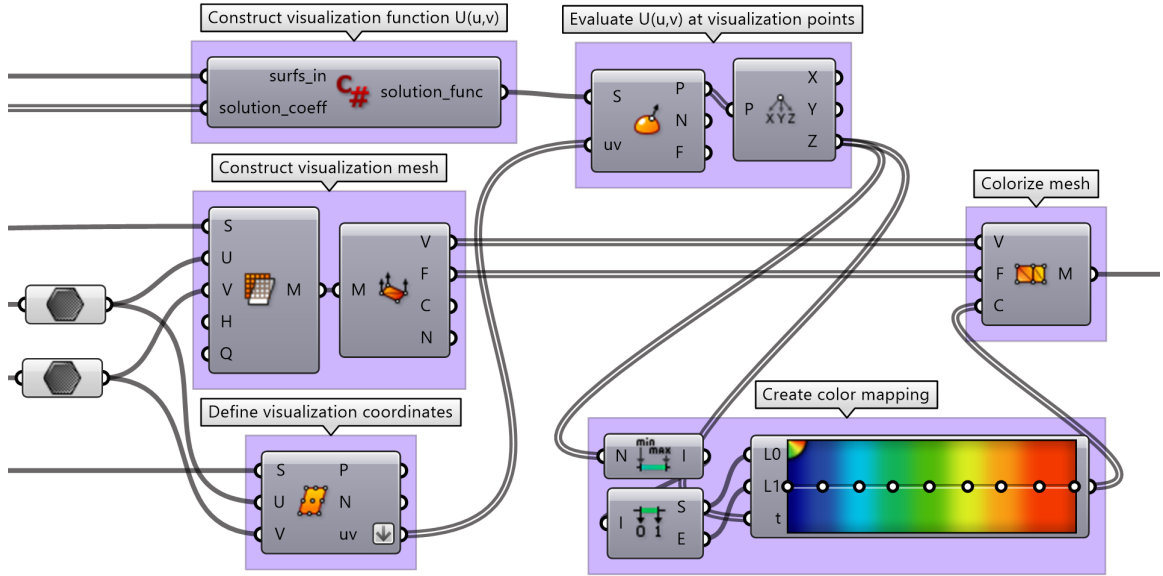


Figure 3.5: Grasshopper implementation of visualization methodology.

implicitly on the design variables χ^t (usually via the field variables $\mathbf{u}(\chi^t)$). The resulting PDE-constrained optimization problem is posed as follows:

$$\begin{aligned}
 & \text{minimize} && \mathcal{J}^t(\mathbf{u}; \chi^t) \\
 & \text{subject to} && \chi^t \in \Omega^t, \\
 & && \mathcal{B}(\mathbf{u}; \chi^t) = 0.
 \end{aligned} \tag{3.5}$$

$\mathcal{J}^t(\mathbf{u}; \chi^t)$, defined below, is calculated for each design-and-analysis iteration; χ^t is the vector of design variables, defined in a later section; and Ω^t is the vector of allowable ranges for each design variable. Recall that, for the family of tube designs, the generative algorithm $\Theta^t(\chi^t)$, shown in Figure 3.3, is employed. This algorithm acts as a preprocessor for the analysis of each design by producing the geometry definition that allows $\mathcal{J}^t(\mathbf{u}; \chi^t)$ to be calculated.

For many structural analyses, and for this problem in particular, it is reasonable to minimize the maximum strain in a design because strain is directly related to many popular failure criteria. As previously discussed, in the isogeometric Kirchhoff–Love thin-shell

formulation [50, 112], the Green–Lagrange strain, \mathbf{E} , is separated into a constant part, due to membrane action, and a linearly varying part, due to bending, as follows:

$$\mathbf{E} = \boldsymbol{\varepsilon} + \xi_3 \boldsymbol{\kappa} , \quad (3.6)$$

where $\boldsymbol{\varepsilon}$ denotes the membrane strain of the midsurface, $\boldsymbol{\kappa}$ denotes the change in curvature of the midsurface due to bending, and ξ_3 is the through-thickness coordinate.

For this example, the expected optimal cross-sectional shape is a circle because it is capable of supporting the entirety of the internal pressure load with only in-plane (membrane) stretching and zero bending action. One can therefore minimize

$$\mathcal{J}^t(\mathbf{u}; \boldsymbol{\chi}^t) = \boldsymbol{\kappa}_{\max}(\mathbf{u}; \boldsymbol{\chi}^t) , \quad (3.7)$$

where $\boldsymbol{\kappa}_{\max}(\mathbf{u}; \boldsymbol{\chi}^t)$ is the maximum component of the maximum curvature change present in the design generated by the design variables $\boldsymbol{\chi}^t$.

3.4.2 Design variables

Much isogeometric shape optimization literature focuses on the optimization of control point locations and, in some cases, weights. While this is reasonable for small-scale problems like this one, it is desirable to reduce the number of design variables to produce only designs within a particular design space of interest. For the internally pressurized tube case, the design space is constrained such that it contains only tubes with a uniform cross section and that are symmetric about two perpendicular planes. Therefore, as Figure 3.6 illustrates, one quarter of the cross section is created using a NURBS curve featuring three control points with weights of $\{1, \frac{\sqrt{2}}{2}, 1\}$. The two end control points are fixed at a radial distance of one unit from the origin, and the middle control point is allowed to move radially towards or away from the origin. Therefore, for this problem, the design variables are defined as

$$\boldsymbol{\chi}^t = \{r\} , \quad (3.8)$$

where r is the radial distance from the origin to the second NURBS control point as illustrated by Figure 3.6. This parameterization allows both square and circular cross sections

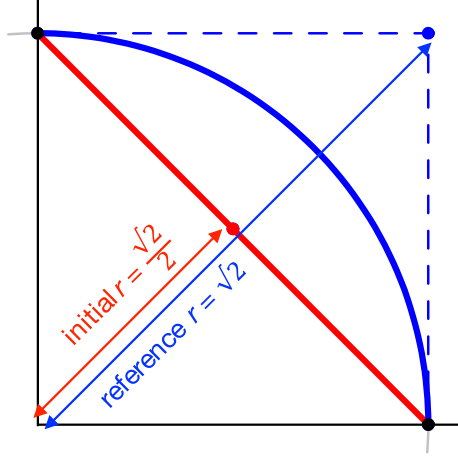


Figure 3.6: Definition of the design variable r for the internally pressurized tube problem.

to be generated by varying a single design variable. The planar curve is then extruded in the perpendicular plane to generate a surface; the surface is replicated, and the replicate surfaces are rotated to create the remaining three quarters of the tube. Recall that the four quarters, each represented by a single patch, are coupled using the bending strip method [97]. This generative geometric procedure, $\Theta^t(\chi^t)$, is shown in Grasshopper in Figure 3.3.

3.4.3 Simulation setup and solution strategies

IGA mesh density is selected using k -refinement³ (degree three in the u and v parametric directions) so as to balance the need for accuracy and the desire to limit analysis time, a critical factor for heuristic optimization techniques. A thickness of 2 cm is used with a Young’s modulus of 0.4 GPa and Poisson’s ratio of zero. The non-variable portions of the cross-sectional radius are fixed at 1 m and the height of the tube is 3 m. An internal pressure of 10 kPa is applied and a single control point is fixed for better numerical stability.

Newton–Raphson iterations are directly applied to converge the residual of this static problem. For each Newton–Raphson iteration, used to converge geometric nonlinearities,

³The processes of knot insertion (h -refinement) and order elevation (p -refinement) do not commute. k -refinement, proposed in Hughes et al. [19], elevates the order of the original curve and then inserts a unique knot value. This process maintains the elevated-order continuity of the curve at the newly inserted knot.

the linear system is solved using an iterative, diagonally-preconditioned conjugate gradient solver. Note that, in this case, there is only one design variable such that the objective function is one-dimensional. A variety of optimization methods contained in the MATLAB Optimization Toolbox [91] can be used. In order to demonstrate that the presented framework is applicable to a wide variety of engineering design problems, MATLAB's generalized pattern search (GPS) algorithm with a positive 2N basis and mesh tolerance of 0.001 is used. Pattern search methods do not suffer from some of the problems associated with gradient-based finite differencing, such as potential oversensitivity or insensitivity to design variable variation, and can therefore be more readily used for many design applications [91, 113], including problems of high dimensionality. Although this approach may not be the most efficient, it is reliable and also has rigorous local convergence properties [114].

3.4.4 Results and discussion

Optimization is performed using 16 GB RAM and a single core of a 2.2 GHz Intel Core i7 processor. A total of 38 designs are evaluated; each function evaluation takes about 25 seconds, yielding a total optimization time of about 16 minutes. Externally routing the optimization procedures through MATLAB requires the reading and writing of design variables and objective function variables, all of which requires less than one second for each function evaluation. Reading and writing analysis model data takes less than 100 ms per function evaluation. The entire process could theoretically be expedited by parallelizing structural analysis, optimization procedures, or both.

Figure 3.7 shows the design variable r plotted versus the objective function value; that is, the maximum curvature change at any point in the design. It is clear that, throughout optimization, the design variable r converges towards the reference solution of $\sqrt{2}$, the radial control point position at which a perfectly circular cross section is achieved. Additionally, the maximum curvature change, $\kappa_{\max}(\mathbf{u}; \boldsymbol{\chi}^t)$, converges to zero.

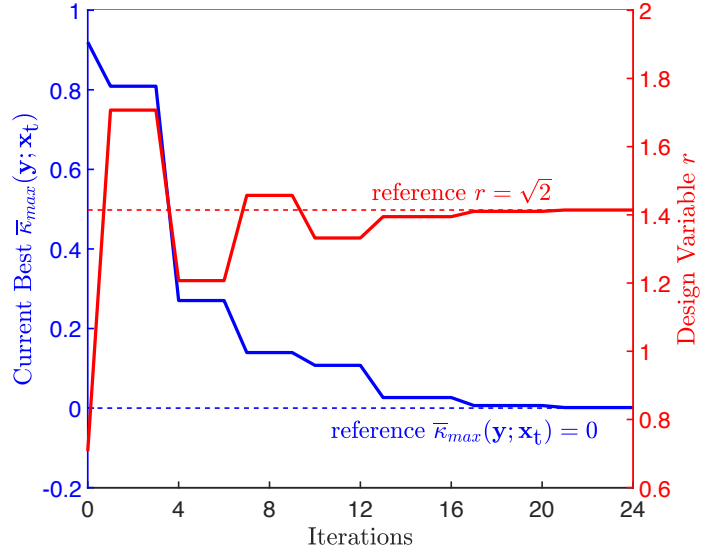


Figure 3.7: Design variable r and current best objective function value versus number of pattern search iterations.

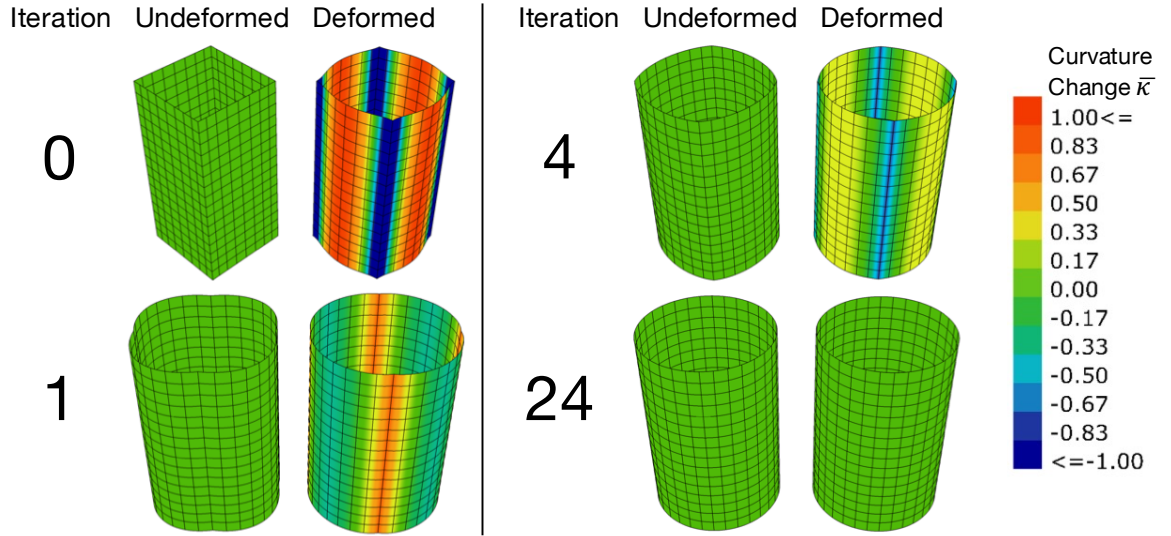


Figure 3.8: Undeformed and deformed shapes of current best tube design at selected optimization iterations. Color contour denotes value of maximum component of curvature change.

These results are corroborated by Figure 3.8, which shows the strain contours on the undeformed and deformed geometries of the current best design at various points in the optimization process. Note that the maximum curvature change occurs in the initial, perfectly square cross section, whereas after the last iteration, iteration 24, there is zero curvature change even after loading. The results are in good agreement with the expected values and demonstrate the framework’s ability to optimize simple parametric designs using IGA and heuristic optimization techniques.

3.5 Wind turbine blade optimization

To demonstrate the applicability of the isogeometric design optimization framework to more complex engineering problems, the framework is used to improve the design of the NREL 5 MW wind turbine blade [115]. More specifically, the baseline design is used to establish a performance benchmark; an optimization algorithm is then used to vary a subset of design parameters to obtain a design with theoretically improved performance.

3.5.1 Objective function

The optimization problem is posed as follows:

$$\begin{aligned}
& \text{minimize} && \mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b) \\
& \text{subject to} && \boldsymbol{\chi}^b \in \boldsymbol{\Omega}^b , \\
& && \mathcal{B}(\mathbf{u}; \boldsymbol{\chi}^b) = 0 , \\
& && \mathcal{C}_i^b(\mathbf{u}; \boldsymbol{\chi}^b) \leq 0 , \quad i = 1, \dots, n_c ,
\end{aligned} \tag{3.9}$$

where $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b)$ is the objective function, defined below, calculated for each design-and-analysis iteration; $\boldsymbol{\chi}^b$ is the vector of design variables; $\boldsymbol{\Omega}^b$ is the vector of allowable ranges for the design variables; and there are n_c inequality constraints, \mathcal{C}_i^b , that the optimized design must satisfy.

Formulating a meaningful objective function for complex engineering designs is challenging, but it is critical for achieving quality optimization results. In the case of wind turbine

blades, a variety of values that derive from the creation and analysis of a computational model may be of interest to the designer. An effective objective function unites these values in a logical and meaningful way, essentially ranking the many design alternatives according to designer-defined objectives.

For this simplified example, the effect that variation of the NREL 5 MW wind turbine blade design has on the machine's overall simple payback period, S_{PP} , is quantified and employed as the objective function. The following relation for payback period is considered:

$$S_{PP} = \frac{K_{CC}}{K_{AAR}} , \quad (3.10)$$

where K_{CC} is the total capital cost of the machine, K_{AAR} is the average annual return, and S_{PP} is the simple payback period for the machine [4]. The simple payback period, as implied by this definition, is the amount of time that it takes for a wind turbine's total revenue production to match initial capital investment. A reduction of payback period indicates that a turbine will be able to produce profit over a larger portion of its operating life.

Rather than computing the capital cost and average annual return for every blade design, one can instead approximately quantify the effect a particular design variation would have on the simple payback period using some assumptions. The following equation, calculated for each blade design variant, is defined:

$$S_{PP}(\chi^b) = \frac{K_{CC_0} + V_{CC}(\chi^b)K_{CC_0}}{K_{AAR_0} + V_{AAR}(\chi^b)K_{AAR_0}} = C_0 \frac{1 + V_{CC}(\chi^b)}{1 + V_{AAR}(\chi^b)} , \quad (3.11)$$

where the subscript zero on K_{CC} and K_{AAR} indicates reference values that are obtained from analysis of a baseline blade design, $V_{CC}(\chi^b)$ indicates the fractional variation of the capital cost as a result of design variation, and $V_{AAR}(\chi^b)$ indicates the fractional variation of the average annual return as a result of design variation. The constant C_0 entails all components of the original capital cost, K_{CC_0} , and original average annual return, K_{AAR_0} , which are unaffected by the blade design variation χ^b . The blade-design-dependent values that most directly influence the simple payback period are the blade's capital cost, estimated using

blade mass in this chapter, and power output. It is assumed that mass is related to $V_{CC}(\mathbf{x}^b)$ in the numerator of Eq. (3.11) and power is related to $V_{AAR}(\mathbf{x}^b)$ in the denominator.

The numerator and the influence of mass variation are considered first. In [116] the International Renewable Energy Agency (IRENA) states that, for 5 MW applications, the blades make up 22.2% of the capital cost of the wind turbine. It further states that the capital cost of the turbine itself comprises 51% of the total capital cost of offshore wind turbine installations; combining these claims, one may surmise that 11.32% of the total capital cost is due to the blades. Other sources may cite varying percentages, but the value of 11.32% is sufficient for this example. If one assumes that the mass of the blade is proportional to its cost (assumed for simplicity in this chapter), the variation in the capital cost due to blade design variation can be captured as follows:

$$V_{CC}(\mathbf{x}^b) = 0.1132 \left(\frac{M(\mathbf{x}^b) - M_0}{M_0} \right), \quad (3.12)$$

where $M(\mathbf{x}^b)$ indicates the mass of a blade design variant and M_0 indicates the baseline NREL 5 MW design's mass.

The denominator of Eq. (3.11), containing V_{AAR} , is now considered. The average annual return, K_{AAR} , is equal to $E_a D_e$, where E_a is the annual energy production and D_e is the price obtained for electricity. One may further recognize that the annual energy production E_a is the product of the nameplate capacity of a machine, P , and the capacity factor, C_F . Thus, $K_{AAR} = PC_F D_e$. Because many blade design variants that might be used interchangeably in the same operating environment are being considered, D_e and C_F are assumed to be constant and are lumped into C_0 . The variation in average annual return due to blade design variation can therefore be defined as

$$V_{AAR}(\mathbf{x}^b) = \frac{P(\mathbf{x}^b) - P_0}{P_0}, \quad (3.13)$$

where $P(\boldsymbol{\chi}^b)$ indicates the power production of a blade design variant and P_0 indicates the baseline NREL 5 MW design's power production.

Substituting Eqs. (3.12) and (3.13) into Eq. (3.11) one is left with

$$S_{PP}(\boldsymbol{\chi}^b) = C_0 \frac{1 + 0.1132 \left(\frac{M(\boldsymbol{\chi}^b) - M_0}{M_0} \right)}{1 + \frac{P(\boldsymbol{\chi}^b) - P_0}{P_0}}. \quad (3.14)$$

Because Eq. (3.14) is to be minimized and because C_0 , a constant, is proportionally related to the rest of the equation, one can finally define the objective function as the non-constant portion of Eq. (3.14), or

$$\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b) = \frac{1 + 0.1132 \left(\frac{M(\boldsymbol{\chi}^b) - M_0}{M_0} \right)}{1 + \frac{P(\boldsymbol{\chi}^b) - P_0}{P_0}}. \quad (3.15)$$

Note that the right-hand side of Eq. (3.15) does not incorporate the blade displacements, \mathbf{u} , explicitly because the displacements are only used to calculate the constraints for this particular example.

The value of $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b)$ indicates a design alternative's payback period in terms of a proportion of the original payback period. The baseline 5 MW blade design has a $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b)$ of 1.0, or 100% the reference payback period; better-performing designs have $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b) < 1$; and poorer-performing designs have $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b) > 1$. Better-performing designs will recover initial investment costs more quickly and should be more profitable overall.

The simple payback period, S_{PP} , is not the only metric used to judge overall cost efficiency of wind turbines. Other metrics include the cost of energy (COE) and levelized cost of energy (LCOE) which could be used in a similar fashion. The simple payback period is used here as a basic demonstration of a multidisciplinary objective. Additionally, numerous factors that would not be superfluous in an actual blade design context are ignored here in order limit the scope of the optimization problem. Such factors include the effect of blade design on power production across the entire possible range of wind conditions (hence ignoring capacity factor variation), the effect of blade mass on tower cost, and modal changes due to mass redistribution. This example is intended merely to demonstrate usage of the isogeometric optimization framework.

3.5.2 Design variables and constraints

While simple geometries are often described using control point locations and weights, such fundamental geometry descriptions are intractable as primary descriptors of more complex models. More highly abstracted parametric relations are established for this reason. Wind turbine blades are generally constructed according to a set of design parameters that are defined at discrete locations along the blade. The geometric parameters are usually a section's radial location, chord length, airfoil shape, and twist degree. The Grasshopper algorithm for generating wind turbine blades using these inputs is shown in Figure 3.4. For this simple optimization problem, the chord length, which has definite implications for both blade mass and power production, is considered.

As can be seen in [115], the original 5 MW blade has nineteen locations, or “stations,” at which the design parameters are defined, corresponding to nineteen chord lengths along the blade span. Instead of using these nineteen chord lengths as the design parameters for the optimization problem, an alternative parameterization strategy is used to reduce the dimensionality of the design space. The strategy consists of creating a variation profile constructed from a quadratic B-spline of six control points evenly spaced along the blade span as shown in Figure 3.9. Varying the four internal control points in the direction of chord profile size allows semi-local control over this chord profile variation. The value of the variation profile at each of the nineteen cross-sectional locations along the blade span is added to the original profile to generate a new profile. The design variables for this example are therefore defined as

$$\boldsymbol{\chi}^b = \{Z_i\}, \quad i = 1, \dots, 4, \quad (3.16)$$

where Z_i is the vertical movement in Figure 3.9 of each of the four internal control points. The generative algorithm $\Theta^b(\boldsymbol{\chi}^b)$, excluding the algorithm for chord profile variation, is shown in Figure 3.4. The different blades in Figure 3.4 were generated using this variation approach.

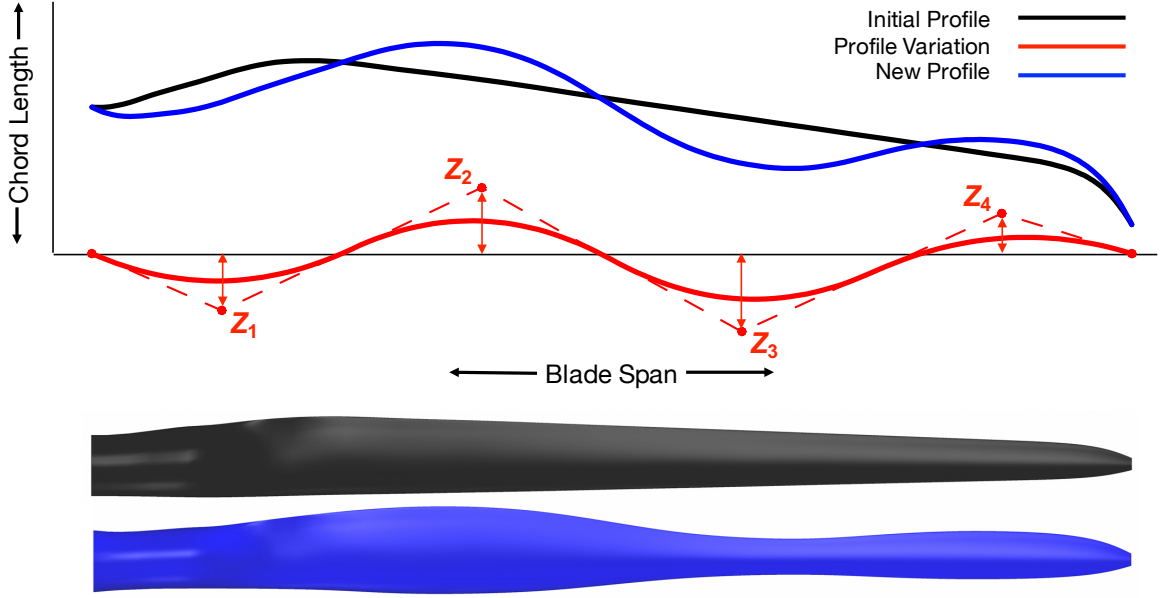


Figure 3.9: Demonstration of strategy for variation of chord profile using reduced number of parameters. Internal control points of a quadratic B-spline are moved and the variation profile is superposed upon the original profile. Original blade geometry is shown in black (second from bottom) and new blade geometry is shown in blue (bottom).

An additional consideration for most realistic optimization problems is the optimization constraints. While the objective function $\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b)$ provides an explicit relationship between blade mass and power production, it does not take into account other potential constraints such as stress and strain or kinematics. Two such constraints are considered here: the maximum tip deflection of the blade, which is associated with tower clearance, and maximum in-plane strain, which is associated with material failure. Two constraint cases are optimized and discussed. For the first case, a single constraint is used:

$$\mathcal{C}_1^b(\mathbf{u}; \boldsymbol{\chi}^b) = \delta_{\text{tip}}^b(\mathbf{u}; \boldsymbol{\chi}^b) - \delta_{\text{tip}_0}^b \leq 0, \quad (3.17)$$

where $\delta_{\text{tip}}^b(\mathbf{u}; \boldsymbol{\chi}^b)$ denotes the out-of-plane tip deflection of a potential blade design and $\delta_{\text{tip}_0}^b$ denotes the out-of-plane tip deflection of the baseline design. For the second constraint case, in addition to (3.17), a second constraint is added:

$$\mathcal{C}_2^b(\mathbf{u}; \boldsymbol{\chi}^b) = \epsilon_{\text{max}}(\mathbf{u}; \boldsymbol{\chi}^b) - \epsilon_{\text{max}_0} \leq 0, \quad (3.18)$$

where $\epsilon_{\max}(\mathbf{u}; \boldsymbol{\chi}^b)$ denotes the maximum in-plane Green–Lagrange strain of a potential blade design and ϵ_{\max_0} denotes the maximum in-plane Green–Lagrange strain of the baseline design. For both constraint cases, if any of the constraints are violated for a given set of design variables, the design variables are no longer considered potentially optimal solutions due to programmatic enforcement of a penalty on the objective function.

3.5.3 Simulation setup and solution strategies

Structural analysis is set up for the NREL 5 MW wind turbine blade and design variants $\Theta^b(\boldsymbol{\chi}^b)$ as follows. Because the geometry of wind turbine blades is critical to the machine’s power production capability, Grasshopper is used to incorporate an aerodynamic analysis module, NREL’s wind turbine analysis tool called FAST [117], which enables both approximate calculation of $P(\boldsymbol{\chi}^b)$ for a given design and the extraction of aerodynamic loads. FAST uses an implementation of blade element momentum theory to quickly produce an aerodynamic torque prediction for a given wind turbine setup. A force vector is calculated for each discrete segment (blade element) of the blade defined in FAST. These forces are distributed into traction vectors that are uniformly applied to the portion of the blade model corresponding to each FAST blade element. FAST is based on purely parametric input so it is easily incorporated into the presented framework. Typical blade design procedures require consideration of many different loading scenarios; however, for the purposes of this work, a single loading scenario is considered. The FAST analyses are based on a standard 5 MW setup described in Jonkman et al. [115] with a no-shear wind speed of approximately 11.3 m/s, nearly the speed at which the turbine should be operating at rated power and out-of-plane tip deflection should be relatively high.

For all blade designs, a simplified composite layup is defined using some of the materials found in Sandia National Laboratory’s composite layup definition for the NREL 5 MW blade [118]. Basic material zones—shell layup, root reinforcement, and spar cap—of uniform thickness are defined, shown in Figure 3.10. The entire blade surface consists of

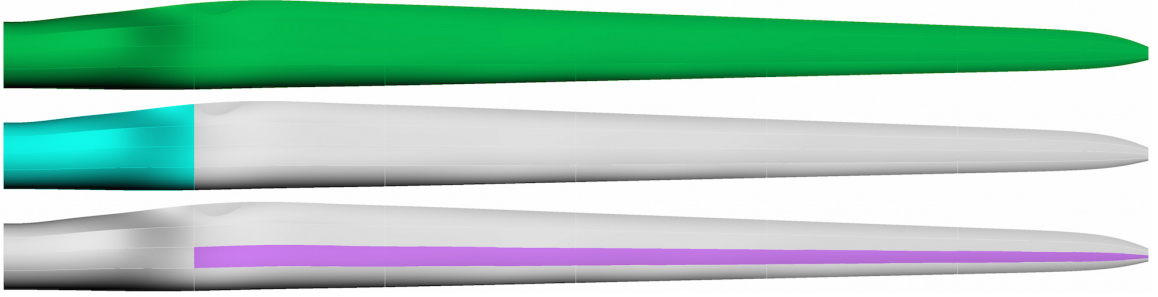


Figure 3.10: Simplified composite layup used for wind turbine blade optimization. Green color (top) indicates base E-LT-5500 fiberglass over entire blade, blue color (middle) indicates root buildup of SNL Triax, and purple color (bottom) indicates spar cap region made up of uni-directional carbon.

uni-directional E-LT-5500 fiberglass with additional uni-directional carbon in the spar cap region and SNL Triax added to the root. Material properties can be found in [118]. Zone thicknesses are chosen such that the maximum tip deflection of the baseline blade design is approximately equal to the deflection specified in [115] under the given wind conditions. 3.25 cm of fiberglass, 1 cm of SNL Triax, and 8 cm of uni-directional carbon are used. The total mass of the baseline blade geometry with this material setup, $M_0 = 40,912$ kg, is much higher than in the reference [115]; this is expected because the shear web structures are omitted for this simple example. A thicker shell definition is thus required to achieve realistic tip deflection under the given loading.

As in the previous example, IGA mesh density is selected using k -refinement (degree three in the u and v parametric directions) so as to balance the need for accuracy and the desire to reduce analysis time. Of course, if a higher degree of accuracy for each function evaluation is required, the mesh density can be increased, also increasing overall optimization time. For this dynamic problem, which terminates once the maximum tip deflection has been reached, the algebraic problem is addressed by direct application of Newton–Raphson iterations to converge the residual at each time step. For each Newton–Raphson iteration, used to converge geometric nonlinearities, the linear system is solved

using a diagonally-preconditioned conjugate gradient method. The objective function is again minimized using MATLAB's generalized pattern search (GPS) algorithm with positive 2N basis and mesh tolerance of 0.01. Because the baseline 5 MW design is assumed to have relatively good performance, the baseline 5 MW design, defined by $\chi^b = 0$, is used as the initial point for pattern search optimization.

3.5.4 Results and discussion

As in the previous example, optimization is performed using a single core of a 2.2 GHz Intel Core i7 processor and 16 GB RAM. Each design evaluation takes approximately 9.5 minutes. Communication of information between MATLAB and Grasshopper takes less than one second per function evaluation, while the reading and writing of model data takes less than 100 ms per function evaluation. The total number of requisite function evaluations for the first and second constraint cases is 128 and 102, respectively. Thus, optimization using the first constraint case takes approximately 20 hours whereas optimization using the second constraint case takes approximately 16 hours. Again, the overall procedure could be expedited by parallelizing structural analysis, optimization procedures, or both. Tabular results of solution values of interest are shown in Table 3.1. Graphs demonstrating both the optimization history of the objective function value and the optimized chord profiles for each constraint case are shown in Figure 3.11. The original and optimized blade geometries in the undeformed configuration are shown side by side in Figure 3.12. Comparison of strain distributions on original and optimized blade designs in their most deformed states are shown in Figure 3.13.

Table 3.1 shows that both optimizations yielded a design with a theoretical payback period slightly lower than the original payback period: a reduction of approximately 0.51% for the first constraint case and approximately 0.22% for the second constraint case, theoretically yielding an additional 6.38 and 2.75 million dollars (see Remark 1) of additional profit, respectively, over the lifetime of a large-scale wind farm. The difference between these two

Table 3.1: Summary of results of interest for original and optimized designs. Only tip deflection is constrained for the first case, whereas both tip deflection and maximum strain are constrained for the second case. Additional profit is defined over the entire lifetime of a large-scale offshore wind farm featuring an given blade design.

Design	Func. Evals	Tip Defl. (m)	Max. Strain	Mass (kg)	Power (kW)	$\mathcal{J}^b(\mathbf{u}; \boldsymbol{\chi}^b)$	Add. Profit (millions \$)
Original	–	5.75	0.0083	40,912	5,265	100.00%	–
Case 1	128	5.75	0.0100	41,650	5,302	99.49%	6.37
Case 2	102	5.12	0.0083	43,265	5,311	99.78%	2.75

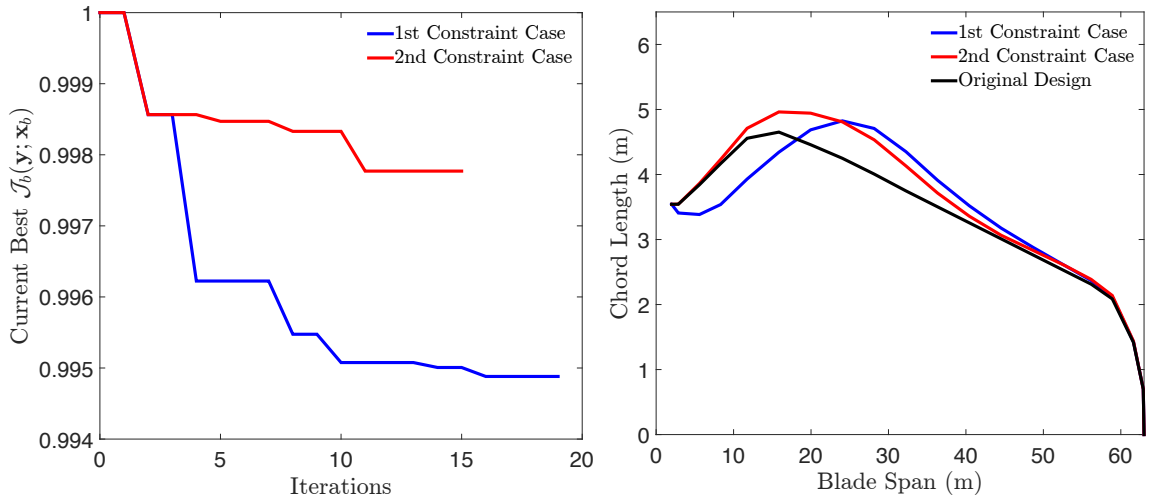


Figure 3.11: Iterative history of best objective function value for each constraint case (left) and chord profiles of original and optimized designs (right).

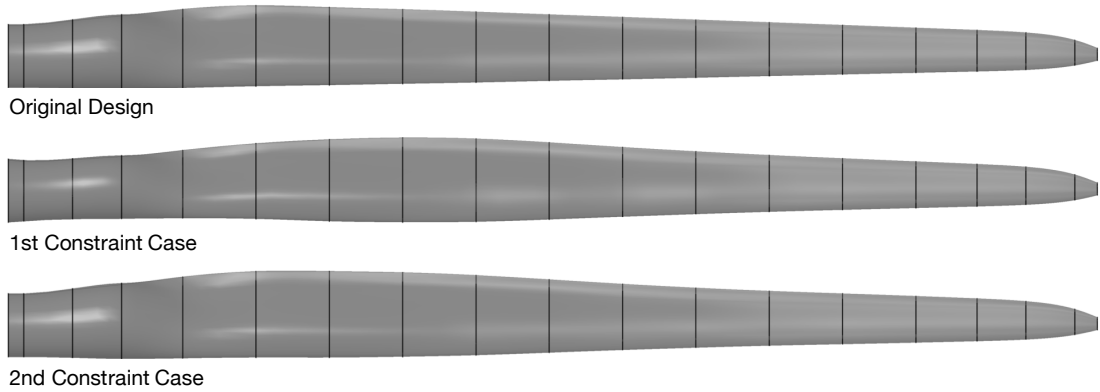


Figure 3.12: Comparison of original and optimized blade shapes viewed from the flapwise direction. Station sizes and locations are indicated by black lines.

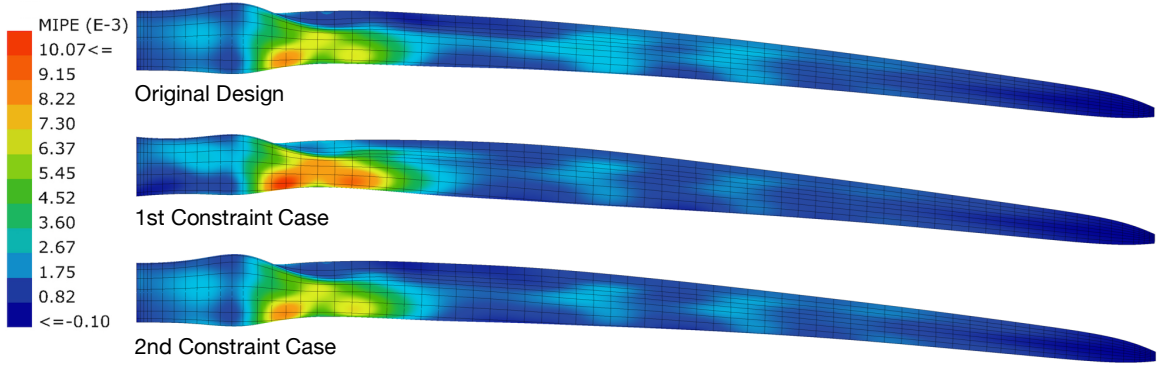


Figure 3.13: Comparison of strain distributions on original and optimized blade designs (shown in their most deformed states). The first principal in-plane strain on the outer surface of the shells is plotted. Blades are pitched 35 degrees from angle used in Figure 3.12 to show region of strain concentration.

results is reasonable because the second case takes maximum strain, a potentially important factor depending on the design scenario, into account. Both optimized designs have larger overall profiles, increasing both mass and potential power output. Because power input increase is inversely related to the payback period in the objective function (3.15) and because it is weighted more heavily than mass, it is reasonable that an increase in both would be justifiable from a payback period perspective.

Remark 1. Using $\mathcal{J}^b(\mathbf{u}; \chi^b)$ to calculate additional profit that is achievable over the life of a wind farm as a result of optimization, as is done in Table 3.1, requires additional assumptions to be made. The dollar values shown in this work are calculated using the Thornton Bank offshore wind farm, made up of 60 turbines with a 5 MW capacity and having a capital cost of 1.25 billion dollars, as a reference [119]. An offshore wind farm capacity factor of 42.4% is used [120] with an assumed price of electricity of 0.11 \$/kWh. As a basic performance measure, the simple payback period, S_{PP} , ignores operations and maintenance costs.

The right plot in Figure 3.11 provides greater insight into the salient design trends in these scenarios. The first optimized design has an increased chord length in the outer

portion of the wind turbine blade’s span, creating greater potential for aerodynamic torque production. Near the blade root, however, where the potential for aerodynamic torque production is the lowest, the blade’s chord size is reduced in an effort to reduce mass. Although there is a greater amount of force across the entirety of the blade and although the root of the blade is smaller, the tip deflection is the same as the baseline because the thicker central portion of the blade provides additional stiffness. The increased maximum strain experienced by this aerodynamically aggressive design, however, may be undesirable.

The maximum in-plane strain is used as an additional constraint for the second optimization. The second optimized design, like the first optimized design, is generally wider to increase energy capture. However, rather than having a thinner root section, which ultimately increased stress concentration, the blade has a larger root. Thus, greater aerodynamic torque production is achieved without also increasing the maximum in-plane strain. Of course, mass is also higher in this case but, governed by the objective function (3.15), mass increase is offset by higher power production potential. Relative strain distributions and deformed blade shapes are shown in Figure 3.13.

The wind turbine blade example clearly demonstrates the benefits of using high-fidelity IGA and optimization in a realistic, multidisciplinary design context. Without giving special attention or *a priori* “knowledge” to the system about particular design concepts that might be intuitive to human designers, the system is able to produce designs that align with human judgment—such as increasing chord size where energy capture potential is high or varying root size according to strain specifications—but in a more precise and less laborious way. Importantly, no effort is expended on finite element mesh generation throughout this entire design process.

3.6 Chapter conclusion

A computational framework for parametric design optimization using isogeometric analysis is presented. In Section 3.3, the principal features of the computational framework are

summarized and the role of each feature in enabling a novel IGA-based parametric design optimization methodology is emphasized. The framework is based on Grasshopper, an algorithmic modeling interface that abides by and uniquely integrates a number of important design philosophies and that also contains powerful geometry manipulation functions that enable the parametric generation of models suitable for IGA. In the context of this unified design framework, which features consistent geometry descriptions throughout design and analysis, analysis-driven optimization even of complex designs is natural and relatively simple. It is a unique framework in that it enables parametric design optimization of a variety of CAD-generated engineering structures using IGA.

In Section 3.4, the framework’s ability to accurately optimize a simple pressurized tube design, a design parameterized with a single design variable, is demonstrated. In Section 3.5, a more realistic design scenario with more abstract design parameters is considered: the design of a wind turbine blade. Optimizing a different design does not require fundamental restructuring of the isogeometric design optimization framework. Instead, optimization merely requires the development of the Grasshopper design algorithm $\Theta^b(\chi^b)$ for wind turbine blades, a relatively simple task for experienced designers made simpler via the use of an interactive interface, and the selection and integration of an appropriate IGA methodology. The theoretical payback period of a 5 MW wind turbine is optimized. It is shown that, under the given assumptions, the payback period could be reduced by approximately 0.22% in the most conservatively constrained optimization case. Even this small improvement could yield an additional profit on the order of 2.75 million dollars over the life of a large-scale offshore wind farm. The optimization also reveals analysis-based trends which are useful to the designer. Consistent with the fundamental goals of IGA, no effort is expended on traditional finite element mesh generation throughout the entire design process.

Overall, this framework demonstrates how the benefits of IGA can be leveraged in realistic engineering design contexts to generate optimized designs and design alternatives based on high-fidelity structural analysis, reducing designer labor. One of the fundamental

goals of computational analysis and design is, simply stated, to achieve optimized designs before experimentation or production even begins. In actual practice, however, the state of the modern engineering workflow is a significant barrier to the realization of this goal. This chapter directly addresses not only the problem of design and analysis, but the design-and-analysis environment itself. Generally, addressing the issues encountered in this context represents an important step towards enabling more effective use of IGA-based parametric design optimization. The framework is also useful in other highly iterative design contexts, such as design exploration (see, for example, [121]). Lastly, the work represents an important first step towards constructing a framework for IGA-based design and optimization of wind turbine blades in that it outlines an approach for reliably and iteratively generating analysis-suitable, NURBS-based wind turbine blade geometries using typical parametric inputs in a CAD environment.

3.7 Acknowledgements

Chapter 3 is, in part, a reprint of material as it appears in: “A framework for parametric design optimization using isogeometric analysis,” (with N. M. Wiese, C. N. Darling, B. Ganapathysubramanian, A. Krishnamurthy, and M.-C. Hsu), *Computer Methods in Applied Mechanics and Engineering*, 316:944–965, 2017. The dissertation author was the primary investigator of this paper.

CHAPTER 4. A PENALTY FORMULATION FOR PATCH COUPLING

4.1 Chapter overview

As emphasized in Chapter 3, the wind turbine blade optimization problem considered in Section 3.5 employed significant simplification, in part in regards to blade structure. Some of these simplifications stemmed from the desire to model the blade using a single NURBS surface patch to avoid instances of C^0 continuity, a limitation imposed by Kirchhoff–Love theory. The need to generate a single-patch model precludes the modeling of shear webs or the use of multiple patches for any other reason, such as to capture material discontinuities. Of course, many geometries besides wind turbine blades are most practically modeled using multiple surfaces. Thus, the coupling of disjoint isogeometric Kirchhoff–Love patches is an important topic in IGA. Chapter 4 therefore presents a practical, effective approach for coupling NURBS patches both with matching and non-matching discretizations. The method is useful in the field of thin-shell IGA broadly, as well as in the specific application of a framework for IGA-based design and optimization of wind turbine blades.

This chapter is outlined as follows. In Section 4.2, the background of patch coupling in the context of isogeometric Kirchhoff–Love thin-shell analysis is given. In Section 4.3, the penalty formulation for patch coupling is presented. An effective selection of penalty parameters is also proposed. In Section 4.4, the coupling formulations are evaluated using several linear and nonlinear benchmark problems, and the effective range of the relatively problem-agnostic penalty coefficient is demonstrated. The effectiveness of the method is demonstrated using linear buckling, vibration, and nonlinear deformation analyses. For all

of these examples, the design-and-analysis pipeline described in Chapter 3 is employed. In Section 4.5, conclusions are drawn.

4.2 Coupling of isogeometric Kirchhoff–Love patches

Models of complex, real-world objects like wind turbine blades tend to be composed of multiple patches due to practical or technical limitations in geometry modeling or to capture design features such as material discontinuities. Despite the many potential advantages offered by Kirchhoff–Love IGA, a few key issues prevent the method from being readily applied to such complex, multi-patch designs. First, even if neighboring patches have conforming discretizations and control points are joined, the C^0 continuity of the approximation functions at the interface implies that, due to the C^1 continuity requirement of the Kirchhoff–Love theory, no bending moment can be transferred. Additionally, in the more general case of non-conforming or even C^{-1} continuity, shared control points between adjacent patches cannot be readily coupled and small gaps or overlaps may appear. In traditional analysis, such topographical information is taken into account during mesh generation, ensuring that the resultant mesh is analysis-suitable. Because IGA circumvents mesh generation procedures, however, the analysis method itself must be capable of representing such topologies in a manner consistent with the designer’s intent.

The issue of patch coupling in the context of IGA has been explored in past works. One of the most commonly used methods, which was employed in Chapter 3, is known as the bending strip method [97]. The bending strip method is a penalty-like approach that introduces patches of fictitious material, having unidirectional bending stiffness and zero membrane stiffness, at patch boundaries. Although it is easy to implement, one of the requirements of the bending strip method is that the patches must be conforming—that is, the control points of the two patches must be co-located at their interface—a requirement that is often difficult to satisfy. Nitsche’s method has also been used for patch coupling and has been shown to be fairly robust [122–124]. However, Nitsche’s method also

depends on the variational formulation employed and, as a result, is relatively difficult to implement for general-purpose analysis codes. Citing the complexity of Nitsche’s method, Coox et al. [125] alternatively propose a Virtual Uncommon-Knot-Inserted Master–Slave (VUKIMS) coupling technique based on master–slave interface constraints derived from the interface knot vectors. Breitenberger et al. [57] have made significant efforts to integrate analysis paradigms with CAD software, specifically with boundary representation (B-Rep) data structures, and utilized a penalty method to constrain patch rotations about coupling interfaces. However, their formulation for rotational continuity was restricted to rotation angles less than $\pi/2$.

Because it admits a plurality of material models, the penalty method is one of the most popular approaches for patch coupling. Penalty methods are also relatively straightforward to implement in a broader analytical framework. Duong et al. [126] proposed a penalty approach posed in the curvilinear coordinate system based, in part, on surface normal vectors. This approach shows promise, although it was not explored in the ubiquitous cases of non-matching interfaces, laminated shells, or large-scale engineering problems. Additionally, one of the key drawbacks of the penalty method is that penalty parameter selection, which is important for balancing coupling constraint satisfaction and feasibility of the algebraic problem, is usually accomplished manually based on the specific problem configuration and user experience. One possibility, as suggested by Breitenberger et al. [57], is to scale the penalty parameter by Young’s modulus. However, this simple approach neglects other factors that can have a strong influence on the problem, such as shell thickness and discretization. Thus, in this chapter, the penalty formulation proposed by Duong et al. [126] is used to handle a variety of matching and non-matching configurations, penalty parameter formulations intended to eliminate the problem-dependent nature of penalty parameters are proposed, and the behavior of the overall penalty approach is thoroughly investigated.

4.3 Penalty-based patch coupling formulation

4.3.1 Coupling methodology

Duong et al. [126] presented a coupling methodology for patches having C^0 or even C^{-1} continuity. The formulation uses a penalty approach and employs the surface normal vectors to fix the relative angle between patches. However, the formulation was never explored in the context of non-matching problems nor was it applied to problems of appreciable complexity. In this work, it is shown that the formulation can be used for non-matching problems. Additionally, a solution to the problem of penalty parameter selection is proposed. It should be noted that, while special attention is given to composites in this work, the coupling formulation presented here is applicable to a wide range of material models.

In the following, it is assumed that there are two patches, \mathcal{S}^A and \mathcal{S}^B , which, in the undeformed configuration, are approximately co-located along an interface curve, \mathcal{L} . Co-location is enforced in the deformed configuration by integrating along the penalty curve, \mathcal{L} , to derive a “penalty virtual work”:

$$\delta W^{\text{pd}} = \alpha_d \int_{\mathcal{L}} \left(\mathbf{u}^A - \mathbf{u}^B \right) \cdot \left(\delta \mathbf{u}^A - \delta \mathbf{u}^B \right) d\mathcal{L} , \quad (4.1)$$

where α_d is a penalty parameter of large magnitude, further discussed in the following section, and \mathbf{u}^A and \mathbf{u}^B are the displacements of corresponding locations on \mathcal{S}^A and \mathcal{S}^B , respectively, along \mathcal{L} . Eq. (4.1) dictates that, if the distance between points on \mathcal{S}^A and \mathcal{S}^B is not the same in the deformed and undeformed configurations, a large penalty energy is introduced into the system.

The coupling methodology must also maintain the angle in between patches \mathcal{S}^A and \mathcal{S}^B . To maintain the angle in between patches having G^0 continuity, a penalty virtual work is constructed as follows:

$$\delta W_{G^0}^{\text{pr}} = \alpha_r \int_{\mathcal{L}} (\cos \varphi - \cos \hat{\varphi}) (\delta \cos \varphi - \delta \cos \hat{\varphi}) d\mathcal{L} , \quad (4.2)$$

where φ is the angle between the normal vectors of patches \mathcal{S}^A and \mathcal{S}^B and α_r is a penalty parameter that will be discussed in detail in the following section. Eq. (4.2) indicates that

φ should be the same in both the deformed and undeformed configurations; otherwise, a large penalty energy is introduced into the system. This formulation can be expressed in terms of the scalar product of the local unit normal vectors, $\mathbf{a}_3^A \cdot \mathbf{a}_3^B$, where the superscripts A and B indicate quantities evaluated on patches \mathcal{S}^A or \mathcal{S}^B , respectively:

$$\delta W_{G^0}^{\text{pr}} = \alpha_r \int_{\mathcal{L}} \left(\mathbf{a}_3^A \cdot \mathbf{a}_3^B - \hat{\mathbf{a}}_3^A \cdot \hat{\mathbf{a}}_3^B \right) \left(\delta \mathbf{a}_3^A \cdot \delta \mathbf{a}_3^B - \delta \hat{\mathbf{a}}_3^A \cdot \delta \hat{\mathbf{a}}_3^B \right) d\mathcal{L} . \quad (4.3)$$

As will be apparent later, Eq. (4.3) alone will not be effective for patches having an angle of π at their interface. Thus, a complementary penalty virtual work is defined:

$$\delta W_{G^1}^{\text{pr}} = \alpha_r \int_{\mathcal{L}} (\sin \varphi - \sin \hat{\varphi}) (\delta \sin \varphi - \delta \sin \hat{\varphi}) d\mathcal{L} , \quad (4.4)$$

Eq. (4.4) can be expressed in terms of the scalar product of the unit normal vector, \mathbf{a}_3^B , and the “in-plane unit normal vector”, \mathbf{a}_n^A , which lies in the plane of patch \mathcal{S}^A and is orthogonal to the penalty curve, \mathcal{L} . Given the natural tangent vector of the penalty curve on patch \mathcal{S}^A , $\tilde{\mathbf{a}}_t^A$, one can compute the unit vector, \mathbf{a}_t^A , using

$$\mathbf{a}_t^A = \frac{\tilde{\mathbf{a}}_t^A}{\|\tilde{\mathbf{a}}_t^A\|} . \quad (4.5)$$

With Eq. (4.5), \mathbf{a}_n^A can be computed as

$$\mathbf{a}_n^A = \mathbf{a}_t^A \times \mathbf{a}_3^A . \quad (4.6)$$

Note that \mathbf{a}_t^A and \mathbf{a}_3^A are orthogonal unit vectors. The penalty virtual work for patches having G^1 continuity can therefore be defined as follows:

$$\delta W_{G^1}^{\text{pr}} = \alpha_r \int_{\mathcal{L}} \left(\mathbf{a}_n^A \cdot \mathbf{a}_3^B - \hat{\mathbf{a}}_n^A \cdot \hat{\mathbf{a}}_3^B \right) \left(\delta \mathbf{a}_n^A \cdot \delta \mathbf{a}_3^B - \delta \hat{\mathbf{a}}_n^A \cdot \delta \hat{\mathbf{a}}_3^B \right) d\mathcal{L} . \quad (4.7)$$

The above formulation works for all patches not forming a kink of $\pi/2$. Together, Eqs. (4.3) and (4.7) constitute a penalty virtual work formulation capable of maintaining all possible angles.

Finally, the virtual work formulation, Eq. (2.11), is augmented by the contributions of Eqs. (4.1), (4.3), and (4.7), and is restated as

$$\delta W = \delta W_{\text{int}} + \delta W^{\text{pd}} + \delta W_{G^0}^{\text{pr}} + \delta W_{G^1}^{\text{pr}} + \delta W_{\text{ext}} = 0 . \quad (4.8)$$

The above equation states the equilibrium condition of virtual work that must be fulfilled for any arbitrary variation of the displacement variables δu_r . Eq. (4.8) is a nonlinear equation system which can be linearized for the purposes of solving the shell problem using the Newton–Raphson method:

$$\frac{\partial^2 W}{\partial u_r \partial u_s} \Delta u_s = -\frac{\partial W}{\partial u_r} . \quad (4.9)$$

The first derivative of the virtual work terms is the residual force vector, whereas the second derivative yields the stiffness matrix. The contributions from the internal and external virtual work are detailed in Kiendl et al. [50]. Here, we present the first and second derivatives of the penalty virtual work. The derivatives of the displacement penalty virtual work, Eq. (4.1), are given as follows:

$$\frac{\partial W^{\text{pd}}}{\partial u_r} = \alpha_d \int_{\mathcal{L}} \left(\mathbf{u}^{\text{A}} - \mathbf{u}^{\text{B}} \right) \cdot \left(\mathbf{u}_{,r}^{\text{A}} - \mathbf{u}_{,r}^{\text{B}} \right) d\mathcal{L} , \quad (4.10)$$

$$\frac{\partial W^{\text{pd}}}{\partial u_r \partial u_s} = \alpha_d \int_{\mathcal{L}} \left(\mathbf{u}_{,s}^{\text{A}} - \mathbf{u}_{,s}^{\text{B}} \right) \cdot \left(\mathbf{u}_{,r}^{\text{A}} - \mathbf{u}_{,r}^{\text{B}} \right) d\mathcal{L} . \quad (4.11)$$

Similarly, the first and second derivatives of the rotation penalty virtual work, Eqs. (4.3) and (4.7), are

$$\frac{\partial W_{G^0}^{\text{pr}}}{\partial u_r} = \alpha_r \int_{\mathcal{L}} \left(\mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} - \mathring{\mathbf{a}}_3^{\text{A}} \cdot \mathring{\mathbf{a}}_3^{\text{B}} \right) \left(\mathbf{a}_{3,r}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} \right) d\mathcal{L} , \quad (4.12)$$

$$\begin{aligned} \frac{\partial W_{G^0}^{\text{pr}}}{\partial u_r \partial u_s} = \alpha_r \int_{\mathcal{L}} & \left(\left(\mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} - \mathring{\mathbf{a}}_3^{\text{A}} \cdot \mathring{\mathbf{a}}_3^{\text{B}} \right) \left(\mathbf{a}_{3,rs}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_{3,r}^{\text{A}} \cdot \mathbf{a}_{3,s}^{\text{B}} + \mathbf{a}_{3,s}^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} + \mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_{3,rs}^{\text{B}} \right) \right. \\ & \left. + \left(\mathbf{a}_{3,s}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_{3,s}^{\text{B}} \right) \left(\mathbf{a}_{3,r}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_3^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} \right) \right) d\mathcal{L} , \end{aligned} \quad (4.13)$$

$$\frac{\partial W_{G^1}^{\text{pr}}}{\partial u_r} = \alpha_r \int_{\mathcal{L}} \left(\mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} - \mathring{\mathbf{a}}_n^{\text{A}} \cdot \mathring{\mathbf{a}}_3^{\text{B}} \right) \left(\mathbf{a}_{n,r}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} \right) d\mathcal{L} , \quad (4.14)$$

$$\begin{aligned} \frac{\partial W_{G^1}^{\text{pr}}}{\partial u_r \partial u_s} = \alpha_r \int_{\mathcal{L}} & \left(\left(\mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} - \mathring{\mathbf{a}}_n^{\text{A}} \cdot \mathring{\mathbf{a}}_3^{\text{B}} \right) \left(\mathbf{a}_{n,rs}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_{n,r}^{\text{A}} \cdot \mathbf{a}_{3,s}^{\text{B}} + \mathbf{a}_{n,s}^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} + \mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_{3,rs}^{\text{B}} \right) \right. \\ & \left. + \left(\mathbf{a}_{n,s}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_{3,s}^{\text{B}} \right) \left(\mathbf{a}_{n,r}^{\text{A}} \cdot \mathbf{a}_3^{\text{B}} + \mathbf{a}_n^{\text{A}} \cdot \mathbf{a}_{3,r}^{\text{B}} \right) \right) d\mathcal{L} . \end{aligned} \quad (4.15)$$

Close investigation of Eqs. (4.13) and (4.15) reveals the importance of having complementary terms for both G^0 and G^1 continuity. Consider patches forming an angle of π

wherein the vectors \mathbf{a}_3^A and \mathbf{a}_3^B are parallel. In this situation, the vector derivative $\mathbf{a}_{3,s}^A$ is orthogonal to \mathbf{a}_3^B and, similarly, $\mathbf{a}_{3,s}^B$ is orthogonal to \mathbf{a}_3^A . Given that the dot product of orthogonal vectors is equal to zero, the entire term $\left(\mathbf{a}_{3,s}^A \cdot \mathbf{a}_3^B + \mathbf{a}_3^A \cdot \mathbf{a}_{3,s}^B\right)$ in Eq. (4.13) would be equal to zero. Further, in geometrically linear analysis and in the first step of geometrically nonlinear analysis, the deformed and undeformed configurations are equivalent and, correspondingly, the term $\left(\mathbf{a}_3^A \cdot \mathbf{a}_3^B - \mathring{\mathbf{a}}_3^A \cdot \mathring{\mathbf{a}}_3^B\right)$ is equal to zero. Thus, in this particular configuration, there is zero penalty stiffness contribution from Eq. (4.13); however, there is a strong penalty stiffness contribution from Eq. (4.15). Conversely, it can be shown that, for configurations wherein patches form an angle of $\pi/2$, the opposite is true: Eq. (4.15) has zero penalty stiffness contribution while Eq. (4.13) has a strong penalty stiffness contribution. The formulations are therefore complementary throughout a range of possible patch angles.

Eqs. (4.14) and (4.15) require the first and second derivatives of \mathbf{a}_n , which are defined below:

$$\mathbf{a}_{t,r} = \frac{1}{\|\tilde{\mathbf{a}}_t\|} \left(\tilde{\mathbf{a}}_{t,r} - (\mathbf{a}_t \cdot \tilde{\mathbf{a}}_{t,r}) \mathbf{a}_t \right) , \quad (4.16)$$

$$\mathbf{a}_{n,r} = \mathbf{a}_{t,r} \times \mathbf{a}_3 + \mathbf{a}_t \times \mathbf{a}_{3,r} , \quad (4.17)$$

and

$$\begin{aligned} \mathbf{a}_{t,rs} = & \frac{1}{\|\tilde{\mathbf{a}}_t\|} (\mathbf{a}_{t,s} \cdot \tilde{\mathbf{a}}_{t,r}) \mathbf{a}_t \\ & + \frac{1}{\|\tilde{\mathbf{a}}_t\|^2} \left(2 (\mathbf{a}_t \cdot \tilde{\mathbf{a}}_{t,r}) (\mathbf{a}_t \cdot \tilde{\mathbf{a}}_{t,s}) \mathbf{a}_t - (\mathbf{a}_t \cdot \tilde{\mathbf{a}}_{t,s}) \tilde{\mathbf{a}}_{t,r} - (\mathbf{a}_t \cdot \tilde{\mathbf{a}}_{t,r}) \tilde{\mathbf{a}}_{t,s} \right) , \end{aligned} \quad (4.18)$$

$$\mathbf{a}_{n,rs} = \mathbf{a}_{t,rs} \times \mathbf{a}_3 + \mathbf{a}_{t,r} \times \mathbf{a}_{3,s} + \mathbf{a}_{t,s} \times \mathbf{a}_{3,r} + \mathbf{a}_t \times \mathbf{a}_{3,rs} . \quad (4.19)$$

Additional details regarding the discretization of the variables used in these expressions can be found in Kiendl [112, pp. 45–48]. Note that, for the ubiquitous case of patch coupling at patch edges, $\tilde{\mathbf{a}}_t$ is simply \mathbf{a}_1 or \mathbf{a}_2 , depending on the edge (\mathbf{a}_1 and \mathbf{a}_2 are generally not unit vectors). Thus, the first and second derivatives of $\tilde{\mathbf{a}}_t$ often correspond to the derivatives of \mathbf{a}_1 or \mathbf{a}_2 . If the penalty formulation is integrated over a trimming curve, it is possible to

use the tangent of the trimming curve’s projection on the patch as $\tilde{\mathbf{a}}_t$. Trimming is outside the scope of the present work and may be investigated in the future.

4.3.2 Implementation

For cases in which the discretization of patches \mathcal{S}^A and \mathcal{S}^B is non-matching along \mathcal{L} , there are various clear possibilities regarding the discretization of \mathcal{L} . For cases which employ patch coupling along patch edges, one can employ the discretization of \mathcal{S}^A , \mathcal{S}^B , or some combination of the two to construct the penalty terms. For the sake of simplicity, the present work employs the discretization of the patch edge which, across the penalty domain \mathcal{L} , has the largest number of elements.

In other configurations, the interface of patches \mathcal{S}^A and \mathcal{S}^B may not be a patch edge. Instead, \mathcal{L} may be an intersection of two NURBS surfaces. In general, it may not be feasible to determine the mathematically exact intersection of the two surfaces; CAD systems are typically tasked with finding approximate, NURBS-curve representations of such intersections according to system- or user-defined tolerances. In this case, the integration domain \mathcal{L} may be defined by the approximate intersection curve and the discretizations of \mathcal{L} , \mathcal{S}^A , and \mathcal{S}^B may not correlate. The proposed methodology is still applicable in such circumstances.

When evaluating Eqs. (4.10)–(4.15), one must integrate over \mathcal{L} whilst incorporating variables defined on both \mathcal{S}^A and \mathcal{S}^B . In the numerical setting, Gaussian integration points are defined along \mathcal{L} . At each integration point, variables with superscripts “A” or “B” are evaluated at the point on either patch \mathcal{S}^A or path \mathcal{S}^B , respectively, which is physically nearest to the integration point on \mathcal{L} . For many simple configurations, the corresponding points on \mathcal{L} and the two patches are exactly co-located.

It is also important to properly assemble the contributions of Eqs. (4.10)–(4.15) with respect to the shell degrees of freedom. This is especially relevant for cases in which the discretization of \mathcal{L} does not match the discretization of one or both of the shell surfaces. In a more traditional approach, one might fully assemble the penalty contributions with

respect to the elements of \mathcal{L} . However, a single element of \mathcal{L} may not precisely correlate with an element in the shells \mathcal{S}^A or \mathcal{S}^B , making proper application of the penalty terms to the shell degrees of freedom difficult. Alternatively, one can assemble the contributions of Eqs. (4.10)–(4.15) on an integration point basis. As described above, each integration point on \mathcal{L} is associated with the nearest points on \mathcal{S}^A and \mathcal{S}^B . For each integration point, the basis function information at each of the points on \mathcal{S}^A and \mathcal{S}^B can be employed to directly apply the penalty contributions to the degrees of freedom of the shell problem.

4.3.3 Penalty parameter formulations

A key drawback of penalty methods is that the penalty parameters must be high enough to ensure constraint satisfaction without creating excessive ill-conditioning. The selection of penalty parameters, usually performed empirically by the analyst, has a strong influence on the solution quality. A common strategy is to directly employ a single value for both displacement and rotation penalties:

$$\alpha_d = \alpha_r = \alpha , \quad (4.20)$$

where α is an adjustable penalty parameter. As will be shown, this strategy does not ensure that a given value of α is appropriate for a variety of problem configurations and, in practice, requires user selection of α based on trial and error.

In this work, it is shown that, rather than requiring user selection of the penalty parameters, the parameters can be formulated according to the problem configuration; that is, according to geometry and material properties. A single value of α , scaled by problem-specific parameters, can then be reliably used for various problems. The most general case considered in this work is the coupling of composite patches with non-matching material definitions at their interface as depicted in Figure 4.1a. The displacement and rotation penalty parameters are formulated with respect to the shell extensional stiffness and bending stiffness, respectively, in order to make the penalty terms dimensionally consistent with

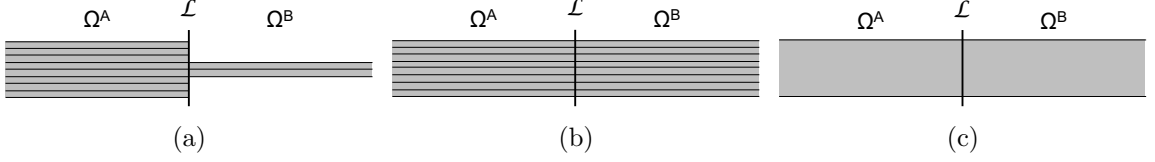


Figure 4.2: Various possible patch connection configurations as indicated by the relative thickness of the coupled patches. The general composite configuration with a material stiffness discontinuity at the coupling location (a); composite configuration with the same stiffness on both patches at the coupling location (b); and isotropic configuration with the same stiffness on both patches at the coupling location (c) are depicted.

the rest of the problem and in order to scale the terms according to kinematically-relevant stiffness properties:

$$\alpha_d = \alpha \frac{\min \left(\max_{i,j} \left(K_{ij}^{\text{exte},A} \right), \max_{i,j} \left(K_{ij}^{\text{exte},B} \right) \right)}{h}, \quad (4.21)$$

$$\alpha_r = \alpha \frac{\min \left(\max_{i,j} \left(K_{ij}^{\text{bend},A} \right), \max_{i,j} \left(K_{ij}^{\text{bend},B} \right) \right)}{h}, \quad (4.22)$$

where α is a penalty parameter; K_{ij}^{exte} and K_{ij}^{bend} are the plate extensional and bending stiffnesses, respectively, derived from laminated plate theory; $h = (h^A + h^B)/2$; h^A and h^B are the lengths of the local elements in the direction most parallel to the penalty curve, \mathcal{L} ; $i = 1, 2$; and $j = 1, 2$. The reason for taking the minimum of the specified stiffness properties on patches \mathcal{S}^A and \mathcal{S}^B in the numerator of these equations is not immediately apparent. A thorough discussion and exploration of this choice will be shown in Section 5.3.5. In particular, it will be shown that this choice can impact the performance of the method for structures with sharp material discontinuities such as wind turbine blades.

For composite materials having the same stiffness on both patches at the coupling loca-

tion, $K_{ij}^{\text{exte,A}} = K_{ij}^{\text{exte,B}}$ and $K_{ij}^{\text{bend,A}} = K_{ij}^{\text{bend,B}}$, as depicted in Figure 4.1b. The formulations therefore reduce to

$$\alpha_d = \alpha \frac{\max_{i,j} \left(K_{ij}^{\text{exte}} \right)}{h} , \quad (4.23)$$

$$\alpha_r = \alpha \frac{\max_{i,j} \left(K_{ij}^{\text{bend}} \right)}{h} . \quad (4.24)$$

For isotropic configurations having the same stiffness on both patches at the coupling location, as depicted in Figure 4.1c, the expressions reduce further:

$$\alpha_d = \alpha \frac{E t}{h (1 - \nu^2)} , \quad (4.25)$$

$$\alpha_r = \alpha \frac{E t^3}{12 h (1 - \nu^2)} , \quad (4.26)$$

where E is Young's modulus, t is the material thickness, and ν is Poisson's ratio.

4.4 Benchmark examples

A variety of geometrically linear and nonlinear benchmark examples are employed to explore the behavior of the coupling methodology and the formulations for the penalty parameters proposed in Eqs. (4.21) and (4.22). Examples featuring different geometries, material properties, and analysis types are selected, and a combination of matching and non-matching discretization strategies are used throughout, all to demonstrate the effectiveness and flexibility of the method. The appropriate range for the master penalty parameter, α , is also explored. The geometrically linear analyses are done by performing only one iteration step of the nonlinear analysis. For all problems, the linear systems are solved using direct solvers. Eigenvalue problems are solved using a SLEPc-based Krylov-Schur solver [127, 128].

4.4.1 Scordelis-Lo roof

The Scordelis-Lo roof is a geometrically linear problem from the well-known shell obstacle course proposed by Belytschko et al. [129] to test accuracy and robustness in complex strain states. The example is used to extensively study the proposed penalty formulation relative to a more naive approach. The Scordelis-Lo roof is a cylindrical shell section as shown in Figure 4.4; the ends of the geometry are supported by rigid diaphragms while the remaining edges are left unconstrained. Figure 4.4 also indicates the dimensions of the geometry. A uniform gravitational load of 90.0 per unit area is applied to the roof, and the resulting linear deformation is quantified by evaluating the vertical displacement of the midpoint of one the edges. For the benchmark problem, the thickness and Young's modulus of the roof are $t = 0.25$ and $E = 4.32 \times 10^8$. Figure 4.4 shows the expected deformation of the roof.

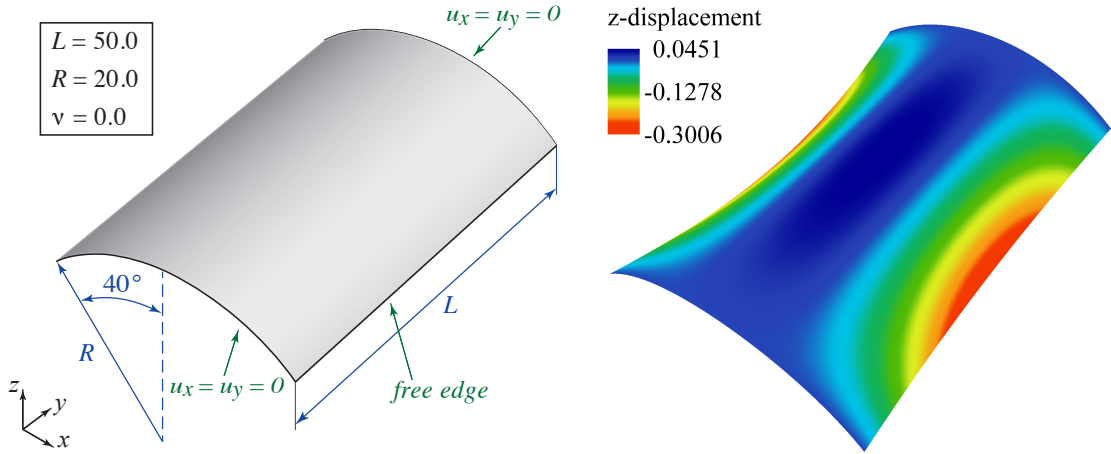


Figure 4.4: Scordelis-Lo roof problem description and deformation (scaled).

The geometry of the Scordelis-Lo roof is modeled using multiple NURBS patches of degree 3 with both matching and non-matching discretizations as shown in Figures 4.5a and 4.5b, respectively. For the purpose of exploring the effective range of α , the deformation of both the matching and non-matching cases is calculated using the proposed penalty parameter formulations, Eqs. (4.21) and (4.22), and the α -only formulation, Eq. (4.20), over

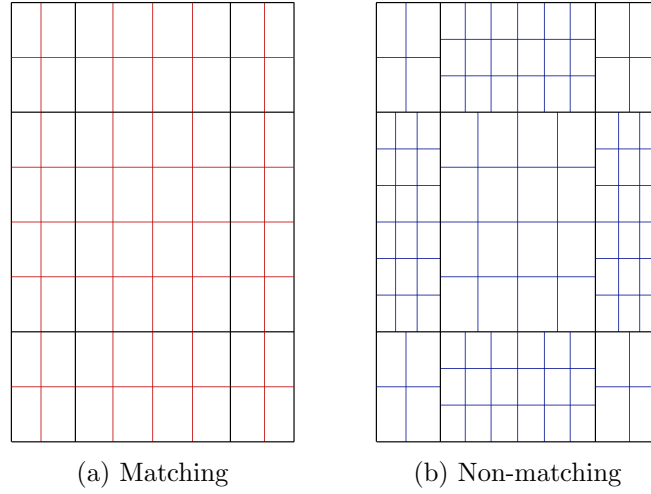


Figure 4.6: Meshes for the matching (a) and non-matching (b) configurations of the Scordelis-Lo roof. Black lines indicate patch boundaries.

a range of values of α . This same study is repeated for cases wherein the shell thickness, t , and Young's modulus, E , have been modified. Throughout these studies, analyses are performed using the meshes shown in Figures 4.5a and 4.5b with 2 levels of h -refinement; the meshes in Figure 4.6 will be used for the convergence study presented later.

The results of the analyses performed using the benchmark material parameters are shown for the proposed penalty formulations in Figure 4.7a and for the α -only penalty formulations in Figure 4.7b. As expected, the results in both cases indicate a range of values of α for which the penalty-based coupling methodology is effective and produces the correct result. Also note that accurate results are obtained for both the matching and non-matching cases. These results clearly show that, if the value of the penalty parameter is too low, the patch coupling constraint is not enforced. If the penalty parameter value is too high, the matrices may become ill-conditioned and the solution of the linear system is prone to large numerical errors. Both scenarios lead to solutions deviating from the reference results. For this reason, the author recommends using a penalty value that is sufficiently high to produce an accurate result but no higher than necessary.

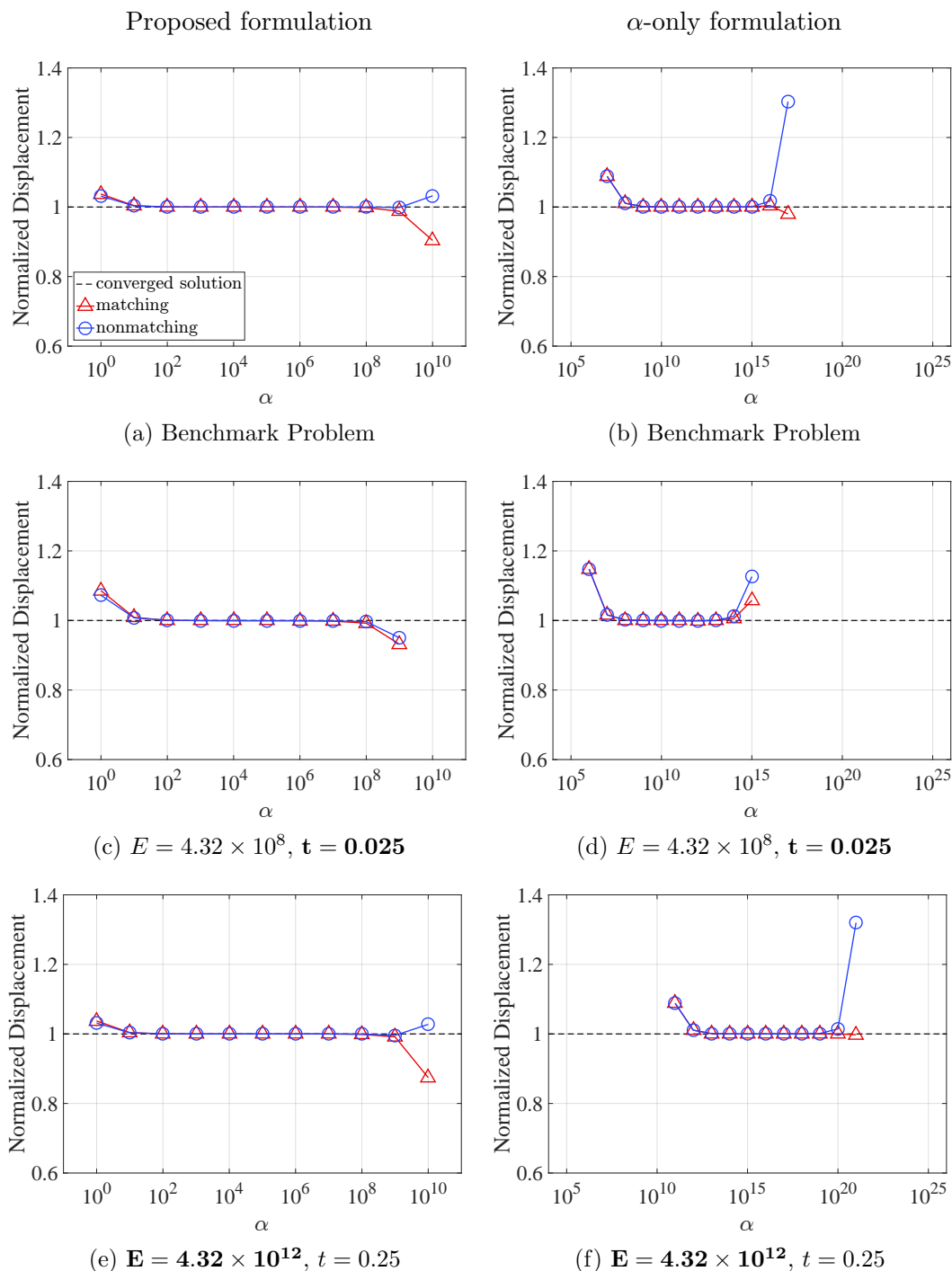


Figure 4.8: Scordelis-Lo roof displacement at midpoint, normalized with respect to reference value, with increasing penalty value α using the proposed penalty approach and the α -only approach for both matching and non-matching discretizations and different combinations of setup variables.

The results in Figures 4.7a and 4.7b alone do not indicate the importance of the proposed penalty formulations. The value can be understood, however, upon performing the same analysis with different sets of material parameters. The results obtained using a thickness of $t = 0.025$ instead of $t = 0.25$ are shown for the proposed and α -only formulations in Figures 4.7c and 4.7d, respectively. Similarly, the results obtained using a Young's modulus of $E = 4.32 \times 10^{12}$ instead of $E = 4.32 \times 10^8$ are shown for the proposed and α -only formulations in Figures 4.7e and 4.7f, respectively.

For the cases employing the α -only penalty formulation, shown in Figures 4.7b, 4.7d, and 4.7f, the range of values of α that produce accurate results shifts significantly, by as much as five orders of magnitude, when the problem parameters are changed. This illustrates one of the key drawbacks of penalty methods: because the effective range of α is problem-dependent, an analyst would be forced to select the penalty parameter based on experience. In contrast, for the cases employing the proposed penalty formulation, shown in Figures 4.7a, 4.7c, and 4.7e, the effective range of α remains consistent for all problem configurations. Specifically, an accurate range of approximately $\alpha = 10^2$ to $\alpha = 10^8$ is observed. Thus, it is recommended that a value of $\alpha = 10^3$ be employed; this value is high enough to reliably produce correct results, but is no higher than necessary.

Solution convergence under mesh refinement can also be demonstrated using the proposed penalty approach. Results are compared to the displacement of a converged, single-patch geometry having a 128×128 element mesh. The coarsest geometries for the matching and non-matching multi-patch configurations are shown in Figures 4.5a and 4.5b, and refinement is performed via global h -refinement. All analyses use $\alpha = 10^3$. The results in Figure 4.9 indicate satisfactory convergence for both the matching and non-matching cases.

4.4.2 T-beam

For the Scordelis-Lo roof example, the patch interfaces having C^0 discontinuity also featured G^1 continuity. Of course, this may not always be the case for general engineering

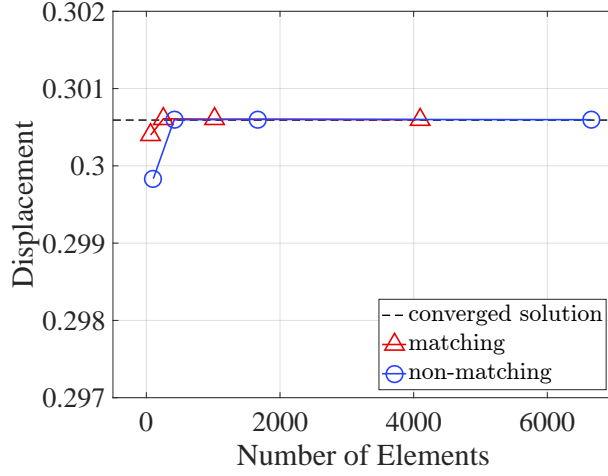


Figure 4.9: Convergence of the roof midpoint displacement under h -refinement for the proposed penalty approach with $\alpha = 10^3$.

applications. Consider, for example, the T-beam geometry presented in Figure 4.11 which features a geometric discontinuity at the patch interface.

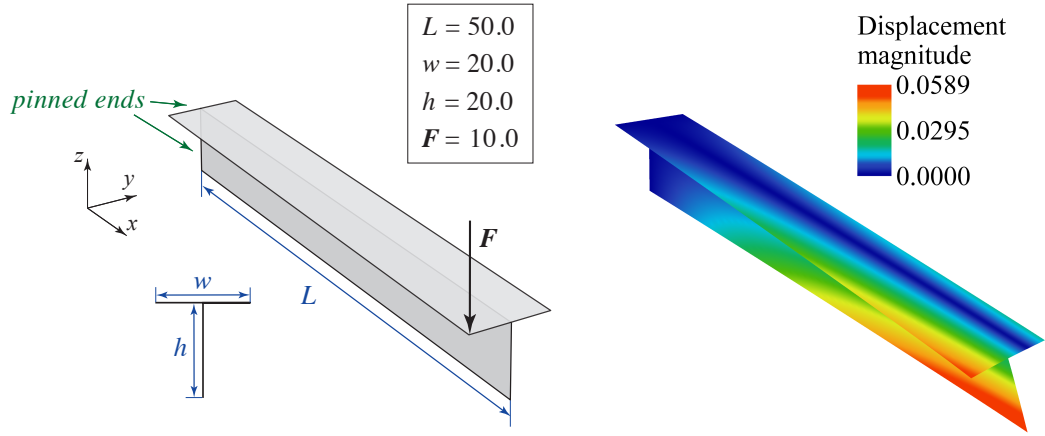


Figure 4.11: T-beam problem description and deformation (scaled).

The T-beam is modeled using two planar geometries which are orthogonal at their interface. As shown in Figure 4.11, one end of the beam is pinned and a force of $F = 10.0$ is applied to one corner of the opposite end in the $-z$ direction. Figure 4.11 also indicates the dimensions of the geometry. The patches have a Young's modulus of $E = 1.0 \times 10^7$, a

thickness of $t = 0.1$, and a Poisson's ratio of $\nu = 0.0$. The expected deformed geometry is also shown in Figure 4.11.

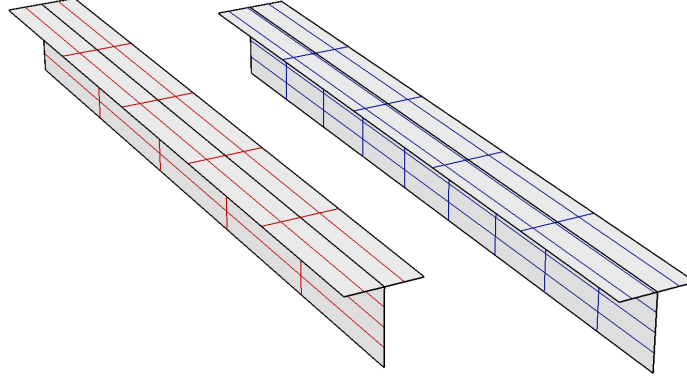


Figure 4.12: Meshes for the matching (left) and non-matching (right) configurations of the T-beam problem. Black lines indicate patch boundaries.

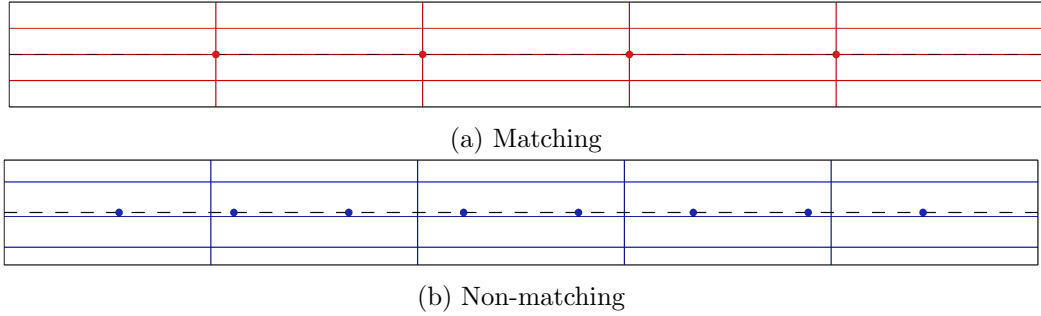


Figure 4.14: Top view of meshes for the matching and non-matching configurations of the T-beam problem. Circular markers indicate discretization of perpendicular patch.

The T-beam is modeled using two NURBS patches of degree 3. Again, both matching and non-matching mesh configurations are constructed as shown in Figures 4.12 and 4.14. For the analyses considered here, two h -refinements are performed on each of the meshes shown in Figures 4.12 and 4.14 (coarser meshes depicted for clarity). Geometrically linear analysis is performed, and the angle between the patches at the end of the beam is calculated. If the patches have been properly coupled, an angle of 90 degrees should be maintained. The left side of Figure 4.16 shows the resultant angle between the two patches for both the matching and non-matching cases for a range of α values. Note that, for relatively low

values of α , the structure is effectively unconstrained at its interface, resulting in an angle of approximately 93.5 degrees between the patches. However, in the range of approximately $\alpha = 10^{-2}$ to $\alpha = 10^2$ the constraint begins to take effect, resulting in the desired angle of 90 degrees between the patches for $\alpha > 10^2$. Again, a wide range of acceptable values of α is observed. In fact, the full range of α values shown to be effective for the Scordelis-Lo problem, $\alpha = 10^2$ to $\alpha = 10^8$, is also shown to be effective for this problem. Thus, the suggested value of $\alpha = 10^3$ remains appropriate in this case.

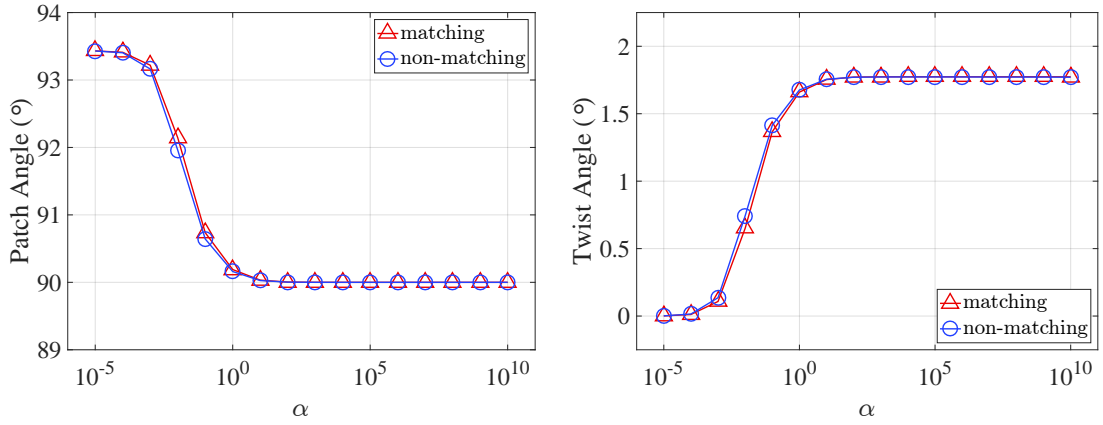


Figure 4.16: Angle between patches of the T-beam (left) and total twist at the end of the vertical patch (right) with increasing penalty value α for both matching and non-matching configurations.

The right side of Figure 4.16 shows the total twist at the free end of the T-beam measured using the vertical patch. When the penalty value is too low, the patches are effectively uncoupled and no strain energy is transferred to the vertical patch, resulting in a twist angle of zero degrees. As the penalty value increases, a consistently reasonable twist angle is observed. Similar results are obtained for both the matching and non-matching configurations.

4.4.3 Plate buckling

Another important type of analysis that is commonly performed in the design of shell structures, such as wind turbine blades, is linear buckling analysis. Thus, the performance

of the proposed penalty formulation is also explored in the context of linear buckling. Linear buckling analysis, or eigenvalue buckling analysis, entails solving the equation

$$\left(\mathbf{K}^{\text{lin}} + \lambda_i \mathbf{K}_g \right) v_i = 0 , \quad (4.27)$$

where \mathbf{K}^{lin} is the linear stiffness matrix of the structure, \mathbf{K}_g is the geometric stiffness matrix based on the applied load, and λ_i is the i^{th} eigenvalue associated with mode vector v_i . The geometric stiffness, \mathbf{K}_g , quantifies the effect of deformation on overall structural stiffness. As such, it is calculated in the deformed configuration after first performing a linear deformation analysis. In this context, an eigenvalue, λ_i , is a scalar multiplier of the applied loads that will, in theory, cause buckling of the structure.

A plate is employed in a simple buckling configuration, as shown in Figure 4.18, with one end clamped and the other supported in the vertical direction. The problem is modeled using both matching and non-matching multi-patch configurations. All of the patches are bivariate cubic NURBS surfaces. The discretization of the patches is shown in Figure 4.20. The patch sizes are intentionally selected such that the patch boundaries do not occur on axes of symmetry. For this problem, Young's modulus is $E = 1.0 \times 10^3$, Poisson's ratio is $\nu = 0.0$, thickness is $t = 0.1$, and the applied distributed force is $P = 1.0$ in terms of force per unit length.

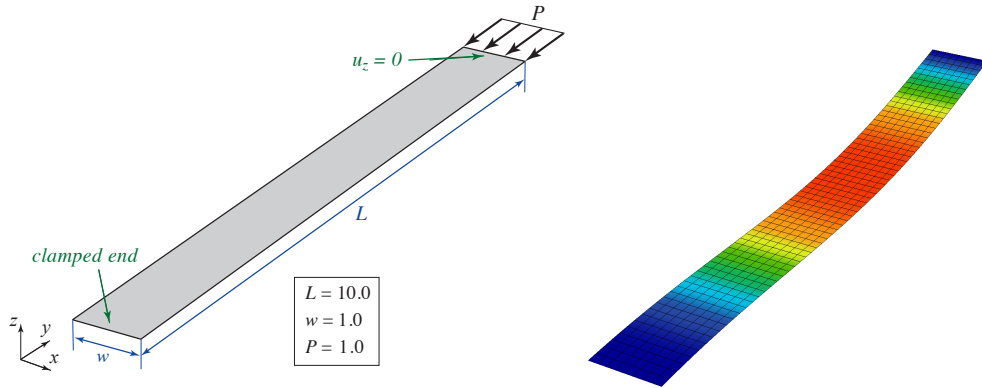


Figure 4.18: Plate buckling problem description and the deformation for the first buckling mode (color contour indicates relative displacement magnitude).

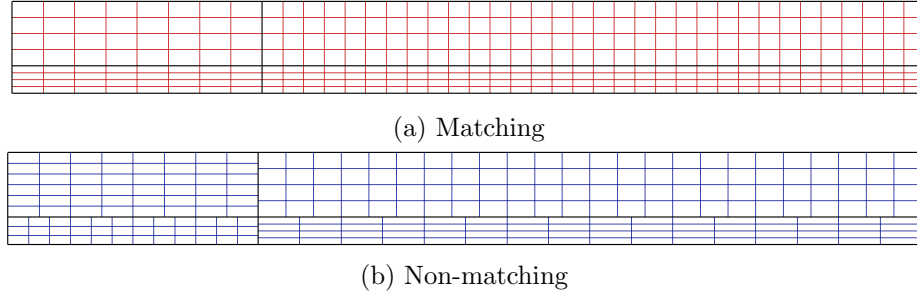


Figure 4.20: Meshes for the (a) matching and (b) non-matching configurations of the plate buckling problem. Black lines indicate patch boundaries.

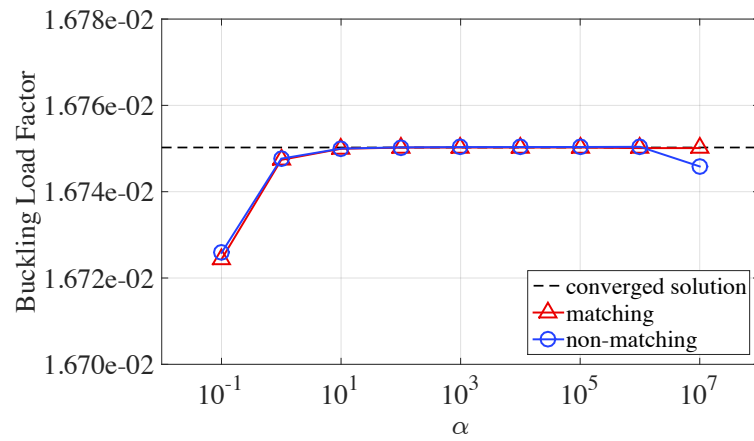


Figure 4.21: Buckling load factor with increasing α for both matching and non-matching configurations of the plate buckling problem.

The plot in Figure 4.21 indicates that, for a similar range of α parameters observed in the previous problems, the multi-patch configurations produce buckling load factors of sufficient accuracy when compared to a converged result obtained using a single-patch configuration. This illustrates the accuracy of the proposed formulation in the context of linear buckling analysis. Also note that the suggested value of $\alpha = 10^3$ is appropriate here.

4.4.4 Nonlinear slit annular plate

All of the examples considered thus far have employed linear analysis. However, the presented methodology is also applicable in the geometrically nonlinear setting. Sze et al. [130] identified and reproduced a number of common benchmark problems for nonlinear analysis, one of which is a slit annular plate subjected to a lifting line force. The slit annular plate problem is reproduced using multi-patch models to verify the proposed coupling methodology in the nonlinear setting.

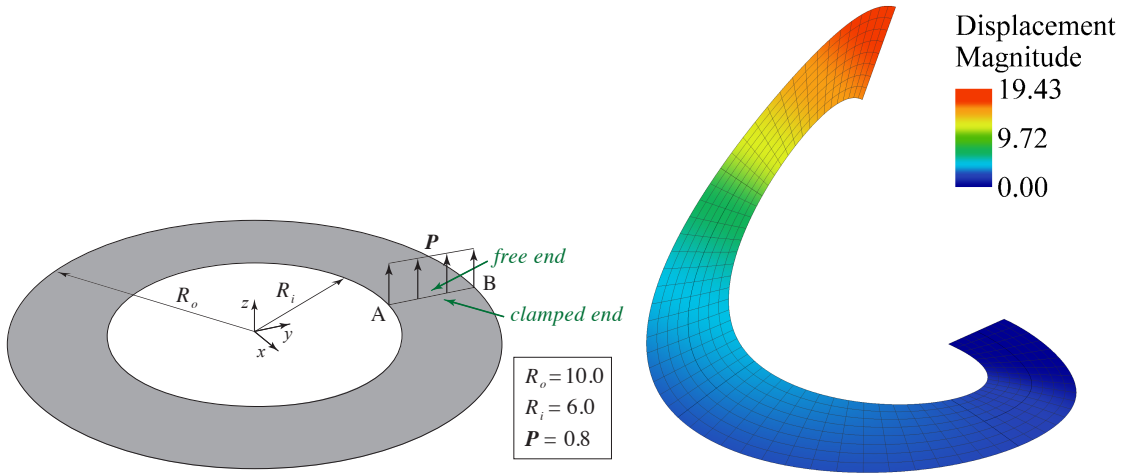


Figure 4.23: Nonlinear slit annular plate problem description and deformation.

The slit annular plate setup and expected deformation are illustrated in Figure 4.23. One side of the slit is clamped, while the other is allowed to freely deform under the applied distributed force, P . The deformation is quantified by tracking the vertical displacement

of points A and B, identified in Figure 4.23, at incremental loads up to $P = 0.8$. Young's modulus is $E = 21.0 \times 10^6$, thickness is $t = 0.03$, and Poisson's ratio is $\nu = 0.0$.

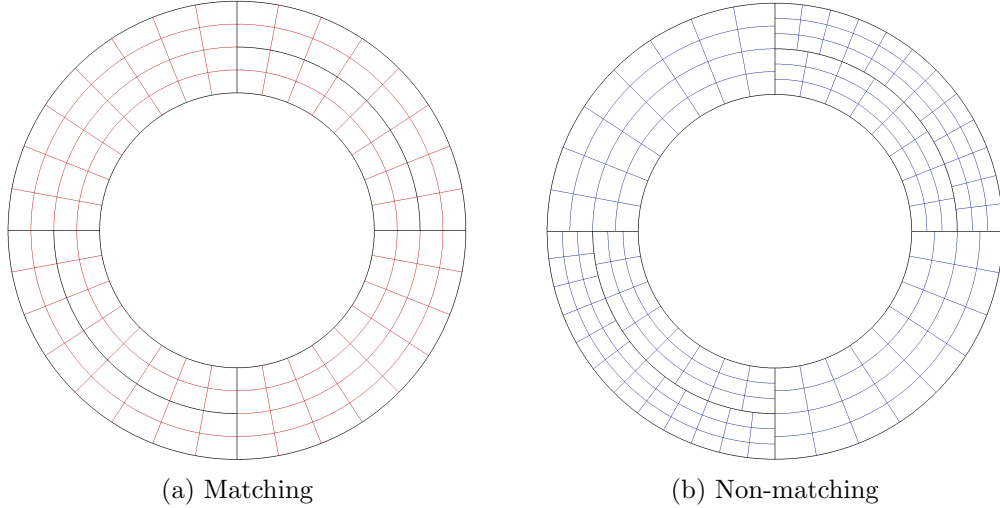


Figure 4.25: Meshes for the (a) matching and (b) non-matching cases of the nonlinear slit annular plate. Black lines indicate patch boundaries.

Refinement studies are not typically performed for the slit annular plate problem in literature. However, because cases with different discretizations are considered in this work, a refinement study is performed for both matching and non-matching configurations to ensure that the results are comparable. For all of the cases, cubic NURBS patches are employed. The coarsest meshes used in the refinement study are shown in Figure 4.25, and the displacement of point B due to the the maximum load of $P = 0.8$ under h -refinement is shown in Figure 4.26. For this analysis, the recommended value of $\alpha = 10^3$ is employed. Convergence is achieved with approximately two h -refinements for both the matching and non-matching cases. Thus, this level of refinement is used for the following verification of α for both matching and non-matching configurations.

Figure 4.28 shows the displacements at A and B for the maximum load of $P = 0.8$ and a range of values of α . The accuracy of the method is clearly demonstrated for $\alpha = 10^2$ through $\alpha = 10^6$, with identical results obtained in both the matching and non-matching cases. In nonlinear analysis, a badly conditioned problem is more likely to exhibit divergent

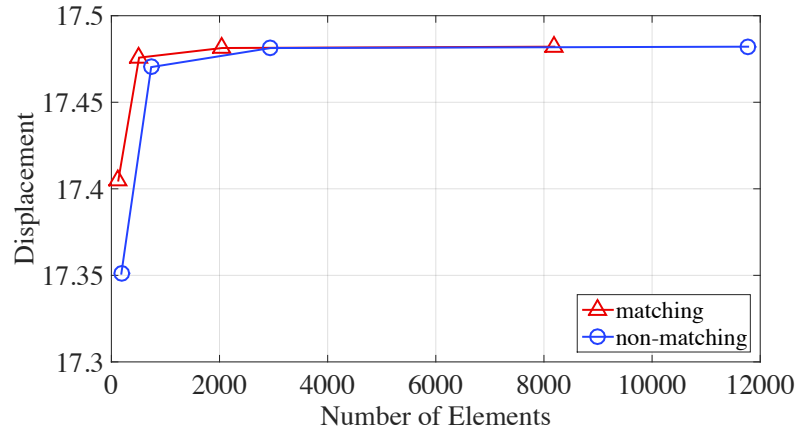


Figure 4.26: Displacement of point B under h -refinement for both matching and non-matching configurations using $\alpha = 10^3$.

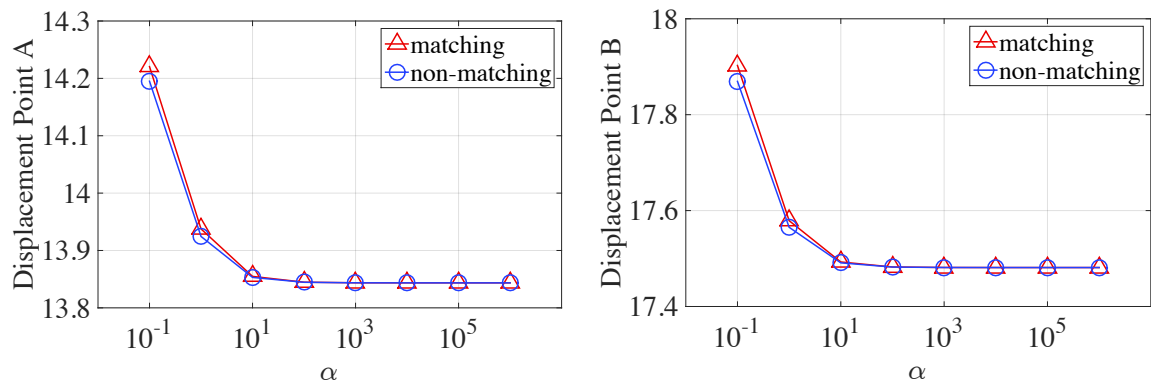


Figure 4.28: Displacement at points A and B of the slit annular plate due to the maximum applied P for varying values of α .

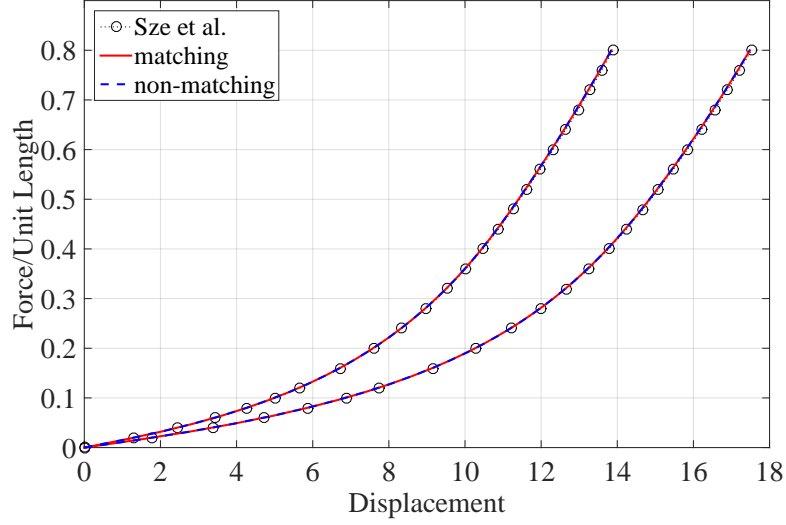


Figure 4.29: Vertical displacement of points A and B versus applied distributed load for the nonlinear slit annular plate with $\alpha = 10^3$.

behavior, as is the case for $\alpha \geq 10^7$ for this problem. Still, the problem is tractable for a wide range of values of α , and the acceptable range is similar to the range observed in the previous examples. The suggestion of $\alpha = 10^3$ remains appropriate.

Figure 4.29 shows the displacements of point A and point B under varying applied forces both for the presented methodology and as reported by Sze et al. [130] using a 6×30 mesh of four-node S4R elements in ABAQUS [131]. A penalty parameter of $\alpha = 10^3$ is used for this comparison, and good agreement with the reference results is observed over the entirety of the load spectrum.

4.5 Chapter conclusion

A new approach for penalty-based coupling of NURBS patches with non-conforming interfaces is proposed. The approach is based on the methodology presented by Duong et al. [126] and is shown to be applicable to non-conforming configurations. Further, formulations for the penalty parameters are proposed. In addition to allowing the penalty parameters

to be dimensionally consistent, the formulations are intended to eliminate the need for experience-based penalty parameter selection for independent problems. The formulations are based on local stiffness properties and are stated for isotropic and composite configurations, as well as for the unique case of composite configurations with stiffness discontinuities at the coupling interface.

The proposed patch coupling approach is demonstrated on a number of benchmark problems from the literature. For all problems, accurate kinematic performance is observed for a relatively consistent range of values of the penalty parameter, α . As a result, it is suggested that a value of $\alpha = 10^3$ be used regardless of problem type or configuration. Through the benchmark problems, the method is shown to be useful for linear, nonlinear, and buckling analyses.

As previously mentioned, one potentially fruitful future use of this methodology would be to use it in the context of trimmed NURBS geometries. For complex geometries, patch intersections are commonly used to trim NURBS patches. In the analysis setting, these interfaces are often considered to be rigidly coupled. Hence, the proposed penalty methodology could be applied to such trimming curves, which are defined in the parametric space of the NURBS surfaces. Of course, some other approach, such as adaptive refinement, would have to be employed to accommodate proper treatment of the trimmed portions of the NURBS surface.

4.6 Acknowledgements

Chapter 4 is, in part, a reprint of material as it appears in: “A patch coupling technique for non-matching isogeometric shells with application to composite wind turbine blades,” (with E.L. Johnson, D. Proserpio, J. Kiendl, and M.-C. Hsu), *Computational Mechanics*, in preparation. The dissertation author was the primary investigator of this paper.

CHAPTER 5. IGA-BASED BLADE DESIGN

5.1 Chapter overview

The methods outlined in Chapters 3 and 4 represent a collection of practical, efficient approaches for generating, analyzing, and optimizing multi-patch parametric geometries using IGA. As discussed in Chapter 1, wind turbine blade design is one area which could greatly benefit from such approaches; thus, in Chapter 5, an overall framework for IGA-based design and optimization of wind turbine blades is presented. The framework integrates the novel IGA methodologies discussed in previous chapters and also, when appropriate, employs reduced-order wind turbine modeling in order to maximize framework flexibility, efficiency, and applicability to modern blade design workflows.

Chapter 5 is outlined as follows. The salient theoretical and technical aspects of the presented framework for IGA-based blade design are reviewed and contextualized in Section 5.2. In Section 5.3, the details of the reference NREL/SNL 5 MW blade model are described, and a NURBS-based model is presented and analyzed. The model is first verified via IGA-based vibration and buckling analysis. Buckling analysis is further employed to perform a mesh refinement study, the results of which are compared to results obtained using traditional FEA. Additionally, the behavior of the proposed penalty parameter formulations relative to some possible alternatives is explored. Lastly, nonlinear deflection analysis is performed. In Section 5.4, the 5 MW blade design is optimized using a combination of reduced-order aeroelastic analyses and IGA-based buckling analysis. Section 5.5 presents conclusions.

5.2 Framework overview

5.2.1 Parametric generation of blade geometry

As Chapter 3 discusses in detail, reliable, parametric generation of IGA-suitable NURBS geometries is a nontrivial task. Given a wind turbine blade's planform, it is critical to employ a set of geometric operations that are applicable across a range of design parameters. Following the work presented in Chapter 3, the CAD software Rhinoceros 3D, a purely NURBS-based platform, and Grasshopper 3D, a visual programming interface for Rhinoceros, are employed. The programmatic approach of Grasshopper allows consistent generation of geometries for a range of parametric inputs whereas its interactive nature enables efficient, intuitive alteration of the underlying algorithms used to generate blade geometries. The algorithm used to generate the wind turbine blade geometry featured in this chapter is fundamentally the same as that shown in Figure 3.4 but with enhancements to facilitate generation of shear web geometry, division of the high- and low-pressure surfaces into distinct material zones, and handling of flatback airfoils. While a single geometry is used in this chapter, the generative algorithm could be used to generate a wide range of blade geometries.

5.2.2 Isogeometric shell analysis

The formulations for isogeometric Kirchhoff–Love shell analysis described in the preceding chapters are employed in the context of IGA-based blade design. Specifically, the virtual work expression given by Eq. (4.8) is employed. Eq. (4.8) has a number of properties that are beneficial in the context of wind turbine blade design. Firstly, because it is based on an isogeometric approach, it enables direct simulation of smooth, NURBS-based blade geometries without further geometry refinement or discretization. Further, because it employs Kirchhoff–Love shell theory, no rotational degrees of freedom are required, limiting the computational cost of the problem. Additionally, as described in Chapter 4, the penalty energies are fully capable of coupling, for example, the shear web, \mathcal{S}^A , and spar, \mathcal{S}^B , of a

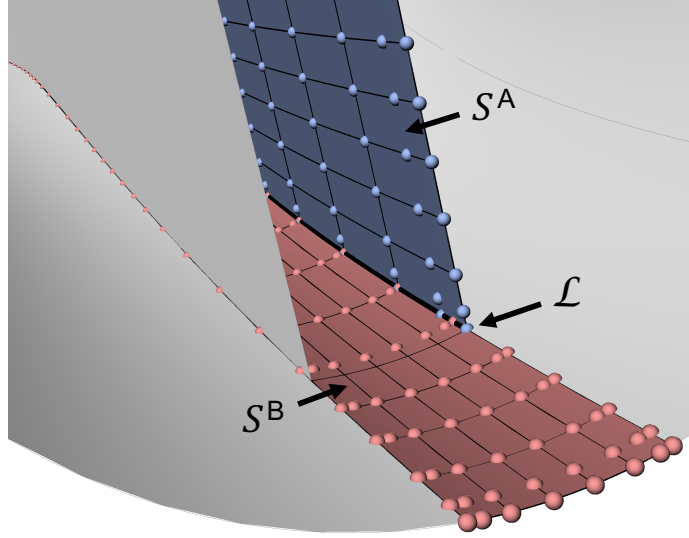


Figure 5.1: Two NURBS surfaces, \mathcal{S}^A and \mathcal{S}^B , representing the shear web and spar cap of a wind turbine blade, which must be coupled along the curve \mathcal{L} . Control point locations indicated by spheres.

wind turbine blade along their interface, \mathcal{L} , even if the patches are not conforming, as is the case for the blade geometry shown in Figure 5.1.

5.2.3 IGA solution strategies

A variety of types of analysis, such as deflection, buckling, and vibration analysis, are typically performed throughout wind turbine blade design. The presented framework for IGA-based blade design has been intentionally developed with such flexibility in mind. Specifically, the in-house developed, Fortran-based IGA code employs the Portable, Extensible Toolkit for Scientific Computation (PETSc) [132, 133], a suite of data structures and routines intended to enable efficient solution of a wide range of computational problems. PETSc provides access to numerous methods for solving linear algebra problems. For example, the use of the penalty method, as specified in Eq. (4.8), can produce ill-conditioned stiffness matrices, suggesting use of one of the direct solvers available in PETSc. Additionally, PETSc provides a convenient interface to the Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems (SLEPc) [127, 128], allowing relatively straightforward and

efficient solution of eigenvalue problems that arise in, for example, vibration analysis and buckling analysis. The use of PETSc and SLEPc ensures that the framework can be tuned and adapted to new types of analyses with relative ease and efficiency.

5.2.4 Integration with aeroelastic simulation

As discussed in Chapter 1, blade analyses of varying levels of fidelity are used throughout blade design. While high-fidelity shell models may be important for performing buckling analysis or for validating lower-fidelity models, many reduced-order models are sufficiently accurate when considering, for example, tip deflection or rotor torque. Advanced aeroelastic codes such as FAST [134] and HAWC2 [135] utilize these reduced-order models to predict overall wind turbine behavior subject to the effects of aerodynamics, structural dynamics, control systems, and other factors. Given the considerable capabilities of these analysis tools, the IGA-based blade design framework is developed as a complement to, rather than as a replacement for, established design and analysis methodologies.

FAST is employed for aeroelastic simulation in this work. Because FAST must be executed numerous times to perform a variety of analyses throughout design, FAST version 8 is implemented in the style of the Framework for Unified Systems Engineering and Design of Wind Plants (FUSED-Wind) [136], an open-source framework for multidisciplinary optimization of wind energy systems developed as an extension to the Python-based, NASA-developed OpenMDAO [137]. This framework allows a designer, within a Python environment, to programmatically dispatch multiple FAST analyses in parallel. Similarly, a Python-based wrapper for the IGA code is developed which allows IGA-based shell analyses to be dispatched using the aerodynamic loads produced in FAST.

5.3 Wind turbine blade analysis

In order to demonstrate the effectiveness of the presented IGA-based approach in the context of blade design, the NREL/SNL 5 MW reference blade is modeled in detail and an-

alyzed. Additionally, because the NREL/SNL 5 MW blade represents a relatively complex, multi-patch engineering structure having a non-uniform composite construction, the blade is used to further explore the effectiveness of the penalty coupling technique introduced in Chapter 4. After defining the reference 5 MW wind turbine blade model in Section 5.3.1, the model is verified through vibration analysis using multiple levels of NURBS discretization and multiple values of the penalty parameter, α , in Section 5.3.2. Next, because buckling analysis is of particular interest, a linear buckling problem is posed and solved in Section 5.3.3. Using buckling analysis, a mesh refinement study is performed in Section 5.3.4; the study underscores the attractive performance of the IGA-based framework relative to traditional FEA methods. In Section 5.3.5, the behavior of the proposed penalty formulations for coupling patches with material discontinuities at the coupling interface is explored relative to some possible alternatives. Lastly, in Section 5.3.6, the challenging case of nonlinear deflection analysis is employed to further demonstrate the effectiveness of the methodology.

5.3.1 NREL/SNL 5 MW blade model

The NREL 5 MW blade was first introduced by Jonkman et al. [115] in the context of a full 5 MW turbine definition intended to be a reference for offshore system development. The turbine is a three-bladed, upwind, variable-speed, variable-pitch machine that is loosely representative of modern utility-scale turbines. The geometric definition of the NREL 5 MW blade itself, however, was only intended to support basic aerodynamic analysis and did not contain enough detail to support construction of a realistic CAD model. Further, only the distributed structural properties of the blade were given, rather than a complete composite layup. Resor [118], as a part of Sandia National Laboratories (SNL), therefore established a rudimentary structural design that approximately reproduces the distributed structural properties described by Jonkman et al. [115]. The design also includes a more detailed geometry definition suitable for generation of high-quality surface geometries. The wind

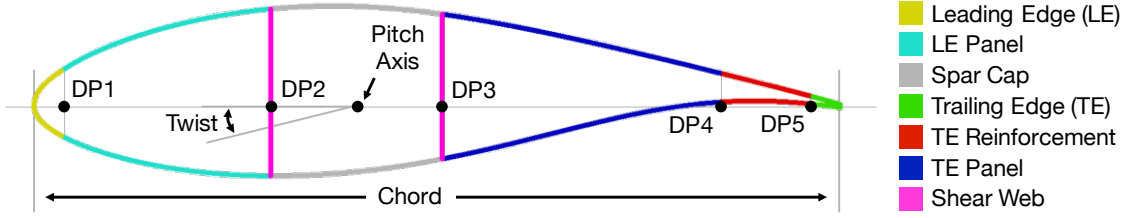


Figure 5.2: Illustration of parameters defined at each station along the blade span in order to define blade geometry and material regions. Parameter values for the NREL/SNL 5 MW blade are given in Table A.1.

turbine blade definition described by Resor et al. [118] will hereafter be referred to as the NREL/SNL 5 MW blade.

The properties of each airfoil cross section describing the NREL/SNL 5 MW blade are given in Table A.1. In addition to defining the properties needed to establish a baseline aerodynamic profile—such as airfoil type, twist degree, and pitch axis location—the table describes the location of the material division points needed to divide the blade into distinct material zones. The parameters that are defined at each spanwise station are illustrated in Figure 5.2.

The parameter values given in Table A.1 are used to develop a NURBS-based shell model consisting of multiple NURBS surfaces, each indicating a material zone with a distinct material stacking sequence. The relatively high spanwise resolution of cross sections in Table A.1 is not actually required to maintain geometric accuracy. However, the modeling software used by Resor [118] assumes that stacks of materials have constant thickness between stations, indicating that many stations must be defined to sufficiently represent high rates of ply drop. In the present work, material thicknesses are defined as piecewise linear functions of blade span which are evaluated at every integration point to determine material distribution. This simplification eliminates the need to use all stations for model construction. More information is given in the footnotes of Table A.1. The baseline IGA model is shown in Figure 5.3 with different colors indicating distinct material zones. The model con-

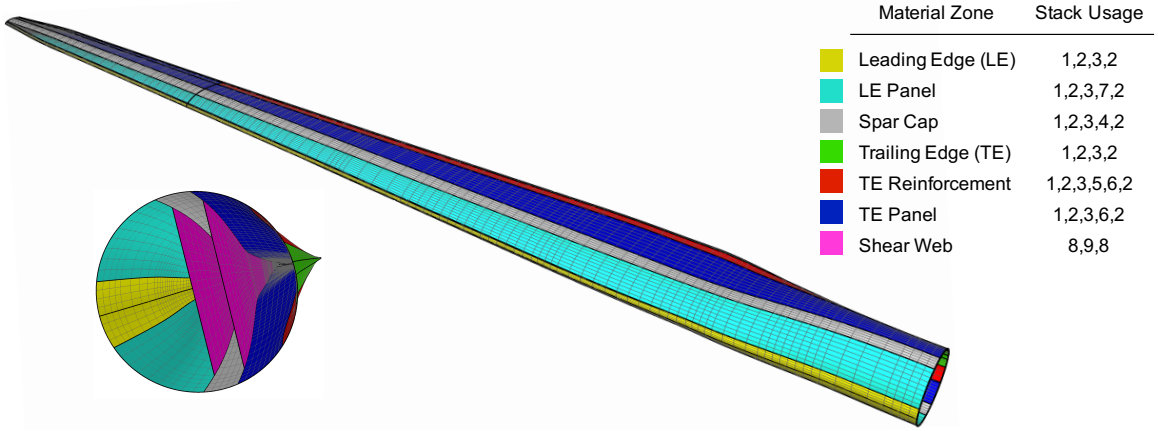


Figure 5.3: A NURBS-based model of the NREL/SNL 5 MW blade. Colors indicate zones with distinct material stacking sequences. Grey lines indicate element edges and black lines indicate surface patch edges. Stacking sequence for each material zone provided; stack definitions can be found in Figure 5.4.

sists of 27 NURBS surfaces of degree 3 in both the u and v directions. The discontinuities present at patch interfaces are used to accurately capture sharp discontinuities in the composite material definition. The shear webs must also be modeled as independent NURBS surfaces. Even in this relatively straightforward geometrical configuration, it is somewhat difficult to ensure conforming spanwise discretization for all patches, highlighting the value of a methodology for coupling non-conforming patches.

Each of the material zones indicated in Figure 5.3 employs a unique sequence of material stacks. Each of these material stacks consists of a single material with a distinct spanwise thickness distribution as defined in Figure 5.4. Material properties are described in Table 5.1 where E_1 and E_2 are the Young's modulus in the first and second material directions, respectively; G_{12} is the shear modulus; ν_{12} is Poisson's ratio; ρ is the density; σ_{UT} is the ultimate tensile strength; and σ_{UC} is the ultimate compressive strength. Further details are given by Resor [118] and Griffith and Ashwill [138]. As Griffith and Ashwill explain, the material properties reported in Table 5.1 for Saertex and SNL Triax represent the homogenized properties of layups having non-uniform, symmetric fiber orientations. Thus,

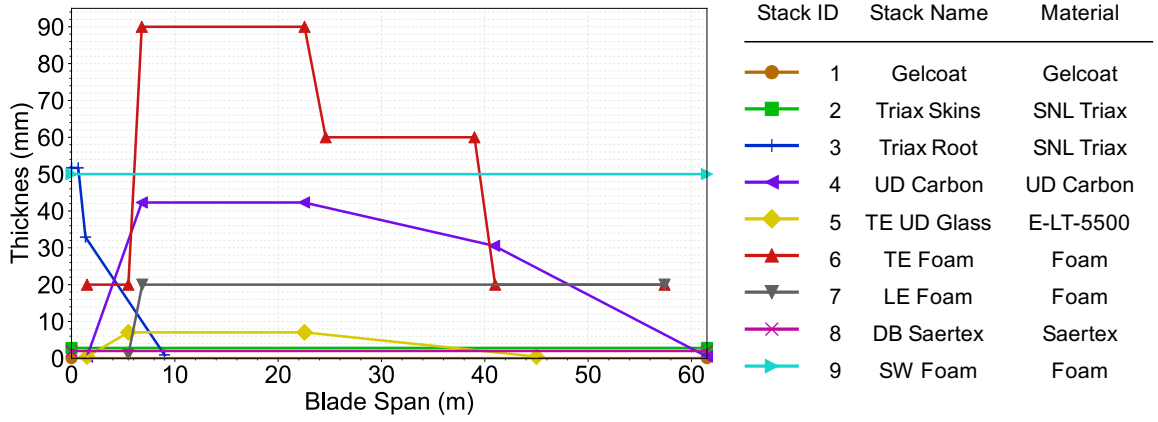


Figure 5.4: Definition of material stack thickness distribution and other relevant information.

in the stacking sequences described in Figure 5.3, these homogenized materials, and all other materials, are assumed to have an overall fiber orientation of zero degrees. In this work, the material axes are aligned with the local Cartesian basis. As a result, the first material direction is aligned with the spanwise parametric direction of the local NURBS surface patch.

Table 5.1: Orthotropic material properties used in the NREL/SNL 5 MW blade design.

Material Name	Layer Thi (mm)	E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12} (-)	ρ (kg/m ³)	σ_{UT} (GPa)	σ_{UC} (GPa)
Gelcoat	0.05	3.440	3.440	1.323	0.30	1235	-	-
E-LT-5500	0.47	41.80	14.00	2.630	0.28	1920	0.972	0.702
SNL Triax	0.94	27.70	13.65	7.200	0.39	1850	0.700	-
Saertex	1.00	13.60	13.30	11.80	0.49	1780	0.144	0.213
Foam	1.00	0.256	0.256	0.022	0.30	200	-	-
UD Carbon	0.47	114.5	8.390	5.990	0.27	1.220	1.546	1.047

5.3.2 Model verification through vibration analysis

Having constructed a model of the NREL/SNL 5 MW blade that is suitable for IGA, a variety of analyses can be performed using the methodology described in Section 5.2. The vibrational frequencies of a wind turbine blade are important to quantify throughout design in order to avoid resonance phenomena. Because vibration analysis incorporates both the mass properties and the stiffness properties of a structure, it is also a good candidate for comparing the presented 5 MW blade model to the reference [118] model. In the finite element context, linear vibration analysis can be performed by considering the eigenvalue problem

$$\left(\mathbf{K}^{\text{lin}} - \lambda_i \mathbf{M}\right) v_i = 0, \quad (5.1)$$

where \mathbf{K}^{lin} is the linear stiffness matrix of the structure, \mathbf{M} is the mass matrix, and λ_i is the i^{th} eigenvalue associated with mode vector v_i . The relation of the i^{th} frequency of vibration, ω_i , to the eigenvalue is given by the equation $\omega_i^2 = \lambda_i$. The eigenvalue problem in Eq. (5.1) is solved using the default SLEPc-based Krylov–Schur solver with a shift-and-invert spectral transformation for extraction of interior eigenvalues [127, 128].

The calculated frequencies of both the reference [118] and the IGA-based model are reported in Table 5.2, with three levels of discretization used for the IGA-based model. Mesh 1 consists of 10,800 cubic NURBS elements and 16,367 control points on the 28 NURBS surfaces. Mesh 2 is obtained by performing a single global h -refinement of Mesh 1, and Mesh 3 is obtained by performing two global h -refinements of Mesh 1. These meshes are shown in Figure 5.5. As discussed previously, the presented model and the reference model use slightly different approaches for capturing material thickness distributions. Additionally, it is likely that some minute geometrical differences exist, and it does not appear that Resor [118] uses a level of discretization that produces fully-converged solutions. Some differences in computed frequencies of vibration are therefore expected. As Table 5.2 shows, however, the first six computed frequencies all agree reasonably well, with a largest difference of

Table 5.2: Frequencies of vibration of the reference model and the IGA-based model for three different levels of discretization. Mesh 1 indicates the coarsest discretization and Mesh 3 indicates the finest. Blade modes, v_i , depicted below for Mesh 1.

	Ref [118]	Mesh 1		Mesh 2		Mesh 3	
		Freq (Hz)	Diff (%)	Freq (Hz)	Diff (%)	Freq (Hz)	Diff (%)
1st flapwise ^a	0.87	0.919	5.63	0.919	5.63	0.919	5.63
1st edgewise ^b	1.06	1.054	0.57	1.054	0.57	1.053	0.66
2nd flapwise ^c	2.68	2.809	4.81	2.808	4.78	2.808	4.78
2nd edgewise ^d	3.91	3.886	0.61	3.884	0.66	3.883	0.69
3rd flapwise ^e	5.57	5.666	1.72	5.660	1.62	5.658	1.58
1st torsion ^f	6.45	6.698	3.84	6.694	3.78	6.692	3.75

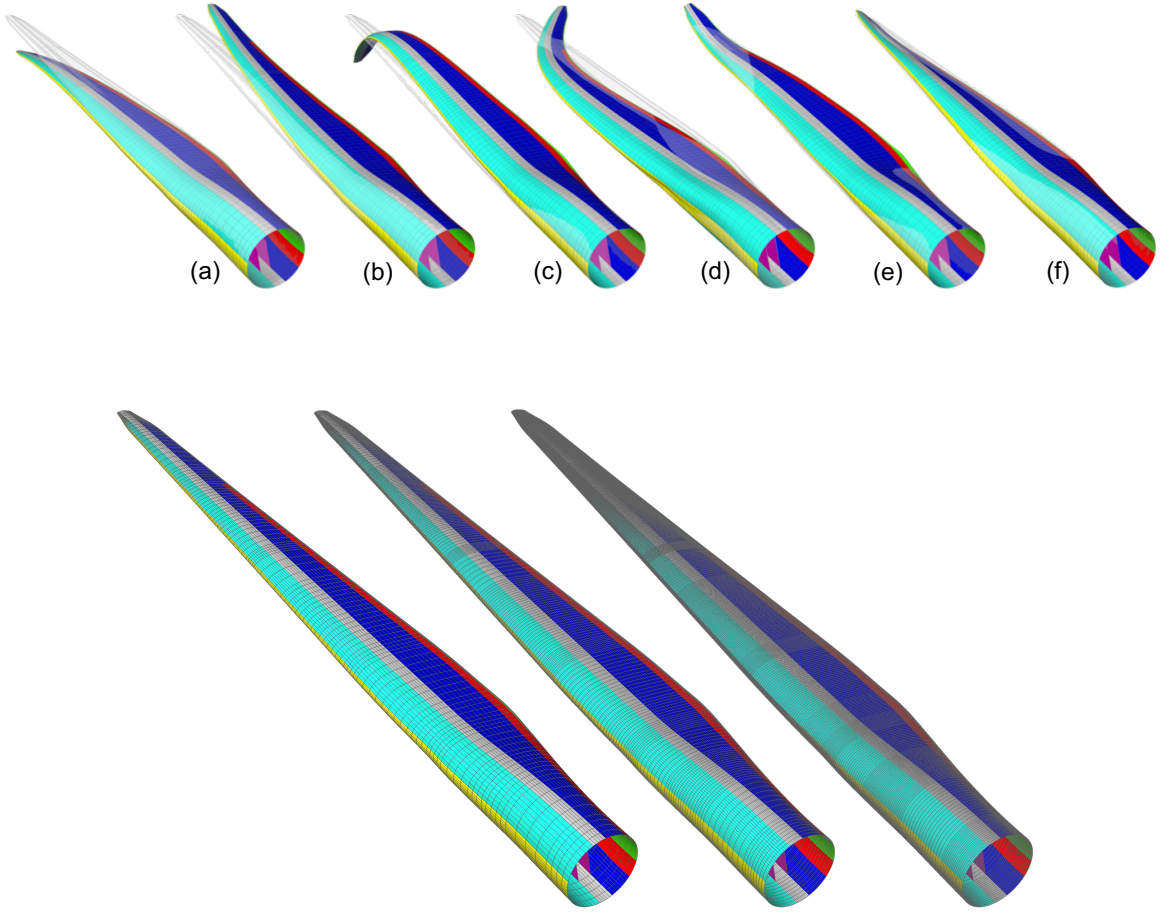


Figure 5.5: Mesh densities for IGA Meshes 1, 2, and 3.

5.63%, or approximately 0.05 Hz, and a lowest difference of 0.61%. Thus, the IGA-based model is considered valid and Mesh 1 is considered to have sufficiently fine discretization.

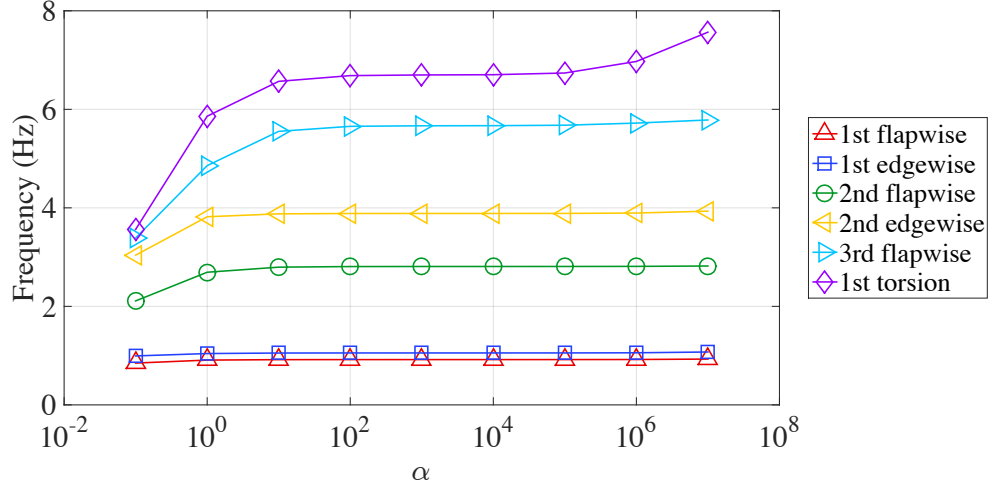


Figure 5.6: NREL/SNL 5 MW blade frequencies of vibration using a range of values for α .

In order to demonstrate the effectiveness of the coupling methodology in the context of vibration analysis of a wind turbine blade, vibration results using a range of α values and Mesh 1 are shown in Figure 5.6. Even in the more complex case of wind turbine blade analysis, consistent behavior is seen over a range of values of α , and $\alpha = 10^3$ appears to remain an appropriate choice for the penalty parameter.

5.3.3 Buckling analysis

As discussed in Chapter 1, buckling load factor prediction is one of the key applications of high-fidelity structural analysis in blade design. Linear buckling analysis, or eigenvalue buckling analysis, entails solving Eq. (4.27); additional details are given in Section 4.4.3. The PETSc-based `SuperLU_DIST` [139] direct solver is employed to solve an initial linear problem derived from Eq. (4.8) prior to constructing \mathbf{K}_g . The eigenvalue problem posed in Eq. (4.27) is addressed by again employing the default SLEPc-based Krylov–Schur solver with a shift-and-invert spectral transformation [127, 128].

Comparison of IGA buckling results to the reference [118] is complicated by the fact that buckling analysis is load-dependent. Load distributions are not provided by Resor [118]; hence, aeroelastic simulations, from which distributed loads can be extracted, are performed. An extreme 50-year wind load case—which corresponds to 70 m/s winds, a fixed rotor, and blades feathered to 90 degrees—with 15 degrees of yaw error is simulated, and the aerodynamic loads associated with the time at which the flapwise root bending moment is greatest are collected. The loads are distributed to all blade surfaces evenly in the chordwise direction and varying in the spanwise direction according to the load profile produced by aeroelastic analysis. Within the IGA-based framework, a buckling factor of 1.61 is calculated. This factor is within 2% of the lowest buckling factor of 1.64 reported by Resor for the same load case [118].

5.3.4 Comparative mesh refinement study

A mesh refinement study using linear vibration analysis was already presented in Section 5.1 and Table 5.2. However, Resor [118] performs a mesh refinement study via linear buckling analysis. Thus, in this section, a similar, IGA-based mesh refinement study is performed for comparison.

Resor [118] records the maximum flapwise bending moment that occurs at each station throughout numerous aerodynamic load cases and then, in buckling analysis, applies a load distribution to recreate this maximum bending moment distribution. While this is an ideal approach, it requires significant additional processing of time histories, loads, and applied forces. Thus, for this study, the loads corresponding to the largest flapwise root bending moment are employed, and it is assumed that the use of somewhat different loading scenarios for the reference blade and the presented blade has a negligible effect with respect to solution convergence. According to both the analyses performed by Resor [118] and analyses performed in this work, the largest bending moment occurs during an extreme 50-year wind with a parked rotor and 15 degrees of yaw misalignment. Resor [118] performs mesh

refinement by incrementally reducing the target element size used in a meshing algorithm, whereas refinement in the IGA framework is performed via h -refinement. The three IGA meshes used in this study correspond to Meshes 1, 2, and 3 referenced in Table 5.3.2 and depicted in Figure 5.5.

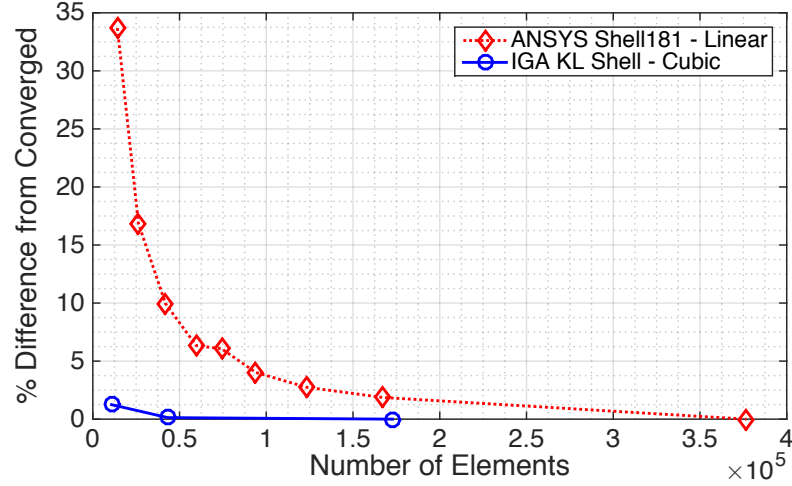


Figure 5.7: Convergence of lowest predicted buckling load factor under mesh refinement relative to the most converged solution for both IGA computations (blue) and computations performed by Resor [118] (red).

Figure 5.7 shows the results of the two mesh refinement studies. The resultant buckling loads are normalized with respect to the most refined solutions. It should be noted that the results in Figure 5.7 do not constitute a one-to-one comparison of traditional FEA to IGA. The element type employed by Resor [118], ANSYS Shell181, is a four-node, thick-shell, linear element with six DOF at each node, whereas the NURBS-based model employs cubic NURBS with three DOF at each node and a thin-shell formulation. The total number of DOF used in an analysis is significant because it indicates the size of the linear algebra problem that must be solved; smaller problems can typically be solved more quickly. Resor [118] only provides the number of elements used in the reference analysis, and the number of elements and DOF cannot be directly related without any knowledge of the mesh topology. Still, the relative number of elements is at least indicative of the relative number of DOF, especially for meshes with fairly uniform discretization.

The results in Figure 5.7 demonstrate the efficiency of the presented IGA approach relative to a more traditional FEA approach. For the coarsest ANSYS-based case having approximately 14,200 elements, an error approximately 34% is observed. In sharp contrast, when using 10,800 elements in the IGA framework an error of only 1.25% is observed, a level of error that is not achieved even by using more than 160,000 ANSYS Shell181 elements. Additionally, moderate refinement quickly drives error towards zero for isogeometric analysis.

5.3.5 Alternative penalty parameter formulations

All of the test examples for the penalty methodology considered in Chapter 4 were isotropic configurations, for which the penalty formulations are given by Eqs. (4.25) and (4.26). The formulations must therefore be further tested in the context of composite materials. Additionally, some composite structures, especially wind turbine blades, may have non-uniform composite definitions. Material discontinuities can occur in between, for example, a blade's spar cap—which has a thick, stiff material definition—and the leading and trailing edge panels, which are comparatively weak. It is therefore necessary to have penalty parameter formulations capable of addressing this discontinuity, such as Eqs. (4.21) and (4.22).

The performance of Eqs. (4.21) and (4.22) relative to a number of possible alternatives is explored in this section. Linear buckling analysis is employed for this study. The same loads that were used in the previous sections are used here. The various approaches that are considered for the penalty parameter formulations in the case of non-uniform composite configurations are described below. In the following, i and j are restricted to $i = 1, 2$ and $j = 1, 2$.

- **Minimum (Proposed):** The minimum local stiffness between patches \mathcal{S}^A and \mathcal{S}^B is selected. The intent of this method is to yield a penalty value that is sufficiently high locally without producing penalty values that are excessively high with respect

to less stiff portions of the model. The formulations for this approach are given by Eqs. (4.21) and (4.22).

- **Maximum:** In this method, the maximum local stiffness between patches \mathcal{S}^A and \mathcal{S}_B is selected. This method prioritizes maximizing the influence of the penalty locally.

$$\alpha_d = \alpha \frac{\max \left(\max_{i,j} \left(K_{ij}^{\text{exte},A} \right), \max_{i,j} \left(K_{ij}^{\text{exte},B} \right) \right)}{h}, \quad (5.2)$$

$$\alpha_r = \alpha \frac{\max \left(\max_{i,j} \left(K_{ij}^{\text{bend},A} \right), \max_{i,j} \left(K_{ij}^{\text{bend},B} \right) \right)}{h}. \quad (5.3)$$

- **Average:** This method dictates that, between patches \mathcal{S}^A and \mathcal{S}_B , the average local stiffness should be used in the penalty formulations. This approach seeks to allow the material properties of both patches to influence the penalty parameter.

$$\alpha_d = \alpha \frac{\max_{i,j} \left(K_{ij}^{\text{exte},A} \right) + \max_{i,j} \left(K_{ij}^{\text{exte},B} \right)}{2h}, \quad (5.4)$$

$$\alpha_r = \alpha \frac{\max_{i,j} \left(K_{ij}^{\text{bend},A} \right) + \max_{i,j} \left(K_{ij}^{\text{bend},B} \right)}{2h}. \quad (5.5)$$

- **Global Minimum:** Unlike the previous approaches, which are based on the local material properties of patches \mathcal{S}^A and \mathcal{S}_B , this method formulates the penalty at all locations according to the minimum stiffness values found throughout the entire structure. The intent of this approach is to prevent extremely high local penalty values which might negatively impact conditioning.

$$\alpha_d = \alpha \frac{\min_{\text{global}} \left(\max_{i,j} \left(K_{ij}^{\text{exte}} \right) \right)}{h}, \quad (5.6)$$

$$\alpha_r = \alpha \frac{\min_{\text{global}} \left(\max_{i,j} \left(K_{ij}^{\text{bend}} \right) \right)}{h}. \quad (5.7)$$

- **Minimum Transverse:** In this approach, the material matrices are rotated such that the e_1 direction of the rotated matrices, $K^{\text{exte}'}$ and $K^{\text{bend}'}$, is consistent with

the local tangential direction of the penalty curve. Then, the stiffness transverse to the penalty curve, or $K_{22}^{\text{exte}'}$ and $K_{22}^{\text{bend}'}$, can be directly used in the formulation. Between the two patches, the minimum $K_{22}^{\text{exte}'}$ or $K_{22}^{\text{bend}'}$ is selected, citing the same logic employed in the “Minimum” approach.

$$\alpha_d = \alpha \frac{\min \left(K_{22}^{\text{exte},A'}, K_{22}^{\text{exte},B'} \right)}{h}, \quad (5.8)$$

$$\alpha_r = \alpha \frac{\min \left(K_{22}^{\text{bend},A'}, K_{22}^{\text{bend},B'} \right)}{h}. \quad (5.9)$$

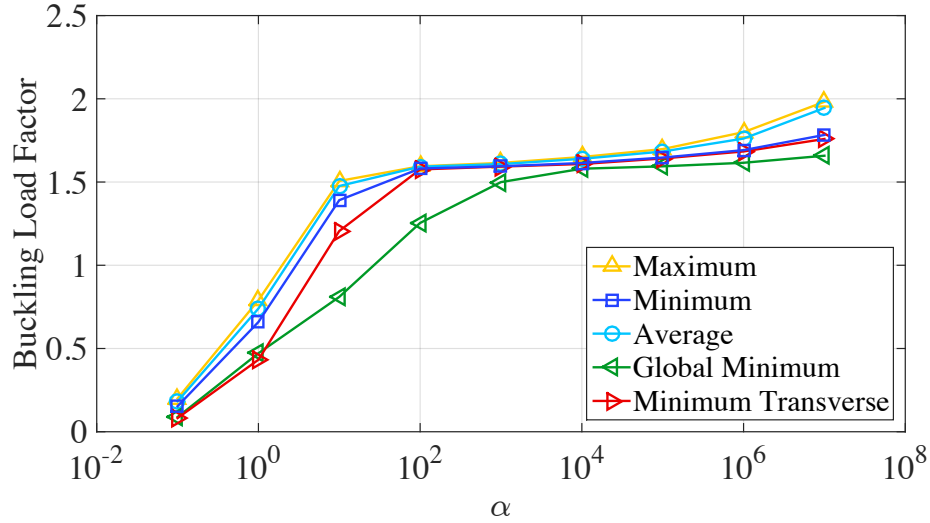


Figure 5.8: Lowest buckling load factor as a function of α for the various possible methods of formulating the penalty parameters.

Each of these methods is used with a range of α values in buckling analysis of the NREL/SNL 5 MW blade, as shown in Figure 5.8. Due to the technical challenges presented in this problem, such as the large number of penalty coupling curves (51) and highly non-uniform material definition across the blade structure, one should not expect a level of α parameter flexibility comparable to that seen in previous benchmark examples. Still, four of the five methods feature a plateau at around $\alpha = 10^2$ to $\alpha = 10^4$, reinforcing the validity of using $\alpha = 10^3$ in general.

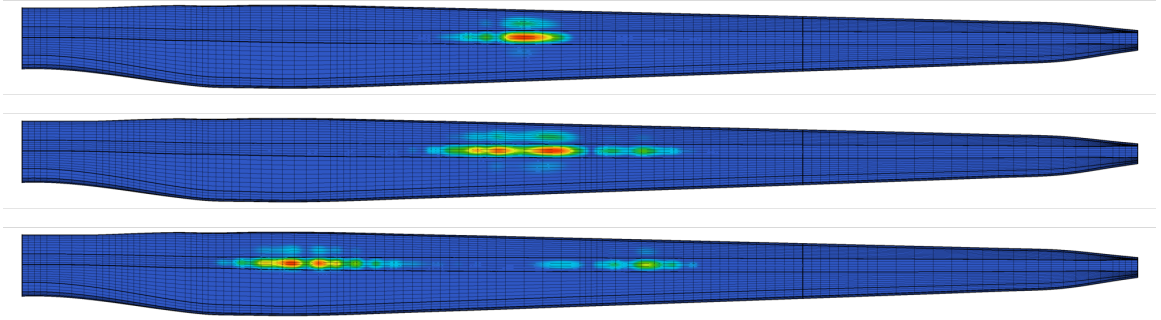


Figure 5.9: First (top), second (middle), and third (bottom) buckling modes of the NREL/SNL 5 MW blade using $\alpha = 10^3$. Color contour indicates relative magnitude of deflection in buckling.

While the “Global Minimum” method also features a plateau, the corresponding values of α are much higher than those suggested in previous benchmarks, which is an undesirable inconsistency. Additionally, the method does not appear to present any advantage in regards to the rate of decreasing accuracy with increasing α . Among the remaining methods, both the “Maximum” and “Average” methods are shown to decrease in accuracy more quickly than the “Minimum” and “Minimum Transverse” methods with increasing α . Because the “Minimum” and “Minimum Transverse” methods utilize similar logic and exhibit comparable performance, the author recommends using the “Minimum” method for the sake of implementation simplicity. The first three buckling modes obtained using this approach and $\alpha = 10^3$ are visualized in Figure 5.9 using the visualization methodology presented in Section 3.3.5 .

5.3.6 Nonlinear deflection analysis

Maximum blade tip deflection is another important consideration in wind turbine blade design. While reduced-order models are usually fairly reliable for tip deflection prediction, finite element models can be used for verification. Additionally, nonlinear deformation analysis of a realistic blade design serves as another test of the proposed IGA methodology.

The same loads that are used in the previous sections are applied and nonlinear deformation analysis is performed for a range of α values, the results of which shown in Figure 5.10.

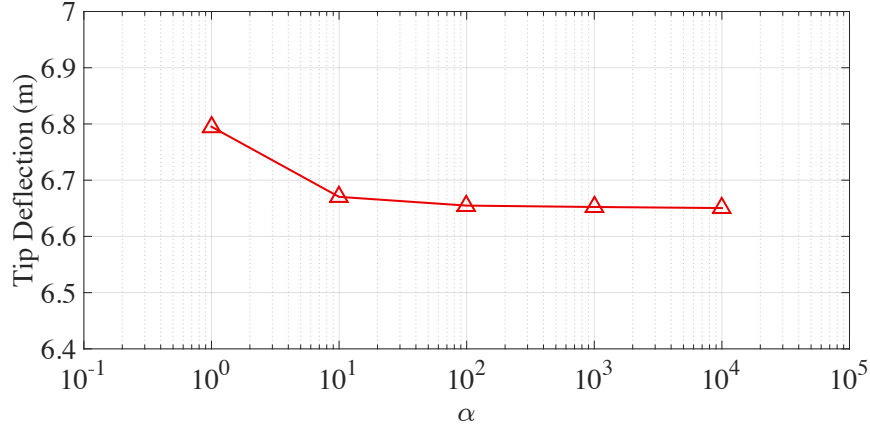


Figure 5.10: Maximum flapwise tip deflection due to nonlinear analysis for a range of values of α .

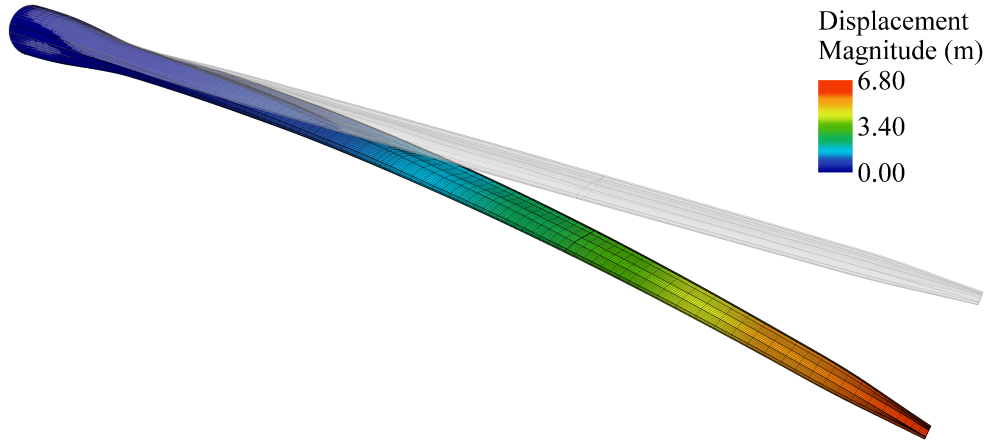


Figure 5.11: Deformation of the NREL/SNL 5 MW blade due to nonlinear analysis using $\alpha = 10^3$.

Because a large number of penalty curves and a variable composite material definition are used, nonlinear convergence is more difficult to achieve for excessively large values of α . Still, a consistent value for tip deflection can be seen from approximately $\alpha = 10^2$ to $\alpha = 10^4$. From $\alpha = 10^2$ to $\alpha = 10^3$, a change in tip deflection of only 0.037% is observed. Similarly, from $\alpha = 10^3$ to $\alpha = 10^4$, a change in tip deflection of less than 0.03% is observed. This

reinforces the validity of using $\alpha = 10^3$ in general and indicates that the proposed method is appropriate for nonlinear analysis of structures featuring material stiffness discontinuities at patch coupling interfaces. The blade deformation obtained using $\alpha = 10^3$ is depicted in Figure 5.11 for which the visualization methodology described in Section 3.3.5 was again employed.

5.4 Wind turbine blade optimization

The great efficiency of the IGA-based framework for wind turbine blade analysis underscores its unique applicability in the context of iterative analysis. Not only can the use of IGA provide time savings, but it can also provide benefits in regards to other practical issues such as mesh generation, data transfer, and data storage. Thus, this section utilizes the aforementioned framework to perform IGA-buckling-constrained optimization of the blade mass and material cost of the NREL/SNL 5 MW blade.

5.4.1 Aeroelastic simulation

A critical task in wind turbine blade design is quantification of the aerodynamic loads, a complicated endeavor due to the interdependence of aerodynamic and structural factors. For the present optimization problem, NREL’s FAST is employed in conjunction with high-fidelity IGA. Load analysis is performed using the methodology presented by Resor [118]. Resor [118] identifies a subset of IEC 61400 [18] design load cases (DLCs) that are considered to be the most likely design drivers for a 5 MW wind turbine blade: DLCs 1.2, 1.3, 1.4, 1.5, 1.6, and 6.3.

For the purposes of comparison, the evaluation of these load cases, as outlined by Resor [118], is reproduced as nearly as possible using FAST. While the exact results obtained differ somewhat from those reported by Resor [118]—assumedly because of the slightly different material distribution, different version of FAST, and potentially different simulation settings that are not described by Resor [118]—the overall trends and conclusions corroborate the

reference analysis. According to the FAST analyses, neither ultimate stress nor fatigue are design-governing, with all stress values falling below the design limits for each material and Miner’s fatigue analysis indicating a material life of well over 20 years. A maximum tip deflection of 6.34 m is observed during a load case featuring a negative gust at rated speed (ECD-R), whereas a maximum flapwise moment of 23,130 kN is observed during the 50-year extreme wind load case with positive 15 degrees of yaw misalignment (EWM50+15).

Given that the NREL/SNL 5 MW reference blade design appears to be governed by the EWM50+15 and ECD-R load cases, only these cases are utilized in the optimization problem, significantly reducing the number of aeroelastic simulations that must be performed for each design iteration. For each set of design parameters, the procedure for determining the objective function value is: (1) calculate the reduced-order blade beam properties using NREL’s PreComp [140], (2) use FAST to perform aeroelastic analysis of the EWM50+15 and ECD-R load cases and calculate maximum stresses and tip deflection, (3) extract instantaneous aerodynamic loads corresponding to the maximum root bending moment in both the flapwise and edgewise directions, (4) perform two independent IGA-based buckling analyses using the loads extracted in the previous step, and (5) calculate the value of the objective function using constraints, calculated via FAST and IGA, and blade mass and material cost values calculated within the IGA code. The details of these procedures are given below.

5.4.2 Objective function

The ultimate goal throughout the design of any wind turbine component is to reduce the overall cost of energy. Quantifying the effect of design changes at the component level on the overall cost of energy, however, is a nontrivial task that requires considerable systems-level design and analysis capabilities. Thus, a simpler approach is adopted in which both blade mass and blade material cost are to be reduced. Both mass reduction and material cost reduction could have a beneficial impact on cost of energy; blade mass reduction entails

reduced loads on the nacelle and tower, which could lead to reduced cost overall, whereas blade material cost is more obviously a contributor to turbine capital cost.

The optimization problem is posed as follows:

$$\begin{aligned}
& \text{minimize} && \mathcal{J}^c(\boldsymbol{\chi}^c) \\
& \text{subject to} && \boldsymbol{\chi}^c \in \boldsymbol{\Omega}^c, \\
& && \mathcal{C}_i^c(\boldsymbol{\chi}^c) \leq 0, \quad i = 1, \dots, n_c,
\end{aligned} \tag{5.10}$$

where $\boldsymbol{\chi}^c$ are the design variables; $\mathcal{J}^c(\boldsymbol{\chi}^c)$ is the objective function; $\mathcal{C}_i^c(\boldsymbol{\chi}^c)$ are the n_c inequality constraints, and $\boldsymbol{\Omega}^c$ are the acceptable ranges for the design variables. The objective function is defined

$$\mathcal{J}^c(\boldsymbol{\chi}^c) = w \frac{M(\boldsymbol{\chi}^c) - M_{\text{ref}}}{M_{\text{ref}}} + (1 - w) \frac{C(\boldsymbol{\chi}^c) - C_{\text{ref}}}{C_{\text{ref}}}, \tag{5.11}$$

where $M(\boldsymbol{\chi}^c)$ is the blade mass for a given set of design variables, M_{ref} is the mass of the reference blade design, $C(\boldsymbol{\chi}^c)$ is the total blade material cost for a given set of design variables, C_{ref} is the reference total material cost, and w is a weighting variable indicating the relative importance of mass and cost reduction, respectively. For this example, a weighting of $w = 0.5$ is used such that mass reduction and material cost reduction are valued equally. Blade mass is calculated using the various material densities given by Resor [118]. The cost of the majority of the materials is dictated by the price per kilogram reported by Griffith and Johanns [141], summarized in Table 5.4. Fiberglass materials are priced as dry fibers; the fiber volume fractions and the cost of resin can be used to determine the infused cost. Alternatively, foam core is priced first on the basis of kitted area and then according to material thickness.

5.4.3 Constraints

One can restate the constrained optimization problem, Eq. (5.10), as an unconstrained optimization problem by incorporating the constraints into the original objective function

Table 5.4: The price of the various materials used in the NREL/SNL 5 MW blade design. Fiberglass costs (E-LT-5500, SNL Triax, and Saertex) are based on dry fiber material whereas UD Carbon material is pre-impregnated.

Material	Price (\$/kg)	Kitting Cost (\$/m ²)	Thickness Cost (\$/mm)
Foam	-	20.00	0.50
Gelcoat	14.00	-	-
E-LT-5500	2.97	-	-
SNL Triax	2.97	-	-
Saertex	2.97	-	-
Resin	4.65	-	-
UD Carbon	26.40	-	-

using the exterior penalty method [34]. This yields a pseudo-objective function

$$\Phi(\boldsymbol{\chi}^c) = w \frac{M(\boldsymbol{\chi}^c) - M_{\text{ref}}}{M_{\text{ref}}} + (1 - w) \frac{C(\boldsymbol{\chi}^c) - C_{\text{ref}}}{C_{\text{ref}}} + \sum_{i=1}^{n_c} \beta_i \max(0, \mathcal{C}_i^c(\boldsymbol{\chi}^c))^2, \quad (5.12)$$

where β_i is a large penalty parameter. An unconstrained optimization problem is ultimately solved:

$$\begin{aligned} & \text{minimize} && \Phi(\boldsymbol{\chi}^c) \\ & \text{subject to} && \boldsymbol{\chi}^c \in \boldsymbol{\Omega}^c. \end{aligned} \quad (5.13)$$

This exterior penalty strategy presents a straightforward but effective way to incorporate constraints into the objective function. The strategy is also beneficial in that it greatly increases the number of optimization methods that can be used to solve the problem.

Upon simulating the EWM50+15 and ECD-R load cases in FAST, many of the constraints, $\mathcal{C}_i^c(\boldsymbol{\chi}^c)$, can be directly calculated. One set of constraints is used to state that the maximum stresses in each structural material across the entirety of the blade should not exceed their maximum tensile strengths with respect to a load factor:

$$\mathcal{C}_j^c(\boldsymbol{\chi}^c) = \sigma_{\max_j}(\boldsymbol{\chi}^c) - \frac{\sigma_{UT_j}}{\gamma_u} \leq 0, \quad j = 1, \dots, n_m, \quad (5.14)$$

where σ_{\max_j} is the maximum stress experienced by each material at any point in time, σ_{UT_j} is the ultimate tensile strength of the material, n_m is the number of materials for

which failure is a design concern, and γ_u is the safety factor used for evaluation of ultimate strength. For this work, $\gamma_u = 1.755$ [118]. Stresses are calculated using the approach described by Resor [118].

A tower clearance constraint can be similarly defined:

$$\mathcal{C}_{n_m+1}^c(\boldsymbol{\chi}^c) = \delta_{\text{tip}}^c(\boldsymbol{\chi}^c) - \frac{\delta_{\text{avail}}}{\gamma_t} \leq 0 , \quad (5.15)$$

where δ_{tip} is the maximum out-of-plane displacement experienced by any of the blades throughout all simulations, δ_{avail} is the available tower clearance, and γ_t is the safety factor used for tip clearance evaluation. Here, $\delta_{\text{avail}} = 10.50$ m and $\gamma_t = 1.485$ [118].

Lastly, after extracting the aerodynamic loads that produce the maximum bending moment in both the flapwise and edgewise directions, IGA-based linear buckling analysis can be performed. The corresponding constraint is given as

$$\mathcal{C}_{n_m+2}^c(\boldsymbol{\chi}^c) = \gamma_b - \min(\lambda_{\text{flap}}(\boldsymbol{\chi}^c), \lambda_{\text{edge}}(\boldsymbol{\chi}^c)) \leq 0 , \quad (5.16)$$

where λ_{flap} is the lowest buckling load factor as a result of the loads corresponding to the maximum flapwise bending moment, λ_{edge} is the lowest buckling load factor as a result of the loads corresponding to the maximum edgewise bending moment, and γ_b is the factor of safety used for linear buckling evaluation; in this case, $\gamma_b = 1.62$ [118].

5.4.4 Design variables

The selection of design variables is governed by both practical constraints and the design objectives. For this problem, a relatively flexible design space is created within which the thickness of many of the material stacks can be varied at different regions over the blade span. Some stack definitions remain invariant, such as the external gelcoat, which does not play an important role in blade structure and which would not be modified for structural purposes, and the triax skins, which are thin layers of fiberglass used consistently throughout the blade structure to maintain general structural integrity. Additionally, the thickness of

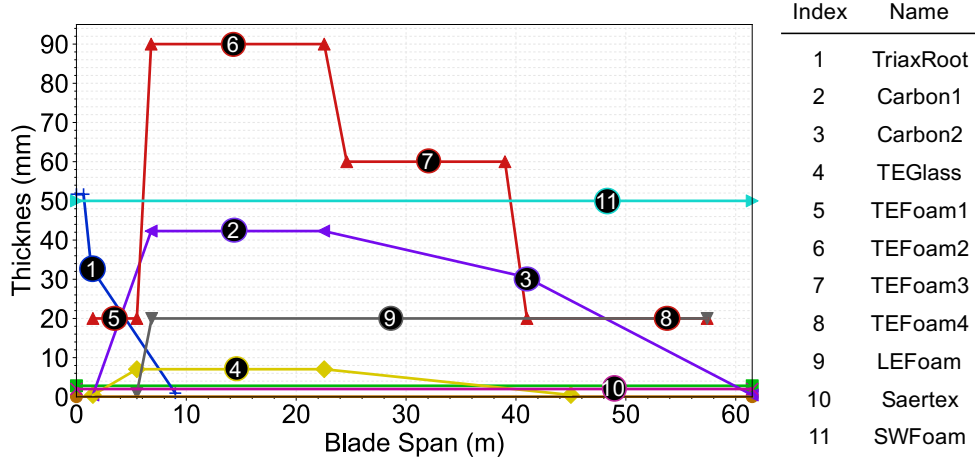


Figure 5.12: Material stack thickness distributions with design variables identified by numerical markers. For variables with markers on horizontal line segments, thickness variation is applied to the line segment end points. For variables with markers directly on points, thickness variation is applied directly to the point. Indices of the variables Δt_i and corresponding names given at right.

the fiberglass at the root is not modified as it is assumed that the reference thickness is defined to account for the installation of bolts.

Figure 5.12 identifies the $n_v = 11$ design variables that are selected. Each variable of index i is associated with either one or two of the thickness distribution definition points illustrated in Figure 5.12. For variables indicated by markers in the middle of a line segment, the thickness of the two associated end points is modified by a common variable Δt_i . For variables indicated by markers directly overlapping a point, the thickness at that single point is modified by a variable Δt_i . This is illustrated by the expression

$$t_i = t_{\text{ref}} + \Delta t_i, \quad i = 1, \dots, n_v, \quad (5.17)$$

where t_i denotes the thickness at the point or points associated with design variable i for a given optimization iteration, t_{ref} denotes the reference thickness of the corresponding point or points, and Δt_i denotes the value of the design variable for a given optimization iteration. The ranges of the design variables are shown in Figure 5.14.

5.4.5 Implementation and solution strategies

The optimization problem represented by Eq. (5.13) is implemented in OpenMDAO [137], a Python-based software for multidisciplinary design analysis and optimization. FAST-based aeroelastic analyses are performed through the OpenMDAO-based FUSED-Wind [136]; thus, the analyses can be performed in parallel. A similar module is developed to dispatch the two IGA-based buckling analyses in serial from within OpenMDAO. Using this setup, each function evaluation takes approximately 12 minutes on two cores of a Linux machine with Intel Xeon E5-2699 v3 2.30 GHz processors. Each optimization problem is solved using two cores, with clear possibilities for improved efficiency and parallelism in the future. OpenMDAO provides a variety of optimization methods as well as the ability to automatically calculate finite-difference-based gradients. The pyOpt-based [142] Sequential Least Squares Programming (SLSQP) method is selected for its relative robustness and ability to incorporate gradient information. A penalty coefficient of $\beta_i = 1000$ is used for all constraints.

Experience indicates that the solution space of this particular optimization problem is quite complicated, with a large number of local minima. Because the SLSQP method features no mechanisms for escaping regions of local minima and because a more thorough exploration of the design space is desired, multiple optimizations are performed with different starting points. Specifically, Latin hypercube sampling, which has been shown to be more effective than purely random sampling [143], is used to generate 15 distinct starting points for the 11 design variables.

5.4.6 Results and discussion

The initial mass and material cost of each sample point, as well as associated optimized masses and material costs, are plotted in Figure 5.13. The average number of function evaluations for each optimization is approximately 424. The wide variety of optimized designs that are generated confirm that the function space has many local minima, some of

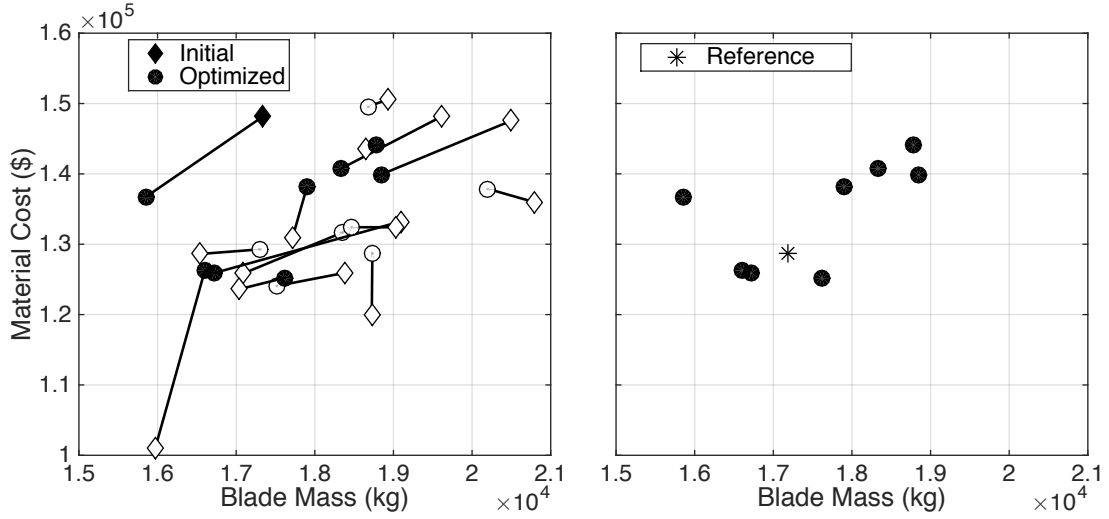


Figure 5.13: At left, blade mass and material cost of each starting point (diamonds) and blade mass and material cost of corresponding optimized designs (circles). Unfilled points signify invalid designs as indicated by an objective function penalization of greater than 0.05. At right, blade mass and material cost of only valid optimized points and reference design [118].

which do not perform well from a blade mass and material cost perspective, and underscores the value of utilizing many starting points. In general, however, one can see that, in the majority of the problems, either mass is decreased, material cost is decreased, the level of constraint violation is decreased, or some combination of these pseudo-objective reduction mechanisms is achieved.

Detailed results of two of the optimizations are given in Table 5.5, and the values of the corresponding design variables are shown in Figure 5.14. The two designs in Table 5.5 are among the best-performing optimized designs and both of them, according to the objective function, Eq. (5.12), perform better than the baseline 5 MW blade design. However, performance improvement is achieved differently in each of these two designs. In the design designated here as the “balanced” design, both blade mass and material cost are reduced, demonstrating that it is possible to achieve improvement of both blade mass and material cost metrics simultaneously. This balanced design technically performs the best from the perspective of Eq. (5.12). The “lightweight” design, however, demonstrates improvement

Table 5.5: Performance details of two of the best-performing optimized designs.

Design Name	Initial Objective	Final Objective	Relative Mass (%)	Relative Cost (%)	Function Evaluations
Balanced	1086.29	-0.026	- 3.39	- 1.80	498
Lightweight	0.08	-0.007	- 7.74	+ 6.32	266

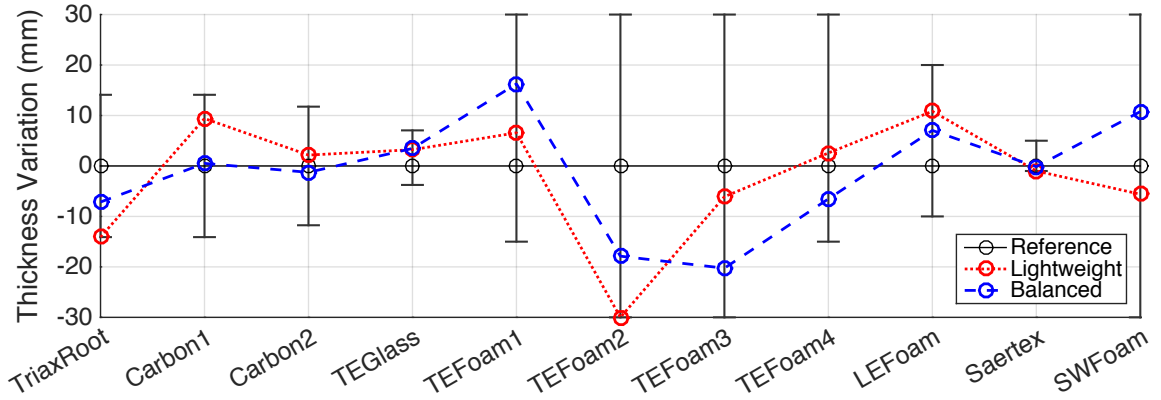


Figure 5.14: Thickness design variable values for the reference, balanced, and lightweight 5 MW blade designs. Bars indicate the allowable range for each design variable.

only of the blade mass metric, with an accompanying material cost increase. While material cost increase is not desirable from the perspective of the objective function, the drastic blade mass reduction of 7.74 % could enable even more cost reduction in downstream components such as the drivetrain or the tower. Such interactions are notoriously difficult to accurately capture on the systems level. Thus, different optimized designs may be preferable for different turbine platforms or to satisfy unique design goals.

Distinct performance metrics are fundamentally achieved by the different material distributions generated by the design variable values in Figure 5.14. The left side of Figure 5.15 shows, for the baseline, balanced, and lightweight 5 MW designs, the total mass of the three primary types of materials: fiberglass, which includes E-LT-5500, SNL Triax, Saertex, and resin; foam; and uni-directional carbon fiber. Similarly, the right side of Figure 5.15 quantifies the contribution of each type of material towards overall material cost in each of the

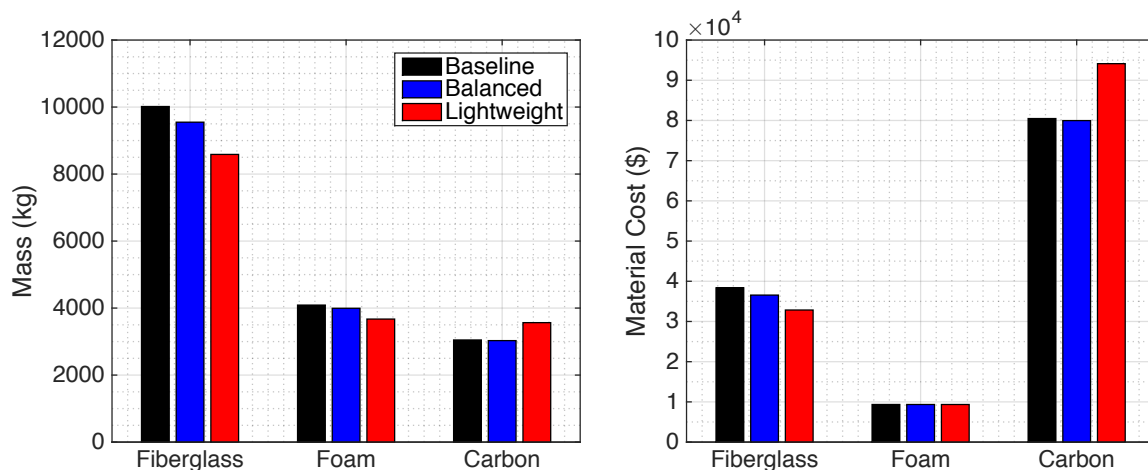


Figure 5.15: The total mass and material cost contributions of three primary types of materials—fiberglass, which includes E-LT-5500, SNL Triax, Saertex, and resin; foam; and carbon—to the baseline, balanced, and lightweight 5 MW blade designs.

three designs. For the balanced design, the mass of all three types of materials is reduced relative to the baseline design. In this case, buckling resistance is improved by more optimally distributing fiberglass and foam. Because the amount of each type of material is reduced in the balanced design, the overall material cost is also reduced.

The lightweight design demonstrates a substantially different approach. In the lightweight design, the amount of carbon is increased, enabling reductions in the amounts of both fiberglass and foam that are used. Carbon’s high strength-to-weight ratio dictates that this exchange is quite beneficial in terms of blade mass, enabling a 7.74% mass reduction, or over 1300 kg. Due to the high cost of carbon, however, the overall material cost is adversely affected, even despite cost reduction stemming from reduced fiberglass mass. Interestingly, the total cost of foam experiences little variation from case to case despite non-zero variables associated with foam thickness. This is because the overall cost that is attributed to foam is dominated by kitting cost, which is determined by the total area over which the foam is applied and which is invariant in this optimization problem.

Table 5.6: Various performance metrics of the reference and optimized 5 MW blade designs.

Design Name	Mass (kg)	Material Cost (\$)	Buckling % of SF	% Max Deflection	% σ_{UT}
Reference	17,184	128,618	100.41	90.71	37.50
Balanced	16,602	126,300	100.02	92.06	37.10
Lightweight	15,853	136,745	100.01	81.73	47.69

The results also illustrate the value of incorporating high-fidelity, IGA-based buckling analysis into such an optimization problem. Table 5.6 shows additional performance details of the reference, balanced, and lightweight designs, including absolute mass, absolute material cost, and relevant constraint values. The buckling load factor for both of the optimized designs is approximately 100% of the design buckling safety factor of 1.62, indicating that the buckling constraint is active. In other words, the 5 MW blade design is governed primarily by buckling in all configurations presented here. If optimization were to be performed without considering this high-fidelity analysis feedback, it is likely that this behavior would not be adequately captured, potentially increasing the workload in later design stages. Each of the two optimizations discussed here took approximately 2.5 days, a reasonable time given the modest computational power utilized.

The first buckling mode for each of the reference, balanced, and lightweight blade designs is shown in Figure 5.16. In the reference design, buckling occurs in the spar cap at approximately midspan. This result is reasonably consistent with the reference [118, Table 29]. In the balanced design, the first buckling mode remains concentrated in the spar cap but is shifted outboard, presumably due to the slightly reduced thickness of carbon fiber beyond 22.5 m as a result of a negative “Carbon2” variation as shown in Figure 5.14. Conversely, the lightweight design shows considerably different buckling behavior, with the first buckling mode concentrated on the inboard region of the shear web nearest to the trailing edge. This is reasonable behavior given the increased thickness of carbon fiber in the spar cap and reduced thickness of both foam and Saertex fiberglass in the shear web.

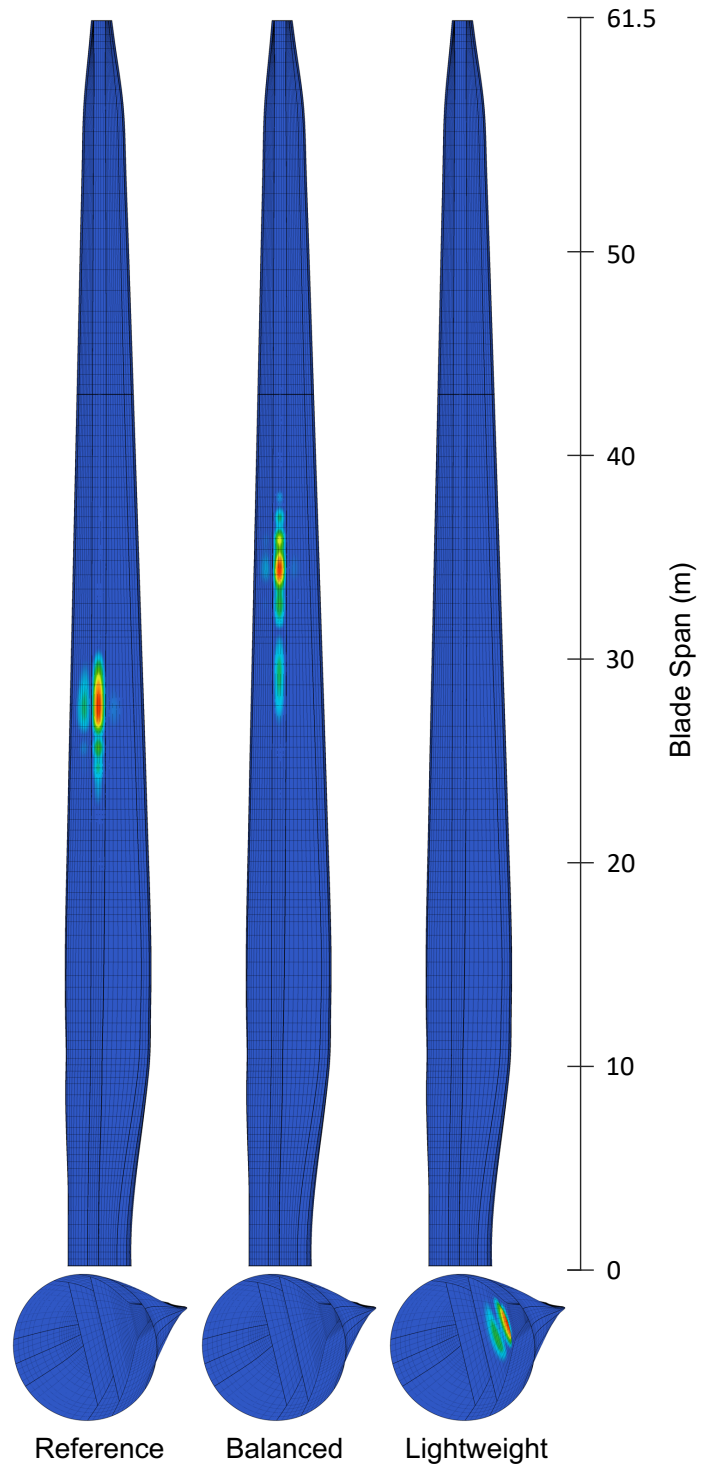


Figure 5.16: The first buckling mode of the reference (left), balanced (middle), and lightweight (right) 5 MW blade designs. Modes are shown in the internal structure of the blade (bottom) and on the suction side of the blade (top).

5.5 Chapter conclusion

A framework is presented through which IGA-based structural analysis of wind turbine blades can be easily performed. The framework employs the efficient IGA Kirchhoff-Love shell formulation for composites and a penalty energy coupling technique to accommodate multi-patch geometry construction. The technique is capable of coupling shell interfaces with non-matching control points and discretization, eliminating the restrictive requirements imposed on the geometry construction by other methods.

For the purposes of demonstration, the NREL/SNL 5 MW blade is modeled within Rhinoceros 3D, a NURBS-based CAD modeling software. Analysis-suitable geometries can be generated from a set of input parameters within the CAD software in a matter of seconds, underscoring the seamless approach enabled by IGA. Upon inspecting the behavior of vibration and buckling solutions under mesh refinement, it is clear that the IGA-based framework is capable of providing accurate solutions using significantly fewer elements relative to a traditional FEA-based approach.

The NREL/SNL 5 MW blade is also optimized within OpenMDAO [137] using both the FAST aeroelastic tool and IGA-based buckling analysis. The objective function is defined such that minimization of both blade mass and total material cost are beneficial. Variation of material thickness profiles is enabled through 11 design variables, and constraints are defined based on the IEC 61400 [18] requirements for material ultimate stress, maximum tip deflection, and resistance to buckling. Optimization is performed on 15 initial blade designs that are generated using Latin hypercube sampling, and multiple designs of potential interest are identified. The optimized design with the absolute lowest objective function evaluation, termed the “balanced” design, features both reduced blade mass and reduced material cost relative to the reference design. Alternatively, the “lightweight” design features a significant blade mass reduction of over 7% achieved through the employment of more carbon fiber in the spar cap. In both of these optimizations, the blade’s buck-

ling behavior governs the design, highlighting one advantage of incorporating high-fidelity structural analysis into blade optimization procedures.

These studies have also generated some peripheral insights. For example, as initial experience suggested and as Figure 5.13 confirms, the objective function space defined in this problem is non-smooth, with numerous local minima. Because both blade mass and material cost have simple, smooth relationships to the design variables, the overall pseudo-objective function complexity appears to stem from the constraints; in particular, the design-governing buckling behavior. It would be reasonable to expect this behavior for other similarly-defined optimization problems. Of course, the optimization procedure presented here is limited in scope. If buckling analysis were performed for loads applied over a wider range of load angles, for example, it is quite possible that the results would differ. Still, this work provides a framework to perform such explorations more easily, more efficiently, and more accurately. It also provides some optimized design alternatives which perform better than the reference design, potentially providing a starting point for future design studies.

5.6 Acknowledgements

Chapter 5 is, in part, a reprint of material as it appears in: “A framework for isogeometric-analysis-based design and optimization of wind turbine blade structures,” (with J. Kiendl and M.-C. Hsu), *Wind Energy*, in review. Chapter 5 is also, in part, a reprint of material as it appears in: “A patch coupling technique for non-matching isogeometric shells with application to composite wind turbine blades,” (with E.L. Johnson, D. Proserpio, J. Kiendl, and M.-C. Hsu), *Computational Mechanics*, in preparation. The dissertation author was the primary investigator of both of these papers.

CHAPTER 6. CONCLUSION AND OUTLOOK

A number of overarching conclusions can be drawn from this work. The first, and perhaps the most significant, conclusion is that, once constructed, an IGA-based framework for wind turbine blade design and optimization can provide substantial savings in regards to analysis time and labor compared to traditional finite-element-based approaches. The efficiency and accuracy of IGA—owed, in part, to the geometric exactness afforded by the IGA approach—corresponds to more data-efficient analysis models which entail significantly less data overhead throughout the design and analysis procedures. Further, the use of the Kirchhoff–Love thin-shell model, which does not require rotational DOF and can be readily used in the context of IGA, is computationally efficient and may be sufficiently accurate, especially with respect to linear buckling, for design purposes. The impact of this efficiency is greatly amplified in highly iterative or optimization-based design procedures which require large numbers of analyses to be performed. Similarly, the advantages of a simplified, single-geometry workflow are especially relevant in the context of optimization, which often requires the entire design-and-analysis workflow to be fully automated.

The value of combining novel, practical, high-fidelity methods with efficient, well-developed reduced-order models is also clearly demonstrated in the optimization of the 5 MW blade. The selected analysis methodologies allow thorough and detailed blade design feedback without requiring excessive computational resources, especially during optimization. In academic settings, it can be tempting to continually pursue more accurate, more sophisticated methodologies regardless of computational cost, practicality, or the broader context of the problem. While such approaches can have merit with respect to addressing and con-

sidering fundamental science problems and long-term goals, their benefits in the context of engineering tend to be limited. Conversely, in the industrial context, one may be tempted to use only tried-and-true methodologies without seriously considering the potential benefits of more cutting-edge approaches. Of course, this tendency is often driven by risk mitigation, which must be seriously considered. Still, the present work seeks to demonstrate the balanced use of both novel and standard analysis approaches.

More concretely, this work provides some quantitative perspective on the tradeoff between blade mass and blade material cost in the context of a carbon-based 5 MW blade design. The existence of a tradeoff between blade mass and blade cost, especially through the use of carbon fiber, can be easily grasped on a conceptual level. Providing actual mass and cost values, however, especially through optimization, is less straightforward. It is not uncommon for blade designers to perform such analyses using less developed, low-fidelity models. Still, insight gleaned from the concrete masses and costs provided in Chapter 5, which correspond to a well-developed, buckling-constrained 5 MW blade design, may be beneficial.

Despite the advantages posed by an IGA-based framework for blade design, the general immaturity of IGA, especially in industrial-grade software, prevents it from being readily adopted in general. In the design context, one must employ methods and tools that are, as much as possible, thoroughly tested and well-documented. Further, if a designer wishes to use the analysis code to justify an engineering design to a certifying body, the analysis code itself must be certified. This is a long and tedious process. One could feasibly use an IGA-based approach for the majority of design activities and could then use a more traditional approach during design certification. This extra complexity, however, may not ultimately be worth the effort.

Although this work addresses many of the fundamental issues associated with the use of an isogeometric Kirchhoff–Love approach in blade design, other IGA-based approaches could also be explored. For example, the impact of adopting a thick-shell, Reissner–Mindlin

formulation [49] for portions or the entirety of the blade structure could be considered. Such an approach might provide improvements in accuracy at the expense of computational efficiency due to the introduction of rotational degrees of freedom. Additionally, because modern wind turbine blades typically have regions which are relatively solid consisting of, for example, glue, the use of solid IGA formulations could be explored. Volumetric NURBS or T-Splines could be employed to model such regions, and coupling of individual entities would again have to be considered. In all of these cases, the use of IGA could lead to improvements with respect to both efficiency and accuracy. Thus, these directions represent potentially fruitful avenues of future research.

BIBLIOGRAPHY

- [1] Lazard. Lazard's levelized cost of energy analysis - version 11.0. <https://www.lazard.com/media/450337/lazard-levelized-cost-of-energy-version-110.pdf>, 2017.
- [2] R. Wiser, M. Bolinger, G. Barbose, N. Darghouth, B. Hoen, A. Mills, J. Rand, D. Millstein, K. Porter, R. Widiss, N. Disanti, F. Oteri, and S. Tegen. 2016 wind technologies market report. Technical Report DOE/GO-102917-5033, U.S. Department of Energy, Office of Energy Efficiency & Renewable Energy, Oak Ridge, TN, 2017.
- [3] GE Renewable Energy. GE announces Haliade-X, the world's most powerful offshore wind turbine. Press Release, March 2018. <https://www.genewsroom.com>.
- [4] J.F. Manwell, J.G. McGowan, and A.L. Rogers. *Wind Energy Explained: Theory, Design and Application, 2nd ed.* John Wiley & Sons, Chichester, 2009.
- [5] C. Money, M. Hand, M. Bolinger, J. Rand, D. Heimiller, and J. Ho. 2015 cost of wind energy review. Technical Report NREL/TP-6A20-66861, National Renewable Energy Laboratory, Golden, CO, 2017.
- [6] K. Cox and A. Echtermeyer. Effects of composite fiber orientation on wind turbine blade buckling resistance. *Wind Energy*, 17(12):1925–1943, 2014.
- [7] K. Cox and A. Echtermeyer. Structural design and analysis of a 10MW wind turbine blade. *Energy Procedia*, 24:194–201, 2012.

- [8] X. Chen, X. Zhao, and J. Xu. Revisiting the structural collapse of a 52.3 m composite wind turbine blade in a full-scale bending test. *Wind Energy*, 17:657–669, 2017.
- [9] E. Lindgaard and E. Lund. Nonlinear buckling optimization of composite structures. *Computer Methods in Applied Mechanics and Engineering*, 199(37–40):2319–2330, 2010.
- [10] F.M. Jensen, P.M. Weaver, L.S. Cecchini, H. Stant, and R.F. Nielsen. The Brazier effect in wind turbine blades and its influence on design. *Wind Energy*, 15(2):319–333, 2012.
- [11] A. Chehouri, R. Younes, A. Ilinca, and J. Perron. Review of performance optimization techniques applied to wind turbine blades. *Applied Energy*, 142:361–388, 2015.
- [12] W. Xudong, W.J. Zhu, J.N. Sørensen, and C. Jin. Shape optimization of wind turbine blades. *Wind Energy*, 12(8):781–803, 2009.
- [13] P. Fuglsang and H.A. Madsen. Optimization method for wind turbine rotors. *Journal of Wind Engineering and Industrial Aerodynamics*, 80(1–2):191–206, 1999.
- [14] R. Lanzafame and M. Messina. Fluid dynamics wind turbine design: Critical analysis, optimization and application of BEM theory. *Renewable Energy*, 32(14):2291–2305, 2007.
- [15] E. Benini and A. Toffolo. Optimal design of horizontal-axis wind turbines using blade-element theory and evolutionary computation. *Journal of Solar Energy Engineering*, 124:357–363, 2002.
- [16] M.S. Campobasso, E. Minisci, and M. Caboni. Aerodynamic design optimization of wind turbine rotors under geometric uncertainty. *Wind Energy*, 19(1):51–65, 2014.

- [17] P. Fuglsang, C. Bak, J.G. Schepers, B. Bulder, T.T. Cockerill, P. Claiden, A. Olesen, and R. van Rossen. Site-specific design optimization of wind turbines. *Wind Energy*, 5:261–279, 2002.
- [18] Wind turbines–Part 1: Design requirements. Technical Report IEC 61400-1, International Electrotechnical Commission, Geneva, Switzerland, 2005.
- [19] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.
- [20] D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. A large deformation, rotation-free, isogeometric shell. *Computer Methods in Applied Mechanics and Engineering*, 200:1367–1378, 2011.
- [21] 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. *Computer Methods in Applied Mechanics and Engineering*, 284:508–520, 2015.
- [22] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *International Journal for Numerical Methods in Fluids*, 65:207–235, 2011.
- [23] Y. Bazilevs, M.-C. Hsu, J. Kiendl, R. Wüchner, and K.-U. Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.
- [24] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49:93–100, 2011.

- [25] Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252:28–41, 2012.
- [26] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.
- [27] A. Korobenko, M.-C. Hsu, I. Akkerman, J. Tippmann, and Y. Bazilevs. Structural mechanics modeling and FSI simulation of wind turbines. *Mathematical Models and Methods in Applied Sciences*, 23(2):249–272, 2013.
- [28] Y. Bazilevs, A. Korobenko, X. Deng, and J. Yan. Novel structural modeling and mesh moving techniques for advanced fluid-structure interaction simulation of wind turbines. *International Journal for Numerical Methods in Engineering*, 102(3-4):766–783, 2014.
- [29] Y. Bazilevs, X. Deng, A. Korobenko, F. Lanza di Scalea, M.D. Todd, and S.G. Taylor. Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data. *Journal of Applied Mechanics*, 82(9), 2015.
- [30] M.-C. Hsu, C. Wang, A.J. Herrema, D. Schillinger, A. Ghoshal, and Y. Bazilevs. An interactive geometry modeling and parametric design platform for isogeometric analysis. *Computers and Mathematics with Applications*, 70:1481–1500, 2015.
- [31] J. Yan, A. Korobenko, X. Deng, and Y. Bazilevs. Computational free-surface fluid–structure interaction with application to floating offshore wind turbines. *Computers & Fluids*, 141:155–174, 2016.
- [32] A.J. Herrema, N.M. Wiese, C.N. Darling, B. Ganapathysubramanian, A. Krishnamurthy, and M.-C. Hsu. A framework for parametric design optimization using iso-

- geometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 316: 944–965, 2016.
- [33] E. Ferede, M.M. Abdalla, and G.J.W. van Bussel. Isogeometric based framework for aeroelastic wind turbine blade analysis. *Wind Energy*, 20(2):193–210, 2017.
- [34] R. Fletcher. *Practical Methods of Optimization*, 2nd ed. John Wiley & Sons, Chichester, 1987.
- [35] S.S. Rao. *Engineering Optimization: Theory and Practice*. John Wiley & Sons, Hoboken, New Jersey, 2009.
- [36] J.R.R.A. Martins and A.B. Lambe. Multidisciplinary design optimization: a survey of architectures. *AIAA journal*, 51(9):2049–2075, 2013.
- [37] H.A. Eschenauer and N. Olhoff. Topology optimization of continuum structures: A review. *Applied Mechanics Reviews*, 54(4):331–390, 2001.
- [38] J.N. Sørensen. Blade-element/momentum theory. In *General Momentum Theory for Horizontal Axis Wind Turbines*, Research Topics in Wind Energy, chapter 7. Springer, Cham, 2016.
- [39] Q. Wang, M.A. Sprague, J. Jonkman, N. Johnson, and B. Jonkman. BeamDyn: a high-fidelity wind turbine blade solver in the FAST modular framework. *Wind Energy*, 20(8):1439–1462, 2017.
- [40] T. Kim, A.M. Hansen, and K. Branner. Development of an anisotropic beam finite element for composite wind turbine blades in multibody system. *Renewable Energy*, 59:172–183, 2013.
- [41] D. Vučina, I. Maranić-Kragić, and Z. Milas. Numerical models for robust shape optimization of wind turbine blades. *Renewable Energy*, 87:849–862, 2016.

- [42] M.A. Elfarra, N. Sezer-Uzol, and I.S. Akmandor. NREL VI rotor blade: numerical investigation and winglet design and optimization using CFD. *Wind Energy*, 17(4): 605–626, 2013.
- [43] M.E. Bechly and P.D. Clausen. Structural design of a composite wind turbine blade using finite element analysis. *Computers & Structures*, 63(3):639–646, 1997.
- [44] N. Buckney, S. Green, A. Pirrera, and P.M. Weaver. On the structural topology of wind turbine blades. *Wind Energy*, 16(4):545–560, 2012.
- [45] G.K.W. Kenway, G.J. Kennedy, and J.R.R.A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, USA, 2010.
- [46] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.
- [47] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*, 2nd ed. Springer-Verlag, New York, 1997.
- [48] M. Bischoff, W.A. Wall, K.-U. Bletzinger, and E. Ramm. Models and finite elements for thin-walled structures. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Solids and Structures, chapter 3. John Wiley & Sons, 2004.
- [49] D.J. Benson, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. *Computer Methods in Applied Mechanics and Engineering*, 199:276–289, 2010.
- [50] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

- [51] A. Korobenko, M.-C. Hsu, I. Akkerman, J. Tippmann, and Y. Bazilevs. Structural mechanics modeling and FSI simulation of wind turbines. *Mathematical Models and Methods in Applied Sciences*, 23(2):249–272, 2013.
- [52] J.N. Reddy. *Mechanics of Laminated Composite Plates and Shells: Theory and Analysis*, 2nd ed. CRC Press, Boca Raton, FL, 2004.
- [53] D. Schillinger, L. Dedè, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and T.J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249–252:116–150, 2012.
- [54] X. Wei, Y. Zhang, L. Liu, and T.J.R. Hughes. Truncated T-splines: Fundamentals and methods. *Computer Methods in Applied Mechanics and Engineering*, 2016. <http://dx.doi.org/10.1016/j.cma.2016.07.020>.
- [55] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, and A. Düster. Geometric modeling, isogeometric analysis and the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 249–252:104–115, 2012.
- [56] 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. *Archives of Computational Methods in Engineering*, 22(3):391–455, 2015.
- [57] 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. *Computer Methods in Applied Mechanics and Engineering*, 284:401–457, 2015.
- [58] Rhino. <http://www.rhino3d.com/>. Accessed 27 May 2016.

- [59] Siemens NX. https://www.plm.automation.siemens.com/en_us/products/nx/. Accessed 27 May 2016.
- [60] Grasshopper. <http://www.grasshopper3d.com/>. Accessed 27 May 2016.
- [61] W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976–2988, 2008.
- [62] D. Fußeder, B. Simeon, and A.-V. Vuong. Fundamental aspects of shape optimization in the context of isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 268:313–331, 2015.
- [63] A.N. Moysidis and V.K. Koumousis. A hysteric formulation for isogeometric analysis and shape optimization of plane stress structures. In *8th GRACM International Congress on Computational Mechanics*, Volos, Greece, 2015.
- [64] S. Julisson, C. Fourcade, P. de Nazelle, and L. Dumas. A novative optimal shape design based on an isogeometric approach: Application to optimization of surface shapes with discontinuous curvature. In *11th World congress on structural and multidisciplinary optimization (WCSMO-11)*, Sydney, Australia, 2015.
- [65] S. Cho and S.-H. Ha. Isogeometric shape design optimization: exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38:53–70, 2009.
- [66] X. Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 199:2059–2071, 2010.
- [67] J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. *Computer Methods in Applied Mechanics and Engineering*, 274:148–167, 2014.

- [68] D.M. Nguyen, A. Evgrafov, and J. Gravesen. Isogeometric shape optimization for electromagnetic scattering problems. *Progress in Electromagnetics Research B*, 45: 117–146, 2012.
- [69] N.D. Manha, A. Evgrafov, A.R. Gersborg, and J. Gravesen. Isogeometric shape optimization of vibrating membranes. *Computer Methods in Applied Mechanics and Engineering*, 200:1343–1353, 2011.
- [70] M. Yoon, M.-J. Choi, and S. Cho. Isogeometric configuration design optimization of heat conduction problems using boundary integral equation. *International Journal of Heat and Mass Transfer*, 89:937–949, 2015.
- [71] P. Nørtoft and J. Gravesen. Isogeometric shape optimization in fluid mechanics. *Structural and Multidisciplinary Optimization*, 48(5):909–925, 2013.
- [72] S.-W. Lee, J. Lee, and S. Cho. Isogeometric shape optimization of ferromagnetic materials in magnetic actuators. *IEEE Transactions on Magnetics*, 52(2):1–8, 2016.
- [73] K.V. Kostas, A.I. Ginnis, C.G. Politis, and P.D. Kaklis. Ship-hull shape optimization with a T-spline based BEM-isogeometric solver. *Computer Methods in Applied Mechanics and Engineering*, 284:611–622, 2015.
- [74] SolidWorks. <http://www.solidworks.com/>. Accessed 27 May 2016.
- [75] Pro/ENGINEER. <http://www.ptc.com/cad/pro-engineer>. Accessed 13 Oct 2016.
- [76] J.J. Shah and M. Mäntylä. *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. John Wiley & Sons, 1995.
- [77] O.W. Salomons, F.J.A.M. van Houten, and H.J.J. Kals. Review of research in feature-based design. *Journal of manufacturing systems*, 12(2):113–132, 1993.
- [78] L.K. Kyprianou. *Shape classification in computer-aided design*. PhD thesis, University of Cambridge, 1980.

- [79] A. Verroust, F. Schonek, and D. Roller. Rule-oriented method for parameterized computer-aided design. *Computer-Aided Design*, 24(10):531–540, 1992.
- [80] I.E. Sutherland. Sketch pad: A man-machine graphical communication system. In *Proceedings of the SHARE design automation workshop*, pages 6–329. ACM, 1964.
- [81] H. Suzuki, H. Ando, and F. Kimura. Geometric constraints and reasoning for geometrical CAD systems. *Computers & Graphics*, 14(2):211–224, 1990.
- [82] A. Borning. ThingLab: An object-oriented system for building simulations using constraints. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pages 497–498. Morgan Kaufmann Publishers Inc., 1977.
- [83] A.A.G. Requicha and H.B. Voelcker. Constructive solid geometry. Technical report, University of Rochester, 1977.
- [84] A. Krishnamurthy. *Parallel GPU Algorithms for Mechanical CAD*. PhD thesis, University of California, Berkeley, 2010. URL <http://eprints.cdlib.org/uc/item/59n1g12w>.
- [85] S. Kodiyalam, R.J. Yang, L. Gu, and C.H. Tho. Multidisciplinary design optimization of a vehicle system in a scalable, high performance computing environment. *Structural and Multidisciplinary Optimization*, 26(3-4):256–263, 2004.
- [86] C.C. Long, A.L. Marsden, and Y. Bazilevs. Shape optimization of pulsatile ventricular assist devices using FSI to minimize thrombotic risk. *Computational Mechanics*, 54(4):921–932, 2014.
- [87] G.K.W. Kenway and J.R.R.A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51:144–160, 2014.

- [88] T. Ashuri, M.B. Zaaijer, J.R.R.A. Martins, G.J.W. van Bussel, and G.A.M. van Kuik. Multidisciplinary design optimization of offshore wind turbines for minimum levelized cost of energy. *Renewable Energy*, 68:893–905, 2014.
- [89] A.L. Marsden. Optimization in cardiovascular modeling. *Annual Review of Fluid Mechanics*, 46:519–546, 2014.
- [90] ANSYS Workbench Platform. <http://www.ansys.com/Products/Platform>. Accessed 27 May 2016.
- [91] Mathworks. Optimization toolbox user’s guide, 2016.
- [92] Dakota. <https://dakota.sandia.gov/>. Accessed 27 May 2016.
- [93] L. Solano and P. Brunet. Constructive constraint-based model for parametric cad systems. *Computer-Aided Design*, 26(8):614–621, 1994.
- [94] X. Chen and C.M. Hoffmann. On editability of feature-based design. *Computer-aided design*, 27(12):905–914, 1995.
- [95] C.M. Hoffmann. Constraint-based computer-aided design. *Journal of Computing and Information Science in Engineering*, 5(3):182–187, 2005.
- [96] J.R. Rossignac, P. Borrel, and L.R. Nackman. Interactive design with sequences of parameterized transformations. In V. Akman, P. J. W. ten Hagen, and P. J. Veerkamp, editors, *Intelligent CAD Systems II: Implementational Issues*, pages 93–125. Springer-Verlag, 1989.
- [97] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199: 2403–2416, 2010.

- [98] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. John Wiley & Sons, Chichester, 2013.
- [99] D.J. Benson, S. Hartmann, Y. Bazilevs, M.-C. Hsu, and T.J.R. Hughes. Blended isogeometric shells. *Computer Methods in Applied Mechanics and Engineering*, 255:133–146, 2013.
- [100] R.N. Simpson, S.P.A. Bordas, J. Trevelyan, and T. Rabczuk. A two-dimensional Isogeometric Boundary Element Method for elastostatic analysis. *Computer Methods in Applied Mechanics and Engineering*, 209–212:87–100, 2012.
- [101] A.I. Ginnis, K.V. Kostas, C.G. Politis, P.D. Kaklis, K.A. Belibassakis, Th.P. Gerostathis, M.A. Scott, and T.J.R. Hughes. Isogeometric boundary-element analysis for the wave-resistance problem using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 279:425–439, 2014.
- [102] M.-C. Hsu, D. Kamensky, F. Xu, J. Kiendl, C. Wang, M.C.H. Wu, J. Mineroff, A. Reali, Y. Bazilevs, and M.S. Sacks. Dynamic and fluid–structure interaction simulations of bioprosthetic heart valves using parametric design with T-splines and Fung-type material models. *Computational Mechanics*, 55:1211–1225, 2015.
- [103] M.-C. Hsu, C. Wang, F. Xu, A.J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.
- [104] M. Levoy. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245–261, 1990.
- [105] W. Martin and E. Cohen. Representation and extraction of volumetric attributes using trivariate splines: A mathematical framework. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, pages 234–240, 2001.

- [106] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.
- [107] P. Cignoni, L. De Floriani, C. Montani, E. Puppo, and R. Scopigno. Multiresolution modeling and visualization of volume data based on simplicial complexes. In *Proceedings of the 1994 Symposium on Volume Visualization*, pages 19–26, 1994.
- [108] A. Knoll, I. Wald, S. Parker, and C. Hansen. Interactive isosurface ray tracing of large octree volumes. In *2006 IEEE Symposium on Interactive Ray Tracing*, pages 115–124, 2006.
- [109] B. Nelson and R.M. Kirby. Ray-tracing polymorphic multidomain spectral/hp elements for isosurface rendering. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):114–125, 2006.
- [110] T. Martin, E. Cohen, and R. M. Kirby. Direct isosurface visualization of hex-based high-order geometry and attribute representations. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):753–766, 2012.
- [111] RhinoCommon. <http://developer.rhino3d.com/guides/#rhinocommon>. Accessed 27 May 2016.
- [112] J. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Lehrstuhl für Statik, Technische Universität München, 2011.
- [113] L. Leifsson and S. Koziel. Variable-fidelity aerodynamic shape optimization. In S. Koziel and X.-S. Yang, editors, *Computational Optimization, Methods and Algorithms*, chapter 9, pages 179–210. Springer-Verlag Berlin Heidelberg, 2011.
- [114] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7:1–25, 1997.

- [115] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, Golden, CO, 2009.
- [116] IRENA. Renewable energy technologies: Cost analysis series, wind power. Technical report, International Renewable Energy Agency, 2012.
- [117] J.M. Jonkman and M.L. Buhl Jr. FAST user’s guide. Technical Report NREL/EL-500-38230, National Renewable Energy Laboratory, Golden, CO, 2005.
- [118] B.R. Resor. Definition of a 5MW/61.5m wind turbine blade reference model. Technical Report SAND2013-2569, Sandia National Laboratories, Albuquerque, NM, 2013.
- [119] B. Snyder and M.J. Kaiser. Ecological and economic cost-benefit analysis of offshore wind energy. *Renewable Energy*, 34:1567–1578, 2009.
- [120] C. Moné, T. Stehly, B. Maples, and E. Settle. 2014 cost of wind energy review. Technical Report NREL/TP-6A20-64281, National Renewable Energy Laboratory, Golden, CO, 2015.
- [121] J. Benzaken, A.J. Herrema, M.-C. Hsu, and J.A. Evans. A rapid and efficient isogeometric design space exploration framework with application to structural mechanics. *Computer Methods in Applied Mechanics and Engineering*, 316:1215–1256, 2017.
- [122] Y. Guo and M. Ruess. Nitsche’s method for a coupling of isogeometric thin shells and blended shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:881–905, 2015.
- [123] 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. *Computer Methods in Applied Mechanics and Engineering*, 316:1157–1178, 2017.

- [124] 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. *Computer Methods in Applied Mechanics and Engineering*, 336:39–79, 2018.
- [125] L. Coox, F. Greco, O. Atak, D. Vandepitte, and W. Desmet. A robust patch coupling method for NURBS-based isogeometric analysis of non-conforming multipatch surfaces. *Computer Methods in Applied Mechanics and Engineering*, 316:235–260, 2017.
- [126] 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. *Computer Methods in Applied Mechanics and Engineering*, 316:43–83, 2017.
- [127] V. Hernandez, J.E. Roman, and V. Vidal. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, 31(3):351–362, 2005.
- [128] J.E. Roman, C. Campos, E. Romero, and A. Tomas. SLEPc users manual. Technical Report DSIC-II/24/02 - Revision 3.7, D. Sistemes Informàtics i Computació, Universitat Politècnica de València, Valencia, Spain, 2016.
- [129] T. Belytschko, H. Stolarski, W.K. Liu, N. Carpenter, and J.S.-J. Ong. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 51:221–258, 1985.
- [130] K.Y. Sze, X.H. Liu, and S.H. Lo. Popular benchmark problems for geometric nonlinear analysis of shells. *Finite Elements in Analysis and Design*, 40:1551–1569, 2004.
- [131] Michael Smith. *ABAQUS/Standard User’s Manual, Version 6.9*. Simulia, 2009.
- [132] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, K. Rupp, B.F.

- Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, Lemont, IL, 2016.
- [133] S. Balay, W.D. Gropp, L.C. McInnes, and B.F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E Arge, AM Bruaset, and HP Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
- [134] J. Jonkman. FAST. <https://nwtc.nrel.gov/FAST>, 2015.
- [135] T.J. Larsen. How 2 HAWC2, the user’s manual. http://orbit.dtu.dk/files/7703110/ris_r_1597.pdf, 2007.
- [136] P.-E. Réthoré and F. Zahle. Framework for unified systems engineering and design of wind plants (FUSED-Wind). <http://www.fusedwind.org/>, 2017.
- [137] J. Gray, K.T. Moore, and B.A. Naylor. OpenMDAO: An open source framework for multidisciplinary analysis and optimization. 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, 2010. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2010-9101>.
- [138] T. Griffith and T.D. Ashwill. The Sandia 100-meter all-glass baseline wind turbine blade: SNL100-00. Technical Report SAND2011-3779, Sandia National Laboratories, Albuquerque, New Mexico, 2011.
- [139] X.S. Li and J.W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Trans. Mathematical Software*, 29(2):110–140, 2003.
- [140] G. Bir and R. Damiani. PreComp. <https://nwtc.nrel.gov/PreComp>, 2014.

- [141] T. Griffith and W. Johanns. Large blade manufacturing cost studies using the Sandia blade manufacturing cost tool and Sandia 100-meter blades. Technical Report SAND2013-2734, Sandia National Laboratories, Albuquerque, NM, 2013.
- [142] R.E. Perez, P.W. Jansen, and J.R.R.A. Martins. pyOpt: A Python-based object-oriented framework for nonlinear constrained optimization. *Structures and Multidisciplinary Optimization*, 45(1):101–118, 2012.
- [143] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42:55–61, 2000.

APPENDIX. NREL/SNL 5 MW BLADE DEFINITION TABLE

Table A.1: Definition of cross section parameters used to construct the NREL/SNL 5 MW blade model. “DP” stands for “division point.”

Blade span (m)	Airfoil	TE Type	Twist (deg)	Chord (m)	Pitch Axis	DP1	DP2	DP3	DP4	DP5
0.00	circle	round	13.31	3.386	0.5	0.0295	0.4114	0.5886	0.8523	0.9705
0.30 †	circle	round	13.31	3.386	0.5	0.0294	0.4114	0.5886	0.8532	0.9706
0.40 †	interp	round	13.31	3.386	0.5	0.0293	0.4114	0.5886	0.8535	0.9707
0.50 †	interp	round	13.31	3.386	0.5	0.0293	0.4114	0.5886	0.8547	0.9707
0.60 †	interp	round	13.31	3.386	0.5	0.0292	0.4114	0.5886	0.8540	0.9708
0.70 †	interp	round	13.31	3.386	0.5	0.0291	0.4114	0.5886	0.8543	0.9709
0.80 †	interp	round	13.31	3.386	0.5	0.0291	0.4114	0.5886	0.8546	0.9709
1.37	circle	round	13.31	3.386	0.5	0.0288	0.4114	0.5886	0.8562	0.9712
1.50 †	interp	round	13.31	3.387	0.4985	0.0287	0.4102	0.5868	0.8565	0.9713
1.60 †	interp	round	13.31	3.388	0.4974	0.0286	0.4094	0.5854	0.8568	0.9714
4.10	interp*	round	13.31	3.629	0.4692	0.0272	0.3876	0.5508	0.8638	0.9728
5.50	interp	round	13.31	3.873	0.4535	0.0265	0.3755	0.5315	0.8677	0.9735
6.83	interp*	flat	13.31	4.124	0.4385	0.0257	0.3639	0.5131	0.8715	0.9743
9.00	interp	flat	13.31	4.461	0.4141	0.0245	0.3450	0.4831	0.8775	0.9755
10.25	DU99-W-405	flat	13.31	4.557	0.4	0.0238	0.3342	0.4658	0.8810	0.9762
12.00	interp	flat	12.53	4.615	0.4	0.0228	0.3313	0.4687	0.8859	0.9772
14.35	DU99-W-350	flat	11.48	4.652	0.4	0.0215	0.3274	0.4726	0.8925	0.9785
17.00	interp	flat	10.68	4.584	0.4	0.0231	0.3230	0.4770	0.8871	0.9769
18.45	interp*	flat	10.16	4.506	0.4	0.0240	0.3206	0.4794	0.8841	0.9760
20.50	interp	flat	9.63	4.374	0.4	0.0253	0.3172	0.4828	0.8800	0.9747
22.55	DU97-W-300	flat	9.01	4.249	0.4	0.0265	0.3138	0.4862	0.8758	0.9735
24.60	interp	flat	8.40	4.132	0.4	0.0278	0.3104	0.4896	0.8716	0.9722
26.65	DU91-W-250	flat	7.79	4.007	0.4	0.0291	0.3070	0.4930	0.8674	0.9709
30.75	DU91-W-250	flat	6.54	3.748	0.4	0.0316	0.3003	0.4997	0.8590	0.9684
32.00	interp	flat	6.18	3.672	0.4	0.0323	0.2982	0.5018	0.8565	0.9677

34.85	DU93-W-210	flat	5.36	3.502	0.4	0.0341	0.2935	0.5065	0.8506	0.9659
37.00	interp	flat	4.75	3.373	0.4	0.0354	0.2899	0.5101	0.8462	0.9646
38.95	DU93-W-210	flat	4.19	3.256	0.4	0.0366	0.2867	0.5133	0.8423	0.9634
41.00	interp	sharp	3.66	3.133	0.4	0.0379	0.2833	0.5167	0.8381	0.9621
42.00 †	interp	sharp	3.40	3.073	0.4	0.0385	0.2817	0.5183	0.8360	0.9615
43.04 ‡	NACA-64-618	sharp	3.13	3.010	0.4	0.0391	0.2799	0.5201	0.8339	0.9609
45.00	interp	sharp	2.74	2.893	0.4	0.0403	0.2767	0.5233	0.8339	0.9597
47.15	NACA-64-618	sharp	2.32	2.764	0.4	0.0416	0.2731	0.5269	0.8339	0.9584
51.25	NACA-64-618	sharp	1.53	2.518	0.4	0.0442	0.2664	0.5336	0.8339	0.9558
54.67	NACA-64-618	sharp	0.86	2.313	0.4	0.0463	0.2607	0.5393	0.8339	0.9537
57.40	NACA-64-618	sharp	0.37	2.086	0.4	0.0479	0.2562	0.5438	0.8339	0.9521
60.13	NACA-64-618	sharp	0.11	1.419	0.4	0.0705	0.1886	0.6114	0.8339	0.9226
61.50	NACA-64-618	sharp	0.00	1.086	0.4	0.0921	0.1236	0.6764	0.8339	0.9079

Rows highlighted in grey indicate cross sections used for aerodynamic analysis. Division points at which a shear web is defined indicated by **boldface**.

* Indicates that an airfoil definition given by Jonkman et al. [115] was replaced by an interpolated profile for the purposes of smooth geometry [118].

† Indicates stations that are ignored during IGA model construction in order to achieve more uniform spanwise NURBS discretization.

‡ Indicates the spanwise location at which the trailing edge reinforcement terminates.