TECHNISCHE UNIVERSITÄT WIEN

DISSERTATION

Shape Optimization of CAD-Compliant Boundary-Conforming Microstructured Geometries

carried out for the purpose of obtaining the degree of Doctor technicae (Dr. techn.), submitted at TU Wien, Faculty of Mechanical and Industrial Engineering, by

Jacques Marvin Zwar
Matrikelnummer: 12020679

under the supervision of

Univ.Prof.in Dr.-Ing.in Stefanie Elgeti
Institute of Lightweight Design and Structural Biomechanics, E317

reviewed by

Assoc. Prof. Dr. rer. nat. Matthias Möller
Delft Institute of Applied Mathematics
TU Delft, The Netherlands

Prof. Dr. rer. nat. Sebastian Schöps
Chair of Computational Electromagnetics
TU Darmstadt, Germany

Preface

This project has received funding from the European Union's Horizon 2020 Research and innovation program under grant agreement No. 862025 (ADAM2).

Affidavit

I declare in lieu of oath, that I wrote this thesis and carried out the associated research myself, using only the literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted for examination elsewhere, nor is it currently under consideration for a thesis elsewhere.

I acknowledge that the submitted work will be checked electronically-technically using suitable and state-of-the-art means (plagiarism detection software). On the one hand, this ensures that the submitted work was prepared according to the high-quality standards within the applicable rules to ensure good scientific practice "Code of Conduct" at the TU Wien. On the other hand, a comparison with other student theses avoids violations of my personal copyright.

Vienna, September 2024

Jacques Marvin Zwar

Summary

Recent developments in additive manufacturing have opened up a vast field of new design possibilities that cannot be fully exploited by traditional engineering methods. These advances require sophisticated numerical design methods, particularly in the form of shape optimization. This thesis addresses this need by presenting a comprehensive framework for the design, analysis, and optimization of periodic two-scale or microstructured designs compatible and compliant with geometry representations from computer-aided design. It is based on a paradigm for constructing boundary-conforming microstructures through functional compositions between splines that naturally conform to the external geometry.

The design variables that control the local geometry prior to its composition into the microstructure are abstracted in order to ensure smooth transitions and to reduce local fluctuations between adjacent tiles. The geometry is modeled using a volumetric representation of the material, which is particularly well-suited for IsoGeometric Analysis. This approach maintains geometric exactness, eliminates the need for computationally expensive meshing techniques, and streamlines analysis, thereby enhancing overall computational efficiency and simulation accuracy. Furthermore, by deriving geometric derivatives of the modeled microstructure, we enable the computation of sensitivities using an adjoint approach. These sensitivities facilitate the use of gradient-based optimization algorithms, reducing the number of iterations required to converge to an optimal design.

The proposed framework is validated through applications in heat transfer and structural optimization. In heat transfer problems, it is employed to optimize temperature profiles with respect to a target temperature. Here, an extrusion die is presented as a potential future application for microstructured geometries. For structural design, the framework minimizes compliance under nonlinear mass constraints applied to a cantilever benchmark. These test cases also serve to compare different optimization algorithms with increasing design complexity. Furthermore, local, macroscopic, and concurrent optimization strategies are evaluated.

Kurzfassung

Die jüngsten Entwicklungen im Bereich der additiven Fertigung haben eine Vielzahl neuer Gestaltungs- und Designmöglichkeiten eröffnet, die mit traditionellen Ingenieurmethoden nicht vollständig ausgeschöpft werden können. Diese Fortschritte erfordern die Entwicklung neuer, numerische Entwurfsmethoden, insbesondere durch Formoptimierung. Die vorliegende Arbeit widmet sich diesem Bedarf durch die Entwicklung eines umfassenden Frameworks für den Entwurf, die Analyse und die Optimierung periodischer Zweiskalen- oder mikrostrukturierter Designs, die mit der Geometrieredarstellung aus dem rechnergestützten Entwurf (CAD) konform und kompatibel sind. Es basiert auf einem Paradigma zur Konstruktion grenzkonformer Mikrostrukturen durch funktionale Verkettung zwischen Splines, die sich entlang der äußeren Geometrie ausrichten.

Die Designvariablen, die die lokale Geometrie vor ihrer Verkettung und Eingliederung in die Mikrostruktur definieren, werden abstrahiert, um glatte Übergänge zu gewährleisten und lokale Schwankungen zwischen benachbarten Zellen zu reduzieren. Die Geometrie wird durch eine volumetrische Darstellung des Materials modelliert, welche sich besonders gut für die Isogeometrische Analyse eignet. Durch diesen Ansatz wird die geometrische Genauigkeit beibehalten, die Notwendigkeit rechenintensiver Meshing-Schritte entfällt, und es wird eine Schnittstelle zwischen Geometriemodellierung und Analyse geschaffen, was zur Steigerung der Gesamtrecheneffizienz führt und die Simulationsgenauigkeit verbessert. Darüber hinaus ermöglicht die Herleitung geometrischer Ableitungen der modellierten Mikrostruktur die Berechnung von Sensitivitäten mittels eines adjungierten Ansatzes. Diese Ableitungen ermöglichen die Verwendung gradientenbasierter Optimierungsalgorithmen und verringern die Anzahl der notwendigen Iterationen, um zu einem optimalen Design zu konvergieren.

Das Framework wird durch Anwendung an Wärmeübertragungsproblemen im Rahmen der Strukturoptimierung validiert. Bei der Wärmeübertragung wird es zur Optimierung von Temperaturprofilen unter Berücksichtigung einer Zieltemperatur eingesetzt. Hier wird ein Extrusionswerkzeug als potenzielle Anwendung für mikrostrukturierte Geometrien vorgestellt. Für das Strukturoptimierungsproblem minimiert das Framework die Nachgiebigkeit unter Berücksichtigung nichtlinearer Massenrandbedingungen, angewendet auf einen Tragbalken Benchmarktest. Diese Anwendungsfälle dienen auch dem Vergleich verschiedener Optimierungsalgorithmen mit zunehmender Designkomplexität. Darüber hinaus werden lokale, makroskopische und kombinierte Optimierungsstrategien bewertet.

Acknowledgments

This thesis is the result of my work as a research assistant at the Institute of Lightweight Design and Structural Biomechanics at TU Wien. First and foremost, I would like to express my heartfelt gratitude to my supervisor, *Stefanie Elgeti*. Thank you for your guidance and for the freedom you allowed me throughout my academic journey. I extend my sincere thanks to *Ludovic Chamoin* for welcoming me to ENS Paris-Saclay during my research stay in Paris. I greatly appreciated your expertise, your extraordinary commitment, and your hospitality.

I would also like to thank my colleagues in Vienna and Aachen, including the biomechanics team and non-scientific staff, for the many fruitful discussions and wonderful distractions during coffee breaks. The office sports activities, Friday night skating, badminton, and game nights provided a much-needed balance to the sometimes stressful times. In particular, I would like to thank *Clemens Fricke*, *Daniel Wolff*, and *Jaewook Lee* (in alphabetical order). At times, you were the only reason I left my comfortable home office and went to the institute. Clemens, thank you for the amazing cakes, sweets, countless coffee breaks, and for being a great companion in sports activities and baking. Daniel, my "partner in crime" in the development of CAMPIGA, you were always there to share the struggles and triumphs. Our countless meetings, CCCs, and discussions kept me motivated and on track; without you, I would never have finished this thesis. Jae, BDFL, you have always had an incredible vision and your constant push has resulted in software that I am - and I hope we are both - very proud of. Thank you for always understanding my twisted thinking and keeping the mood light.

Furthermore, I am grateful to the people I met at conferences who became friends along the way, including *Michelangelo*, *Jacob*, and many others. Thank you for the great discussions and memorable moments. On that note, I would also like to thank the G+smo team for their incredible tech support and assistance with my implementations.

Last but not least, I am infinitely thankful to my friends and family — my parents, my sister, and my amazing partner — whose unconditional support and love have been my foundation throughout this stressful journey. The times abroad have been challenging, but I always knew I had you only one call away when things became overwhelming. I would not be here if it weren't for your constant encouragement, and I am forever grateful. Danke für alles.

Vienna, April 2024 *Jacques Marvin Zwar*

Contents

List of Figures

Acronyms

AD Algorithmic Differentiation. 30, 86

AM Additive Manufacturing. 1, 7, 8, 9, 22, 24, 89

BiCGStab BiConjugate Gradient Stabilized method. 42

BOBYQA Bound Optimization BY Quadratic Approximation. 60, 61, 63

B-Rep Boundary Representation. 39

BC Boundary Condition. 33, 34, 36, 37, 40, 42

CAA Computer Aided Analysis. 37, 42

CAD Computer Aided Design. 1, 2, 8, 9, 10, 15, 19, 22, 37, 38, 39, 42, 47, 48, 55, 87, 88

CAE Computer Aided Engineering. XIII, 1, 2, 42

COBYLA Constrained Optimization BY Linear Approximation. 60, 63

DOF Degrees Of Freedom. 2, 38, 50, 52, 58, 59, 60, 68, 79, 83, 88

FD Finite Differences. 30, 51, 86

FDM Finite Difference Method. 37

FEM Finite Element Method. 2, 19, 20, 33, 37, 38, 39, 42, 45, 47, 57

FFD Free-Form Deformation. 15, 19, 20, 47

FVM Finite Volume Method. 37

G+SMO Geometry + Simulation Modules. 42, 49

IGA IsoGeometric Analysis. XIII, XX, 2, 3, 4, 9, 19, 21, 22, 23, 33, 37, 38, 39, 40, 41, 42, 47, 48, 55, 56, 57, 58, 87, 92

ILSB Institute of Lightweight Design and Structural Biomechanics. 12

KKT Karush-Kuhn-Tucker. 46, 65

LBFGS Low-storage Broyden–Fletcher–Goldfarb–Shanno. 61

MMA Method of Moving Asymptotes. 61, 63, 64

NLopt Nonlinear Optimization Framework. 50, 60, 61

NMS Nelder-Mead Simplex. 60, 61, 63

NURBS Non-Uniform Rational B-Splines. 9, 10, 11, 12, 13, 15, 18, 40, 47

PDE Partial Differential Equation. XIX, 2, 4, 33, 34, 35, 36, 37, 39, 45, 46, 49, 50, 51, 86, 87, 92

SIMP Solid Isotropic Material with Penalization. 8, 48

SLMVM Shifted Limited-Memory Variable-Metric. 61

SLSQP Sequential Least-Squares Quadratic Programming. XIII, 50, 61, 62, 63, 64, 65, 69, 79, 83

splinepy Library for prototyping spline geometries of arbitrary dimensions and degrees, and IGA. 12, 49

THB Truncated Hierarchical B-Spline. 10

TO Topology Optimization. 4, 7, 8, 44, 48

TPMS Triply Periodic Minimal Surface. 22, 48

V-Rep Volumetric Representation. 21, 22, 39, 42, 88

Notation

Variables associated with spline construction

$\mathcal{B}_i^p$	i-th univariate Bernstein polynomial of degree p
$\mathbf{C}$	Vector of control points $\mathbf{C}$
p	Degree associated to a spline basis, for multivariate splines, its vector equivalent $\mathbf{p}$ is used
$\mathbf{J}$	Jacobian of a mapping ϕ, e.g., a spline, defined as $J_{ij} = {}^{\partial\phi_i}/_{\partial\xi_j}$
$\mathbb{A}_{\Xi}$	Knot insertion matrix that maps the original control points to the ones associated to the new knot vector after knot insertion
Ξ	Knot vector associated to a B-Spline or NURBS, with $\Xi = [t_1, \dots, t_{n_\mathbf{C}+p+1}]$
$\boldsymbol{i}$	Multidimensional index $\{i, j, k, \dots\}$ to identify a control point or weight in a multi-dimensional control point grid
$\mathbf{n}$	Normalized normal vector
$n_\mathbf{C}$	Number of control points $\mathbf{C}$
$\boldsymbol{\xi}$	Coordinates in parametric space $\boldsymbol{\xi} = [\xi_1, \dots, \xi_{\tilde{d}}]$
$\tilde{d}$	Dimension of the parametric domain
$\mathbf{x}$	Coordinates in the physical space $x = [x_1, \dots, x_d]$
d	Dimension of the physical domain
$\mathcal{N}_i^p$	i-th univariate B-Spline basis function of degree p
$\mathcal{R}_i^p$	i-th univariate NURBS basis function of degree p
$\mathcal{C}$	Univariate spline / curve
$\mathcal{S}$	Bivariate spline / surface

$\mathbf{W}$	Vector of weights $[\mathrm{w}_1, \ldots, \mathrm{w}_{n_\mathbf{C}}]$ associated to a control point of a rational spline
$\mathcal{W}$	Denominator (spline) of rational splines

Variables associated with microstructure construction

$\mathcal{T}$	Deformation function, defining the macro or outer shape of the microstructure
$\boldsymbol{\chi}$	Vector of independent design variables
$\boldsymbol{\mathcal{M}}$	Complete microstructure $\boldsymbol{\mathcal{M}} = \cup_i \mathcal{M}$
M	Microtile, base cell to be inserted into the microstructure periodically
$\mathbf{x}^M$	Evaluation points defined in the unit cube along with the micro-tile parametrization $\tilde{M}$, used to retrieve local micro-tile parameters $\tilde{P}^M$
$\tilde{M}$	Parametrization of the microtile prior to its insertion into the deformation function, based on the micro-tile parameters $\tilde{P}^M$
$\tilde{P}^M$	Design variables, characterizing the form of one individual micro-tile before composition
$n_{\boldsymbol{\chi}}$	Total number of design variables
$\mathcal{P}$	Parameter spline, used to retrieve local micro-tile parameters $\tilde{P}^M$
$\mathbf{P}^M$	Vector of design variables $\tilde{P}^M$, associated to the parameter abstraction

Entities for Numerical Analysis

ε_{abs}	Absolute convergence tolerance
$\mathbf{k}$	Additional constraints, e.g., mass constraints, manufacturing constraints, etc.
$a_\mathbf{u}$	Bilinear form of the elasticity formulation, with $a_\mathbf{u} = a_{\mathbf{u},\lambda} + a_{\mathbf{u},\mu_1} + a_{\mathbf{u},\mu_2}$
a_θ	Bilinear form of the heat equation
$\partial\Omega$	Boundary of the computational domain
$\boldsymbol{\sigma}$	Stresses associated to the elasticity problem
$\hat{\mathbf{u}}$	Coefficients of the displacement field
$\hat{\theta}$	Coefficients of the temperature field discretization
ρ	Material density
Γ_D	Boundary, where Dirichlet conditions (displacements, temperatures) are imposed
$\mathbf{g}$	Displacement constraints, Dirichlet boundary condition to the elasticity problem
$\mathbf{u}$	Displacement field $\mathbf{u}$, its discretization $\mathbf{u}^h$ and discretized field in parametric space $\tilde{\mathbf{u}}^h$

$\mathcal{S}_{\mathbf{u}}$	Function space for displacement field in model problem
λ	First Lamé constant
$\mathbf{h}$	Forces as as applied on Neumann boundary
Ψ	Geometric mapping from the parametric to the physical domain
$\bar{\Psi}$	Mapping of the reference element $\bar{\Omega}_e$ into the parametric domain
$\bar{\Omega}_e$	Reference element
$\mathcal{O}^{\Psi}$	Symbolic representation of the geometry kernel
J^{Γ}	Surface metric for integration on embedded geometries
$\mathbf{q}$	Heat flux
q_{Γ}, $\bar{q}_{\Gamma}$	Heat flux (scaled with material parameters $1/\rho c$) on boundary
f, $\bar{f}$	Heat sources (scaled with material parameters $1/\rho c$)
B	Arbitrary small body within Ω
Q	Internal energy in a given domain
M^{Ψ}	Minors of the geometric mappings Jacobian
$\boldsymbol{\lambda}$	Lagrange multipliers in the optimization problem
$\mathcal{L}$	Lagrangian of the optimization problem
$L_{\mathbf{u}}$	Linear form of the elasticity problem
L_{θ}	Linear form of the heat equation
$\boldsymbol{\epsilon}$	Stresses associated to the elasticity problem
Γ_N	Boundary, where Neumann conditions (forces, heat-fluxes) are imposed
$n_{\hat{\mathbf{u}}}$	Number of Coefficients in the displacement field
$n_{\hat{\theta}}$	Number of Coefficients in the temperature field
n_D	Number of coefficients on the Dirichlet boundary Γ_D in both mentioned PDEs
$\mathcal{J}$	Objective function of the optimization problem evaluates to $J = \mathcal{J}(\bullet)$, $\hat{\mathcal{J}}$ denotes the simplified notation, which only depends on the design variables.
$\mathcal{O}^{\mathbf{X}}$	Symbolic representation of the optimization driver
$\tilde{\Omega}$	Parametric domain
$\tilde{\nabla}$	Gradient in the parametric/reference domain, defined as $\{\partial/\partial\xi_1, \ldots, \partial/\partial\xi_{\tilde{d}}\}$
$\mathbf{c}$	PDE constraint in residual form

Ω	Physical problem domain computational domain
∇	Gradient in the physical domain, defined as $\{\partial/\partial x_1, \ldots, \partial/\partial x_d\}$
ν	Poisson's ratio
μ	Second Lamé constant (shear modulus)
$\mathcal{H}^1$	First-order Sobolev space
e	Specific energy
c	Specific heat capacity
$\mathbf{K}$	System matrix resulting from IGA discretization for the Heat Transfer Problem $\boldsymbol{\theta}$ and elasticity problem $\mathbf{u}$
$\mathbf{F}$	Right-hand side of the linear system of equations resulting from IGA discretization for the Heat Transfer Problem $\boldsymbol{\theta}$ and elasticity problem $\mathbf{u}$
θ_t	Target temperature field
T_Γ	Temperature constraints, Dirichlet boundary condition to the heat transfer problem
θ^h, θ	(discretized) Temperature field in a heat transfer problem. $\tilde{\theta}^h$ is the discretized field in the parametric domain
$\mathcal{S}_\theta^h$, $\mathcal{S}_\theta$	(finite dimensional) Function space for temperature field in model problem
w, w^h	(discretized) Test function. Discrete function in parametric space written as $\tilde{w}^h$. The vector valued test function for the displacement function is written $\mathbf{w}$
$\mathcal{V}^h$, $\mathcal{V}$	(finite dimensional) Test function space for given model problem
k	Thermal conductivity
λ	Thermal diffusivity defined as $k/\rho c$
t	Time
m, m_t	Total (target) mass of a structure
$\tilde{\rho}$, $\tilde{\rho}_t$	(Target) volumetric density
$\mathbf{f}$	Volumetric forces, such as gravitational forces
E	Young's modulus

Publications and Copyrights

Results shown in this thesis have been published in the following articles:

- Jacques Zwar, Gershon Elber, and Stefanie Elgeti. "Shape optimization for temperature regulation in extrusion dies using microstructures". In: *Journal of Mechanical Design* 145.1 (2023)

1 Introduction

This thesis investigates the development and evaluation of methods for the numerical design and optimization of boundary-conforming microstructures that integrate seamlessly into existing Computer Aided Engineering (CAE) workflows. A central aspect is the unification of geometry representation using Computer Aided Design (CAD)-based models for creation, analysis and optimization, and the implementation of a new paradigm for microstructure design based on functional composition between spline-based geometries.

1.1. Motivation

Engineered microstructures are cellular materials with (at least) two different scales, one describing the local geometry, i.e., the local cells or lattices, and the other describing the macroscopic shape of an object. With this definition, a microstructure does not necessarily have to be microscopic, but finds applications across a wide range of different scales. For example, on the nano- to millimeter-sized scale, these microstructures are used in lightweight meta-materials to design structures with high stiffness [24] or auxetic material behavior [35, 87]. In the mid-range, they are used in the design of orthopedic insoles [84] or implants and prosthetics [42, 19], but also in industrial applications such as heat exchangers [106]. As an extension of the definition of a microstructure, there are even efforts to apply the cellular design approach to buildings and structures several meters in length in civil engineering using cement [170].

The use of man-made non-stochastic structures such as honeycomb sandwich laminates in lightweight structures dates back at least to the beginning of the 20^{th} century [44]. These cellular materials are said to be bio-inspired and are often motivated by their low relative density and anisotropic properties [92]. With recent developments in the field of Additive Manufacturing (AM), the use of these microstructures has seen a new upsurge, making novel, topologically intricate designs more accessible. The global AM equipment and materials market has experienced a compound annual growth rate of 25.9 % over the past 28 years, and AM products are projected to reach a market share of approximately 5 % by 2030 [126]. Moreover, AM is increasingly used across various engineering fields, from classical lightweight structures to civil engineering [176] and even the food industry [141]. However, the vast new design space associated with 3D manufacturing methods cannot be fully exploited using conventional engineering approaches, hence the need for numerical design and optimization.

Sophisticated numerical design methods for exploring new design spaces can be divided into three main components: geometric modeling, analysis, and optimization. In the following chapters, we will examine each of these components and discuss their contributions to enhancing the CAE workflow for microstructure design.

Traditionally, microstructured geometries are modeled as cells that are periodically repeated in a Cartesian grid to form the macroscopic or "outer" geometry, e.g., [6, 137, 234, 229]. However, these approaches face two major challenges. First, structures built on such a grid often struggle to accurately represent boundaries, as individual cells are not always completely inside or outside the macro shape. This requires a binary choice for cell inclusion or the development of complex solutions [17] to cut cells without disrupting the outline of the microstructure. Second, these structures are typically defined as mesh connections between points [211], implicit functions [155, 111], or single parametrized cells that are repeated in the macro shape. Converting these into CAD-compatible geometry representations requires extensive post-processing for further steps in the numerical design workflow. Since CAD is often the first step in production, its geometry representation is often the interface for subsequent steps such as analysis, optimization, and manufacturing. Recently, new design paradigms for microstructure design have emerged based on explicit spline representations [72]. These approaches solve both problems by creating spline-based microstructure designs that naturally conform to the macroscopic geometry of a structural component, ensuring compatibility with CAD systems, and improving the overall efficiency of the design workflow.

The second major component, the analysis of microstructured geometries, corresponds to the second step in the traditional CAE workflow. Conventional methods often employ model order reduction and homogenization [11] techniques to cope with the high number of Degrees Of Freedom (DOF) required to represent the topologically complex microstructured geometry. This greatly simplifies the structure by ignoring individual cells and focusing on global (or macroscopic) properties. While it reduces the computational cost, it is not suitable if the scale of the local geometry is not several orders of magnitude smaller than the outer geometry. In cases with about 10 to 50 cells per characteristic global length, however, full-scale models are inevitable. These models, which include all the intricate geometric features, require a large number of DOFs, making them computationally expensive and cumbersome to create, especially if the geometry is created implicitly. In addition, traditional methods such as the Finite Element Method (FEM) with linear elements can only approximate curved or freeform surfaces in a piecewise linear fashion, introducing geometric errors that increase with element size. This approach also requires a separate meshing step, which adds significantly to the overall simulation time. IsoGeometric Analysis (IGA) offers a promising solution to these challenges [53]. By using spline representations, IGA maintains geometric accuracy without the need for meshing, which reduces pre-processing time and has also shown to improve simulation accuracy per DOF [116, 191]. In addition, this method seamlessly integrates with the spline-based CAD models used in modern design processes. As a result, IGA has the potential to improve both the efficiency and accuracy of numerical analysis of microstructured geometries, making it a valuable tool in the optimization framework.

Finally, the third component is optimization itself. The quality of a design is evaluated using an objective function that takes into account factors such as design parameters, field quantities (e.g., temperatures, displacements, or stresses), and/or object properties (e.g., mass). The field quantities are typically the solution to a Partial Differential Equation

(PDE), such as those governing heat transfer, continuum mechanics, or fluid dynamics, and require appropriate analysis. In addition, optimization routines generally perform better with more detailed information, so accurate gradient calculation is critical to the efficiency of the design workflow.

IsoGeometric Analysis pairs well with shape optimization because the design coefficients can serve as design variables, streamlining further geometry parametrization. Consequently, the combination of IGA and shape optimization has received increasing research attention over the last 15 years since IGA was introduced to the scientific community in 2005 [118]. This reflects in the number of publications on these topics[1] as shown in Figure 1.1.

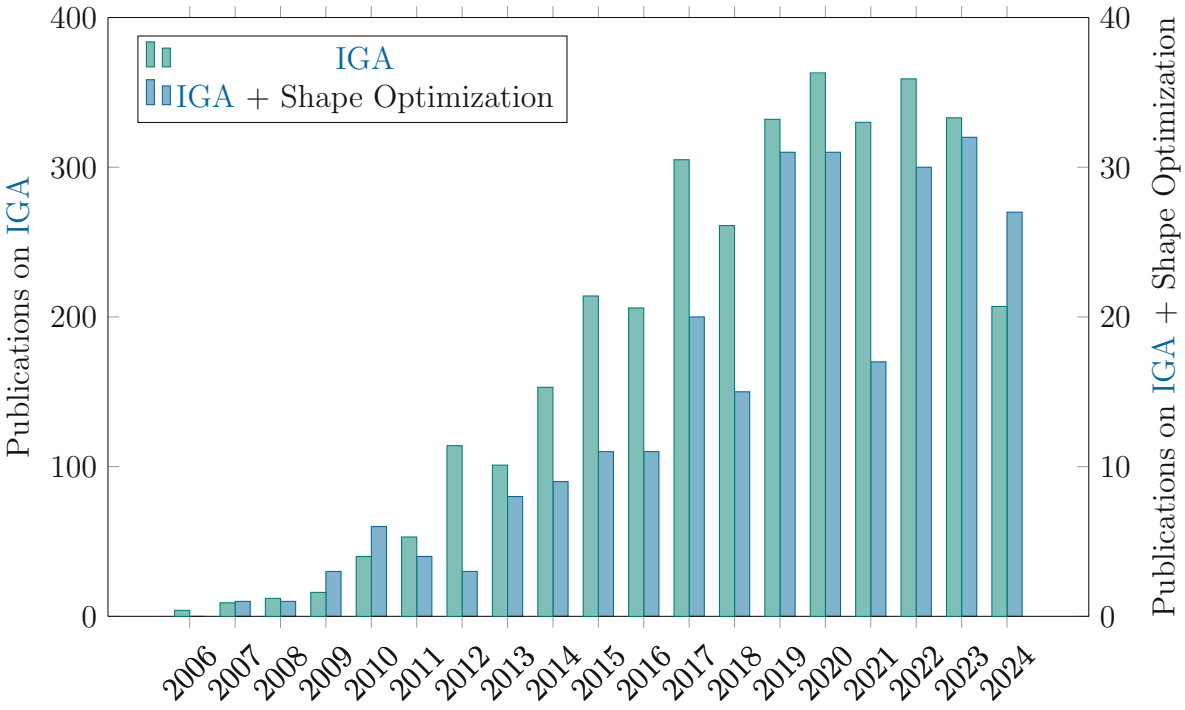


Figure 1.1.: Publications on IsoGeometric Analysis and in combination with shape optimization respectively retrieved from Scopus Search Engine [22].

1.2. Objectives

Traditionally, design, analysis, and optimization have been separate disciplines within the product-development workflow. This thesis aims to provide the necessary methods and concepts for the design and optimization of microstructured geometries that can be seamlessly integrated into existing design frameworks without the need for remeshing or post-processing. By using a consistent geometry representation of the microstructures for geometric modeling and optimization, the goal is to facilitate streamlining and automation. Specifically, this thesis aims to

[1] Search queries were set to include the keywords “isogeometric analysis” and “shape optimization” as of June 16th, 2024. Search results were not manually screened.

Unify the three components of shape optimization into one framework Given a unified geometry representation, the thesis aims to build a combined workflow that integrates geometric modeling, analysis, and design. Building on advanced analysis techniques like IGA, this framework will provide as much information as possible to improve the convergence of optimization algorithms. Starting with an initial design of the macroscopic geometry, the microstructure can then be designed with a minimal set of input variables, enabling a more efficient exploration of the design space.

Investigate applications of boundary-conforming microstructures This thesis investigates the use of boundary-conforming microstructures in various model problems, particularly in the context of heat transfer and structural optimization. By assessing the current state of the art and ongoing research efforts, we aim to evaluate the performance of microstructured geometries in different contexts and compare it to existing workflows. This will help to identify possible applications and further research potential.

Explore different optimization strategies and algorithms By investigating different optimization algorithms and strategies, i.e., objective functions or design parameters, this thesis attempts to evaluate the applicability of conformal microstructures to increasingly complex geometries and model problems.

In order to achieve these objectives, we first need to introduce the relevant concepts and equations, laying the groundwork for the detailed investigations presented in subsequent chapters.

1.3. Thesis Outline

After this brief introduction and motivation for the work presented, the remainder of this document is organized as follows:

Chapter 2 deals with the construction of microstructures and focuses on providing insights into geometry modeling. First, we provide an overview of the state-of-the-art in microstructure design and characterize the different classes of microstructures. Then, in Section 2.2.3, the mathematical prerequisites for the construction of specific microstructures are introduced, with a deep dive into spline fundamentals, where different spline types and the necessary mathematical tools and spline operations for the construction of boundary-conforming microstructures are introduced. It then details the actual construction process and the geometric derivatives, which play a key role in the optimization framework.

Chapter 3 focuses on the analysis and optimization aspects. We start with an introduction to the physical background and the model problem. This is followed by a concise introduction to IsoGeometric Analysis (IGA) and the necessary tools to numerically solve the derived Partial Differential Equations (PDEs). After that, Section 3.3 provides an introduction to shape optimization, drawing an analogy to Topology Optimization (TO) and introducing the optimization driver and framework. It also provides the necessary ingredients to properly define the optimization problem. The following section is dedicated to discrete adjoints, showing how derivatives of the optimization problem can be efficiently computed. The chapter concludes with a short discussion comparing the chosen parametric shape optimization approach with alternative strategies.

In Chapter 4, we apply the methods to heat transfer problems. First, we cover preliminary work and introduce and compare different objective functions from literature. We then apply the optimization framework to a simple problem in a rectangular 2D domain and use this test case to compare the cost of different optimization algorithms. We then increase the complexity in Section 4.3 by applying the methods to an extrusion die. After introducing the design challenges, we demonstrate the application of the presented methods to more realistic and intricate 3D geometries.

In Chapter 5, the microstructures are used to optimize elastic structures, considering linear elasticity for a variety of structures subjected to external loads. We begin by introducing additional constraints and describing the objective function. Section 5.2 demonstrates the performance of the methods on a benchmark problem (cantilever design). Then, we increase the geometric complexity by applying the methods to a bending arch, highlighting the benefits of boundary-conforming lattice structures. This test case is used to compare different optimization strategies, including local, global, and concurrent approaches. The chapter concludes with a brief discussion of local parameter abstraction in the context of structural optimization.

Chapter 6 concludes the thesis by summarizing the present work, highlighting the contributions made, and identifying potential areas for future research based on the results of this work.

2 Construction of Microstructures

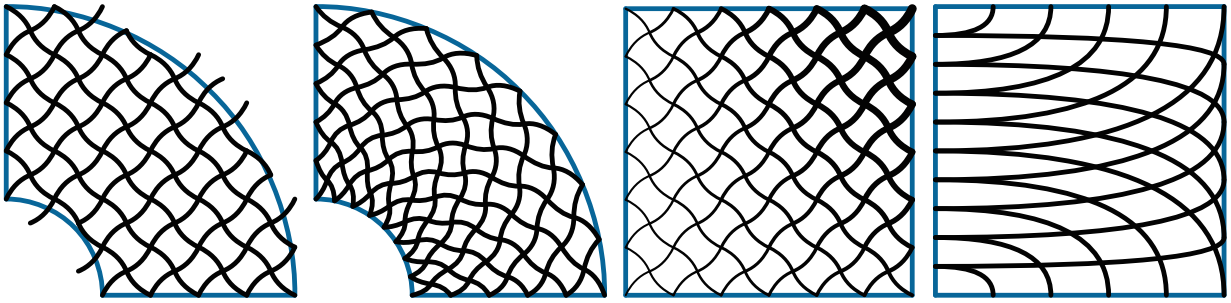
2.1. An Introduction to Microstructures

Microstructured geometries, encompassing both micro and macro scales, are inspired by nature's intricate designs such as the ones found in bone structures, wood, and sponges [91]. These structures often exhibit structural behaviour tailored to their loading conditions, like excellent stiffness-to weight ratio relative to comparable foam-structures [63], which is why engineers have long been inspired by these natural designs when constructing new materials or geometries, leading to a wide range of applications in various fields of engineering. While they are most prominent in classical lightweight applications such as aerospace [83] or automotive engineering [1], they have also proven valuable in medical applications such as bone implants [5, 19] or orthopedic devices [84] and in the design of compliant mechanisms [74]. It is important to note that the term *microstructure*[1] in this context does not refer exclusively to small-scale structures, but rather emphasizes the presence of different spatial scales.

Since the advent of Additive Manufacturing (AM) technologies, new design methods have been explored to fully exploit the vast design space made accessible through this development. Through intricate micro scale geometries, these structures can mimic complex anisotropic material behavior on a global scale, e.g., by exploiting local instabilities [127], or create meta-materials exhibiting auxetic properties such as negative Poisson's ratio [35, 73].

There are several methods for designing microstructures. These construction methods can be distinguished based on the global characteristics of the resulting microstructure, as derived from [128], see Figure 2.1. Most prominently, this classification distinguishes between periodic and non-periodic structures. Non-periodic structures, cf. Figure 2.1d, such as those resulting from force flow-based approaches [138, 3] or those based on global Topology Optimization (TO) [230], and - to a certain extent - foams [92], do not exhibit a regular, recurring pattern. In contrast, periodic structures (Figures 2.1a to 2.1c) have one or more repeating base cells that fill the entire structure in a grid-like fashion. Periodic microstructures are further categorized in terms of their conformity. While conformal microstructures, see Figure 2.1b, align with the exterior geometry, non-conformal – or uniform – structures result from the repeated insertion of a parametrized base cell into a

[1] We will use the terms *microstructure* and *lattice structure* interchangeably throughout this work, although an alternative nomenclature may be used in other literature.



(a) Non-conformal microstructure **(b)** Conformal microstructure **(c)** Functionally graded microstructure **(d)** Non-periodic microstructure

Figure 2.1.: Classification of microstructured geometries, based on [128]

(Cartesian) grid, progressively filling the entire computational domain, cf. e.g. [137, 234]. This approach represents the macro shape in a binary fashion, which can require a high resolution to accurately represent intricate geometries, and special precautions must be taken at the domain boundary, since cells may be only partially inside the domain [17]. To illustrate this, Figure 2.1a only deletes elements if they are fully outside the predefined outer geometry, which leads to partial penetration of the boundary.

Further, periodic microstructures can be categorized as either functionally graded or homogeneous. Homogeneous structures replicate the base cell without modification, while functionally graded structures, see Figure 2.1c, adapt the parametrized base cell based on local constraints or loading conditions, introducing variations in, for example, thickness, angles, and other geometric parameters. In the context of (structural) optimization, inhomogeneous or functionally graded microstructures are often used in combination with TO, particularly with the Solid Isotropic Material with Penalization (SIMP) method and its variants [30]. These methods aim to find an optimal material distributions based on external loads and constraints. Approaches based on TO are twofold. On the one hand, it is used to design an optimized macro-geometry, where the outcome of this optimization serves as an input for generating a conformal mesh from the boundary representation. This mesh is then systematically filled with microstructures [188]. On the other hand, parametrized cells are utilized to achieve intermediate material densities on a global scale, i.e., discarding the penalization step [38], often involving methods for homogenization to reduce computational efforts during optimization [234]. TO-based approaches will be the subject of further discussion in Section 3.3.4.

Unlike uniform microstructures, where the initial structure is often limited to box-like geometries, conformal structures provide a more natural representation because the meshes are aligned with the macroscopic geometry. In the context of this thesis, we will focus on conformal meshes that are recoverable in a Computer Aided Design (CAD)-compliant representation. Comprehensive discussions on different construction techniques can be found in literature, e.g., with a focus on geometric modeling and microstructural design [192, 210], on Additive Manufacturing (AM) and possible applications [69, 194], or on structural optimization [181, 143, 128].

Recently, there have been a growing number of efforts to develop methods for conformal structures. In some of these approaches, the macroscopic structure is first discretized using a piecewise linear hexahedron mesh [77] with consecutive truss size optimization [167], while the inner structure is defined based on control points within the unit cell. These

control points are subsequently mapped onto the hexagonal mesh, and the lattice structure interpolates between these mapped control points in an explicit manner. Other approaches consist of representing the macro geometry as a set of vertices and connections and then use implicit functions in a local coordinate system to represent the local cell geometry [211]. Because both methods represent the macrogeometry in a piecewise linear fashion, the resulting microstructure cannot accurately represent curved boundaries. In addition, although the latter method can be used to sample voxel-based geometries for AM, both methods require extensive post-processing to recover a CAD-compatible representation. In [208], building from an existing CAD representation, sphere packing is applied, where a spline based truss structure is created by connecting the center nodes, building a boundary conforming, non-periodic lattice structure. However, the resulting microstructure also only approximates the boundary of the macro shape with piecewise linear trusses and nodes.

To address the challenges posed by the reduced continuity of the boundary and the need for CAD compatibility, the method proposed by Elber [72] employs functional composition of tensor product splines. In particular, this approach excels in its ability to control the degree of continuity to a certain extent. Moreover, it integrates a fully explicit volumetric representation of the domain, rendering it particularly suitable for IsoGeometric Analysis (IGA) (cf. Section 3.2). While the use of functional composition of splines in the design process is a relatively recent development, its application has already been demonstrated in the design of wing structures [102], where stringers have been modeled using spline compositions.

In the following, Section 2.2 provides a brief introduction to splines, which form the basis for representing free-form geometries in computer graphics and engineering. This chapter will also provide insights into the various spline operations in Section 2.2.3, that are required to construct microstructures using the method proposed in [72]. The construction process is then detailed in Section 2.3, along with the available extensions. Finally, Section 2.4 provides the mathematical tools to differentiate the resulting geometry with respect to its parametrization.

2.2. Splines

Splines are a versatile mathematical tool for describing complex geometries in arbitrary dimensions. Most commonly used in computer graphics and Computer Aided Design (CAD), they play a central role in representing mechanical drawings, architectural designs, and a variety of intricate geometries. Historically, modern spline representations originated in the automotive industry, when French engineers Pierre Bézier [32] of Renault and Paul Faget de Casteljau of Citroën developed similar ideas for representing curves and surfaces based on interpolation with Bernstein polynomials [31]. These advances ultimately led to the development of one of the first CAD systems UNISURF [33]. Shortly thereafter, in the 1970s, new fundamental contributions were made with Richard Riesenfeld's dissertation on B-Splines [187], followed by Kenneth Versprille's dissertation [220] on rational B-Splines. The latter - usually referred to as Non-Uniform Rational B-Splines (NURBS) - have since become the de facto industry standard for CAD. For a more detailed historical overview, the reader is referred to [189].

Many of the fundamental advantages of splines have led to their widespread adoption. They allow the creation of smooth and arbitrarily continuous geometries, providing great

flexibility for geometric modeling, including the construction of objects in arbitrary dimensions and embedded geometries such as three-dimensional surfaces. In addition, splines are particularly well suited for data interpolation because their piecewise polynomial (rational) definition making them less susceptible to oscillations, compared to the classical Lagrange interpolation [81, Chapter 7.4].

Over time, a multitude of specialized spline types has evolved from Bézier splines and B-Splines, each tailored to their specific application. For example, Truncated Hierarchical B-Spline (THB) [90] and T-Splines [201] allow for local refinement and hence are used in CAD to reduce the number of control points required for the same level of accuracy to approximate local features, e.g., in surface reconstruction from point clouds [89].

2.2.1. Non-Uniform Rational B-Splines (NURBS)

NURBS allow the construction of intricate geometries, including conic sections such as circles. A NURBS curve $\mathcal{C}$ of degree p is determined by its knot vector $\boldsymbol{\Xi} = [t_1, \dots, t_{n_\mathbf{C}+p+1}]$ with $t_i \leq t_j$ if $i < j$, a set of positive non-zero weights $\mathbf{W} = [\mathrm{w}_1, \dots, \mathrm{w}_{n_\mathbf{C}}]$ and a vector of control points $\mathbf{C} = [\mathbf{C}_1, \dots, \mathbf{C}_{n_\mathbf{C}}]$. The curve is then constructed by interpolating between the control points, using piecewise rational polynomial basis functions $\mathcal{R}_i^p$ in the form

$$\mathcal{C}(\xi) = \sum_{i=1}^{n_\mathbf{C}} \mathcal{R}_i^p(\xi)\mathbf{C}_i \quad \text{for } \xi \in \left[t_{p+1}, t_{n_\mathbf{C}+1}\right[\quad . \tag{2.1}$$

In this representation, the control points $\mathbf{C}_i \in \mathbb{R}^d$ are of an arbitrary physical dimension d and can even be used to interpolate scalars if $d = 1$. The rational basis functions are composed of their polynomial B-Spline counterparts $\mathcal{N}_i^p$ using

$$\mathcal{R}_i^p(\xi) = \frac{\mathrm{w}_i \mathcal{N}_i^p(\xi)}{\sum_{j=1}^{n_\mathbf{C}} \mathrm{w}_j \mathcal{N}_j^p(\xi)} = \frac{\mathrm{w}_i \mathcal{N}_i^p(\xi)}{\mathcal{W}(\xi)} \quad . \tag{2.2}$$

Here, the denominator function – which is a scalar spline itself – is denoted $\mathcal{W}$ to simplify the expression. This construction can be interpreted as a perspective projection from a $(d+1)$ dimensional space into dimension d, cf. [178]. The polynomial B-Spline basis functions $\mathcal{N}$ have very distinct properties, notably

- non-negativity, i.e., $\mathcal{N}_i^p(\xi) \geq 0$
- formation of a partition of unity, i.e., $\sum_i \mathcal{N}_i^p(\xi) = 1$
- $\mathcal{N}_i^p$ attaining exactly one maximum on the interval $\left[t_{p+1}, t_{n_\mathbf{C}+1}\right[$ if $p > 0$

The same properties also hold for the rational basis functions $\mathcal{R}$. From the partition of unity and Equation (2.2) follows, that polynomial basis functions are equal to their rational counterparts if all weights are equal. B-Spline basis functions are defined recursively using the Cox-de Boor formula:

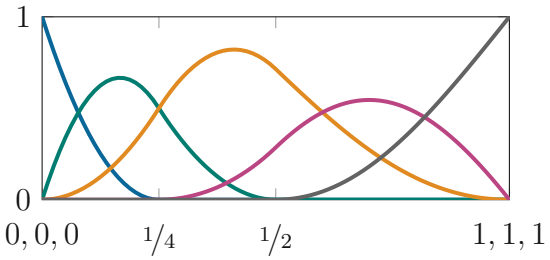
$$\mathcal{N}_i^0(\xi) := \begin{cases} 1 & \text{if} \quad t_i \leq \xi < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad , \tag{2.3}$$

$$\mathcal{N}_i^p(\xi) := \frac{\xi - t_i}{t_{i+p} - t_i} \mathcal{N}_i^{p-1}(\xi) + \frac{t_{i+p+1} - \xi}{t_{i+p+1} - t_{i+1}} \mathcal{N}_{i+1}^{p-1}(\xi). \tag{2.4}$$

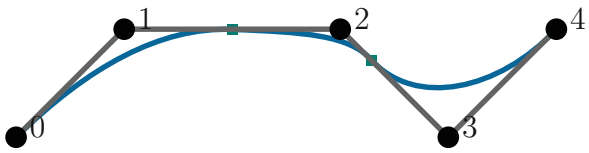
The interval between two consecutive knots $\left[t_i, t_{i+1}\right[$, is called a knot-span. According to Equation (2.3), the B-Spline basis function $\mathcal{N}_i^p$ is only non-zero on the interval $\left[t_i, t_{i+p+1}\right[$, i.e., it exhibits only local support on $p+1$ knot-spans. This is an important feature, as it limits the influence of a control point locally, allowing for more flexibility in geometric design. The knot vector also determines the continuity of the basis functions. A knot vector in the form

$$\Xi = \Big[\underbrace{t_1, \ldots, t_1}_{p+1}, t_{p+2}, \ldots, t_{n_\mathbf{C}}, \underbrace{t_{n_\mathbf{C}+1}, \ldots, t_{n_\mathbf{C}+1}}_{p+1}\Big] \quad , \tag{2.5}$$

is called open if the multiplicity of the first and last knot is $(p+1)$. Although it is possible to describe periodic knot vectors without repetitions of the first and last knot, open (or clamped) knot vectors are preferred as they lead to an interpolatory property at the first and last control point of the curve. The multiplicity of consecutive knots further determines the continuity of the basis function and therefore the continuity of the spline curve. The spline is C^∞-continuous everywhere and C^{p-m_i} at knot t_i, where m_i denotes the multiplicity of knot t_i. The continuity of the associated spline can be deduced directly from the basis functions. Further, any linear transformation to the knot vector, i.e., multiplication or addition with a non-zero scalar, will not alter the physical representation of the spline. An exemplary NURBS spline curve and its associated basis functions are depicted in Figure 2.2.



(a) Rational basis functions over knot-vector, each associated to one control point



(b) Nurbs curve in 2D with control points and element borders (green marks)

$\Xi =$	$\left[0,\ 0,\ 0,\ 1/4,\ 1/2,\ 1,\ 1,\ 1\right]$
$\mathbf{C} =$	$\left[(0\ 0),\ (1\ 1),\ (3\ 1),\ (4\ 0),\ (5\ 1)\right]$
$\mathbf{W} =$	$\left[1,\ 1,\ 1,\ 4/5,\ 1\right]$

(c) Knot vector, control point list and weights required to construct the spline

Figure 2.2.: Basis function and spline representation of a NURBS of degree $p = 2$, created with `splinepy` [136].

The same considerations and properties of a curve can be applied to higher dimensional splines such as surfaces and volumes/multivariates. These geometric entities are constructed in a tensor-product fashion, resulting in multivariate functions. For example,

a NURBS surface is constructed of a control grid $\mathbf{C}_{i,j}$ with associated weights $\mathbf{W}_{i,j}$ using basis functions $\mathcal{R}^{p,q}_{i,j}$, with:

$$\mathcal{R}^{p,q}_{i,j}(\xi,\eta) = \frac{\mathrm{w}_{i,j}\mathcal{N}^p_i(\xi)\hat{\mathcal{N}}^q_j(\eta)}{\sum_{k,l}\mathrm{w}_{k,l}\mathcal{N}^p_k(\xi)\hat{\mathcal{N}}^q_l(\eta)} \quad . \tag{2.6}$$

Here, both basis functions $\mathcal{N}$ and $\hat{\mathcal{N}}$ are assigned an accompanying knot vector $\boldsymbol{\Xi}$ and $\hat{\boldsymbol{\Xi}}$, and degrees p and q, which define the basis functions in their respective parametric dimension. To simplify the notation, a vector-valued notation is used in the following, with the parametric coordinates $\boldsymbol{\xi} = \{\xi, \eta, \dots\}$, the degrees-vector $\mathbf{p} = \{p_1, p_2, \dots\}$ and a unique multi-index $\boldsymbol{i} = \{i, j, \dots\}$, leading to

$$\mathcal{S}(\boldsymbol{\xi}) = \sum_{\boldsymbol{i}} \mathcal{R}^{\mathbf{p}}_{\boldsymbol{i}}(\boldsymbol{\xi})\mathbf{C}_{\boldsymbol{i}} \quad . \tag{2.7}$$

A more extensive description on NURBS and B-Spline geometries, including efficient algorithms for their evaluation, differentiation and construction can be found in [178, 189], the open-source library splinepy used in the context of this thesis for the construction, visualization, and manipulation of splines is provided by ILSB [136].

2.2.2. (Rational) Bézier Splines

Even though (rational) Bézier splines predate NURBS historically, they can be considered a special case of NURBS. That is, they contain no interior knots. In the following, we further assume that all Bézier splines map from the (multivariate) unit cube $[0,1]^d$ into physical space. Under these conditions, the recursive expression (2.3) describes a Bernstein polynomial

$$\mathcal{B}^p_i(\xi) = \binom{p}{i-1}\xi^{i-1}(1-\xi)^{p-i+1}, \quad i = 1, \dots, n_{\mathbf{C}} \quad , \tag{2.8}$$

that forms the basis for a Bézier spline (with $n_{\mathbf{C}} = p+1$)[2]. As for NURBS, rational Bézier splines are obtained using a projection as in Equation (2.2). However, contrary to B-Splines and NURBS, Bézier type splines exhibit global support, as they contain only a single knot span.

In principle, it is possible to represent any (rational) Bézier spline using a NURBS or B-spline, and vice versa, it is possible to represent a NURBS as a set of Bézier splines. While the transformation from a Bézier type to a NURBS is trivial, the reverse transformation can be achieved using the methods presented in Section 2.2.3.

2.2.3. Spline Operations

A wide array of geometric algorithms is available to manipulate and modify splines. These include, for example, the complementary operation to knot insertion, i.e., knot removal,

[2] In other literature, the indexation for Bézier splines might be from $0, \dots, p$. In the context of this thesis, control points and basis functions will be counted $1, \dots, n_{\mathbf{C}}$ to conform to B-Spline-type spline notation.

and techniques to adjust the degree of a given spline without introducing geometric or parametric alterations. Furthermore, many algorithms and techniques related to creating geometries exist, including interpolations, projections, and reparametrizations. However, a detailed discussion of these operations is beyond the scope of this thesis. For a more comprehensive exploration of the vast realm of spline operations, interested readers are referred to [178].

Knot Insertion and Bézier Extraction

The term knot insertion refers to the change of basis, which alters the knot vector $\boldsymbol{\Xi}$ without changing the spline geometrically and without directly affecting the image of the spline function. The process for knot insertion is demonstrated for B-Spline curve $\mathcal{C}$, however, the same considerations are also valid for NURBS, where the same algorithm is applied to the weights. Here, an additional knot is introduced into the knot vector, effectively creating a new spline $\tilde{\mathcal{C}}$, which meets the condition:

$$\mathcal{C}(\xi) = \tilde{\mathcal{C}}(\xi) \quad \forall \xi \in \left[t_{p+1}, t_{n_{\mathbf{C}}+1}\right] \quad . \tag{2.9}$$

In order to satisfy this equation, a new set of control points $\tilde{\mathbf{C}}$ must be determined - since the basis functions result directly from the knot vector. Considering the insertion of the new knot $\tilde{t} \in [t_k, t_{k+1})$, the new control points can be obtained using

$$\tilde{\mathbf{C}}_i = \alpha_i \mathbf{C}_i + (1 - \alpha_i)\mathbf{C}_{i-1} \quad , \tag{2.10}$$

with

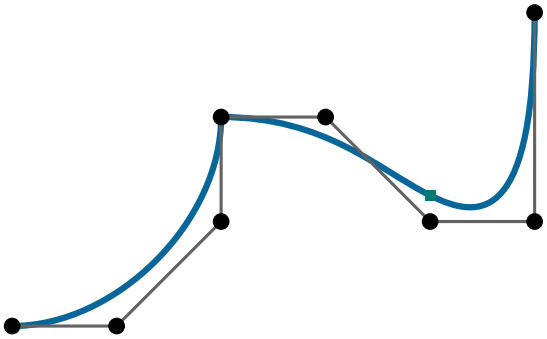
$$\alpha_i = \begin{cases} 1 & i \leq k - p \\ \frac{\tilde{t} - t_i}{t_{i+p} - t_i} & k - p + 1 \leq i \leq k \\ 0 & k + 1 \leq i \end{cases} \quad . \tag{2.11}$$

This process is demonstrated in Figure 2.3. Knot insertion is commonly used for refinement, allowing for more precise, localized control and, thus, design flexibility. Further, by increasing the multiplicity of a specific knot through its repeated insertion into the knot vector, the continuity between knot spans diminishes. This process can be repeated until the spline's continuity is reduced to C^0, which allows the individual knot spans to be reinterpreted as new (Bézier) splines. In the context of geometric modeling, this reinterpretation is generally referred to as Bézier extraction.

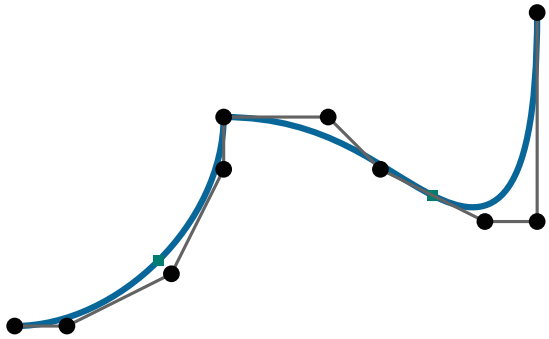
The method described by Equation (2.10) can be extended to insert a single knot multiple times in one step [178] and can also be used to insert multiple knots in one pass. The equation further establishes a linear relationship between the control points of the original spline and the control points of the refined spline. This observation can be utilized to rewrite Equation (2.10) as a matrix product in the following format

$$\tilde{\mathbf{C}} = \mathbb{A}_{\boldsymbol{\Xi}}\, \mathbf{C} \quad . \tag{2.12}$$

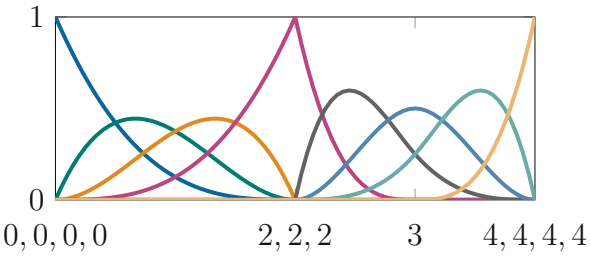
This equation can also be used for higher dimensional splines to perform knot insertions along several parametric dimensions in succession, resulting in a series of matrix multiplications. The sparsity of the resulting matrix increases the computational efficiency of this operation and enables storage of the matrix in memory after computing it once, even if the control points undergo changes in the meantime.



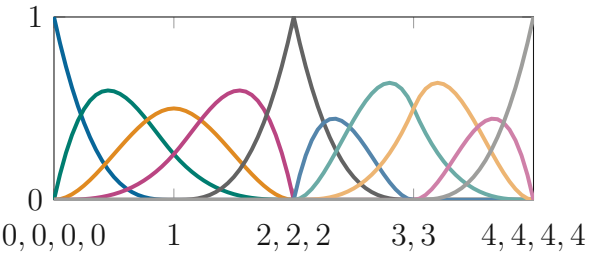
(a) Spline and control point polygon before knot insertion



(b) Spline and control point polygon after knot insertion



(c) Associated basis functions before knot insertion



(d) Associated basis functions before knot insertion at positions $[1, 3]$

Figure 2.3.: Modification of the control points and basis function through knot insertion. The individual knots are displayed under the graphs along with their multiplicities. When the multiplicity is as high as the degree, i.e., $p = m_i$, the basis functions become only C^0 continuous and the spline becomes interpolatory at the position of the repeated knot.

Addition

Besides refinement operations, which primarily increase design flexibility, it is also possible to define arithmetic operations between two splines, where the result is again a spline. Although their use may seem rather theoretical, they can, for example, be used to interpolate between data sets. To calculate the sum of non-rational splines, it is necessary to first unify their representations so that they share the same basis functions. Thereafter, their sum can be obtained by simply adding their respective control points. For (multivariate) B-Splines, this requires (1) degree elevation [51] of the lower order spline until both splines share the same degrees in all parametric dimensions and (2) knot insertion until both splines share the same knot vectors.

This approach is also applicable to rational splines. However, in cases where they possess non-matching weight functions, it is necessary to first find a common denominator via cross-multiplying with the respective denominator splines. Note, that the exact same procedure can be employed for subtraction. For a more comprehensive description, the interested reader is referred to [71].

Multiplication

Similar to addition, it is also possible to exactly compute the product of two splines and display it as a spline itself. This requires that the multiplication of their respective control point types is defined. For instance, a scalar-valued spline can be multiplied by a vector-valued spline, resulting in a vector-valued spline. The multiplication of two Bézier spline curves $\mathcal{C}$ and $\tilde{\mathcal{C}}$ with degree p and $\tilde{p}$ results in a new Bézier spline with degree $(p+\tilde{p})$. Similarly, when multiplying multi-variate Bézier splines, the degree of the resulting spline along each parametric axis is the sum of the degrees of the original splines along that respective axis. The new control points $\bar{\mathbf{C}}$ of the resulting spline can be determined using [200, 71][3]

$$\bar{\mathbf{C}}_{k+1} = \sum_{i=\max(0,k-\tilde{p})}^{\min(k,p)} \mathbf{C}_{i+1} \cdot \tilde{\mathbf{C}}_{k-i+1} \frac{\binom{p}{i}\binom{\tilde{p}}{k-i}}{\binom{p+\tilde{p}}{k}} \quad \text{for } k = 0, \dots, (p+\tilde{p}) \quad . \tag{2.13}$$

This formula can be extended for higher-dimensional geometries, e.g. as done in [239], allowing the multiplication of surface-surface, or volume-volume splines. It also holds for rational Bézier splines. These require the numerator and denominator to be multiplied individually.

The same calculations are also attainable for NURBS and B-Splines. Corresponding algorithms are presented in, e.g., [46] and [179]. It is further possible to determine the control points of the product spline by setting up and solving a linear system of equations through evaluation of the basis functions and products at a specific set of points. This approach is adopted to avoid the high implementation effort and computational costs associated with these methods [71]. In the following, only (rational) Bézier splines will be considered for the construction of microstructures – which can be extracted from NURBS and B-Splines.

Composition

By employing the techniques described above, it becomes feasible to establish a symbolic composition, $\mathcal{S} \circ \tilde{\mathcal{S}}$, between two splines. Notably, Bézier splines are continuously differentiable C^∞ across their entire parametric domain, making them particularly suited for performing compositions. In contrast, the composition between two tensor-product B-Splines is generally not a tensor-product spline [62]. Therefore, our focus in this context will be directed toward the composition involving (rational) Bézier splines. Spline composition has existed for several decades [71]; however, recently there have been concerted efforts to explore its applications more thoroughly [207]. For example, compositions serve as a valuable extension to the traditional Free-Form Deformation (FFD) technique [199, 95] (cf. Section 3.3), particularly in shape optimization for CAD geometries [114]. Moreover, they find utility in transforming trimmed surfaces into tensor product surfaces [75].

In the following, the method will be demonstrated for a surface-curve composition, although the same methods extend to higher dimensions so long as the physical dimension of the inner function matches the parametric dimension of the outer function. Let $\mathcal{C}$ be

[3]Note, to ensure uniform indexation, there is an index-shift compared to the references in [200, 71]

a univariate (non-rational) Bézier curve with degree p in 2D, and $\mathcal{S}$ be a bivariate Bézier surface with degrees $\mathbf{p} = (\tilde{p}_1, \tilde{p}_2)$ in 2D, then the composed spline $\tilde{\mathcal{C}}$ fulfills

$$\begin{aligned}\tilde{\mathcal{C}}(\xi) &= (\mathcal{S} \circ \mathcal{C})(\xi) = \mathcal{S}\big(\mathcal{C}^{\mathrm{x}}(\xi), \mathcal{C}^{\mathrm{y}}(\xi)\big) \\ &= \sum_{i=1}^{\tilde{p}_1+1} \sum_{j=1}^{\tilde{p}_2+1} \mathcal{B}_i^{\tilde{p}_1}\big(\mathcal{C}^{\mathrm{x}}(\xi)\big)\, \mathcal{B}_j^{\tilde{p}_2}\big(\mathcal{C}^{\mathrm{y}}(\xi)\big)\, \mathbf{C}_{ij} \quad ,\end{aligned} \tag{2.14}$$

where $\mathcal{C}^{\mathrm{x/y}}$ describe the x and y components of the spline curve, respectively. Equation (2.14) shows, how the problem of spline composition is reduced to finding the composition between a scalar-valued spline, e.g., $\mathcal{C}^{\mathrm{x}}$, with a Bernstein basis function, cf. Equation (2.8). Note, that this is only possible, given $\mathcal{C}^{\mathrm{x/y}}(\xi) \in [0,1]\, \forall \xi \in [0,1]$. With

$$\mathcal{B}_i^p\big(\mathcal{C}(\xi)\big) = \binom{p}{i-1} \big(1 - \mathcal{C}(\xi)\big)^{p-i+1} \big(\mathcal{C}(\xi)\big)^{i-1} \quad , \tag{2.15}$$

this operation is made possible using solely the expressions presented in the previous sections[4]. The result of this B-Spline composition is a scalar-valued Bézier curve itself, meaning that the scalar control points α_{kl}^{ij}, associated with the multiplication $\mathcal{B}_i^{\tilde{p}_1}(\mathcal{C}^{\mathrm{x}})\mathcal{B}_j^{\tilde{p}_2}(\mathcal{C}^{\mathrm{y}})$ are multiplied with the surface's 3D control points $\mathbf{C}_{ij}$, prior to addition. Figures 2.4a, 2.4b and 2.4c show an example composition between a polynomial curve and surface in 2D.

Analogous considerations hold for rational splines. Specifically, the resulting composition will also be rational if either function is rational. The composition then must be applied separately to the denominator and numerator splines if the outer function is rational. If the inner function is rational, Equation (2.15) can be rewritten as

$$\mathcal{B}_i^p\big(\mathcal{C}(\xi)\big) = \binom{p}{i-1} \frac{\big(\mathcal{W}(\xi) - c(\xi)\big)^{p-i+1} \big(c(\xi)\big)^{i-1}}{\mathcal{W}(\xi)^p} \quad , \tag{2.16}$$

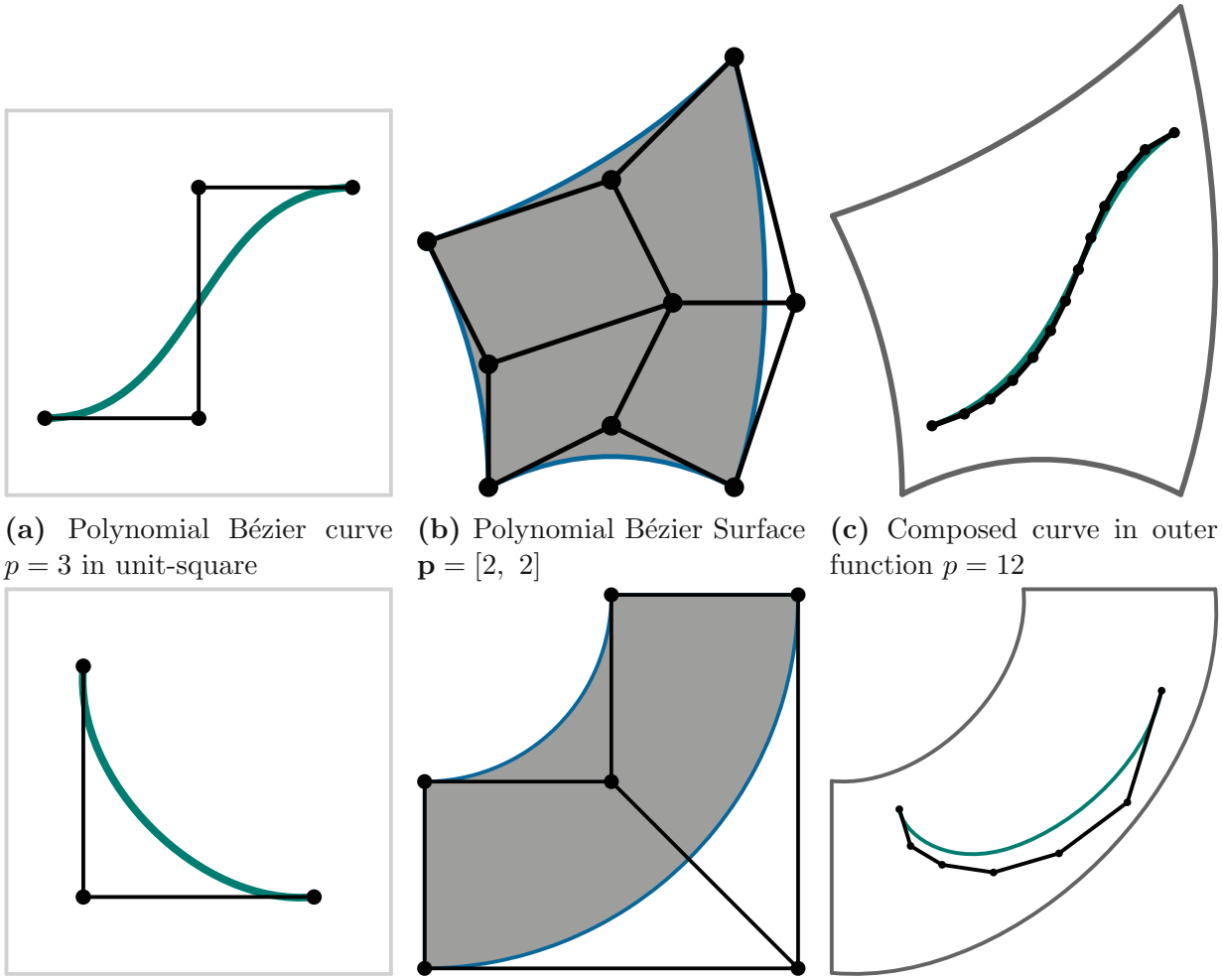
where $\mathcal{W}$ and c denote the denominator and numerator of the rational inner spline function, respectively. The denominator term in Equation (2.16) cancels out if the outer function is rational as well. A rational surface curve composition is depicted in Figures 2.4d, 2.4e and 2.4f.

From the aforementioned description of spline multiplications, it was established that the degree of the product spline equals the sum of the degrees of the factor splines. In the context of composition, the splines are raised to an integer power, specifically, the degree of the outer function. This indicates that the degrees of the inner spline and the corresponding basis function multiply. This process is repeated across all tensor product basis functions. Consequently, the degrees $\mathbf{q}$ of the composition of an outer spline with degrees $\mathbf{p}$ and an inner function with degrees $\tilde{\mathbf{p}}$ follow

$$q_i = \tilde{p}_i \sum_j p_j \quad , \tag{2.17}$$

both for rational and polynomial spline compositions. This results in exceedingly high orders, significantly increasing the computational cost associated with manipulating and

[4] Although not explicitly listed, the scalar 1 can be interpreted as a spline of zeroth degree with control points 1, hence it is possible to express the difference by computing $(1 - \mathbf{C}_i) \forall i$



(a) Polynomial Bézier curve $p = 3$ in unit-square

(b) Polynomial Bézier Surface $\mathbf{p} = [2,\ 2]$

(c) Composed curve in outer function $p = 12$

(d) Rational Bézier curve (quarter circle) $p = 2$ in unit-square

(e) Rational Surface (quarter ring) $\mathbf{p} = [2,\ 1]$

(f) Composed rational curve $p = 6$

Figure 2.4.: Surface-Curve composition between a (rational) Bézier curve and a (rational) Bézier surface. Note, that the degrees of the composed curves are significantly higher.

evaluating such splines. For instance, even composing a quadratic curve with a bi-cubic surface yields a spline of degree 12. Exact degree reduction is feasible in cases where the spline aligns with a parametric axis of the outer function. If this is not enough, lower-order splines can be employed to approximate the resulting geometry.

Derivatives of Splines

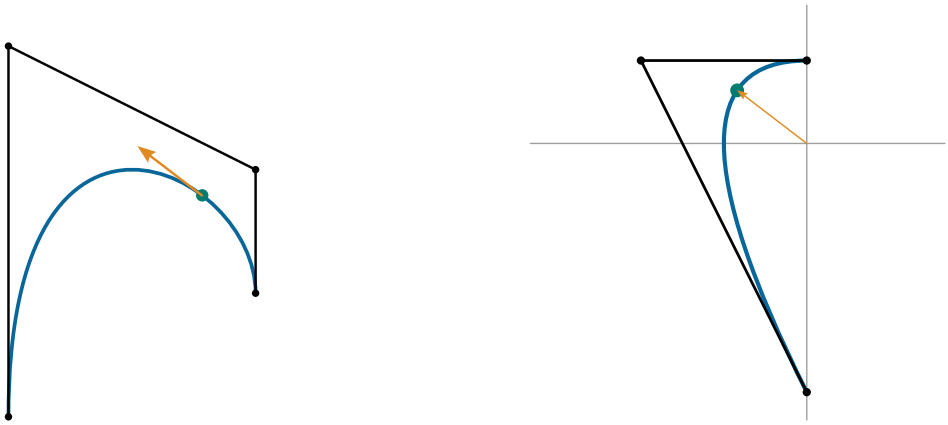
If a spline has at least C^1 continuity over its entire definition range, it is possible to express its derivative as a closed-form representation in terms of another spline function – and therefore this also holds for higher-order derivatives given a sufficient continuity; that is $\mathcal{C}'(\xi) = \frac{\partial}{\partial \xi}\mathcal{C}(\xi)$. This derivative spline describes the rate of change with respect to the parametric coordinates. In the context of curves, this spline is called a hodograph [52]. Such derivative representations find practical applications in various fields of engineering, e.g., in trajectory planning for unmanned aerial vehicles (UAVs) [216]. Furthermore, it

plays an important role in geometric analysis, for example [148] use the exact representation of the determinant of a spline to evaluate the quality of a given spline parametrization.

For a non-rational Bézier curve $\mathcal{C}$ of degree $p > 0$, its derivative $\mathcal{C}'$ is a non-rational Bézier curve of degree $p' = p - 1$ with control points

$$\mathbf{C}'_i = p\big(\mathbf{C}_{i+1} - \mathbf{C}_i\big) \quad . \tag{2.18}$$

An example of a close-form derivative of a Bézier spline is shown in Figure 2.5.



(a) Bézier Spline curve with degree $p = 3$

(b) Spline representation of the derivative with degree $p = 2$

Figure 2.5.: Close-form derivative of a cubic Bézier spline curve. The yellow arrow represents the derivative evaluated at the position of the green point.

Similar considerations can be made for B-Splines, if the multiplicity of any interior knot does not exceed $p - 1$, i.e., the spline remains at least continuously differentiable. Given a B-Spline Curve $\mathcal{C}$ of degree p, with control points $\mathbf{C}$ and an open knot-vector $\boldsymbol{\Xi}$, the close-form of the derivative $\mathcal{C}'$ is a B-Spline of degree $p-1$ and its knot vector and control points are computed as

$$\boldsymbol{\Xi}' = \Big[\underbrace{t_1, \dots, t_1}_{p}, t_{p+2}, \dots, t_{n_\mathbf{C}}, \underbrace{t_{n_\mathbf{C}+1}, \dots, t_{n_\mathbf{C}+1}}_{p}\Big] \quad \text{(indices referring to } \boldsymbol{\Xi}) \quad ,$$

$$\mathbf{C}'_i = \frac{p}{t_{i+p+1} - t_{i+1}}\big(\mathbf{C}_{i+1} - \mathbf{C}_i\big) \quad . \tag{2.19}$$

Piegl and Tiller [179] further show how Equation (2.19) extends to higher order derivatives. For tensor-product splines, Equation (2.19) has to be applied separately for every "layer", i.e., along a given parametric axis for every new control point in the control polygon. Similar considerations apply to rational Bézier splines and NURBS; however, the quotient rules must be applied then. Given a rational spline $\mathcal{C}_r = \bar{\mathcal{C}}/\mathcal{W}$, the quotient rule reads

$$\mathcal{C}'_r(\xi) = \frac{\bar{\mathcal{C}}'(\xi)\mathcal{W}(\xi) - \bar{\mathcal{C}}(\xi)\mathcal{W}'(\xi)}{\big(\mathcal{W}(\xi)\big)^2} \quad , \tag{2.20}$$

which requires the use of spline multiplication and hence results in a spline of degree[5] $p' = 2p$. Additionally, efficient algorithms are available for the computation of basis function

[5] Order elevation of the numerator is required to obtain a valid spline representation.

derivatives and for evaluating higher-order derivatives at specific parametric coordinates without the need to compute the closed-form representation beforehand. Interested readers are directed to references [178, 52, 189, 81] for a more detailed and comprehensive overview.

2.2.4. A Distinction Between Composition and Free-Form Deformation

Node-based shape optimization within the FEM framework - i.e., where each node in the mesh is considered a free design variable - requires constraints on mesh motion to prevent entanglement, i.e., inverted (irregular) elements with a negative Jacobian. In addition, optimizing the shape of the boundary may require expensive techniques to update the internal parametrization, such as the use of elastic deformations [27]. Therefore, Sederberg and Parry [202] introduced Free-Form Deformation (FFD) as a cost-effective method for mesh modification. FFD has since found various applications in shape optimization, such as in [215, 129].

The FFD technique involves defining a spline (mostly B-Spline or Bézier) around the bounding box of a mesh. The parametric domain usually aligns with its physical counterpart to facilitate the mapping. The spline in the surrounding area is modified by adjusting its control points. Subsequently, the nodes of the finite element mesh are updated to the new physical domain without altering the node-element connectivity. This technique has also been extended to the use of non-rational splines. Additional measures, such as mapping the Jacobian [149] (as implemented in the `splinepy` library), can be employed to ensure that the surrounding spline remains untangled.

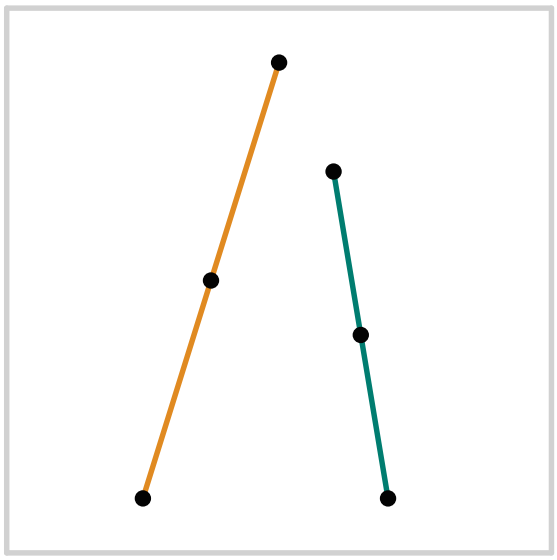
This approach can also be applied to spline geometries by mapping the control points. Contrary to functional spline composition presented in Equation (2.14), only the position of the control points is set into the spline representation of the outer geometry. For the "composition" of a line with degree p and control points $\mathbf{C}$ into a surface $\mathcal{S}$, this results in the mapping

$$\tilde{\mathcal{C}}_{\mathrm{FFD}}(\xi) = \sum_{i=1}^{n_{\mathbf{C}}} \mathcal{R}_i^p(\xi)\, \mathcal{S}(\mathbf{C}_{ij}) \quad . \tag{2.21}$$

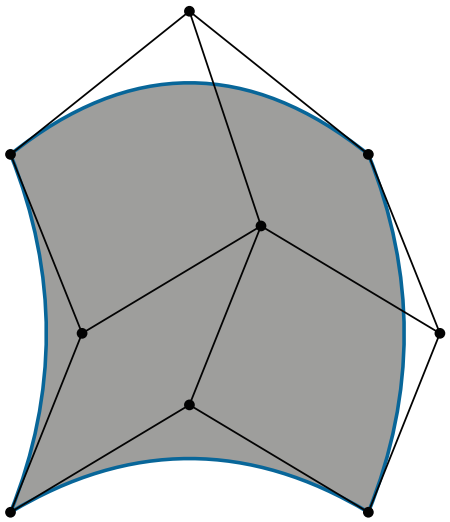
The method has already been utilized in shape optimization problems within the context of IsoGeometric Analysis (IGA) [190]. However, it is important to note that while both approaches originate from a composition, the continuity is generally not preserved when applying FFD, as shown in Figure 2.6. This is a key distinction from the functional composition approach, which preserves continuity as long as the geometry of the outer spline is maintained. However, the functional composition approach comes with a trade-off. The degree of the composed spline is significantly higher than the degree of the original spline (cf. Equation (2.17)).

2.3. Construction Process

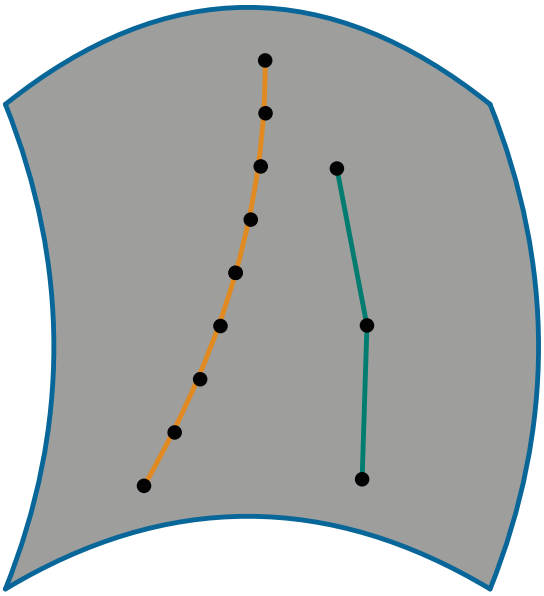
The methodology for creating CAD-compliant microstructures is based on the design paradigm through functional composition between splines outlined by Elber in [72]. In a general sense, the entire microstructure comprises two key components: (1) the micro



(a) Two lines, each consisting of linear B-Splines with an interior knot. Although the two line segments are G^1 in practice, the parametrization is only C^0. They are shown in the unit square $[0,1]^2$.



(b) A biquadratic Bézier Spline describing a given deformation. Its parametric domain is the unit square.



(c) The yellow line is deformed by means of functional composition with the Bézier spline, whereas the green spline is deformed by mapping only its control points corresponding to the classical Free-Form Deformation.

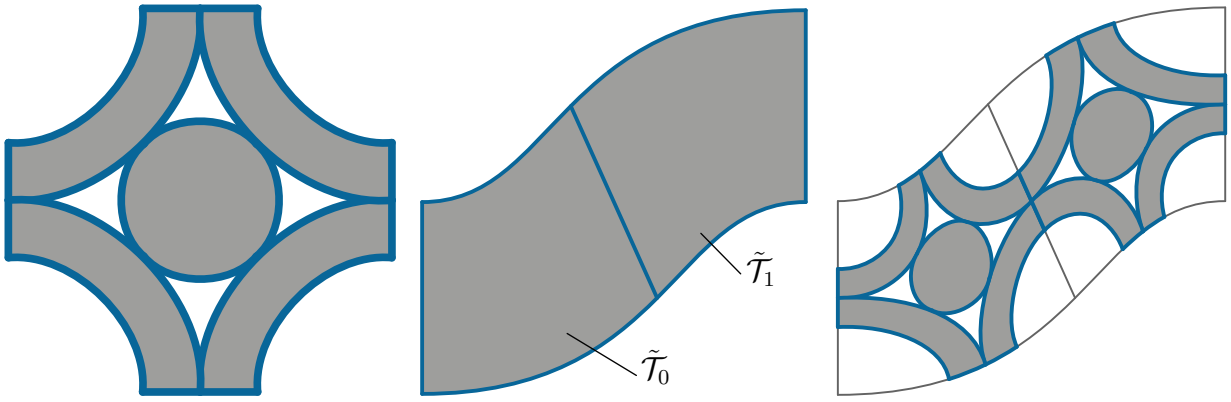
Figure 2.6.: Here, we demonstrate the distinction between the Free-Form Deformation method, which is commonly used in FEM-based shape optimization. Both lines consist of two colinear segments before deformation. The green line is deformed using FFD, which is applied to the control points of the spline representation, while the yellow line is deformed through functional composition with the new outer geometry. Although the latter *ensures continuity*, it significantly *increases the degree* (to degree 4, see Equation (2.17)).

representation, forming the basis which is periodically inserted to fill the geometry, and (2) a macro-representation, defining the overall structure's contour. In this context, we will refer to the (set of) patches defining the local geometry as the microtiles $M_{\boldsymbol{i}}$ and the macro geometry as the deformation function $\mathcal{T}$. Additionally, we define the number of tiles per parametric dimension of the deformation function as the grid dimensions $(n_{\xi_1}, \ldots, n_{\xi_{\tilde{d}}})$. We continue to presume that all tiles fit into the unit cube $[0,1]^d$.

The individual tiles can be some arbitrary combination of polynomial or rational splines. The original proposal for the microstructure construction paradigm also allowed for trimmed spline geometries and embedded geometries, e.g., surfaces or lines in 3D. In the context of shape optimization, however, we will only consider Volumetric Representation (V-Rep) geometries – that is, a representation of the body, rather than only its boundary – to facilitate the integration of IGA, see Chapter 3.2.

As described in the previous sections, compositions cannot generally be performed across knot lines. Consequently, the deformation function is first divided into its constituent C^∞ Bézier subsections $\tilde{\mathcal{T}}$, with $\cup_{\boldsymbol{i}} \tilde{\mathcal{T}}_{\boldsymbol{i}} = \mathcal{T}$. Knot insertion is performed prior to this step in order to attain the desired grid dimensions. The Bézier extraction further omits the necessity for a linear substitution to fit the microtiles into the individual knot spans, because their parametric domain spans the unit cube by definition. These individual Bézier patches are then subjected to functional composition with the splines that form the microtile. The resulting geometry can then be written as

$$\boldsymbol{\mathcal{M}} = \bigcup_{\boldsymbol{i}} \tilde{\mathcal{T}}_{\boldsymbol{i}}(M_{\boldsymbol{i}}) = \bigcup_{\boldsymbol{i}} \mathcal{M}_{\boldsymbol{i}} \quad . \tag{2.22}$$



(a) Microtile definition in unit cube M

(b) Knot span elements (Bézier patches) of the deformation function $\mathcal{T}$

(c) Complete microstructure $\boldsymbol{\mathcal{M}}$

Figure 2.7.: Construction of microstructures. The microtiles, defined in the unit cube, are placed into the parametric domain of the individual knot spans of the deformation function using functional composition.

The construction process is illustrated in Figure 2.7. As a result of the composition, the microtile cells are not only periodically repeated, but also deformed, resulting in the overall conformity of the structure within the macro representation.

Another advantage of this construction method is the controllable continuity of the resulting lattice structure. The continuity of the composition is governed by the lowest

continuity of either the inner and outer function spline [72]. This implies that continuity can be determined by analyzing patches on opposite faces of the unit cube prior to the composition. Further, if both the microtile and the deformation function are regular, i.e. $\det(\mathbf{J}) > 0$, the resulting structure will also be regular.

This construction process has since been extended in various ways. Massarwi et al. [158] introduced the utilization of bifurcation tiles, enabling a hierarchical microstructure that incorporates multi-scale components to some extent. Additionally, Hong and Elber [110] demonstrated the application of this composition-based method to trimmed trivariates with the aim of using this method with more intricate macro-geometries. The same authors also investigated the use of implicit functions to describe complex micro-geometries in [111]. The particular interest in implicit functions is driven by the use of Triply Periodic Minimal Surface (TPMS), which have been subject to noticeable research attention in recent years in the realm of microstructure design [155]. Further, in an effort to create shell structures, Dahiya, Shein, and Elber [57] used the composition based approach to tessellate an embedded surface representation in 3D.

The spline-based result of the microstructure construction process facilitates post-processing, as it is inherently in a CAD-compliant format. Additionally, the V-Rep geometry supports analysis through IGA naturally. Being based on the same spline representation, this paradigm allows seamless integration into established design processes. It also streamlines the back-propagation of geometry changes in the design pipeline's later stages, which might result from production constraints, such as those arising from Additive Manufacturing (AM). Consequently, this paradigm was proposed as the basis for numerical optimization in [14] and has since found application in [238], where it was utilized to optimize heat transfer in extrusion dies, cf. Section 4. In order to establish a sound mathematical formulation for the optimization problem, it is necessary to represent the geometry through a set of design variables. These variables should encompass both the local (micro) geometry, as well as the macroscopic geometry, i.e., the deformation function.

2.3.1. Parametrization of the deformation function

The deformation function captures both the outer shape of the microstructure and the arrangement of its interior knots determines the position of the cells within the macro shape. Consequently, the deformation function can be utilized to control both the external contour and the distribution of tiles within the structure.

In order to alter the deformation function, we consider two cases: modification of the internal parametrization and contour modification. In the application case discussed in [238], the microstructure surrounded a flow channel in an extrusion die, with the outer surface constrained by tool dimensions. Thus, the outer contour was assumed to be optimized or fixed, limiting the design space for the deformation function to the internal structure. Here, the objective is to adjust the internal parametrization without changing the outline of the macro geometry. One way to achieve this distribution modification is by altering the position of interior knots by means of knot insertion. However, the effectiveness of knot insertion is limited to the placement of additional knots. Hence, this method may prove insufficient if the deformation function already encompasses multiple knot spans, even though it might, in some cases, be possible to remove some of the original knots without affecting the precision too much [146]. Alternatively, an approximate representation of the outer geometry can be constructed, e.g., through spline fitting within production tol-

erances. Methods for spline fitting are manifold [153, 178] and can result in much simpler representations of the macro geometry. Moreover, [238] utilized the inner control points of the tensor product spline to modify the internal parametrization, as they do not contribute to the outer geometry. However, caution is advised as this can lead to irregular regions within the domain.

The main drawback associated with modifying the tile distribution by altering the deformation function originates from the inherent composition process of microstructures. As tiles are set into the parametric domain via functional composition, those situated in denser regions of the macro-spline simultaneously become thinner. This thinning can negatively impact the overall performance of a given structure, depending on the underlying physical problem. For example, in a heat transfer problem with isotropic material, the overall heat transfer is practically proportional to the volume density of a microstructure, which does not change by the composition, given a "smooth enough" deformation function. Ergo, changing the local distribution of tiles will have a negligible impact on the overall performance because the thinning of individual tiles counteracts the benefits of a denser tile distribution.

More design freedom is provided when the outer geometry of the macro geometry can be modified during optimization. This may be the case if (part of) the shape is not constrained by manufacturing or design limitations. In this case, all control points on the outside of the control mesh can be considered as design variables. However, this approach introduces new challenges since the regularity of the deformation function depends on the control points. This means that too much modification of the control mesh can result in irregular regions with negative Jacobian determinants, leading to a tangled microstructure with self-intersections.

This risk of irregularity affects not only the surface, but also the internal parametrization. In fact, creating the smoothest possible internal parametrization based on a boundary representation is a common problem in the context of IGA and shape optimization. Therefore, much research effort is directed at modifying internal control points to produce smooth internal parametrizations, either by computing the internal control points as a function of the external control points [94] or by introducing additional constraints to the optimization problem [140].

2.3.2. Parametrization of the microtile

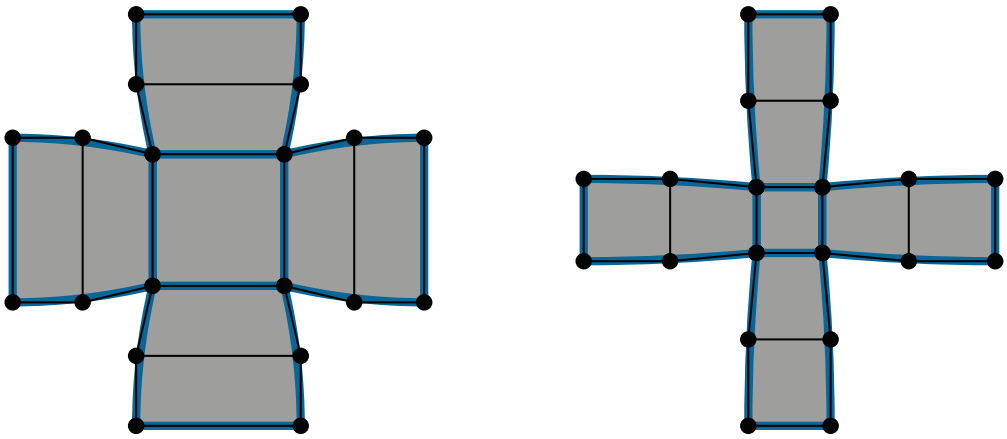
As stated in the previous section, a denser distribution ultimately leads to thinning of individual tiles after their composition - two effects that may cancel each other out in terms of structural performance. To counteract these effects, the microtiles can be locally modified prior to their insertion into the macro-geometry. Here the possibilities are manifold and are not limited to geometric aspects, but can also be extended to material properties.

A primitive approach consists of designing multiple microtiles and setting them into their respective regions within the macro-geometry. However, this method is generally not a well-suited approach to numerical shape optimization because (1) the initial placement of the microtiles requires a lot of insight into the problem and design intuition, and (2) this method leads to a combinatorial problem that becomes exponentially more complex with the number of tiles in the microstructure.

Regarding elastic structures, the most intuitive way to modify the local properties of a microstructure is through manipulation of the material properties. For instance, selecting a

material with a higher Young's modulus in a region that requires greater stiffness. However, while this idea translates well into simulations, it is very difficult production-wise. To offer an array of material properties, it is necessary to either provide a multitude of different materials [193] or create meta-materials during production [18]. Although approaches have been developed to incorporate functionally graded materials in the design stages – using trivariate spline representations with an additional material dimension and even producing real prototype demonstrators [80] – AM production of multi-material designs still poses practical difficulties. Therefore, we will restrict the conversation to pure shape optimization in the context of this thesis, altering solely the microtile's geometry and leaving material properties untouched.

The options for altering the microtile shape are multifaceted and depend on the objective of the optimization at hand. In the application examples discussed in later Chapters 4 and 5, this can be accomplished by locally modifying the volume density, which is akin to changing the volume density of a tile prior to its insertion. For example, for a tile consisting of a cross, this can be achieved by modifying the thickness of its branches, which, in the context of heat transfer, can correlate with higher heat fluxes in a given area. An example of such a structure is depicted in Figure 2.8.



(a) Cross-shaped tile with branch width 0.4 in the unit square

(b) Cross-shaped tile with branch width 0.2 in the unit square

Figure 2.8.: Example of a parametrized microtile. A cross with individually modified branch thicknesses. In this example, all four branches have individual widths. Here, the parametrization only acts on the position of the control points in the unit square.

Similar approaches also apply for structural problems. Here, a higher volume density generally correlates with a higher stiffness. More recently, methods have further evolved to align the microscopic structure with the stresses in order to distribute forces more efficiently throughout the structure [4]. Based on this idea, it is also possible to create and alter tiles to achieve comparable outcomes, e.g., by using rotations. However, because tiles are created locally in a unit cube, maintaining continuity between adjacent parametrized tiles can be challenging.

By using this method for microtile parametrization, each microtile is defined as a function of design variables that are associated with the tile's specific characteristics. In the example tile in Figure 2.8, each branch in the cross cell possesses a specific thickness associated with

the corresponding design variable. The microtile resulting from this process, prior to its insertion into the deformation function, can therefore be expressed as follows

$$M_{\boldsymbol{i}} = \tilde{M}(\tilde{P}^M_{\boldsymbol{i}}) \quad , \tag{2.23}$$

where $\tilde{M}$ indicates the parametrized base-cell as a function of the microtile's design variable(s) $\tilde{P}^M$ at position $\boldsymbol{i} = \{i, j, \dots\}$ (indices in the tile-grid). Note, that the parametrization in this case only acts on the control points of the microtile definition, leaving all other spline properties untouched. As a consequence, the individual patches, composing the microtile $\tilde{\mathcal{S}} \in M_{\boldsymbol{i}}$, can be written as

$$\tilde{\mathcal{S}}(\boldsymbol{\xi})|_{\tilde{P}^M_{\boldsymbol{i}}} = \sum_{\boldsymbol{i}} \mathcal{R}^{\mathcal{S}}_{\boldsymbol{i}}(\boldsymbol{\xi})\, \mathbf{C}_{\boldsymbol{i}}(\tilde{P}^M_{\boldsymbol{i}}) \quad . \tag{2.24}$$

2.3.3. Parameter Abstraction

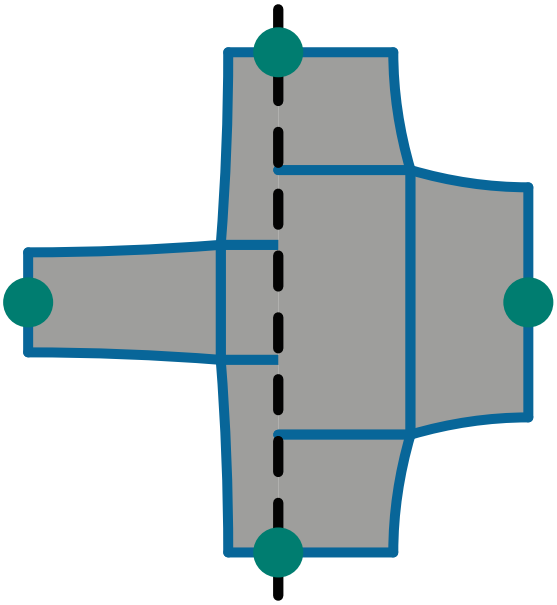
As highlighted in the previous section, various parameters can be associated to each individual microtile to fully exploit localized parametrization. Consequently, given the vast number of microtiles involved in certain applications, such an optimization can quickly exceed 100 000 design variables. In practice, such a large number of optimization parameters is not practical, especially if the optimization is performed by black-box drivers without access to gradient information. Therefore, this motivates rearranging the parameter set utilizing a small subset of superordinate parameters $\mathbf{P}^M$, in order to decrease the number of degrees of freedom in the optimization.

This abstraction has the additional purpose of assigning similar values to microtiles in close proximity, resulting in smooth transitions between tile characteristics. It can also make it easier to maintain continuity between adjacent tiles, e.g. by ensuring that adjacent tiles have the same thickness at their connecting interface.

One intuitive approach to accomplish this is by defining a function in the parametric domain of the deformation function. This function needs to meet certain criteria:

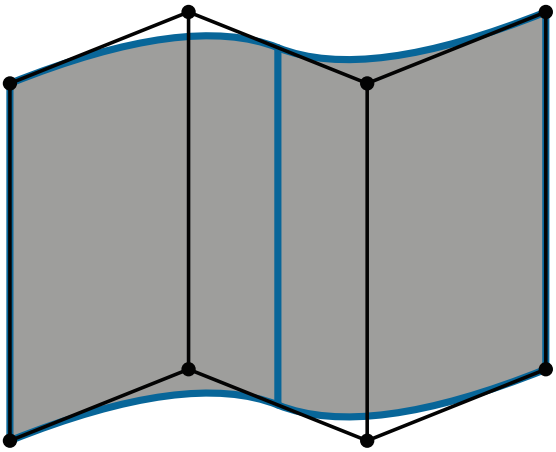
- continuity within the parametric domain of the deformation function
- intuitive representation and local control
- extensibility

One natural choice for these functions is splines, since they can be chosen to be arbitrarily continuous and allow for extensibility through knot refinement. They also offer a clear idea of the resulting structure due to their interpolatory properties with non-negative basis functions. The spline coefficients or control points, which can be interpreted as the superordinate design variables, provide an easy design intuition when projected onto the deformation function. This method, as proposed in [238], building upon approaches presented in [14], denotes the resulting spline as the parameter spline $\mathcal{P}$. The parameter spline and deformation function do not necessarily require sharing the same representation. They must, however, be defined over the same parametric domain and the image of the parameter spline must be within the admissible range for the local design parameters in $\tilde{M}$. It may be necessary to fulfill additional constraints, for example, if the deformation has two connected boundaries, the parameter spline has to fulfill the same restrictions.

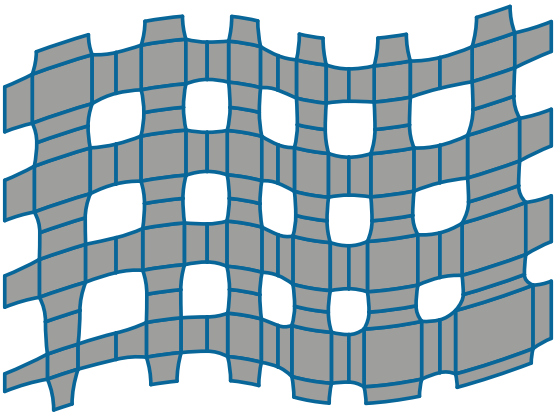


(a) Parametrized cross tile and evaluation points $\mathbf{x}^M$ (green), shown with branch thicknesses 0.1 (left) and 0.23 (right).

(b) Parameter spline and its coefficients mapped onto a square with values ranging from 0.05 (lower left) to 0.35 (lower right).



(c) Deformation function; Linear-Quadratic B-Spline one interior knot



(d) Resulting microstructure with tiling 3×4 in every element

Figure 2.9.: Microstructure synthesis in a nutshell. The microtile in Subfigure (a) is defined within the unit-square along with evaluation points (green), that are used to retrieve local parameters from the parameter spline displayed in Subfigure (b), based on its position within the deformation function (Subfigure (c)). The resulting microstructure in Subfigure (d) then reflects this abstraction in the "thickness"-distribution.

Using these parameter splines as abstractions, the microstructure synthesis can be described as follows. First, the tile parametrization is established, along with a set of evaluation points $\mathbf{x}^M$. The position of these points within the deformation function is then used to assess the parameter spline and retrieve the corresponding local spline parameters. If evaluation points fall on the boundary (e.g., at position [0.5, 1] for a 2D microtile), adjacent tiles will receive the same value as long as the parameter spline is at least C^0 continuous. Afterwards, the microtile is constructed based upon the updated parameters

and inserted into the deformation function via functional composition. This process is illustrated in Figure 2.9.

To conclude, the microstructure construction process is illustrated in Algorithm 2.1. The number of coefficients in the parameter spline corresponds to the number of design variables associated with the internal parametrization of the optimization procedure. This significantly reduces the complexity and helps to accelerate the convergence to the optimal solution. In this thesis, the $n_{\boldsymbol{\chi}}$ design variables $\boldsymbol{\chi}$ considered for the optimization problem therefore comprise (a subset of) the control points $\mathbf{C}_{\mathcal{T}}$ of the deformation function to modify the macro-shape and the coefficients of the parameter spline $\mathbf{P}^M$, which yields

$$\boldsymbol{\chi} = \{\mathbf{C}^{\mathcal{T}}, \mathbf{P}^M\} \quad . \tag{2.25}$$

2.4. Geometric Derivatives

As outlined in the preceding chapters, the resulting microstructure is fully described by the deformation function, the parameter spline and the parametrization of the microtile. Based on this input, the microstructure is built explicitly, facilitating the analytical differentiation of the geometry with respect to the design variables. This attribute is an important advantage in terms of the optimization strategy, which will be presented in Chapter 3.4.

We will first examine the derivative of the composed microtile in relation to the deformation function's control points, which is required for optimization of the macro-shape. Recalling the definition of the microstructure construction from Equation (2.22) the derivative of the microstructure can be written as

$$\begin{aligned}\frac{\partial \boldsymbol{\mathcal{M}}}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} &= \bigcup_k \frac{\partial \mathcal{M}_k}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} = \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k(M_k)}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} \\ &= \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} \circ M_k \quad .\end{aligned} \tag{2.26}$$

The first term represents the derivative of the Bézier patch $\tilde{\mathcal{T}}$ with respect to the control points of the deformation function $\mathcal{T}$. However, these control points pertain to the original deformation function before knot insertion and Bézier extraction. The chain rule simplifies since the microtile parametrization is independent of the deformation function's control points. As a result, the first term in Equation (2.26) can be split up into the derivatives (1) with respect to the patch control points, and (2) the sensitivities of the deformation function's control points

$$\begin{aligned}\frac{\partial \tilde{\mathcal{T}}_k}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} &= \frac{\partial \tilde{\mathcal{T}}_k}{\partial \mathbf{C}_{\boldsymbol{j}}^{\tilde{\mathcal{T}}_k}} \frac{\partial \mathbf{C}_{\boldsymbol{j}}^{\tilde{\mathcal{T}}_k}}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} \\ &= \mathcal{R}_{\boldsymbol{j}}^{\tilde{\mathcal{T}}_k} \, \mathbb{A}_{\boldsymbol{\Xi}}^{\boldsymbol{j} \to \boldsymbol{i}} \quad ,\end{aligned} \tag{2.27}$$

which leads to

$$\frac{\partial \boldsymbol{\mathcal{M}}}{\partial \mathbf{C}_{\boldsymbol{i}}^{\mathcal{T}}} = \bigcup_k \mathcal{R}_{\boldsymbol{j}}^{\tilde{\mathcal{T}}_k}(M_k) \, \mathbb{A}_{\boldsymbol{\Xi}}^{\boldsymbol{j} \to \boldsymbol{i}} \quad . \tag{2.28}$$

Algorithm 2.1: Microstructure synthesis during optimization adapted from Antolin et al. [14], extending it by parametrization of the deformation function as well as local parameter retrieval

Input:

$\tilde{M}$	parametrized microtile in parametric space as a function of $\tilde{P}^M$
$\mathcal{T}_0$	Initial geometry of the deformation function
$(n_{\xi_1}, \ldots, n_{\xi_{\tilde{d}}})$	Grid dimensions of the microstructure
$\mathcal{P}$	Spline representation of the parameter space that is projected into the deformation function

Output:

$\boldsymbol{\mathcal{M}}$	complete microstructure
$\boldsymbol{\chi} : \{\mathbf{C}^{\mathcal{T}}, \mathbf{P}^M\}$	optimized set of parameters

Algorithm:

$\boldsymbol{\mathcal{M}} \leftarrow \emptyset$;

Parametrization Deformation function

$\mathcal{T} \leftarrow \mathbf{C}^{\mathcal{T}}$; `// Update coefficients`

Parameter spline update

$\mathcal{P} \leftarrow \mathbf{P}^M$; `// Update coefficients`

for $\boldsymbol{i}$ **in** $(n_{\xi_1}, \ldots, n_{\xi_{\tilde{d}}})$ **do**

  Parametrization Microtile

  $\tilde{P}^M_{\boldsymbol{i}} \leftarrow \mathcal{P}(\xi_{\boldsymbol{i}})$; `// Retrieve microtile parameters based on position of the microtile`

  $M_{\boldsymbol{i}} \leftarrow \tilde{M}(\tilde{P}^M_{\boldsymbol{i}})$; `// Construct Microtile in Parameter space`

  Construct Microstructure

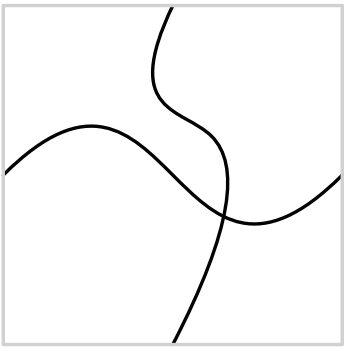
  $\tilde{\mathcal{T}}_{\boldsymbol{i}} \leftarrow \mathcal{T}$; `// Extract corresponding Bézier patch from Deformation Function`

  $\mathcal{M}_{\boldsymbol{i}} \leftarrow \tilde{\mathcal{T}}_{\boldsymbol{i}}(M_{\boldsymbol{i}})$; `// Perform composition`

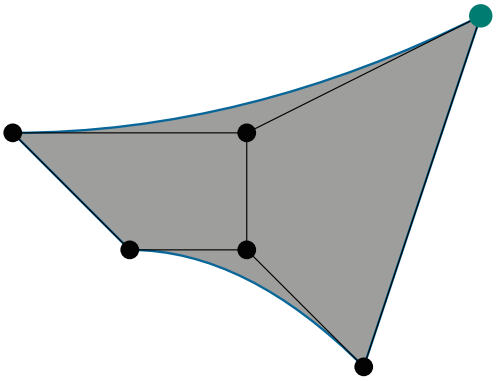
  $\boldsymbol{\mathcal{M}} \leftarrow \boldsymbol{\mathcal{M}} \cup \mathcal{M}_{\boldsymbol{i}}$; `// Add new tile to Microstructure`

end

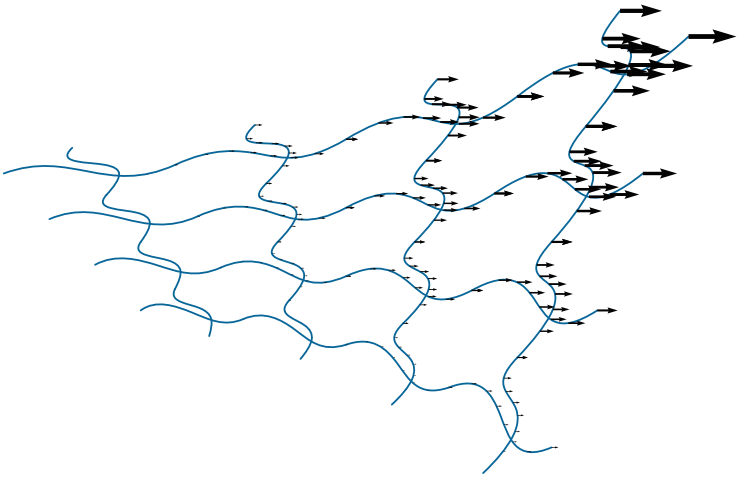
The derivative with respect to a control point yields a matrix-valued spline, which can be difficult to handle in practice. Therefore, in most cases, the individual vector contributions are derived separately.



(a) Microtile comprised of 2 Bézier curves of degree 3 and 4.



(b) Linear-quadratic deformation function. The resulting microstructure is derived with respect to the x-component of the green control point in the upper right corner.



(c) Resulting microstructure and control point derivative plotted as a field.

Figure 2.10.: Microstructure and derivative (field) with respect to the x-component of the 6^{th} control point, marked green. The field is computed by functional composition of the microtiles with the basis functions of the deformation function.

It is also of interest to consider Equation (2.28) in regards of computational efficiency. The order of evaluation has a significant impact on the number of required computations. The individual terms are independent of each other, allowing for significant reduction in the number of multiplications through their pre-combination and computation. Furthermore, depending on the implementation, the computation of the basis function compositions can be an intermediate step of the spline composition itself, resulting in very little overhead. The knot insertion matrices for Bézier extraction are independent of the position of the control points of the deformation function, hence they can be computed in pre-processing, too. Moreover, the underlying spline representation, i.e., the type, degree and knot vector,

of the basis function composition is equivalent to the spline composition itself. This is because the subsequent addition of basis function contributions does not change the order of the splines. This correspondence creates a one-to-one relationship between the control points of the microstructure and the (vector-valued) close-form derivative's control points.

Similar considerations can be made for the derivation of the remaining design variables, the coefficients of the parameter spline $\mathbf{P}^M$. To accomplish this, Equation (2.22) will be revisited, but taking into account the abstraction of the design variables using the parameter splines. This can be expressed as follows

$$\frac{\partial \boldsymbol{\mathcal{M}}}{\partial \mathbf{P}_{\boldsymbol{i}}^M} = \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k\big(M_k(\tilde{P}_k^M)\big)}{\partial \mathbf{P}_{\boldsymbol{i}}^M} = \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k\Big(M_k\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)\Big)}{\partial \mathbf{P}_{\boldsymbol{i}}^M} \quad , \tag{2.29}$$

where $\bar{\mathbf{x}}_k^M$ denote the microtile's evaluation points mapped into the parametric domain of the deformation function. The chain rule must be applied successively to calculate the analytical derivatives of the microstructure, leading to

$$\begin{aligned} \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k\Big(M_k\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)\Big)}{\partial \mathbf{P}_{\boldsymbol{i}}^M} &= \bigcup_k \frac{\partial \tilde{\mathcal{T}}_k}{\partial M_k^o}\Big(M_k\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)\Big) \frac{\partial M_k^o\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)}{\partial \mathbf{P}_{\boldsymbol{i}}^M} \\ &= \bigcup_k \underbrace{\frac{\partial \tilde{\mathcal{T}}_k}{\partial M_k^o}\Big(M_k\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)\Big)}_{\tilde{\mathcal{T}}_{k,\xi^o} \circ M_k} \frac{\partial M_k^o\big(\mathcal{P}(\bar{\mathbf{x}}_k^M)\big)}{\partial \tilde{P}^M} \underbrace{\frac{\partial \mathcal{P}(\bar{\mathbf{x}}_k^M)}{\partial \mathbf{P}_{\boldsymbol{i}}^M}}_{\mathcal{R}^{\mathcal{P}}_{\mathbf{P}_{\boldsymbol{i}}^M}(\bar{\mathbf{x}}_k^M)} \quad . \end{aligned} \tag{2.30}$$

Upper indices correspond to the parametric dimension of the deformation function and, thus, to the physical dimension of the microtile. The first term of the equation $\partial \tilde{\mathcal{T}}_k / \partial M_k$ corresponds to the closed-form derivative of the deformation function patches with respect to its parametric coordinates as described in Equation (2.18) and Equation (2.19). The last term $\partial \mathcal{P} / \partial \mathbf{P}^M$ can be rewritten as the parameter spline's basis functions. The derivative with respect to the local design variables $\tilde{P}^M$ must either be analytically derived for a given tile or determined using numerical methods, like Algorithmic Differentiation (AD) [162, 96] or Finite Differences (FD).

It was established in Chapter 2.2.3, that the degree of the composed spline is a function of the outer function's degrees and the inner function's degrees, cf. Equation (2.17). In addition, the degree of a polynomial spline is reduced in the derivative along the parametric dimension in which it is differentiated. Since the microtiles are parametrized solely in terms of their control points, the degree spline representation of the microtile derivative with respect to the microtile parameters $\tilde{P}^M$ remains unaffected. If we now consider Equation (2.30) for a certain parametric dimension o, the degree of the deformation function $\mathbf{p}$ in this parametric direction is first reduced by 1, so that the sum of the degrees is also reduced by 1. On the other hand, the degree of all parametric directions of the composed spline $\mathbf{p}^{\mathcal{M}}$ is increased by the respective degrees of the derived microtile $\tilde{\mathbf{p}}$ through its multiplication, resulting in the degree of the derivative

$$p_i^{\mathcal{M}} = \tilde{p}_i \left[\sum_{j, j \neq o} p_j + (p_o - 1) \right] + \tilde{p}_i = \tilde{p}_i \left[\left(\sum_j p_j \right) - 1 \right] + \tilde{p}_i = \tilde{p}_i \sum_j p_j \quad , \tag{2.31}$$

which is equivalent to the microstructured geometries inner parametrization.

If the control point of the deformation function remains unaltered during the optimization process, the closed-form derivatives can be precomputed once. However, this is not possible if the deformation function is optimized at the same time. In addition, the multiplication with the microtile derivative must be performed once for each microtile parameter required to define the structure. Therefore, the computational cost of determining the derivative of the microstructure with respect to the parameter spline coefficient is significantly higher than the previously introduced derivative with respect to the control points of the deformation function. Therefore, in order to minimize the required computations, it is crucial to take advantage of the local support of the parametrization function and determine the derivative only in the regions where the basis functions are non-zero.

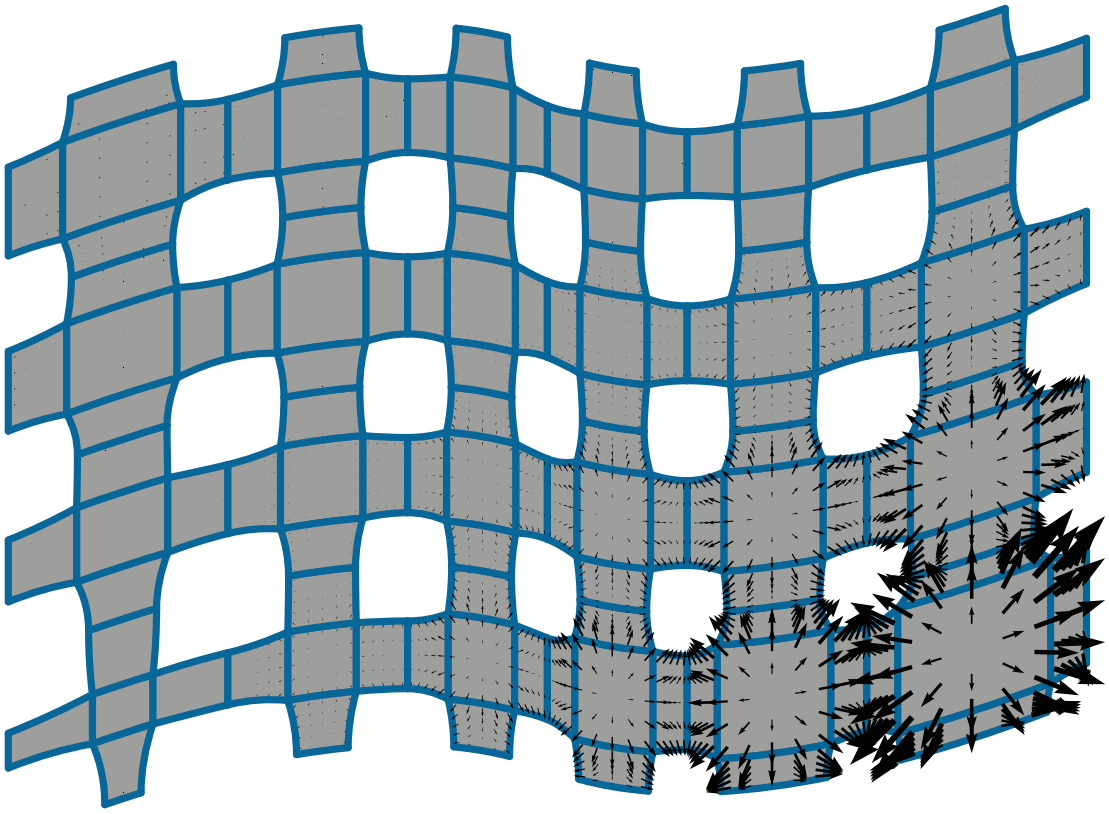


Figure 2.11.: Derivative of the parametrized microstructure displayed in Figure 2.9 with respect to the third coefficient of the parameter spline, here displayed in the lower right corner. The field vectors are scaled for better visibility.

Recalling the microstructure construction process shown in Figure 2.9, the resulting structure is now derived with respect to the coefficient in the lower right corner of the parameter spline shown in Figure 2.9b. The derivative field is mapped onto the structure in Figure 2.11. The figure shows that the derivative is non-zero only within the support of the basis function associated to the coefficient (here in the lower right corner of the deformation function). This property is particularly useful in the context of shape optimization, as it allows for localized adaptation of the microstructure.

3 Numerical Analysis

This chapter provides a concise overview of the theoretical foundations of computational modeling. The initial section presents the required differential equations by formulating the integral conservation laws. In our optimization examples, we will consider two distinct cases: (1) a heat transfer problem and (2) an elasticity problem. This section will also introduce the underlying assumptions and constitutive equations to provide the closed Partial Differential Equation (PDE) which will be discretized in later parts of this chapter.

Following this, we examine IsoGeometric Analysis (IGA), comparing it to the traditional Finite Element Method (FEM) and highlighting its applications in shape optimization. In order to provide a better perspective, we break down the steps of IGA, touching on its strengths and challenges. This section presents the strong and variational forms, followed by an illustration of the discretization process. Then, the numerical concepts and implementation details are addressed.

Moving on to shape optimization, this section outlines and classifies various design optimization methods and introduces the key concept of optimization drivers, setting the stage for the later chapters. Finally, we demonstrate how to efficiently compute sensitivities using the adjoint approach.

3.1. Physical Background

For our derivations of the required PDEs, we will consider an open and connected set in $\mathbb{R}^d$ that defines our *physical* domain Ω. We will further denote its boundary $\partial\Omega := \bar{\Omega} \setminus \Omega$. The domain and its boundary are illustrated in Figure 3.1, along with some exemplary boundary subdomains, which play a role in the the definition of the necessary BCs.

3.1.1. Heat Transfer

As a first example, we will look at a heat transfer problem in a solid material, i.e. without convection. The following considerations are based on the derivations in [144]. To construct the necessary equations, we will first consider an arbitrary small body B, fully contained within the domain Ω, with $B \subseteq \Omega$. Given the specific (heat) energy e, a directional heat

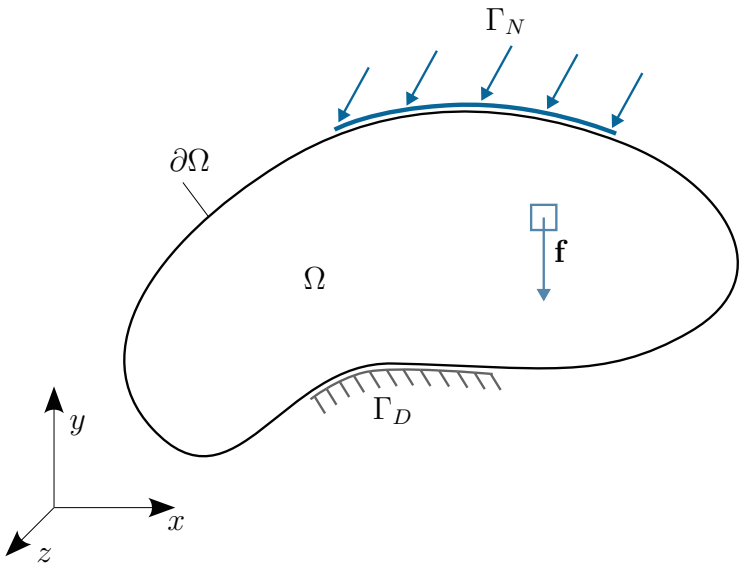


Figure 3.1.: Representative body with Neumann and Dirichlet conditions. All boundaries that are not further specified will be considered with zero-Neumann Boundary Conditions.

flux $\mathbf{q}$ on the boundary of the body ∂B with its outward facing normal $\mathbf{n}$, and local heat sources f, the energy balance equation can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_B e\,\mathrm{d}V = -\int_{\partial B}\mathbf{q}\cdot\mathbf{n}\,\mathrm{d}S + \int_B f\,\mathrm{d}V \quad , \tag{3.1}$$

which states that the change in thermal (internal) energy within the body must be equal to the heat transferred across its boundary ∂B and the heat introduced from additional sources f. The integral is negative because the transferred energy is considered added to the system if the heat flux is pointing into the body.

Assuming that heat flux $\mathbf{q}$ is a continuously differentiable field within domain B and continuous on ∂B, we can apply the divergence theorem to the vector field

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_B e\,\mathrm{d}V = -\int_B \nabla\cdot\mathbf{q}\,\mathrm{d}V + \int_B f\,\mathrm{d}V \quad . \tag{3.2}$$

Given that body B is independent of time, we can pull the time derivative into the integral and rearrange the terms onto one side. Further, the integral must hold on every subset of the computational domain Ω, giving rise to the PDE

$$\frac{\mathrm{d}e}{\mathrm{d}t} + \nabla\cdot\mathbf{q} - f = 0 \quad . \tag{3.3}$$

To obtain a closed system of equations, we need to introduce constitutive relationships between the internal energy e and the heat flux $\mathbf{q}$ in terms of the temperature field θ. Considering a rigid body without motion, the internal energy can be written as [152]

$$e = \rho c\theta \quad , \tag{3.4}$$

where ρ is the local material density, c denotes the specific heat capacity and θ the temperature. Furthermore, we introduce Fourier's law of heat conduction for isotropic materials, which states

$$\mathbf{q} = -k\nabla\theta \quad . \tag{3.5}$$

Assuming that the material constants are time-independent, the full PDE can be rewritten as

$$\rho c \frac{\mathrm{d}\theta}{\mathrm{d}t} - \nabla \cdot (k \nabla \theta) - f = 0 \quad . \tag{3.6}$$

If the material parameters are considered constant, the parameters are often rearranged in terms of their thermal diffusivity $\lambda = {}^{k}/_{\rho c}$. Then Equation (3.6) reads

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} - \lambda \Delta \theta - \frac{f}{\rho c} = 0 \quad . \tag{3.7}$$

This system can be simplified in the stationary case without external heat sources ($f = 0$), when the temperature field satisfies Laplace's equation

$$\Delta \theta = 0 \quad . \tag{3.8}$$

Heat transfer problems (or other types of diffusion dominated systems) are not only frequently considered for demonstration purposes in simulations due to their simplicity, but also play an important role in many fields of engineering, with applications ranging from heat transfer in electrical systems in fatigue analysis under thermo-mechanical loading [108], to heat dissipation in braking systems [233], to extrusion die manufacturing [238] (cf. Chapter 4).

3.1.2. Continuum Mechanics - Linear Elasticity

Continuum mechanics is the study of the mechanical behavior under different loading conditions of a continuous, compact set of material points. It describes the macroscopic behavior of a body and is used in almost all areas of modern engineering. While a full discussion of continuum mechanics is beyond the scope of this work, there is extensive literature on this topic and the interested reader is referred to, for example, [12, 36, 109, 132]. In the context of this work, we limit ourselves to the consideration of linear elastic, isotropic material behavior, which is a reversible deformation that does not depend on the load history or the orientation of the material. Furthermore, only stationary problems are addressed.

The derivation of the corresponding equations follows a similar principle as in the previous model. For this purpose, an infinitesimal element is virtually cut free from a solid to form the balance of forces. This approach yields the conservation equation (cf., e.g., [49])

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = \mathbf{0} \quad , \tag{3.9}$$

with the Cauchy stress tensor $\boldsymbol{\sigma}$ and a volumetric force field $\mathbf{f}$.

Equation (3.9) describes a general principle and is not limited to a specific material. However, this system of equations is not complete and requires additional constitutive equations that describe the idealized material behavior and determine the relationship between strains (as a function of the displacement field), describing the deformation of a body and the resulting stresses.

As strain measure, we consider the linearized Green-Lagrange strain [12], or infinitesimal strain tensor, which reads

$$\boldsymbol{\epsilon} = \frac{1}{2} \left[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T \right] \quad , \tag{3.10}$$

as a function of the displacement field $\mathbf{u}$.

Finally, we need to establish a relationship between stress and strain to close the system of equations. To do this, we formulate the constitutive equation for an ideal material, the isotropic, linearly elastic solid, according to Hook's law. Linear elasticity is often used as a first estimate of the mechanical behavior of new designs. In fact, for many applications, engineers try to stay within the linear regime of the material used to ensure that the mechanism does not exhibit permanent deformation or even fracture. As the name suggests, this material idealization assumes a linear relationship between the strain measure, i.e., the deformation, and the applied load, and a fully reversible deformation when the load is removed. This idealization can provide a good estimate of material behavior when deformations remain very small. It reads

$$\boldsymbol{\sigma} = \lambda \mathrm{tr}\,(\boldsymbol{\epsilon})\,\mathbf{I} + 2\mu\boldsymbol{\epsilon} \quad , \tag{3.11}$$

where λ and μ are known as Lamé's coefficients, which are to be determined from appropriate experiments. These constants can also be expressed in terms of the more commonly used parameters Young's modulus E, which is a proportionality factor between the tensile/compressive loading and the corresponding tensile/compressive stress, and Poisson's ratio ν, which characterizes the transverse contraction behavior using[1]

$$\mu = \frac{\mathrm{E}}{2\,(1+\nu)} \qquad \lambda = \frac{\mathrm{E}\nu}{(1-2\nu)(1+\nu)} \quad . \tag{3.12}$$

3.1.3. Boundary Conditions

Both PDEs described in Equations (3.3) and (3.9) yield a parabolic boundary value problem given the presented constitutive equations and require appropriate Boundary Conditions (BCs) in order to form a well-posed problem.

The BCs considered in this thesis are twofold. First, Dirichlet-type conditions are applied where the fields have an explicitly defined value of the unknown quantity. Dirichlet-type boundaries are applied to a (part of a) spatial boundary $\Gamma_D \subset \partial\Omega$. In the case of heat transfer, this is reflected in a specific temperature T_Γ on the surface,

$$\theta = T_\Gamma \quad \text{on } \Gamma_D \quad , \tag{3.13}$$

while in the case of linear elasticity, a specified displacement $\mathbf{g}$ is forced

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma_D \quad . \tag{3.14}$$

Dirichlet BCs are typically used to describe surfaces attached to other components, such as heating elements, mechanical mounts, housings, or external mechanisms, and therefore must have the same value as their counterpart.

Neumann, or second-type BCs are applied to the external boundary $\Gamma_N \subset \partial\Omega$, with $\Gamma_N \cup \Gamma_D = \partial\Omega$ and $\Gamma_N \cap \Gamma_D = \emptyset$. Contrary to Dirichlet BCs, which impose values on the solution itself, these act on the (first-order) derivatives of the fields described by the

[1] The presented simplified equations originate from the 3D linear case and their validity heavily depends on underlying hypothesis about the state of strain and stress in the 2D case. Here, we indirectly assume that all strain components along the third axis are zero, which in practice is referred to as the plane strain assumption.

system of PDEs. In the context of heat transfer, this involves prescribing the net heat flux density q_Γ over the boundary, c.f. Equation (3.5))

$$\mathbf{q} = -k\nabla\theta \cdot \mathbf{n} = q_\Gamma \quad \text{on } \Gamma_N \quad , \tag{3.15}$$

where $\mathbf{n}$ represents the unit normal vector. If the prescribed flux over the boundary q_Γ is negative, heat is added to the system. In the case of linear elasticity, Neumann BCs are expressed as a prescribed surface traction $\mathbf{h}$, i.e., external pressures, loads or forces, which are imposed using

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} \quad \text{on } \Gamma_N \quad . \tag{3.16}$$

If not otherwise specified, zero-Neumann conditions are assumed, i.e., $q_\Gamma = 0$ and $\mathbf{h} = \mathbf{0}$. For heat transfer, this implies an adiabatic surface, while for elasticity problems it implies a traction-free boundary.

3.2. Isogeometric Analysis

3.2.1. An Overview

PDEs like the ones introduced in Chapter 3.1 can be solved numerically using various techniques. Among the most prevalent are the Finite Difference Method (FDM) (e.g., [133]), the Finite Volume Method (FVM) (e.g., [58]) and the Finite Element Method (FEM) (e.g., [67, 117, 236]). Since its introduction in the aerospace industry in the 1950s, FEM has become the de facto industry standard in structural mechanics and many other fields of engineering.

Most modern mechanical engineering applications use Computer Aided Design long before physical prototypes to reduce production costs. The standard workflow links the geometric design stage with CAD to Computer Aided Analysis (CAA) through meshing, a costly and time-consuming process. In this step, the original geometry is discretized and (in most cases) defeatured using a computational mesh, while geometric complexity is reduced to a minimum in order to make the analysis manageable. Even without defeaturing, this step introduces geometric errors into the system, as in most cases rounded surfaces in the original geometry are approximated using piecewise polynomial cells. Furthermore, the necessary preparatory steps and meshing procedures collectively account for a significant proportion of the total analysis time [53]. Moreover, if the given mesh accuracy proves insufficient, mesh refinement requires knowledge of the original geometry to capture geometric details of the boundary and the discretization has to be repeated.

Driven by considerations of cost efficiency within the design workflow, Hughes, Cottrell, and Bazilevs [118] introduced the paradigm of IsoGeometric Analysis in 2005 with the goal of seamlessly linking the analysis stage with geometric design. This approach incorporates splines, the mathematical geometry description traditionally used in CAD (cf. Chapter 2), as the basis for solving PDEs, therefore eliminating the meshing stage. The accuracy provided from CAD geometries is generally insufficient for analysis, where, e.g., displacement fields or stresses have to be computed on a very fine scale compared to their geometric features. However, refinement strategies such as h-, p-, and k-refinement [54] can be introduced using the methods presented in Chapter 2.2.3, which increase the field resolution

without altering the geometry. It is important to note that while h- and p-refinement have counterparts in FEM, k-refinement has a distinctive status in IGA. This method allows to simultaneously reduce the element size and increase the polynomial order without compromising a given level of continuity.

The high continuity in between elements, as well as the direct utilization of the CAD-model confer multiple benefits to the analysis, with particular emphasis on the following aspects:

Integration of Optimization and Design A fundamental aspect of this thesis is the inherent compatibility between IGA and shape optimization. Using IGA for analysis within the shape optimization loop, the control points of the discretized geometry directly serve as the design variables [223]. After successful optimization, these control points can be easily reinserted into the original design because they contain the same geometry representation. This eliminates the need for additional post-processing steps that are otherwise required in classical FEM when using a node-based optimization strategy. In the latter case, the original design must be adjusted based on the modified mesh, which can be a costly, error-prone process. As a result, IGA has found many applications in shape optimization, such as fluid dynamics for minimizing drag or pressure loss [173, 177] shell structures [102, 104], and also truss-based lattice structures [228, 227].

Geometrically exact As mentioned above, eliminating the mesh not only reduces the number of steps required for analysis, but also limits the geometric error that would otherwise be introduced by piecewise polynomial approximations using standard FEM. One of the major advantages of the specific construction methods for microstructured geometries presented in Section 2.3 is their exact conformity to the macro shape. It is therefore natural to maintain this exactness in the analysis step. Furthermore, the approximation can significantly affect the simulation result, especially when the analysis is particularly sensitive to the underlying geometry, e.g. in buckling [174]. The use of spline-based geometries is also beneficial for many applications that require a specific continuity, such as some shell models [23]. To that extent, modeling geometries with consistently high continuity is an active area of research, see for example [151].

High Accuracy Compared to conventional Finite Element Methods, it is typically possible to achieve higher accuracy per Degrees Of Freedom, employing the aforementioned refinement strategies [116, 191]. This has already been demonstrated in various fields, notably, structural analysis [54, 55] and fluid mechanics [2, 145].

Of course, these advantages do not come without drawbacks. The high continuity, including across element boundaries, not only improves solution accuracy but also typically results in a larger number of non-zero elements in the assembled matrices and reduces the efficiency of the linear solver compared to standard finite elements. Generally, this increases the computational costs and memory requirements for both assembly and solving, particularly when conventional methods adapted from standard FEM are used (cf. Da Veiga et al. [56] and references therein for more details). Methods to counteract these effects are an active topic of current research, with measures ranging from increasing the efficiency of assembly through more efficient quadrature rules [43] or sum factorization [39], to developing efficient multi-grid solvers [205]. The primary motivation for developing IGA was to provide a natural integration of CAD and numerical analysis. However, a significant chal-

lenge arises from the different representation modes. In particular, CAD predominantly uses a Boundary Representation (B-Rep) - describing the surface between the inside and outside of the domain. While a surface representation (in 3D) may be sufficient for certain shell models [219, 125, 218] used to describe the behavior of thin-walled structures, continuum mechanics applications, whether structural or fluid, require Volumetric Representations (V-Reps) of the computational domain. Contrary to the original intent, this requires a separate "meshing" step to generate an analysis-suitable V-Rep geometry. Many methods have recently been developed to recover such geometries or to make them more suitable for numerical analysis, for example those presented in [121, 101]. Recognizing this limitation, Cohen et al. [50] called for a shift towards analysis-aware modeling and proposed appropriate guidelines. Furthermore, recent efforts towards modeling with V-Reps, including multivariate trimmed B-Splines [156, 157], have already yielded promising results in IGA [13]. Efforts towards volumetric modeling have also been proposed in the field of additive manufacturing, where it is essential to represent the region of interest in detail, also suggesting the incorporation of spline types that can be locally refined [66].

IsoGeometric Analysis is still a relatively new and active area of research. While a comprehensive discussion is beyond the scope of this thesis, the interested reader is referred to the (probably most influential) book [53] as well as the review papers on fluid mechanics [25], structural mechanics [98], design optimization [225], as well as implementation details [168] and the treatment of trimmed B-Reps [154], for further reading.

In the remainder of this section, we will elucidate the basic principles of IsoGeometric Analysis and explain the essential equations and concepts. To this end, we will follow the derivation of one of the PDEs introduced earlier and discuss the underlying concepts and spaces. Although other forms of IGA exist, such as collocation methods [20, 196], immersed IGA [195, 65], but also locally adaptive methods [41], we will only consider the more established Galerkin method with standard tensor product splines, which is more closely related to standard FEM.

3.2.2. Strong and Variational Form

To derive the necessary components for IGA, we will consider the boundary value problem of the heat equation as introduced in Equation (3.7) in the stationary, isotropic case. For simplicity, we will rewrite $\bar{f} = {}^{f}/_{\rho c}$ and $\bar{q}_\Gamma = -{}^{q_\Gamma}/_{\rho c}$. In this context, the minus sign is introduced to represent the flux from the perspective of an external observer for readability. Simply put, a positive flux value indicates the addition of energy to the system. The objective is to find a temperature field $\theta : \bar{\Omega} \to \mathbb{R}$, such that (cf. Section 3.1)

$$\lambda \Delta \theta + \bar{f} = 0 \qquad \text{on } \Omega \quad , \tag{3.17a}$$

$$\theta = T_\Gamma \qquad \text{on } \Gamma_D \quad , \tag{3.17b}$$

$$\lambda \nabla \theta \cdot \mathbf{n} = \bar{q}_\Gamma \qquad \text{on } \Gamma_N \quad . \tag{3.17c}$$

The above equation is referred to as the strong form of the model problem. Following a classical Galerkin approach, the first step is to write Equation (3.17a) in the weak or variational form. This is achieved by multiplying both sides with an arbitrary test function

w and integrating over the domain Ω. After applying Green's identity to the resulting equation, the weak form of the problem reads

$$-\int_\Omega \lambda \nabla w \cdot \nabla\theta \,\mathrm{d}V + \int_{\partial\Omega} \lambda w \,(\nabla\theta \cdot \mathbf{n}) \,\mathrm{d}S + \int_\Omega w\bar{f} \,\mathrm{d}V = 0 \quad , \tag{3.18}$$

which only contains first order spatial derivatives, hence reducing the continuity requirements of the solution.

Considering the BCs from Equation (3.17b), we can now define the set of trial solutions $\mathcal{S}_\theta$ the set of weighting functions $\mathcal{V}$ as (see e.g., [53])

$$\mathcal{S}_\theta = \left\{\theta \,|\, \theta \in \mathcal{H}^1(\Omega),\ \theta = T_\Gamma \text{ on } \Gamma_D\right\} \quad , \tag{3.19a}$$

$$\mathcal{V} = \left\{w \,|\, w \in \mathcal{H}^1(\Omega),\ w = 0 \text{ on } \Gamma_D\right\} \quad . \tag{3.19b}$$

where $\mathcal{H}^1$ denotes the first order Sobolev space. All trial solutions $\theta \subset \mathcal{S}_\theta$ naturally fulfill the Dirichlet BCs on Γ_D, i.e., $\theta|_{\Gamma_D} = T_\Gamma$. The weighting functions can therefore be chosen homogeneous on the Dirichlet boundary, that is, $w = 0$ on Γ_D.

As in the given example the BCs only consists of Dirichlet-type and Neumann-type boundaries, the boundary integral of Equation (3.18) reduces to an integral over Γ_N, where the flux over the boundary is known. Rearranging the terms in terms of unknowns and replacing the boundary integral yields the weak form of the problem: Given $\bar{f}$, $\bar{q}_\Gamma$, find $\theta \subset \mathcal{S}_\theta$, such that for every $w \subset \mathcal{V}$

$$\underbrace{\int_\Omega \lambda \nabla w \cdot \nabla\theta \,\mathrm{d}V}_{a_\theta(\theta,w)} = \underbrace{\int_{\Gamma_N} w\ \bar{q}_\Gamma \,\mathrm{d}S + \int_\Omega w\ \bar{f} \,\mathrm{d}V}_{L_\theta(w)} \quad . \tag{3.20}$$

Here, a_θ and L_θ denote the bilinear and linear form respectively. This new equation will serve as a basis for discretization in the following steps.

3.2.3. Discretization

The key to solving Equation (3.20) numerically is to consider discrete functions θ^h and w^h, of a set of finite dimensional function spaces $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{S}_\theta^h \subset \mathcal{S}_\theta$ yielding the problem formulation: Find $\theta^h \in \mathcal{S}_\theta^h$ such that

$$a_\theta(\theta^h, w^h) = L_\theta(w^h) \qquad \forall w^h \in \mathcal{V}^h \quad . \tag{3.21}$$

Following the core concept of IGA, we will further consider the geometric mapping $\Psi : \tilde{\Omega} \rightarrow \Omega$, which is set to be some tensor-product (NURBS) spline[2], that maps from a parametric domain $\tilde{\Omega}$ into the physical space Ω. We consider the mapping to be bijective, i.e., $\det(\mathbf{J}_\Psi) > 0$, where $\mathbf{J}_\Psi$ denotes the Jacobian matrix of the geometric mapping with $(\mathbf{J}_\Psi)_{ij} = {}^{\partial x_i}/_{\partial \xi_j}$. Recalling Equation (2.7), the mapping consists of

$$\Psi = \sum_i^{n_\mathbf{C}} \mathcal{R}_i \mathbf{C}_i \quad , \tag{3.22}$$

[2]Here, we will consider one spline, or a single-patch geometry. For completeness, a short remark on multipatch geometries has been included at the end of this chapter.

with control points $\mathbf{C}_i$ in physical space. Building on this mapping, we can define our (finite dimensional) solution space based on all linear combinations of the set of basis functions $\mathcal{R}_i$. Therefore, an approximation of the unknown field $\tilde{\theta}^h : \tilde{\Omega} \rightarrow \mathbb{R}$ can be written as

$$\tilde{\theta}^h = \sum_i^{n_{\hat{\theta}}} \mathcal{R}_i \hat{\theta}_i + \sum_i^{n_D} \mathcal{R}_i^D \hat{\theta}_i^D \quad , \tag{3.23}$$

with a set of $n_{\hat{\theta}}$ independent coefficients $\hat{\theta}$, and a set of n_D nodes $\hat{\theta}^D$ on the Dirichlet boundary Γ_D. We assume that a set of boundary coefficients $\hat{\theta}_i^D$ exists to represent the function T_Γ at the boundary Γ_D, otherwise an approximation must be computed using methods such as L_2-projections. Note that in the general case, the number of coefficients in the field approximation may differ from the number of control points associated with the geometry representation, e.g., when refinement is applied. To reduce the number of computations required, the refinement could be applied only to the field and not to the geometry, hence $n_{\mathbf{C}} \neq n_D + n_{\hat{\theta}}$. Although this is done in practice, we will not consider this case in the derivation of the necessary equations.

Since we require the mapping to be invertible, we obtain a suitable function θ^h using the composition

$$\theta^h = \tilde{\theta}^h \circ \Psi^{-1} \quad . \tag{3.24}$$

Pulling back Equation (3.20) into the parametric domain, Equation (3.21) becomes:

$$\begin{aligned} \int_{\tilde{\Omega}} \lambda \, (\tilde{\nabla} \tilde{w}^h \mathbf{J}_\Psi^{-1}) \cdot (\tilde{\nabla} \tilde{\theta}^h \mathbf{J}_\Psi^{-1}) \det(\mathbf{J}_\Psi) \, \mathrm{d}V = & \int_{\Psi^{-1}(\Gamma_N)} \tilde{w}^h (\bar{q}_\Gamma \circ \Psi) \, J^\Gamma \, \mathrm{d}S + \dots \\ & \dots + \int_{\tilde{\Omega}} \tilde{w}^h (\bar{f} \circ \Psi) \det(\mathbf{J}_\Psi) \, \mathrm{d}V \quad , \end{aligned} \tag{3.25}$$

with test functions in the parametric domain $\tilde{w}^h = w^h \circ \Psi$. These set of $n_{\hat{\theta}}$ test functions is defined in a similar fashion to Equation (3.23), but zero on the Dirichlet boundary. Since the integrals are all linear with respect to the test functions, the test function coefficients can be omitted and only the basis functions will be considered.

Here, $\tilde{\nabla}$ denotes the gradient in the parametric domain, such that $\tilde{\nabla}_i = {}^{\partial}/_{\partial \xi_i}$, with coordinates in parametric space $\boldsymbol{\xi}$. Pulling back the gradient from the physical into the parametric domain with Equation (3.24) yields

$$\nabla = \tilde{\nabla} \, \mathbf{J}_\Psi^{-T} \tag{3.26}$$

Further, J^Γ is the surface metric on the boundary of the geometry, defined as $J^\Gamma = \| {}^{\partial \Psi}/_{\partial \tilde{\xi}} \|$ in 2D and $J^\Gamma = \| {}^{\partial \Psi}/_{\partial \tilde{\xi}_1} \times {}^{\partial \Psi}/_{\partial \tilde{\xi}_2} \|$ in 3D, with local coordinates $\tilde{\xi}$ defined on the respective boundary in parametric space.

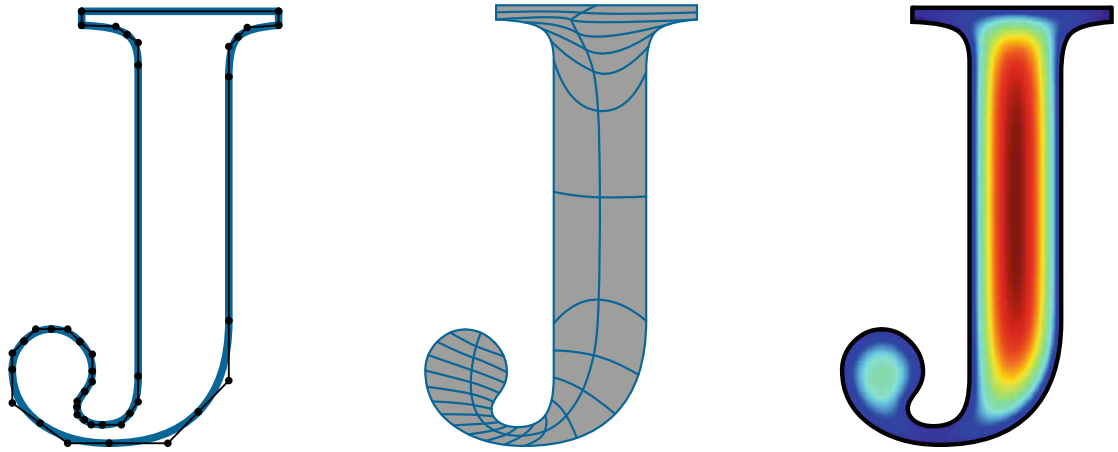
Equation (3.25) results in a system of $n_{\hat{\theta}}$ linear equations

$$\mathbf{K}\boldsymbol{\theta} = \mathbf{F} \quad . \tag{3.27}$$

with the vector of unknown coefficients $\boldsymbol{\theta} = [\hat{\theta}_1, \dots, \hat{\theta}_{n_{\hat{\theta}}}]^T$ and entries

$$\begin{aligned} K_{ij} &= a_\theta(\mathcal{R}_i, \mathcal{R}_j) \quad , \\ F_i &= L_\theta(\mathcal{R}_i) \quad . \end{aligned} \tag{3.28}$$

An illustration of the workflow from geometry representation to IGA solution is shown in Figure 3.2.



(a) Linear and quadratic Bézier curves comprise the letter "J" in the GNU FreeFont FreeSerif [185].

(b) Analysis suitable geometry, generated using the methods presented in [120, 119] with G+SMO.

(c) Solution of the Poisson equation $\Delta u = c$ with homogeneous BCs computed on the geometry using IGA.

Figure 3.2.: Illustration of the Computer Aided Engineering workflow with IGA. Taking the spline representation of a geometry, for example from a CAD design (here in Figure (a) extracted from a font-file, which are generally stored as Bézier curves), a V-Rep geometry suitable for analysis is generated, see Figure (b). Finally, with CAA, the Poisson equation is solved using the methods presented in this chapter, resulting in the solution field shown in Figure (c).

Some Remarks on the Numerical Implementation

In practice, the integrals described in Equations (3.25) are evaluated by numerical integration. Although finding more efficient methods is an active topic of current research [21, 100, 43], in the context of this thesis, the classical Gauss-Legendre quadrature and Bézier extraction is adopted. This method involves defining a set of weights and quadrature points on a reference element, which is subsequently mapped (using a linear mapping $\bar{\Psi}$) into the parametric space. Analogous to the FEM approach, an element is defined as a (tensor product) non-zero knot span of the spline patch. For a clearer perspective, Figure 3.3 provides an overview of the various domains and mappings utilized in this chapter.

The assembly of the linear system is subsequently performed element-wise. In the context of this thesis, we use the open-source software Geometry + Simulation Modules (G+SMO) [123] (pronounced gismo) for assembling these linear systems of equations. G+SMO also provides linear solvers using the Eigen library [97]. For the simulations conducted in this thesis, we will exclusively use the iterative solver BiCGStab, see the Eigen documentation or [217] for further detail.

So far, we have only dealt with a single spline at a time. For most applications, however, a single spline cannot adequately describe the complex topologies of the physical domain. The computational domain therefore consists of a set of patches, each represented by a spline (or patch).

There are several ways to couple the individual patches during the assembly of the linear system [107, 15], such as weak interface coupling using Nitsche's method [169] or mortar method approaches [40], but here we consider only strong coupling, where neighboring

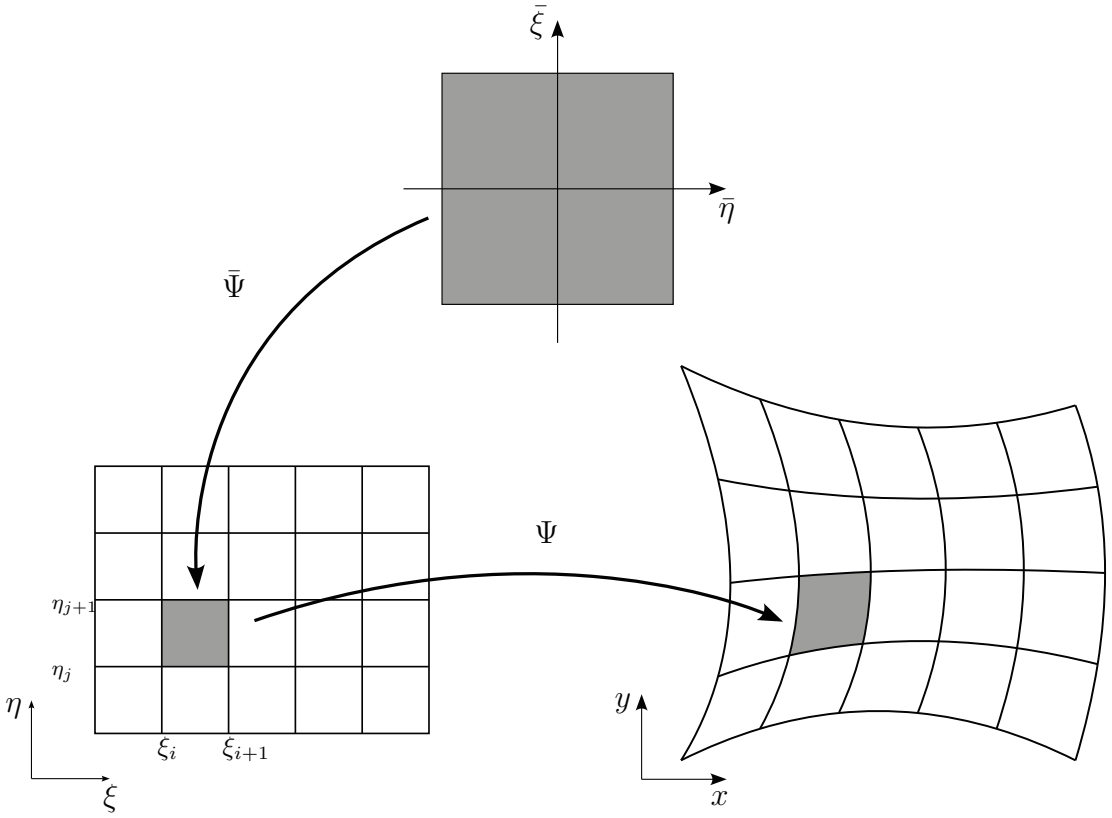


Figure 3.3.: Overview over the different mappings and domains. During assembly, quadrature points are defined on the reference element, which are then projected into the parametric domain $\tilde{\Omega}$ using a linear mapping $\bar{\Psi}$. The trial and test functions are defined on the parametric domain and then mapped into physical space Ω using the geometric mapping Ψ.

spline patches share control points and coefficients on their common interface. As a consequence, we not only require C^0-continuity between adjacent splines, but also assume that they share a common tangential parametrization of the boundary surface. However, they may have different internal parametrizations along normal parametric axes.

3.3. Shape Optimization

3.3.1. Strategies for Numerical Optimization

Structural optimization is a fundamental part of modern engineering, focused on determining the optimal design under given constraints. In fact, numerical optimization as a field of active research goes back as early as the 1960s [197]. Conceptually, design optimization can be divided into three categories [59], with increasing design flexibility:

Sizing Optimization Given a predefined layout, the optimization process focuses on determining the optimal set of thicknesses, widths, or cross-sections. Examples include the appropriate sizing of trusses in a lattice structure [213] and also the optimization of channel geometries using specific set of widths [79]

Shape Optimization Here, the objective is to find the best geometric configuration under prescribed constraints without altering the topology of the design.

Topology Optimization This strategy introduces a higher degree of design freedom by allowing the distribution of material within a specified design space. This may involve creating new voids and structural elements, essentially altering the topological layout of the design.

It's worth noting that the distinction between sizing and shape optimization is sometimes blurry, as sizing optimization can be viewed as a trivial subset within the broader spectrum of shape optimization techniques. Conversely, a 2D topology optimization with density field can equally be interpreted as a sizing problem of the individual cells. Nevertheless, this categorization provides a global understanding of the evolution of design optimization techniques with increasing levels of design flexibility. The different approaches are illustrated by a simple example in Figure 3.4, inspired by [30].

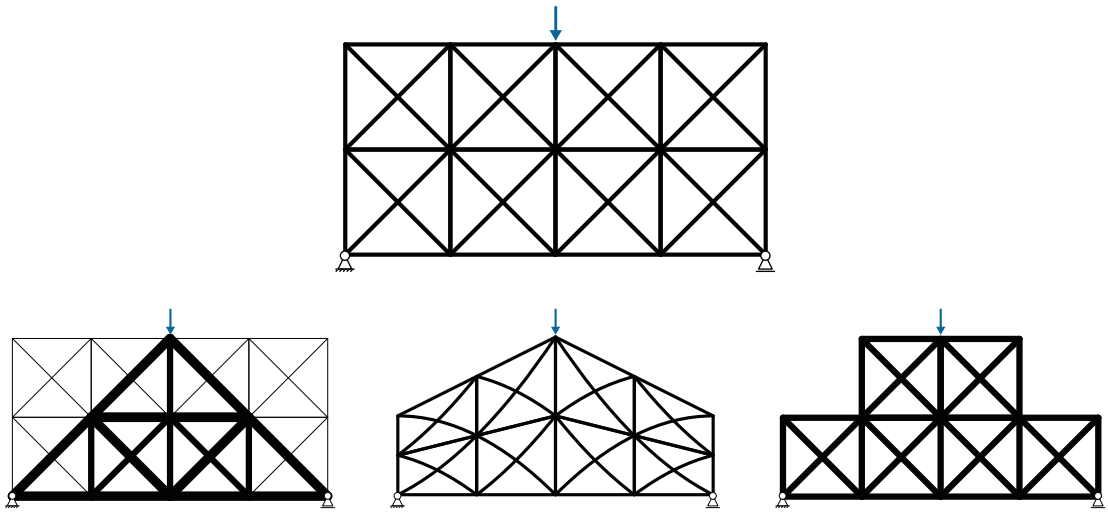


Figure 3.4.: Comparison between the different categories of design optimization applied to lattice structures, inspired by [30]. Starting from a base geometry (top), sizing optimization (left) acts on a predefined set of individual model parameters, such as beam thickness. On the other hand, shape optimization (center) acts – more globally – on the geometry of the domain. Finally, Topology Optimization (right) determines features and can change the connectivity, here shown by removing individual cells. In practice, the distinction between these methods is usually blurry. The geometries shown are not the result of optimization, but are used for illustration purposes only.

In the context of microstructure optimization, a further distinction can be made between concurrent optimization, which involves the simultaneous modification of multiple scales or design features [165], and single-scale optimization, where only local or global features are considered, e.g., the individual truss diameters within a lattice structure [166]. In general, concurrent lattice optimization couples the designs at different scales. It is, therefore, preferable to single-scale optimization, as research suggests that macro scale and micro scale effects strongly interact with each other [231].

3.3.2. PDE-constraint Optimization

The shape optimization problems discussed in this thesis and structural optimization problems in engineering applications generally belong to the class of PDE-constraint optimization problems, where structural performance or the design quality is evaluated in the context of a model problem. The shape optimization problem consists of three main components. The first component is the *geometry module* $\mathcal{O}^{\Psi}$, which defines the structural design as a function of a set of geometric design variables $\boldsymbol{\chi}$ (cf. Equation (2.25)). The geometry module provides the optimization process with an appropriate geometry for the second component - the *solver* for numerical analysis. This analysis is performed based on the generated geometry to provide the necessary fields required to asses the current design's performance. Finally, the *optimization driver* $\mathcal{O}^{\boldsymbol{\chi}}$ updates the design variables and thus the underlying geometry. The general outline of the optimization framework is illustrated in Figure 3.5. The geometry module includes an optional step of discretization, which may be necessary to prepare an analysis-suitable geometry representation. This may involve modifying or refining the mesh, particularly in the context of Finite Element Method (FEM), or appropriately refining the underlying geometry. Furthermore, some optimization drivers require information about the gradient of the objective function, which will be covered in detail in Section 3.4.

Considering the objective function $\mathcal{J}$, evaluating the structural performance of a given design, the shape optimization problem in its basic form can be written as

$$\begin{aligned} &\min_{\forall \boldsymbol{\chi}} \; \mathcal{J}\left(\theta,\, \boldsymbol{\chi}\right) && , \\ \text{subject to} \quad &\mathbf{c}\left(\Omega(\boldsymbol{\chi}),\, \theta\right) = \mathbf{0} && , \end{aligned} \tag{3.29}$$

where $\mathbf{c}$ represents the PDE constraint in its residual form. In structural optimization however, the domain Ω and the solution field θ can also be regarded as a function of the design variables $\boldsymbol{\chi}$, i.e., $\theta(\boldsymbol{\chi})$. Furthermore, additional equality constraints $\mathbf{k}_E$ and inequality constraints $\mathbf{k}_I$ can be imposed on the optimization problem. In the fabrication and design of microstructures, these can include, e.g., restricting the total mass [137], limiting the design space to admissible design variables $\boldsymbol{\chi}$, or introducing manufacturing constraints [6, 212, 198]. These constraints can be a function of the domain (e.g., mass constraints), of the design variables (e.g., box constraints to maintain an admissible range of design parameters) and of the solution field (e.g., maximum stresses, temperature limits), and therefore can be written as $\mathbf{k}\left(\theta(\boldsymbol{\chi}),\, \Omega(\boldsymbol{\chi}),\, \boldsymbol{\chi}\right)$, making them a function that depends purely on the design variables. Using this information, we can introduce a new (albeit less rigorous) definition of the objective function $\hat{\mathcal{J}}(\boldsymbol{\chi}) = \mathcal{J}\left(\theta(\boldsymbol{\chi}),\, \boldsymbol{\chi}\right)$ to facilitate the treatment of these additional constraints in the form

$$\begin{aligned} &\min_{\boldsymbol{\chi}} \; \hat{\mathcal{J}}(\boldsymbol{\chi}) && , \\ \text{subject to} \quad &\mathbf{k}_I\left(\boldsymbol{\chi}\right) \leq \mathbf{0} && , \\ &\mathbf{k}_E\left(\boldsymbol{\chi}\right) = \mathbf{0} && , \end{aligned} \tag{3.30}$$

with Lagrangian

$$\hat{\mathcal{L}}(\boldsymbol{\chi},\, \boldsymbol{\lambda}) = \hat{\mathcal{J}}(\boldsymbol{\chi}) + \boldsymbol{\lambda}_I^T\, \mathbf{k}_I(\boldsymbol{\chi}) + \boldsymbol{\lambda}_E^T\, \mathbf{k}_E(\boldsymbol{\chi}) \quad , \tag{3.31}$$

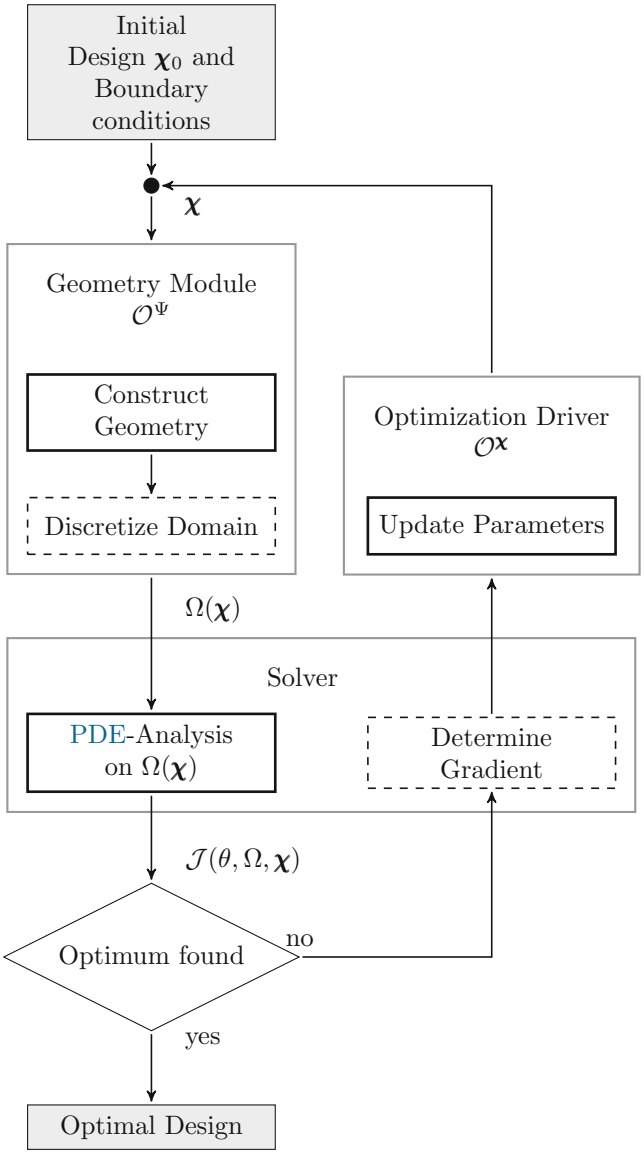


Figure 3.5.: General outline of the optimization framework for PDE constraint optimization, based on [238]. Here, we assume that the gradient is computed on the solver side, which is typically the case when using an adjoint approach. Furthermore, the termination criterion is in most cases also evaluated by the optimization driver.

with Lagrange multipliers $\boldsymbol{\lambda} = \boldsymbol{\lambda}_I \cup \boldsymbol{\lambda}_E$. This Lagrangian $\hat{\mathcal{L}}$ is a prerequisite for defining the first-order necessary condition for optimality, also known as the Karush-Kuhn-Tucker (KKT)-conditions [131]. It states, that for any local solution $\boldsymbol{\chi}^*$, there exists a Lagrange multiplier $\boldsymbol{\lambda}^*$, such that the following conditions are satisfied

$$\partial_{\boldsymbol{\chi}} \hat{\mathcal{L}}(\boldsymbol{\chi}^*,\ \boldsymbol{\lambda}^*) = \mathbf{0} \quad , \tag{3.32a}$$

$$\mathbf{k}_I\,(\boldsymbol{\chi}^*) \leq \mathbf{0} \quad , \tag{3.32b}$$

$$\mathbf{k}_E\,(\boldsymbol{\chi}^*) = \mathbf{0} \quad , \tag{3.32c}$$

$$\boldsymbol{\lambda}_I^* \geq \mathbf{0} \quad , \tag{3.32d}$$

$$\boldsymbol{\lambda}_I^{*T}\ \mathbf{k}_I = 0 \quad . \tag{3.32e}$$

Equation (3.32e) implies that (row-wise) either the inequality constraint is active, i.e., $\mathbf{k}_I(\boldsymbol{\chi}) = 0$ or the Lagrange multiplier is 0. The active inequality constraints are referred to as the active set. For a more comprehensive explanation and the definition of second-order necessary and sufficient conditions, we refer to [172], which also provides a mathematically sound explanation, aspects of which were neglected here for simplicity. The optimization problem defined in Equation (3.30) is treated by the optimization driver $\mathcal{O}^{\boldsymbol{\chi}}$. Together with the geometry module $\mathcal{O}^{\Psi}$, the analysis driver provides the required function evaluations and derivatives. Computing the derivative of the Lagrangian with respect to the design variables $\boldsymbol{\chi}$ requires differentiating the objective function as stated in Equation (3.21). This is not a trivial task and will be discussed in detail in section 3.4.

3.3.3. IsoGeometric Analysis and Shape Optimization of Lattice Structures

As highlighted in Section 3.2, the integration of IsoGeometric Analysis, which inherently bases its numerical analysis on the shape representation, constitutes a natural complement to the classical shape optimization framework. It allows (a subset of) the geometry's control points to serve as design variables for the optimization problem. This largely omits the intermediate step of providing an appropriate analysis mesh to the FEM-model and eliminates the reintegration of geometric changes into the CAD-model. Despite numerous applications of isogeometric analysis in design optimization, as evidenced by works such as [86], its use in the specific context of lattice structures has been relatively limited. In many applications using IGA with microstructure geometries, the focus is often solely on designing either scale, for example to find local geometries which lead to auxetic macroscopic behavior [87], sometimes analyzed in combination with reduced order models [164].

IGA has also been applied to lattice structures comprised of beam elements, where the one-dimensional element description is used to save computational costs. In [227], Weeger, Narayanan, and Dunn used a global optimization approach to modify the control points defining the beam centerline to minimize the compliance of the structures. The same approach was later extended by the author to include other cross-sectional properties, such as thicknesses and material properties, and has been used in various applications, such as the design of meta-material unit cells or shape memory composites [226]. However, this approach treats all centerline control points as individual design variables, requiring the implementation of specific constraints to minimize deviation from the original shape. As a result, this method is limited to specific macro geometries, and macro shape optimization cannot be performed easily.

In an effort to optimize conformal microstructures, the study presented by Choi and Cho [47] uses a different methodology based on the concept of Free-Form Deformation. In their approach, the lattice structure is initiated by constructing a spline geometry on a planar surface and then using FFD on sample points to map the lattices onto a curved surface (see Section 2.2.4). The optimization goal is to achieve a conformal microstructure within the surface representation, so the samples are fitted with a NURBS curve. They further used IGA-beam elements and analytical derivatives for the optimization of the lattices. However, despite efforts to minimize the fitting error through refinement, the spline itself may deviate outside the surface, introducing a geometric error.

Finally, Antolin et al. [14] suggested using the microstructure design process described in Section 2.3 for shape optimization. This approach allows the optimization of both the local geometry and the macro shape, individually or simultaneously. In contrast to the previous methods, analytical conformity is ensured up to machine accuracy and the geometry output is inherently CAD-compliant.

3.3.4. An Analogy with Topology Optimization

Topology Optimization (TO) encompasses a range of methods aimed at determining the optimal layout or distribution of material within a given design space to achieve specific performance objectives. These methods include density-based approaches, such as the Solid Isotropic Material with Penalization (SIMP) method [28], free material design techniques [29], homogenization method [11], approaches based on stochastic optimization strategies like genetic algorithms [124], evolutionary strategies [186], and many more. This thesis will not cover a comprehensive review of topology optimization methods; instead, the interested reader is referred to [30, 10, 9].

At their core, homogenization approaches, SIMP and free material design techniques incorporate fictitious materials or fields as design variables for optimization. While all entries of the elasticity tensor can serve as design variables for free material design, the SIMP method scales the elasticity tensor with a density distribution field. In SIMP, the penalization step forces the density to take values closer to 0 and 1 in different post-processing stages to extract a "real" shape. In the 2D case, a density field can be interpreted as a plate with varying thickness. Thus, the problem can be interpreted as sizing the individual cells, again blurring the boundaries between sizing, shape, and topology optimization.

The idea of using a density field for optimization also translates to microstructure construction and has already found applications in other construction methods for microstructures. Previous work has often leveraged these density fields by (1) homogenizing structures composed of lattice geometries to extract their material properties and (2) employing topology optimization methods to select the cell that best suits the local loading requirements. In [113], the authors applied IGA at the macro level, using an approach similar to SIMP within a B-Spline representing the macro geometry. Following this analysis, the density distribution obtained was used as a basis for designing the microstructure using Triply Periodic Minimal Surface (TPMS)-based local geometries defined in the parametric domain of the spline. Similarly, in [224] the authors used SIMP alongside asymptotic homogenization [16] to optimize compliance. However, no IGA-based analysis or optimization was directly applied to the full model of the microstructure itself.

In the parametrization technique proposed in this work, the abstraction of local cell parameters in the form of a parameter spline can be interpreted as a density field, analogous to TO approaches. However, in contrast to these techniques, our concurrent parametrization – i.e., the simultaneous modification of local and global properties – extends this idea to include, to some extent, the interaction between macroscopic and microscopic effects. The deformation of the outer geometry and the resulting orientation of the local structure promises a better force distribution of the structure. In addition, by abstracting the thickness distribution, we can vary the local structure and achieve a continuous variation of the thickness through the smooth non-zero basis functions of splines.

3.3.5. On the Optimization Driver

In the iterative process of a common optimization framework, the optimization driver plays a critical role in determining the evolution of design variables toward an optimal solution. This updating step employs various optimization algorithms that rely on information acquired by previous iterations. The vast field of optimization algorithms can generally be divided into two distinct groups based on the information provided. Gradient-based approaches, such as steepest descent methods [159] or quasi-Newton methods [222], are known for their faster convergence rates, resulting in fewer objective function evaluations and better overall performance. However, their requirement for gradient information, which is often analytically inaccessible, poses a challenge.

In contrast, gradient-free methods [182] require only objective function evaluations. Such methods are widely used as computing the gradient can be an expensive and cumbersome task. Recent advances have explored the use of machine learning as gradient-free optimization algorithms in shape optimization [85]. In addition, evolutionary algorithms have found applications in microstructure optimization [64].

When the gradient is required, techniques such as adjoint methods [214] (see Section 3.4), algorithmic differentiation [161, 96], or finite differences [70] are used to approximate the gradient. While adjoint methods are computationally efficient, they present challenges in terms of implementation complexity, especially in systems with black-box components. Conversely, (forward) algorithmic differentiation[3] and finite differences are more accessible but come with additional costs, generally requiring an additional evaluation of the PDE constraint for each design parameter. This makes them computationally expensive, especially for complex geometry parametrizations. In the following section, we introduce the adjoint method, a powerful technique for efficiently computing gradients in shape optimization.

3.3.6. Short Remark on the Optimization Framework

The framework for the optimizations conducted in this thesis is constructed in Python and follows the principles shown in Figure 3.5. For geometry construction, we use splinepy, our in-house prototyping tool for geometry construction. splinepy provides all the methods and functionality needed to construct the microstructures described in Section 2 as well as their derivatives. splinepy utilizes a fast `C++` backend for expensive operations such as spline composition and evaluation, while providing a simple Python interface and many input/output options, including for Geometry + Simulation Modules (G+SMO) [123]. G+SMO, the open-source simulation module, is used for discrete adjoint analysis and computation. To facilitate the implementation of new PDEs, G+SMO provides an expression assembler, which has also been used in the context of this thesis. Special Python bindings have been written to facilitate integration into the optimization framework, which can be found in reference [237][4].

[3] Although computing the gradient using backward algorithmic differentiation is efficient for a high number of variables, its implementation is very complex and is associated with high memory usage. Therefore, it is not considered here.

[4] At the time this was written, Python bindings for the expression assembler were unavailable. However, G+SMO's rapid development now includes a new framework supporting the expression assembler natively through just-in-time compilation.

The optimization driver uses the Python bindings provided by the Nonlinear Optimization Framework (NLopt) [122], a freely available library designed for nonlinear optimization tasks. NLopt provides a unified interface for accessing various optimization routines from different online repositories, as well as original algorithms developed within the NLopt project. This allows the selection and testing of different optimization algorithms without major intervention in the optimization framework. The selected algorithm is specified for each application in the following chapters. Additionally, the SLSQP implementation provided by the `scipy` [221] library will be used in some of the test cases.

3.4. Computing Derivatives

Adjoint methods, based on the principles of Lagrange multipliers, have been widely used in shape optimization because of their computational efficiency and versatility. These methods provide a powerful means to efficiently compute gradients, which play a crucial role in sensitivity analysis for shape optimization. For a deeper understanding and more comprehensive exploration of adjoint methods, the interested reader is referred to [8, 45, 93]. In the following section, we will present the adjoint method by deriving the necessary equations from the Lagrangian formulation based on the model problem introduced in Chapter 3.2[5]. The objective is to find the derivatives of the objective function with respect to the design variables $\boldsymbol{\chi}$, therefore we will disregard all additional constraints $\mathbf{k}$. These either do not depend on the solution of the PDE-constraint or are treated separately by the optimization driver, as discussed in Section 3.3.2.

Recalling the optimization problem introduced in Equation (3.29), the Lagrangian can be defined as

$$\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) = \mathcal{J}\left(\boldsymbol{\chi}, \theta\right) + \boldsymbol{\lambda}^T \mathbf{c}\left(\theta, \Omega(\boldsymbol{\chi})\right) \quad , \tag{3.33}$$

where $\mathcal{L}$ defines the Lagrangian of the optimization problem, together with a vector of Lagrange-multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{n_{\hat{\theta}}}$, i.e., one for every DOF. The superscript h indicating the discretized functions is omitted for readability. As the PDE constraint $\mathbf{c}$ is written in residual form, i.e., $\mathbf{c} = \mathbf{0}$, it follows that

$$\begin{aligned} \mathrm{d}_{\boldsymbol{\chi}}\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) &= \mathrm{d}_{\boldsymbol{\chi}}\mathcal{J}\left(\boldsymbol{\chi}, \theta\right) \\ &= \partial_{\boldsymbol{\chi}}\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) + \partial_{\theta}\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) \mathrm{d}_{\boldsymbol{\chi}}\theta \quad . \end{aligned} \tag{3.34}$$

Here, we use the notation $\partial(\bullet)$ to write partial derivatives, and $\mathrm{d}(\bullet)$ to write a total derivative. The total derivatives $\mathrm{d}_{\boldsymbol{\chi}}\theta$ denote implicit derivatives, which are computationally unattainable without solving for every component of $\boldsymbol{\chi}$. Therefore, crucially, we chose $\boldsymbol{\lambda}$ to satisfy the adjoint equation

$$\begin{aligned} \partial_{\theta}\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) &= \mathbf{0} \\ &= \partial_{\theta}\mathcal{J}\left(\boldsymbol{\chi}, \theta\right) + \boldsymbol{\lambda}^T \partial_{\theta}\mathbf{c}\left(\theta, \Omega(\boldsymbol{\chi})\right) \quad , \end{aligned} \tag{3.35}$$

which eliminates the second term in Equation (3.34) and thus reduces the total derivative of the Lagrangian to

$$\mathrm{d}_{\boldsymbol{\chi}}\mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) = \partial_{\boldsymbol{\chi}}\mathcal{J}\left(\boldsymbol{\chi}, \theta\right) + \boldsymbol{\lambda}^T \partial_{\boldsymbol{\chi}}\mathbf{c}\left(\theta, \Omega(\boldsymbol{\chi})\right) \quad . \tag{3.36}$$

[5] The same principles can be applied to other PDEs.

From Equations (3.35) and (3.36), we can develop a cost efficient procedure to determine the gradient of the objective function. Starting from the design variables $\boldsymbol{\chi}$, we first

1. solve the PDE-constraint $\mathbf{c}(\theta, \Omega(\boldsymbol{\chi})) = \mathbf{0}$ using Equation (3.27)
2. determine the Lagrange multipliers $\boldsymbol{\lambda}$ using

$$\boldsymbol{\lambda} = -\left(\partial_\theta \mathbf{c}\left(\theta, \Omega(\boldsymbol{\chi})\right)\right)^{-T} \partial_\theta \mathcal{J}\left(\boldsymbol{\chi}, \theta\right)^T \tag{3.37}$$

3. evaluate the gradient using Equation (3.36)

This procedure allows the computation of the gradient with only one additional solution of a linear system of equations, as opposed to $n_{\boldsymbol{\chi}}$ additional linear solutions that would be required by the Finite Differences approach.

Without loss of generality, the above equations can be derived in the same manner with respect to the control points of the geometry representation, i.e.,

$$\mathrm{d}_{\boldsymbol{\chi}} \mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) = \mathrm{d}_{\mathbf{C}} \mathcal{L}\left(\theta, \boldsymbol{\chi}, \boldsymbol{\lambda}\right) \partial_{\boldsymbol{\chi}} \mathbf{C} \quad . \tag{3.38}$$

In Section 2.4, the derivative of the microstructure based on the functional composition was derived and a one-to-one relationship between the control points of the geometry and its derivative representation was established, which means that Equation (3.38) can be written as a matrix multiplication. The derivative with respect to the control points can also be used to freely expand the design space once the microstructure has been constructed. Therefore, in the following we will consider the derivatives with respect to the control points and then use the matrix multiplication to restrict our design space to the proposed construction process.

The derivative of the PDE-constraint function $\mathbf{c}$ with respect to the (here) temperature field θ, required to solve the adjoint Equation (3.35), is already computed during the analysis step, and can be recovered, the derivatives of the objective function are mostly trivial to compute and will be discussed separately for each application. In the case of linear elasticity and heat transfer problems, we also benefit from the symmetry of the system matrix. Recalling the definitions of the bilinear forms in Equation (3.25) in the pulled back configuration and by applying the product rule to the individual terms of the bilinear form, the derivative of the constraint function with respect to the control points $\partial_{\mathbf{C}} \mathbf{c}$ yields

$$\partial_{\mathbf{C}} \mathbf{c}\left(\theta, \Omega(\boldsymbol{\chi})\right) = \frac{\partial a_\theta(w, \theta)}{\partial \mathbf{C}} - \frac{\partial L_\theta(w)}{\partial \mathbf{C}} \quad , \tag{3.39}$$

with the temperature solution of the heat transfer problem θ, $\tilde{\theta}$ and a the test functions w, $\tilde{w}$ in their pulled back (parametric) configuration, respectively

$$\begin{aligned} \frac{\partial a_\theta(w, \theta)}{\partial \mathbf{C}_i} = &\int_{\tilde{\Omega}} \lambda \left(\tilde{\nabla}\tilde{w} \, \frac{\partial \mathbf{J}_\Psi^{-1}}{\partial \mathbf{C}_i}\right) \cdot \left(\tilde{\nabla}\tilde{\theta} \, \mathbf{J}_\Psi^{-1}\right) \det\left(\mathbf{J}_\Psi\right) \mathrm{d}V \\ &+ \int_{\tilde{\Omega}} \lambda \left(\tilde{\nabla}\tilde{w} \, \mathbf{J}_\Psi^{-1}\right) \cdot \left(\tilde{\nabla}\tilde{\theta} \, \frac{\partial \mathbf{J}_\Psi^{-1}}{\partial \mathbf{C}_i}\right) \det\left(\mathbf{J}_\Psi\right) \mathrm{d}V \\ &+ \int_{\tilde{\Omega}} \lambda \left(\tilde{\nabla}\tilde{w} \, \mathbf{J}_\Psi^{-1}\right) \cdot \left(\tilde{\nabla}\tilde{\theta} \, \mathbf{J}_\Psi^{-1}\right) \frac{\partial \det\left(\mathbf{J}_\Psi\right)}{\partial \mathbf{C}_i} \mathrm{d}V \quad , \end{aligned} \tag{3.40}$$

and

$$\frac{\partial L_\theta(w)}{\partial \mathbf{C}_i} = \int_{\Psi^{-1}(\Gamma_N)} \tilde{w}\ (\bar{q}_\Gamma \circ \Psi)\ \frac{\partial J^\Gamma}{\partial \mathbf{C}_i}\,\mathrm{d}S + \int_{\tilde{\Omega}} \tilde{w}\ (\bar{f} \circ \Psi)\ \frac{\partial \det\left(\mathbf{J}_\Psi\right)}{\partial \mathbf{C}_i}\,\mathrm{d}V \quad . \tag{3.41}$$

Here, we make use of the following identities[6] and apply Jacobi's formula

$$\frac{\partial \mathbf{J}_\Psi^{-1}}{\partial \mathbf{C}_i} = -\mathbf{J}_\Psi^{-1}\frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i}\mathbf{J}_\Psi^{-1} \tag{3.42a}$$

$$\frac{\partial \det(\mathbf{J}_\Psi)}{\partial \mathbf{C}_i} = \det(\mathbf{J}_\Psi)\mathrm{tr}\left(\mathbf{J}_\Psi^{-1}\frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i}\right) \quad . \tag{3.42b}$$

Using the definition of the geometric mapping and it's geometry from Equation (3.22), we obtain[7]

$$\frac{\partial \Psi}{\partial \mathbf{C}_i^k} = \mathcal{R}_i \quad , \tag{3.43}$$

$$\frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i^k} = \tilde{\nabla}\mathcal{R}_i \quad , \tag{3.44}$$

where the index k refers to the k^{th} component (physical dimension) of the control point. Differentiating $^{\partial \mathbf{J}_\Psi}/_{\partial \mathbf{C}_i}$ would result in a third order tensor. In practice, the individual components are computed separately and are treated as independent DOFs, which also helps to avoid the necessity for higher dimensional matrices.

Substituting the derivatives in Equation (3.42) yields

$$\begin{aligned}\frac{\partial a_\theta(w,\theta)}{\partial \mathbf{C}_i} = &-\int_{\tilde{\Omega}} \lambda\ (\tilde{\nabla}\tilde{w}\ \mathbf{J}_\Psi^{-1}\ \frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i}\ \mathbf{J}_\Psi^{-1}) \cdot (\tilde{\nabla}\tilde{\theta}\ \mathbf{J}_\Psi^{-1}) \det\left(\mathbf{J}_\Psi\right)\,\mathrm{d}V \\ &-\int_{\tilde{\Omega}} \lambda\ (\tilde{\nabla}\tilde{w}\ \mathbf{J}_\Psi^{-1}) \cdot (\tilde{\nabla}\tilde{\theta}\ \mathbf{J}_\Psi^{-1}\ \frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i}\ \mathbf{J}_\Psi^{-1}) \det\left(\mathbf{J}_\Psi\right)\,\mathrm{d}V \\ &+\int_{\tilde{\Omega}} \lambda\ (\tilde{\nabla}\tilde{w}\ \mathbf{J}_\Psi^{-1}) \cdot (\tilde{\nabla}\tilde{\theta}\ \mathbf{J}_\Psi^{-1}) \det(\mathbf{J}_\Psi)\ \mathrm{tr}\left(\mathbf{J}_\Psi^{-1}\ \frac{\partial \mathbf{J}_\Psi}{\partial \mathbf{C}_i}\right)\,\mathrm{d}V \quad .\end{aligned} \tag{3.45}$$

Similar considerations also apply to the boundary integral with the surface metric J^Γ. To differentiate the linear form and to get a generalized equation, we rewrite the metric in a dimension-independent expression [78, 139], which yields

$$J^\Gamma = \left(\sum_{j=1}^{d} \det\left(M_{jk}^\Psi\right)^2\right)^{\frac{1}{2}} \quad , \tag{3.46}$$

where M_{jk}^Ψ denotes the j,k^{th} minor of the Jacobian matrix $\mathbf{J}_\Psi$, namely where the j^{th} row and the k^{th} column are removed, the index k corresponds to the parametric dimension

[6] These arise from applying the product rule to $^{\partial\left(\mathbf{J}_\Psi^{-1}\mathbf{J}_\Psi\right)}/_{\partial \mathbf{C}_i} = 0$

[7] As mentioned before, the geometry representation Ψ and trial functions might differ due to refinement. While the different function spaces may be considered in the implementation to reduce computational cost, here the difference will be left out to simplify the necessary equations.

that is normal to the boundary in parametric space. This expression allows calculating the derivative with respect to the control points of the geometry. It reads

$$\frac{\partial J^{\Gamma}}{\partial \mathbf{C}_i} = \frac{1}{J^{\Gamma}} \sum_{j=1}^{d} \left[\det\left(M_{jk}^{\Psi}\right) \mathrm{tr}\left(\left(M_{jk}^{\Psi}\right)^{-1} \frac{\partial M_{jk}^{\Psi}}{\partial \mathbf{C}_i} \right) \right] \quad , \tag{3.47}$$

where the derivative of the Jacobian minor with respect to the control points is the minor of the derivative of the Jacobian itself, see Equation (3.44). It is essential to note that, in the context of this work, the surface integrals exclusively appear on the Neumann boundary Γ_N. In the majority of shape optimization problems, however, this boundary remains immutable. Its mention here is included for the sake of completeness. The optimization framework is summarized in Algorithm 3.1.

Algorithm 3.1: Solution strategy for a shape optimization problem with sensitivities obtained using the method of discrete adjoints.

Input:

$\boldsymbol{\chi}_0$	Initial vector of design variables
$\mathcal{O}^{\Psi}$	Geometry Module
$\mathcal{O}^{\boldsymbol{\chi}}$	Optimization driver

Output:

$\boldsymbol{\chi}$	Vector of optimized design variables

Algorithm:

$\boldsymbol{\chi} \leftarrow \boldsymbol{\chi}_0$; // Initialize system

while *not optimized* **do**

  Geometry Module

  $\Omega(\boldsymbol{\chi}), \partial_{\boldsymbol{\chi}}\mathbf{C} \leftarrow \mathcal{O}^{\Psi}(\boldsymbol{\chi})$; // Determine parametrized geometry and its derivatives with respect to the vector of design variables $\boldsymbol{\chi}$ (cf. Algorithm 2.1)

  PDE Solver on $\Omega(\boldsymbol{\chi})$

  $\boldsymbol{\theta} \leftarrow (\mathbf{K})^{-1}\,\mathbf{F}$; // Solve constraint function and determine solution field (cf. Equation (3.27))

  $J \leftarrow \mathcal{J}(\boldsymbol{\theta})$; // Evaluate objective function based on solution field

  $\boldsymbol{\lambda} \leftarrow (\mathbf{K})^{-T}\,\partial_{\boldsymbol{\theta}} J^T$; // Determine Lagrange multipliers, requires solving an additional linear system of equations (cf. Equation (3.35))

  $\mathrm{d}_{\mathbf{C}} J \leftarrow \partial_{\mathbf{C}} J + \boldsymbol{\lambda}^T\,\partial_{\mathbf{C}}\mathbf{c}\,(\theta, \Omega(\boldsymbol{\chi}))$; // Determine sensitivities of objective function using adjoint approach (cf. Equation (3.39))

  $\mathrm{d}_{\boldsymbol{\chi}} J \leftarrow \mathrm{d}_{\mathbf{C}} J\,\partial_{\boldsymbol{\chi}}\mathbf{C}$; // Map the control point sensitivities onto the design variables using their geometric derivatives (cf. Equation (3.38))

  Optimization Driver

  $\boldsymbol{\chi} \leftarrow \mathcal{O}^{\boldsymbol{\chi}}(J, \partial_{\boldsymbol{\chi}} J)$; // Update design variables based on solver results

end

3.5. Discussion

With node-based methods, where every control point of an original design serves as a design variable, IsoGeometric Analysis provides a CAD-compliant optimization method for free, "unconstrained" shape optimization, enabling the design of microstructures with minimal limitations. In contrast, the proposed method for constructing microstructures appears to be overly restrictive, as it only allows for composition-based outcomes, significantly limiting design freedom. At the same time, the resulting geometry representation has high polynomial degrees, which are associated with high computational costs for both analysis and geometry construction. Therefore, the question naturally arises:

Why use this construction method at all?

First, the proposed method achieves **analytical conformity** by populating a domain using functional composition, aligning the local tiles both with the surface and the internal parametrization of the macroshape. This offers a clear advantage over traditional methods, where only point-wise boundary conformity is achieved, or the internal structure remains in a Cartesian grid while only the boundary is conformal. Conformity plays a key role in the adoption of this technique, as the macroscopic shape is often designed to optimize the stress distribution in structural mechanics or the heat flux in thermal problems, guided by external factors. Aligning the microtiles with the original design that fulfills these objectives promises undisturbed flow through the microstructured design and can potentially outperform Cartesian grid geometries. Furthermore, concurrent optimization of outer and local geometries allows tile distribution and orientation adjustments via control points, while stress variations can be addressed by modifying the local tile geometry, e.g., through increasing thickness. This approach works even when the design's outer contour remains unchanged, by adjusting only internal control points.

The composition-based approach only utilizes components from the initial design, **facilitating integration** into the established design workflow. In other words, when a pre-designed part is used as the outer shape, or deformation function, the design can be populated with a microstructured geometry without affecting surrounding or connecting components. In classical approaches that fill macro-domains in a Cartesian grid-like fashion, geometric features and connections to adjacent parts require special attention to ensure watertightness and prevent "floating" lattices, especially in areas of high surface curvature. The composition-based approach can circumvent these issues by using closed tiles at the boundary, defined once in the reference element, which naturally seal the boundary if needed (see Section 4.3).

Restricting the design space to local tile parameters and the control points of the deformation function — rather than using all control points from an initial microstructure — helps **mitigate entanglement issues** and other challenges associated with node-based approaches. In traditional node-based methods, additional constraints must be imposed on all patches and nodes to prevent overlapping regions in the model. In contrast, the composition-based approach ensures regularity as long as both the inner and outer functions are regular. However, the parametrized microtile naturally fulfills this requirement. As a result, only the deformation function, where control points can be modified freely, requires checking, significantly reducing the number of necessary constraints.

Moreover, appropriate parametrization of the local structure can be used to **ensure manufacturability**. By carefully setting bounds on the parametrization of the local

structure, manufacturability constraints such as minimum thickness requirements can be addressed directly, or in combination with characteristics of the deformation function before insertion. This minimizes the need for strict constraints during the design process.

In addition, the design space is **low dimensional**, consisting only of the parameter spline coefficients and the control points of the deformation function. Low dimensionality is typically associated with faster convergence in the context of shape optimization, especially when information about the gradient of the objective function is not available.

Finally, it is possible to construct pairs of microtiles that, together, form a conformal representation of the unit cube in the parametric domain. This expands the range of possible applications to designing microstructures and their corresponding **inverse geometries** for fluid-structure interaction problems, such as static mixers for plastic extrusion processes.

However, it is also important to acknowledge certain limitations or challenges associated with the proposed method. While IGA has proven to be a powerful tool for structural analysis, its application to microstructures can be **computationally expensive**, especially due to the high-fidelity models required to represent the intricate local structures and the high polynomial orders from the construction process. In response, recent advancements have been introduced to mitigate computational costs. The high polynomial orders and continuities are especially numerically demanding during the assembly process, among other things, because numerical integration requires a high number of quadrature points. This holds especially true for 3D applications. Hirschler, Antolin, and Buffa [103] proposed a multi-scale assembly procedure based on lookup tables and polynomial approximation on the macro scale to reduce the computational demand associated to assembling the linear system of equations. The same authors also discuss the development of inexact solvers in [105]. They exploit the inherent periodicity of cells to take advantage of their mechanical and geometric similarities, reducing the complexity of the system to a few main local problems using cell-wise operators.

4 Microstructure Optimization for Heat Transfer Problems

4.1. Introduction

Heat transfer problems play an important role in several industrial and real-world applications, ranging from conjugate heat transfer in systems involving both solids and fluids [147] to thermal management of electronic components [26]. Optimization of thermal processes is therefore essential in system design to improve energy efficiency and ensure the reliability of critical components. While a comprehensive examination of all aspects of heat transfer is beyond the scope of this thesis, the review by Dbouk [60] provides valuable insights into topology optimization techniques for heat transfer problems. Therein, the author investigates different solution strategies and objective functions, highlighting the advantages and disadvantages of different approaches.

4.1.1. Preliminary Work

In the following chapter we will explore the application of microstructures in the context of heat transfer problems. First, we will introduce appropriate objective functions and their differentiation. Then, we demonstrate the presented methods on a minimal example. Through this test case, we also evaluate the performance of different optimization algorithms in the context of heat diffusion systems. Finally, we will present a real-world application with a more complex geometry where we apply our method to optimize an extrusion die.

While the data provided in this thesis has been generated specifically for this work, it is important to acknowledge that the test cases discussed herein are based on the findings presented in

---

[238] Jacques Zwar, Gershon Elber, and Stefanie Elgeti. "Shape optimization for temperature regulation in extrusion dies using microstructures". In: *Journal of Mechanical Design* 145.1 (2023)

As an extension to the test cases presented therein, the applications featured in the following sections include (1) the adoption of IsoGeometric Analysis instead of standard FEM with linear elements, (2) the utilization of gradients obtained through discrete adjoints,

and (3) the use of true 2D problems, as opposed to quasi-2D, where 3D crosses were used to fill an extruded domain. In addition, the local geometry was slightly modified to cope with the high number of DOFs. The changes are highlighted in the respective sections.

4.1.2. Objective Function

The objective function of an optimization problem assesses the quality of a given design. Therefore, the choice of an appropriate objective function depends on the specific application. In the context of (diffusive) heat transfer in solids, several approaches have been proposed in literature to evaluate designs. For example, the authors in [235] state that high temperatures are a primary failure factor in some electronic devices and therefore choose an objective function that can capture the highest temperatures. This results in a min-max type optimization. However, since the location of the highest temperature can change, the objective function becomes non-differentiable, which is generally a problem for gradient-based optimization algorithms. Therefore, the authors introduce a thermal performance index based on the local flux density and temperature gradient. A very similar strategy is adopted in [88], where it is referred to as the heat potential capacity. Mathematically, this objective function is similar to the compliance often utilized in structural optimization (cf. Section 5.1.1), which facilitates the use of discrete adjoints.

Donoso [68] takes a different approach, focusing on minimizing the quadratic mean temperature gradient within the structure. This can potentially reduce the thermo-mechanical stresses within the structure. Both of these ideas are considered in [150], where a multi-objective optimization approach is chosen, minimizing the mean temperature on the one hand, while simultaneously reducing the variance of the temperature to penalize local deviations of the temperature.

In the context of this thesis, we will consider problems where, given a set of boundary conditions, the temperature profile is optimized to best fit a given distribution. More specifically, we will consider a body with a known temperature at one part of the structure Γ_D and a known heat flux at another part Γ_N. We will consider the inverse problem, where the quality of the design is evaluated based on the deviation from the desired temperature profile on the part with a given heat flux, i.e., where the Neumann boundary condition is applied. This model problem has already been described in [34], in the context of hierarchical optimization using IGA.

Given a target temperature profile $\theta_t : \Gamma_N \to \mathbb{R}$ on the Neumann boundary Γ_N, the objective function therefore reads

$$J = \mathcal{J}(\theta) = \int_{\Gamma_N} (\theta - \theta_t)^2 \, \mathrm{d}S \tag{4.1}$$

$$= \int_{\Psi^{-1}(\Gamma_N)} (\theta - \theta_t)^2 \, J^\Gamma \, \mathrm{d}S \quad . \tag{4.2}$$

In order to compute the gradients using algorithm 3.1, we will additionally require the partial derivatives of the objective function with respect to both the control points as well as the solution field, which yield

$$(\partial_{\boldsymbol{\theta}} J)_i = \frac{\partial \mathcal{J}(\theta)}{\partial \hat{\theta}_i} = 2 \int_{\Psi^{-1}(\Gamma_N)} \mathcal{R}_i \ (\theta - \theta_t)\, J^\Gamma \,\mathrm{d}S \quad , \tag{4.3}$$

$$(\partial_{\mathbf{C}} J)_i = \frac{\partial \mathcal{J}(\theta)}{\partial \mathbf{C}_i} = \int_{\Psi^{-1}(\Gamma_N)} (\theta - \theta_t)^2 \frac{\partial J^\Gamma}{\partial \mathbf{C}_i} \,\mathrm{d}S \quad , \tag{4.4}$$

where the quantity $\partial J^\Gamma / \partial \mathbf{C}_i$ can be evaluated using Equation 3.47. However, to facilitate the optimization, we will not consider cases with moving Neumann boundaries, such that the term evaluates to zero. This also simplifies the assembly of the derivation of the constraint function with respect to the control points $\partial_{\mathbf{C}} \mathbf{c}$, see Equation 3.41.

4.2. Optimization of a Rectangular Microstructured Domain

As a first test case, we consider a rectangular domain with dimensions of 4 m by 2 m, described by a bilinear B-Spline function. This domain is filled with a grid of 10 by 5 crosstiles (see Figure 4.1). The crosstile branches have an adjustable thickness, which is controlled by 4 evaluation points, each located in the center of the faces of the unit square. To facilitate the imposition of boundary conditions and to accurately represent the macro shape at the boundary, a closing layer is introduced into the microstructure, where the tiles form a watertight surface at the top and bottom. The thickness is defined relative to the parametric domain of the deformation function by inserting nodes at positions 0.05 and 0.95.

The individual microtiles consist of 5 patches and the closing patches consist of 4 patches each. Here, the branches of the cross are linear quadratic to achieve G^1 smoothness at the junctions to adjacent tiles, and bilinear at the center and boundary patches. In total, the discritized geometry comprises 330 patches with 4290 control points of which 3141 are "free", non-duplicate control points adjusting for internal patch interfaces. By composing the tiles with the bilinear surface, the patches are of degrees 2 and 4 (from linear and quadratic splines, respectively). The resulting geometry is depicted in Figure 4.1. Additionally, two consecutive uniform h-refinements are applied to obtain more accurate results. This results in a total of 12 071 DOFs in the linear system of equations.

During the analysis, we assume no internal heat sources or sinks and an isotropic environment with a thermal diffusivity of $1\,\mathrm{m^2\,s^{-1}}$ for simplicity[1]. A Dirichlet condition is applied to the bottom surface Γ_D with constant temperature θ of 0 °C. On the top surface Neumann boundary condition are imposed in form of an irregular heat flux along the x-direction, described by

$$\bar{q}_\Gamma = \frac{27\ (4 - x)\ x^2}{256} + 1 \quad \text{in } \left[\mathrm{m\,K\,s^{-1}}\right] \quad . \tag{4.5}$$

[1] All units are the same as for the 3D case. Analogous to the plane-strain assumption in 2D structural mechanics, we will assume an infinitely long z-axis to perform the 2D computations.

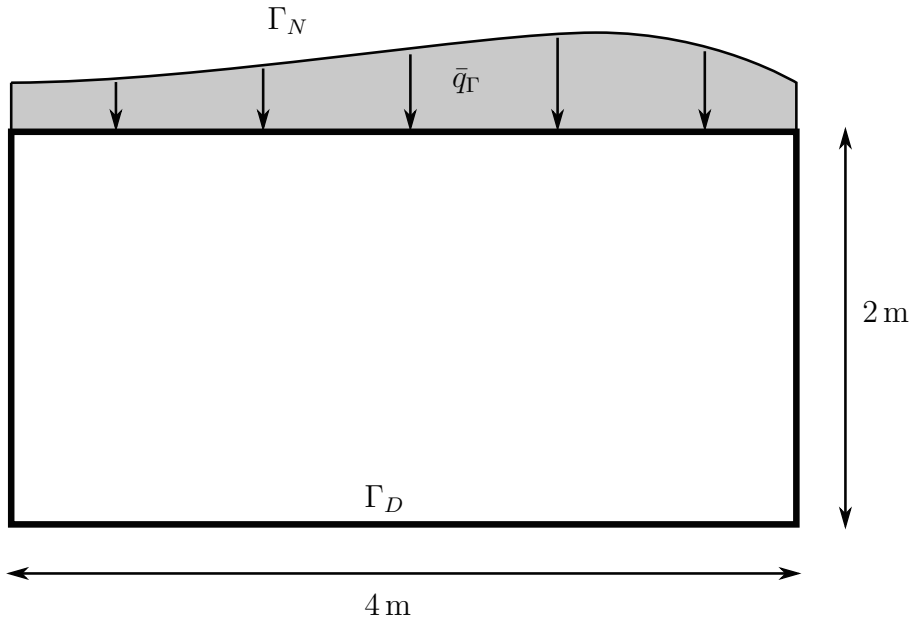


Figure 4.1.: Schematic representation of the macro geometry and boundary conditions of the rectangle test case.

This irregular heat flow is chosen to create an asymmetric temperature distribution on the top surface. The analysis is performed using the framework described in Section 3.3.6.

As in the reference test case, only the local microstructure is modified, i.e. the external geometry remains constant throughout the optimization. To parametrize the local microstructure, we consider a parameter spline that is quadratic with an interior knot along the parametric dimension that maps to the x-axis, and linear with no interior knot along the y-axis, resulting in a control point grid with 4×2 coefficients, or 8 independent design variables.

In the initial configuration, all coefficients are set to the same value, resulting in a uniform thickness distribution. As expected, uniform tiling results in a temperature profile that closely resembles the progression of the applied heat flux, with a pronounced peak in the region of maximum heat flux. The initial temperature of the microstructure is shown in Figure 4.3a. The temperature profile is plotted in Figure 4.2.

This test case is relatively inexpensive due to the use of a (relatively) low polynomial order in the spline representation, as well as the small number of resulting DOFs. As such, we will use it to evaluate the efficiency of different optimization algorithms applied to the local optimization of microstructures. The choice of an appropriate optimization function depends strongly on the objective function and the problem at hand. Many studies are therefore dedicated to finding suitable algorithms for optimization. For example, the authors in [203] compared algorithms in the context of shape optimization of pipe flows.

We will consider both gradient-based and gradient-free algorithms, all provided by NLopt. In the class of gradient-free algorithms, we consider the Constrained Optimization BY Linear Approximation (COBYLA) [183] and Bound Optimization BY Quadratic Approximation (BOBYQA) [184] algorithm, as well as the Nelder-Mead Simplex (NMS) algorithm [163, 37]. While all of these methods allow the imposition of bound constraints, only COBYLA allows the imposition of nonlinear inequality constraints. All three algo-

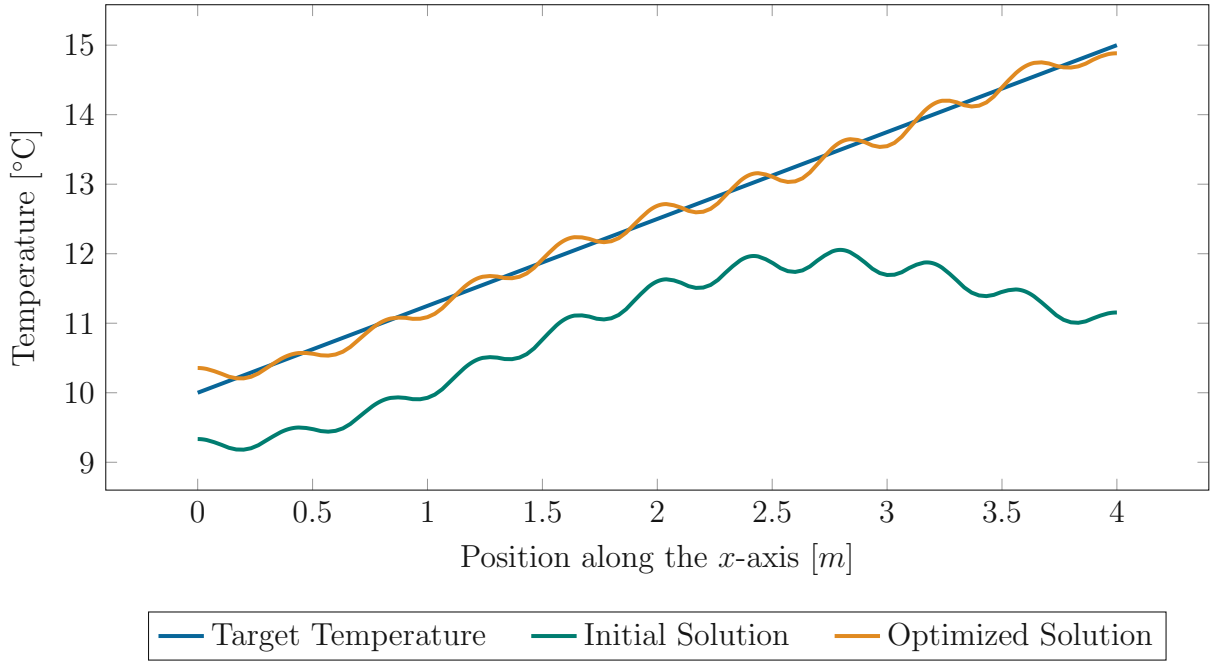
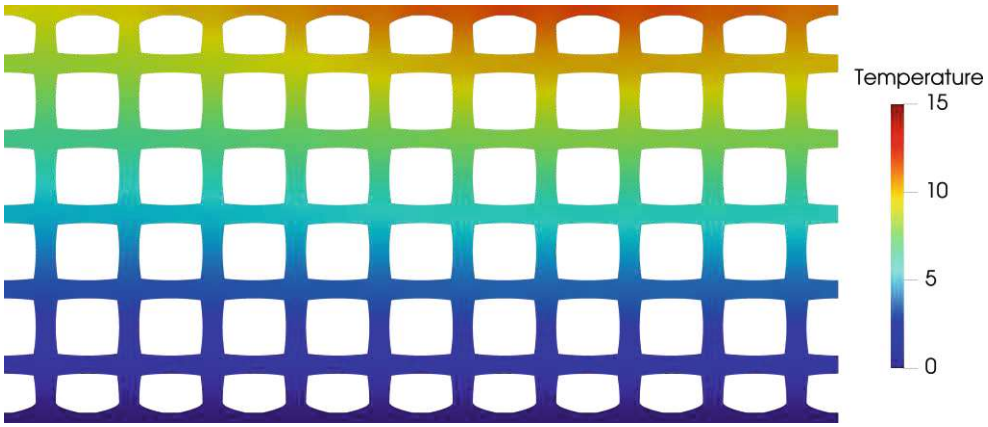


Figure 4.2.: Temperatures on the rectangle surface before and after optimization. The ripples in the temperature distribution show the effect of the microstructure, as the temperature dips, where the branches connect to the surface.

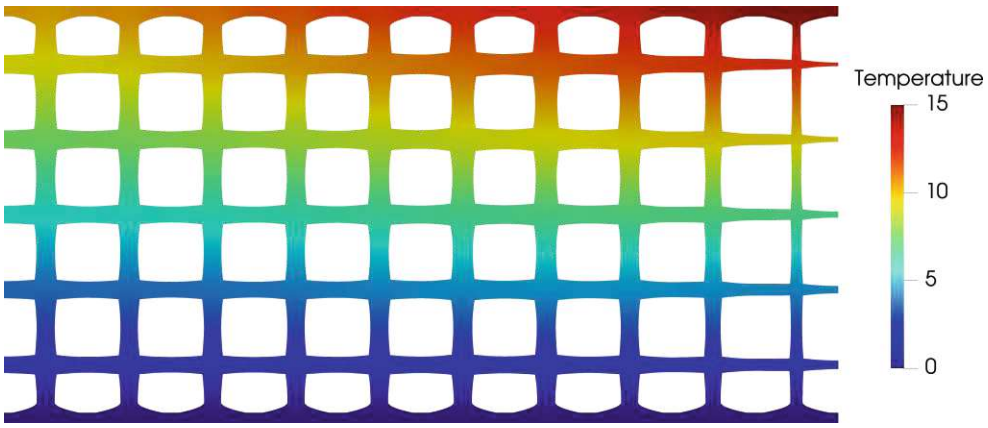
rithms were also considered in [203], where BOBYQA showed the best performance for a large number of design variables, while NMS showed the best convergence near the local minimum.

We will further test the performance of six gradient-based optimization algorithms provided by NLopt, namely Method of Moving Asymptotes (MMA) [209], Sequential Least-Squares Quadratic Programming (SLSQP) [130], Low-storage Broyden–Fletcher–Goldfarb–Shanno (LBFGS) [142], Shifted Limited-Memory Variable-Metric (SLMVM) [222] and a Preconditioned Truncated Newton algorithm [61]. Of these gradient-based algorithms, only MMA and SLSQP support the use of nonlinear inequality constraints in the current implementation; the rest support bound-constrained or unconstrained problems only. Notably, MMA, developed by Svanberg, is one of the de facto standard algorithms in the field of structure optimization and is widely used especially in topology optimization, which is why we included it in this comparison. While a detailed discussion and explanation of the utilized algorithms is beyond the scope of this thesis, the interested reader is referred to [172] for a comprehensive review of numerical optimization, the different approaches and concepts, and the available algorithms and methods.

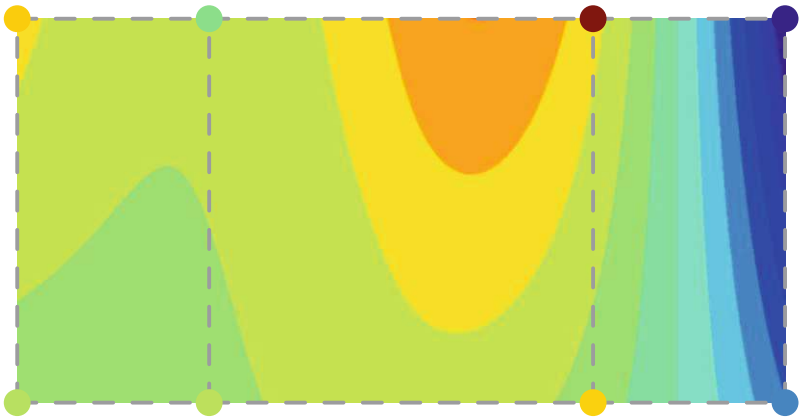
All methods were applied to the model problem by setting a threshold on the value of the objective function, which was determined in a preliminary optimization. Figure 4.4 shows the number of iterations required to reach this threshold. Initial tests have shown that each computation of the gradient (in this particular problem) adds a computational cost equivalent to about 139 % of a normal iteration. This cost is associated with the additional solution of another linear system of equations to solve the Adjoint Equation (3.37), as well as the assembly of the terms specified in Equation (3.39), which is more costly than the assembly of the bilinear form of the forward problem. The adjusted “cost” in terms



(a) Initial Geometry with a constant thickness throughout the microstructure. The overall temperature on the top surface is too low, indicating that the structure in the initial configuration is to thick.



(b) Optimized Geometry. Thinner branches in the upper right corner reduce the heat flux to increase the temperature on the top surface.



(c) Parameter spline associated to the optimized geometry. Blue indicates smaller, red indicates higher values, i.e. thinner or thicker branches in the microstructure. The parameter spline is projected into the deformation function.

Figure 4.3.: Comparison of the initial and optimized geometry obtained with the SLSQP algorithm after 15 iterations.

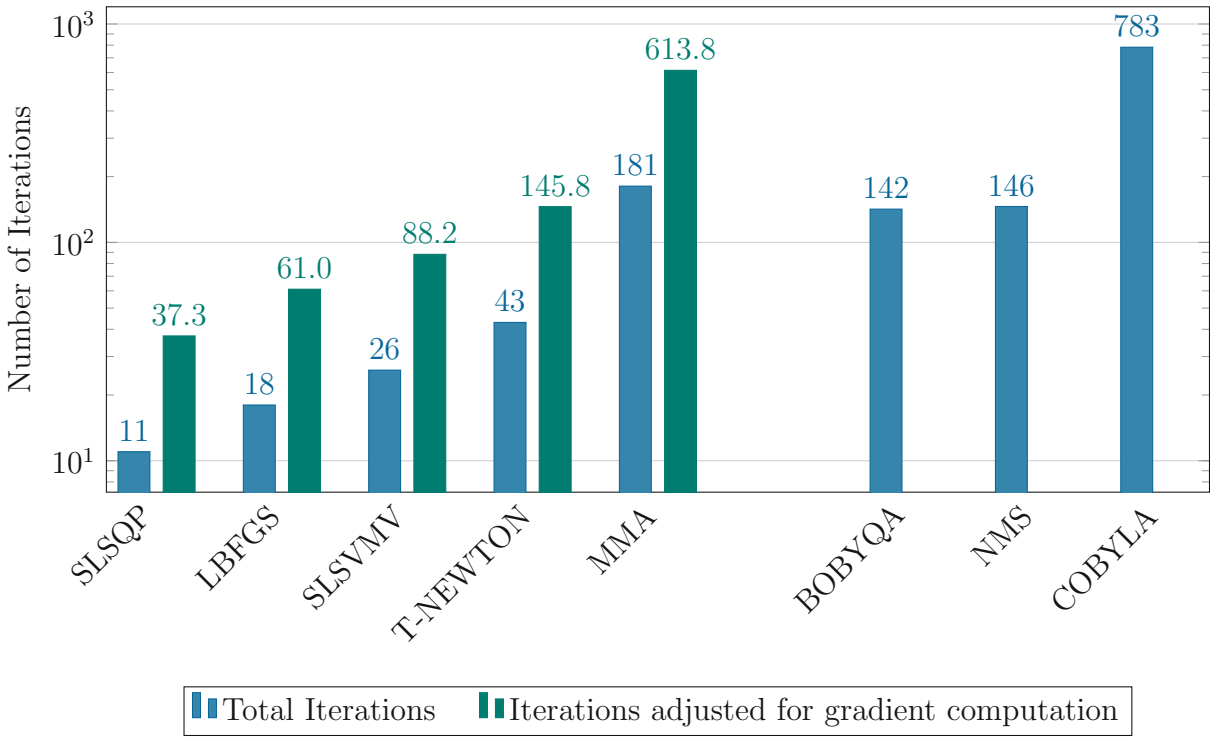


Figure 4.4.: Comparison of different optimization algorithms. The gradient based algorithms also show an adjusted count in green, taking the cost of gradient computations into account. The graph shows the number of required iterations until the objective function falls under a specified threshold determined in preliminary optimizations. The number of iterations is plotted logarithmically.

of total iterations is shown in green for all gradient-based algorithms. Note that the cost of the optimization routines is plotted logarithmically for better visualization. We provided all optimization algorithms with the same initial conditions and only applied bound constraints to the design space. To achieve optimal performance of the provided algorithms, we scaled both the objective function and the design variables to be close to 1, evaluated at the initial conditions.

Figure 4.4 shows that, as expected, gradient-based algorithms generally outperform gradient-free methods, even after accounting for the additional cost of evaluating the gradient. The SLSQP algorithm showed the fastest convergence to the minimum of all tested methods. The same results could be repeated for an extended design space, i.e. with an increase of the available design variables. In this particular test, MMA could not achieve the same rate of convergence. Furthermore, MMA proved to be very sensitive to initial conditions, while SLSQP was more consistent in its rate of convergence.

Among the gradient-free methods considered in this test, both BOBYQA and NMS showed fast convergence, regardless of the initial conditions imposed on the problem. However, of the algorithms tested, only MMA, SLSQP, and COBYLA support nonlinear inequality constraints, which are required when, for example, additional mass constraints are introduced into the optimization problem. These additional requirements on the available constraints become a crucial factor for the appropriate choice of the algorithm, especially for structural optimization in the context of lightweight design.

Of course, the optimal choice of optimization algorithm may vary depending on the specific test case and constraints imposed. However, in our case, SLSQP emerged as the preferred option due to its consistent performance across different initial conditions and its inherent support for nonlinear inequality constraints. Other algorithms, such as MMA, may be better suited for larger optimization problems with significantly higher-dimensional design spaces, with potentially millions of design variables, where this algorithm will ultimately fall short due to its use of dense matrices in its second-order approximation of the objective function [122]. However, our abstraction of the design space effectively reduces the number of design variables so that we end up - at times - with around 1×10^3.

The convergence behavior of the SLSQP algorithm is discussed in more detail below. To create a fair comparison of the optimization algorithms, we initially chose a minimum value of the objective function as the termination criterion. However, to allow the algorithm to iterate closer towards a local minimum, another termination criterion of the form

$$\varepsilon_{\text{abs}} > |J^{k+1} - J^k| \quad , \tag{4.6}$$

is chosen, with an absolute function value tolerance ε_{abs}. Here, the optimization is terminated once the objective function value changes less then 1×10^{-5} in two consecutive iterations.

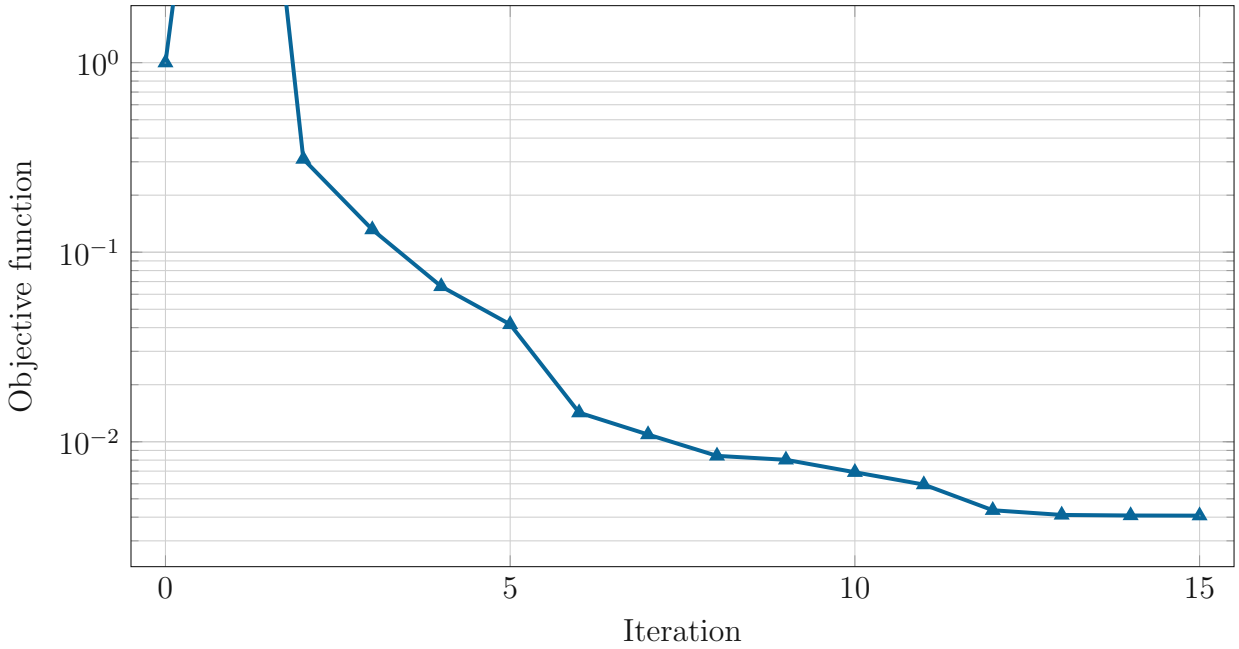
The evolution of the objective function value is plotted in Figure 4.5a. Analyzing the results, we also notice that the bounded constraints were never met during the optimization, so Figure 4.5b also shows the norm of the objective function gradient, or sensitivities. These must tend to 0 in order to satisfy the necessary condition of optimality, defined in Equation (3.32). For ease of interpretation, both curves are plotted logarithmically. In the first iteration, overshooting can be observed, as the initial step size was chosen too large by the optimization algorithm, thereafter, the objective function slowly converges towards the local minimum. A similar trend is also visible in the sensitivity graph. Overall the objective function value was decreased by 99.6 % with respect to the initial configuration. The optimized geometry alongside the optimized parameter spline are shown in Figure 4.3.

4.3. Application to Extrusion Die Geometries

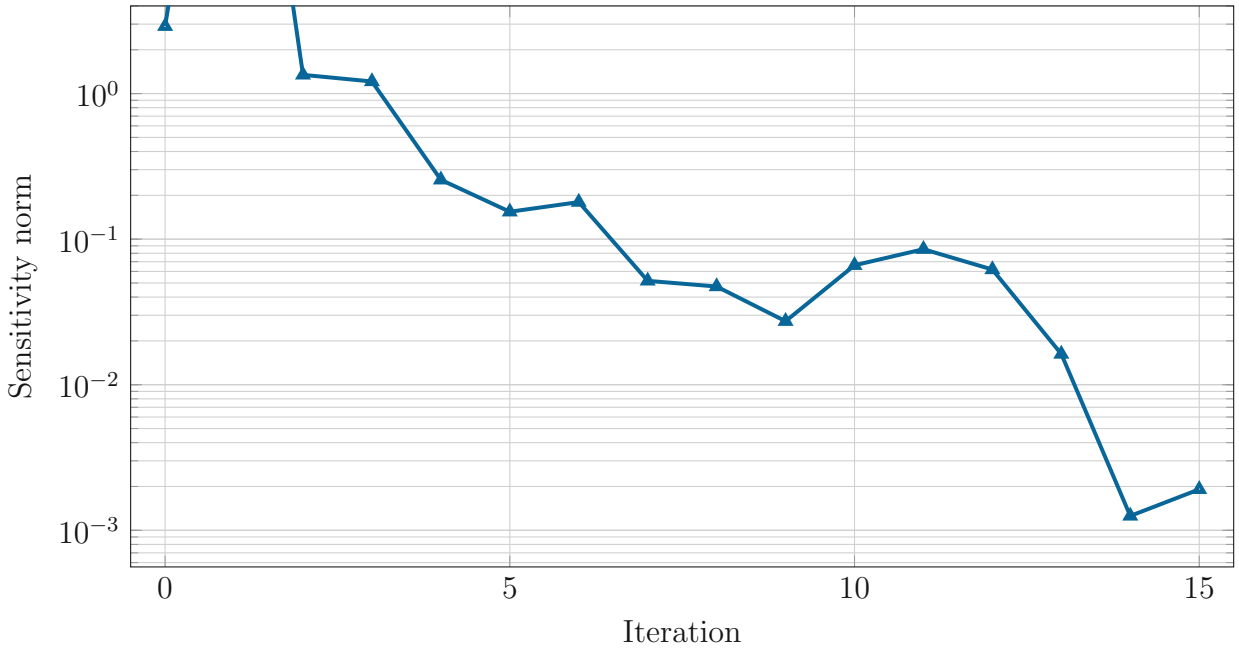
4.3.1. Design Challenges in Extrusion Die Design

Plastic extrusion is a continuous manufacturing process used to produce profiles with a constant cross-section, such as window profiles or pipes. A schematic layout of the primary part of a typical production line is shown in Figure 4.6. First, material in the form of small plastic beads, along with additives and colorants, enters the process through a funnel (or feed hopper). It then passes into a screw, which moves the solid granulate forward toward an extrusion die. With external heating and due to shear heating, the pellets gradually liquefy into a highly viscous melt. This melt then flows through the shaping component of the process, the extrusion die, where it is formed into its final geometry. Therefore, the geometry and structure of the die are critical to the quality of the final product. After exiting the die, the profile is cooled and cut to length. For more details on the extrusion process, the interested reader is referred to [48, 180].

The design of an extrusion die poses numerous challenges to engineers due to the interdependence of various process parameters, the highly nonlinear behavior of the melt [175],



(a) Evolution of the objective function J. In the first iteration the step size is too large leading to a sharp increase in the objective function value.



(b) Norm of the sensitivities $\|\nabla_{\boldsymbol{\chi}} J\|$. No constraints are active during the optimization, hence the norm of the sensitivities must tend towards 0 to fulfill the KKT-condition at the minimum.

Figure 4.5.: Evolution of the objective function using the SLSQP algorithm. Both the sensitivities and the objective function have been scaled by a constant factor so that the initial objective function value is close to 1.

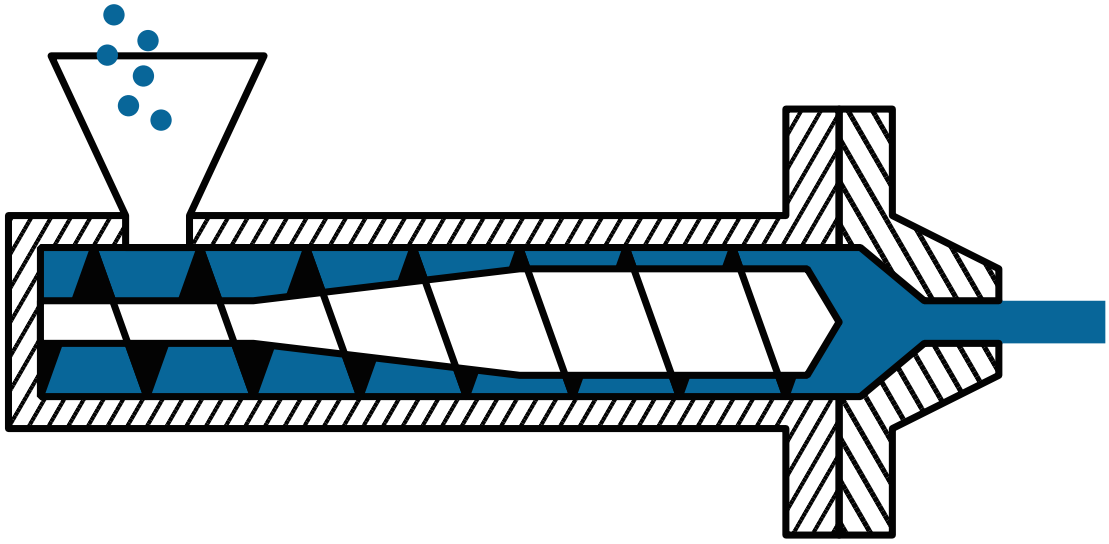


Figure 4.6.: Schematic outline of the extrusion process, based on [115]. New material is fed into the system and pushed through the die via a screw, here shown from left to right. The extrusion die, where the final profile geometry is formed is attached to the extruder on the right hand side.

complex effects such as die swelling, and temperature constraints [112]. At the same time, the list of design objectives is long; For example, the die design must optimize pressure drop and provide a uniform flow rate, the latter generally being considered the most important design factor [171]. Deviations from a uniform velocity profile at the die outlet can lead to residual stresses in the final product, which can cause warpage, i.e., deformation from the desired shape.

The traditional design process often relies on a trial-and-error approach, guided by engineers' domain-specific knowledge and experience. As a result, the process is associated with high costs for prototyping and calibration of manufacturing parameters [112]. Consequently, there has been a growing emphasis on numerical optimization of the extrusion die and its associated parts to become more sustainable and cost effective. In the past, the focus of these optimization approaches has mostly been on the flow channel geometry. Either by considering a specific set of geometric features [79, 232], or by employing spline-based approaches to find an optimal channel geometry [76].

In most of these optimizations, the flow is assumed to be isothermal [79, 76] or with constant wall temperatures [134]. These assumptions regarding temperature are generally justified by stable conditions in a calibrated extrusion process. They significantly reduce the complexity of the analysis and also reduce the simulation problem to the melt itself, disregarding the temperature distribution in the surrounding extrusion die. However, the temperature distribution of the flow within an extrusion die critically affects the material properties of the melt and therefore the velocity profile. In general, a homogeneous temperature in the flow channel would be ideal for rheological design [112, Chapter 8], but local temperature control in combination with flow analysis can also provide new leverage for optimization. By changing the temperature, the melt can be made more or less viscous, which, in addition to the shape of the flow channel, is an additional parameter in achieving homogeneous velocities at the extrusion die outlet. In computer simulations,

this idea translates into modifying the temperature boundary conditions and optimizing the temperature distribution within the flow channel.

In [135], Lebaal, Schmidt, and Puissant employed this idea, optimizing the temperature of a given channel with a coat-hanger design, where the die cooled the melt to counteract shear-induced energy dissipation. However, no consideration has been given to how this specific temperature profile is achieved. In practice, heating of the extrusion die, e.g. for thermoplastics, is usually done by means of heating bands attached to the outside of the extrusion die. These generally provide only global temperature control.

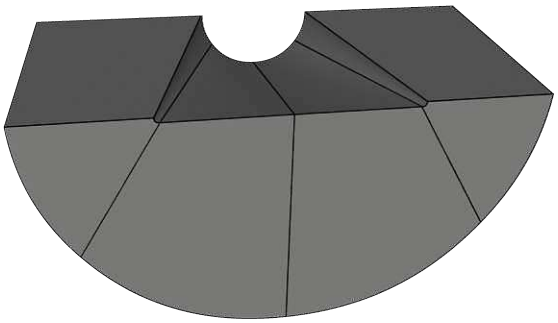
The die itself is usually made of metal by wire erosion or milling. In these dies, the only way to control temperature is through holes or active fluid-based temperature control. However, with recent advances in additive manufacturing techniques, 3D metal printing can become a viable alternative to traditional extrusion die manufacturing. By using carefully designed internal structures as opposed to massive solids, passive local heat control can be achieved despite only global control on the heater band. With this in mind, the use of microstructures for extrusion die design was explored in [238], which will be analyzed in the following section.

It's important to stress that this test case serves primarily as a conceptual exploration, demonstrating the potential application of microstructure geometries in more complex outer geometries, with a focus on extrusion die designs. The ability of the microstructure to withstand high pressure requires thorough computational analysis to assess the feasibility of using microstructured extrusion dies in practical applications. These considerations must also take the interactions between the temperatures in the die and the melt into account. In spite of this uncertainty, there are several potential advantages to the use of such dies. Their hollow structure offers ease of handling and material cost savings, while the cavities provide space for attaching sensors for testing and maintenance. In addition, the incorporation of more complex designs could introduce tubes through the microstructure for active temperature control.

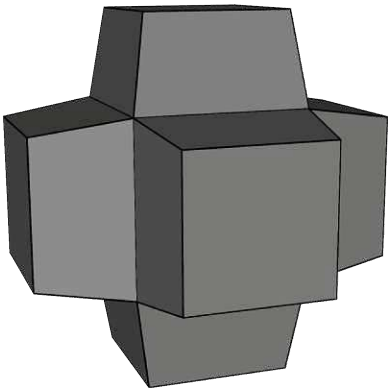
4.3.2. Optimization of Heat Transfer in Extrusion Dies

Building on the results of Section 4.2, the same principles are applied to a more complex 3D geometry of an extrusion die. It is assumed that the channel geometry is already optimized and therefore only the local structure is considered during optimization. The desired cross section of the extrusion die is a slit profile with circular edges as shown in Figure 4.7a. The extrusion die is cylindrical with an outside diameter of 0.2 m and a length of 0.1 m. The slit in the center of the die has a total width of 0.1 m and a height of 0.004 m. The diameter of the inlet is 0.05 m where the melt leaves the extruder. The die is symmetrical both along the slot axis and about the center of the die, so only a quarter of the extruder is included in the calculations. This test case is taken from [238] with minor modifications. The outer geometry is constructed by linear interpolation between the inlet geometry from Figure 4.8a and the outlet geometry from Figure 4.8b. The resulting B-Spline approximation of the die is used as deformation function for microstructure construction. It has degrees $\mathbf{p} = [2, 1, 1]$ and 1 knot along the quadratic angular axis at position 0.4, connecting the round part of the slit profile with the straight edge.

The microtile is constructed using 7 patches connecting the center cube to the faces of the unit cube. Each branch has a distinct thickness, which is obtained using the evaluation points at the centers of the faces of the unit cube. An exemplary microtile is shown in

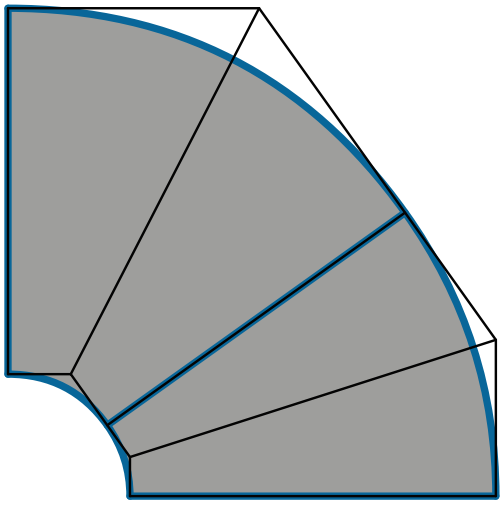


(a) Half of the Extrusion die, showing the channel geometry in the middle. B-Spline with degrees $\mathbf{p} = [2, 1, 1]$

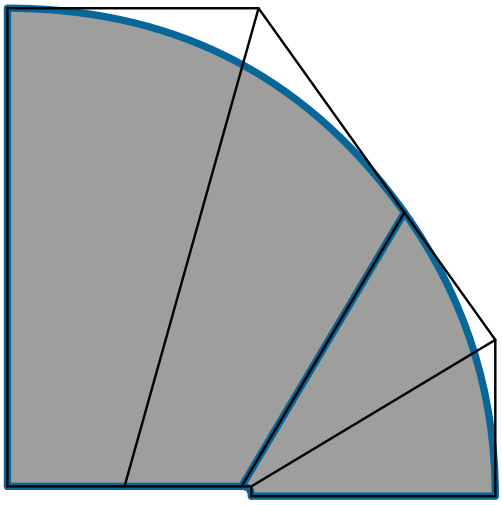


(b) Microtile comprised of 7 trilinear patches. The center cuboid's width is 1.2 times the average branch thickness

Figure 4.7.: Deformation function and microtile parametrization.



(a) Inlet geometry of the extrusion die, where the round channel entry is attached to the extruder.



(b) Outlet geometry of the extrusion die, showing the slit-geometry of the final profile.

Figure 4.8.: Spline representation with control grid of the inlet and outlet side of the extrusion die. The inlet and outlet geometry are linearly interpolated to obtain the geometry of the deformation function.

Figure 4.7b with a constant branch width of 0.4. The width of the central cube is 1.2 times the average branch thickness. These two components are used to construct a microstructure with a total of $12 \times 7 \times 9$ tiles.

This microstructure serves as a basis of optimization. The local thickness distribution is described by a B-Spline with degrees $\mathbf{p} = [2, 1, 1]$ defined using a control grid of $4 \times 2 \times 2$ coefficients leading to a total of 16 design variables. This relatively small number of control parameters was chosen to ensure maximum concordance with the test case from [238]. The resulting microstructure consists of 7452 patches with degree 4 along every parametric axis, defined by 672 025 non-duplicate control points. Subtracting coefficients at the Dirichlet boundary gives a total of 656 220 DOFs for the heat transfer problem. Based on the

previous results, the SLSQP algorithm is employed, implemented via the `scipy` Python library, with a function tolerance ε_{abs} of 1×10^{-6}.

In order to obtain results that are more physical and closer to reality, we have chosen material parameters that are very similar to those of stainless steel. The material parameters used to simulate the heat transfer problem are listed in Table 4.1.

Parameter	Symbol	Value	Unit
Thermal Conductivity	k	20	$\mathrm{W\,m^{-1}\,K^{-1}}$
Specific Heat Capacity	c	420	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Density	ρ	7850	$\mathrm{kg\,m^{-3}}$
Thermal Diffusivity	λ	6.066×10^{-6}	$\mathrm{J\,m^{-3}\,K}$

Table 4.1.: Material parameters used for the extrusion die in optimization.

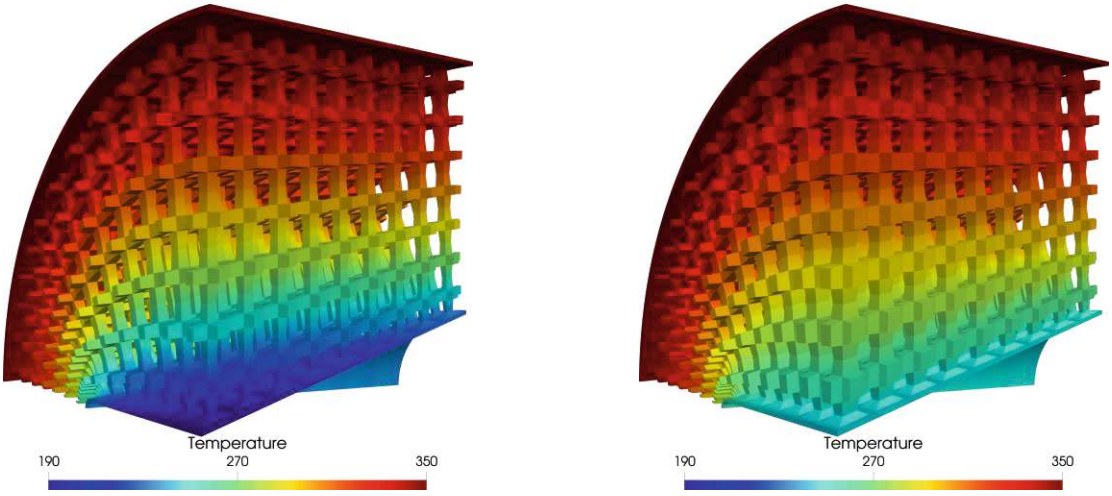
Evidently, the true temperature distribution in the die can only be calculated in a coupled simulation that includes the melt and (to some extent) the ambient air. However, for the purposes of this demonstration, we will assume that the outside of the die, where the heater band is attached, has a constant temperature of 350 °C. At the channel wall, on the inside of the die, we assume a time-invariant heat flux. We will now consider two cases, one where the wall temperature, and thus the melt temperature, is constant with a locally dependent heat flux, and another where the heat flux is constant but the target temperature varies. These two cases correspond to two different concepts. In the first case, the melt is assumed to have heated up irregularly due to pressure drop and mechanical shear, as analyzed in [135]. This is modeled here by an irregular heat flux into the melt. In the second case, the heat flux is assumed to be constant at the boundary, but the target temperature is chosen to increase slightly at the slit boundary. This increase in temperature is thought to locally reduce the viscosity and thus counteract the high flow resistance as the melt sticks to the sidewall.

In the temperature optimization test case presented in [135], the melt heats up toward the boundary due to increased mechanical shear and energy dissipation. This temperature increase also causes a reduction of the viscosity, resulting in higher velocities near the boundary. To counteract this effect, the temperature of the extrusion die was increased towards the center during optimization in order to balance the temperature distribution of the melt. From the perspective of the die, this temperature difference can be represented as a higher heat flux. To model this (qualitatively) in the optimization problem, the scaled heat flux $\bar{q}_\Gamma$ at the channel wall in this example is set to

$$\bar{q}_\Gamma(x, y, z) = -\frac{4500}{\rho c} \cdot \left(1 + \frac{1}{2}\sqrt{(1 - 10z)(1 - 20x)}\right) \quad \text{in } \left[\mathrm{m\,K\,s^{-1}}\right] \tag{4.7}$$

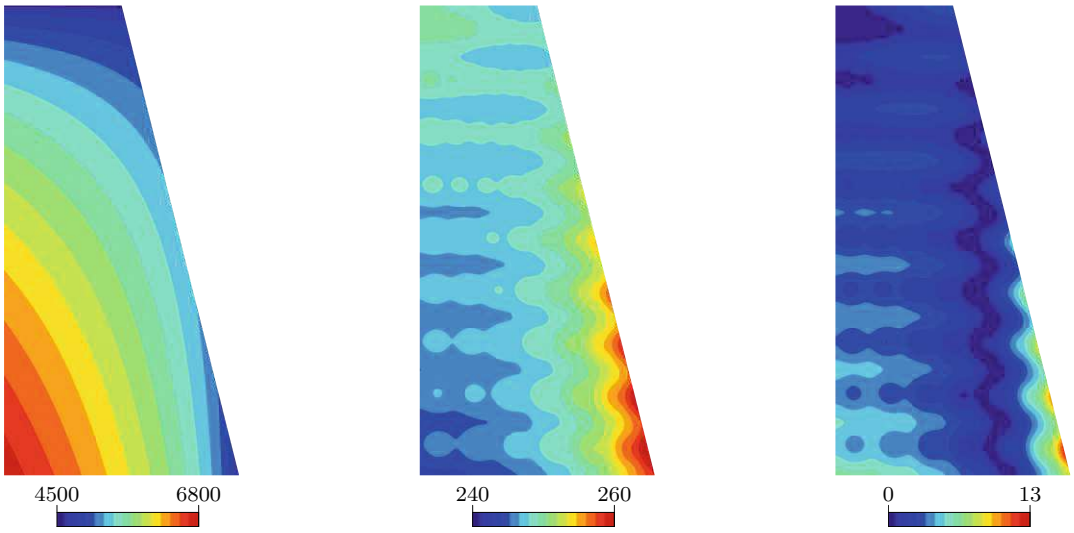
Equation (4.7) applies an inward facing flux of $4500\,\mathrm{W\,m^{-2}}$ to the melt (i.e., cooling the die), which increases by an additional 50 % toward the center, where the lower temperatures are expected. The flux is shown in Figure 4.10a.

The target temperature is set to $\theta_t = 250\,°\mathrm{C}$. Initially all cross tiles are set to a constant width. The initial and optimized geometry along with their temperature distributions are shown in Figure 4.9. In the initial configuration, the lowest temperature is 60 °C below the target temperature, indicating that the overall structure restricts the heat flow too much. As a result, the average thickness of the cross tiles increases, adding approximately



(a) Initial geometry and temperature **(b)** Optimized geometry and temperature

Figure 4.9.: Geometries and temperature fields associated to test case varying heat flux and constant target temperature.



(a) Magnitude Heat flux **(b)** Optimized Temperature **(c)** Error $|\theta - \theta_t|$

Figure 4.10.: Irregular heat flux at the channel, along with the optimized temperature and error plotted on the channel wall. The channel is projected onto the xz-plane. The target temperature is $\theta_t = 250\,°\mathrm{C}$

18.5 % to the total volume of the microstructure. This effect is most pronounced in the center of the die, where the heat flow is highest. The temperature distribution of the optimized geometry as well as the difference between the actual temperature and the target temperature θ_t is shown in Figure 4.10. The ripples in the channel wall temperature are attributed to the microtile branches adhering to the surface.

The temperature deviates the furthest from the target temperature θ_t along the edge of the slit profile. This is explained by the distance to the heat source, which is smallest at the outer part of the geometry. As a result, although the average branch thickness increases

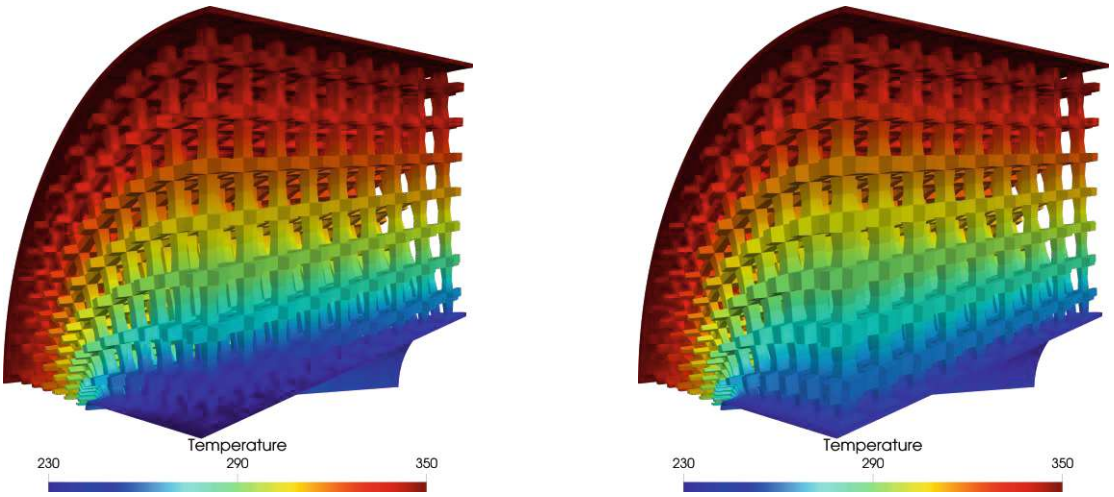
significantly, the branches of the tiles attached to the narrow radius become thinner with respect to the initial geometry.

In this first instance, it was inferred that mechanical friction and the associated heat lead to a velocity increase at the boundary. However, in the case of a wide flow channel, this may not necessarily be the case, especially if the high viscosity of the melt is the dominant perturbation of the velocity profile. In this case, it may be beneficial to increase the temperature at the narrow boundary to reduce the viscosity and improve the velocity profile. Assuming a constant heat flux $|q_\Gamma| = 4500\,\mathrm{W\,m^{-2}}$ and without changing other parameters of the simulation, the following target temperature profile is set as an optimization objective for the second test case

$$\theta_t(x, y, z) = 250 + 5 \cdot \max\left(\frac{x - 0.03}{0.02},\ 0\right)(1 - 10z) \quad \text{in } [^\circ\mathrm{C}] \quad . \tag{4.8}$$

This temperature profile is shown in Figure 4.12a. The temperature is constant with a value of 250 °C in most of the flow channel and increases linearly along the edges and towards the outlet by an additional 5 °C.

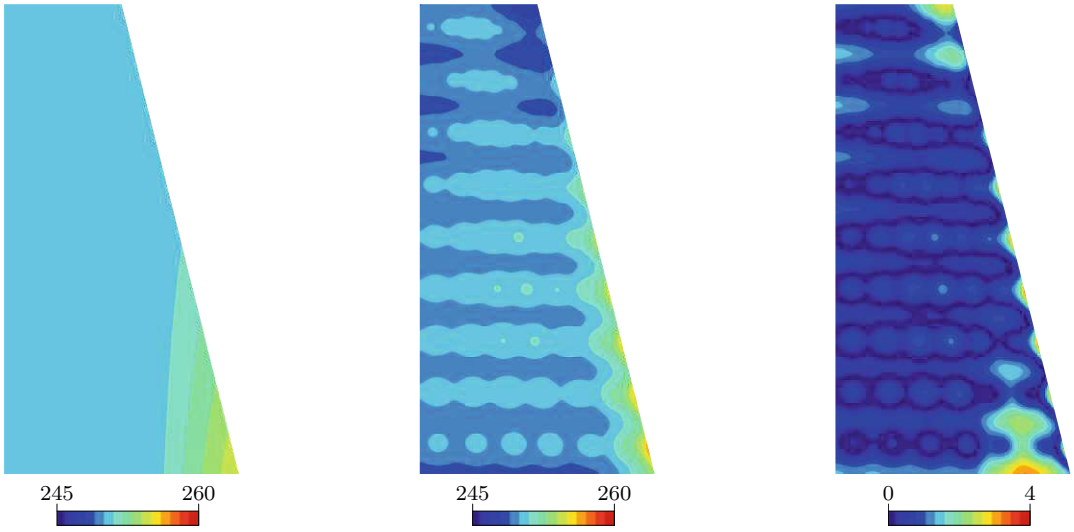
The initial geometry identical to the first application, its associated temperature field, and the optimized geometry and temperature are shown in Figure 4.11. As in the previous example, the overall temperature in the initial configuration is too low and thicker microtiles are required to increase the total heat flux through the structure. In the rest of the structure, the branches tend to thin, even along the narrow channel edge, where a higher temperature is now sought. The temperature field at the flow channel, as well as the deviation from the target temperature distribution are shown in Figure 4.12.



(a) Initial geometry and temperature **(b)** Optimized geometry and temperature

Figure 4.11.: Geometries and temperature fields associated to test case with increased temperatures towards the edge of the channel wall.

Although the overall temperature distribution matches the target temperature profile closely, local outliers can be observed in the optimized structure. How pronounced these outliers become, as well as how they affect convergence to the optimal temperature profile, depends on the objective function. One strategy to eliminate these outliers is to divide the flow channel into different sections and then evaluate the performance on each section. This penalizes local outliers, but taking the largest outlier as the performance metric



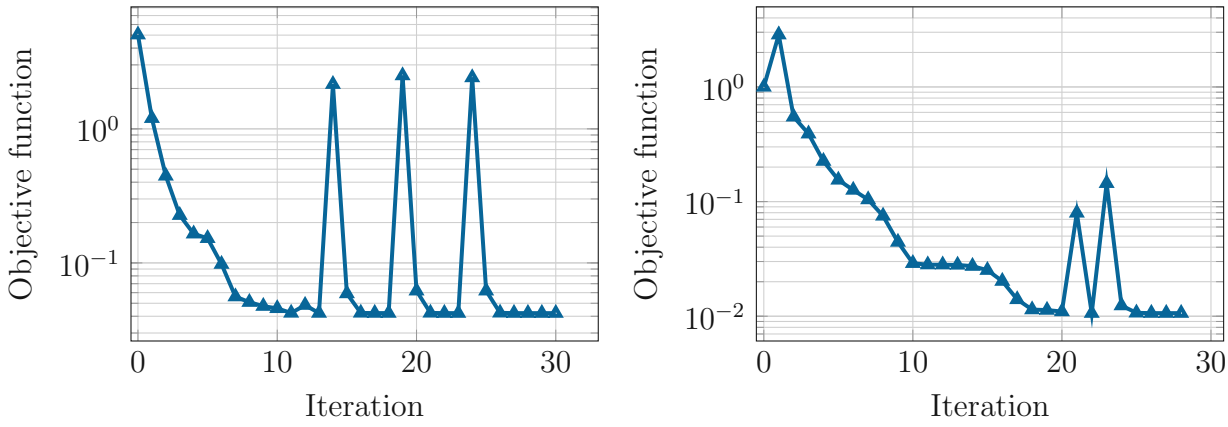
(a) Target temperature **(b)** Optimized temperature **(c)** Error $|\theta - \theta_t|$

Figure 4.12.: Target temperature distribution, along with the optimized temperature and error plotted on the channel wall. The channel is projected onto the xz-plane.

makes the objective function non-differentiable, making the use of gradient-based algorithms impractical. Another approach is to utilize a different exponent in the integral, which can prioritize outliers over average performance. The aforementioned strategies can be integrated into the same framework, although the convergence of the optimizer may be constrained by the non-differentiability of some of the objective functions. Ultimately, the appropriate choice of the objective function is a fundamental design decision.

The evolutions of the objective function over the course of the optimization iterations are shown in Figure 4.13. Initially, both optimization routines exhibit fast convergence towards their respective local minima. This behavior coincides with a significant decrease in the norm of the sensitivities by 3 to 4 orders of magnitude. However, as they approach the minimum, significant spikes occur despite the absence of constraint violations. It is observed that in both cases the step size starts to increase as the optimization progresses towards the minimum, which may result in overshooting.

In the case of a constant target temperature, the objective function could be reduced by approximately 99.16 % with respect to the initial configuration. 31 iterations and 15 gradient evaluations were required to reach this local minimum. Similarly the second test case showed an improvement of 99.63 %, with 29 iterations and 21 gradient evaluations. As the gradient computations are significantly more expensive than the forward simulation, the optimization driver only determines the gradient, once a better solution has been found.



(a) Evolution of the objective function J for the first test case, with constant target temperature.

(b) Evolution of the objective function J for the second test case, with modified target temperature. First spike is due to inadequate initial step size.

Figure 4.13.: Evolution of the objective function over the iterations of the optimization. In both cases, the design variables as well as the objective function values have been scaled to ensure optimal performance of the optimization algorithm.

5 Application to Structural Optimization Problems

Design optimization under structural and mechanical considerations is the main driving force for numerical shape and topology optimization. Its applications are manifold and it is used in almost every field of engineering. As a result, there is extensive literature on the subject, for example [99], which focuses more on providing a general understanding and briefly discusses various methods and tools, or [11, 206], which provide details about the mathematical foundations.

In this chapter, we demonstrate the application of the design paradigm in the context of structural optimization. After first introducing the necessary preliminaries, including microtile geometry, objective functions and constraints, along with their derivatives, we will apply the presented method to a benchmark test case and compare it to classical approaches. In the second part, we show microstructures on curved geometries and compare different optimization strategies combining the ideas presented in the previous chapters.

5.1. Preliminaries

Throughout this chapter we will use the microtile shown in Figure 5.1, which consists of a vertical box with diagonally crossed trusses in the center. The microtile consists of 20 bilinear Bézier splines. Its truss thickness is parametrized per cell, with an evaluation point at the center of the unit cube, resulting in one design parameter per cell. The vertical and horizontal patches are only half as wide because they are attached to their respective adjacent cells.

5.1.1. On the Choice of the Objective Function

In the field of structural mechanics, particularly in lightweight design, the primary objective of numerical design optimization is to minimize deformation under a given load while maintaining the lowest possible overall mass of the structure. In practice, these two design specifications are inherently incompatible, as a (geometrical) stiffness is typically achieved by incorporating additional material. Therefore, defining an optimal balance between these two objectives is an essential aspect of the design strategy and depends on the specific application.

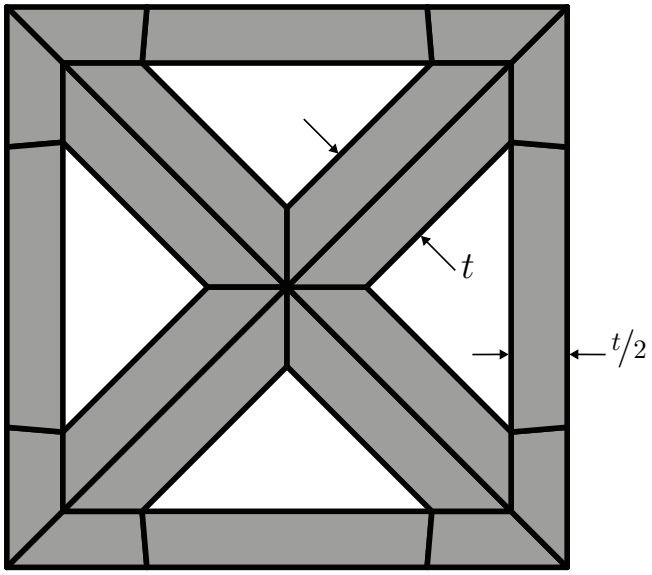


Figure 5.1.: Lattice microtile parametrization, used throughout this chapter where the thickness t is evaluated at the center of the tile within the parametric domain of the deformation function.

Over time, various strategies have evolved to attain these objectives. These generally follow one of two general paths. In the first strategy, the optimization is stress-based, where the design for a given loading scenario is identified by minimizing the structural weight while not exceeding a maximum permissible stress for the given material. In this strategy, the weight serves as the objective function and some measure of the stresses serves as constraint, e.g., von Mises stresses. However, this approach presents several numerical challenges, especially if a strict limit is imposed on the maximum stress. The specific location may change under design variations, making the objective function non-differentiable. In contrast, the second approach is stiffness-based, where the weight is constrained and the stiffness of the structure is maximized. A comparison between these two approaches in the context of lattice structures was made in [82], where a p-mean of Hill's yield criterion was used in the stress-based approach. The authors concluded that although the stress-based approach inhibited yield more rigorously, the resulting structures were found to be less stiff when compared to their counterparts of the same weight. Both strategies can be incorporated into the existing framework; however, this work will focus on the second strategy exclusively.

Although other measures of overall stiffness have been explored to quantify the deformability of a given structure, see Appendix B, the stiffness of a structure in response to external forces is typically quantified by its compliance. This measure is widely used particularly in the context of structural topology and shape optimization [11, 230], also in the context of lattice structures, see e.g., [212]. Compliance can be interpreted as the deformation energy, with its formulation in the context of linear elasticity outlined as [160]

$$\mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) = \int_{\Omega} \mathbf{f} \cdot \mathbf{u} \, \mathrm{d}V + \int_{\Gamma_N} \mathbf{h} \cdot \mathbf{u} \, \mathrm{d}S \quad . \tag{5.1}$$

The popularity of this objective functional is due to its compatibility with the adjoint-based sensitivity analysis discussed in section 3.4. In the following, we will use the vari-

ational form of the elasticity equation as well as the associated derivatives, which are described in detail in Appendix A.1. Using these derived expressions, the objective function from Equation (5.1) can be rewritten in the form

$$\mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) = L_{\mathbf{u}}(\mathbf{u}) \quad . \tag{5.2}$$

Building on the linear form $L_{\mathbf{u}}$ and bilinear form $a_{\mathbf{u}}$ established in Appendix A, we can build a linear system of equations in the form (cf. Equations (A.4) to (A.7))

$$\mathbf{K}\mathbf{u} = \mathbf{F} \quad . \tag{5.3}$$

As the test and trial functions build on the same function space, it is possible to rewrite Equation (5.2) using only components of the linear system of equations, which have already been computed during analysis [223]. This demonstrated in more detail in Appendix A.2. The objective function then reads

$$\mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) = \mathbf{F}^T\mathbf{u} \quad . \tag{5.4}$$

This expression for compliance further facilitates the differentiation of the objective function required for sensitivity analysis, which can be computed using

$$\partial_{\mathbf{C}}\mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) = \frac{\partial L_{\mathbf{u}}(\mathbf{u})}{\partial \mathbf{C}_i} \quad , \tag{5.5}$$

$$\partial_{\mathbf{u}}\mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) = \mathbf{F}^T \quad . \tag{5.6}$$

Substituting Equation (5.6) into Equation (3.37) shows that, in this particular setup, the Lagrange multipliers are related to the displacement coefficients of the solution field with $\boldsymbol{\lambda} = -\mathbf{u}$, due to the symmetry of the stiffness matrix. This so-called *self-adjoint* [7] behavior is a key factor in the popularity of the objective function in the context of design optimization. It significantly reduces the computational cost associated with determining the sensitivities of the parametrization, obviating the necessity to solve an additional system of linear equations. These can be computed by multiplication alone, using

$$\mathrm{d}_{\boldsymbol{\chi}}J = \left(\partial_{\mathbf{C}}J - \mathbf{u}^T\partial_{\mathbf{C}}\mathbf{c}\right)\ \partial_{\boldsymbol{\chi}}\mathbf{C} \quad . \tag{5.7}$$

$\partial L_{\mathbf{u}}(\mathbf{u})/\partial \mathbf{C}_i$ and $\partial_{\mathbf{C}}\mathbf{c}$ are further detailed in Appendix A.1 in Equations (A.10d) and Equations (A.10a) to (A.10c).

5.1.2. Constraints

As discussed in the previous section, the total mass of the structure is constraint during optimization. The mass of the microstructure $\boldsymbol{\mathcal{M}}$ can be computed using

$$m_{\boldsymbol{\mathcal{M}}} = \int_{\Omega} \rho \,\mathrm{d}V = \int_{\Psi^{-1}\,\circ\,\tilde{\Omega}} \rho\ \det(\mathbf{J}_{\Psi})\,\mathrm{d}V \quad , \tag{5.8}$$

and the derivatives of the constraint function can be determined using the identities in Equation 3.42, which yield

$$\frac{\partial m_{\boldsymbol{\mathcal{M}}}}{\partial \mathbf{C}_i} = \int_{\Psi^{-1}\,\circ\,\tilde{\Omega}} \rho\ \det(\mathbf{J}_{\Psi})\ \mathrm{tr}\left(\mathbf{J}_{\Psi}^{-1}\frac{\partial \mathbf{J}_{\Psi}}{\partial \mathbf{C}_i}\right)\,\mathrm{d}V \quad . \tag{5.9}$$

These are later multiplied with $\partial_{\boldsymbol{\chi}}\mathbf{C}$ in order to obtain the parameter sensitivities.

However, in practical applications, especially for constant densities ρ, the volume constraint is often given in terms of a global volumetric density. The volumetric density compares the weight/volume of the microstructure to the dense structure, represented by the deformation function $\mathcal{T}$. It therefore reads

$$\tilde{\rho} = \frac{\int_{\mathcal{M}} \mathrm{d}V}{\int_{\mathcal{T}} \mathrm{d}V} \quad , \tag{5.10}$$

where the denominator remains constant only if the deformation function does not change during optimization.

5.2. Optimization of a Cantilever Design

The first test case demonstrates the performance of the proposed method on a cantilever beam. The cantilever beam is one of the de facto standard benchmark tests in structural optimization, especially in the field of topology optimization [30]. Consequently, it has also served as a benchmark in the context of lattice structure design, as demonstrated for example in [38, 138]. In fact, in the review [181] it is used to evaluate different methods, but comparison has proven difficult because there is no general agreement on aspect ratios and target volume densities. In this section, the test case design is based on the numerical example presented in [229], using the microtile geometry shown in Figure 5.1.

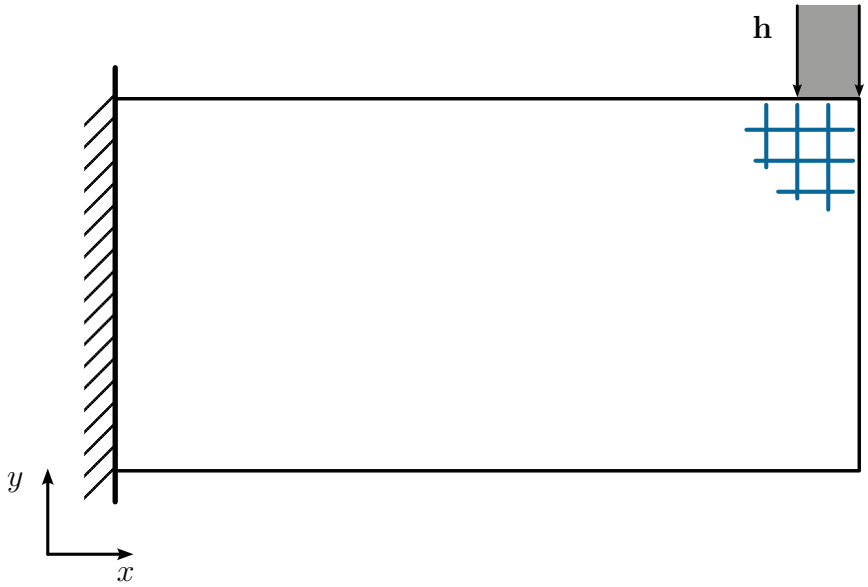


Figure 5.2.: Schematic outline of the cantilever benchmark test case. The force is applied to the two outermost microtiles in the structure, the left side is clamped.

The schematic outline of the test case is shown in Figure 5.2. A rectangle with dimensions $2\,\mathrm{m} \times 1\,\mathrm{m}$ is filled with a regular grid of 24×12 tiles. The structure is clamped on the left side with both x and y displacements set to 0. Unlike the example in [229], where a point force is applied, the force is evenly distributed over two columns of tiles on the top surface of the rectangle in the form of a unit load of 1 Pa. The material parameters

are also taken from [229] and summarized in the Table 5.1. Note that arguably the only "decisive" parameter that influences the outcome of the optimization is Poisson's Ratio, since Young's Modulus, structure size, and applied force scale the compliance linearly, cf. Equation (5.4). Consequently, the optimization would yield the same results if a stiffer material with a Poisson's Ratio of 0.3, such as steel, was used[1].

Parameter	Symbol	Value	Unit
Young's Modulus	E	1	Pa
Poisson's Ratio	ν	0.3	-
First Lamé constant	λ	0.5769	Pa
Second Lamé constant (shear modulus)	μ	0.3846	Pa

Table 5.1.: Fictitious material used for the cantilever optimization, taken from [229].

The macro geometry is constructed using a bilinear Bézier spline. Although in this particular example the tiles could have been periodically stacked using only linear transformations, the microstructure is constructed using functional composition, resulting in biquadratic microtile patches in the overall microstructure. The entire geometry is described by 5760 patches described by 25 561 non-duplicate control points, for a total of 50 976 DOFs.

The total mass of the structure is constraint, aiming for a target volume fraction of solid material $\tilde{\rho}_t$ of 30 %. This is imposed using an additional inequality constraint

$$\mathbf{k}_I(\boldsymbol{\chi}) = \tilde{\rho} - \tilde{\rho}_t \leq 0 \quad . \tag{5.11}$$

The initial thickness is chosen, such that the inequality constraints is active, i.e. $\tilde{\rho}_{\text{initial}} = 0.3$. In the benchmark test case, the individual cell thicknesses are specified individually. In order to achieve this without changing the framework, a zero-order spline is adopted that has the same internal nodes as the deformation function (with tiling).

Employing the optimization framework detailed in section 3.3.6, the SLSQP-algorithm from the `scipy` Python library is used with a function tolerance of ε_{abs} of 1×10^{-4} as termination criterion, see Equation (4.6). The evolution of the objective function over the course of the optimization is plotted in Figure 5.3.

Figure 5.4 shows the displacement fields before and after optimization. Comparing the two designs, the maximum displacement (located at the top right) is reduced by 56.22 %, which also indicates a higher stiffness of the optimized design.

The compliance is reduced by 55.22 % after a total of 35 optimization iterations. Figure 5.5 show snapshots taken at different iterations, which show the evolution of the design. Initially, the material shifts to the top and bottom of the structure, increasing the bending stiffness in the leftmost part of the structure. After only five iterations, the final design begins to emerge, and the typical design with two diagonals crystallizes toward the end of the optimization. This can be seen more clearly in the parameter spline plot projected onto the deformation function. Dark blue areas appear where the design variable is set to the minimum allowed thickness.

In the original test case from [229], the structure is further modified by penalizing intermediate values. On the one hand, this resulted in a more refined structural outline - or

[1] With adjusted convergence scaling and convergence criteria.

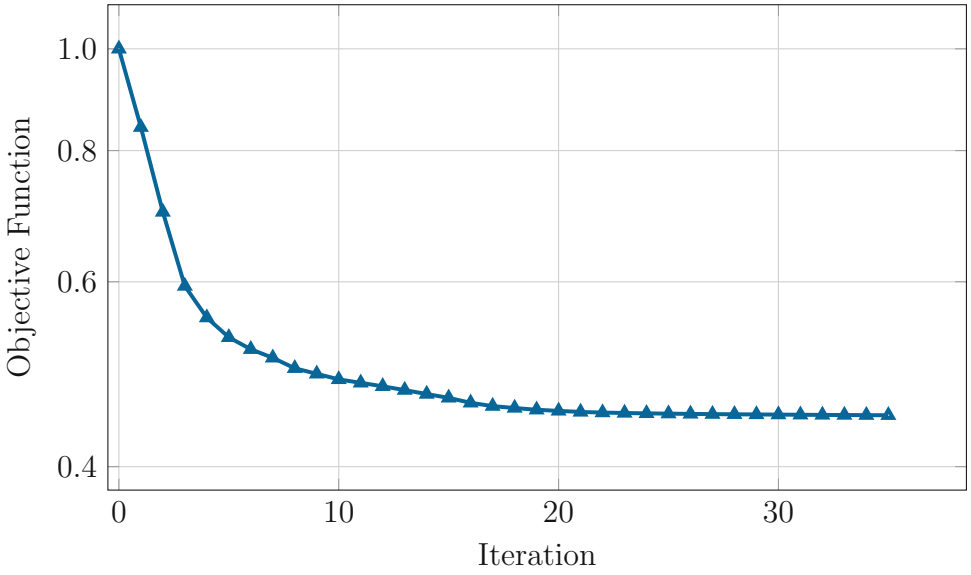
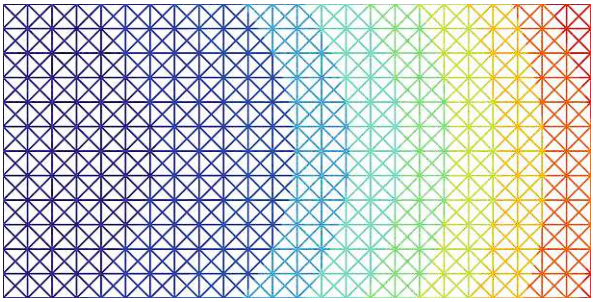
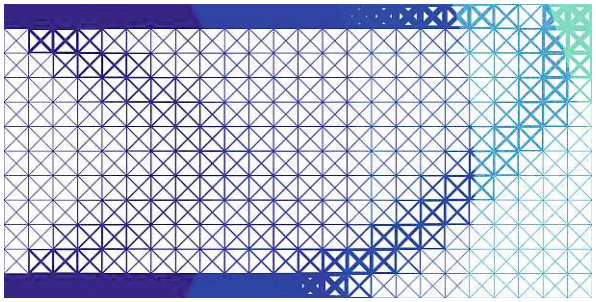


Figure 5.3.: Evolution of the objective function over the course of the iterations, plotted logarithmically. The objective function is scaled with a constant factor, such that the objective function in the initial configuration evaluates to 1.



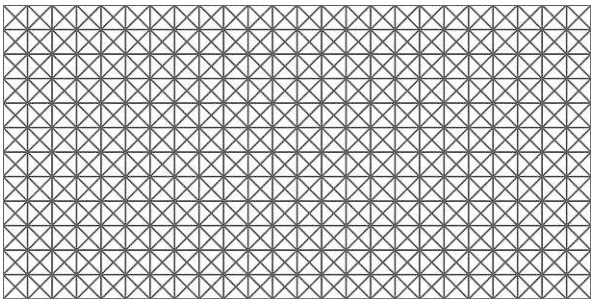
(a) Magnitude of the displacement before the optimization.



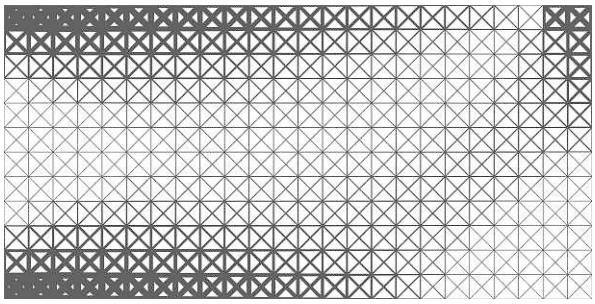
(b) Magnitude of the displacement after optimization.

Figure 5.4.: Qualitative comparison of the displacement magnitude before and after optimization. The deformation results from a unit force applied to the top right microtiles of the microstructure. Since the magnitude scales linearly with the applied force and the objective function describes the stiffness of the entire structure, a color bar was intentionally omitted. Both use the same color scheme for comparison.

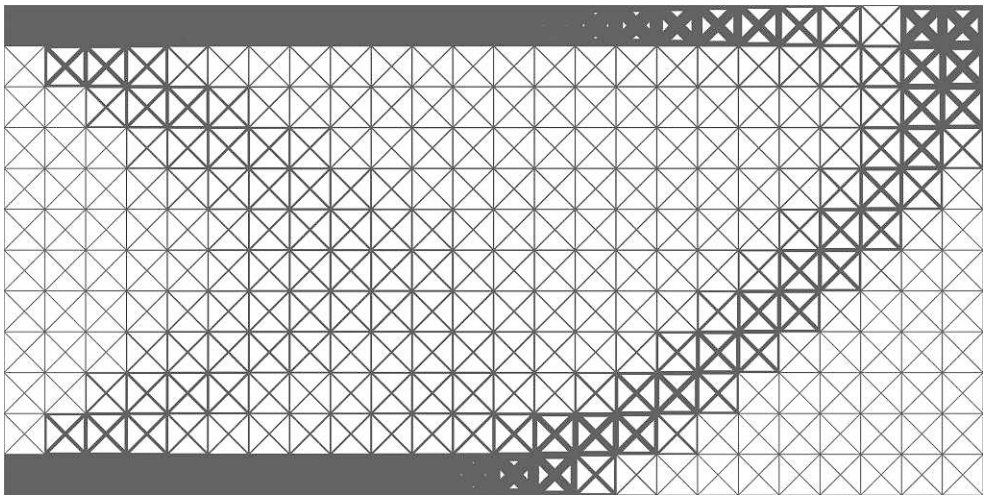
a "cleaner" design - and allowed the authors to more aggressively eliminate all cells with a thickness below a certain threshold. This reduced design can be beneficial for fabrication, as fewer cells with intricate features are present in the final structure. On the other hand, penalizing intermediate values also increased the overall compliance of the system, hence resulting in a less favorable design. In the context of this work, the topology of the geometry remains unchanged over the design iterations. Compared to the non-penalized geometry, the result of the presented approach qualitatively is in good agreement with the benchmark test.



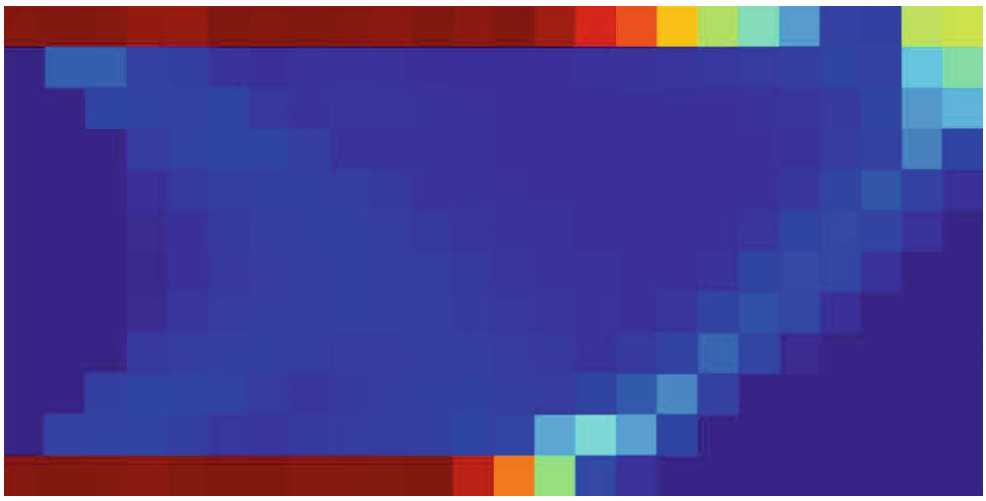
(a) Initial Design



(b) Iteration 5



(c) Iteration 35 (final)



(d) Parameter spline (zeroth degree)

Figure 5.5.: Design evolution of the cantilever during optimization shown at different iterations during the optimization. The thickness of the initial design is chosen such that $\tilde{\rho} = \tilde{\rho}_t$. Figure (d) shows the parameter spline of the finalized design.

5.3. Optimization of a Bending Arch

As mentioned in the previous section, a composition-based approach was not necessarily required to fill the rectangular domain with a microstructure. Thus, in the second test case, the method is demonstrated on a curved outer geometry, highlighting the key advantages of the boundary-conformal microstructure approach. The paradigm has already been demonstrated on a similar geometry in [103] to highlight the benefits of reduced order modeling in analysis of microstructured geometries.

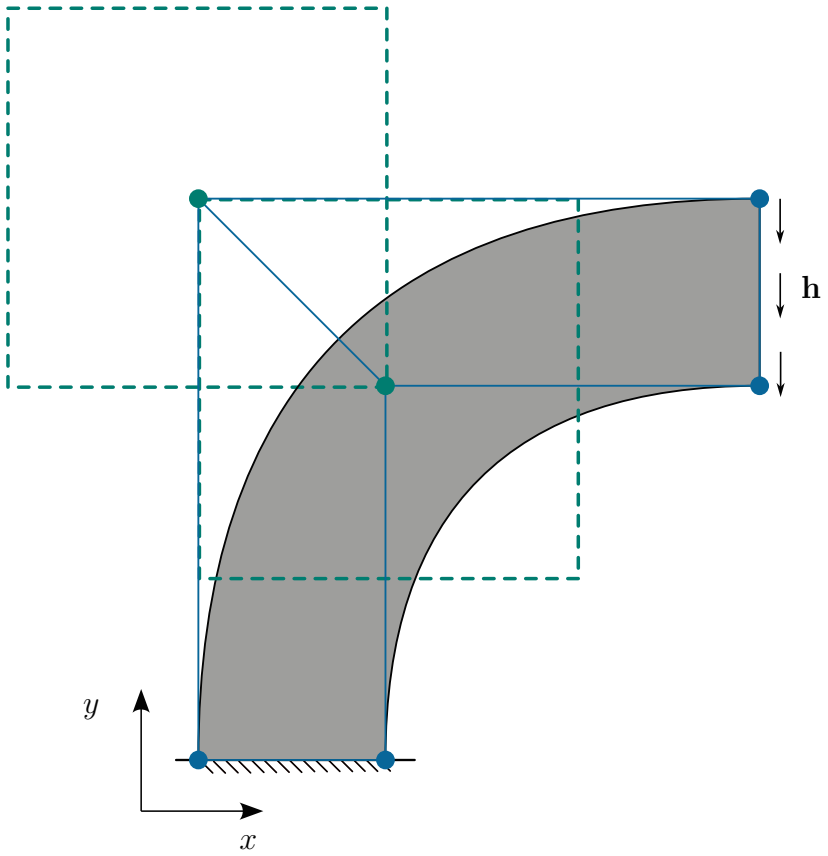


Figure 5.6.: Schematic outline of the bending arch test case. The deformation function (macro geometry) is defined using a linear-quadratic Bézier spline. Control points 2 and 5 (green) are treated as design variables. The bounds of the control point displacement are illustrated with the dashed rectangles.

The schematic outline of the second example is shown in Figure 5.6. The outer geometry is a polynomial approximation of a quarter circle described by a B-Spline, which is quadratic in the angular direction and linear in the radial direction. The control point mesh of the spline is shown in blue. The outer dimensions of the macro geometry are $3\,\mathrm{m} \times 3\,\mathrm{m}$, the width of the arc is $1\,\mathrm{m}$, so the inner and outer radii are $2\,\mathrm{m}$ and $3\,\mathrm{m}$, respectively. The horizontal boundary on the left side is clamped, and a vertical unit traction of $1\,\mathrm{Pa}$ is applied to the vertical side on the right. The material parameters are chosen as in the first example and can be found in Table 5.1

Unlike in the first example, the macro geometry is not kept constant during optimization. Instead, the control points $\mathbf{C}_2$ and $\mathbf{C}_5$, shown in green in the Figure 5.6, are treated as design variables. Both the horizontal and vertical position of the two control points can

be modified by the optimizer within a box constraint of $\pm 1\,\mathrm{m}$, indicated by the dashed rectangles. Additionally, the local thickness of the microtiles is controlled using a bilinear B-spline with 5×5 control points with evenly spaced inner nodes. In contrast to the first example, where all tile parameters are assigned individually, the abstraction of the thickness with linear spline basis functions ensures smoother transitions between neighboring tiles.

The deformation function is filled with a grid of 32×8 microtiles, each consisting of 20 bi-linear Bézier patches. The entire microstructure thus consists of 5120 patches of degree 3 and is defined by 50 025 non-duplicate control points. This results in a total of 99 904 DOFs, accounting for coefficients on the Dirichlet boundary.

The objective of the optimization is to reduce the compliance of the structure while maintaining a mass associated with a volumetric density $\tilde{\rho}_t$ of 50 % in the initial, undeformed macro shape. However, since the deformation function $\mathcal{T}$ will not necessarily remain unchanged over the course of the optimization, the total mass must be constrained, with $m_t = \tilde{\rho}_t \; m_{\mathcal{T}_0}$, resulting in

$$\mathbf{k}_I(\boldsymbol{\chi}) = m - m_t \leq 0 \quad . \tag{5.12}$$

This configuration is chosen specifically to compare different optimization strategies. Using this setup, three different cases can be considered:

Local optimization – where only the *local* tile thickness is modified, and the deformation function remains unchanged, resulting in 25 design variables.

Macro optimization – where the control points can be modified and the thickness of the tiles is adjusted *globally* with a single value to fulfill the mass constraint, yielding 5 design variables.

Concurrent optimization – where both the *local* tile thickness and the control points can be modified, leading to a total of 29 design variables.

All three strategies were tested in the same framework described in section 3.3.6, using the SLSQP algorithm from the `scipy` library in Python. Again, all initial values of the objective function are scaled to 1 at the beginning of the optimization, similarly the design variables are scaled to ensure optimal performance of the optimization driver.

The evolution of the objective function for all three cases is shown in Figure 5.7. All three cases show similar convergence behavior and reach a local minimum within 15 iterations. Notably, only the macro and concurrent optimizations exceed the maximum mass constraint in some of the first couple of iterations.

The optimized designs are shown in Figure 5.8. In the local optimization, the material is concentrated at the boundary of the design, which makes the structure stiffer against bending, similar to the concept of sandwich structures. In addition, there is more material in the vertical sections of the arch, especially at the inner radius, suggesting that the vertical load is concentrated in this area. Overall, the compliance of the optimized design is reduced by approximately 61 % compared to the original design.

In the macro optimization, the structure becomes wider, which also makes it more rigid against bending. In addition, both control points move significantly closer to the Neumann boundary. Although this changes the contour of the outer geometry only slightly, it affects the inner parametrization of the deformation function, which in turn determines the distribution of the microtiles within the microstructure. As the control points get

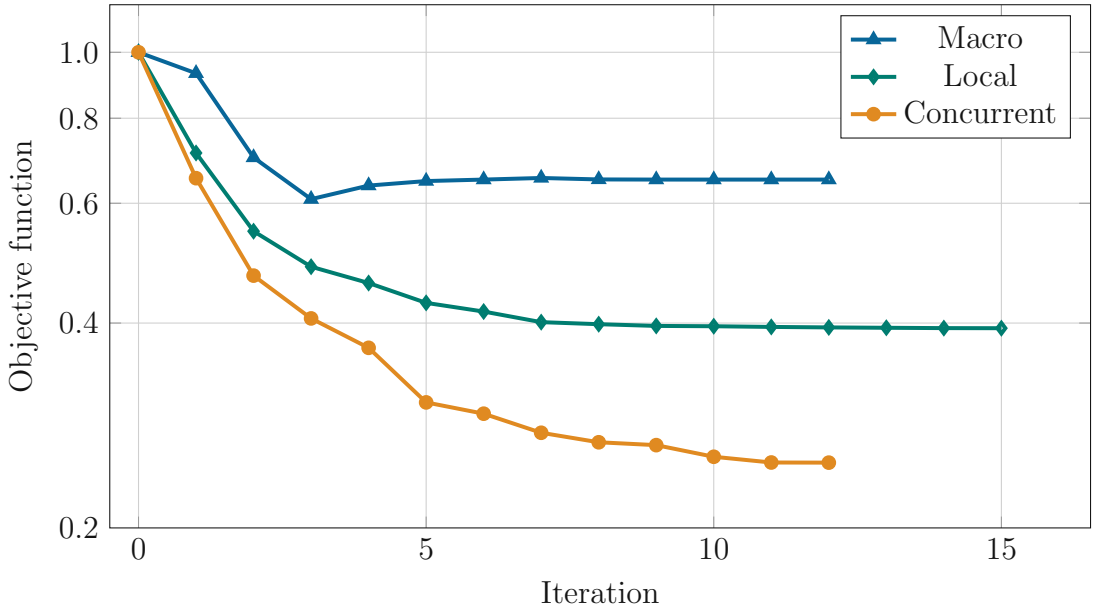


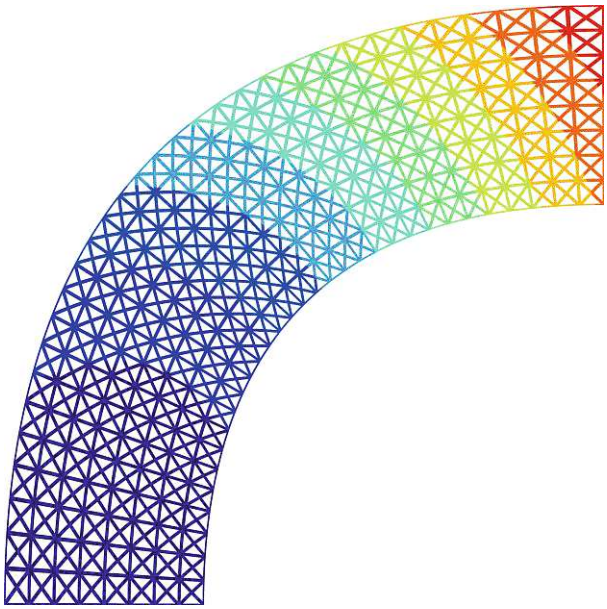
Figure 5.7.: The evolution of the objective function over the course of the iterations, plotted logarithmically. The objective function is scaled by a constant factor so the initial configuration evaluates to 1. The thickness of the initial design is chosen so that $m_t = m$. The lowest value in the fourth iteration of the macro geometry optimization is not a valid minimum, because the mass constraints are violated.

closer to the upper right surface, the "tile density", i.e. the number of tiles in a given area, increases. As a result, the tiles on the right are squeezed horizontally and the diagonal trusses are arranged at a steeper angle, making it easier to distribute the vertical load. The same is true for the tiles in the lower left, where the elongation of the tiles creates trusses that are more upright. However, the effects of the macro optimization are not as significant as the local optimization, only reducing the compliance by about 39 %.

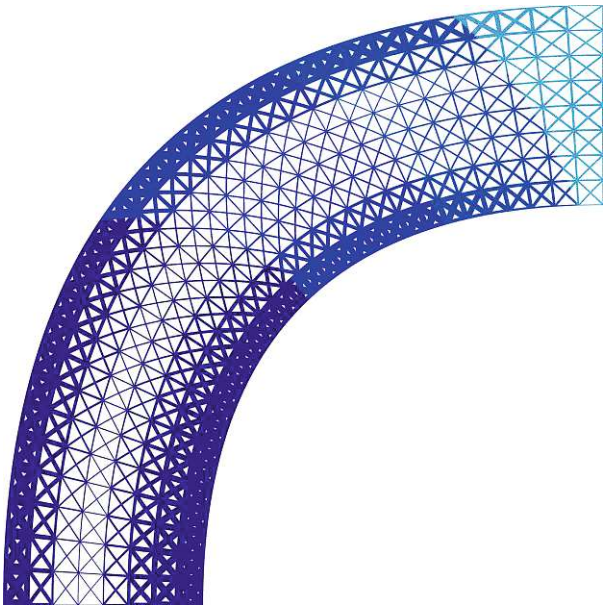
Optimization Type	Compliance reduction [%]	Reduction of maximum displacement [%]
Macro Optimization	39.1615	34.2198
Local Micro Optimization	60.6879	60.2637
Concurrent Optimization	75.0576	71.1745

Table 5.2.: Summary of the results of the individual optimization strategies.

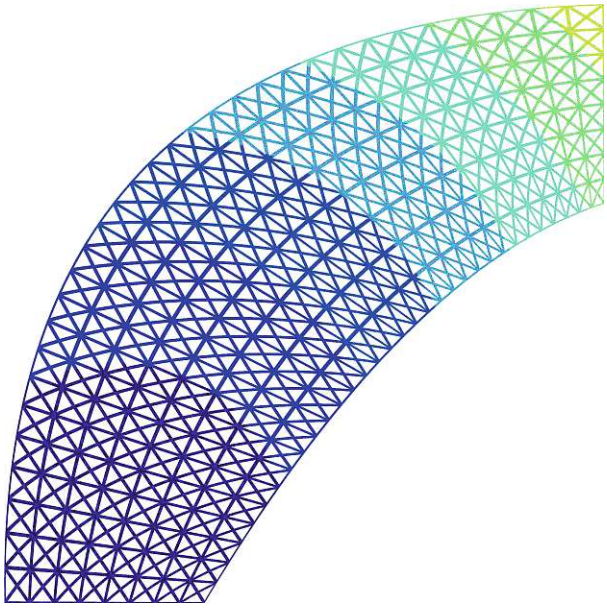
The concurrent optimization combined these two effects and achieved a total reduction of the compliance of about 75 %. The summary of the results of the different optimization strategies can be found in Table 5.2. As an alternative measure of the stiffness, it also shows the reduction of maximum displacement.



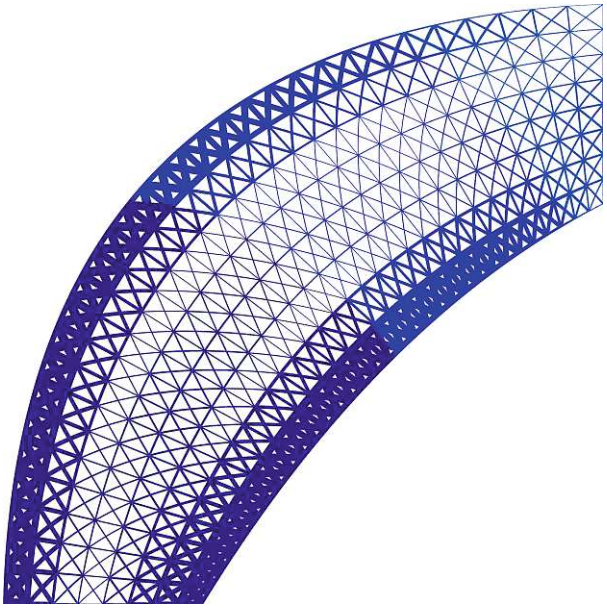
(a) Initial design of the bending arch



(b) Result of the local optimization



(c) Result of the macro optimization



(d) Result of the concurrent optimization

Figure 5.8.: Comparison of the final geometries after optimization. The figures all use the same color scheme to represent the magnitude of the displacements. The displacements scale proportionally to the applied force. The thickness of the initial design is chosen that $m_t = m$.

5.4. Discussion

The paradigm of constructing boundary-conforming microstructures through functional composition allows even complex geometries to be seamlessly filled without the need to find solutions for cells that lie on the boundary, which is a common issue for cells stacked in a Cartesian grid. This is particularly useful in the context of structural optimization, where straight edges and rectangular designs rarely occur. As mentioned in the previous sections, the macro-shape of a structural part is generally designed to efficiently "guide" forces through the structure. With boundary-conforming microstructures that naturally align with the external geometry, the local structure will therefore potentially follow the flow of forces in the same way.

In the first example, the tile parameters were set individually. In fact, using a higher parameter spline degree with a smaller number of design variables in the abstraction ultimately reduces the design freedom and limits the minimum value that can be achieved (for example, in the first example, using a bilinear parameter spline with a control mesh of 12×6 coefficients only allows a reduction of 50.18 % instead of 55.22 %). However, it can still be advantageous to use the spline-based abstraction. Parameter splines with non-zero, smooth basis functions tend to regularize the problem and help avoid undesirable phenomena such as checkerboard patterns[2]. These patterns are naturally filtered out by the layer of abstraction. This additional blurring of design variables also ensures smoother transitions between high and low density regions where neighboring cells have similar tile parameters. This results in more intuitive designs. It also helps to avoid regions with extremely narrow structures that can cause manufacturing problems. Additionally, denser regions might in some cases prevent stress concentrations, as the load can be distributed among more tiles, as a preventive measure against damage formation.

Finally, the additional layer of abstraction also significantly reduces the number of design parameters, as described in section 2.3. Arguably, the cost of gradient computation is not (significantly) reduced with a smaller number of design variables when using the adjoint method. However, gradient information may not always be available, or may require the use of FD, or AD, for example, when black-box solvers are used, or the implementation of adjoints would be too cumbersome for complex nonlinear PDEs. Here, a reduction of the design space can provide remedy and allow a faster convergence of the optimization algorithm.

[2] Checkerboard patterns, numerical instabilities observed in density-based topology optimization approaches [204], manifest as alternating regions of high and low density. While we have observed instances of high fluctuation between adjacent tiles in our approach, it is uncertain whether these phenomena are directly related, as there is no penalization of intermediate values in the spline based abstraction.

6 Conclusion and Outlook

6.1. Conclusion

This thesis presents a comprehensive framework for the design, analysis, and optimization of microstructured CAD-compliant geometries. This framework adopts the design paradigm based on a functional composition between splines. Using this approach, we further introduced an abstraction of the local geometry parameters to reduce the number of design variables. This reduction allows the use of less performant optimization algorithms and minimizes local fluctuations in the geometries, resulting in smooth transitions between neighboring splines.

In addition, we derived geometric derivatives in terms of parameters that control both the macroscopic and local geometry, involving the spline abstraction. This allowed the application of an adjoint approach to compute sensitivities, which are critical for gradient-based optimization algorithms. These algorithms generally achieve faster convergence to an optimal solution. To verify and test these results, we developed a modular, open-source framework. This framework allows the testing of different optimization algorithms and strategies, ensuring the robustness and flexibility of the proposed methods.

As a first example, the framework was tested on a heat transfer problem to achieve a predefined temperature distribution on a specified subdomain under Dirichlet and Neumann boundary conditions, controlling only the local geometry. This setup allowed the comparison of different optimization algorithms in the context of PDE-constrained shape optimization. To further test the applicability of the framework to complex geometries, an extrusion die test case was recreated. The new framework showed improvements in both analysis and optimization over the reference. The use of IGA maintained geometric accuracy and eliminated the need for remeshing. In addition, the use of an adjoint approach made sensitivity calculations more efficient, compared to preceeding test cases, which relied on finite difference methods for derivative calculations.

The framework was then applied to structural optimization to minimize the compliance of a microstructure under nonlinear mass constraints. Initially, the framework was validated using a cantilever benchmark test, yielding results consistent with traditional topology optimization methods. However, traditional methods do not yield boundary-conforming microstructures where the local geometry naturally aligns with the external shape. Therefore, a load-bearing arch composed of microtiles was optimized to demonstrate the full potential of this paradigm. This test case compared the effects of different

optimization strategies, including local, macroscopic, and concurrent optimization. The results highlighted the potential of combining global and local optimization and further demonstrated the benefits of boundary-conforming microstructures in the context of structural design.

6.2. Outlook

While the current work has laid a foundation for the design of microstructured, CAD-compliant geometries, it has also revealed several potential future research directions. When it comes to the design of microstructured geometries, a shift in the design approach to V-Rep geometries can facilitate the use of this paradigm for topologically complicated, (trimmed,) multi-patch geometries that can be used as the deformation function. While there has been research on modeling microstructures with trimmed V-Rep splines [110] and bifurcating tiles [158], efforts are still needed to incorporate these findings in the context of optimization.

Another significant potential for improving shape optimization lies in improving the performance of the analysis. Analysis is currently the bottleneck of the iterative process, primarily due to the high polynomial degrees of the microstructures, but also due to their complex topologies and number of DOFs. While some recent research has focused on reducing the analysis time [103, 105], there is still potential to explore model order reduction techniques or other methods potentially including scientific machine learning to speed up the analysis and thus reduce the computational cost per iteration.

Extending the study of microstructure simulation to fluid-structure interactions holds promise for a variety of applications. For example, cooling systems or heat exchangers could benefit from evaluating fluid flow through the microstructure, as shown in Figure 6.1. In addition, extrusion die simulation could be extended to include the melt itself. In a coupled die-melt simulation, system performance could be evaluated based on the behavior of the melt, with structural integrity serving as a constraint to withstand the high pressures within the flow channel.

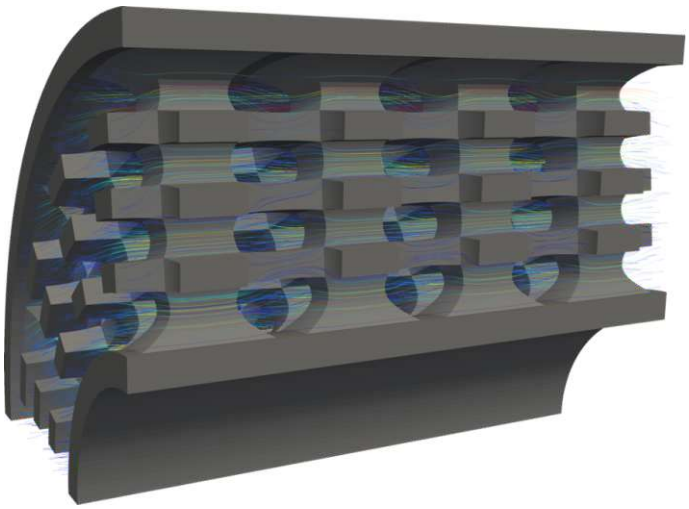


Figure 6.1.: Simulation of fluid flow through a microstructure as a demonstration. Courtesy of Daniel Wolff.

Moreover, it is worth investigating structural issues more thoroughly to account for the nonlinear behavior of the structures using the boundary-conforming paradigm. Including stresses or potential energy in the objective function could broaden the scope of design applications. Recent advances in AM, particularly the development of novel high-performance materials, have created new opportunities for the design and implementation of complex microstructures. These potential applications can be found in many fields of engineering, from robotics to damping systems in automotive parts and beyond.

A Additional Remarks on the Linear Elasticity Formulation

A.1. Variational Form and Derivatives

Recalling the strong form of the linear elasticity problem

$$\begin{aligned} & \nabla \cdot \boldsymbol{\sigma} + \mathbf{f} = \mathbf{0} && \text{on } \Omega \\ \text{with} \quad & \boldsymbol{\sigma} = \lambda \mathrm{tr}\,(\boldsymbol{\epsilon})\,\mathbf{I} + 2\mu\boldsymbol{\epsilon} \\ \text{and} \quad & \boldsymbol{\epsilon} = \frac{1}{2}\left[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T\right] \\ & \mathbf{u} = \mathbf{g} && \text{on } \Gamma_D \\ & \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{h} && \text{on } \Gamma_N \quad , \end{aligned} \tag{A.1}$$

With the function spaces

$$\mathcal{S}_{\mathbf{u}} = \left\{\mathbf{u} \,|\, \mathbf{u} \in \mathcal{H}^1(\Omega),\ \mathbf{u} = \mathbf{g} \text{ on } \Gamma_D\right\} \quad , \tag{A.2a}$$

$$\mathcal{V} = \left\{\mathbf{w} \,|\, \mathbf{w} \in \mathcal{H}^1(\Omega),\ \mathbf{w} = \mathbf{0} \text{ on } \Gamma_D\right\} \quad . \tag{A.2b}$$

Using the techniques introduced in Section 3.2, we obtain the variational (or weak form) as

$$\int_\Omega \left[\lambda \nabla \cdot \mathbf{u}\mathbf{I} + \mu\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)\right] : (\nabla \mathbf{w})\, \mathrm{d}V = \int_{\partial\Omega} \mathbf{h} \cdot \mathbf{w}\, \mathrm{d}S + \int_\Omega \mathbf{f} \cdot \mathbf{w}\, \mathrm{d}V \quad , \tag{A.3}$$

or, separated into simpler terms[1]

$$\begin{aligned} &\underbrace{\int_\Omega \lambda\,(\nabla \cdot \mathbf{u})\,(\nabla \cdot \mathbf{w})\, \mathrm{d}V}_{a_{\mathbf{u},\lambda}(\mathbf{u},\mathbf{w})} + \underbrace{\int_\Omega \mu\,(\nabla \mathbf{u}) : (\nabla \mathbf{w})\, \mathrm{d}V}_{a_{\mathbf{u},\mu_1}(\mathbf{u},\mathbf{w})} + \dots \\ &\dots \underbrace{\int_\Omega \mu\,(\nabla \mathbf{u})^T : (\nabla \mathbf{w})\, \mathrm{d}V}_{a_{\mathbf{u},\mu_2}(\mathbf{u},\mathbf{w})} = \underbrace{\int_{\Gamma_N} \mathbf{h} \cdot \mathbf{w}\, \mathrm{d}S + \int_\Omega \mathbf{f} \cdot \mathbf{w}\, \mathrm{d}V}_{L_{\mathbf{u}}(\mathbf{w})} \end{aligned} \tag{A.4}$$

[1] Here, the term $\boldsymbol{\epsilon}(\mathbf{u}) : \nabla \mathbf{w}$ is often written as $\boldsymbol{\epsilon}(\mathbf{u}) : \boldsymbol{\epsilon}(\mathbf{w})$, which originates from the identity $A : B = 0$ if A is symmetric and B is antisymmetric.

Recalling the IGA-approach and the definition of the trial functions, we can write the displacement field similar to Equation (3.23) as

$$\tilde{\mathbf{u}}^h = \sum_i^{n_{\hat{\mathbf{u}}}} \mathcal{R}_i \hat{\mathbf{u}}_i + \sum_i^{n_D} \mathcal{R}_i^D \hat{\mathbf{u}}_i^D \quad , \tag{A.5}$$

in the parametric domain, with $\mathbf{u}^h = \tilde{\mathbf{u}}^h \circ \Psi$, the number of independent displacement coefficients $n_{\hat{\mathbf{u}}}$, as well as the number of coefficient on the Dirichlet boundary n_D.

Building on this, the terms introduced in Equation (A.4) can be used to build a linear system of equations using

$$\begin{aligned} K_{ij} &= a_{\mathbf{u}}(\mathcal{R}_i, \mathcal{R}_j) \\ &= a_{\mathbf{u},\lambda}(\mathcal{R}_i, \mathcal{R}_j) + a_{\mathbf{u},\mu_1}(\mathcal{R}_i, \mathcal{R}_j) + a_{\mathbf{u},\mu_2}(\mathcal{R}_i, \mathcal{R}_j) \quad , \qquad \text{(A.6)} \\ F_j &= L_{\mathbf{u}}(\mathcal{R}_j) \quad . \qquad \text{(A.7)} \end{aligned}$$

Using the following identities (with $(\mathbf{J}_\Psi)_{ij} = {}^{\partial x_i}/_{\partial \xi_j}$),

$$\begin{aligned} \nabla \mathbf{u} &= \frac{\partial \mathbf{u}_i}{\partial x_j} = \frac{\partial \mathbf{u}_i}{\xi_k} \frac{\xi_k}{\partial x_j} \\ &= \tilde{\nabla} \mathbf{u} \mathbf{J}_\Psi^{-1} \quad , \qquad \text{(A.8a)} \\ \nabla \cdot \mathbf{u} &= \frac{\partial \mathbf{u}_i}{\partial x_i} = \frac{\partial \mathbf{u}_i}{\partial x_j} \delta_{ij} = \operatorname{tr}(\nabla \mathbf{u}) \\ &= \operatorname{tr}\left(\tilde{\nabla} \mathbf{u} \mathbf{J}_\Psi^{-1}\right) = \tilde{\nabla} \mathbf{u} : \mathbf{J}_\Psi^{-T} \quad , \qquad \text{(A.8b)} \end{aligned}$$

we can pull back Equation (A.4) into parametric space, which yields

$$\begin{aligned} &\int_{\tilde{\Omega}} \lambda \left(\tilde{\nabla} \mathbf{u} : \mathbf{J}_\Psi^{-T}\right) \left(\tilde{\nabla} \mathbf{w} : \mathbf{J}_\Psi^{-T}\right) \det(\mathbf{J}_\Psi) \, \mathrm{d}V + \int_{\tilde{\Omega}} \mu \left(\tilde{\nabla} \mathbf{u} \mathbf{J}_\Psi^{-1}\right) : \left(\tilde{\nabla} \mathbf{w} \mathbf{J}_\Psi^{-1}\right) \det(\mathbf{J}_\Psi) \, \mathrm{d}V + \ldots \\ &\quad \ldots + \int_{\tilde{\Omega}} \mu \left[\mathbf{J}_\Psi^{-T} (\tilde{\nabla} \mathbf{u})^T\right] : \left(\tilde{\nabla} \mathbf{w} \mathbf{J}_\Psi^{-1}\right) \det(\mathbf{J}_\Psi) \, \mathrm{d}V = \ldots \\ &\quad \ldots = \int_{\Psi^{-1}(\Gamma_N)} \mathbf{h} \cdot \mathbf{w} \; J^\Gamma \, \mathrm{d}S + \int_{\tilde{\Omega}} \mathbf{f} \cdot \mathbf{w} \, \det(\mathbf{J}_\Psi) \, \mathrm{d}V \quad . \end{aligned} \tag{A.9}$$

To compute derivatives with the adjoint approach, we need to introduce the derivatives of the PDE-constraint with respect to the control point components. Therefore, we need

to differentiate the (bi-)linear functions. Using the identities from Equations (A.8) and (3.42) these can be written as (cf . Section 3.4)

$$\begin{aligned}
\frac{\partial a_{\mathbf{u},\lambda}(\mathbf{u},\mathbf{w})}{\partial \mathbf{C}_i} = &- \int_{\tilde{\Omega}} \lambda \left[\tilde{\nabla}\mathbf{u} : \left(\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathcal{R}^i)^T\mathbf{J}_\Psi^{-T}\right)\right]\left(\tilde{\nabla}\mathbf{w} : \mathbf{J}_\Psi^{-T}\right)\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&- \int_{\tilde{\Omega}} \lambda \left[\tilde{\nabla}\mathbf{u} : \mathbf{J}_\Psi^{-T}\right]\left[\tilde{\nabla}\mathbf{w} : \left(\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathcal{R}^i)^T\mathbf{J}_\Psi^{-T}\right)\right]\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&+ \int_{\tilde{\Omega}} \lambda \left[\tilde{\nabla}\mathbf{u} : \mathbf{J}_\Psi^{-T}\right]\left(\tilde{\nabla}\mathbf{w} : \mathbf{J}_\Psi^{-T}\right)\det(\mathbf{J}_\Psi)\mathrm{tr}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\right)\,\mathrm{d}V \qquad \text{(A.10a)} \\
\frac{\partial a_{\mathbf{u},\mu_1}(\mathbf{u},\mathbf{w})}{\partial \mathbf{C}_i} = &- \int_{\tilde{\Omega}} \mu \left[\tilde{\nabla}\mathbf{u}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\mathbf{J}_\Psi^{-1}\right)\right] : \left(\tilde{\nabla}\mathbf{w}\mathbf{J}_\Psi^{-1}\right)\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&- \int_{\tilde{\Omega}} \mu \left(\tilde{\nabla}\mathbf{u}\mathbf{J}_\Psi^{-1}\right) : \left[\tilde{\nabla}\mathbf{w}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\mathbf{J}_\Psi^{-1}\right)\right]\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&+ \int_{\tilde{\Omega}} \mu \left(\tilde{\nabla}\mathbf{u}\mathbf{J}_\Psi^{-1}\right) : \left(\tilde{\nabla}\mathbf{w}\mathbf{J}_\Psi^{-1}\right)\det(\mathbf{J}_\Psi)\mathrm{tr}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\right)\,\mathrm{d}V \qquad \text{(A.10b)} \\
\frac{\partial a_{\mathbf{u},\mu_2}(\mathbf{u},\mathbf{w})}{\partial \mathbf{C}_i} = &- \int_{\tilde{\Omega}} \mu \left[\left(\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathcal{R}^i)^T\mathbf{J}_\Psi^{-T}\right)\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathbf{u})^T\right] : \left(\tilde{\nabla}\mathbf{w}\mathbf{J}_\Psi^{-1}\right)\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&- \int_{\tilde{\Omega}} \mu \left[\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathbf{u})^T\right] : \left(\tilde{\nabla}\mathbf{w}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\mathbf{J}_\Psi^{-1}\right)\right)\det(\mathbf{J}_\Psi)\,\mathrm{d}V \\
&+ \int_{\tilde{\Omega}} \mu \left[\mathbf{J}_\Psi^{-T}(\tilde{\nabla}\mathbf{u})^T\right] : \left(\tilde{\nabla}\mathbf{w}\mathbf{J}_\Psi^{-1}\right)\det(\mathbf{J}_\Psi)\mathrm{tr}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\right)\,\mathrm{d}V \qquad \text{(A.10c)} \\
\frac{\partial L_{\mathbf{u}}(\mathbf{w})}{\partial \mathbf{C}_i} = &\int_{\Psi^{-1}(\Gamma_N)} \mathbf{h}\cdot\mathbf{w}\frac{\partial J^\Gamma}{\partial \mathbf{C}_i}\,\mathrm{d}S + \int_\Omega \mathbf{h}\cdot\mathbf{w}\det(\mathbf{J}_\Psi)\mathrm{tr}\left(\mathbf{J}_\Psi^{-1}(\tilde{\nabla}\mathcal{R}^i)\right)\,\mathrm{d}V \qquad \text{(A.10d)}
\end{aligned}$$

A.2. Computing the Compliance

In this section we show how the compliance used in Chapter 5 is written using components of the linear system.

Given the compliance given in the form

$$L_{\mathbf{u}}(\mathbf{u}) = \int_{\partial\Omega} \mathbf{h}\cdot\mathbf{u}\,\mathrm{d}S + \int_\Omega \mathbf{f}\cdot\mathbf{u}\,\mathrm{d}V \quad . \qquad \text{(A.11)}$$

As the test functions $\mathbf{w}$ are the same as the trial functions associated to the solution field coefficients, the above equation can be rewritten as follows

$$\begin{aligned}
L_{\mathbf{u}}(\mathbf{u}) &= \int_{\partial\Omega} \mathbf{h}\cdot\left(\sum_i \mathcal{R}^i\hat{\mathbf{u}}_i\right)\,\mathrm{d}S + \int_\Omega \mathbf{f}\cdot\left(\sum_i \mathcal{R}^i\hat{\mathbf{u}}_i\right)\,\mathrm{d}V \quad , \qquad \text{(A.12)} \\
&= \sum_i \int_{\partial\Omega} \mathbf{h}\,\mathcal{R}^i\,\mathrm{d}S\cdot\hat{\mathbf{u}}_i + \int_\Omega \mathbf{f}\,\mathcal{R}^i\,\mathrm{d}V\cdot\hat{\mathbf{u}}_i \quad , \qquad \text{(A.13)} \\
&= \sum_i \Bigg(\underbrace{\int_{\partial\Omega} \mathbf{h}\,\mathcal{R}^i\,\mathrm{d}S + \int_\Omega \mathbf{f}\,\mathcal{R}^i\,\mathrm{d}V}_{\mathbf{F}_i}\Bigg)\cdot\underbrace{\hat{\mathbf{u}}_i}_{\mathbf{u}_i} \quad . \qquad \text{(A.14)}
\end{aligned}$$

B

Alternative Measures for structural compliance

In Chapter 5 of this thesis, we explored shape optimization using a compliance measure related to deformation energy to evaluate the structural performance of the design, see Equation (5.4). However, as mentioned before, there are other measures of stiffness, one of which is described in this appendix.

Instead of considering the applied external forces, this measure focuses solely on the deformation within a specific region of interest. Specifically, it determines stiffness by the amount of deformation in the area where the external force is applied, known as the Neumann boundary Γ_N. This can be expressed as

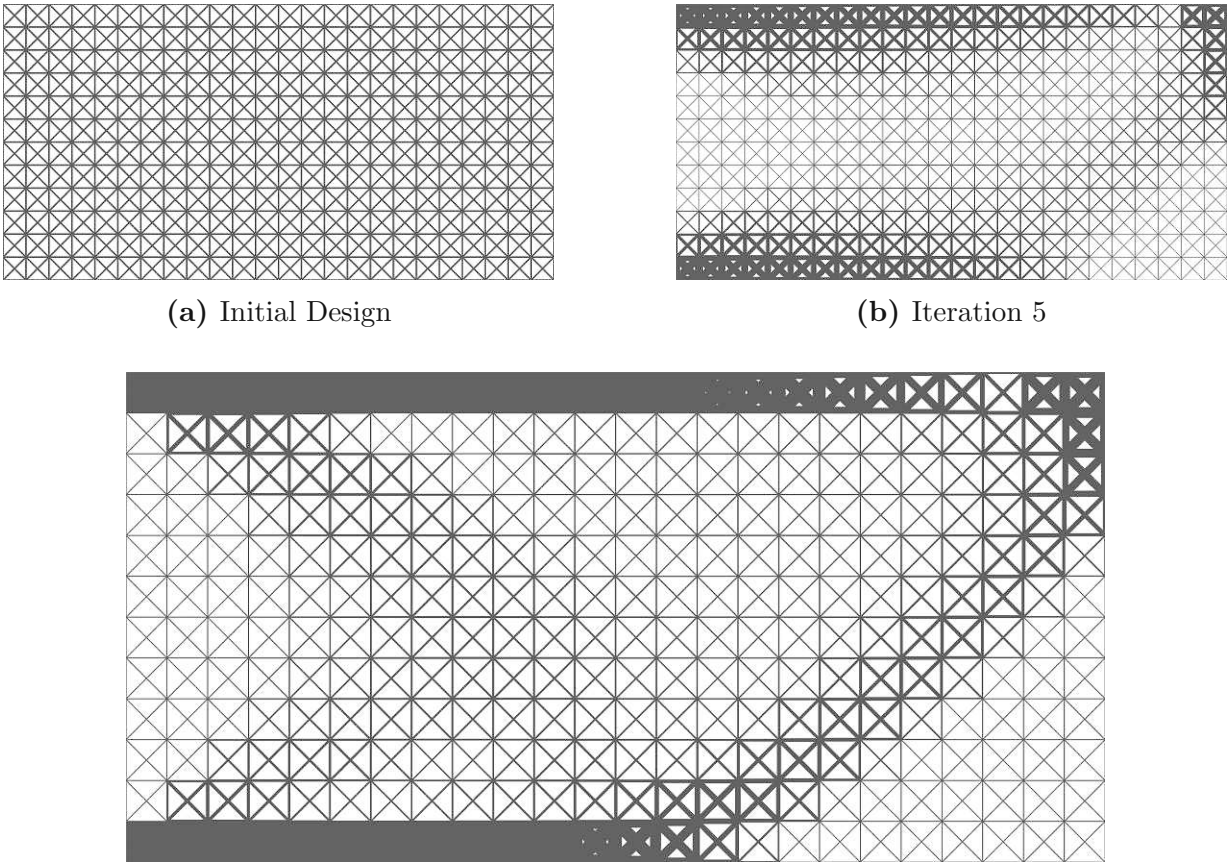
$$\begin{aligned} \mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u}) &= \int_{\Gamma_N} \mathbf{u} \cdot \mathbf{u} \, \mathrm{d}S \quad , \\ &= \int_{\Psi^{-1}(\Gamma_N)} \mathbf{u} \cdot \mathbf{u} \, J^{\Gamma} \, \mathrm{d}S \quad . \end{aligned} \tag{B.1}$$

In order to compute adjoint sensitivities using the adjoint approach, we further introduce the derivatives of Equation (B.1), which read

$$(\partial_{\mathbf{u}} J)_i = \frac{\partial \mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u})}{\partial \hat{\mathbf{u}}_i} = 2 \int_{\Psi^{-1}(\Gamma_N)} \mathcal{R}_i \, \mathbf{u} \, J^{\Gamma} \, \mathrm{d}S \quad , \tag{B.2}$$

$$(\partial_{\mathbf{C}} J)_i = \frac{\partial \mathcal{J}(\Omega(\boldsymbol{\chi}), \mathbf{u})}{\partial \mathbf{C}_i} = \int_{\Psi^{-1}(\Gamma_N)} \mathbf{u} \cdot \mathbf{u} \, \frac{\partial J^{\Gamma}}{\partial \mathbf{C}_i} \, \mathrm{d}S \quad . \tag{B.3}$$

This measure was also applied to the cantilever benchmark test case introduced in Chapter 5.2. Unlike the original compliance, the calculation of sensitivities requires the additional solution of a linear system of equations, which makes it more computationally expensive. Thus, despite a similar number or iterations required to converge to a local minimum, the total computation time exceeds that of the original compliance. Qualitatively, the results of the cantilever optimization are equivalent to those computed with the energy-based compliance. The optimized designs are shown in Figure B.1 along with snapshots throughout the iterative process. Figure B.2 shows the parameter spline abstraction associated with the optimized design.



(a) Initial Design

(b) Iteration 5

(c) Iteration 32 (final)

Figure B.1.: Design evolution of the cantilever design during optimization using the $\mathcal{L}^2$ norm of the displacement as objective function.

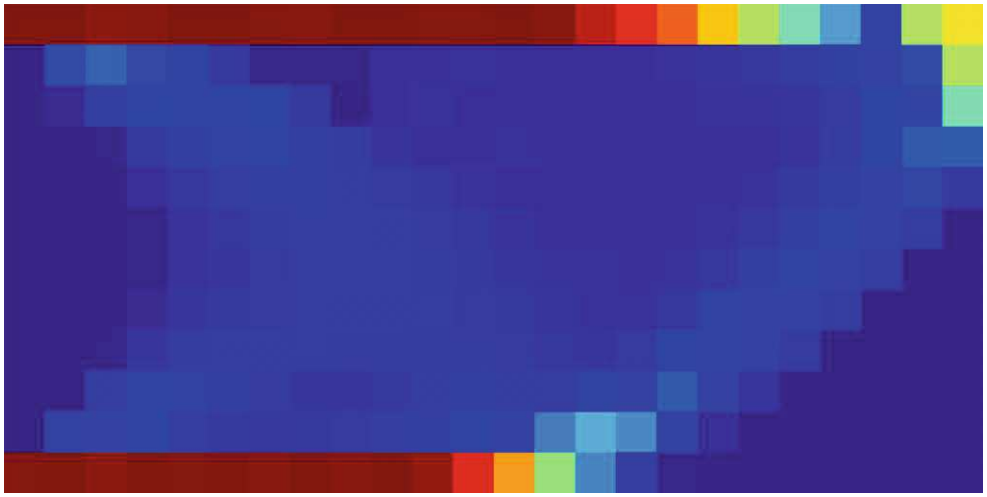


Figure B.2.: Parameter-spline of the optimized geometry with the alternative compliance measure. Here we assigned every tile their parameters individually, which can be interpreted as a zero-th degree spline in the proposed framework.

Bibliography

[1] Meisam Abdi, Ian Ashcroft, and Ricky D. Wildman. "Design optimisation for an additively manufactured automotive component". In: *International Journal of Powertrains* 7.1 (2018), p. 142. ISSN: 1742-4267, 1742-4275. DOI: 10.1504/IJPT.2018.090371 (cit. on p. 7).

[2] I. Akkerman et al. "The role of continuity in residual-based variational multiscale modeling of turbulence". In: *Computational Mechanics* 41.3 (Feb. 2008), pp. 371–378. ISSN: 0178-7675, 1432-0924. DOI: 10.1007/s00466-007-0193-7 (cit. on p. 38).

[3] Monzer Al Khalil et al. "A Biomimetic Design Method for 3D-Printed Lightweight Structures Using L-Systems and Parametric Optimization". In: *Applied Sciences* 12.11 (May 29, 2022), p. 5530. ISSN: 2076-3417. DOI: 10.3390/app12115530 (cit. on p. 7).

[4] Monzer Al Khalil et al. "A Biomimetic design method for 3D-printed lightweight structures using L-systems and parametric optimization". In: *Applied Sciences* 12.11 (2022), p. 5530 (cit. on p. 24).

[5] Enrique Alabort, Daniel Barba, and Roger C. Reed. "Design of metallic bone by additive manufacturing". In: *Scripta Materialia* 164 (Apr. 2019). ISSN: 13596462. DOI: 10.1016/j.scriptamat.2019.01.022 (cit. on p. 7).

[6] Joe Alexandersen and Boyan S. Lazarov. "Topology optimisation of manufacturable microstructural details without length scale separation using a spectral coarse basis preconditioner". In: *Computer Methods in Applied Mechanics and Engineering* 290 (June 2015), pp. 156–182. ISSN: 00457825. DOI: 10.1016/j.cma.2015.02.028 (cit. on pp. 2, 45).

[7] Grégoire Allaire. *Conception optimale des structures*. Mathématiques et Applications 58. Berlin Heidelberg: Springer, 2007. 278 pp. ISBN: 978-3-540-36710-9 (cit. on p. 77).

[8] Grégoire Allaire. "A review of adjoint methods for sensitivity analysis, uncertainty quantification and optimization in numerical codes". In: *Ingénieurs de l'Automobile* 863 (2915), pp. 33–36 (cit. on p. 50).

[9] Grégoire Allaire, Charles Dapogny, and François Jouve. "Shape and topology optimization". In: *Handbook of Numerical Analysis*. Vol. 22. Elsevier, 2021, pp. 1–132. ISBN: 978-0-444-64305-6. DOI: 10.1016/bs.hna.2020.10.004 (cit. on p. 48).

[10] Grégoire Allaire and Antoine Henrot. "On some recent advances in shape optimization". In: *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics* 329.5 (May 2001), pp. 383–396. ISSN: 16207742. DOI: 10.1016/S1620-7742(01)01349-6 (cit. on p. 48).

[11] Grégoire Allaire et al. “Shape optimization by the homogenization method”. In: *Numerische Mathematik* 76 (1997), pp. 27–68 (cit. on pp. 2, 48, 75, 76).

[12] Holm Altenbach. *Kontinuumsmechanik*. Springer, 2012 (cit. on p. 35).

[13] Pablo Antolin, Annalisa Buffa, and Massimiliano Martinelli. “Isogeometric Analysis on V-reps: First results”. In: *Computer Methods in Applied Mechanics and Engineering* 355 (Oct. 2019), pp. 976–1002. ISSN: 00457825. DOI: 10.1016/j.cma.2019.07.015 (cit. on p. 39).

[14] Pablo Antolin et al. “Optimizing micro-tiles in micro-structures as a design paradigm”. In: *Computer-Aided Design* 115 (2019), pp. 23–33 (cit. on pp. 22, 25, 28, 48).

[15] Andreas Apostolatos et al. “A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis”. In: *International Journal for Numerical Methods in Engineering* 97.7 (Feb. 17, 2014), pp. 473–504. ISSN: 00295981. DOI: 10.1002/nme.4568 (cit. on p. 42).

[16] Sajad Arabnejad and Damiano Pasini. “Mechanical properties of lattice materials via asymptotic homogenization and comparison with alternative homogenization methods”. In: *International Journal of Mechanical Sciences* 77 (Dec. 2013), pp. 249–262. ISSN: 00207403. DOI: 10.1016/j.ijmecsci.2013.10.003 (cit. on p. 48).

[17] A.O. Aremu et al. “A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing”. In: *Additive Manufacturing* 13 (Jan. 2017), pp. 1–13. ISSN: 22148604. DOI: 10.1016/j.addma.2016.10.006 (cit. on pp. 2, 8).

[18] Meisam Askari et al. “Additive manufacturing of metamaterials: A review”. In: *Additive Manufacturing* 36 (2020), p. 101562 (cit. on p. 24).

[19] Arash Ataee et al. “Ultrahigh-strength titanium gyroid scaffolds manufactured by selective laser melting (SLM) for bone implant applications”. In: *Acta Materialia* 158 (Oct. 2018), pp. 354–368. ISSN: 13596454. DOI: 10.1016/j.actamat.2018.08.005 (cit. on pp. 1, 7).

[20] F. Auricchio et al. “Isogeometric Collocation Methods”. In: *Mathematical Models and Methods in Applied Sciences* 20.11 (Nov. 2010). ISSN: 0218-2025, 1793-6314. DOI: 10.1142/S0218202510004878 (cit. on p. 39).

[21] F. Auricchio et al. “A simple algorithm for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (Dec. 2012), pp. 15–27. ISSN: 00457825. DOI: 10.1016/j.cma.2012.04.014 (cit. on p. 42).

[22] Elsevier B.V. *Scopus Search Engine*. https://www.scopus.com/. note Data queried 16.06.2024 using : ”((TITLE-ABS-KEY(isogeometric AND analysis AND shape AND optimization) AND KEY(shape AND optimization) AND KEY(isogeometric AND analysis)))” and ”((TITLE-ABS-KEY(isogeometric AND analysis) AND KEY(isogeometric AND analysis)))”. June 16, 2024 (cit. on p. 3).

[23] Viacheslav Balobanov et al. “Kirchhoff–Love shells within strain gradient elasticity: Weak and strong formulations and an H3-conforming isogeometric implementation”. In: *Computer Methods in Applied Mechanics and Engineering* 344 (Feb. 2019), pp. 837–857. ISSN: 00457825. DOI: 10.1016/j.cma.2018.10.006 (cit. on p. 38).

[24] J. Bauer et al. “Approaching theoretical strength in glassy carbon nanolattices”. In: *Nature Materials* 15.4 (Apr. 2016), pp. 438–443. ISSN: 1476-1122, 1476-4660. DOI: 10.1038/nmat4561 (cit. on p. 1).

[25] Yuri Bazilevs et al. “Computational aerodynamics with isogeometric analysis”. In: *Journal of Mechanics* 39 (Jan. 23, 2023), pp. 24–39. ISSN: 1811-8216. DOI: 10.1093/jom/ufad002 (cit. on p. 39).

[26] Tamara Bechtold, Evgenii B Rudnyi, and Jan G Korvink. “Dynamic electro-thermal simulation of microsystems—a review”. In: *Journal of Micromechanics and Microengineering* 15.11 (Nov. 1, 2005), R17–R31. ISSN: 0960-1317, 1361-6439. DOI: 10.1088/0960-1317/15/11/R01 (cit. on p. 57).

[27] Marek Behr and Feby Abraham. “Free-surface flow simulations in the presence of inclined walls”. In: *Computer Methods in Applied Mechanics and Engineering* 191.47 (Nov. 2002), pp. 5467–5483. ISSN: 00457825. DOI: 10.1016/S0045-7825(02)00444-9 (cit. on p. 19).

[28] M. P. Bendsøe. “Optimal shape design as a material distribution problem”. In: *Structural Optimization* 1.4 (Dec. 1989), pp. 193–202. ISSN: 0934-4373, 1615-1488. DOI: 10.1007/BF01650949 (cit. on p. 48).

[29] M. P. Bendsoe et al. “An Analytical Model to Predict Optimal Material Properties in the Context of Optimal Structural Design”. In: *Journal of Applied Mechanics* 61.4 (Dec. 1, 1994), pp. 930–937. ISSN: 0021-8936, 1528-9036. DOI: 10.1115/1.2901581 (cit. on p. 48).

[30] Martin P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Berlin ; New York: Springer, 2003. 370 pp. ISBN: 978-3-540-42992-0 (cit. on pp. 8, 44, 48, 78).

[31] Serge Bernstein. “Démonstration du théorème de Weierstrass fondé sur le calcul des probabilitеś”. In: *Communications of the Kharkov Mathematical Society* 13 (1912), pp. 1–2 (cit. on p. 9).

[32] Pierre Bézier. “Definition numérique des courbes et surface”. In: *Automatisme* 11.4 (1966), pp. 625–632 (cit. on p. 9).

[33] Pierre E Bézier. “Example of an existing system in the motor industry: the Unisurf system”. In: *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 321.1545 (1971), pp. 207–218 (cit. on p. 9).

[34] Louis Blanchard et al. *Hierarchical Optimization in Isogeometric Analysis : Application to Thermal Conduction*. Research report 7470. Dec. 2010, p. 16 (cit. on p. 58).

[35] Gp. Bonneau, S. Hahmann, and J. Marku. “Geometric construction of auxetic metamaterials”. In: *Computer Graphics Forum* 40.2 (May 2021), pp. 291–304. ISSN: 0167-7055, 1467-8659. DOI: 10.1111/cgf.142633 (cit. on pp. 1, 7).

[36] Allan F. Bower. *Applied mechanics of solids*. Boca Raton, Fla.: CRC Press, 2010. 794 pp. ISBN: 978-1-4398-0247-2 (cit. on p. 35).

[37] M. J. Box. "A New Method of Constrained Optimization and a Comparison With Other Methods". In: *The Computer Journal* 8.1 (Apr. 1, 1965), pp. 42–52. ISSN: 0010-4620, 1460-2067. DOI: 10.1093/comjnl/8.1.42 (cit. on p. 60).

[38] D Brackett, I Ashcroft, and R Hague. "Topology Optimization for Additive Manufacturing". In: 2011 international solid freeform fabrication symposium. University of Texas at Austin, 2011 (cit. on pp. 8, 78).

[39] A. Bressan and S. Takacs. "Sum-factorization techniques in Isogeometric Analysis". In: *Computer Methods in Applied Mechanics and Engineering* 352 (Aug. 2019), pp. 437–460. ISSN: 00457825. DOI: 10.1016/j.cma.2019.04.031 (cit. on p. 38).

[40] Ericka Brivadis et al. "Isogeometric mortar methods". In: *Computer Methods in Applied Mechanics and Engineering* 284 (Feb. 2015), pp. 292–319. ISSN: 00457825. DOI: 10.1016/j.cma.2014.09.012 (cit. on p. 42).

[41] Annalisa Buffa et al. "Mathematical Foundations of Adaptive Isogeometric Analysis". In: *Archives of Computational Methods in Engineering* 29.7 (Nov. 2022), pp. 4479–4555. ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-022-09752-5 (cit. on p. 39).

[42] Fabrizia Caiazzo, Vittorio Alfieri, and Brahim David Bujazha. "Additive manufacturing of biomorphic scaffolds for bone tissue engineering". In: *The International Journal of Advanced Manufacturing Technology* 113.9 (Apr. 2021), pp. 2909–2923. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/s00170-021-06773-5 (cit. on p. 1).

[43] F. Calabrò, G. Sangalli, and M. Tani. "Fast formation of isogeometric Galerkin matrices by weighted quadrature". In: *Computer Methods in Applied Mechanics and Engineering* 316 (Apr. 2017), pp. 606–622. ISSN: 00457825. DOI: 10.1016/j.cma.2016.09.013 (cit. on pp. 38, 42).

[44] Bruno Castanie, Christophe Bouvet, and Malo Ginot. "Review of composite sandwich structure in aeronautic applications". In: *Composites Part C: Open Access* 1 (Aug. 2020), p. 100004. ISSN: 26666820. DOI: 10.1016/j.jcomc.2020.100004 (cit. on p. 1).

[45] Ludovic Chamoin. "Stratégie de vérification et validation de modèles dédiée au calcul de quantités d'intérêt en ingénierie mécanique". PhD thesis. Ecole Normale Supérieure de Cachan, 2013 (cit. on p. 50).

[46] Xianming Chen, Richard F Riesenfeld, and Elaine Cohen. "An algorithm for direct multiplication of B-splines". In: *IEEE transactions on automation science and engineering* 6.3 (2009), pp. 433–442 (cit. on p. 15).

[47] Myung-Jin Choi and Seonho Cho. "Constrained isogeometric design optimization of lattice structures on curved surfaces: computation of design velocity field". In: *Structural and Multidisciplinary Optimization* 58.1 (July 2018), pp. 17–34. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-018-2000-9 (cit. on p. 47).

[48] Chan I. Chung. *Extrusion of Polymers: Theory & Practice*. 3rd ed. München: Carl Hanser Verlag GmbH & Co. KG, Oct. 7, 2019. ISBN: 978-1-56990-609-5. DOI: 10.3139/9781569907382 (cit. on p. 64).

[49] Philippe Gaston Ciarlet. *Mathematical elasticity*. Studies in mathematics and its applications v. 20, 27, 29. Amsterdam New York New York, N.Y., U.S.A: North-Holland Sole distributors for the U.S.A. and Canada, Elsevier Science Pub. Co, 1988. ISBN: 0-444-70259-8 (cit. on p. 35).

[50] E. Cohen et al. "Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 199.5 (Jan. 2010), pp. 334–356. ISSN: 00457825. DOI: 10.1016/j.cma.2009.09.010 (cit. on p. 39).

[51] Elaine Cohen, Tom Lyche, and Larry L Schumaker. "Degree raising for splines". In: *Journal of Approximation Theory* 46.2 (1986), pp. 170–181 (cit. on p. 14).

[52] Elaine Cohen, Richard F Riesenfeld, and Gershon Elber. *Geometric modeling with splines: an introduction*. CRC Press, 2001 (cit. on pp. 17, 19).

[53] J Austin Cottrell, Thomas JR Hughes, and Yuri Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009 (cit. on pp. 2, 37, 39, 40).

[54] J.A. Cottrell, T.J.R. Hughes, and A. Reali. "Studies of refinement and continuity in isogeometric structural analysis". In: *Computer Methods in Applied Mechanics and Engineering* 196.41 (Sept. 2007), pp. 4160–4183. ISSN: 00457825. DOI: 10.1016/j.cma.2007.04.007 (cit. on pp. 37, 38).

[55] J.A. Cottrell et al. "Isogeometric analysis of structural vibrations". In: *Computer Methods in Applied Mechanics and Engineering* 195.41 (Aug. 2006), pp. 5257–5296. ISSN: 00457825. DOI: 10.1016/j.cma.2005.09.027 (cit. on p. 38).

[56] L. Beirão Da Veiga et al. "Mathematical analysis of variational isogeometric methods". In: *Acta Numerica* 23 (May 2014), pp. 157–287. ISSN: 0962-4929, 1474-0508. DOI: 10.1017/S096249291400004X (cit. on p. 38).

[57] Sumita Dahiya, Avi Shein, and Gershon Elber. "Shell-lattice construction based on regular and semi-regular tiling via functional composition". In: *Proc. of SMI'2021 fabrication and sculpting event*. 2021, pp. 12–23 (cit. on p. 22).

[58] Marwan Darwish and Fadl Moukalled. *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. Springer, 2016 (cit. on p. 37).

[59] Sachin D. Daxini and Jagdish M. Prajapati. "Parametric shape optimization techniques based on Meshless methods: A review". In: *Structural and Multidisciplinary Optimization* 56.5 (Nov. 2017), pp. 1197–1214. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-017-1702-8 (cit. on p. 43).

[60] T. Dbouk. "A review about the engineering design of optimal heat transfer systems using topology optimization". In: *Applied Thermal Engineering* 112 (Feb. 2017), pp. 841–854. ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2016.10.134 (cit. on p. 57).

[61] Ron S. Dembo and Trond Steihaug. "Truncated-Newton algorithms for large-scale unconstrained optimization". In: *Mathematical Programming* 26.2 (June 1983), pp. 190–212. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/BF02592055 (cit. on p. 61).

[62] Tony D DeRose et al. “Functional composition algorithms via blossoming”. In: *ACM Transactions on Graphics (TOG)* 12.2 (1993), pp. 113–135 (cit. on p. 15).

[63] V S Deshpande, N A Fleck, and M F Ashby. “Effective properties of the octet-truss lattice material”. In: *J. Mech. Phys. Solids* (2001) (cit. on p. 7).

[64] Nathaniel Després et al. “Evolutionary computation to design additively manufactured optimal heterogeneous lattice structures”. In: *Progress in Additive Manufacturing* 8.3 (June 2023), pp. 615–627. ISSN: 2363-9512, 2363-9520. DOI: 10.1007/s40964-022-00352-0 (cit. on p. 49).

[65] Sai C. Divi et al. “Error-estimate-based adaptive integration for immersed isogeometric analysis”. In: *Computers & Mathematics with Applications* 80.11 (Dec. 2020), pp. 2481–2516. ISSN: 08981221. DOI: 10.1016/j.camwa.2020.03.026 (cit. on p. 39).

[66] Tor Dokken, Vibeke Skytt, and Oliver Barrowclough. “Trivariate spline representations for computer aided design and additive manufacturing”. In: *Computers & Mathematics with Applications* 78.7 (2019), pp. 2168–2182 (cit. on p. 39).

[67] Jean Donea and Antonio Huerta. *Finite element methods for flow problems*. John Wiley & Sons, 2003 (cit. on p. 37).

[68] Alberto Donoso. “Numerical Simulations in 3D Heat Conduction: Minimizing the Quadratic Mean Temperature Gradient by an Optimality Criteria Method”. In: *SIAM Journal on Scientific Computing* 28.3 (Jan. 2006), pp. 929–941. ISSN: 1064-8275, 1095-7197. DOI: 10.1137/060650453 (cit. on p. 58).

[69] Anton Du Plessis et al. “Properties and applications of additively manufactured metallic cellular materials: A review”. In: *Progress in Materials Science* 125 (Apr. 2022), p. 100918. ISSN: 00796425. DOI: 10.1016/j.pmatsci.2021.100918 (cit. on p. 8).

[70] David Eberly. *Derivative Approximation by Finite Differences*. 2020 (cit. on p. 49).

[71] Gershon Elber. “Free form surface analysis using a hybrid of symbolic and numeric computation”. PhD thesis. Dept. of Computer Science, University of Utah, 1992 (cit. on pp. 14, 15).

[72] Gershon Elber. “Precise construction of micro-structures and porous geometry via functional composition”. In: *Mathematical Methods for Curves and Surfaces: 9th International Conference, MMCS 2016, Tønsberg, Norway, June 23–28, 2016, Revised Selected Papers 9*. Springer. 2017, pp. 108–125 (cit. on pp. 2, 9, 19, 22).

[73] Gershon Elber. “A Review of a B-spline based Volumetric Representation: Design, Analysis and Fabrication of Porous and/or Heterogeneous Geometries”. In: *Computer-Aided Design* 163 (Oct. 2023). ISSN: 00104485. DOI: 10.1016/j.cad.2023.103587 (cit. on p. 7).

[74] Gershon Elber. “Heterogeneous Conforming Compliant Microstructure Mechanisms”. In: (2023) (cit. on p. 7).

[75] Gershon Elber and Myung-Soo Kim. “Modeling by composition”. In: *Computer-Aided Design* 46 (2014), pp. 200–204 (cit. on p. 15).

[76] S. Elgeti et al. "Numerical shape optimization as an approach to extrusion die design". In: *Finite Elements in Analysis and Design* 61 (Nov. 2012), pp. 35–43. ISSN: 0168874X. DOI: 10.1016/j.finel.2012.06.008 (cit. on p. 66).

[77] Sarah Engelbrecht et al. "Cellular Structures for Optimal Performance". In: International Solid Freeform Fabrication Symposium. University of Texas at Austin, 2009 (cit. on p. 8).

[78] Jost-Hinrich Eschenburg and Jürgen Jost. *Differentialgeometrie und Minimalflächen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. ISBN: 978-3-642-38522-3. DOI: 10.1007/978-3-642-38522-3 (cit. on p. 52).

[79] H.J. Ettinger et al. "Parameterization and optimization strategies for the automated design of uPVC profile extrusion dies". In: *Structural and Multidisciplinary Optimization* 28.2 (Sept. 2004). ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-004-0440-x (cit. on pp. 43, 66).

[80] Ben Ezair, Daniel Dikovsky, and Gershon Elber. "Fabricating functionally graded material objects using trimmed trivariate volumetric representations". In: *Proceedings of SMI*. University of California Berkeley, CA, USA. 2017 (cit. on p. 24).

[81] Gerald E Farin. *Curves and surfaces for CAGD: a practical guide*. Morgan Kaufmann, 2002 (cit. on pp. 10, 19).

[82] Rossana R. Fernandes and Ali Y. Tamijani. "Design optimization of lattice structures with stress constraints". In: *Materials & Design* 210 (Nov. 2021), p. 110026. ISSN: 02641275. DOI: 10.1016/j.matdes.2021.110026 (cit. on p. 76).

[83] Carlo Giovanni Ferro et al. "Lattice structured impact absorber with embedded anti-icing system for aircraft wings fabricated with additive SLM process". In: *Materials Today Communications* 15 (June 2018), pp. 185–189. ISSN: 23524928. DOI: 10.1016/j.mtcomm.2018.03.007 (cit. on p. 7).

[84] Nicola Ferro et al. "Design of cellular materials for multiscale topology optimization: application to patient-specific orthopedic devices". In: *Structural and Multidisciplinary Optimization* 65.3 (Mar. 2022), p. 79. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-021-03163-z (cit. on pp. 1, 7).

[85] Clemens Fricke et al. "Investigation of reinforcement learning for shape optimization of 2D profile extrusion die geometries". In: *Advances in Computational Science and Engineering* 1.1 (2023), pp. 1–35. DOI: 10.3934/acse.2023001 (cit. on p. 49).

[86] D. Fußeder, B. Simeon, and A.-V. Vuong. "Fundamental aspects of shape optimization in the context of isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 286 (Apr. 2015), pp. 313–331. ISSN: 00457825. DOI: 10.1016/j.cma.2014.12.028 (cit. on p. 47).

[87] Jie Gao et al. "Topology optimization for auxetic metamaterials based on isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 352 (Aug. 2019), pp. 211–236. ISSN: 00457825. DOI: 10.1016/j.cma.2019.04.021 (cit. on pp. 1, 47).

[88] T. Gao et al. "Topology optimization of heat conduction problem involving design-dependent heat load effect". In: *Finite Elements in Analysis and Design* 44.14 (Oct. 2008), pp. 805–813. ISSN: 0168874X. DOI: 10.1016/j.finel.2008.06.001 (cit. on p. 58).

[89] C Giannelli et al. "Leveraging moving parameterization and THB-splines for CAD surface reconstruction". In: (2023) (cit. on p. 10).

[90] Carlotta Giannelli et al. "THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 299 (2016), pp. 337–365 (cit. on p. 10).

[91] Lorna J. Gibson. "Biomechanics of cellular solids". In: *Journal of Biomechanics* 38.3 (Mar. 2005), pp. 377–399. ISSN: 00219290. DOI: 10.1016/j.jbiomech.2004.09.027 (cit. on p. 7).

[92] Lorna J. Gibson and Michael F. Ashby. *Cellular Solids: Structure and Properties*. 2nd ed. Cambridge University Press, May 1, 1997. ISBN: 978-0-521-49911-8. DOI: 10.1017/CBO9781139878326 (cit. on pp. 1, 7).

[93] Michael B. Giles and Niles A. Pierce. "An Introduction to the Adjoint Approach to Design". In: *Flow, Turbulence and Combustion* 65.3 (2000), pp. 393–415. ISSN: 13866184. DOI: 10.1023/A:1011430410075 (cit. on p. 50).

[94] Jens Gravesen et al. "Planar parametrization in isogeometric analysis". In: *Mathematical Methods for Curves and Surfaces: 8th International Conference, MMCS 2012, Oslo, Norway, June 28–July 3, 2012, Revised Selected Papers 8*. Springer. 2014, pp. 189–212 (cit. on p. 23).

[95] Josef Griessmair and Werner Purgathofer. "Deformation of Solids with Trivariate B-Splines." In: *Eurographics*. Vol. 89. 1989, pp. 137–148 (cit. on p. 15).

[96] Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. SIAM, 2008 (cit. on pp. 30, 49).

[97] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. 2010 (cit. on p. 42).

[98] Vibhushit Gupta et al. "An Insight on NURBS Based Isogeometric Analysis, Its Current Status and Involvement in Mechanical Applications". In: *Archives of Computational Methods in Engineering* 30.2 (Mar. 2023), pp. 1187–1230. ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-022-09838-0 (cit. on p. 39).

[99] Raphael T. Haftka and Zafer Gürdal. *Elements of Structural Optimization*. Red. by G. M. L. Gladwell. Vol. 11. Solid Mechanics And Its Applications. Dordrecht: Springer Netherlands, 1992. ISBN: 978-0-7923-1505-6. DOI: 10.1007/978-94-011-2550-5 (cit. on p. 75).

[100] René R. Hiemstra et al. "Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 316 (Apr. 2017), pp. 966–1004. ISSN: 00457825. DOI: 10.1016/j.cma.2016.10.049 (cit. on p. 42).

[101] Jochen Hinz, Matthias Möller, and Cornelis Vuik. "Elliptic grid generation techniques in the framework of isogeometric analysis applications". In: *Computer Aided Geometric Design* 65 (2018), pp. 48–75 (cit. on p. 39).

[102] T Hirschler et al. "The embedded isogeometric Kirchhoff–Love shell: From design to shape optimization of non-conforming stiffened multipatch structures". In: *Computer Methods in Applied Mechanics and Engineering* 349 (2019), pp. 774–797 (cit. on pp. 9, 38).

[103] T. Hirschler, P. Antolin, and A. Buffa. "Fast and multiscale formation of isogeometric matrices of microstructured geometric models". In: *Computational Mechanics* 69.2 (Feb. 2022), pp. 439–466. ISSN: 0178-7675, 1432-0924. DOI: 10.1007/s00466-021-02098-y (cit. on pp. 56, 82, 88).

[104] T. Hirschler et al. "Isogeometric sizing and shape optimization of thin structures with a solid-shell approach". In: *Structural and Multidisciplinary Optimization* 59.3 (Mar. 2019), pp. 767–785. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-018-2100-6 (cit. on p. 38).

[105] T. Hirschler et al. "Reduced order modeling based inexact FETI-DP solver for lattice structures". In: *International Journal for Numerical Methods in Engineering* (Jan. 22, 2024), e7419. ISSN: 0029-5981, 1097-0207. DOI: 10.1002/nme.7419 (cit. on pp. 56, 88).

[106] J.Y. Ho, K.C. Leong, and T.N. Wong. "Additively-manufactured metallic porous lattice heat exchangers for air-side heat transfer enhancement". In: *International Journal of Heat and Mass Transfer* 150 (Apr. 2020), p. 119262. ISSN: 00179310. DOI: 10.1016/j.ijheatmasstransfer.2019.119262 (cit. on p. 1).

[107] Christoph Hofer. "Fast Multipatch Isogeometric Analysis Solvers". Dissertation. Johannes Kepler Universität Linz, 2018 (cit. on p. 42).

[108] Paul Hoffmann. "Thermo-mechanically induced fatigue damage in thin metal films: simulation, comparison to experiments, and fatigue damage parameter calibration". PhD thesis. Technische Universität Wien, 2023 (cit. on p. 35).

[109] Gerhard A Holzapfel. *Nonlinear solid mechanics: a continuum approach for engineering science*. 2002 (cit. on p. 35).

[110] Q Youn Hong and Gershon Elber. "Conformal microstructure synthesis in trimmed trivariate based v-reps". In: *Computer-Aided Design* 140 (2021), p. 103085 (cit. on pp. 22, 88).

[111] Q Youn Hong, Gershon Elber, and Myung-Soo Kim. "Implicit Functionally Graded Conforming Microstructures". In: *Computer-Aided Design* 162 (2023), p. 103548 (cit. on pp. 2, 22).

[112] Christian Hopmann et al. *Extrusion dies for plastics and rubber: design and engineering computations*. 4th Edition. Cincinnati: Hanser Publictions, 2016. 451 pp. ISBN: 978-1-56990-623-1 (cit. on p. 66).

[113] Chuanfeng Hu et al. "Isogeometric Analysis-Based Topological Optimization for Heterogeneous Parametric Porous Structures". In: *Journal of Systems Science and Complexity* 36.1 (Feb. 2023), pp. 29–52. ISSN: 1009-6124, 1559-7067. DOI: 10.1007/s11424-022-1290-6 (cit. on p. 48).

[114] Sebastian Hube, Roxana Pohlmann, and Stefanie Elgeti. "Seamless Integration of Analysis and Design: Automatic CAD Reconstruction of Post-Analysis Geometries". In: *arXiv preprint arXiv:2205.04356* (2022) (cit. on p. 15).

[115] Sebastian Andreas Hube. "Numerical shape optimization for dynamic mixing elements in single-screw extruders". Doctoral Thesis. Aachen: RWTH Aachen University, July 6, 2023. 124 pp. (cit. on p. 66).

[116] Thomas J.R. Hughes, John A. Evans, and Alessandro Reali. "Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems". In: *Computer Methods in Applied Mechanics and Engineering* 272 (Apr. 2014), pp. 290–320. ISSN: 00457825. DOI: 10.1016/j.cma.2013.11.012 (cit. on pp. 2, 38).

[117] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012 (cit. on p. 37).

[118] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement". In: *Computer methods in applied mechanics and engineering* 194.39-41 (2005), pp. 4135–4195 (cit. on pp. 3, 37).

[119] Ye Ji et al. "Constructing high-quality planar NURBS parameterization for isogeometric analysis by adjustment control points and weights". In: *Journal of Computational and Applied Mathematics* 396 (Nov. 2021), p. 113615. ISSN: 03770427. DOI: 10.1016/j.cam.2021.113615 (cit. on p. 42).

[120] Ye Ji et al. "Penalty function-based volumetric parameterization method for isogeometric analysis". In: *Computer Aided Geometric Design* 94 (Mar. 2022), p. 102081. ISSN: 01678396. DOI: 10.1016/j.cagd.2022.102081 (cit. on p. 42).

[121] Ye Ji et al. "On an improved PDE-based elliptic parameterization method for isogeometric analysis using preconditioned Anderson acceleration". In: *Computer Aided Geometric Design* 102 (2023), p. 102191 (cit. on p. 39).

[122] Steven G. Johnson. *The NLopt nonlinear-optimization package*. https://github.com/stevengj/nlopt. 2007 (cit. on pp. 50, 64).

[123] Bert Jüttler et al. "Geometry + Simulation Modules: Implementing Isogeometric Analysis". In: *PAMM* 14.1 (Dec. 2014), pp. 961–962. ISSN: 1617-7061, 1617-7061. DOI: 10.1002/pamm.201410461 (cit. on pp. 42, 49).

[124] Couro Kane and Marc Schoenauer. "Topological optimum design using genetic algorithms". In: *Control and Cybernetics* 25.5 (1996), pp. 1059–1088 (cit. on p. 48).

[125] Josef Kiendl et al. "Isogeometric Kirchhoff–Love shell formulations for general hyperelastic materials". In: *Computer Methods in Applied Mechanics and Engineering* 291 (July 2015), pp. 280–303. ISSN: 00457825. DOI: 10.1016/j.cma.2015.03.010 (cit. on p. 39).

[126] Robin Kleer and Frank T. Piller. "Local manufacturing and structural shifts in competition: Market dynamics of additive manufacturing". In: *International Journal of Production Economics* 216 (Oct. 2019), pp. 23–34. ISSN: 09255273. DOI: 10.1016/j.ijpe.2019.04.019 (cit. on p. 1).

[127] Dennis M Kochmann and Katia Bertoldi. "Exploiting microstructural instabilities in solids and structures: from metamaterials to structural transitions". In: *Applied mechanics reviews* 69.5 (2017), p. 050801 (cit. on p. 7).

[128] Michael Kofler, Jacques Zwar, and Stefanie Elgeti. "Lightweight Design Principles: Numerical Optimization of Lattice Structures". In: *Encyclopedia of Continuum Mechanics. Berlin: Springer* (in review), pp. 156–182 (cit. on pp. 7, 8).

[129] Anwar Koshakji, Alfio Quarteroni, and Gianluigi Rozza. "Free form deformation techniques applied to 3D shape optimization problems". In: *Communications in Applied and Industrial Mathematics* 4.452 (2013). DOI: https://doi.org/10.1685/journal.caim.452 (cit. on p. 19).

[130] Dieter Kraft. *A software package for sequential quadratic programming*. Research Report 88-28. Oberpfaffenhofen: Institut für Dynamik der Flugsysteme, July 1988, p. 33 (cit. on p. 61).

[131] H W Kuhn and A W Tucker. "NONLINEAR PROGRAMMING". In: *Nonlinear Programming*. Proceedings of 2nd Berkeley Symposium. Berkeley: Berkeley: University of California Press, 1951, pp. 481–492 (cit. on p. 46).

[132] W Michael Lai, David Rubin, and Erhard Krempl. *Introduction to continuum mechanics*. Butterworth-Heinemann, 2009 (cit. on p. 35).

[133] Hans Petter Langtangen and Svein Linge. *Finite difference computing with PDEs: a modern software approach*. Springer Nature, 2017 (cit. on p. 37).

[134] Nadhir Lebaal, Fabrice Schmidt, and Stephan Puissant. "Design and optimization of three-dimensional extrusion dies, using constraint optimization algorithm". In: *Finite Elements in Analysis and Design* 45.5 (2009), pp. 333–340 (cit. on p. 66).

[135] Nadhir Lebaal, Fabrice Schmidt, and Stephan Puissant. "Optimisation of extrusion flat die design and die wall temperature distribution, using Kriging and response surface method". In: *International Journal of Materials and Product Technology* 38.2 (2010), p. 307. ISSN: 0268-1900, 1741-5209. DOI: 10.1504/IJMPT.2010.032107 (cit. on pp. 67, 69).

[136] Jaewook Lee et al. *tataratat/splinepy: 0.0.26*. https://doi.org/10.5281/zenodo.8091753. Version v0.0.26. June 2023. DOI: 10.5281/zenodo.8091753 (cit. on pp. 11, 12).

[137] Dawei Li et al. "Optimal design and modeling of gyroid-based functionally graded cellular structures for additive manufacturing". In: *Computer-Aided Design* 104 (Nov. 2018), pp. 87–99. ISSN: 00104485. DOI: 10.1016/j.cad.2018.06.003 (cit. on pp. 2, 8, 45).

[138] Shuaishuai Li et al. "Design for additive manufacturing from a force-flow perspective". In: *Materials & Design* 204 (June 2021), p. 109664. ISSN: 02641275. DOI: 10.1016/j.matdes.2021.109664 (cit. on pp. 7, 78).

[139] Asger Limkilde. "Efficient 3D Shape Optimization with Isogeometric Analysis". PhD thesis. Technical University of Denmark, 2020 (cit. on p. 52).

[140] Asger Limkilde et al. "Practical isogeometric shape optimization: parametrization by means of regularization". In: *Journal of computational design and engineering* 8.2 (2021), pp. 547–558 (cit. on p. 23).

[141] Jeffrey I. Lipton et al. "Additive manufacturing for the food industry". In: *Trends in Food Science & Technology* 43.1 (May 2015), pp. 114–123. ISSN: 09242244. DOI: 10.1016/j.tifs.2015.02.004 (cit. on p. 1).

[142] Dong C. Liu and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Mathematical Programming* 45.1 (Aug. 1989), pp. 503–528. ISSN: 0025-5610, 1436-4646. DOI: 10.1007/BF01589116 (cit. on p. 61).

[143] Yuan Liu et al. "A Survey of Modeling and Optimization Methods for Multi-Scale Heterogeneous Lattice Structures". In: *Journal of Mechanical Design* 143.4 (Apr. 1, 2021), p. 040803. ISSN: 1050-0472, 1528-9001. DOI: 10.1115/1.4047917 (cit. on p. 8).

[144] J David Logan. *Applied partial differential equations*. Springer, 2014 (cit. on p. 33).

[145] J. Lotz, M. ten Eikelder, and I. Akkerman. *A space-time framework for periodic flows with applications to hydrofoils*. Nov. 20, 2022 (cit. on p. 38).

[146] Tom Lyche and Knut Mørken. "Knot removal for parametric B-spline curves and surfaces". In: *Computer Aided Geometric Design* 4.3 (1987), pp. 217–230 (cit. on p. 22).

[147] David S. Makhija and Philip S. Beran. "Concurrent shape and topology optimization for steady conjugate heat transfer". In: *Structural and Multidisciplinary Optimization* 59.3 (Mar. 2019), pp. 919–940. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-018-2110-4 (cit. on p. 57).

[148] Angelos Mantzaflaris et al. "Low rank tensor methods in Galerkin-based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 1062–1085 (cit. on p. 18).

[149] Angelos Mantzaflaris et al. "Low rank tensor methods in Galerkin-based isogeometric analysis". In: *Computer Methods in Applied Mechanics and Engineering* 316 (Apr. 2017), pp. 1062–1085. ISSN: 00457825. DOI: 10.1016/j.cma.2016.11.013 (cit. on p. 19).

[150] Gilles Marck et al. "Topology Optimization Using the SIMP Method for Multiobjective Conductive Problems". In: *Numerical Heat Transfer, Part B: Fundamentals* 61.6 (June 2012), pp. 439–470. ISSN: 1040-7790, 1521-0626. DOI: 10.1080/10407790.2012.687979 (cit. on p. 58).

[151] Michelangelo Marsala, Angelos Mantzaflaris, and Bernard Mourrain. "G1 – Smooth biquintic approximation of Catmull-Clark subdivision surfaces". In: *Computer Aided Geometric Design* 99 (Nov. 2022), p. 102158. ISSN: 01678396. DOI: 10.1016/j.cagd.2022.102158 (cit. on p. 38).

[152] Jerrold E Marsden and Thomas JR Hughes. *Mathematical foundations of elasticity*. Courier Corporation, 1994 (cit. on p. 34).

[153] Tobias Martin, Elaine Cohen, and Mike Kirby. "Volumetric parameterization and trivariate B-spline fitting using harmonic functions". In: *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pp. 269–280 (cit. on p. 23).

[154] Benjamin Marussig and Thomas J. R. Hughes. "A Review of Trimming in Isogeometric Analysis: Challenges, Data Exchange and Simulation Aspects". In: *Archives of Computational Methods in Engineering* 25.4 (Nov. 2018), pp. 1059–1127. ISSN: 1134-3060, 1886-1784. DOI: 10.1007/s11831-017-9220-9 (cit. on p. 39).

[155] I. Maskery et al. "Insights into the mechanical properties of several triply periodic minimal surface lattice structures made by polymer additive manufacturing". In: *Polymer* 152 (Sept. 2018), pp. 62–71. ISSN: 00323861. DOI: 10.1016/j.polymer.2017.11.049 (cit. on pp. 2, 22).

[156] Fady Massarwi, Pablo Antolin, and Gershon Elber. "Volumetric untrimming: Precise decomposition of trimmed trivariates into tensor products". In: *Computer Aided Geometric Design* 71 (May 2019), pp. 1–15. ISSN: 01678396. DOI: 10.1016/j.cagd.2019.04.005 (cit. on p. 39).

[157] Fady Massarwi and Gershon Elber. "A B-spline based framework for volumetric object modeling". In: *Computer-Aided Design* 78 (2016), pp. 36–47 (cit. on p. 39).

[158] Fady Massarwi et al. "Hierarchical, random and bifurcation tiling with heterogeneity in micro-structures construction via functional composition". In: *Computer-Aided Design* 102 (2018), pp. 148–159 (cit. on pp. 22, 88).

[159] Thomas McSherry. "A General Steepest Descent Algorithm". In: *IEEE Transactions on Aerospace and Electronic Systems* AES-12.1 (Jan. 1976), pp. 12–22. ISSN: 0018-9251. DOI: 10.1109/TAES.1976.308210 (cit. on p. 49).

[160] Cornel Marius Murea and Dan Tiba. "Topological optimization and minimal compliance in linear elasticity". In: *Evolution Equations & Control Theory* 9.4 (2020), pp. 1115–1131. ISSN: 2163-2480. DOI: 10.3934/eect.2020043 (cit. on p. 76).

[161] Uwe Naumann. *The Art of Differentiating Computer Programs: An Introduction to Algorithmic Differentiation*. Society for Industrial and Applied Mathematics, Jan. 2011. ISBN: 978-1-61197-206-1. DOI: 10.1137/1.9781611972078 (cit. on p. 49).

[162] Uwe Naumann. *The art of differentiating computer programs: an introduction to algorithmic differentiation*. SIAM, 2011 (cit. on p. 30).

[163] John A. Nelder and Roger Mead. "A simplex method for function minimization". In: *The Computer Journal* 7.4 (1965), pp. 308–313 (cit. on p. 60).

[164] Chuong Nguyen et al. "Three-dimensional topology optimization of auxetic metamaterial using isogeometric analysis and model order reduction". In: *Computer Methods in Applied Mechanics and Engineering* 371 (Nov. 2020), p. 113306. ISSN: 00457825. DOI: 10.1016/j.cma.2020.113306 (cit. on p. 47).

[165] Cong Hong Phong Nguyen and Young Choi. "Concurrent density distribution and build orientation optimization of additively manufactured functionally graded lattice structures". In: *Computer-Aided Design* 127 (Oct. 2020), p. 102884. ISSN: 00104485. DOI: 10.1016/j.cad.2020.102884 (cit. on p. 44).

[166] Jason Nguyen, Sang-in Park, and David Rosen. "Heuristic optimization method for cellular structure design of light weight components". In: *International Journal of Precision Engineering and Manufacturing* 14.6 (June 2013), pp. 1071–1078. ISSN: 2234-7593, 2005-4602. DOI: 10.1007/s12541-013-0144-5 (cit. on p. 44).

[167] Jason Nguyen et al. “Conformal Lattice Structure Design and Fabrication”. In: University of Texas at Austin, 2012 (cit. on p. 8).

[168] Vinh Phu Nguyen, Stéphane P. A. Bordas, and Timon Rabczuk. “Isogeometric analysis: an overview and computer implementation aspects”. In: *Mathematics and Computers in Simulation* 117 (Nov. 2015), pp. 89–116. ISSN: 03784754. DOI: 10.1016/j.matcom.2015.05.008 (cit. on p. 39).

[169] Vinh Phu Nguyen et al. “Nitsche’s method for two and three dimensional NURBS patch coupling”. In: *Computational Mechanics* 53.6 (June 2014), pp. 1163–1182. ISSN: 0178-7675, 1432-0924. DOI: 10.1007/s00466-013-0955-3 (cit. on p. 42).

[170] Vuong Nguyen-Van et al. “Bioinspired cellular cementitious structures for prefabricated construction: Hybrid design & performance evaluations”. In: *Automation in Construction* 119 (Nov. 2020), p. 103324. ISSN: 09265805. DOI: 10.1016/j.autcon.2020.103324 (cit. on p. 1).

[171] J. M. Nóbrega et al. “Flow Balancing in Extrusion Dies for Thermoplastic Profiles: Part I: Automatic Design”. In: *International Polymer Processing* 18.3 (Sept. 1, 2003), pp. 298–306. ISSN: 2195-8602, 0930-777X. DOI: 10.3139/217.1745 (cit. on p. 66).

[172] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. 2nd ed. Springer series in operations research. OCLC: ocm68629100. New York: Springer, 2006. 664 pp. ISBN: 978-0-387-30303-1 (cit. on pp. 47, 61).

[173] Peter Nørtoft and Jens Gravesen. “Isogeometric shape optimization in fluid mechanics”. In: *Structural and Multidisciplinary Optimization* 48.5 (Nov. 2013), pp. 909–925. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-013-0931-8 (cit. on p. 38).

[174] Bastian Oesterle et al. “A study on the approximation power of NURBS and the significance of exact geometry in isogeometric pre-buckling analyses of shells”. In: *Computer Methods in Applied Mechanics and Engineering* 397 (July 2022), p. 115144. ISSN: 00457825. DOI: 10.1016/j.cma.2022.115144 (cit. on p. 38).

[175] Tim A. Osswald and Natalie Rudolph. *Polymer rheology: fundamentals and applications*. München: Hanser, 2015. 225 pp. ISBN: 978-1-56990-517-3 (cit. on p. 64).

[176] Alexander Paolini, Stefan Kollmannsberger, and Ernst Rank. “Additive manufacturing in construction: A review on processes, applications, and digital planning methods”. In: *Additive Manufacturing* 30 (Dec. 2019), p. 100894. ISSN: 22148604. DOI: 10.1016/j.addma.2019.100894 (cit. on p. 1).

[177] Byong-Ug Park et al. “Shape optimization of the stokes flow problem based on isogeometric analysis”. In: *Structural and Multidisciplinary Optimization* 48.5 (Nov. 2013), pp. 965–977. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-013-0939-0 (cit. on p. 38).

[178] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 1996 (cit. on pp. 10, 12, 13, 19, 23).

[179] Les Piegl and Wayne Tiller. “Symbolic operators for NURBS”. In: *Computer-Aided Design* 29.5 (1997), pp. 361–368 (cit. on pp. 15, 18).

[180] J F T Pittman. “Computer-aided design and optimization of profile extrusion dies for thermoplastics and rubber: a review”. In: *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* 225.4 (Nov. 2011), pp. 280–321. ISSN: 0954-4089, 2041-3009. DOI: 10.1177/0954408911415324 (cit. on p. 64).

[181] János Plocher and Ajit Panesar. “Review on design and structural optimisation in additive manufacturing: Towards next-generation lightweight structures”. In: *Materials & Design* 183 (Dec. 2019), p. 108164. ISSN: 02641275. DOI: 10.1016/j.matdes.2019.108164 (cit. on pp. 8, 78).

[182] M J D Powell. “A view of algorithms for optimization without derivatives”. In: *Mathematics Today-Bulletin of the Institute of Mathematics and its Applications* 43.5 (2007), pp. 270–274 (cit. on p. 49).

[183] M. J. D. Powell. “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation”. In: *Advances in Optimization and Numerical Analysis*. Ed. by Susana Gomez and Jean-Pierre Hennart. Dordrecht: Springer Netherlands, 1994, pp. 51–67. ISBN: 978-94-015-8330-5. DOI: 10.1007/978-94-015-8330-5_4 (cit. on p. 60).

[184] Michael J. D. Powell. “The BOBYQA algorithm for bound constrained optimization without derivatives”. In: *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge* 26 (2009), pp. 26–46 (cit. on p. 60).

[185] GNU Project. *GNU FreeFont*. 2002. URL: https://www.gnu.org/software/freefont/ (visited on 03/18/2024) (cit. on p. 42).

[186] O.M. Querin, G.P. Steven, and Y.M. Xie. “Evolutionary structural optimisation using an additive algorithm”. In: *Finite Elements in Analysis and Design* 34.3 (Feb. 2000), pp. 291–308. ISSN: 0168874X. DOI: 10.1016/S0168-874X(99)00044-X (cit. on p. 48).

[187] Richard Franklin Riesenfeld. *Applications of b-spline approximation to geometric problems of computer-aided design*. Syracuse University, 1973 (cit. on p. 9).

[188] J. Robbins et al. “An efficient and scalable approach for generating topologically optimized cellular structures for additive manufacturing”. In: *Additive Manufacturing* 12 (Oct. 2016), pp. 296–304. ISSN: 22148604. DOI: 10.1016/j.addma.2016.06.013 (cit. on p. 8).

[189] David F Rogers. *An introduction to NURBS: with historical perspective*. Morgan Kaufmann, 2001 (cit. on pp. 9, 12, 19).

[190] Filippo Salmoiraghi et al. “Isogeometric analysis-based reduced order modelling for incompressible linear viscous flows in parametrized shapes”. In: *Advanced Modeling and Simulation in Engineering Sciences* 3.1 (Dec. 2016), p. 21. ISSN: 2213-7467. DOI: 10.1186/s40323-016-0076-6 (cit. on p. 19).

[191] Espen Sande, Carla Manni, and Hendrik Speleers. “Explicit error estimates for spline approximation of arbitrary smoothness in isogeometric analysis”. In: *Numerische Mathematik* 144.4 (2020), pp. 889–929 (cit. on pp. 2, 38).

[192] Gianpaolo Savio et al. “Geometric Modeling of Cellular Materials for Additive Manufacturing in Biomedical Field: A Review”. In: *Applied Bionics and Biomechanics* 2018 (2018), pp. 1–14. ISSN: 1176-2322, 1754-2103. DOI: 10.1155/2018/1654782 (cit. on p. 8).

[193] Krishna Kumar Saxena, Emilio P Calius, and Raj Das. “Tailoring cellular auxetics for wearable applications with multimaterial 3D printing”. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 50633. American Society of Mechanical Engineers. 2016, V009T12A063 (cit. on p. 24).

[194] Tobias A. Schaedler and William B. Carter. “Architected Cellular Materials”. In: *Annual Review of Materials Research* 46.1 (July 1, 2016), pp. 187–210. ISSN: 1531-7331, 1545-4118. DOI: 10.1146/annurev-matsci-070115-031624 (cit. on p. 8).

[195] Dominik Schillinger et al. “An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces”. In: *Computer Methods in Applied Mechanics and Engineering* 249-252 (Dec. 2012), pp. 116–150. ISSN: 00457825. DOI: 10.1016/j.cma.2012.03.017 (cit. on p. 39).

[196] Dominik Schillinger et al. “Isogeometric Collocation: Cost Comparison with Galerkin Methods and Extension to Adaptive Hierarchical NURBS Discretizations”. In: *PAMM* 13.1 (Dec. 2013), pp. 107–108. ISSN: 1617-7061, 1617-7061. DOI: 10.1002/pamm.201310049 (cit. on p. 39).

[197] Lucien A Schmit. “Structural design by systematic synthesis”. In: 2nd Conference on electronic computation. Pittsburgh, Pa.: American Society of Civil Engineers, Sept. 9, 1960 (cit. on p. 43).

[198] Oliver Schmitt et al. “On the formulation and implementation of geometric and manufacturing constraints in node–based shape optimization”. In: *Structural and Multidisciplinary Optimization* 53.4 (Apr. 2016), pp. 881–892. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-015-1359-0 (cit. on p. 45).

[199] Thomas W Sederberg and Scott R Parry. “Free-form deformation of solid geometric models”. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 1986, pp. 151–160 (cit. on p. 15).

[200] Thomas W Sederberg and Alan K Zundel. “Scan line display of algebraic surfaces”. In: *ACM SIGGRAPH Computer Graphics* 23.3 (1989), pp. 147–156 (cit. on p. 15).

[201] Thomas W Sederberg et al. “T-spline simplification and local refinement”. In: *ACM transactions on graphics (TOG)* 23.3 (2004), pp. 276–283 (cit. on p. 10).

[202] Thomas W. Sederberg and Scott R. Parry. “Free-form deformation of solid geometric models”. In: *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. SIGGRAPH ’86. New York, NY, USA: Association for Computing Machinery, Aug. 31, 1986, pp. 151–160. ISBN: 978-0-89791-196-2. DOI: 10.1145/15922.15903 (cit. on p. 19).

[203] Roland Siegbert et al. “Comparing Optimization Algorithms for Shape Optimization of Extrusion Dies”. In: *PAMM* 14.1 (Dec. 2014), pp. 789–794. ISSN: 1617-7061, 1617-7061. DOI: 10.1002/pamm.201410377 (cit. on pp. 60, 61).

[204] O. Sigmund and J. Petersson. "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima". In: *Structural Optimization* 16.1 (Aug. 1998), pp. 68–75. DOI: 10.1007/BF01214002 (cit. on p. 86).

[205] Jarle Sogn and Stefan Takacs. "Multigrid solvers for isogeometric discretizations of the second biharmonic problem". In: *Mathematical Models and Methods in Applied Sciences* 33.9 (Aug. 2023), pp. 1803–1828. ISSN: 0218-2025, 1793-6314. DOI: 10.1142/S0218202523500422 (cit. on p. 38).

[206] Jan Sokolowski and Jean-Paul Zolesio. *Introduction to Shape Optimization*. Vol. 16. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. ISBN: 978-3-642-63471-0. DOI: 10.1007/978-3-642-58106-9 (cit. on p. 75).

[207] Boris van Sosin and Gershon Elber. "Crossing knot lines in composition of freeform B-spline geometry". In: *Computer Aided Geometric Design* 62 (2018), pp. 217–227 (cit. on p. 15).

[208] Boris van Sosin et al. "The construction of conforming-to-shape truss lattice structures via 3D sphere packing". In: *Computer-Aided Design* 132 (2021), p. 102962 (cit. on p. 9).

[209] Krister Svanberg. "A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations". In: *SIAM Journal on Optimization* 12.2 (Jan. 2002), pp. 555–573. ISSN: 1052-6234, 1095-7189. DOI: 10.1137/S1052623499362822 (cit. on p. 61).

[210] Francesco Tamburrino, Serena Graziosi, and Monica Bordegoni. "The design process of additively manufactured mesoscale lattice structures: a review". In: *Journal of Computing and Information Science in Engineering* 18.4 (2018), p. 040801 (cit. on p. 8).

[211] Yunlong Tang, Guoying Dong, and Yaoyao Fiona Zhao. "A hybrid geometric modeling method for lattice structures fabricated by additive manufacturing". In: *The International Journal of Advanced Manufacturing Technology* 102.9 (June 2019), pp. 4011–4030. ISSN: 0268-3768, 1433-3015. DOI: 10.1007/s00170-019-03308-x (cit. on pp. 2, 9).

[212] Yunlong Tang et al. "Lattice Structure Design and Optimization With Additive Manufacturing Constraints". In: *IEEE Transactions on Automation Science and Engineering* 15.4 (Oct. 2018), pp. 1546–1562. ISSN: 1545-5955, 1558-3783. DOI: 10.1109/TASE.2017.2685643 (cit. on pp. 45, 76).

[213] S Teufelhart and G Reinhart. "Optimization of Strut Diameters in Lattice Structures". In: 2012 International Solid Freeform Fabrication Symposium. University of Texas at Austin, 2012 (cit. on p. 43).

[214] Dominique Thévenin and Gábor Janiga, eds. *Optimization and Computational Fluid Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-72152-9. DOI: 10.1007/978-3-540-72153-6 (cit. on p. 49).

[215] Steffen Tillmann et al. "Shape-optimization of extrusion-dies via parameterized physics-informed neural networks". In: *PAMM* 23.4 (Dec. 2023), e202300203. ISSN: 1617-7061, 1617-7061. DOI: 10.1002/pamm.202300203 (cit. on p. 19).

[216] Antonios Tsourdos, Brian White, and Madhavan Shanmugavel. *Cooperative path planning of unmanned aerial vehicles*. Vol. 32. John Wiley & Sons, 2010 (cit. on p. 17).

[217] H. A. Van Der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (Mar. 1992), pp. 631–644. ISSN: 0196-5204, 2168-3417. DOI: 10.1137/0913035 (cit. on p. 42).

[218] H. M. Verhelst, J. H. Den Besten, and M. Möller. *An Adaptive Parallel Arc-Length Method*. Mar. 2, 2023 (cit. on p. 39).

[219] H. M. Verhelst et al. *Goal-Adaptive Meshing of Isogeometric Kirchhoff-Love Shells*. July 17, 2023 (cit. on p. 39).

[220] Kenneth James Versprille. *Computer-aided design applications of the rational b-spline approximation form*. Syracuse University, 1975 (cit. on p. 9).

[221] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2 (cit. on p. 50).

[222] Jan Vlček and Ladislav Lukšan. "Shifted limited-memory variable metric methods for large-scale unconstrained optimization". In: *Journal of Computational and Applied Mathematics* 186.2 (Feb. 2006), pp. 365–390. ISSN: 03770427. DOI: 10.1016/j.cam.2005.02.010 (cit. on pp. 49, 61).

[223] Wolfgang A. Wall, Moritz A. Frenzel, and Christian Cyron. "Isogeometric structural shape optimization". In: *Computer Methods in Applied Mechanics and Engineering* 197.33 (June 2008), pp. 2976–2988. ISSN: 00457825. DOI: 10.1016/j.cma.2008.01.025 (cit. on pp. 38, 77).

[224] Yingjun Wang, Hang Xu, and Damiano Pasini. "Multiscale isogeometric topology optimization for lattice materials". In: *Computer Methods in Applied Mechanics and Engineering* 316 (Apr. 2017), pp. 568–585. ISSN: 00457825. DOI: 10.1016/j.cma.2016.08.015 (cit. on p. 48).

[225] Yingjun Wang et al. "Structural Design Optimization Using Isogeometric Analysis: A Comprehensive Review". In: *Computer Modeling in Engineering & Sciences* 117.3 (Dec. 29, 2018), pp. 455–507. ISSN: 15261492. DOI: 10.31614/cmes.2018.04603 (cit. on p. 39).

[226] Oliver Weeger. "Isogeometric sizing and shape optimization of 3D beams and lattice structures at large deformations". In: *Structural and Multidisciplinary Optimization* 65.2 (Feb. 2022), p. 43. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-021-03131-7 (cit. on p. 47).

[227] Oliver Weeger, Bharath Narayanan, and Martin L. Dunn. "Isogeometric shape optimization of nonlinear, curved 3D beams and beam structures". In: *Computer Methods in Applied Mechanics and Engineering* 345 (Mar. 2019), pp. 26–51. ISSN: 00457825. DOI: 10.1016/j.cma.2018.10.038 (cit. on pp. 38, 47).

[228] Oliver Weeger et al. “Optimal Design and Manufacture of Active Rod Structures with Spatially Variable Materials”. In: *3D Printing and Additive Manufacturing* 3.4 (Dec. 2016), pp. 204–215. ISSN: 2329-7662, 2329-7670. DOI: 10.1089/3dp.2016.0039 (cit. on p. 38).

[229] Zijun Wu et al. “Topology optimization of hierarchical lattice structures with substructuring”. In: *Computer Methods in Applied Mechanics and Engineering* 345 (Mar. 2019), pp. 602–617. ISSN: 00457825. DOI: 10.1016/j.cma.2018.11.003 (cit. on pp. 2, 78, 79).

[230] Qi Xia, Michael Yu Wang, and Tielin Shi. “A method for shape and topology optimization of truss-like structure”. In: *Structural and Multidisciplinary Optimization* 47.5 (May 2013), pp. 687–697. ISSN: 1615-147X, 1615-1488. DOI: 10.1007/s00158-012-0844-y (cit. on pp. 7, 76).

[231] X. Yan et al. “Concurrent topology optimization of structures and their composite microstructures”. In: *Computers & Structures* 133 (Mar. 2014), pp. 103–110. ISSN: 00457949. DOI: 10.1016/j.compstruc.2013.12.001 (cit. on p. 44).

[232] Oktay Yilmaz, Hasan Gunes, and Kadir Kirkkopru. “Optimization of a profile extrusion die for flow balance”. In: *Fibers and Polymers* 15.4 (Apr. 2014), pp. 753–761. ISSN: 1229-9197, 1875-0052. DOI: 10.1007/s12221-014-0753-3 (cit. on p. 66).

[233] Przemyslaw Zagrodzki. “Numerical analysis of temperature fields and thermal stresses in the friction discs of a multidisc wet clutch”. In: *Wear* 101.3 (1985), pp. 255–271 (cit. on p. 35).

[234] Pu Zhang et al. “Efficient Design-Optimization of Variable-Density Hexagonal Cellular Structure by Additive Manufacturing: Theory and Validation”. In: *Journal of Manufacturing Science and Engineering* 137.2 (Apr. 1, 2015), p. 021004. ISSN: 1087-1357, 1528-8935. DOI: 10.1115/1.4028724 (cit. on pp. 2, 8).

[235] Yongcun Zhang and Shutian Liu. “The optimization model of the heat conduction structure”. In: *Progress in Natural Science* 18.6 (June 2008), pp. 665–670. ISSN: 10020071. DOI: 10.1016/j.pnsc.2008.01.010 (cit. on p. 58).

[236] Olek C Zienkiewicz, Robert L Taylor, and Jian Z Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 2005 (cit. on p. 37).

[237] Jacques Zwar. *Tataratat/pygadjoints: python binding for gismo frontend - adjoints for diffusion and linear elasiticity*. https://github.com/tataratat/pygadjoints/tree/main. 2024 (cit. on p. 49).

[238] Jacques Zwar, Gershon Elber, and Stefanie Elgeti. “Shape optimization for temperature regulation in extrusion dies using microstructures”. In: *Journal of Mechanical Design* 145.1 (2023) (cit. on pp. XXI, 22, 23, 25, 35, 46, 57, 67, 68).

[239] Jacques Zwar et al. *tataratat/bezman: 0.2.0*. https://doi.org/10.5281/zenodo.8369310. Version 0.2.0. Sept. 2023. DOI: 10.5281/zenodo.8369310 (cit. on p. 15).