8-2024

Linear and Nonlinear Topology Optimization with Morphing Beam Networks

Andrew S. Montalbano
Clemson University, asmonta@g.clemson.edu

Linear and Nonlinear Topology Optimization with Morphing Beam Networks

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mechanical Engineering

by
Andrew Montalbano
August 2024

Accepted by:
Dr. Gang Li, Committee Chair
Dr. Georges Fadel
Dr. Huijuan Zhao
Dr. Oliver Myers
Dr. Vincent Blouin

Abstract

Topology optimization (TO) is an engineering design discipline dedicated to optimizing material distribution within a given domain. In traditional gradient-based topology optimization, the solid domain is discretized into small volumetric elements. Using finite element analysis (FEA) of the structure, the gradient of the objective function with respect to the design variables (the pseudo densities) is computed, and these design variables are updated iteratively until convergence is achieved. Although gradient-based TO methods are well-established, sensitivity analyses of objective functions and constraints can be both mathematically complex and computationally intensive. The nonconvex nature of most TO problems often complicates efficient convergence. Furthermore, traditional TO methods are limited in their ability to design structures that achieve specified nonlinear deformation behaviors; specifically, they struggle to match load-by-load deformations to consistently produce a topology that delivers the desired nonlinear load-deformation characteristics.

In this work, with the observation that topology-optimized structures frequently resemble sparsely arranged beams, we propose to replace the initial domain of solid elements with a sparse network of morphing beam elements to largely reduce the computational cost while retaining the result fidelity. Although the substitution of solid elements with beam elements has been explored in existing literature, previous methods typically face challenges related to scalability, manufacturability, and design space limitations. More importantly, no prior efforts have focused on developing a beam network-based topology optimization method specifically aimed at achieving targeted nonlinear deformation responses, despite its potential engineering benefits. This work introduces an efficient approach to creating topology-optimized structures using a morphing beam network, thereby reducing the associated degrees of freedom and computational costs. This method can be leveraged to address more complex topology optimization problems, such as optimization for nonlinear deformation.

Two types of optimization problems are examined using the proposed approach. First, to validate the method's ability to maintain the result fidelity found in gradient-based topology optimization with solid elements, we conduct an in-depth investigation of compliance minimization and compare the optimized structures with those obtained using solid elements. The second optimization investigation employs a nonlinear beam theory and leverages the efficiency of a sparse morphing beam network to generate structures with specified nonlinear load-displacement curves. Numerical results demonstrate that, by optimizing the nodal locations and beam widths in a sparse network of discretized beams, both linear and nonlinear topology-optimized structures can be produced with significantly reduced degrees of freedom and computational cost compared to traditional methods.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In engineering design, the strategic placement of materials within a structure is critical to achieving optimal performance and efficiency. Proper material distribution not only enhances the strength and durability of the structure but also minimizes weight and reduces resource consumption. Traditional design methods often rely on an engineer's intuition and incremental improvements to existing structures, which can lead to sub-optimal use of materials and resources. Improving material placement can minimize costs, reduce environmental impact, and enhance a structure's performance. This poses a significant challenge: how to best allocate material within a given domain to meet specific design objectives. This challenge is particularly critical in fields where weight reduction without compromising structural integrity can lead to substantial benefits, such as in aerospace and automotive industries.

Topology, as a concept, refers to the fundamental layout or configuration of an object, including the distribution and arrangement of its components and materials. The arrangement of material can significantly impact how an object responds to stresses and strains. Consequently, topological configurations are extensively studied in engineering fields such as civil, automotive, and mechanical engineering[1–3]. For instance, material placement within a bridge greatly impacts its structural integrity, while effective material placement in buildings maximizes available floor space and minimizes cost. In automotive engineering, material distribution directly affects a vehicle's performance. Furthermore, material distribution within a mechanism can significantly influence its properties[4]. Given that an object's topological representation is heavily coupled with its performance, optimizing its topology is a natural step in achieving design objectives.

Optimization in engineering design seeks to find the best solution given a model, a set of parameters and objectives, and any constraints attached to the problem [5]. In the context of engineering topology, the model becomes the topological representation of an object. To modify the object's topology, parameters defining its material placement are optimized to best fulfill a given objective, considering any constraints. By tuning these parameters, the overall topology of the object morphs into a form that best fulfills the given objective. This process of morphing an object's topology to obtain an optimal material distribution is appropriately named Topology Optimization.

1.1 Motivation

By modifying and optimizing an object's topology, its performance can be increased relative to its initial configuration, such as decreasing a frame's weight while simultaneous increasing its overall strength. While this promise of achieving increased performance for an object is compelling, the computational demands and requirements to perform topology optimization often pose significant limitations and challenges to mass adoption. The nature of the topology optimization process often involves repeated analyses of an engineering system, which alone can require a large amount of computational resources. This poses challenges for widespread adoption of topology optimization in engineering design, as it can either become prohibitively costly or require analysis-simplifications that can result in suboptimal or over-designed products.

This challenge leads to the first motivation of the work: to decrease the overall computational cost required for topology optimization. Decreasing the computational cost required for optimization reduces the barrier to entry in performing topology optimization, as well as increasing its prevalence in engineering design. As discussed in the following literature review section, topology optimization can be split into three different sub-processes: the topological representation of the object, the analysis method, and the optimization strategy. With the analysis method heavily depending on the physics considered, this work will investigate the reducing the computational cost of the topological representation of the object coupled with the optimization method.

Starting with the topological representation of an object, it was observed that existing topology-optimized structures frequently resemble a network of beams. For example, the topology-optimized structure shown in fig. 1.1 was obtained with solid elements whose final topology resembles a network of beams [6]. This poses the question of whether the same result could be obtained by representing and subsequently optimizing a sparse network of beams from the beginning. Representing the same topology with a beam network in place of a solid domain has two advantages. First, constructing the beam-like features from solid elements requires a large number of solid elements, while the same feature can be represented using a single beam element. Secondly, the white space in fig. 1.1 corresponds to areas of negligible material distribution representing a void. However, these regions are still represented in the analysis, incurring a computational cost for regions that negligibly impact the per-

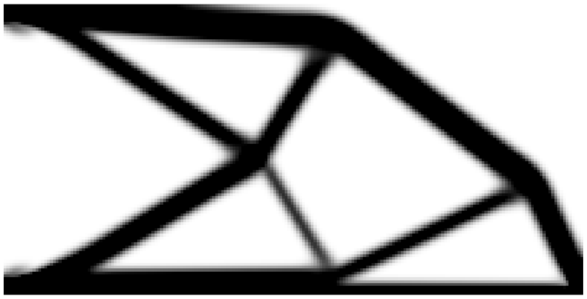


Figure 1.1. An optimized topology obtained with solid elements [6]

formance of the object. These regions can be better represented using beam elements for two reasons. First, regions without a beam present have no impact on the object's performance, naturally behaving as void cavities. Secondly, when regions with negligible impact on performance are included in the analysis, the sparse nature of beam networks mitigates the computational penalties incurred from modeling these regions.

As discussed in the following literature review, these capabilities of beam networks have not gone unnoticed and beam elements have been previously pursued in topology optimization. However, current beam network implementations in topology optimization do not fully approximate a solid domain, resulting in a reduced design space.. While beam elements can potentially match the same result fidelity obtained with solid elements, this requires morphing the structure, which increases the difficulty of calculating analytical derivatives. Because of this, the more efficient gradient algorithms often implemented with solid elements are replaced with heuristic algorithms, which have their own computational penalty. Alternatively, implementations may seek to reduce the topology being represented or make concessions to reduce the computational cost. In addition to investigating topological representations, the first portion of this work investigates the use of gradient algorithms with beam elements to reduce the computational cost associated with more expensive algorithms.

Topology optimization often requires repeated analysis of the object's topology. This necessity often leads to simplifications, such as reducing the scope of physics considered during analysis. In light of this, the second portion of this work leverages the computational cost savings achieved in the first portion to investigate topology optimization considering more advanced physics. By considering advanced physics models, the true behavior of an object can be more accurately represented, resulting in the optimized object's digital representation better describing its true behavior. Furthermore, considering additional physics allows for new optimization objectives to be pursued in topology optimization.

In summary, the motivations of this work: are first is to reduce the computational cost required by topology optimization by representing the topology of an object with a sparse network of beams which is subsequently optimized by a gradient-based algorithm. Secondly, to leverage these computational savings to consider analyses that more accurately depict the true behavior of an object. The following section discusses the current state of the art in topology optimization with respect to the motivations of this work.

1.2 Literature Review

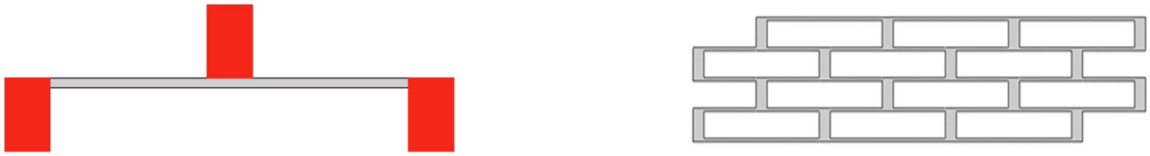
Topology optimization in engineering design can be considered consisting of three essential and distinct processes. The first process involves defining the topological representation governing the placement of materials targeted for optimization. This includes definitions for parameters, variables, domains, and other necessary elements to accurately characterize the topology under consideration. The second process centers on the methodology employed to measure the performance of the previously defined topology, including the physics considered, performance metrics, and interactions influencing to object's behavior. Finally, the third process involves the application of optimization methods and techniques, including algorithm selection, constraint considerations, and optimization formulation used to modify the topology to best achieve a user selected design objective(s).

These three processes largely operate independently. For instance, one can use the same topology definition and optimization technique while considering different physical models and analysis methods. Moreover, each process constitutes an active research field in its own right, often advancing autonomously. Given the interaction between these processes, this section aims to provide a comprehensive overview of current state-of-the-art practices with respect to topology optimization. This section is divided into three subsections: topological representations, analysis techniques, and optimization methods.

1.2.1 Topological Representations in Topology Optimization

By in large the most common representation of material across multiple different topology optimization problems are solid elements [1–3, 6–13] and shell elements [13, 14]. These representations take a domain representing the bounds of an object, and discretizes said domain into much smaller individual elements. Often, these elements are arranged in a square-orthogonal grid [6, 7], however non-grid representations have been tested to better approximate the domain being optimized [8].

Another technique to represent a material topology is to use so called "element function geometries (EFG)" and construct the material distribution within the periodic unit cells by connecting the EFGs in various ways [15, 16]. These geometries are designed to achieve specific objectives within each cell. Subsequently, these cells are interconnected to form a lattice structure, whereby the overall properties of the lattice emerge from the collective behaviors of the individual cells. For example, the lattice depicted in fig. 1.2b was assembled using an arrangement of the unit cell illustrated in fig. 1.2a [15]. The topology of each individual unit cell was tailored to achieve a specific design objective, which manifests in the final lattice structure.



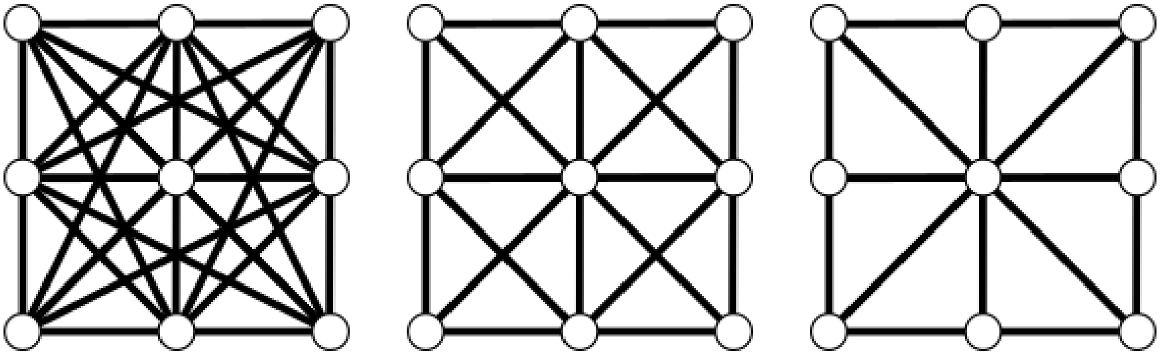
(a) Unit Cell

(b) Resulting Lattice

Figure 1.2. Topology Representation using Unit Cells

As discussed in the motivation, the similarity between optimized topologies and a network of beams has been observed and there are several papers in the literature investigating these beam-based topological representations in topology optimization [17–24]. Similar to solid elements, these methods often start by a grid of nodes, but instead of forming solid element representations of the topology, they connect the nodes together to form a network of beams. These connections play a large role in determining not only the topology being represented, but also in the computational cost needed to store, analyze, and optimize the network and in the final optimized topologies.

The simplest way to connect a set of nodes to construct a network is to fully connect each node with every other node, resulting in a fully connected network [17, 22, 23]. An example of this arrangement on a 3 × 3 grid of nodes may be seen in fig. 1.3a. While this is the simplest method to construct a network, it is also incredibly computationally expensive as the number of elements scales with the number of nodes to the fourth power [25]. This sharp increase in elements directly increases the computational cost required to store, analyze, and optimize the network.



(a) Fully Connected **(b)** Neighbor Connections **(c)** Periodic Connections

Figure 1.3. Example beam networks on a 3 × 3 grid of nodes

Additionally, depending on the grid layout, several elements may be coincident with one another, introducing feasibility concerns and redundant elements.

Rather than connecting each node with one another, nodes can be connected only with a subset of the total number of nodes, which greatly reduces the computational cost [19–23]. As of this writing, there does not appear to be consensus in the literature as to the best method to form connections between a smaller subset of nodes. As such, multiple different strategies have been considered and deployed. One methodology starts from a rectangular grid, and forms connections directly with its neighbors [20]. An example of this grid may be seen in fig. 1.3b. The most obvious benefit of this method is the drastic reduction in the computational cost compared to the fully connected grid [26]. However, by connecting elements to their neighbors results in diagonal elements that overlap with one another, which is physically impossible. There are ways to implement such designs in the real world, such as moving around the connection in the third dimension, but this work considers 2D topologies making such representations unfeasible. As a result, network constructions that feature overlapping elements are not considered in this work. The overlapping diagonal elements could be removed from this method, resulting in a orthogonal grid of connections. This would eliminate the feasibility concerns and reduce the overall computational cost [27], but likely results in a loss in result fidelity as the author were unable to find papers that implemented such a method. It is possible the result fidelity could be increased by increasing the density of the nodes, but this quickly becomes analogous to a solid element domain, and obtains the computational burden previously described.

Alternatively, rather than connecting nodes to their immediate neighbors, additional nodes and connections can be inserted, an example of one such design is depicted in fig. 1.3c [19]. This method eliminates any overlapping elements found with the previous method, but results in a periodic mesh that can effect the results obtained from topology optimization. Additionally, this method increases the number of elements compared to the previous connection method, and therefore increases the computational cost associated with this representation.

Lastly, under most circumstances, the extension from 2D to 3D is straightforward, but incurs increased computational cost as it introduces an additional dimension whose topology needs to be represented, analyzed, and optimized. As such, most topology optimizations are performed in two dimensions. However, this does not imply that 3D design has been neglected. There are studies that investigate such designs [2, 7, 10, 14]. Nevertheless, these studies must consistently consider performance characteristics.

These presented works show a wide variety of methods used to represent the topology of objects. However, what is of interest to this work are the representation of thin elements. As apparent from the multiple techniques used to connect nodes into a network, there is lack of general consensus which highlights a problem this work seeks to address. The remaining sections of the literature review investigate the analysis techniques commonly used in topology optimization, followed by the optimization methods.

1.2.2 Analysis Techniques in Topology Optimization

Once the topology representation of an object has been determined, physical analysis needs to be performed to measure its performance. To perform the physical analysis, topology, finite elements methods are often used [1–3]. These methods break down complex structures into smaller, manageable elements for analysis. For the physical analysis, both the physics considered and element formulations are needed.

There are a multitude of physical phenomena considered in topology optimization, such as heat transfer [2, 9], fluid dynamics [10, 11], or vibrations [3, 12]. But the most common investigations are in analyzing an object's mechanical response [1, 6–8, 20, 21]. As this is the physics this work is interested in, the remainder of this literature review will discuss the state of the art of topology optimization considering mechanical responses of structures.

The most common mechanical analysis considered in topology optimization is measuring and subsequently minimizing the internal work done within a subject to an applied load, often called compliance [1, 6–8, 20, 21]. With u referring to a structure's deformation and k to its stiffness, the value for compliance, c, may be obtained and is shown in eq. (1.1) [6]. Since the topology of an object determines where its material is placed, which is directly related to its stiffness and subsequent displacement under load, minimizing a structure's compliance decreases the internal work it performs, effectively increasing the structure's stiffness.

$$c = u \cdot k \cdot u \tag{1.1}$$

Similarly, some research has focused on creating structures with mechanical responses tailored to match specific targets [15, 16], although this area is not studied as extensively as compliance. These methods involve measuring the displacement of a specific region within a topology, such as a node, and then modifying the topology to adjust the region's response until it aligns with a user-defined target.

While these descriptions outline the performance aspects under analysis, equally crucial is the methodology used for their evaluation. Given the inherent expense associated with nonlinear analysis and the consideration of additional physical phenomena, much of the research focused on mechanical responses has concentrated on topology optimization using linear-elastic elements [6–8, 19]. However, there has been some exploration into topology optimization that incorporates nonlinear effects. For example, studies have examined nonlinear deformations [15] and internal contact [16] using solid element representations. Additionally, a study utilizing beam elements explored nonlinear behavior [17] but did not comprehensively explore the entire topological space. Similarly, akin to the computational efficiencies gained with beam elements, shell elements have also been investigated as a strategy to mitigate the computational costs associated with nonlinear analysis [14]. These studies demonstrate that topology optimization can indeed be conducted with nonlinear analysis, albeit often necessitating compromises to manage the associated increase in computational demands.

In finite elements methods, there are two common representations of thin features. Truss elements whose degrees of freedom are solely defined by their nodal-displacements, and frame elements which additionally consider a rotation degree of freedom [28, 29]. This work will refer to these elements as beam elements. Both thin-feature representations have been investigated in topology optimization, with trusses commonly used to model structures that traditionally do not generally consider rotation effects, common in bridges or frame design [19, 23]. Alternatively, beam elements are often used to explore structures that do consider these effects [18, 21]. The choice between truss and beam elements in finite elements often comes down to whether or not rotation of the nodes will play a role in determining the stiffness response of the object. As one would expect, the consideration of rotation can impact the optimized results. However, due to similar topological representations, extending or translating between the two is relatively straightforward and both kinds of elements are covered in this section.

This section has shown the current state of the art when it comes to analysis in topology optimization. While most analyses have only considered linear effects, there have been some investigations into nonlinear effects. But these often make concessions to reduce the increased computational cost required by analysis. The next section will discuss various optimization methods in topology optimization.

1.2.3 Optimization Methods in Topology Optimization

There are several optimization techniques used to optimize the topology of an object to maximize a performance metric. Most methods, irrespective of the focus of analysis, implement a gradient based approach [1–3]. Gradient methods are optimization techniques that utilize the gradient of the objective function and constraints to guide the search for optimal solutions. A common method is to iteratively move the design variables in the direction of the steepest descent, as indicated by the gradient, to minimize the objective function [5]. Gradient methods are widely used due to their efficiency and effectiveness in handling smooth and differentiable functions, but necessitate the calculation of the gradient of the objective function and constraints. Determining the gradient of the objective function is more involved compared to solid or shell elements, resulting in a scarcity of studies investigating gradient approaches compared to solid elements.

When gradient methods are used to optimize beam networks, rather than optimizing the full topological space, a subspace is considered. One such approach is to constrain the positions of the nodes of the network [17, 19]. While these studies were successful in producing topology optimized results, by neglecting node position changes can fail to appropriately search the full topological space. This can be mitigated by increasing the density of the network, but this incurs a computational penalty which offsets any potential gains from using a beam network.

Instead of gradient based approaches to topology optimize a beam network, an alternative method is to deploy genetic algorithms [20, 21, 23]. Genetic algorithms are a class of optimization techniques inspired by the principles of natural selection and genetics. They operate by generating a population of potential solutions and then iteratively evolving these solutions through selection, crossover, and mutation processes. Over successive generations, the algorithm selects the fittest individuals from the population to produce offspring, gradually converging towards an optimal solution. An optimization method similar to genetic algorithms is ant colony optimization which also has investigated beam element structures [30]. Both genetic algorithms and ant colony techniques are heuristic in practice, and thus require a large number of model executions for each optimization iteration. Further, these optimization techniques can often be inefficient at exploring the domain. Both of these drawbacks can significantly increase the computational cost associated with optimization, again mitigating any potential increase obtained by a morphing beam network.

All of the aforementioned studies indicate that topology optimization using beam elements is both feasible and of considerable interest to the scientific community. However, there are two primary shortcomings. First, there is a lack of consensus regarding the appropriate beam network representation for modeling the topology of an object. Second, there is an absence of a fast, efficient, gradient-based approach to topology optimization involving beam elements. The objective of this work is to develop an efficient beam network based method and to investigate a nonlinear topology optimization using morphing beam networks

1.3 Contribution

This work introduces a novel topology optimization methodology using a morphing beam network. The network is constructed by a set of nodes connected to one another by a sequence of elements, collectively approximating a solid domain. Through adjusting the node positions and beam dimensions of the network, the topology it represents can be optimized to meet specific optimization objectives.

This research has two primary contributions. The first aims to reduce the computational costs associated with topology optimization, achieved through two strategies. Traditionally, optimized solid element structures often feature thin components akin to beams. Representations of these features typically require a large number of solid elements, whereas the same behavior could be captured by using a few beam elements. Thus, replacing solid elements with a beam elements significantly reduces computational costs for both analysis and optimization while maintaining the same topological features and performance. Additionally, in optimized topologies, there are often regions that have negligible impact on the structure's performance but still need to be represented in the analysis, incurring unnecessary computational costs. Beam elements offer a more efficient solution in such cases as regions without a beam present have no impact on the object's performance, naturally representing cavities.

The second significant source of computational expense in topology optimization arises from the algorithms employed. This work addresses this problem by adopting gradient-based algorithms. These algorithms iteratively adjust optimization variables by following the steepest descent direction, evaluated at each iteration in optimization. This iterative process progressively minimizes the objective function, thereby morphing the structure's topology to best meet a design objective. To further streamline the optimization process and reduce computational overhead, this study obtains and provides the necessary analytical derivatives.

Once the initial goal of reducing computational costs in topology optimization is achieved, the second objective is to harness this cost reduction to optimize the topology of structures that account for nonlinear effects in their analysis. Specifically, this study focuses on nonlinear deformations, often termed as large displacements. This analytical approach considers how a structure's stiffness changes due to internal forces within each element and the deformations experienced by those elements. As these forces and deformations evolve throughout the analysis, the mechanical response of each element varies, often resulting in nonlinear force-displacement curves as a result of geometric nonlinearity. Therefore, the second goal of this study is to optimize and adjust these force-displacement curves obtained from a beam-network representation of topology to align with a user-defined response.

To achieve these objectives, this study establishes a methodology to optimize a network of beams that can represent a arbitrary topology effectively and efficiency. The proposed approach outlined in this study involves defining the initial network construction, implementing the optimization algorithm and strategy, and obtaining the necessary gradients required for its two optimization case studies.

1.4 Summary of Content

The dissertation of the manuscript details the methodology for the proposed approach and is organized into six chapters. The first half of the work encompasses defining the network used in optimization, deriving the formulation employed in analysis, and outlining the optimization method (chapters 2 to 4). Following these foundational chapters, the subsequent chapters (chapters 5 and 6) field the method in two different optimization problems. This dissertation is then concluded with closing remarks and a discussion on avenues for future research.

Specifically, chapter 2 details the construction of the beam network utilized in subsequent topology optimization analyses. This chapter covers the definitions used to define the topology, placement of nodes and the connections between them, and the discretization process.

In the following chapter, chapter 3, the finite element formulation is presented. This chapter begins by deriving the stiffness matrices obtained from the co-rotation formulation, which accounts for nonlinear geometric effects. Once the nonlinear stiffness matrix has been derived, the linear-elastic stiffness matrices are obtained as a special case of the co-rotation formulation, but which matches traditional derivations. The remainder of the chapter outlines the solution algorithm employed to determine the equilibrium path when considering nonlinear elements.

With the groundwork laid for topology and finite element definitions, chapter 4 presents the optimization algorithm and strategy used to achieve the results detailed in the subsequent chapters. This chapter also includes the necessary derivatives concerning node locations and positions, presented in sections 4.3 and 4.5 respectfully. Additionally the constraints considered in optimization are discussed.

Chapter 5 shows the first optimization cases in which a network of beams is optimized to minimize the compliance, or strain energy, of a structure. This chapter begins with the formal definition of the compliance minimization followed by several progressively complex test cases.

Chapter 6 extends the proposed topology optimization method to accommodate changes in stiffness due to nonlinear effects. The objective investigated in this chapter is to match a user-defined force-displacement curve. Similar to the previous chapter, this section begins with the formal optimization definition used in subsequent test cases.

Chapter 7 summarizes and concludes the work while suggesting future opportunities.

Chapter 2

Topological Construction

In the previous chapter, the motivation, literature review, and contributions of this work were discussed. In short, the primary objective of this study is to enhance the efficiency of representing solid object topologies using a sparse beam network. This chapter aims to elaborate on this concept by defining the proposed beam network and its application in topology optimization. It begins by describing the solid domains typically used in topology optimization, followed by the introduction of the proposed beam network representation. Unlike dense solid element methods that discretize the entire volume, the proposed beam network consists of nodes (discrete points within the structure) and beams that connect these nodes, creating a skeletal framework that approximates the object's topology. The connectivity between nodes, which defines the beams in the network, is introduced along with its initial construction. The chapter concludes with the edge discretization method used in nonlinear analysis within the proposed beam network framework.

2.1 Topology Representation by Morphing Beam Networks

The conventional approach in topology optimization involves optimizing material distribution within a solid isotropic material domain, denoted as Ω, with regions where boundary conditions are applied called Γ. The boundary Γ comprises three distinct regions: Γ_f where forces, moments, or surface tractions, $\mathbf{f}$, are applied. Γ_u denotes regions of known or given displacement. Finally, Γ_o denotes the rest of the boundary, i.e., the free surface of Ω. This makes Γ defined as $\Gamma = \{\Gamma_o, \Gamma_f, \Gamma_u\}$. An illustration of Ω and Γ for an arbitrary domain is depicted in fig. 2.1.

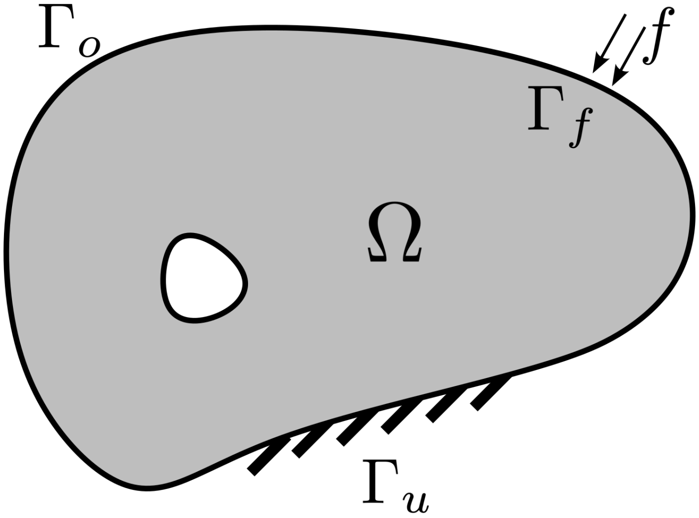


Figure 2.1. Arbitrary domain represented with a solid material distribution

As outlined earlier, this work proposes to represent the solid domain using a morphing beam network. This network replaces the traditional solid domain representation with discrete beams of varying lengths and thicknesses. By adjusting the placement and material distribution along these beams, the structure's internal topology and geometry can be approximated, without the need of modeling the large space of the voids.

To translate the solid topology Ω into a beam network, the domain Ω is divided into two subdomains: Ω_e and Ω_s. Ω_s represents the region of the domain containing material, i.e. the network of beams, while Ω_e corresponds to the portion of Ω where material is absent, forming voids, i.e. $\Omega_e + \Omega_s = \Omega$. Initially, Ω_s is constructed using a uniform grid of nodes connected with one another to form individual beam elements. This initial network configuration is subsequently adjusted in each optimization iteration to achieve the defined optimization objective described in chapter 4. Figure 2.2 shows the domain previously described in fig. 2.1 whose internal topology is now represented by an orthogonal beam network .

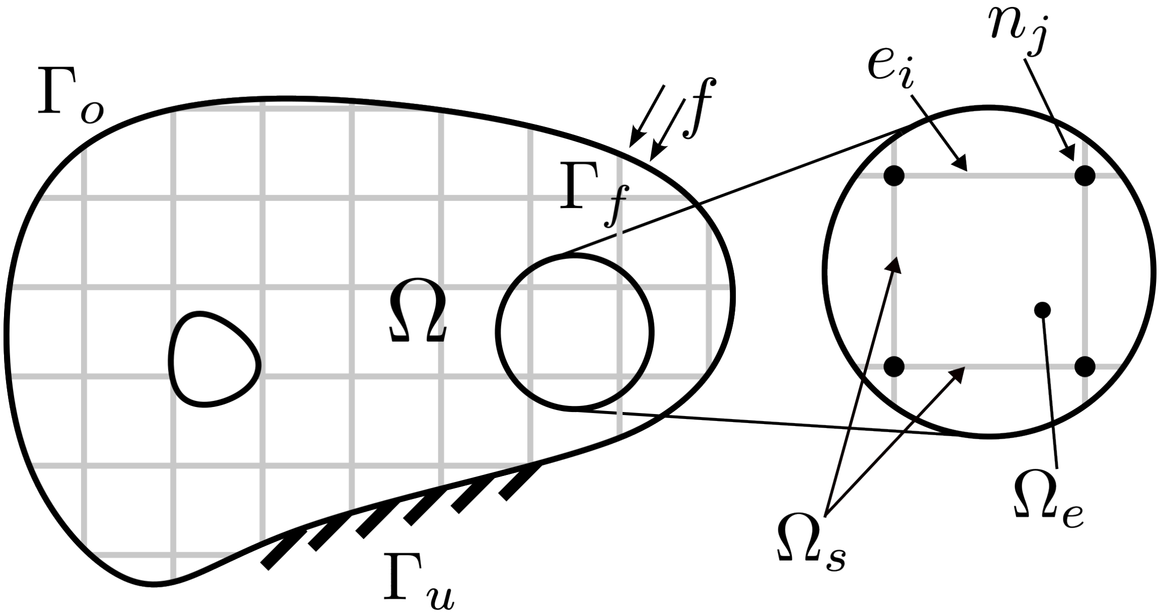


Figure 2.2. Arbitrary domain represented with a beam network

The orthogonal grid in fig. 2.2 represents where material is placed within the structure, Ω_s, and is defined by a set of nodes, $\mathbf{n}$, which are connected together to form a set of elements, $\mathbf{e}$. Each node n is defined by its x and y locations, with the set locations of all nodes in the network defined as the set $\mathbf{x}$ and $\mathbf{y}$, with x_j and y_j referring to the x and y locations of the jth node.

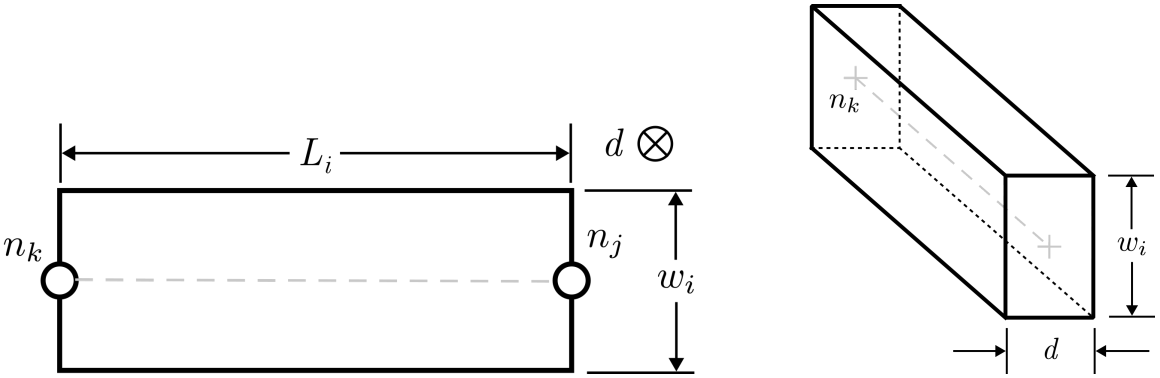
The elements comprising Ω_s require two definitions with the first being to defines the individual topology of each element within the network. Being 1D generic elements, and each element's cross-section geometry can take on various shapes. Square and circular cross-sections could be implemented due to their simplicity and uniform stress distribution characteristics, representing rods or bars. I-beams are another potential candidate due with their large bending resistance. Alternatively, the topology of the cross-section profiles can be optimized. As this work is interested in creating 2D topologies, this work will investigate elements with rectangular cross-sections, which will be discussed in the following section. The section following that will discuss the connection pattern used to construct the initial network.

2.2 Geometric Properties of the Considered Elements

This work will investigate beam networks constructed from rectangular elements. To fully define a rectangular cross-section, two lengths are needed. The first is the in-plane dimension called width, denoted by w with the second dimension being the out-of-plane dimension, referred to as depth, which is denoted by d. Two free-body diagrams of the same rectangular element may be seen in fig. 2.3, with an in-plane, or side view, of the rectangular element shown in fig. 2.3a, and an isometric version shown in fig. 2.3.

As described in full later in chapter 4, only the width of each element is varied in optimization, while its depth is held constant. Similarly, each element in the network's width is independent from one another, while the depth of each element is

[1] The symbol $\otimes$ in fig. 2.3a refers to the engineering-drawing symbol that denotes an out-of-plane dimension



(a) In-plane diagram[1]

(b) Isometric diagram

Figure 2.3. Free body diagrams of the rectangular elements considered

identical. $\mathbf{w}$ will be defined as the set of all widths in the network, with $\mathbf{w}_i$ referring to the width of the *ith* element, while $\mathbf{L}$ refers to lengths of all elements within the structure. With these definitions in mind, the various properties for element e_i, the cross-sectional, moment of inertia given, and the length given are given by eqs. (2.1) to (2.3)

$$A_i = \mathbf{w}_i \times d \tag{2.1}$$

$$I_i = \frac{\mathbf{w}_i^3 d}{12} \tag{2.2}$$

$$L_i = \sqrt{(n_{k,x} - n_{j,x})^2 + (n_{k,y} - n_{j,y})^2} \tag{2.3}$$

where n_k and n_j refer to the two nodes that define the element

From fig. 2.3, it is apparent that volume V of the element is given by $V = A \times L$. With this, the volume of material present within the structure is obtained by eq. (2.4).

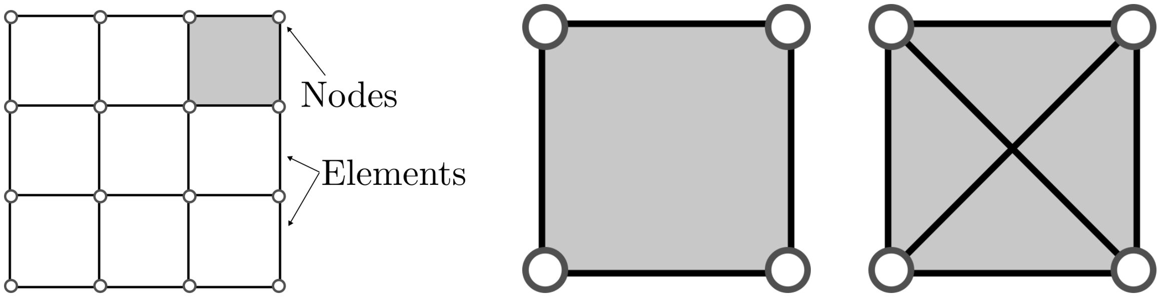
$$V = \sum_{i=0}^{\text{num. elem.}} A_i \times L_i \tag{2.4}$$

With these definitions, the geometric characterization of a single element is complete. However, a major component of defining the topology of the beam network lies in the connections between nodes that form the network. Therefore, the next section will delve into the methodology used to establish these connections, which are essential for constructing the beams that comprise the network.

2.3 Beam Network Construction Methodology

The connections used to construct the elements of the network significantly impact both the feasibility and the quality of optimized structures. Feasibility relates to whether a structure can be physically manufactured. For example, while two overlapping but non-connected elements can be modeled computationally, they cannot be manufactured. Quality of results refers to the optimized structures ability to reach a optimization objective. Given its importance, this section discusses the method used to connect the nodes together to form the elements of the network.

Consider a simple rectangular domain represented by a uniform 2D grid of nodes shown in fig. 2.4a. The grey square in fig. 2.4a represents a single unit cell, which can be configured in various structural arrangements. The simplest configuration is to connect the nodes in the unit cell orthogonality to one another, as shown in fig. 2.4b. However, due to the grid's sparsity, implementing and optimizing a network constructed from orthogonal unit cells failed to accurately replicate the behavior of solid element domains without using an excessively dense mesh or unrealistically thick beams. An alternative unit cell configuration, incorporating diagonal connections as shown in fig. 2.4c, has been explored in previous literature [19]. This approach significantly enhances the quality of the results and produced optimized structures comparable to those achieved with solid elements. However, the structures obtained with diagonal unit-cells are unfeasible in two-dimensions due to their overlapping, yet unconnected, elements.



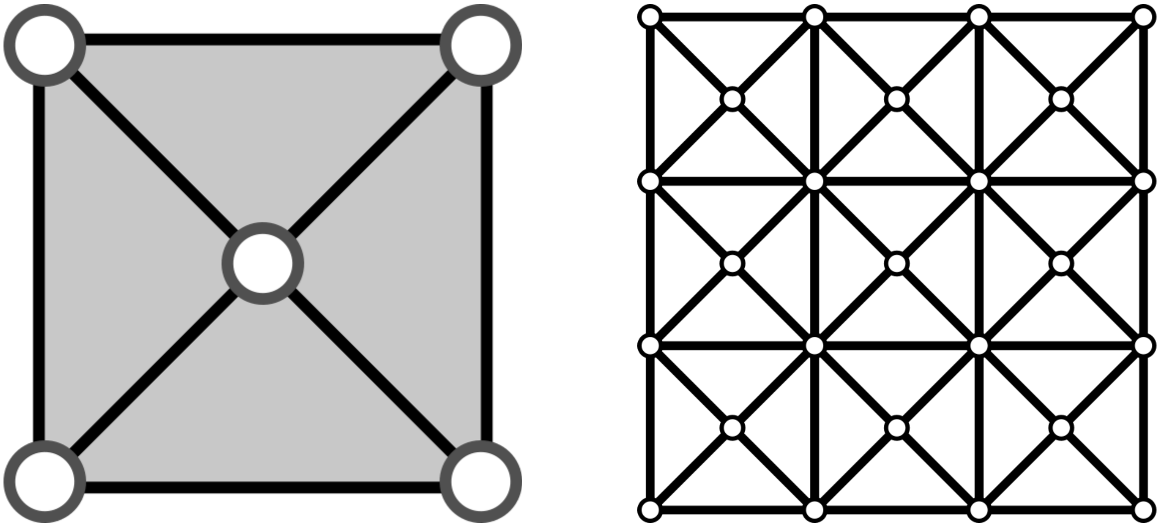
(a) Uniform grid of unit-cells **(b)** Simple unit cell **(c)** Diagonal unit cell

Figure 2.4. Unit cell configuration for a 4 × 4 grid of nodes

To retain the result fidelity obtained by diagonal unit-cells while allowing for feasible structures, an additional node is inserted at the point of intersection, as depicted in fig. 2.5a. This splits the two unconnected elements into four interconnected elements. This modification effectively eliminates the overlap, enabling the creation of structures that are feasible for manufacturing without compromising result fidelity. Therefore, the unit cell depicted in fig. 2.5a is used as the initial unit in this work.

Returning to the previous 4 × 4 grid of nodes and populating each unit cell with fig. 2.5a produces the structure shown in fig. 2.5b, which would be used in the subsequent optimization. It is important to note that this process is specifically used to generate the initial Ω_s. Throughout optimization, the connections between nodes that define the beam elements remain consistent and are preserved.

Using the unit-cell described in fig. 2.5a, the method for representing a topology using a network of beams is fully established. Figures 2.2, 2.3b and 2.5a may be combined together to form a complete description of the network considered in the work, which is shown in fig. 2.6. In fig. 2.6, the previously described solid domain shown in fig. 2.1 is replaced by a grid of nodes. These nodes are connected both orthogonally and diagonally, with an additional node inserted at the point of



(a) Diagonal unit cell with center node **(b)** Final Beam Network

Figure 2.5. Implemented unit cell and its corresponding 4×4 example structure

intersection to construct Ω_s. The whitespace within Ω represent regions void of material, Ω_e. A black-circle cutout highlights a single unit-cell, shown in magenta. The unit-cell structure is formed by nodes connected to form elements, as indicated by a red-circle cutout representing a single node n_j, and a blue cutout illustrating a single element e_i.

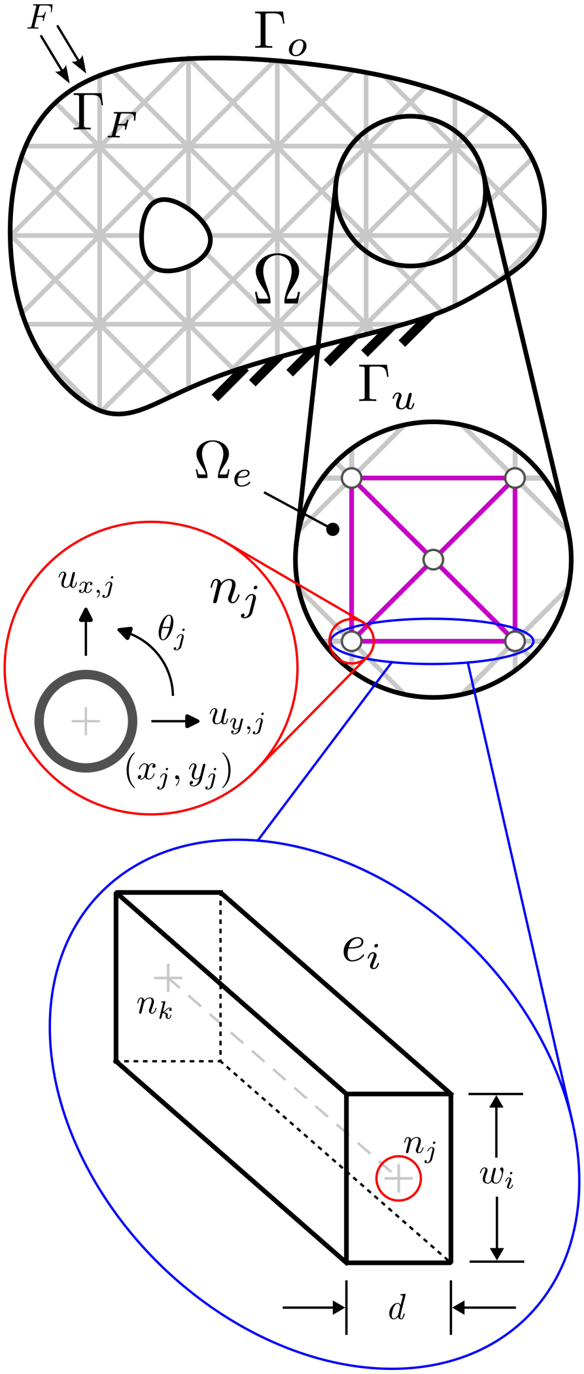


Figure 2.6. Summary of the beam network definitions

This completes the method used to construct the beam network. The following section will discuss the discretization needed for nonlinear analysis. For those primarily interested in linear analysis, it is recommended to proceed directly to the subsequent chapter, chapter 3, where the finite element method considered in this work is presented.

2.4 Edge Discretization Method

Mesh discretization plays a large role in accurately capturing a structure's behavior, especially undergoing large rotations [28, 29, 31]. When analyzing the network with nonlinear geometries, the edges of the morphing beam network are discretized into smaller elements to more accurately predict the structure's deformation response.

To achieve this, the edges of the network defined in the previous section are subdivided into smaller elements. This approach results in two networks: the first network is designated as the reference network while the second network, constructed by subdividing the edges of the reference network, is referred to as the active network. The reference network is optimized directly, whereas the active network is used in analysis. It is important to note that this method is pursued for topology optimizations incorporating nonlinear effects. In contrast, in cases where only linear-elastic effects are present the previously described network, referred to here as the referent, is both optimized and analyzed.

Consider a referent beam which is used to produce an active beam after a single subdivision, depicted in 2.7. In the figure, the black element represents the referent beam, while the grey elements denotes the active beam.

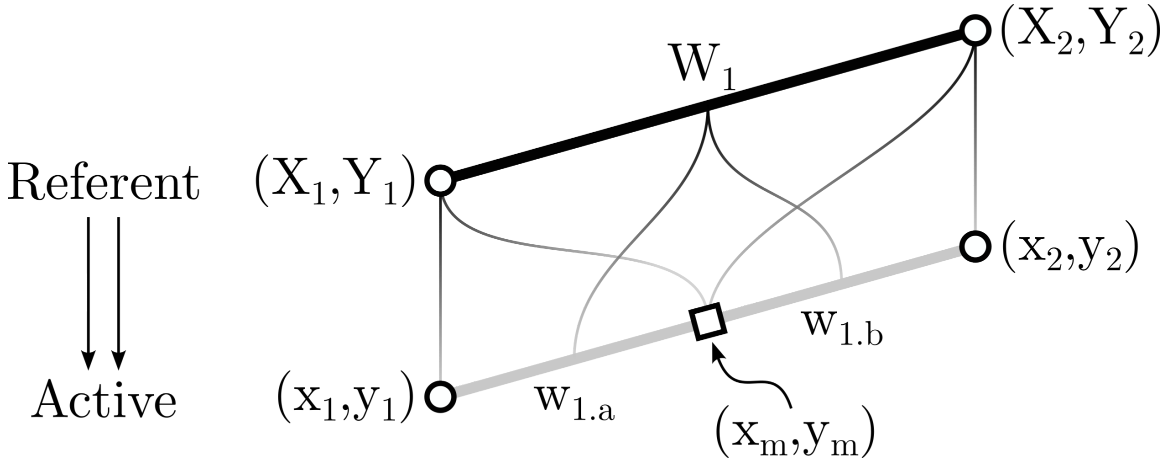


Figure 2.7. Relationship between a referent and active element

It is apparent from fig. 2.7 that the node positions of the active beam are determined by interpolating the node locations of the referent beam, while retaining identical widths. Generalizing to S discretizations, the active structure's new node positions and element widths are given by eq. (2.5). These relations are then used to generate the active structure from the referent structure.

$$
\begin{aligned}
x_s &= X_1 + \frac{s}{S+1}(X_2 - X_1), \quad s = 1, 2, ..., S \\
y_s &= Y_1 + \frac{s}{S+1}(Y_2 - Y_1), \quad s = 1, 2, ..., S \\
w_s &\equiv W
\end{aligned}
\tag{2.5}
$$

With the active structure use in analysis, these relations are inverted to map the analysis results to calculate the gradient with respect to the referent's optimization variables. The gradients are then used to update the referent, and a new active structure is generated using eq. (2.5) for the next optimization iteration.

With the topological definitions complete, the previously solid domain can be represented by a sparse beam network. The next chapter will discuss the finite element method used to analysis the network.

Chapter 3

Finite Element Analysis and Solution Algorithm

In the previous chapter, the topology considered in this work was presented where the traditional solid domain typically used in topology optimization is replaced by a network of beams. These beams are constructed from a sequence of nodes connected to form individual elements, creating a network that approximates a solid domain. This work is interested on analyzing the mechanical response of the network. A well-established method for determining a structure's mechanical behavior is finite element analysis. The formulation this work considers will be presented in this chapter.

Finite elements analysis encompasses a broad field in engineering, with various formulations designed to address a wide range of structural problems. Given the interest in approximating a solid domain with thin members, this work uses 1D elements. To better match the behavior of solid elements, the elements considered here have degrees of freedom in $\vec{x}$ and $\vec{y}$ directions, along with a rotational degree, θ. These elements are commonly referred to as beam or frame elements, with this work using the latter. There are two different formulations presented in the work. The first assumes linear-elastic behavior, while the second accounts for nonlinear behavior arising from internal loads and axial deformations, often referred to as large deformations. Other sources of nonlinear behavior are not included in this study. Specifically, for nonlinear elements co-rotation formulation is considered and derived in the following section, section 3.1. Once the stiffness matrices for nonlinear elements have been obtained, the stiffness matrices for linear-elastic elements are derived as a special case of the co-rotation formulation by setting stiffness coefficients related to internal load and axial deformations to zero and are presented in section 3.3.

When dealing with linear-elastic elements, the structure's displacement can be directly solved. However, the stiffness coefficients of the nonlinear elements discussed in this work depend on their displacements. To solve the system of equations, the Newton-Raphson method is used, as detailed in section 3.4.

Finally, for clarity: bold uppercase letters denote the presence of a matrix of two dimensions, e.g., $\mathbf{K}$ refers to a matrix of $a \times b$ dimensions, while bold lowercase letters denote the presence of a 1D matrix, e.g., $\mathbf{u}$ refers to an $a \times 1$ matrix. A common alternative notation would be the use of overlines to denote the dimensions. Using the previous examples results in $\overline{\overline{K}}$ and $\overline{u}$.

3.1 Corotation Formulation

Linear stiffness matrices assume material behavior follows Hooke's law with constant properties, suitable for small deformations and linear stress-strain relationships. In contrast, the corotated stiffness matrix addresses larger rigid-body motions by separating this motion from local deformations using a reference configuration that translates and rotates with each element. This method allows for more accurate representation of nonlinear geometric effects while maintaining linear formulations for small strain deformations, making it particularly effective for analyzing structures subjected to significant rigid motions and small local deformations[29]. To describe the beam element's mechanical response when undergoing large displacements, the corotational formulation is used with Euler-Bernoulli elements in this work. Only nonlinearity stemming from geometric changes are considered, with the mechanical properties, beam cross-section, and loading conditions assumed to be constant.

A single element is subjected to two kinds of motion: the first being translation and rotation, which are rigid body motions that moves the element from its initial configuration to its corotated configuration; and the second being deformation, which results in the deformed element. These can be seen in fig. 3.1, with the solid black element representing the initial state of the element, the dashed line representing the corotated state, and the grey line representing the deformed state of the element. The circle denotes the first node that defines the element, and the square denotes the end node of the element, with their spatial locations being (x_1, y_1) and (x_2, y_2), respectively. From these node locations, the initial length of the element, L_0, can be calculated, along with its initial angle, β_0.

The displacements applied to each node in the element are defined by $u_{x,1}$ referring to the x displacement observed by node 1, $u_{y,1}$ referring to the y displacement observed by node 1, and θ_1 being the rotation observed by node 1. With node 2's displacements defined similarly. From the motion of the nodes, we derive three components of displacement that produce our deformed configuration: the elongation of the element ΔL with respect to the initial configuration, and the two rotations of the nodes with respect to the corotated configuration, $\theta_{n,1}$ and $\theta_{n,2}$. From these, we define our displacement vectors, with $\mathbf{u}$ representing the displacements with respect to the initial configuration and $\mathbf{d}$ representing the displacements arising from the

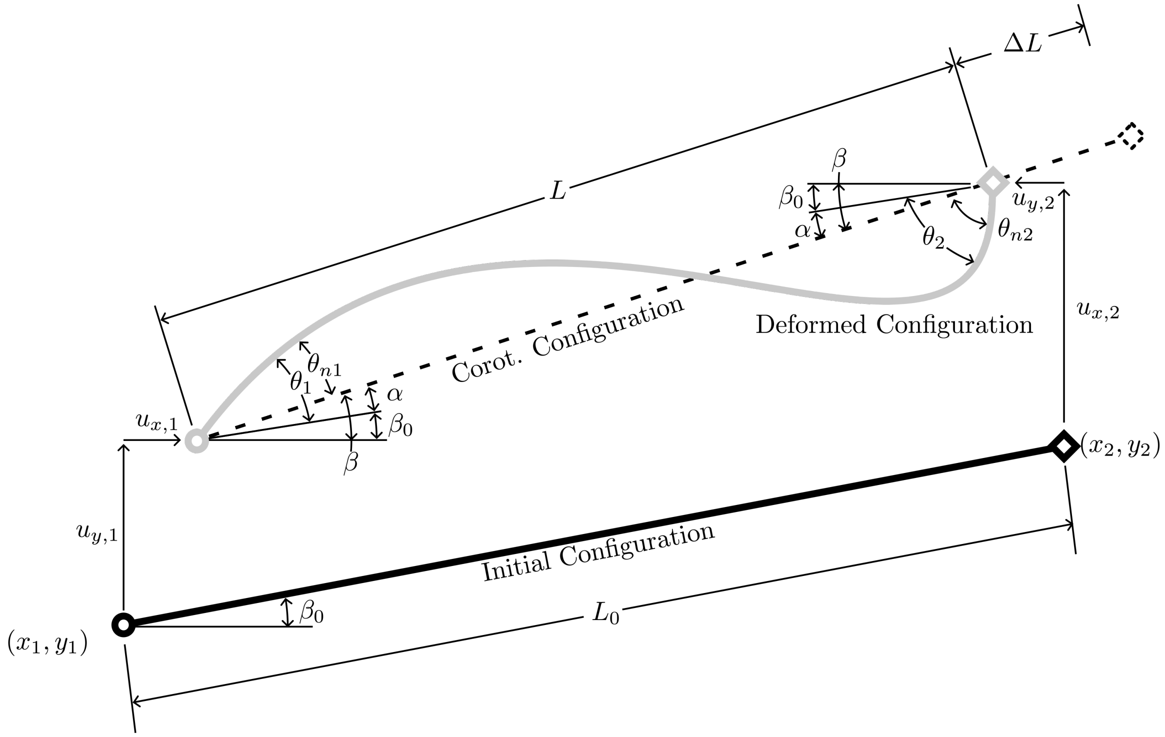


Figure 3.1. Types of motion a corotated element experiences

motion of the nodes:

$$\mathbf{u} = [u_{x,1}, u_{x,1}, \theta_1, u_{x,2}, u_{x,2}, \theta_2] \tag{3.1}$$

$$\mathbf{d} = [\Delta L, \theta_{n,1}, \theta_{n,2}] \tag{3.2}$$

Starting with the elongation of the element, ΔL, is simply the change of length of the element between the initial configuration and the corotated states. For example fig. 3.1 shows a contraction. It is obvious that the definitions for ΔL, L_0 and L are:

$$\Delta L = L - L_0 \tag{3.3}$$

with

$$\begin{aligned} L_0 &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ L &= \sqrt{((x_2 + u_{x,2}) - (x_1 + u_{x,1}))^2 + ((y_2 + u_{y,2}) - (y_1 + u_{y,1})^2)} \end{aligned} \tag{3.4}$$

Internal forces within the element are generated by the elongation of the beam. Assuming linear material behavior and a constant cross-section, Hooke's Law can be used to obtain the axial load which is given by:

$$P = \frac{EA}{L_0}\Delta L \tag{3.5}$$

where

$$\begin{aligned} E &= \text{the modulus of elasticity of the material} \\ A &= \text{the cross-sectional area of the element} \\ \Delta L &= \text{the change in length of the element} \\ L_0 &= \text{the original, undeformed, length of the element} \end{aligned}$$

While this work considers beam elements, A will be kept generic to promote readability.

Moving to rotations, based upon node translations, the angle between the initial configuration and the corotated configuration may change, defined as α. To obtain the difference in angle between the corotated configuration and the deformed configuration, the difference between α and the node rotations θ is taken, resulting in the following:

$$\begin{aligned} \theta_{n,1} &= \theta_1 - \alpha \\ \theta_{n,2} &= \theta_2 - \alpha \end{aligned} \tag{3.6}$$

with

$$\alpha = \beta - \beta_0 \tag{3.7}$$

Similar to the axial loads, end moments are generated as a result of the node rotations. Keeping with the previous assumptions, these end-moments are obtained using the bending stiffness EI, and the relative rotations of the nodes which are given by Hook's Law as:

$$M_1 = \frac{4EI}{L_0}\theta_{n,1} + \frac{2EI}{L_0}\theta_{n,2} \tag{3.8}$$

$$M_1 = \frac{2EI}{L_0}\theta_{n,1} + \frac{4EI}{L_0}\theta_{n,2} \tag{3.9}$$

These moments, along with the axial forces, define the internal force state of the beam element when considering to nonlinear deformations. The axial load and end moments previously defined may be combined in matrix form as:

$$\begin{bmatrix} P \\ M_1 \\ M_2 \end{bmatrix} = \begin{bmatrix} EA/L_0 & 0 & 0 \\ 0 & 4EI/L_0 & 2EI/L_0 \\ 0 & 2EI/L_0 & 4EI/L_0 \end{bmatrix} \begin{bmatrix} \Delta L \\ \theta_{n,1} \\ \theta_{n,2} \end{bmatrix} \rightarrow \mathbf{f}_d = \mathbf{Cd} \tag{3.10}$$

where $\mathbf{f}_d$ and $\mathbf{C}$ refer to force vector arising from node displacements and the elastic coefficient matrix respectfully.

Up to this point, the displacements discussed have been respective to a single element's local coordinate system. To obtain the relationship between a local element's coordinate system and a global coordinate system, consider a small displacement, $\delta\mathbf{d}$, applied to a corotated configuration of an element as shown in fig. 3.2. $\vec{e}_1$ is the unit vector defining the direction of the corotated configuration, and $\vec{e}_2$ is the vector orthogonal to $\vec{e}_1$. $\delta\mathbf{d}$, $\delta\alpha$, and $\delta\Delta L$ are changes in position, rotational angle, and length respectfully between the corotated configuration and the configuration after a small movement.

With this in mind, $\delta\mathbf{f}$ may be obtained via the chain rule:

$$\begin{bmatrix} \delta P \\ \delta M_1 \\ \delta M_2 \end{bmatrix} = \begin{bmatrix} EA/L_0 & 0 & 0 \\ 0 & 4EI/L_0 & 2EI/L_0 \\ 0 & 2EI/L_0 & 4EI/L_0 \end{bmatrix} \begin{bmatrix} \delta\Delta L \\ \delta\theta_{n,1} \\ \delta\theta_{n,2} \end{bmatrix} \tag{3.11}$$

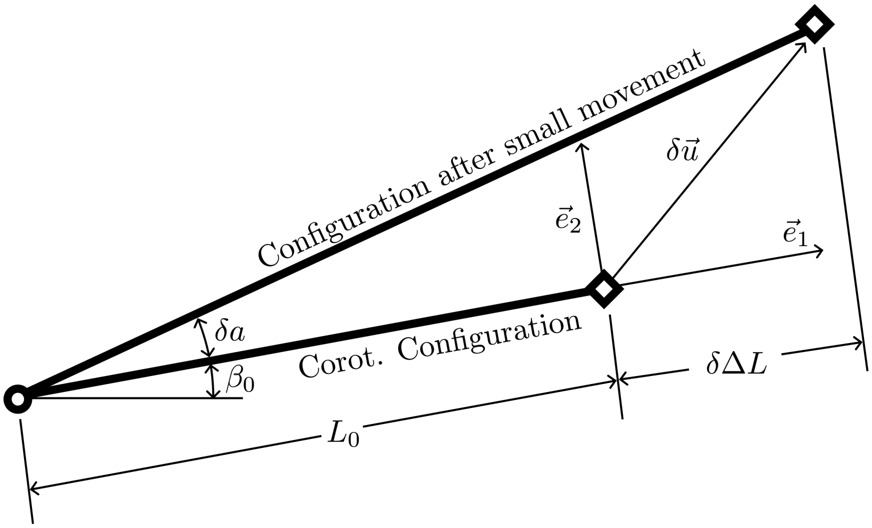


Figure 3.2. Perturbation of a corotated element

From fig. 3.2, it is apparent:

$$\vec{e}_1 = [\cos(\beta), \sin(\beta)]^T \tag{3.12}$$

$$\vec{e}_2 = [-\sin(\beta), \cos(\beta)]^T \tag{3.13}$$

$$\delta\mathbf{d} = [\delta\mathbf{d}_{x,2} - \delta\mathbf{d}_{x,1}, \delta\mathbf{d}_{y,2} - \delta\mathbf{d}_{y,1}]^T \tag{3.14}$$

From these terms, the relationship between the element elongation, $\delta\Delta L$, and the rigid body motion, $\delta\alpha$, and the nodal displacements in the global systems are:

$$\begin{aligned}\delta\Delta L &= [-\cos\beta, -\sin\beta, 0, \cos\beta, \sin\beta, 0] \cdot [\delta\mathbf{d}_{x,1}, \delta\mathbf{d}_{y,1}, \delta\theta_1 \delta\mathbf{d}_{x,2}, \delta\mathbf{d}_{y,2}, \delta\theta_2]^T \\ &= \mathbf{a} \cdot \delta\mathbf{d}_g && (3.15) \\ \delta\alpha &= \frac{1}{L}[\sin\beta, -\cos\beta, 0, -\sin\beta, \cos\beta, 0] \cdot [\delta\mathbf{d}_{x,1}, \delta\mathbf{d}_{y,1}, \delta\theta_1, \delta\mathbf{d}_{x,2}, \delta\mathbf{d}_{y,2}, \delta\theta_2]^T \\ &= \frac{1}{L}\mathbf{b} \cdot \delta\mathbf{d}_g && (3.16)\end{aligned}$$

Equations (3.15) and (3.16) are the expressions of the rigid-body motion of displacement and rotation for the elements. The non-rigid body rotation, θ_n, must also be translated into the global system. Using the global variant of the local corotated configurations described in eq. (3.6) with eq. (3.16) describes the angle variation in global coordinates as:

$$\begin{aligned}\begin{bmatrix} \delta\theta_{n,1} \\ \delta\theta_{n,2} \end{bmatrix} &= \left[\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{L} \begin{bmatrix} \mathbf{b}^T \\ \mathbf{b}^T \end{bmatrix} \right] \delta\mathbf{d}_g \\ &= \mathbf{A}^T \delta\mathbf{d}_g && (3.17)\end{aligned}$$

Combining eqs. (3.15) and (3.17) yields the relationship between the local and global displacements:

$$\begin{bmatrix} \delta\Delta L \\ \delta\theta_{n,1} \\ \delta\theta_{n,2} \end{bmatrix} = \begin{bmatrix} \mathbf{a}^T \\ \mathbf{A}^T \end{bmatrix} \delta\mathbf{d}_g \tag{3.18}$$

$$\delta\mathbf{d}_n = \mathbf{T}^T \delta\mathbf{d}_g \tag{3.19}$$

T describes the rotation operations necessary to translate the local displacement to the global coordinate system, which will appear in the subsequent force derivations. Its expanded form is:

$$\mathbf{T} = \begin{bmatrix} -\cos\beta & -\sin\beta/L & -\sin\beta/L \\ -\sin\beta & \cos\beta/L & -\cos\beta/L \\ 0 & 1 & 0 \\ \cos\beta & \sin\beta/L & \sin\beta/L \\ \sin\beta & -\cos\beta/L & -\cos\beta/L \\ 0 & 0 & 1 \end{bmatrix} \tag{3.20}$$

Translating forces applied to the element from the local system to the global system follows a similar procedure as transforming displacements. The forces initially applied in the local coordinate system are first transformed to the corotated coordinate system and then further transformed to the global coordinate system.

Consider a single reference element as shown in fig. 3.3. The grey forces represent the forces in the local coordinates, the black open arrows denote the forces in the corotated system, and the solid black arrows indicate the forces in the global system. It is apparent from fig. 3.3 that the relationship between the local and global

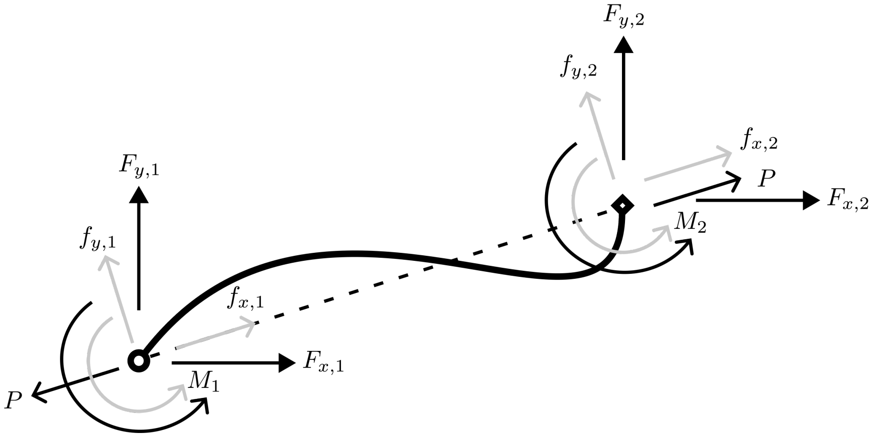


Figure 3.3. Internal and external forces experienced by the element

systems is:

$$f_{x,1} = -P, \quad f_{x,2} = P, \quad f_{y,1} = \frac{M_1 + M_2}{L} \quad f_{y,2} = -\frac{M_1 + M_2}{L} \tag{3.21}$$

In matrix form, eq. (3.21) becomes:

$$\begin{bmatrix} f_{x,1} \\ f_{y,1} \\ M_1 \\ f_{x,2} \\ f_{y,2} \\ M_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1/L & 1/L \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1/L & -1/L \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P \\ M_1 \\ M_2 \end{bmatrix}$$

$$\mathbf{f}_l = \mathbf{B}\mathbf{f}_d \tag{3.22}$$

Using the fig. 3.3, the global forces relate to the corotated forces by:

$$\begin{aligned}
F_{x,1} &= f_{x,1}\cos(\beta) - f_{y,1}\sin(\beta) \\
F_{y,1} &= f_{x,1}\sin(\beta) + f_{y,1}\cos(\beta) \\
F_{x,2} &= f_{x,2}\cos(\beta) - f_{y,2}\sin(\beta) \\
F_{y,2} &= f_{x,2}\sin(\beta) + f_{y,2}\cos(\beta)
\end{aligned} \tag{3.23}$$

In matrix form eq. (3.23) becomes:

$$\begin{bmatrix} F_{x,1} \\ F_{y,1} \\ M_1 \\ F_{x,2} \\ F_{y,2} \\ M_2 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & & & \\ \sin(\beta) & \cos(\beta) & 0 & & \text{zeros} & \\ 0 & 0 & 1 & & & \\ & & & \cos(\beta) & -\sin(\beta) & 0 \\ & \text{zeros} & & \sin(\beta) & \cos(\beta) & 0 \\ & & & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{x,1} \\ f_{y,1} \\ M_1 \\ f_{x,2} \\ f_{y,2} \\ M_2 \end{bmatrix}$$

$$\mathbf{f}_n = \mathbf{R}\mathbf{f}_l \tag{3.24}$$

To translate the natural forces to the global forces, eqs. (3.22) and (3.24) combine to yield:

$$\mathbf{f} = \mathbf{T}\mathbf{f}_n \tag{3.25}$$

where $\mathbf{T}$ is defined previously in eq. (3.20)

With the relationships between the natural and global coordinates known, the variation in forces as the beam deforms must be determined, and is obtained as:

$$\delta\mathbf{f} = \mathbf{T}\delta\mathbf{f}_n + \mathbf{d}; ta\mathbf{T}f_n \tag{3.26}$$

For derivation, $\mathbf{T}$ will be split into its columns, i.e. $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3]$. Starting with $\mathbf{t}_1$, $\delta\mathbf{t}_1$ is obtained as:

$$\begin{aligned}\delta\mathbf{t}_1 &= [\sin(\beta), -\cos(\beta), 0, -\sin(\beta), \cos(\beta), 0]^T \delta\beta \\ &= \mathbf{b}\delta\beta \end{aligned} \tag{3.27}$$

From fig. 3.1, it is apparent that $\delta\beta \equiv \delta\alpha$. This, and with eq. (3.16), eq. (3.27) becomes:

$$\delta\mathbf{t}_1 = \frac{1}{L}\mathbf{b}\mathbf{b}^T\delta\mathbf{d}_g \tag{3.28}$$

The second column, T_2 may be written as:

$$T_2 = -\frac{1}{L}\mathbf{b} + [0, 0, 1, 0, 0, 0]^T \tag{3.29}$$

Similarly, the third column, T_3, may be written as:

$$T_3 = -\frac{1}{L}\mathbf{b} + [0, 0, 0, 0, 0, 1]^T \tag{3.30}$$

The later vectors found in both eqs. (3.29) and (3.30) are constant, making their derivatives 0, making $\delta T_2 \equiv \delta T_3$. Knowing this, and using the product rule yields:

$$\delta T_{1,2} = \left(-\delta\left(\frac{1}{L}\right)\mathbf{b} + -\frac{1}{L}\delta\mathbf{b}\right) \tag{3.31}$$

The term $\delta(-1/L)$ in eq. (3.31) may be obtained as follows:

$$\delta\left(-\frac{1}{L}\right) = \frac{\delta L}{L^2} = \frac{\Delta L}{L^2} \tag{3.32}$$

Using eq. (3.15), eq. (3.32) becomes:

$$\delta\left(-\frac{1}{L}\right) = \frac{1}{L^2}\mathbf{a}\delta\mathbf{d}_g \tag{3.33}$$

The second unknown term in eq. (3.31) is $\delta\mathbf{b}$. Knowing $\delta\beta \equiv \delta\alpha$ and with eq. (3.16), $\delta\mathbf{b}$ is found as:

$$\delta\mathbf{b} = -\mathbf{a}\delta\alpha = -\frac{1}{L}\mathbf{a}\mathbf{b}^T\delta\mathbf{d}_g \tag{3.34}$$

With eqs. (3.33) and (3.34), eq. (3.31) becomes:

$$\delta T_{1,2} = \frac{1}{L^2}(\mathbf{b}\mathbf{a}^T + \mathbf{a}\mathbf{b}^T)\delta\mathbf{d}_g \tag{3.35}$$

At this point, there are no longer any unknowns which allows for the creation of the stiffness matrix.

3.2 Stiffness matrix derivation

To derive our final stiffness matrices, we return to eq. (3.26) and expand δT into each of its columns:

$$\delta\mathbf{f} = \mathbf{T}\delta\mathbf{f}_n + P\delta\mathbf{t}_1 + M_1\delta\mathbf{t}_2 + M_2\delta\mathbf{t}_3 \tag{3.36}$$

Substituting eqs. (3.10), (3.11), (3.19), (3.28) and (3.35) into eq. (3.36) yields $\delta\mathbf{f}$ for the element as:

$$\delta\mathbf{f} = \mathbf{T}\cdot\mathbf{C}\cdot\mathbf{T}^T\delta\mathbf{d}_g + \frac{P}{L}\mathbf{b}\mathbf{b}^T\delta\mathbf{d}_g + \frac{M_1+M_2}{L}\big(\mathbf{b}\mathbf{a}^T + \mathbf{a}\mathbf{b}^T\big)\delta\mathbf{d}_g \tag{3.37}$$

Knowing the equilibrium equation, $\mathbf{ku} = \mathbf{f}$, we may now obtain the final stiffness matrix $\mathbf{k}$ as:

$$\mathbf{k} = \underbrace{\mathbf{T} \cdot \mathbf{C} \cdot \mathbf{T}^T}_{\mathbf{k}_e} + \underbrace{\frac{P}{L}\mathbf{b}\mathbf{b}^T + \frac{M_1 + M_2}{L}\left(\mathbf{b}\mathbf{a}^T + \mathbf{a}\mathbf{b}^T\right)}_{\mathbf{k}_g} \tag{3.38}$$

Equation (3.39) shows the expanded matrix form of eq. (3.38).

$$\mathbf{k} = \begin{bmatrix} k_1 & -k_3 & -k_5 & -k_1 & k_3 & -k_5 \\ & k_2 & k_4 & k_3 & -k_2 & k_4 \\ & & k_7 & k_5 & -k_4 & k_6 \\ & & & k_1 & -k_3 & k_5 \\ & \text{sym.} & & & k_2 & -k_4 \\ & & & & & k_7 \end{bmatrix} \tag{3.39}$$

where $k_i = k_{i,e} + k_{i,g}$, with:

$$
\begin{aligned}
k_{1,e} &= (EAL^2 \cos^2(\beta) + 12EI \sin^2(\beta))/(L^2 L_0) \\
k_{2,e} &= (EAL^2 \sin^2(\beta) + 12EI \cos^2(\beta))/(L^2 L_0) \\
k_{3,e} &= ((12EI - EAL^2) \cos(\beta) \sin(\beta))/(L^2 L_0) \\
k_{4,e} &= (6EI \cos(\beta))/(LL_0) \\
k_{5,e} &= (6EI \sin(\beta))/(LL_0) \\
k_{6,e} &= (2EI)/(L_0) \\
k_{7,e} &= (4EI)/(L_0) \\
k_{1,g} &= (PL \sin^2(\beta) - 2(M_1 + M_2) \cos(\beta) \sin(\beta))/(L^2) \\
k_{2,g} &= (PL \cos^2(\beta) + 2(M_1 + M_2) \cos(\beta) \sin(\beta))/(L^2) \\
k_{3,g} &= (PL \cos(\beta) \sin(\beta) - (M_1 + M_2)(\cos^2(\beta) - \sin^2(\beta)))/(L^2) \\
k_{4,g} &= k_{5,g} = k_{6,g} = k_{7,g} = 0
\end{aligned}
\tag{3.40}
$$

For understanding, the obtained governing equation (eq. (3.38)) and stiffness matrix (eq. (3.39)) can be divided into two components: $\mathbf{k}_e$ which represents the stiffness due to the elastic behavior of the element, while $\mathbf{k}_g$ accounts for the stiffness resulting from internal loads and deformations experienced by the element.

Consider an arbitrary beam aligned to the global x-axis, i.e. $\beta = 0$. This simplification significantly reduces the stiffness coefficients presented in eq. (3.40). The resulting stiffness matrices for linear-elastic and nonlinear behavior are provided in eq. (3.41) and eq. (3.42) respectively.

$$
\mathbf{k}_e = \begin{bmatrix}
\frac{EA}{L_0} & 0 & 0 & -\frac{EA}{L_0} & 0 & 0 \\
 & \frac{12EI}{L_0^2 L} & \frac{6EI}{L_0 L} & 0 & -\frac{12EI}{L_0^2 L} & \frac{6EI}{L_0 L} \\
 & & \frac{4EI}{L_0} & 0 & -\frac{6EI}{L_0 L} & \frac{2EI}{L_0} \\
 & & & \frac{EA}{L_0} & 0 & 0 \\
 & \text{sym.} & & & \frac{12EI}{L_0^2 L} & -\frac{6EI}{L_0 L} \\
 & & & & & \frac{4EI}{L_0}
\end{bmatrix} \tag{3.41}
$$

$$
\mathbf{k}_g = \begin{bmatrix}
0 & \frac{(M_1+M_2)}{L^2} & 0 & 0 & -\frac{(M_1+M_2)}{L^2} & 0 \\
 & \frac{P}{L} & 0 & -\frac{(M_1+M_2)}{L^2} & -\frac{P}{L} & 0 \\
 & & 0 & 0 & 0 & 0 \\
 & & & 0 & \frac{(M_1+M_2)}{L^2} & 0 \\
 & \text{sym.} & & & \frac{P}{L} & 0 \\
 & & & & & 0
\end{bmatrix} \tag{3.42}
$$

This subsection derived the stiffness matrix for a single element using the co-rotation formulation, which is employed for nonlinear analysis in chapter 6. The stiffness matrices for linear-elastic elements used in chapter 5 are obtained by setting the internal forces and moments to zero in eq. (3.39), as derived in the subsequent subsection.

3.3 Linear Element Formulation

The previous subsection defined the stiffness matrix using the co-rotation formulation, which accounts for stiffness variations due to internal deformation and geometric changes when the element is loaded. In contrast, the linear-elastic elements analyzed in chapter 5 assume constant stiffness regardless of loading. This behavior can be viewed as a special case of the stiffness matrices derived from the co-rotation formulation, achieved by setting the internal loads and deformations experienced by the element to zero, i.e., $P = M_1 = M_2 = 0$ and $\mathbf{u} = 0$.

To obtain the stiffness matrices for linear-elastic elements, setting $\mathbf{u} = 0$ in eq. (3.4) reveals $L_0 = L$. Similarly, setting $P = M_1 = M_2 = 0$ in eq. (3.40) yields $k_g = 0$. Thus, the final stiffness matrix for linear-elastic elements is:

$$\mathbf{k}_{le} = \begin{bmatrix} k_{1,le} & -k_{3,le} & -k_{5,le} & -k_{1,le} & k_{3,le} & -k_{5,le} \\ & k_{2,le} & k_{4,le} & k_{3,le} & -k_{2,le} & k_{4,le} \\ & & k_{7,le} & k_{5,le} & -k_{4,le} & k_{6,le} \\ & & & k_{1,le} & -k_{3,le} & k_{5,le} \\ & \text{sym.} & & & k_{2,le} & -k_{4,le} \\ & & & & & k_{7,le} \end{bmatrix} \tag{3.43}$$

with:

$$\begin{aligned} k_{1,le} &= (EAL^2\cos^2(\beta) + 12EI\sin^2(\beta))/(L_0^3) \\ k_{2,le} &= (EAL^2\sin^2(\beta) + 12EI\cos^2(\beta))/(L_0^3) \\ k_{3,le} &= ((12EI - EAL^2)\cos(\beta)\sin(\beta))/(L_0^3) \\ k_{4,le} &= (6EI\cos(\beta))/(L_0^2) \\ k_{5,le} &= (6EI\sin(\beta))/(L_0^2) \\ k_{6,le} &= (2EI)/(L_0) \\ k_{7,le} &= (4EI)/(L_0) \end{aligned} \tag{3.44}$$

The stiffness matrix for an arbitrary element aligned with the x-axis may be seen in eq. (3.45), which yields a similar result to eq. (3.41).

$$
\mathbf{k}_e = \begin{bmatrix}
\frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\
 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\
 & & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\
 & & & \frac{EA}{L} & 0 & 0 \\
 & \text{sym.} & & & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\
 & & & & & \frac{4EI}{L}
\end{bmatrix} \tag{3.45}
$$

With eq. (3.39) defining the stiffness matrix for nonlinear elements and eq. (3.43) defining the stiffness matrix for linear-elastic elements, the complete global stiffness matrix for the entire network for each formulation can be constructed by assembling the local stiffness matrices **k** of each element into the global stiffness matrix **K** in the usual manor.

Using the stiffness matrices derived in this section, the displacement field of a network of beams can be computed. In the case of linear-elastic elements, the solution to the system of equations represented by the stiffness matrix can be directly obtained since all coefficients in **K** are known. However, for nonlinear elements, certain coefficients in the stiffness matrix are unknown. For instance, as defined in eq. (3.4), L depends on the current deformations $\mathbf{u}$ experienced by the element which is unknown. Consequently, the remainder of chapter 3 will detail the solution method for nonlinear analysis, with section 3.4 outlining the solution algorithm and summarizing the analysis process for nonlinear elements.

For readers primarily interested in the linear topology optimization method and its results, it is recommended to proceed directly to section 4.6 for the optimization method or to chapter 5 for the optimization results.

3.4 Nonlinear Finite Element Solution

In the previous section, the local stiffness matrix for a single element was derived using the co-rotational formulation which includes both elastic and geometric stiffness coefficients. The elastic coefficients represent the inherent stiffness properties of the structure, while the geometric stiffness matrix accounts for stiffness due to deformations and internal forces and moments. Since the geometric stiffness coefficients depend on the current deformation of the structure, which is initially unknown, the system cannot be solved analytically and requires numerical methods for solution

The method selected for this work is the Newton-Raphson method, in finite elements commonly known as the load control method. It begins with the structure in its unloaded state, where increments of total load are applied sequentially. The system of equations is iteratively solved until the internal and external forces balance, achieving an equilibrium state. Once equilibrium is attained for a particular load increment, the next increment is applied, and the iterative process repeats to find a new equilibrium point. This process continues until the target load is reached, resulting in a sequence of equilibrium points that trace the structure's equilibrium path.

To achieve equilibrium, the internal forces and moments within the structure must balance the externally applied loads, expressed as $\mathbf{f} = \mathbf{p}$, where $\mathbf{f}$ is the vector describing the internal forces and moments at the degrees of freedom of the structure, and $\mathbf{p}$ is the external load applied to the structure. As explained in the preceding chapter, the internal forces experienced by the beam element are dependent on its displacement, hence $\mathbf{f}(\mathbf{u})$. Introducing a load parameter, λ, which ranges from 0 (unloaded) to 1 (fully loaded), allows us to express the equilibrium equation as shown ineq. (3.46).

$$\mathbf{f}(\mathbf{u}) = \mathbf{p}(\lambda) \tag{3.46}$$

To obtain the equilibrium path, the values of $\mathbf{p}$ and $\mathbf{u}$ are incrementally increased. This is achieved using increments denoted as $\Delta\mathbf{p}$ and $\Delta\mathbf{u}$. The values at the current increment are denoted by subscript i, and subscript $i-1$ refers to the values at the previous step. Therefore, the values of $\mathbf{p}$ and $\mathbf{u}$ at the current and previous increments can be expressed as follows:

$$\begin{aligned} \mathbf{p}_i &= (\lambda_{i-1} + \Delta\lambda_i)\mathbf{p} \\ \mathbf{u}_i &= u_{i-1} + \Delta u_i \end{aligned} \tag{3.47}$$

With the structure being in equilibrium at each previous step, the equilibrium equation for the i-th step involves only the changes in the load ratio λ and the nodal displacements $\mathbf{u}$. Therefore, it can be expressed as:

$$\mathbf{f}(\Delta\mathbf{u}_i) = \Delta\lambda_i\mathbf{p} \tag{3.48}$$

Here, $\mathbf{f}_i(\mathbf{u}_i)$ represents the internal forces and moments as a function of the nodal displacements $\mathbf{u}_i$, and $\mathbf{p}_i$ denotes the applied external load at the i-th step. This equation ensures that the forces within the structure are in equilibrium with the applied loads for each incremental step during the Newton-Raphson process.

The stiffness matrix $\mathbf{K}$ obtained in the previous chapter relates the forces to the nodal displacements when can be used to predict the solution at the next increment. From the equilibrium condition at the i-th step, we derive two key equations: one relating the internal forces $\mathbf{f}_i$ to the nodal displacements $\mathbf{u}_i$, given by eq. (3.49), and another relating the applied external load $\mathbf{p}_i$ to the stiffness matrix $\mathbf{K}$, expressed by eq. (3.50).

$$\mathbf{f}(\Delta\mathbf{u}_i) = \mathbf{K}_{i-1}\Delta\mathbf{u}_i \tag{3.49}$$

$$\mathbf{K}_{i-1}\Delta\mathbf{u}_i = \Delta\lambda_i\mathbf{p} \tag{3.50}$$

Since the geometric stiffness coefficients within **K** depend on the current deformation increment, a difference arises between Equations 3.49 and 3.50, known as the residual. This residual represents the imbalance between the internal forces $\mathbf{f}_i$ computed from the displacements $\mathbf{u}_i$ and the applied external loads $\mathbf{p}$. Achieving equilibrium at each step requires minimizing this residual through Newton-Raphson iterations until a new equilibrium state is found.

Similar to the increments described in the previous subsection, the load ratio, λ, and nodal displacements, $\mathbf{u}$, are updated incrementally within each step. For a given step indexed by subscript i, the superscript j denotes iterations within that step. Thus, the load ratio and displacement at the j-th iteration within step i are:

$$\begin{aligned} \Delta\bar{\lambda}_i^j &= (\Delta\lambda_i^{j-1} + \delta\lambda_i^j) \\ \Delta\mathbf{u}_i^j &= (\Delta\mathbf{u}_i^{j-1} + \delta\mathbf{u}_i^j) \end{aligned} \tag{3.51}$$

Once an equilibrium state has been found, the values for λ and $\mathbf{u}$ are updated for use in the next step. At convergence, typically denoted by the j-th iteration within each step i, the total updated values used in the analysis are obtained as:

$$\begin{aligned} \bar{\lambda}_i^j &= (\lambda_i^{j-1} + \Delta\lambda_i^j) \\ \mathbf{u}_i^j &= (\mathbf{u}_i^{j-1} + \Delta\mathbf{u}_i^j) \end{aligned} \tag{3.52}$$

The aforementioned residual is defined as the difference between the external and internal forces, as shown in eq. (3.53). Assuming a small step-size, the residual forces can be approximated by a Taylor series expansion considering the first derivatives, as depicted in eq. (3.54).

$$\mathbf{r}_i^j = \lambda_i^j \mathbf{p} - \mathbf{f}(\mathbf{u}_i^j) \tag{3.53}$$

$$\mathbf{r}_i^j \approx \mathbf{r}_i^{j-1} + \frac{\partial \mathbf{r}}{\partial \mathbf{u}} \delta U_i^j + \frac{\partial \mathbf{r}}{\partial \lambda} \delta \lambda_i^j + \text{H.O. terms} \tag{3.54}$$

The first derivative present in the Taylor series expansion is obtained as:

$$\frac{\partial \mathbf{r}}{\partial \mathbf{u}} = (-\frac{\partial \mathbf{f}}{\partial \mathbf{u}}) = -\mathbf{K}_i^{j-1} \tag{3.55}$$

The second derivative present in the Taylor series expansion is obtained as:

$$\frac{\partial \mathbf{r}}{\partial \lambda} = (\frac{\partial \mathbf{p}}{\partial \lambda}) = \mathbf{p} \tag{3.56}$$

Knowing the goal is reduce the residual to zero, eqs. (3.54) to (3.56) may be combined to form eq. (3.57) which is used to calculate the iterative displacement value, $\delta \mathbf{u}_i^j$. To begin the iterative process, the values obtained in the previous step for λ and $\mathbf{u}$ are used.

$$\mathbf{K}_i^{j-1} \delta \mathbf{u}_i^j = \delta \lambda_i^j \mathbf{p} + \mathbf{r}_i^{j-1} \tag{3.57}$$

The system of equations described in eq. (3.57) consists of N equations, one for each degree of freedom in $\mathbf{u}$, but it involves $N + 1$ unknowns: the displacements $\delta \mathbf{u}$ associated with the degrees of freedom and the increment in the loading ratio $\delta \lambda_i^j$. This necessitates an additional equation to define the increment of the loading ratio [29].

Various methods have been considered to determine $\delta\lambda_i^j$, particularly around regions of potential instability [29]. Instabilities like snap-through, where a structure buckles suddenly, can lead to singular stiffness matrices and pose challenges in both analysis and optimization [4]. Near these critical points, the stiffness matrix becomes ill-conditioned, affecting the accuracy of analysis results.

To mitigate numerical issues stemming from instability, this paper introduces a buckling constraint in the topology optimization (section 4.1.2) to prevent critical loads-limits from being exceeded. Consequently, the structures analyzed in this study are unlikely to reach critical points, reducing the robustness requirements for the analysis solution algorithm. Therefore, this paper employs the conventional Newton-Raphson method, also known as the loading control method.

The loading control method applies a fixed load increment in each step, which remains constant throughout the iteration process, effectively setting $\delta\lambda_i^j = 0$. This simplification allows solving a single linear system at each iteration, significantly enhancing computational efficiency compared to other methods, which is advantageous for subsequent optimization processes.

With the steps and iterations defined in this subsection, and the stiffness matrices established in the previous section (section 3.1), the methodology for conducting nonlinear finite element analysis is complete. In summary, an applied load is incrementally increased in a series of steps. At the beginning of each step, the stiffness matrix is computed using the displacement field from the previous step. An iterative process then seeks equilibrium, and once achieved, the next step begins with an incremented load. This iterative sequence is illustrated in fig. 3.4.

A tool was created in MATLAB to perform the analysis for both linear-elastic elements and nonlinear geometric elements. The final subsection in this chapter, section 3.5, the tool will be validated against a benchmark problem to ensure its accuracy and reliability. The remaining chapters will implement said tool in their respective topology optimization problems.

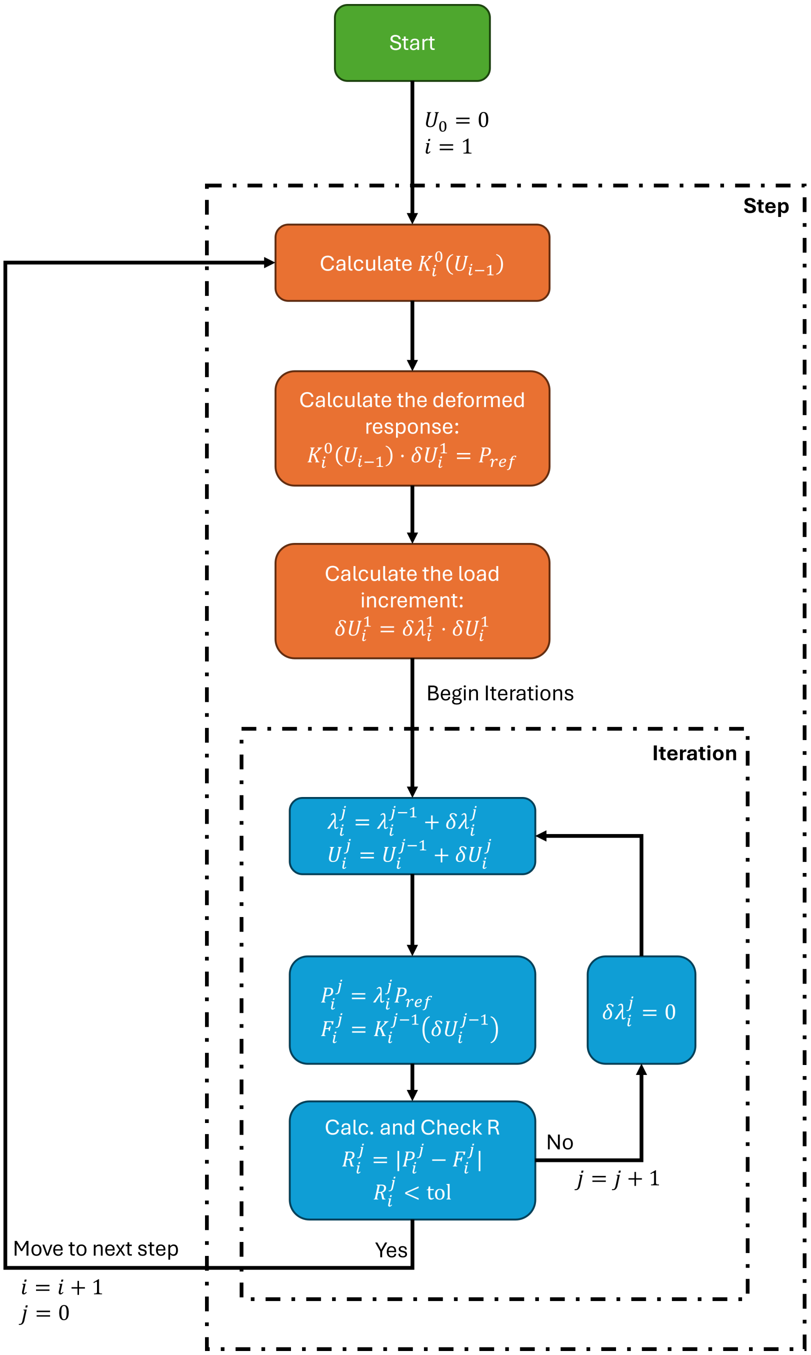

Start
$U_0 = 0$
$i = 1$
Step
Calculate $K_i^0(U_{i-1})$
Calculate the deformed response:
$K_i^0(U_{i-1}) \cdot \delta U_i^1 = P_{ref}$
Calculate the load increment:
$\delta U_i^1 = \delta\lambda_i^1 \cdot \delta U_i^1$
Begin Iterations
Iteration
$\lambda_i^j = \lambda_i^{j-1} + \delta\lambda_i^j$
$U_i^j = U_i^{j-1} + \delta U_i^j$
$P_i^j = \lambda_i^j P_{ref}$
$F_i^j = K_i^{j-1}(\delta U_i^{j-1})$
$\delta\lambda_i^j = 0$
Calc. and Check R
$R_i^j = |P_i^j - F_i^j|$
$R_i^j <$ tol
No
$j = j + 1$
Yes
Move to next step
$i = i + 1$
$j = 0$

Figure 3.4. Analysis algorithm used to produce the equilibrium path considering nonlinear geometric elements.

3.5 Benchmark Problem

To validate our implementation of the nonlinear beam solver, this section investigates a common benchmark problem featuring a cantilever beam with a concentrated moment applied to its tip, as shown in fig. 3.5. With specific properties, this structure deforms in a perfect circle [29, 32].

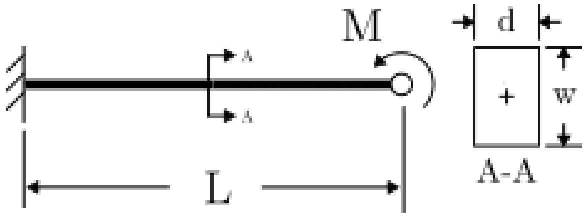
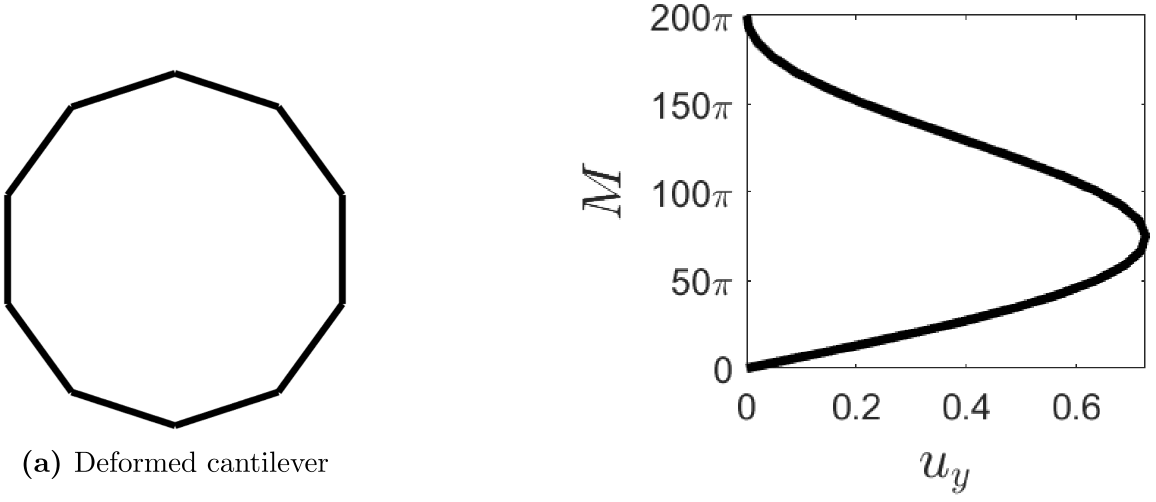


Figure 3.5. FBD of a cantilever with a concentrated moment

Setting $L = 1, w = \frac{1}{5}\sqrt{\frac{3}{10}}, w = \frac{1}{2\sqrt{30}}$ and $M = 200\pi$ results in the final deformed geometry shown in fig. 3.6a whose force-displacement relation of the end-tip is given by fig. 3.6b.



(a) Deformed cantilever

(b) Force-displacement relation

Figure 3.6. Deformed cantilever and equilibrium path for a cantilever with a concentrated moment

With λ referring to the loading ratio, the motion of the deformation can be visualized by fig. 3.7.

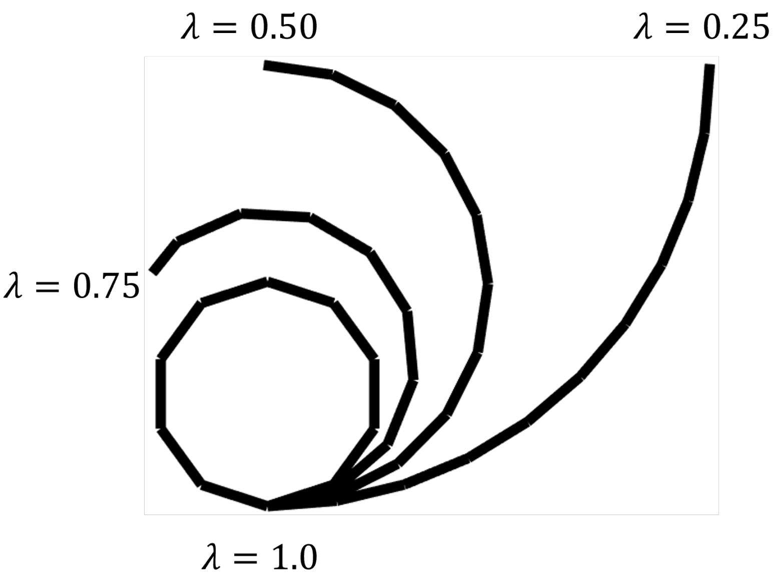


Figure 3.7. Motion of the cantilever beam

The results illustrated in figs. 3.6 and 3.7 align with both analytical and finite element outcomes, as referenced in [33] and [32].

Utilizing the beam network representation described in the preceding chapter (chapter 2) in conjunction with the finite element formulation discussed in the current chapter, we have established a comprehensive procedure for both the construction and physical analysis of a beam network. The subsequent chapter will finalize the formulation of the proposed method and elaborate on the optimization framework, including the constraints, methodology, and gradients required to adjust the topology in accordance with two optimization objectives.

Chapter 4

Optimization Formulation

With the topology formulation described in chapter 2 and the analysis method described in chapter 3, this chapter focuses on the optimization methodology employed to explore various optimizations in subsequent chapters.

The beam network under consideration is defined by the positions of its nodes, the connections between these nodes, and the distribution of material across these connections. Material distribution within the network is characterized by the width, depth, and density of each beam element. This study manipulates the network by adjusting node positions and beam widths, while maintaining fixed connections, structural depth, and material density. Although optimizing these additional parameters could expand the range of representable topologies, it is outside the scope of this work.

Here, $\mathbf{x}$ denotes the set of node positions along the x-axis, $\mathbf{y}$ along the y-axis, and $\mathbf{w}$ represents the set of beam widths. Collectively, $\mathbf{z}$ encompasses all optimization variables: $\mathbf{z} = \{\mathbf{x}, \mathbf{y}, \mathbf{w}\}$.

This chapter is divided into four sections. The first section, section 4.1, introduces the constraints shared between both optimization investigations. The subsequent sections, section 4.2 and section 4.4, focus on compliance and targeted deformations respectively and provide formal optimization definitions, with section 4.3 and section 4.5 providing the necessary derivatives for linear and nonlinear respectfully. Finally, section 4.6 provides an overview of the general optimization algorithm employed.

4.1 Constraints

In optimization, a constraint is a condition or limitation imposed on the solution space to ensure that the results are feasible and adhere to specific requirements, viability, or for numerical stability. In the context of a morphing beam network investigated in this work, we have identified four constraints that are shared across both optimization problems: a volume constraint, a width constraint, and two node position constraints. For nonlinear deformations, there is an additional width constraint to prevent a sudden loss of stiffness. For organization, these constraints are described in the following subsections, section 4.1.1 discusses the volume constraint, section 4.1.2 discusses a common width constraint followed by an additional constraint for nonlinear geometries, and the node constraints are discussed in section 4.1.3.

4.1.1 Volume Constraint

A fundamental constraint in topology optimization is limiting the amount of material within the domain, commonly known as a volume constraint [6]. Without such a constraint, beams may distribute material in unrealistic proportions, leading to impractical or degenerate solutions. Therefore, the volume contained within Ω_s is constrained to a user-defined ratio of the total volume of Ω:

$$\frac{V(\mathbf{z})}{V_0} \leq V_f \tag{4.1}$$

where V_f is the user-defined volume fraction, V_0 is the total volume of Ω, and $V(\mathbf{z})$ being the volume of material present in Ω_s

This constraint can be included into the problem by the Lagrange multiplier method. As to not lose generality, obj refers to the objective function of the current study.

$$\mathcal{L}(\mathbf{z}, \Lambda) = \mathrm{obj}(\mathbf{z}) + \Lambda(V(\mathbf{z}) - V_f) \tag{4.2}$$

where $\mathcal{L}$ denotes the Lagrange function and Λ is the Lagrangian multiplier.

The Karush–Kuhn–Tucker first-order optimality condition of eq. (4.2) becomes [34]:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{z}} = \frac{\partial \mathrm{obj}(\mathbf{z})}{\partial \mathbf{z}} + \Lambda \frac{\partial V(\mathbf{z})}{\partial \mathbf{z}} = 0 \tag{4.3}$$

$$\frac{\partial \mathcal{L}}{\partial \Lambda} = V(\mathbf{z}) - V_f = 0 \tag{4.4}$$

From eq. (4.3), the derivatives of both the objective function and volume with respect to the structure's design variables, $\mathbf{z}$, are required. With the derivatives of the objective functions presented in sections 4.2 and 4.4, this section will obtain the derivatives for volume with respect to $\mathbf{z}$. With three types of variables, x location, y location, and width, three different of derivatives are obtained.

Using eq. (2.4), the derivative of volume with respect to the design variables

$\mathbf{z}$ are found as:

$$\frac{\partial V}{\partial w_i} = d_i L_i \tag{4.5}$$

$$\frac{\partial V}{\partial x_n} = \sum_{e;n_1,n_2 \in e}^{\text{num. elem.}} \frac{w_e d_e}{L_e}(x_1 - x_2) \tag{4.6}$$

$$\frac{\partial V}{\partial y_n} = \sum_{e;n_1,n_2 \in e}^{\text{num. elem.}} \frac{w_e d_e}{L_e}, (y_1 - y_2) \tag{4.7}$$

where $\frac{\partial V(\mathbf{z})}{\partial \mathbf{z}} = \{\frac{\partial V}{\partial \mathbf{x}}, \frac{\partial V}{\partial \mathbf{y}}, \frac{\partial V}{\partial \mathbf{w}}\}$, $n_1, n_2 \in e$ refers to the nodes that make up element e

This completes the volume fraction constraint. The next subsection will discuss the constrains applied to beam width.

4.1.2 Beam Width Constraints

Material distribution within the network is determined by the width of each element, stored in the vector $\mathbf{w}$. This work considers two types of constraints on the beam widths: one constraint is common to both problems, while the other is specific when considering nonlinear deformation.

The first constraint is the positivity of widths: negative widths are physically incorrect. Therefore, beam widths are constrained to be positive. While having no width is representative of the removal of the beam from the network, setting width of an element to zero represents numerical challenges. To maintain numerical stability, beam widths are constrained to be greater than a small value, denoted as w_{min}, which is set slightly above zero (e.g., $w_{\text{min}} << 1$), akin to methods such as SIMP [6]. Thus, the first constraint applied to beam widths in the network is:

$$0 < w_{\text{min}} < \mathbf{w} \tag{4.8}$$

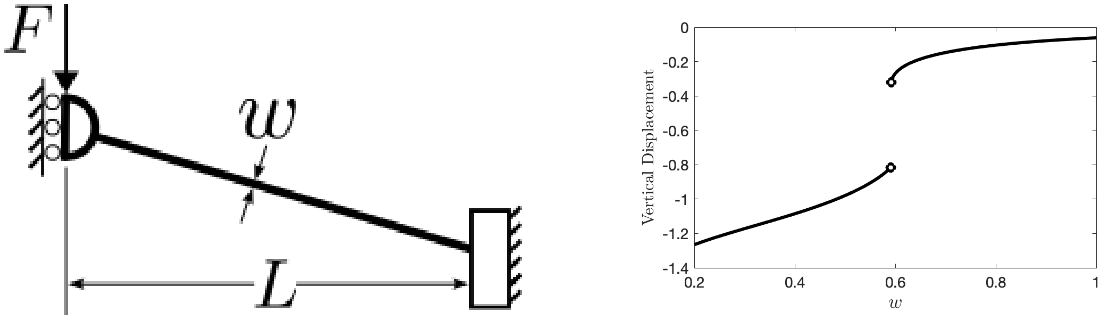
For those interested solely in linear-elastic compliance, it is recommended to skip to the next section, section 4.1.3, as the next constraint only applies to nonlinear deformations.

4.1.2.1 Nonlinear Buckling Constraint

Depending on a structure's geometry and loading conditions, certain elements may undergo a sudden loss of stiffness when loaded, commonly referred to as buckling or snap-through [4]. While these are valid mechanical responses to applied loads, this abrupt loss of stiffness poses significant challenges in optimization.

Consider William's Frame [35], a simple structure exhibiting snap-through. A free-body diagram of this frame is shown in fig. 4.1a. Setting $F = 100$, $E = 1 \times 10^7$, $d = 0.6$, and $L = 30$, and varying the element widths w within the range $w \in \{0.2, 1.0\}$ and measuring the vertical displacement yields the plot shown in fig. 4.1b. This plot shows an abrupt change and discontinuity at $w \approx 0.59$ as the beam loses or gains snap-through.

At the moment of snap-through, there is a sudden and abrupt change in displacement, showing a lack of solutions for vertical displacements that lie between the critical thresholds. Additionally, at the point of snap-through both the displacement and the derivative are undefined, which poses a direct challenge in optimization. Given these issues, this paper introduces a constraint to prevent elements in the structure from undergoing snap-through.



(a) FBD for William's frame

(b) Vertical Response vs beam width

Figure 4.1. FBD for William's frame and its corresponding design space

Assuming the axial load applied to each element determines the stability of that element, Euler's critical load is used. From classical mechanics, the critical load for an axially loaded fixed-free rectangular column is determined by:

$$P_{cr} = \frac{\pi^2 E w^3 d}{24L^2} \tag{4.9}$$

Since L is dependent on the node locations of the element (eq. (3.4)) all optimization variables are present in eq. (4.9). However, because the positions of nodes affect multiple elements, applying the buckling constraint directly to node positions can adversely affect other parts of the structure. Therefore, the buckling constraint will not be applied to node positions but will instead be enforced by constraining the beam widths of the elements.

Before eq. (4.9) can be used to calculate the minimum widths, there are several additional things to consider. First, while frequently the maximum axial load an element experiences does occurs on the last time step, occasionally it may occur earlier as structure bends to the applied load. Therefore, to calculate minimum width, w_{cr}, the maximum axial load occurring across each time-step in finite elements is used. Additionally, across optimization iterations, changes in the design variables will cause the axial load experienced by the elements to change slightly, potentially increasing the load beyond the calculated minimum width, ergo for robustness, the calculated width is scaled by 1.5. Finally, elements within the structure may experience tension instead of compression. When that occurs, the elements are instead constrained to be larger than the previously defined value, w_{min}. Additionally it should be noted that while the active structure described in section 2.4 is used in analysis, this buckling constraint only considers axial loads which are identical across each beam for the element.

With these in mind, the minimum widths of each element are set as:

$$\mathbf{w}_{cr} = \begin{cases} 2\sqrt[3]{\frac{4\max(\mathbf{P})L^2}{E\pi^2}} & \max(\mathbf{P}) > 0 \\ w_{\min} & \max(\mathbf{P}) < 0 \end{cases} \tag{4.10}$$

where $\mathbf{P}$ refers to the set of all axial loads experienced by the elements of the structure throughout loading.

With the constraints applied to the width of the structure defined, the following subsection will discuss constraints applied to the nodal positions of the structure.

4.1.3 Node Position Constraints

To modify the shape of the network, the x and y coordinates of the nodes constituting the network are adjustable, but like the widths of the beams, they are subject to constraints.

Firstly, to ensure the structure's topology remains within Ω, the nodes must lie within Γ. This is achieved by enforcing the nodes to be clipped to the nearest point on Γ, a process commonly known as 'clipping'. An illustration of this process is shown in fig. 4.2, where the solid line represents the boundary Γ, the solid dot depicts the node's original position, and the dotted arrow indicates the direction in which the node would move. If the node would move outside Γ, as depicted by the dashed circle, it is instead clipped to the nearest point on Γ, denoted by the grey point.

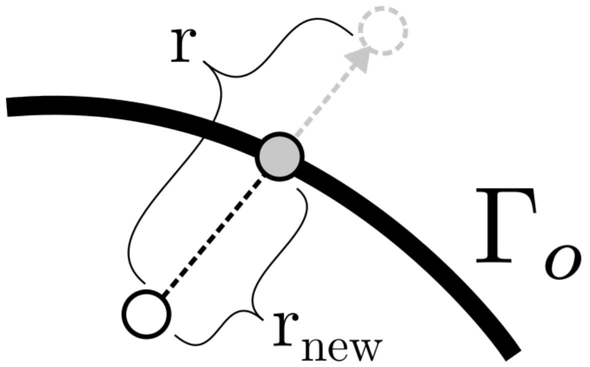


Figure 4.2. A clipping method is used to constrain nodes to lie within Ω

In general, this method works by preserving the direction the node travels in, determined by its gradient, but limits the amount such that it remains within Ω. While this process can be achieved with in a cartesian system, it is which is more easily performed considering polar coordinates. Let the amount the node wishes to move be given by $\vec{r}$, the length of the change be given by r, and the angle of the vector be given by γ_r, the final position of the node can be given by $r \cdot [\cos(\gamma_r), \cos(\gamma_r)]$. To ensure that the node does not move outside Γ, r is reduced until it lies on the edge of Γ, this distance is called r_{new}, which can easily be found by a 1D line search. This makes the final position of the node obtained by:

$$(\mathbf{x}_{\text{clip}}, \mathbf{y}_{\text{clip}}) = r_{\text{new}}(\cos(\gamma_r), \cos(\gamma_r)) \tag{4.11}$$

The second constraint imposed on the nodes aims to prevent numerical issues that arise when two or more nodes become coincident. Elements formed by coincident nodes have zero length, leading to numerical instability similar to how a zero width does. While various alternatives exist to constrain nodal positions, this work employs a method that restricts node positions to regions closest to their original locations, commonly known as Voronoi cells.

Using the previous example structure shown in fig. 2.5b, the structure along with the Voronoi cells, whose boundaries are displayed as grey dashes, is shown in fig. 4.3.

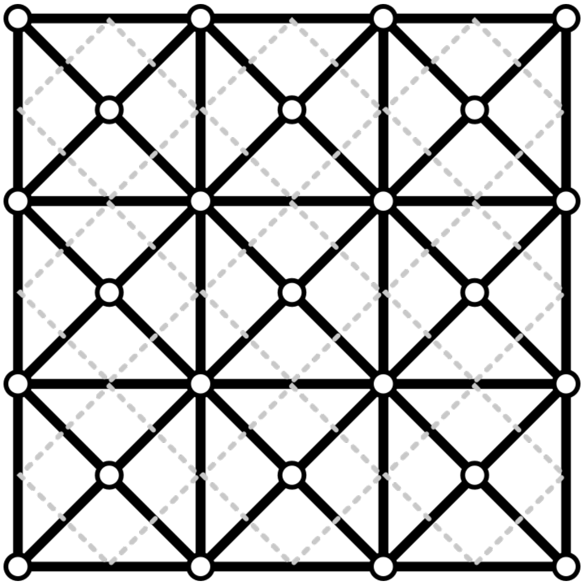


Figure 4.3. Example structure with highlighted Voronoi edges

Upon closer inspection of fig. 4.3, the Voronoi cells are arranged in a uniform square grid, though at a 45° angle. To restrict the nodes from leaving their Voronoi cells, let the minimum distance from the cell edge to the node location be defined as p_c, the original x and y locations of the nodes defined as $\mathbf{x}_{nat}$ and $\mathbf{y}_{nat}$, and a rotation matrix T_{45} defined by:

$$\mathbf{T}_{45} = \begin{bmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \tag{4.12}$$

The Voronoi constraint may be achieved by:

$$[\mathbf{x}, \mathbf{y}] = \min(p_c, \max(-p_c, [(\mathbf{x}_{nat}, \mathbf{y}_{nat}) - (\mathbf{x}, \mathbf{y})] \cdot T_{45})) \cdot T_{45}^T \tag{4.13}$$

While not a constraint, to prevent updates to the boundary conditions the positions of the nodes with applied forces or known displacements are excluded from having their positions updated.

For brevity, eqs. (4.11) and (4.13) will be combined into a relation given in eq. (4.14), where Ω_v refers to the valid region in Ω node $\mathbf{n}$ is in.

$$\mathbf{n} \in \Omega_v \tag{4.14}$$

With each variable type's constraints defined, this completes the definitions for constraints used in the work. The next section discusses the step-sized used in optimization. Following that, the formal optimization definitions are provided.

4.1.4 Step Size and Variable Updates in Optimization

In optimization, step size refers to the magnitude of the change applied to variables in each iteration of the algorithm, determining the distance each variable moves in the direction of the gradient [5]. To ensure smooth and stable optimization, a maximum step size, denoted as $\Delta\mathbf{z}$, restricts how much each variable can change between iterations.

Through experimentation, it was observed that setting the maximum step size, $\Delta x, \Delta y$, to be less than or equal to 1% of the initial element length yields effective results. However, applying the same principle to Δw becomes challenging as the number of elements increases exponentially with mesh density. Initially, with uniformly distributed material across the mesh, more elements result in smaller initial widths, thereby necessitating very small Δw values. To address practical concerns, separate step sizes are employed for x, y, and w, where $\Delta\bar{p}$ scales with the initial element length, while Δw remains constant regardless of mesh density.

In a broad sense, the algorithm used to generate the results discussed in subsequent chapters operates with two main loops. The outer loop calculates both the mechanical response and objective function values for the structure in each optimization iteration using current design variables $\mathbf{z}$. The inner loop updates the Lagrangian multiplier to satisfy volume constraints outlined in the second term of eq. (4.3). This adjustment involves applying a scaling factor to each design variable, as detailed in [34].

$$s = \frac{\frac{\partial}{\partial \mathbf{z}}\text{obj}}{\Lambda \frac{\partial}{\partial \mathbf{z}} V} \tag{4.15}$$

Combining these three steps together, $\mathbf{z}_{new}$ is found by:

$$\mathbf{z}_{new} = \mathbf{z}_{min} \leq \mathbf{z} - \Delta\mathbf{z} \leq \mathbf{z} \times \sqrt{s} \leq \mathbf{z} + \Delta\mathbf{z} \leq \mathbf{z}_{max}, \quad \mathbf{z} \in \{\bar{p}, w\} \tag{4.16}$$

with the square root acting as a numerical dampener.

This concludes the discussions of the constraints and variable updates. The remainder of the chapter discusses the optimization definitions for both optimization problems, with the next section discussing compliance minimization, and section 4.4 discussing targeted deformations.

4.2 Linear Compliance Minimization

Compliance minimization is a widely studied objective in structural optimization, particularly in topology optimization. Compliance, also referred to as strain energy, quantifies the internal work performed by a structure under an applied load. This section develops the optimization formulation for linear compliance minimization problems discussed in chapter 5.

The compliance equation is derived from the work equation and the equilibrium equation, as given in eqs. (4.17) and (4.18) respectfully.

$$c = \mathbf{f} \cdot \mathbf{u} \tag{4.17}$$

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{f} \tag{4.18}$$

where c is the energy, or compliance, **f** is the force, **u** the displacement, and **K** the stiffness

Rearranging eqs. (4.17) and (4.18) produces a single equation for compliance as:

$$c = \mathbf{u}^T \cdot \mathbf{K} \cdot \mathbf{u} \tag{4.19}$$

Equation (4.19) is used as our objective function for the linear compliance optimization. With this work considering gradient methods, the derivative of eq. (4.19) is needed. With **K** and **u** being functions of our variables, **z**, the derivative of eq. (4.19) may be obtained from the chain rule as:

$$\begin{aligned} c &= \mathbf{u}^T \cdot \mathbf{K} \cdot \mathbf{u} \\ \frac{d}{d\mathbf{z}}c &= \frac{d}{d\mathbf{z}}(\mathbf{u}^T\mathbf{K}\mathbf{u}) \\ &= \mathbf{u}^T\frac{d}{d\mathbf{z}}(\bar{K})\mathbf{u} + 2\mathbf{u}^T\mathbf{K}\frac{d}{d\mathbf{z}}(\mathbf{u}) \end{aligned} \tag{4.20}$$

While obtaining $\frac{d}{d\mathbf{z}}(\bar{K})$ is relatively straight forward, $\frac{d}{d\mathbf{z}}(\mathbf{u})$ has no easily obtainable analytical solution. To work around this, the derivative of the equilibrium equation, eq. (4.18), is used:

$$\begin{aligned}\mathbf{K}\mathbf{u} &= \mathbf{f}\\ \frac{d}{d\mathbf{z}}(\mathbf{K}\mathbf{u}) &= \underbrace{\frac{d}{d\mathbf{z}}(\mathbf{f})}_{=0}\\ \mathbf{K}\frac{d\mathbf{u}}{d\mathbf{z}} &= -\frac{d\mathbf{K}}{d\mathbf{z}}\mathbf{u}\end{aligned} \tag{4.21}$$

Combining eqs. (4.20) and (4.21) yields the derivative of compliance as:

$$\frac{d}{d\mathbf{z}}c = -\mathbf{u}^T\left(\frac{d\mathbf{K}}{d\mathbf{z}}\right)\mathbf{u} \tag{4.22}$$

As to not break flow, the derivatives of $\left(\frac{d\mathbf{K}}{d\mathbf{z}}\right)$ are provided in the following section, section 4.3.

By acquiring the derivative of compliance, we have everything necessary for minimizing the compliance of a morphing beam network. With our objective function given by eq. (4.19), the constraints for volume (eq. (4.1)), width (eq. (4.8)), and position (eq. (4.14)), and the scale factor eq. (4.15), the formal optimization definition used to minimize a morphing beam network is:

$$\begin{aligned}\min_{\mathbf{z}} \quad & c(\mathbf{z}) = \mathbf{u}^T \cdot \mathbf{K} \cdot \mathbf{u}\\ \text{s.t.} \quad & \frac{V(\mathbf{z})}{V_0} \leq V_f\\ & 0 < w_{\min} < \mathbf{w}\\ & \mathbf{n} \in \Omega_v\end{aligned} \tag{4.23}$$

with

$$\mathbf{z}_{new} = \mathbf{z}_{min} \leq \mathbf{z} - \Delta\mathbf{z} \leq \mathbf{z} \times \sqrt{\frac{\mathbf{u}^T\left(\frac{d\mathbf{K}}{d\mathbf{z}}\right)\mathbf{u}}{\Lambda\frac{\partial}{\partial\mathbf{z}}V}} \leq \mathbf{z} + \Delta\mathbf{z} \leq \mathbf{z}_{max}$$

This completes the optimization formulation for minimizing the compliance of a morphing beam network. The next section will provide the Jacobian for **K** considering linear-elastic elements. Section 4.4 will do a similar process outlined in this section for the targeted deformation problem. For those exclusively interested in on compliance minimization, it is recommended to proceed directly to either section 4.6 for a summary of this chapter or to chapter 5 which contains the results of the compliance minimization of a morphing beam network.

4.3 Derivatives for linear-elastic K

From eq. (3.43), the derivative of **K** with respect to **z** can be obtained and will be produced for a single element, with the derivative for the full matrix being obtainable by the usual stiffness creation. Upon inspection of eq. (3.43), the optimization variables **x** and **y** are coupled in the functions, e.g. L, while w remains independent. As such, for brevity ζ will introduced as $\zeta = \{x_1, x_2, y_1, y_2\}$. With this simplification, the derivatives of with respect to w are in eq. (4.24), while the derivatives with respect to ζ are in eq. (4.25).

$$
\begin{aligned}
\frac{\partial k_{1,le}}{\partial w} &= \frac{3\,E\,A\,w\sin(\beta)^2 + E\,d\,L^2\cos(\beta)^2}{L^3}\\
\frac{\partial k_{2,le}}{\partial w} &= \frac{3\,E\,A\,w\cos(\beta)^2 + E\,d\,L^2\sin(\beta)^2}{L^3}\\
\frac{\partial k_{3,le}}{\partial w} &= -\frac{\cos(\beta)\,\sin(\beta)\,\left(E\,d\,L^2 - 3\,E\,A\,w\right)}{L^3}\\
\frac{\partial k_{4,le}}{\partial w} &= \frac{3\,E\,A\,w\,\cos(\beta)}{2\,L^2}\\
\frac{\partial k_{5,le}}{\partial w} &= \frac{3\,E\,A\,w\,\sin(\beta)}{2\,L^2}\\
\frac{\partial k_{6,le}}{\partial w} &= \frac{E\,A\,w}{2\,L}\\
\frac{\partial k_{7,le}}{\partial w} &= \frac{E\,A\,w}{L}
\end{aligned}
\tag{4.24}
$$

$$
\begin{aligned}
\frac{\partial k_{1,le}}{\partial \zeta} &= \frac{2\,A\,E\,\cos(\beta)\,\tfrac{\partial}{\partial \zeta}\cos(\beta)}{L} - \frac{A\,E\cos(\beta)^2\,\tfrac{\partial}{\partial \zeta}L}{L^2}\cdots \\
&+ \frac{24\,E\,I\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}\sin(\beta)}{L^3} - \frac{36\,E\,I\sin(\beta)^2\,\tfrac{\partial}{\partial \zeta}L}{L^4}\cdots \\
\frac{\partial k_{2,le}}{\partial \zeta} &= \frac{2\,A\,E\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}\sin(\beta)}{L} - \frac{A\,E\sin(\beta)^2\,\tfrac{\partial}{\partial \zeta}L}{L^2}\cdots \\
&+ \frac{24\,E\,I\,\cos(\beta)\,\tfrac{\partial}{\partial \zeta}\cos(\beta)}{L^3} - \frac{36\,E\,I\cos(\beta)^2\,\tfrac{\partial}{\partial \zeta}L}{L^4} \\
\frac{\partial k_{3,le}}{\partial \zeta} &= \frac{12\,E\,I\,\cos(\beta)\,\tfrac{\partial}{\partial \zeta}\sin(\beta)}{L^3} + \frac{12\,E\,I\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}\cos(\beta)}{L^3}\cdots \\
&- \frac{36\,E\,I\,\cos(\beta)\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}L}{L^4} + \frac{A\,E\,\cos(\beta)\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}L}{L^2}\cdots \\
&- \frac{A\,E\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}\cos(\beta)}{L} - \frac{A\,E\,\cos(\beta)\,\tfrac{\partial}{\partial \zeta}\sin(\beta)}{L} \\
\frac{\partial k_{4,le}}{\partial \zeta} &= \frac{6\,E\,I\,\tfrac{\partial}{\partial \zeta}\cos(\beta)}{L^2} - \frac{12\,E\,I\,\cos(\beta)\,\tfrac{\partial}{\partial \zeta}L}{L^3} \\
\frac{\partial k_{5,le}}{\partial \zeta} &= \frac{6\,E\,I\,\tfrac{\partial}{\partial \zeta}\sin(\beta)}{L^2} - \frac{12\,E\,I\,\sin(\beta)\,\tfrac{\partial}{\partial \zeta}L}{L^3} \\
\frac{\partial k_{6,le}}{\partial \zeta} &= -\frac{2\,E\,I\,\tfrac{\partial}{\partial \zeta}L}{L^2} \\
\frac{\partial k_{7,le}}{\partial \zeta} &= -\frac{4\,E\,I\,\tfrac{\partial}{\partial \zeta}L}{L^2}
\end{aligned}
\tag{4.25}
$$

with $\zeta \in \{x_1, x_2, y_1, y_2\}$, the derivatives for the $L, \cos(\beta)$ and $\sin(\beta)$ are:

$$\begin{aligned}
\frac{\partial}{\partial x_1} L &= \frac{x_1 - x_2}{L} \\
\frac{\partial}{\partial y_1} L &= \frac{y_1 - y_2}{L} \\
\frac{\partial}{\partial x_1} \cos(\beta) &= \frac{(x_1 - x_2)\,(\Delta x)}{\left((\Delta x)^2 + (\Delta y)^2\right)^{3/2}} - \frac{1}{L} \\
\frac{\partial}{\partial y_1} \cos(\beta) &= \frac{(y_1 - y_2)\,(\Delta x)}{\left((\Delta x)^2 + (\Delta y)^2\right)^{3/2}} \\
\frac{\partial}{\partial x_1} \sin(\beta) &= \frac{(x_1 - x_2)\,(\Delta y)}{\left((\Delta x)^2 + (\Delta y)^2\right)^{3/2}} \\
\frac{\partial}{\partial y_1} \sin(\beta) &= \frac{(y_1 - y_2)\,(\Delta y)}{\left((\Delta x)^2 + (\Delta y)^2\right)^{3/2}} - \frac{1}{L} \\
\frac{\partial}{\partial x_2} L &= -\frac{\partial}{\partial x_1} L \quad \frac{\partial}{\partial y_2} L = -\frac{\partial}{\partial y_1} L \\
\frac{\partial}{\partial x_2} \cos(\beta) &= -\frac{\partial}{\partial x_1} \cos(\beta) \quad \frac{\partial}{\partial y_2} \cos(\beta) = -\frac{\partial}{\partial y_1} \cos(\beta) \\
\frac{\partial}{\partial x_2} \sin(\beta) &= -\frac{\partial}{\partial x_1} \sin(\beta) \quad \frac{\partial}{\partial y_2} \sin(\beta) = -\frac{\partial}{\partial y_1} \sin(\beta)
\end{aligned} \tag{4.26}$$

Equations (4.25) and (4.26) can be combined to obtain the derivatives for each variable in ζ.

4.4 User-defined Nonlinear Mechanical Response

The preceding sections detailed the optimization approach used to minimize compliance in a linear-elastic morphing beam network. This section performs similar operations to formulate optimizations that aim to match a user-defined deformation response.

Consider an optimization task aimed at matching a user-defined target value, denoted as Φ. From chapter 3, the displacement vector, $\mathbf{u}$, of a beam network can be obtained. To obtain the a single scalar value from the structure's deformation vector that matches the user's target node and direction, a sparse vector with a single 1 in the index pertaining the interested node's degree of freedom, called **i**, is multiplied by **u**. The square difference between the network's displacement and the user-defined target is then minimized until the structure matches the user-defined target.

To obtain a target force-displacement curve, a sequence of the network's displacement and user-defined displacement is minimized, providing the following objective function:

$$\text{obj.} = \sum_{t=1}^{N} \cdot \big(\mathbf{i} \cdot \mathbf{u}(f_t) - \mathbf{\Phi}(f_t)\big)^2 \tag{4.27}$$

where $\mathbf{\Phi}$ is a vector containing our set of unique force to user-defined targets, N is the total number of extraction points, and f_t refers to the force applied to the structure for the tth extraction point.

Similar to the compliance minimization problem presented in the previous section, section 4.2, to minimize eq. (4.27), gradient descent is used. Considering a single term of eq. (4.27), its gradient is given by:

$$\frac{d}{d\mathbf{z}}\text{obj.}_{\text{single}} = 2(\mathbf{i} \cdot \mathbf{u} - \mathbf{\Psi}) \cdot \mathbf{i} \cdot \mathbf{J} \tag{4.28}$$

where $\mathbf{J}$ is the Jacobian matrix; $\frac{\partial}{\partial \mathbf{z}}\mathbf{u}$

The derivative of eq. (4.27) necessitates the derivative of **u**. This can be found by taking the derivative of the equilibrium equation, eq. (4.18), beginning with:

$$\begin{aligned}\mathbf{Ku} &= \mathbf{f}\\ \frac{d}{d\mathbf{z}}(\mathbf{Ku}) &= \underbrace{\frac{d}{d\mathbf{z}}(\mathbf{f})}_{=0}\\ \frac{d}{d\mathbf{z}}(\mathbf{K})\mathbf{u} + \mathbf{K}\frac{d}{d\mathbf{z}}(\mathbf{u}) &= 0\end{aligned} \tag{4.29}$$

At this point, it is important to note that the stiffness matrices that were derived in chapter 3 (eqs. (3.41) and (3.42)) are a function of both $\mathbf{z}$ and $\mathbf{u}$, with $\mathbf{u}$ also being a function of $\mathbf{z}$, e.g. $\mathbf{K}(\mathbf{u}(\mathbf{z}), \mathbf{z})$. With this in mind, as a result of the product rule, emergying from the derivative of $\mathbf{K}$ with respect to $\mathbf{z}$ is ${}^{\partial\mathbf{K}}/_{\partial\mathbf{u}}$, a 3D hyper-matrix with each page referring to the derivative of $\mathbf{K}$ with respect to each degree of freedom in $\mathbf{u}$. The operation $\odot$ in this paper refers to taking the dot product of along the depth of ${}^{\partial\mathbf{K}}/_{\partial\mathbf{u}}$. With this noted, continuing from eq. (4.29):

$$\begin{aligned}\frac{d}{d\mathbf{z}}(\mathbf{K})\mathbf{u} + \mathbf{K}\frac{d}{d\mathbf{z}}(\mathbf{u}) &= 0\\ \left(\frac{\partial\mathbf{K}}{\partial\mathbf{u}} \otimes \frac{\partial\mathbf{u}}{\partial\mathbf{z}} + \frac{\partial\mathbf{K}}{\partial\mathbf{z}}\right)\cdot\mathbf{u} + \mathbf{K}\frac{\partial}{\partial\mathbf{z}}\mathbf{u} &= \vdots\\ \frac{\partial\mathbf{K}}{\partial\mathbf{u}} \odot \frac{\partial\mathbf{u}}{\partial\mathbf{z}}\cdot\mathbf{u} + \mathbf{K}\frac{\partial}{\partial\mathbf{z}}\mathbf{u} &= -\frac{\partial\mathbf{K}}{\partial\mathbf{z}}\mathbf{u}\\ \left(\mathbf{u}^T \odot \frac{\partial\mathbf{K}}{\partial\mathbf{u}}\right)\cdot\frac{\partial\mathbf{u}}{\partial\mathbf{z}} + \mathbf{K}\frac{\partial}{\partial\mathbf{z}}\mathbf{u} &= \vdots\\ \frac{\partial\mathbf{u}}{\partial\mathbf{z}} &= -\left(\frac{\partial\mathbf{K}}{\partial\mathbf{u}} \odot \mathbf{u} + \mathbf{K}\right)^{-1}\cdot\frac{\partial\mathbf{K}}{\partial\mathbf{z}}\cdot\mathbf{u}\end{aligned} \tag{4.30}$$

Now that the derivative of displacement is known, with the objective function eq. (4.27), the constraints for volume (eq. (4.1)), width (eq. (4.10)), and position (eq. (4.14)), and the scale factor eq. (4.15), the formal optimization definition used to achieve a target deformation pattern from a user supplied sequence, Ψ with a morphing beam network is:

$$\begin{aligned} \min_{\mathbf{z}} \quad & \text{obj}(\mathbf{z}) = \sum_{g=1}^{N} \cdot \big(\mathbf{i} \cdot \mathbf{u}(f_g) - \mathbf{\Phi}(f_g)\big)^2 \\ \text{s.t.} \quad & \frac{V(\mathbf{z})}{V_0} \leq V_f \\ & 0 < w_{\min} < \mathbf{w}_{cr} < \mathbf{w} \\ & \mathbf{n} \in \Omega_v \end{aligned} \tag{4.31}$$

with

$$\mathbf{z}_{new} = \mathbf{z}_{min} \leq \mathbf{z} - \Delta\mathbf{z} \leq \mathbf{z} \times \sqrt{\frac{-\left(\frac{\partial \mathbf{K}}{\partial \mathbf{u}} \odot \mathbf{u} + \mathbf{K}\right)^{-1} \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{z}} \cdot \mathbf{u}}{\Lambda \frac{\partial}{\partial \mathbf{z}} V}} \leq \mathbf{z} + \Delta\mathbf{z} \leq \mathbf{z}_{max}$$

This completes the optimization definition used to optimize a beam network to achieve a target deformation response, the results of which are displayed in chapter 6. The next section provides the Jacobian for **K** considering nonlinear elements, and the final section in this chapter summarizes the optimization method described in this chapter.

4.5 Derivatives for nonlinear K

Similar to linear-elastic compliance, the derivatives of **K** can be obtained considering nonlinear deformations. For brevity, ζ will again be introduced. However, with ${}^{\partial \mathbf{K}}/_{\partial \mathbf{u}}$ the derivatives of **u** need to be considered, resulting in a larger set, which will be defined as $\zeta_1 = \{x_1, x_2, y_1, y_2, u_{1,x}, u_{2,x}, u_{1,y}, u_{2,y}\}$. Similarly, ζ_2 will be introduced as $\zeta_2 = \{x_1, x_2, y_1, y_2, \mathbf{u}, w\}$. With this simplification, the derivatives **K** with nonlinear elements is contained in eqs. (4.32) to (4.35). As observed in section 4.3 the derivatives for the nodal coordinates are the positive/negative of one another.

This trend continues with nonlinear geometries, and is shared by the rotation and displacements of the nodes as well. For brevity, only the derivatives for the first coordinate will be provided, with the derivative for the second node being the negative of the other of the first coordinate.

$$\begin{aligned}
\frac{\partial k_{1,e}}{\partial w} &= \frac{3\,E\,A\,w\sin(\beta)^2 + E\,d\,L^2\cos(\beta)^2}{L^2\,L_0}\\
\frac{\partial k_{2,e}}{\partial w} &= \frac{3\,E\,A\,w\cos(\beta)^2 + E\,d\,L^2\sin(\beta)^2}{L^2\,L_0}\\
\frac{\partial k_{3,e}}{\partial w} &= -\frac{\cos(\beta)\,\sin(\beta)\,(E\,d\,L^2 - 3\,E\,A\,w)}{L^2\,L_0}\\
\frac{\partial k_{4,e}}{\partial w} &= \frac{3\,E\,A\,w\,\cos(\beta)}{2\,L\,L_0}\\
\frac{\partial k_{5,e}}{\partial w} &= \frac{3\,E\,A\,w\,\sin(\beta)}{2\,L\,L_0}\\
\frac{\partial k_{6,e}}{\partial w} &= \frac{E\,A\,w}{2\,L_0}\\
\frac{\partial k_{7,e}}{\partial w} &= \frac{E\,A\,w}{L_0}
\end{aligned} \tag{4.32}$$

$$
\begin{aligned}
\frac{\partial k_{1,e}}{\partial \zeta_1} &= \frac{2\,A\,E\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L_0} - \frac{A\,E\cos(\beta)^2\,\frac{\partial}{\partial \zeta_1}L_0}{{L_0}^2}\dots \\
&+ \frac{24\,E\,I\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L^2\,L_0} - \frac{12\,E\,I\sin(\beta)^2\,\frac{\partial}{\partial \zeta_1}L_0}{L^2\,{L_0}^2}\dots \\
&- \frac{24\,E\,I\sin(\beta)^2\,\frac{\partial}{\partial \zeta_1}L}{L^3\,L_0} \\
\frac{\partial k_{2,e}}{\partial \zeta_1} &= \frac{2\,A\,E\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L_0} - \frac{A\,E\sin(\beta)^2\,\frac{\partial}{\partial \zeta_1}L_0}{{L_0}^2}\dots \\
&+ \frac{24\,E\,I\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L^2\,L_0} - \frac{12\,E\,I\cos(\beta)^2\,\frac{\partial}{\partial \zeta_1}L_0}{L^2\,{L_0}^2}\dots \\
&- \frac{24\,E\,I\cos(\beta)^2\,\frac{\partial}{\partial \zeta_1}L}{L^3\,L_0} \\
\frac{\partial k_{3,e}}{\partial \zeta_1} &= \frac{12\,E\,I\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L^2\,L_0} + \frac{12\,E\,I\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L^2\,L_0}\dots \\
&- \frac{12\,E\,I\,\cos(\beta)\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L_0}{L^2\,{L_0}^2} - \frac{24\,E\,I\,\cos(\beta)\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L}{L^3\,L_0}\dots \\
&+ \frac{A\,E\,\cos(\beta)\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L_0}{{L_0}^2} - \frac{A\,E\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L_0}\dots \\
&- \frac{A\,E\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L_0} \\
\frac{\partial k_{4,e}}{\partial \zeta_1} &= \frac{6\,E\,I\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L\,L_0} - \frac{6\,E\,I\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}L_0}{L\,{L_0}^2} - \frac{6\,E\,I\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}L}{L^2\,L_0} \\
\frac{\partial k_{5,e}}{\partial \zeta_1} &= \frac{6\,E\,I\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L\,L_0} - \frac{6\,E\,I\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L_0}{L\,{L_0}^2} - \frac{6\,E\,I\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L}{L^2\,L_0} \\
\frac{\partial k_{6,e}}{\partial \zeta_1} &= -\frac{2\,E\,I\,\frac{\partial}{\partial \zeta_1}L_0}{{L_0}^2} \\
\frac{\partial k_{7,e}}{\partial \zeta_1} &= -\frac{4\,E\,I\,\frac{\partial}{\partial \zeta_1}L_0}{{L_0}^2}
\end{aligned}
\tag{4.33}
$$

$$
\begin{aligned}
\frac{\partial k_{1,g}}{\partial \zeta_1} &= \frac{\sin(\beta)^2\,\frac{\partial}{\partial \zeta_2}P}{L} - \frac{P\sin(\beta)^2\,\frac{\partial}{\partial \zeta_1}L}{L^2}\dots \\
&+ \frac{2\,P\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L} + \frac{\cos(\beta)\,\sin(\beta)\,\left(2\,\frac{\partial}{\partial \zeta_2}M_1 + 2\,\frac{\partial}{\partial \zeta_2}M_2\right)}{L^2}\dots \\
&+ \frac{\cos(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L^2}\dots \\
&+ \frac{\sin(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L^2}\dots \\
&- \frac{2\,\cos(\beta)\,\sin(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}L}{L^3} \\
\frac{\partial k_{2,g}}{\partial \zeta_1} &= \frac{\cos(\beta)^2\,\frac{\partial}{\partial \zeta_2}P}{L} + \frac{2\,P\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L}\dots \\
&- \frac{P\cos(\beta)^2\,\frac{\partial}{\partial \zeta_1}L}{L^2} + \frac{\cos(\beta)\,\sin(\beta)\,\left(2\,\frac{\partial}{\partial \zeta_2}M_1 + 2\,\frac{\partial}{\partial \zeta_2}M_2\right)}{L^2}\dots \\
&+ \frac{\cos(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L^2}\dots \\
&+ \frac{\sin(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L^2}\dots \\
&- \frac{2\,\cos(\beta)\,\sin(\beta)\,\left(2\,M_1 + 2\,M_2\right)\,\frac{\partial}{\partial \zeta_1}L}{L^3} \\
\frac{\partial k_{3,g}}{\partial \zeta_1} &= \frac{P\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)}{L} + \frac{P\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta)}{L}\dots \\
&+ \frac{\cos(\beta)\,\sin(\beta)\,\frac{\partial}{\partial \zeta_2}P}{L} - \frac{P\,\cos(\beta)\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}L}{L^2}\dots \\
&+ \frac{\left(\cos(\beta)^2 - \sin(\beta)^2\right)\,\left(\frac{\partial}{\partial \zeta_2}M_1 + \frac{\partial}{\partial \zeta_2}M_2\right)}{L^2}\dots \\
&+ \frac{\left(2\,\cos(\beta)\,\frac{\partial}{\partial \zeta_1}\cos(\beta) - 2\,\sin(\beta)\,\frac{\partial}{\partial \zeta_1}\sin(\beta)\right)\,\left(M_1 + M_2\right)}{L^2}\dots \\
&- \frac{2\,\left(\cos(\beta)^2 - \sin(\beta)^2\right)\,\left(M_1 + M_2\right)\,\frac{\partial}{\partial \zeta_1}L}{L^3}
\end{aligned}
$$

with $\zeta_1 \in \{x_1, x_2, y_1, y_2, u_{1,x}, u_{2,x}, u_{1,y}, u_{2,y}\}$, the derivatives for the rigid body motion of the element are found as:

$$\begin{aligned}
\frac{\partial}{\partial x_1}L &= \frac{\Delta u_x + \Delta x}{L}\\
\frac{\partial}{\partial y_1}L &= \frac{\Delta u_y + \Delta y}{L}\\
\frac{\partial}{\partial u_{x,1}}L &= \frac{\Delta u_x + \Delta x}{L}\\
\frac{\partial}{\partial u_{y,1}}L &= \frac{\Delta u_y + \Delta y}{L}\\
\frac{\partial}{\partial x_1}L_0 &= \frac{\Delta x}{L_0}\\
\frac{\partial}{\partial y_1}L_0 &= \frac{\Delta y}{L_0}\\
\frac{\partial}{\partial x_1}\cos(\beta) &= -\frac{(\Delta u_y + \Delta y)^2}{L^3}\\
\frac{\partial}{\partial y_1}\cos(\beta) &= \frac{(\Delta u_y + \Delta y)\ (\Delta u_x + \Delta x)}{L^3}\\
\frac{\partial}{\partial u_{x,1}}\cos(\beta) &= -\frac{(\Delta u_y + \Delta y)^2}{L^3}\\
\frac{\partial}{\partial u_{y,1}}\cos(\beta) &= \frac{(\Delta u_y + \Delta y)\ (\Delta u_x + \Delta x)}{L^3}\\
\frac{\partial}{\partial x_1}\sin(\beta) &= \frac{(\Delta u_x + \Delta x)\ (\Delta u_y + \Delta y)}{L^3}\\
\frac{\partial}{\partial y_1}\sin(\beta) &= -\frac{(\Delta u_x + \Delta x)^2}{L^3}\\
\frac{\partial}{\partial u_{x,1}}\sin(\beta) &= \frac{(\Delta u_x + \Delta x)\ (\Delta u_y + \Delta y)}{L^3}\\
\frac{\partial}{\partial u_{y,1}}\sin(\beta) &= -\frac{(\Delta u_x + \Delta x)^2}{L^3}
\end{aligned} \tag{4.34}$$

with:

$$\Delta u_x = u_{1,x} - u_{2,x} \quad \Delta u_y = u_{1,y} - u_{2,y}$$

$$\Delta x = x_1 - x_2 \quad \Delta y = y_1 - y_2$$

Finally, the derivatives with respect to the internal load and moments are:

$$\begin{aligned}
\frac{\partial}{\partial w}P &= E\,d - \frac{E\,d\,L_0}{L} \\
\frac{\partial}{\partial u_{x,1}}P &= \frac{E\,d\,w\,L_0\,\left(\Delta u_x + \Delta x\right)}{L^3} \\
\frac{\partial}{\partial u_{y,1}}P &= \frac{E\,d\,w\,L_0\,\left(\Delta u_y + \Delta y\right)}{L^3} \\
\frac{\partial}{\partial w}M_1 &= \frac{E\,A\,w\,\left(2\,\theta_1 + \theta_2 - 3\,\mathrm{atan2}\left(-\Delta u_y, -\Delta u_x + L_0\right)\right)}{2\,L_0} \\
\frac{\partial}{\partial u_{x,1}}M_1 &= \frac{E\,d\,w^3\,\left(\Delta u_y\right)}{2\,\left(\left(\Delta u_y\right)^2 + \left(-\Delta u_x + L_0\right)^2\right)\,L_0} \\
\frac{\partial}{\partial u_{y,1}}M_1 &= \frac{E\,d\,w^3\,\left(-\Delta u_x + L_0\right)}{2\,\left(\left(\Delta u_y\right)^2 + \left(-\Delta u_x + L_0\right)^2\right)\,L_0} \\
\frac{\partial}{\partial \theta_1}M_1 &= \frac{E\,d\,w^3}{3\,L_0} \\
\frac{\partial}{\partial w}M_2 &= \frac{E\,A\,w\,\left(\theta_1 + 2\,\theta_2 - 3\,\mathrm{atan2}\left(-\Delta u_y, -\Delta u_x + L_0\right)\right)}{2\,L_0} \\
\frac{\partial}{\partial u_{x,1}}M_2 &= \frac{E\,d\,w^3\,\left(\Delta u_y\right)}{2\,\left(\left(\Delta u_y\right)^2 + \left(-\Delta u_x + L_0\right)^2\right)\,L_0} \\
\frac{\partial}{\partial u_{y,1}}M_2 &= \frac{E\,d\,w^3\,\left(-\Delta u_x + L_0\right)}{2\,\left(\left(\Delta u_y\right)^2 + \left(-\Delta u_x + L_0\right)^2\right)\,L_0} \\
\frac{\partial}{\partial \theta_1}M_2 &= \frac{E\,d\,w^3}{6\,L_0} \\
\frac{\partial}{\partial x_1}P \;\&\; &\frac{\partial}{\partial y_1}P \;\&\; \frac{\partial}{\partial x_1}M_1 \;\&\; \frac{\partial}{\partial y_1}M_1 \;\&\; \frac{\partial}{\partial x_1}M_2 \;\&\; \frac{\partial}{\partial y_1}M_2 \rightarrow \exists
\end{aligned} \tag{4.35}$$

The derivatives of internal forces with respect to nodal positions, $\frac{\partial}{\partial x_1}P \cdots \frac{\partial}{\partial y_1}M_2$ while straightforward to compute, are incredibly large and difficult to succinctly display. As such they are left to reader as an exercise, with the relation at the end of eq. (4.35) noting their existence.

4.6 Optimization Algorithm

Throughout this chapter, the optimization framework utilized in this study has been developed. Section 4.1 detailed the constraints applied to the three types of optimization variables: $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{w}$. Subsequently, objective functions for both compliance and targeted deformation patterns were defined, with their derivatives presented in sections 4.2 and 4.4 respectfully. This subsection summarizes the chapter by presenting the optimization algorithm formulated in this study, show below in algorithm 1.

Algorithm 1 General Beam Topology Optimization Algorithm

```
Input: $V_f$, $\Omega$, $\Gamma$, $\Delta\mathbf{z}$, $iter_{max}$, tol, objective
Input: Target Def. Only $\mathbf{\Psi}$
Output: $\mathbf{z}$
  Generate grid over the solid domain
  Connect Nodes together to form the beam network                ▷ Chapter 2
  Calculate $V_o = V_f \cdot V$
  while $iter \leq iter_{max}$ and $|c_{iter-1} - c_{iter}| < tol$ do
      Solve the equilibrium equation                              ▷ Chapter 3
      Calculate $\frac{\partial}{\partial \mathbf{z}}\text{obj}$, $\frac{\partial}{\partial \mathbf{z}}V$          ▷ Section 4.2 or section 4.4
      $\Lambda_1 = 0, \Lambda_2 = 1e9, \Lambda = \frac{\Lambda_2 - \Lambda_1}{2}$
      while $\Lambda < tol$ do
          $\mathbf{z}_{new} = \min\Big(\mathbf{z} + \Delta\mathbf{z}, \max\Big(\mathbf{z} - \Delta\mathbf{z}, \mathbf{z} \times \sqrt{\frac{\frac{\partial}{\partial X}\text{obj}}{\Lambda \frac{\partial}{\partial X} V}}\Big)\Big)$
          $\mathbf{z}_{new} = \min\Big(\mathbf{z}_{max}, \max\Big(\mathbf{z}_{min}, \mathbf{z}_{new}\Big)\Big)$
          From $x_{new}, y_{new}$, calculate $L_{new}$
          if $(\mathbf{w}_{new} \times d_e) \cdot \mathbf{L}_{new} - V_o > 0$ then
              $\Lambda_1 = \Lambda$
          else
              $\Lambda_2 = \Lambda$
          end if
          $\Lambda = \frac{\Lambda_2 - \Lambda_1}{2}$
      end while
  end while
```

With the topological formulation constructed in chapter 2, the analysis method in chapter 3, and the optimization method completed in this chapter, the next chapters will present results obtainable by this method. Chapter 5 will showcase the linear-elastic compliance minimization problems, while chapter 6 will demonstrate the targeted deformation response problem.

Chapter 5

Compliance Minimization

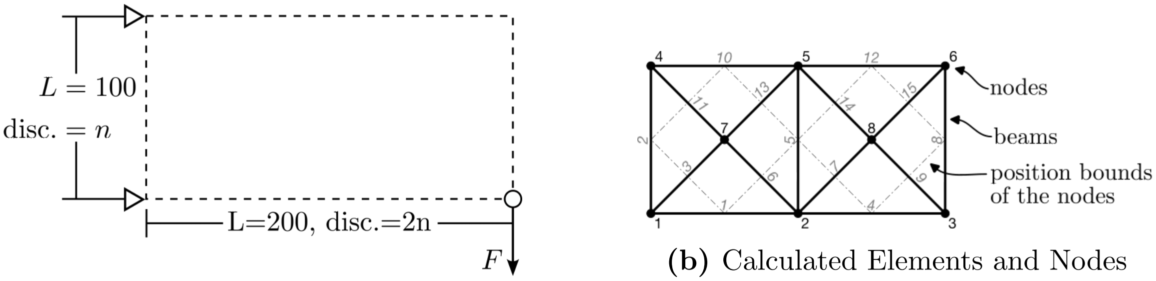
While the method described in the previous chapters forms the basis for a topology optimization method, it is important to first verify its ability to reproduce the results of existing topology optimization methods. This chapter has two primary goals. First, it aims to provide several examples of topologies obtained with the proposed method and compare them to solutions obtained using solid elements. Second, it seeks to demonstrate optimization convergence by increasing the number of nodes used to define the initial domain.

This chapter is divided into five sections. The first four optimize the topologies for four types of problems: a simple cantilever, a bridge, an L-bracket, and a pincer, as discussed in sections 5.1 and 5.3 to 5.5 respectfully, while the last section simply presents additional designs. The cantilever beam section, section 5.1, goes more in-depth to highlight the proposed methods abilities.

To produce the following results, each network was constructed using the previously defined topology of a morphing beam network given in chapter 2, the analysis method detailed inchapter 2, the analysis method detailed in chapter 3, the optimization formulation described in section 4.2 using algorithm 1.

5.1 Cantilever with Point Load

In this section, we demonstrate the application of the proposed topology optimization method across several cantilever beam problems. The first example has its boundary condition shown in fig. 5.1a. Using a cell discretization of 2 × 1 and employing the partially connected domain, the initial nodes, elements, and bounds are illustrated in fig. 5.1b.

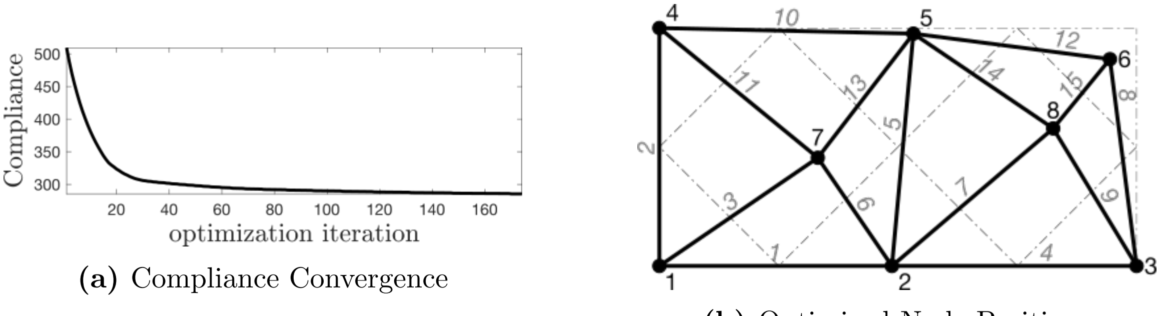


(a) Cantilever BC

(b) Calculated Elements and Nodes

Figure 5.1. Topology creation processes for a cantilever beam

Running the full optimization with $\Delta\mathbf{z} = 0.1$ produces the structure shown in fig. 5.2b, with the compliance at each optimization iteration presented in fig. 5.2a. Scaling the widths to match the optimized values results in the final structure shown in fig. 5.3, where beams with widths set to the minimum are hidden due to their negligible stiffness.



(a) Compliance Convergence

(b) Optimized Node Positions

Figure 5.2. Optimization Process for a cantilever beam

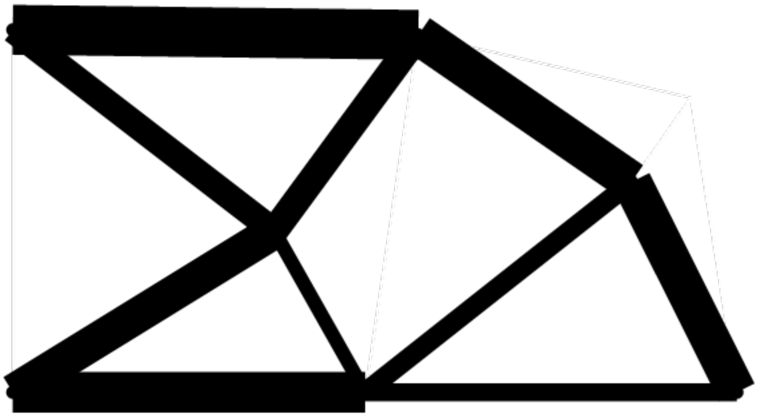


Figure 5.3. Final Plot with proportional beam thicknesses

Using the labels shown in fig. 5.1b, the final node positions and element widths are given in table 5.1. A figure whose beam's widths are scaled to match those values is viewable in fig. 5.3.

Table 5.1. Optimized node locations and element widths

(a) Final Node Positions

position	Node Number							
	1	2	3	4	5	6	7	8
x	0	97.315	200	0	101.053	187.276	74.318	171.872
y	0	0	0	100	100	79.669	48.601	57.687

(b) Optimized Element Widths

width	Element Number														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w	3.17	0	2.56	1.29	0	1.17	1.74	0	2.95	3.49	2.21	0	1.8	3.11	0

Progressively doubling the initial grid density results in domains that resemble their coarser counterparts, ensuring that optimized structures either match the optimized structure found in the coarser domain or, with a finer mesh, achieve structures with lower compliance. This is because finer grids retain the variables of less fine grids, while possessing additional variables, thereby providing a larger design space. Inversely, the design space of coarser grids can be viewed as a subset of the expanded design space available with finer grids. To demonstrate this, several cantilevers were optimized with progressively doubled grid densities, as shown in fig. 5.4, which displays the resulting structures alongside their optimized compliance values. The jagged jagged edges observed in dense meshes result from short elements arranged in a jagged pattern, similar to the checker-boarding effect that occurs with solid elements.

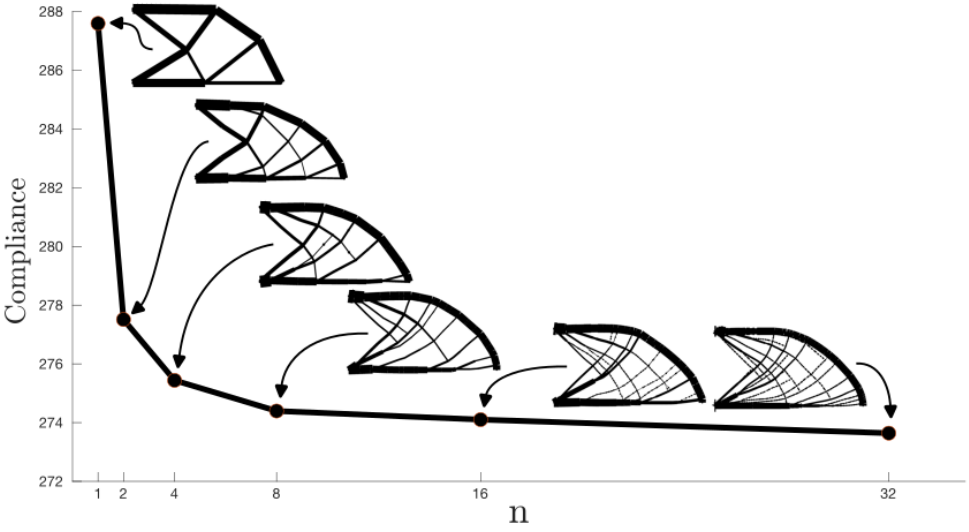
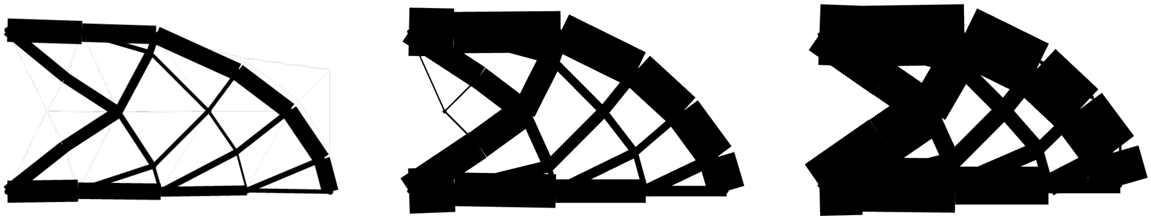


Figure 5.4. Convergence behavior for cantilever beam as n increases

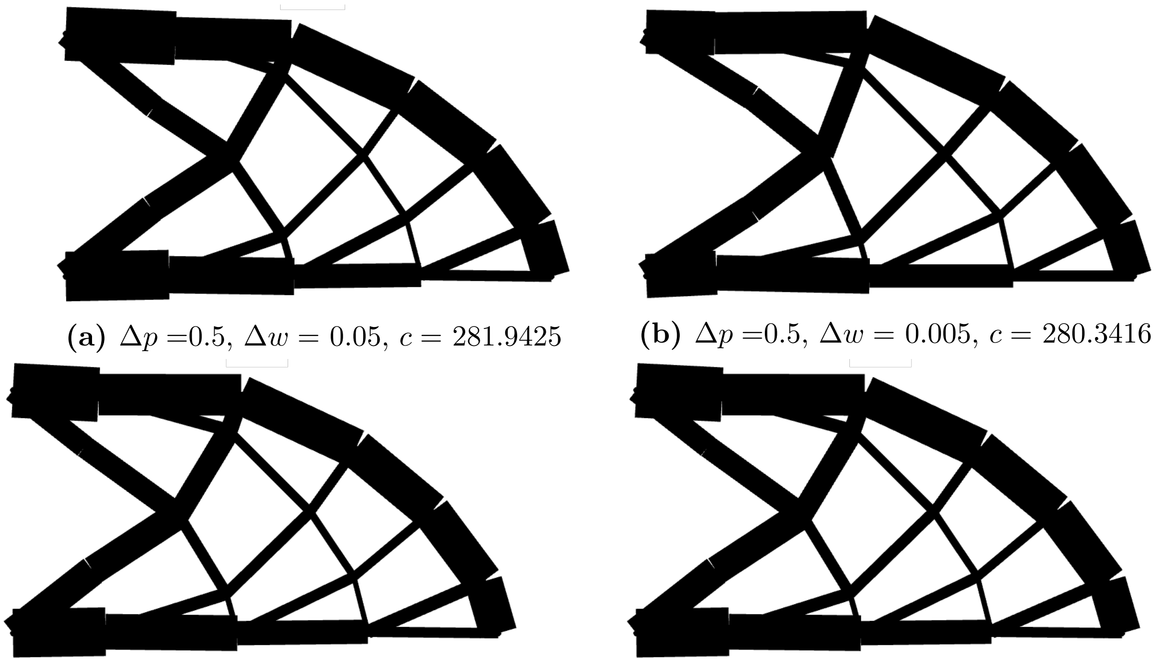
The designs produced by the method can be greatly influenced by the amount of allowable material within the domain, which is controlled by the volume fraction. To show the effects that changing the volume fraction has on the final optimized structure, fig. 5.5 shows the results of three different cantilever beams produced by $V_f \in \{0.10, 0.30, 0.50\}$. With less material available for optimization, the material that is present is consolidated into a limited number of features. Optimizations with a larger volume fraction allow for features to emerge that strengthen the structure as whole, but whose individual effect would be negligible if the structure's material would be scaled to a smaller volume fraction.



(a) $V_f = 0.10$ **(b)** $V_f = 0.30$ **(c)** $V_f = 0.5$

Figure 5.5. Volume fraction effects on the cantilever beam problem

Figure 5.6 shows the effect of changing Δp and Δw has on the optimization results. Running four cantilever beams with $\Delta p \in \{0.05, 0.001\}$ and $\Delta w \in \{0.05, 0.005\}$ returns the structures shown in fig. 5.6.



(a) Δp =0.5, $\Delta w = 0.05$, $c = 281.9425$ **(b)** Δp =0.5, $\Delta w = 0.005$, $c = 280.3416$

(c) Δp =0.05, $\Delta w = 0.05$, $c = 276.3385$ **(d)** Δp =0.05, $\Delta w = 0.005$, $c = 275.6214$

Figure 5.6. Effect $\Delta \mathbf{z}$ has on the final cantilever beam topology

5.2 Computational Cost Comparison to Solid Elements

Switching from solid to beam elements has a measurable impact on computational cost in both structure simulation and optimization. Consider a single solid element and a single beam element shown in fig. 5.7a. The solid element has two degrees of freedom per node: displacement in each principal direction. In contrast, the nodes in a beam element possess these two degrees of freedom plus an additional rotational degree of freedom. In optimization, both element types have a variable dedicated to material distribution—Young's Modulus for solid elements and width for beam elements. However, beam elements also have an additional variable type for node positioning.

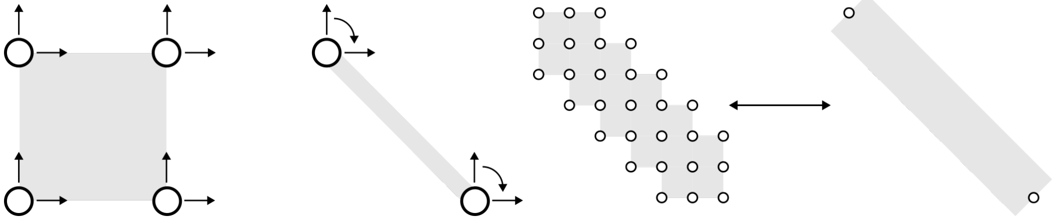

(a) Comparison of Single elements **(b)** Beam Approximation

Figure 5.7. Comparison and approximation of solid and beam elements

Consider a grid of $n \times n$ nodes. These nodes can be interconnected to form either square solid elements or a beam network using the unit-cell described in fig. 2.5. An example of each is shown in section 5.2, where the left grid displays a mesh of solid elements and the right grid displays a mesh of beam elements. Using this grid as a benchmark, we can calculate the number of nodes, elements, degrees of freedom present in finite element analysis, and optimization variables, as summarized in table 5.2.

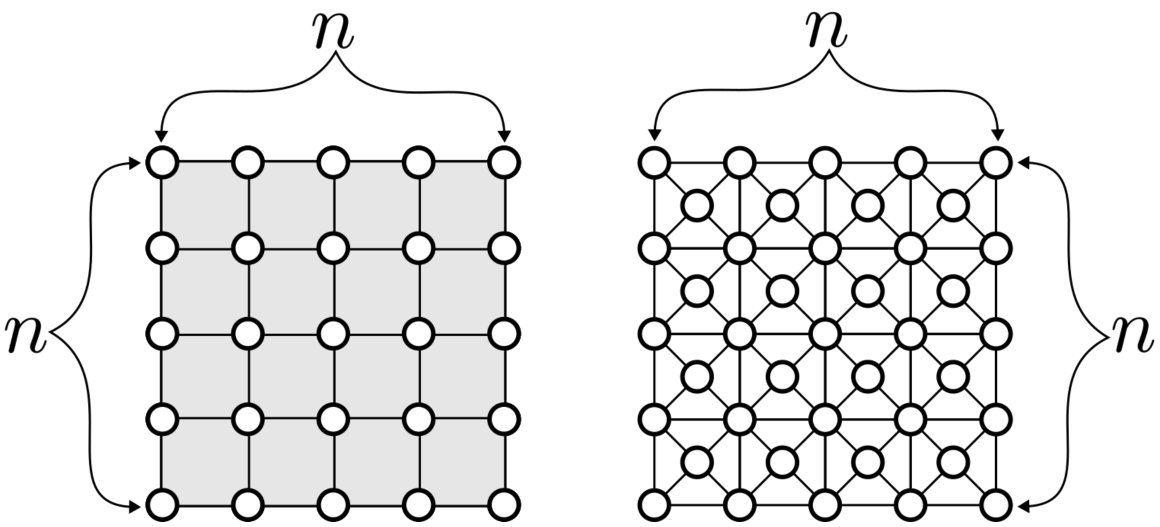


Figure 5.8. $n \times n$ domain of solid and beam elements

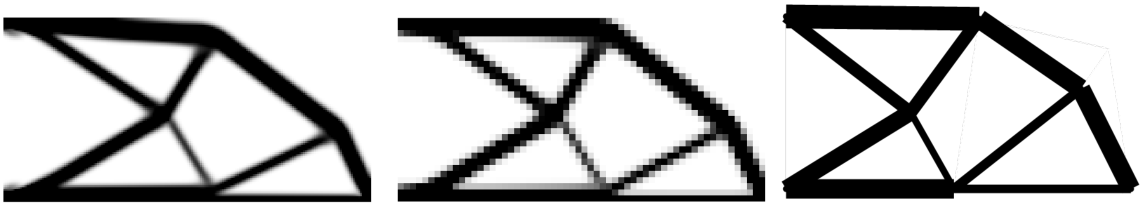
Table 5.2. Numerical comparison between beam and solid elements in a square grid

$n \times n$	Beam elements	Solid elements
num. Nodes	$n^2 - (n-1)^2$	n^2
num. elements	$2(3n-2)(n-1)$	$(n-1)^2$
FEA D.o.F.	$3n^2 - 3(n-1)^2$	$2(n-1)^2$
Opt. Variables	$2n^2 - 2(n-1)^2 + 2(3n-2)(n-1)$	$(n-1)^2$

Table 5.2 shows that solid elements out perform beam elements, making beam elements only suitable if the initial grid may be made considerably coarser, which the remainder of this subsection will show is the case. The reason why the grid consisting of beam elements may be made more course is due to multiple solid elements needed to represent thin features, while the same feature can be represented by a single beam element as shown in fig. 5.7b.

Returning to the cantilever previously discussed in fig. 5.1a and optimizing with solid elements on a grid of 100 × 200 returns fig. 5.9a. The comparison of computational between both beam and solid elements displayed in table 5.3. As expected, beam elements vastly outperform solid elements due to their lower element count. However, the quality of results produced with solid elements correlates with the initial grid density, introducing some subjectivity into determining an acceptable grid size. Reducing the grid size of solid elements to 30×60 returns a very coarse result shown in fig. 5.9b and whose computational complexity is also shown in table 5.3. The smooth edges observed in fig. 5.9a are replaced by blocky edges in fig. 5.9b due to the coarser grid, raising concerns about the acceptability of the result. Even if one finds results from a coarser grid acceptable, the computational cost remains several orders of magnitude higher compared to using beam elements.

Moving forward, this paper prioritizes achieving smooth edges over blocky ones. While this criterion is subjective and favors the proposed method, solid elements still do not match beam elements in terms of computational efficiency and performance.



(a) S.E. result on an initial grid of 100 × 200

(b) S.E. result on an initial grid of 30 × 60

(c) B.E. result on an initial grid of 1 × 2

Figure 5.9. Comparison between optimized cantilevers between element types

Table 5.3. Complexity Comparison between beam and solid elements

	Beam elements	Solid elements (fine)	Solid elements (coarse)
Grid Size (Ele.)	1×2	100×200	30×60
num. Nodes	8	$20,301$	$1,891$
num. elements	15	$20,000$	$1,800$
FEA D.o.F.	24	$61,812$	$5,952$
Opt. Variables	31	$20,000$	$1,800$

Comparing the optimized compliance values between the proposed method using beam elements and existing solid element methods is difficult due to differing approaches in material distribution. In the proposed method, beam element widths are calculated at their midpoints, allowing material to extend slightly beyond domain boundaries. This results in beam elements having a larger effective perimeter compared to solid elements, where material must fully reside within the domain. Since a structure's stiffness primarily relies on material near its perimeter, optimized structures with beam elements often exhibit lower compliance than those produced with solid elements. Scaling the optimized beam structures to precisely fit within the domain reduces material volume, thereby increasing compliance but also altering the structure's boundary conditions, influencing compliance results. To avoid any misleading comparisons, compliance values for solid elements are not provided.

The performance trends observed between beam and solid elements in chapter 5 are consistent across the other examples discussed. For brevity, a detailed comparison between element types is not shown for the rest of the problems in the work problems. For the rest of chapter 5, the grid density of the beam based Topology Optimization is chosen to reproduce the solid element results.

5.3 Bridge with Point Load

Moving beyond cantilevers, another commonly explored topology optimization problem is a bridge with a point load, whose boundary conditions are shown in fig. 5.10.

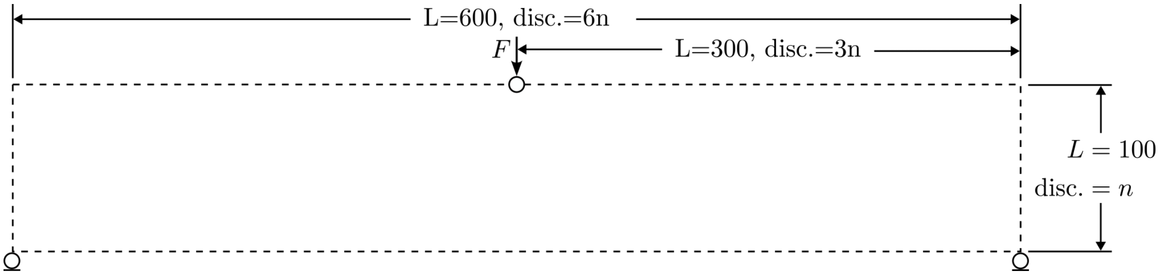
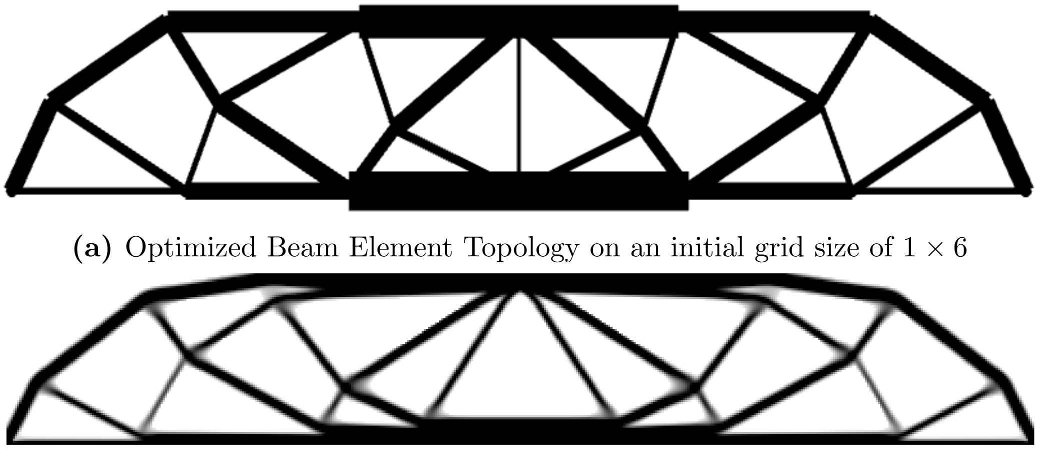


Figure 5.10. Bridge BC

Using algorithm 1 with $n = 1$, V.F. $= 0.3$, $(\Delta p, \Delta w) = 0.001{,}0.1$ produces fig. 5.11a with its Solid Element counterpart presented in fig. 5.11b. A comparison between the computational cost may be seen in table 5.4



(a) Optimized Beam Element Topology on an initial grid size of 1 × 6

(b) Optimized Solid Element Bridge on an initial grid size of 100 × 600

Figure 5.11. Topology Optimized Bridge utilizing Beam And Solid Elements

Table 5.4. Complexity Comparison of Topology Optimizing a Bridge Domain

	Beam Elements	Solid Elements
Grid Size (Ele.)	1×6	100×600
num. Nodes	20	$60,701$
num. Elements	43	$60,000$
FEA D.o.F.	60	$184,212$
Opt. Variables	83	$60,000$

To analyze the optimization performance across multiple instances, multiple bridges were optimized on the range $n \in [0, 5]$, producing the convergence curve in fig. 5.12. As n increases, the number of thick abutments initially rises which then get supported by thinner ones as n continues to increase.

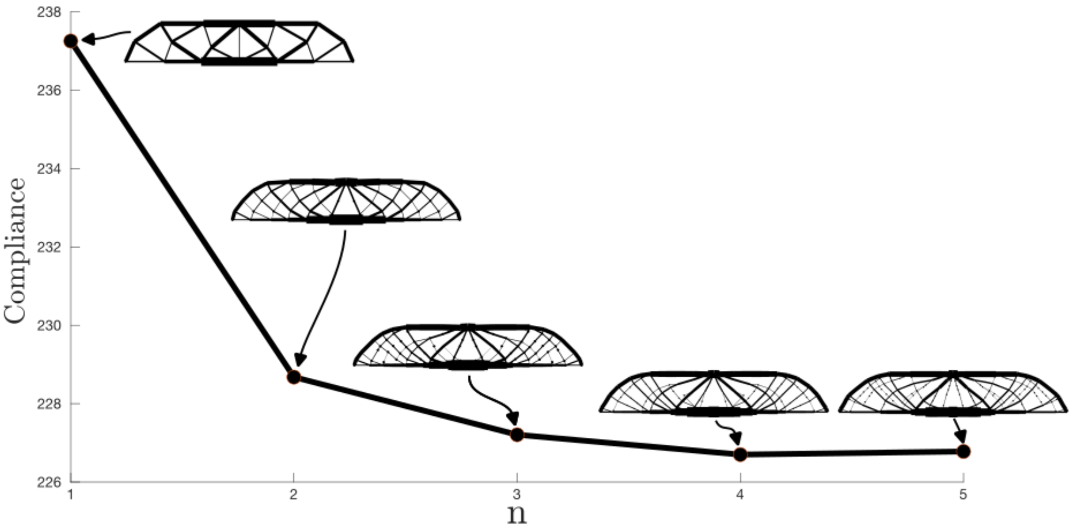
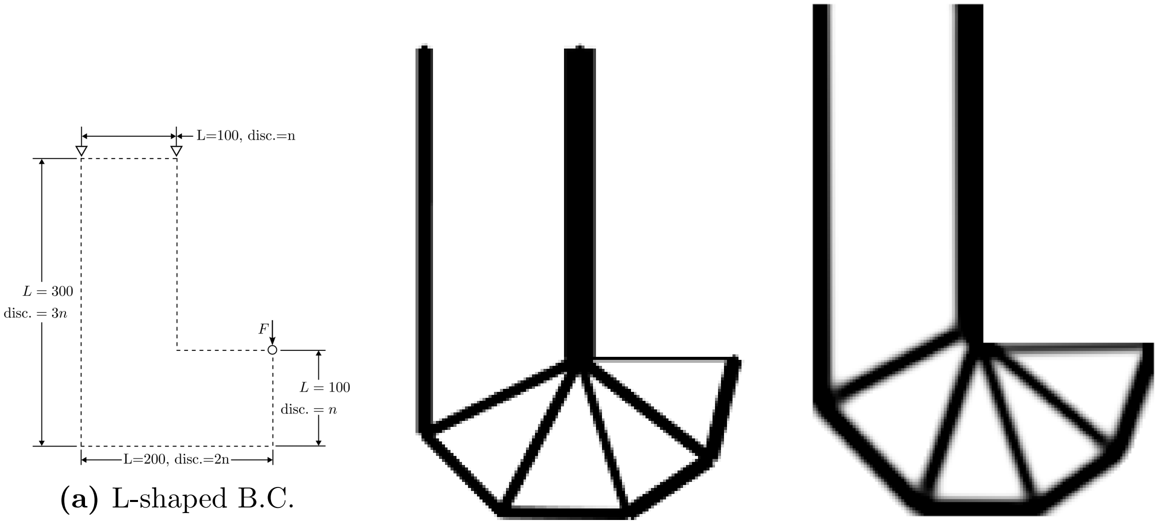


Figure 5.12. Increasing n shows the Bridge converges around 4 to 5 elements

5.4 L-Bracket with Point Load

To show an example with a non-rectangular Ω, a large Γ_s is inserted to form Ω into an "L" shape and whose boundary conditions are shown in fig. 5.13a. Nodes that would move onto into Γ_s during optimization are instead clipped to the nearest point in their path on Γ_o as described by eq. (4.11). Using algorithm 1 with $n = 1$, V.F. = 0.3, $(\Delta p, \Delta w)$ = (0.001,0.1) may be seen in fig. 5.13b with its Solid Element counterpart shown in fig. 5.13c. Table 5.5 shows the complexity comparison between the two structures.



(a) L-shaped B.C.

(b) Beam Topology Optimized L-Bracket

(c) Solid Element Topology Optimized L-Bracket

Figure 5.13. L-shaped Bracket Results

Table 5.5. Complexity Comparison of optimized L-Bracket using Beam and Solid Elements

	Beam Elements	Solid Elements
Grid Size (Ele.)	2×3	120×180
num. Nodes	18	$21,901$
num. Elements	41	$21,600$
FEA D.o.F.	54	$66,612$
Opt. Variables	77	$21,600$

Running multiple L-Brackets on the range $n \in [2 : 2 : 8]$ returns the convergence curve shown in fig. 5.14. It can be observed that as the number of discretizations increases, the bottom dome progressively becomes smoother, and the internal supports are more evenly spread out.

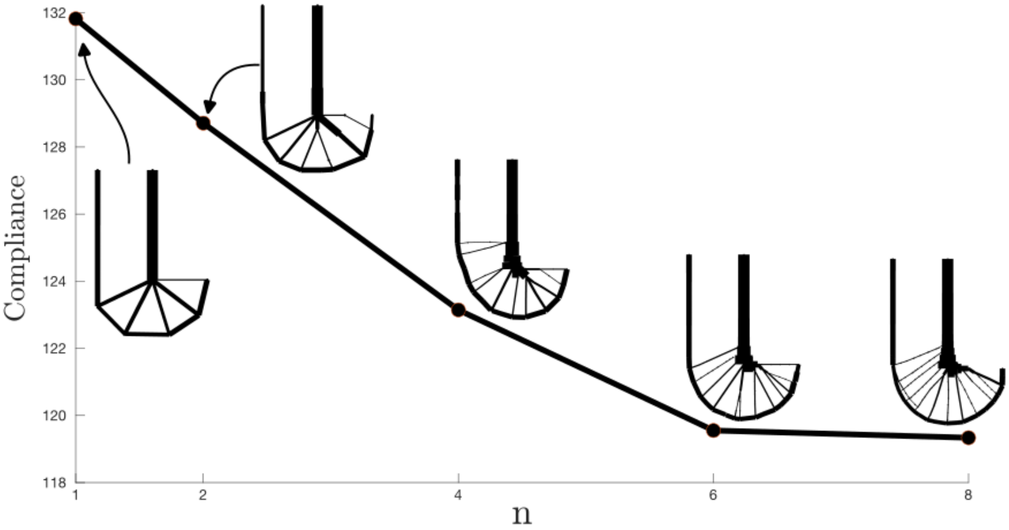
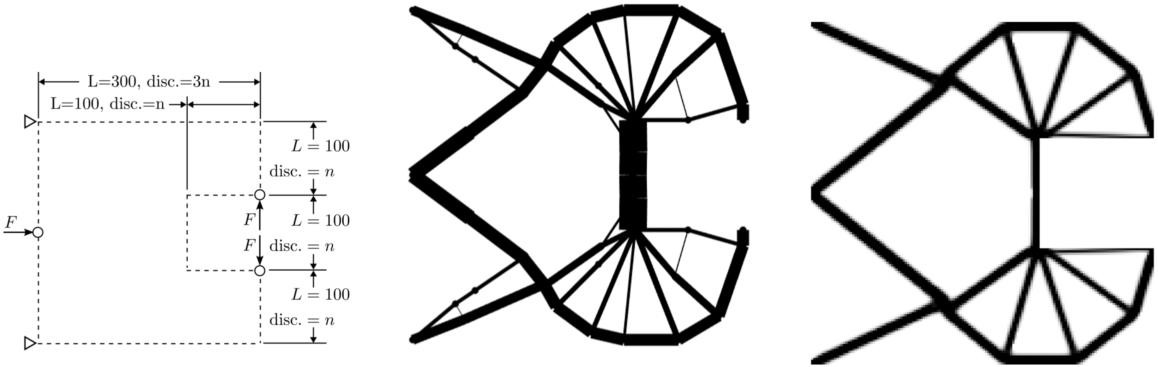


Figure 5.14. Increasing n shows the L-Bracket converges around $n = 6 - 8$ elements

5.5 Linear Pincer Mechanism

The final problem that is compared against solid elements increases complexity and produces a compliant mechanism, whose the boundary conditions are shown in fig. 5.15a. The structures produced by fig. 5.15a takes a horizontal input force, and translates that force to a vertical pinching motion on the 'jaw's of the structure found on the side opposite of the input force. Using algorithm 1 with $n = 1$, V.F. $= 0.2$, $(\Delta p, \Delta w) = 0.001{,}0.1$ produce the compliant mechanisms shown in fig. 5.15b followed by a comparison of their complexity in table 5.6.



(a) Pincer B.C.

(b) Obtained with beam elements (initial grid (6 × 6)

(c) Obtained with solid elements (150 × 150)

Figure 5.15. Pincer results

Table 5.6. Complexity Comparison of optimized Pincer using Beam and solid elements

	Beam elements	Solid elements
Grid Size (Ele.)	6×6	150×150
num. Nodes	85	$22,801$
num. Elements	228	$22,500$
FEA D.o.F.	255	$69,312$
Opt. Variables	398	$22,500$

Running multiple clamp optimizations on the range $n \in [0, 5]$ returns the convergence curve shown in fig. 5.16. As n increases, dome above the jaw progressively comes smoother while the internal supports become more even-spread.

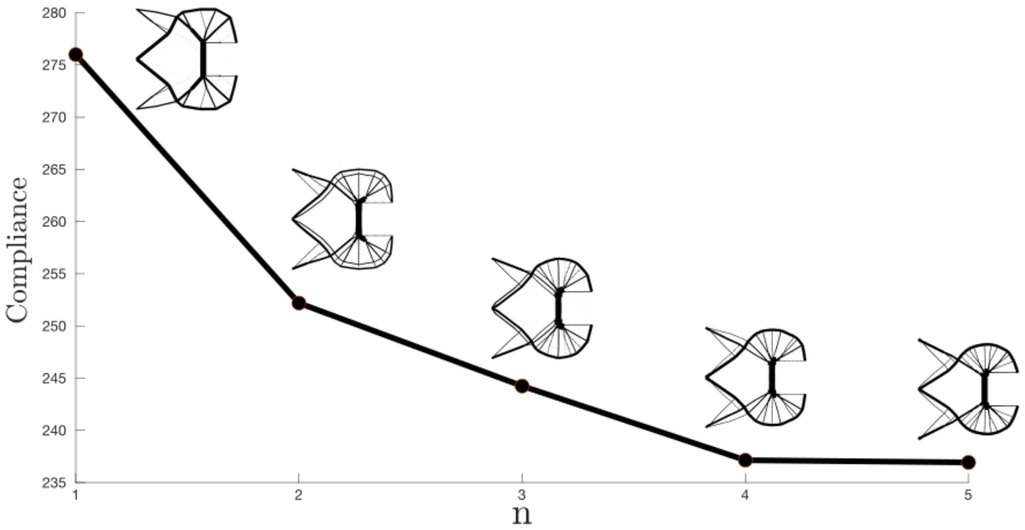
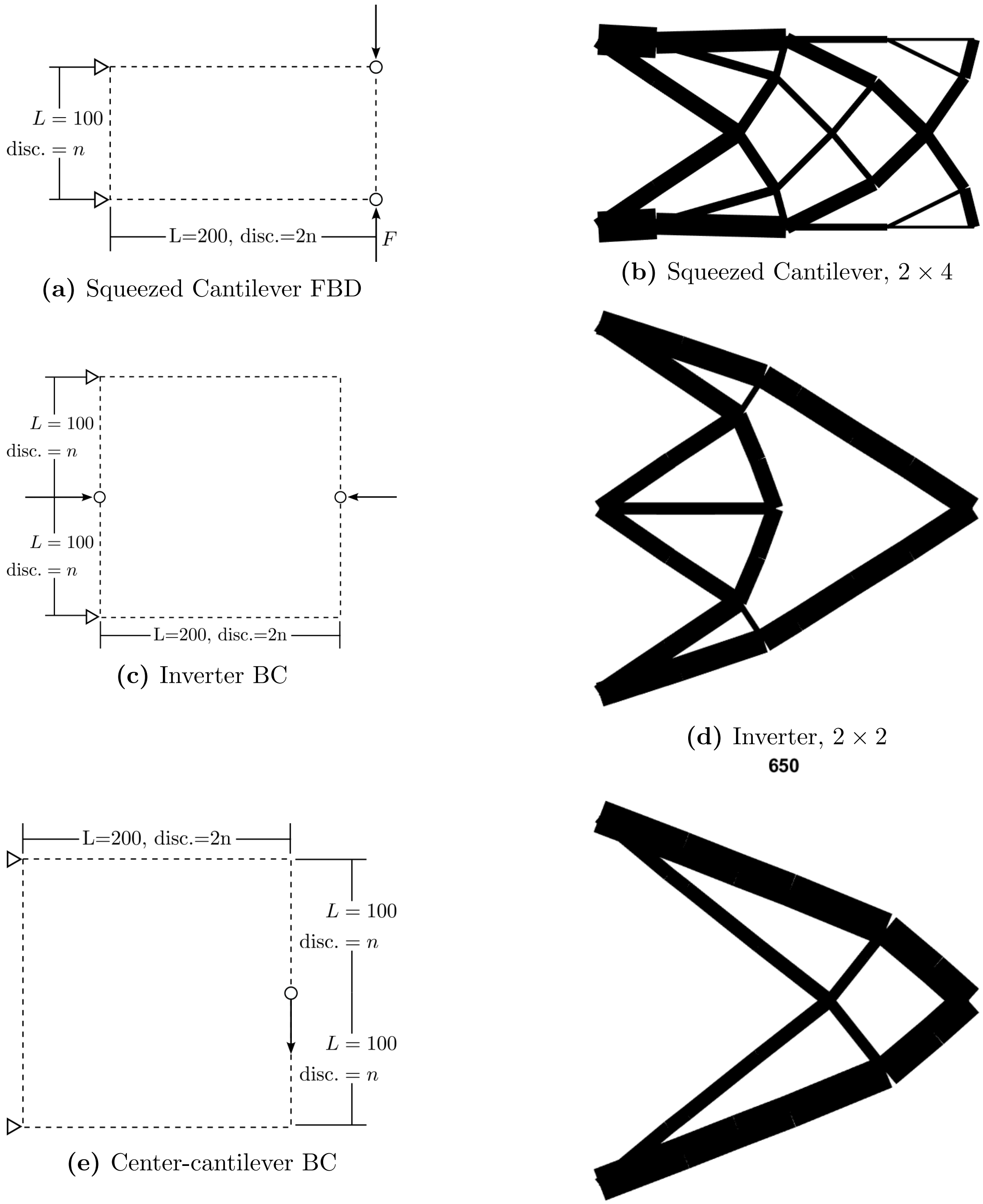


Figure 5.16. Increasing n shows the Pincer converges around $n = 4$ elements

5.6 Additional Examples

The previous sections compared structures obtained by the proposed method against those obtained from solid elements. This section aims to showcase additional compliance minimization problems to further highlight the method's efficacy. The examples in this chapter are presented with the boundary conditions for each problem positioned to the left or directly above the resulting optimized topologies. A comparison between the proposed method and solid elements is not provided.



(a) Squeezed Cantilever FBD

(b) Squeezed Cantilever, 2 × 4

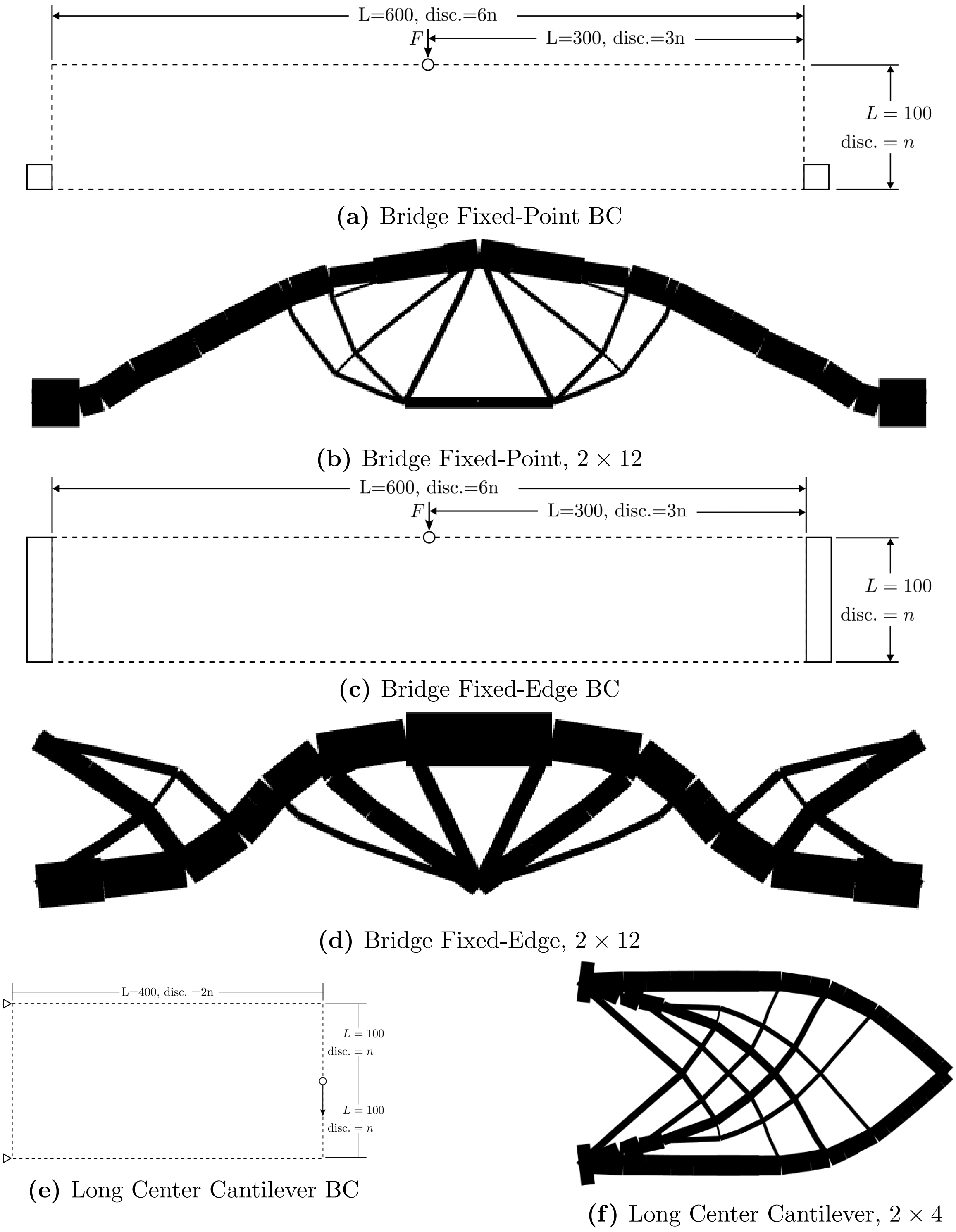
(c) Inverter BC

(d) Inverter, 2 × 2

(e) Center-cantilever BC

(f) Center-cantilever, 2 × 2

Figure 5.17. First set of additional compliance minimization problems



(a) Bridge Fixed-Point BC

(b) Bridge Fixed-Point, 2 × 12

(c) Bridge Fixed-Edge BC

(d) Bridge Fixed-Edge, 2 × 12

(e) Long Center Cantilever BC

(f) Long Center Cantilever, 2 × 4

Figure 5.18. Second set of additional compliance minimization problems

This section has shown the ability for a morphing beam network to produce quality topology optimization results by using a morphing beam network. Constructed with beam elements, the method achieves competitive results comparable to traditional solid element approaches while offering significant computational advantages. The examples illustrated throughout this work, from simple cantilevers to more complex structures like bridges and compliant mechanisms, underscore the method's versatility and fidelity in producing optimized structures. The next chapter seeks to optimize structures that posses a user-defined loading response.

Chapter 6

User-defined Nonlinear Mechanical Response

With the previous chapter showcasing the ability of a morphing beam network to achieve the same result fidelity as solid elements, this chapter aims to extend the previously described method to consider nonlinear geometry in analysis, using the approach outlined in chapter 3.

Extending compliance minimization to include nonlinear geometry is straightforward; therefore, this work focuses on producing structures whose mechanical response to an applied load matches a user-defined response. Specifically, this paper aims to produce structures whose force-displacement curves align with a user-defined response. A moment-displacement curve optimization is also included to further showcase the method's ability. Unless otherwise noted, the vertical displacement of the load node in each problem is optimized, with the target displacement curves provided as a function of displacement for a given force. Five points along this function are selected as targets used in eq. (4.27).

For clarity, functions will define the displacement as a function of the loading ratio, λ, as defined in section 3.4. This is the ratio of the maximum load, ranging from 0, unloaded, to 1, fully loaded. The resulting force-displacement curves are rotated so that the unloaded force-displacement, $(0, 0)$, is always located at the bottom-left corner, and the force-displacement relation for the maximum load is in the top-right corner.

To generalize the optimized topologies obtained by the method, the loading and displacements are rearranged into non-dimensional forms. These parameters are defined as follows:

$$\begin{aligned} F^* &= \frac{FL_c^2}{EI_i} \\ M^* &= \frac{ML_c}{EI_i} \\ U^* &= \frac{U}{L_c} \end{aligned} \tag{6.1}$$

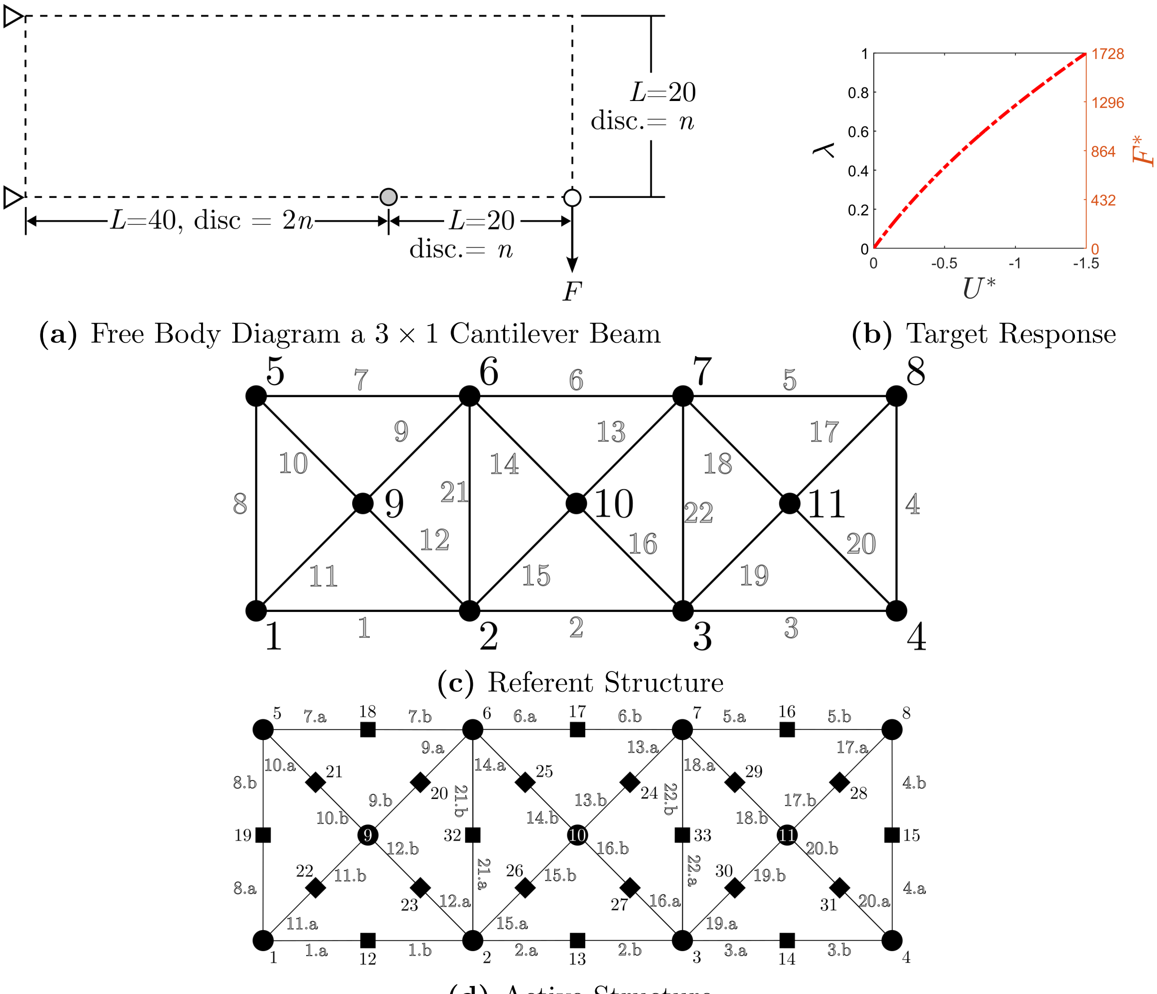
where L_c is the characteristic length, or the maximum length in the original structure, I_i is the original moment of inertia of an element in the initial network.

This chapter will investigate the same topologies explored in the linear cause: a simple cantilever, a bridge, an L-Bracket, and a Pincer, showcased in sections 6.1 to 6.4 respectfully. Similar to the linear case, section 6.1 is much more detailed than its following sections.

6.1 Nonlinear Cantilever with Point Load

The objective of the problems discussed in this and the following sections are to produce structures that match a user-defined force-displacement relation. Specifically, this section investigates a sequence of cantilever beams.

The user-supplied target curve is defined by $U^*(\lambda) = -\frac{1}{L_c}(\lambda^2 + 0.5\lambda)$, as shown in fig. 6.1b which also shows the relation between λ and $\mathbf{F}^*$. To match the target curve, displacement values the structure are extracted along loading ratios of $\lambda = \left\{\frac{n}{5} \mid n = 0, 1, \ldots, 5\right\}$. Based on the boundary conditions (fig. 6.1a), the reference structure used in optimization is generated and shown in fig. 6.1c, with its active counterpart produced by a single discretization shown in fig. 6.1d.



(a) Free Body Diagram a 3 × 1 Cantilever Beam

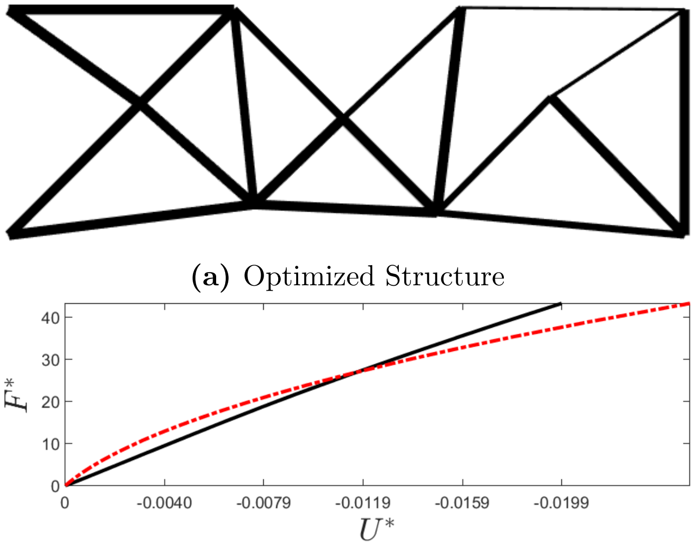
(b) Target Response

(c) Referent Structure

(d) Active Structure

Figure 6.1. Topology creation process for a nonlinear cantilever beam

Using the algorithm 1, the optimized structure to is shown in fig. 6.2a whose mechanical response is shown alongside the target curve in fig. 6.2b.



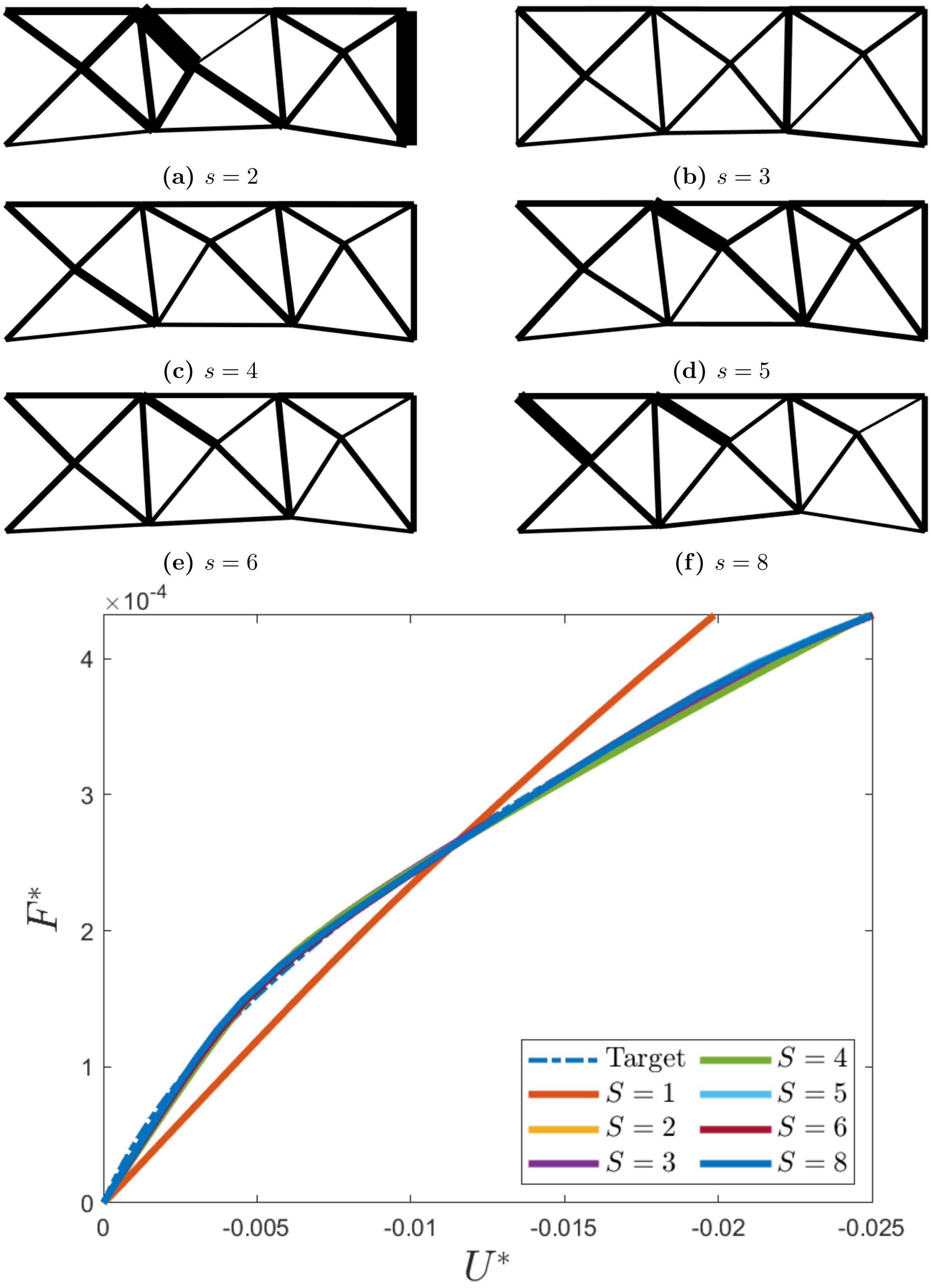
(a) Optimized Structure

(b) Opt. Response

Figure 6.2. Optimized topology for the cantilever beam described in fig. 6.1

The optimized structure's mechanical response exhibits negligible nonlinear behavior and is therefore unable to align with the user-defined nonlinear target curve with only a single subdivision. Since the number of discretizations between the active and referent structures plays a significant role in capturing the nonlinear behavior of the structure, increasing the number of subdivisions allows for a more accurate prediction and optimization of nonlinear behavior. Figure 6.3 shows structures and their force-displacement relations obtained by repeating the previous optimization of the cantilever, but changing the number of subdivisions to be $\{1 \ldots 6, 8\}$. Structures with two or more subdivisions can accurately model the nonlinear behavior within the beams and produce structures that achieve the target curve.

With a low number of subdivisions, structures cannot properly calculate the internal forces generated within their elements, making them very sensitive to any parameter changes. This sensitivity propagates into the derivatives, causing the structures to converge to local minima. This sensitivity, or the inability to accurately calculate the nonlinear response, is why the structure shown in fig. 6.2a was unable to match the target curve. In contrast, with higher discretizations the structures are both less sensitive to variable changes and can more accurately predict nonlinear behavior, resulting in a smoother gradient and nearly identical results.



(a) $s = 2$

(b) $s = 3$

(c) $s = 4$

(d) $s = 5$

(e) $s = 6$

(f) $s = 8$

(g) Resulting force-displacement curves for each optimized structure

Figure 6.3. Sequence of cantilevers with increasing element subdivisions

To select the number of subdivisions for future optimizations, the difference between each structure's force-displacement curve and the given target, $U^* = \frac{1}{L_c}(\lambda^2 + 0.5\lambda)$, is shown in fig. 6.4. As predicted in eq. (3.42), during the initial loads, the internal forces generated within the beam are relatively small, resulting in nearly linear behavior. Consequently, each structure under-predicts the initial loading responses. However, as the applied load increases, so does the nonlinear behavior, allowing the structure to adjust its stiffness to better match the target curve. To select the number of subdivisions for future examples, the percent difference between the structure's response and the given target curve was calculated. Starting from $s = 6$, the percent error was less than 2%. As such, the following examples presented in this chapter will create active structures with six subdivisions.

To further showcase the proposed method, fig. 6.5 shows three more examples of cantilevers matching a user-defined target. Each cantilever was constructed from the referent structure shown in fig. 6.1c with 6 subdivisons, and 5 values were selected along their given curves. Of note is the structure shown in fig. 6.5e whose target curve is non polynomial.

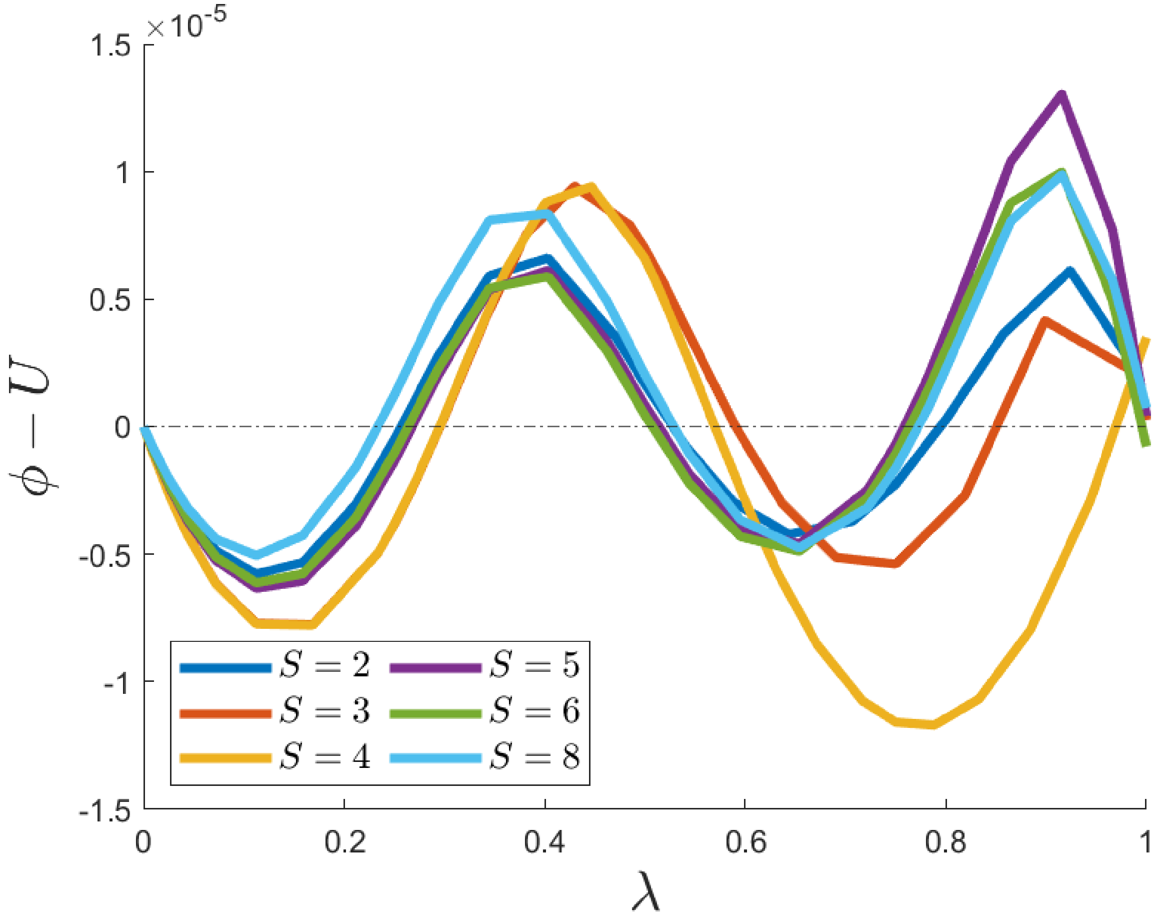
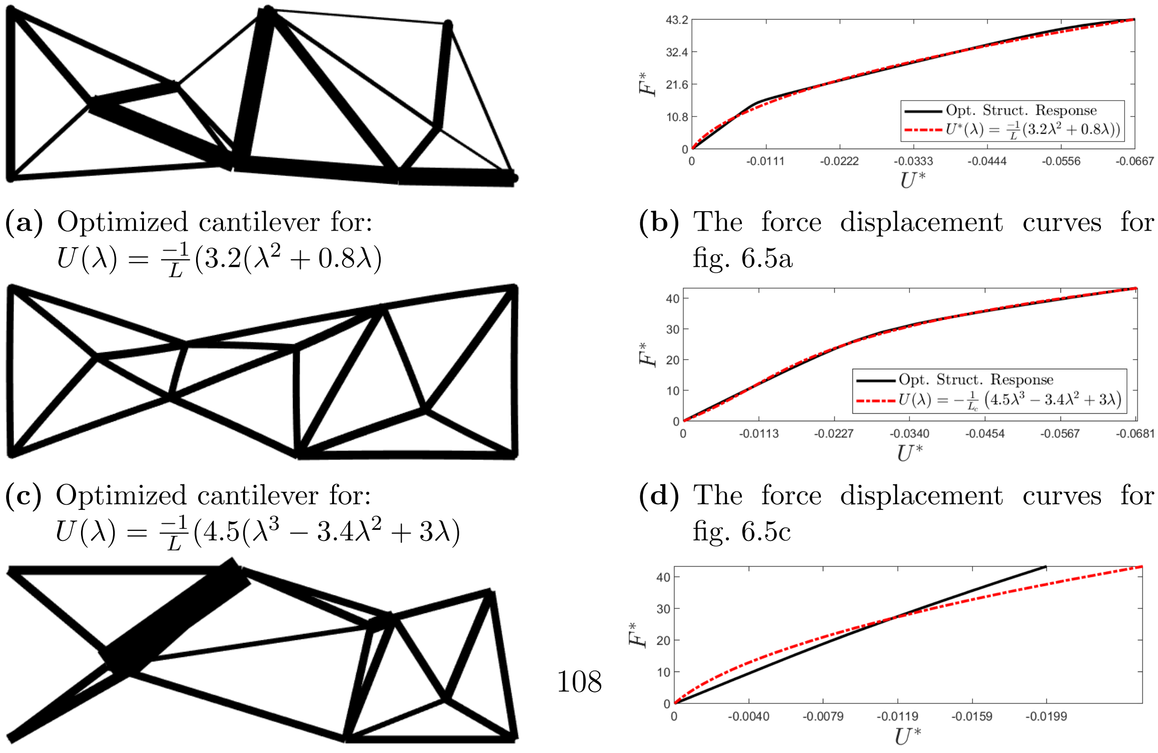


Figure 6.4. Error difference between cantilevers with various subdivisions



(a) Optimized cantilever for: $U(\lambda) = \frac{-1}{L}(3.2(\lambda^2 + 0.8\lambda)$

(b) The force displacement curves for fig. 6.5a

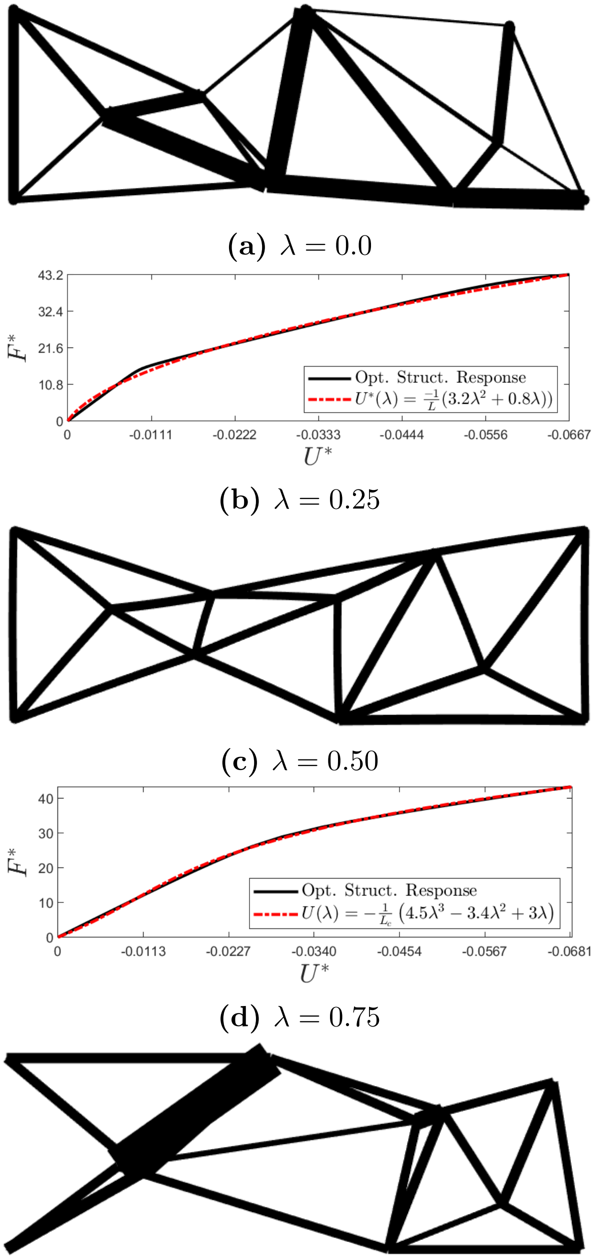
(c) Optimized cantilever for: $U(\lambda) = \frac{-1}{L}(4.5(\lambda^3 - 3.4\lambda^2 + 3\lambda)$

(d) The force displacement curves for fig. 6.5c

(e) Optimized cantilever for: $U(\lambda) = \frac{2}{3}\left(\lambda^2 - \cos(\pi\lambda)\right) + \frac{2}{3}$

(f) The force displacement curves for fig. 6.5e

The motion of the structure depicted in fig. 6.5c in response to the applied load is displayed in fig. 6.6, with λ referring to the load ratio, or the percent of the maximum applied load. The other cantilevers obtained follow similar loading behaviors and are not shown.



(a) $\lambda = 0.0$

(b) $\lambda = 0.25$

(c) $\lambda = 0.50$

(d) $\lambda = 0.75$

(e) $\lambda = 1.00$

Figure 6.6. Motion of the cantilever shown in fig. 6.5c

Finally, NlFrame2D [29] is an open source finite element program which is used to validate the finite element method and the resulting mechanical response obtained by the developed. NlFrame2D was selected as the developed tool was constructed following its formulation. Other finite element tools such as Abaqus use different formulations and solution methods, resulting a difference between the final results. Using the structure shown in fig. 6.5a as an example, it was recreated in NlFrame2D and a comparison between the resulting force-displacement curves between both tools may be viewed in fig. 6.7.

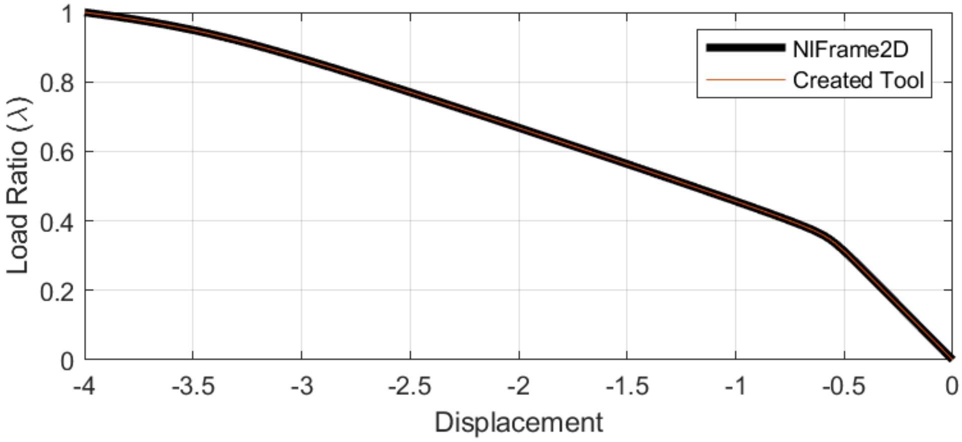
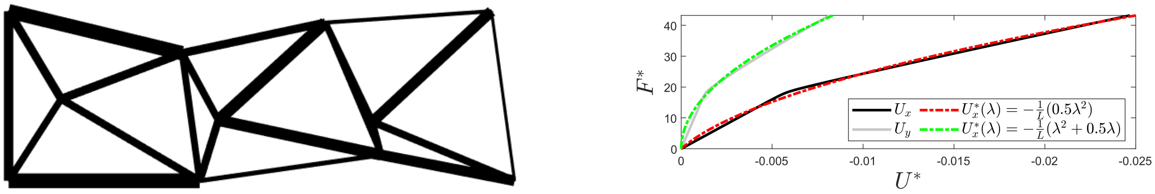


Figure 6.7. NLFrame2D validation for fig. 6.5a

The displacement field obtained from finite elements, $\mathbf{u}$, contains the motion of each node in the active structure, allowing for a node to be optimized in multiple directions. This was done using the same cantilever design, and whose resulting topology and force-displacement curve for both nodes are shown in fig. 6.8

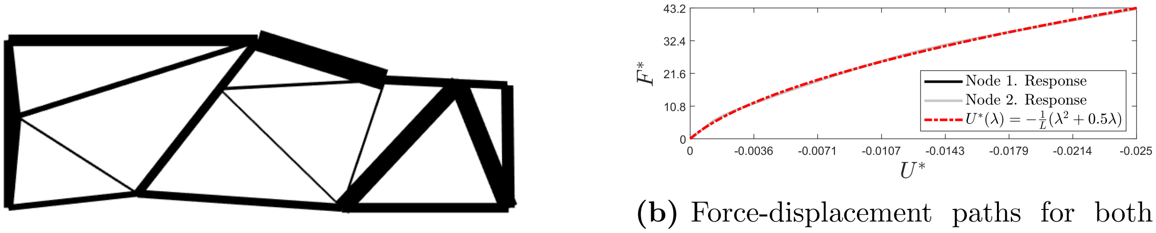


(a) Optimized Structure

(b) Resulting force-displacement paths

Figure 6.8. Cantilever whose end-tip was optimized in multiple directions

In a similar vein multiple nodes may be optimized at once. Figure 6.9 shows a structure who has both its loading node and the labeled grey node in fig. 6.1a optimized simultaneously, effectively optimizing the motion of lower-right edge. Both nodes follow the target curve given by $U^*(\lambda) = -\frac{1}{L_c}(\lambda^{*2} + 0.5\lambda)$, and whose structure and force-displacement curves are viewable in fig. 6.9. The motion of the tip node matches the target curve used in figs. 6.2 and 6.3, showing how the topologies produced by this method are non-unique.



(a) Optimized Structure

(b) Force-displacement paths for both nodes

Figure 6.9. Cantilever with multiple nodes following different paths

6.2 Fixed-Fixed Bridge with Point Load

This section will present the results obtained by optimizing the topology of a fixed-fixed bridge. This structure possesses a stiffening response under load, the boundary conditions used to define the bridge which are shown in fig. 6.10.

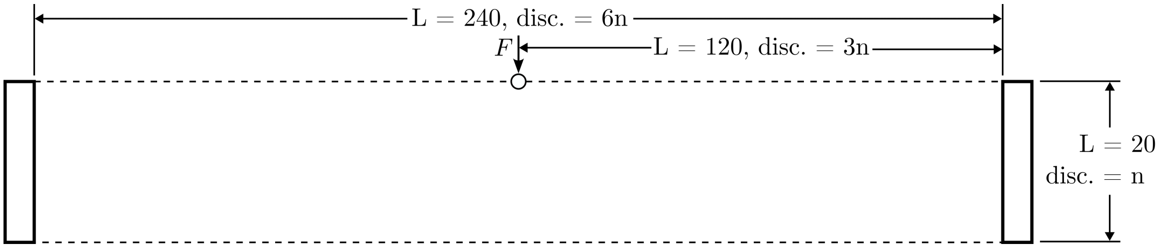
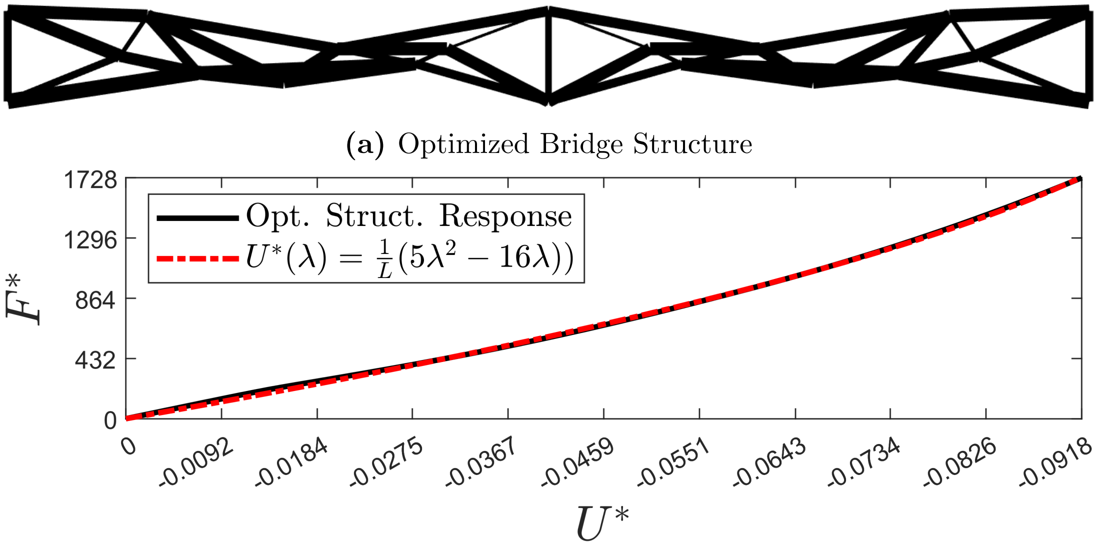


Figure 6.10. Bridge Boundary Conditions

With the target curve defined by: $U(\lambda) = -5(\lambda)^2 + 16(\lambda)$ with $F = 1e6$, the optimized structure and its corresponding force-displacement curve may be seen in figs. 6.11a and 6.11b respectfully.

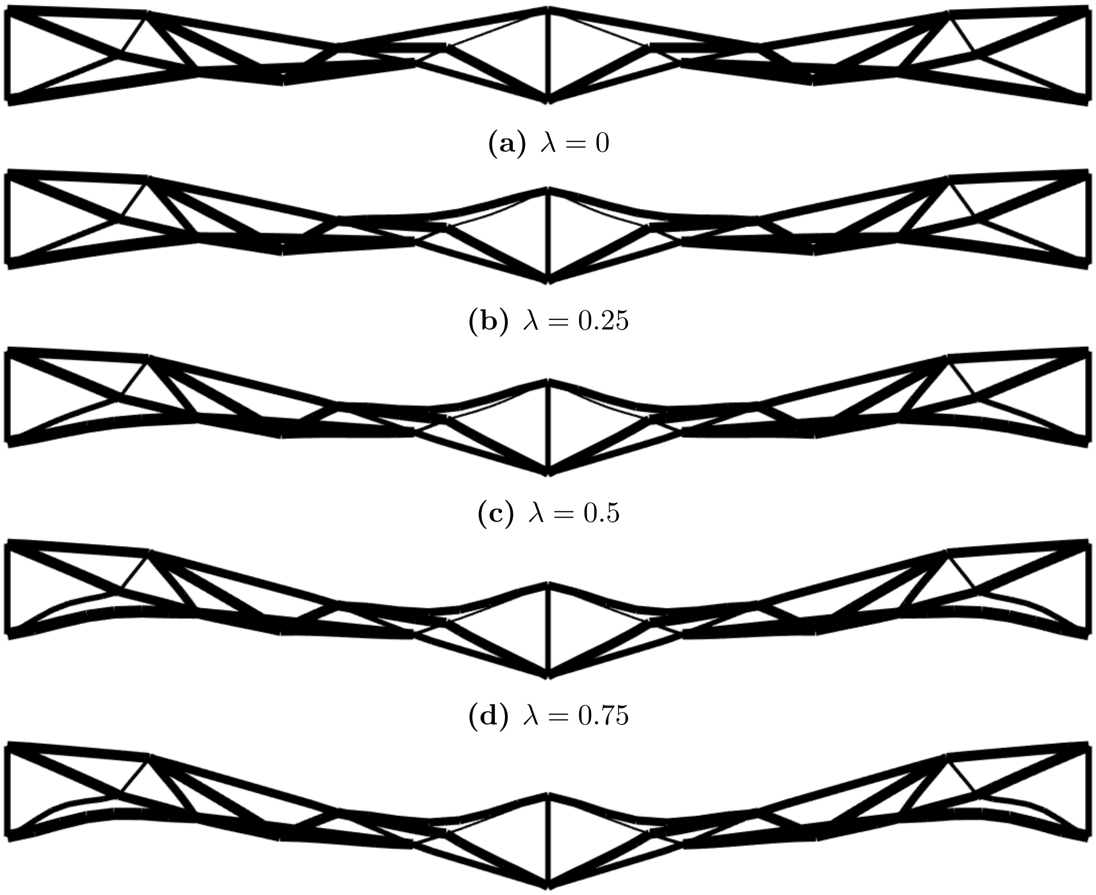


(a) Optimized Bridge Structure

(b) Force-displacement curve for fig. 6.11a

Figure 6.11. Optimized Bridge following stiffening force-displacement path

The motion of the bridge shown in fig. 6.11a is displayed in fig. 6.12



(a) $\lambda = 0$

(b) $\lambda = 0.25$

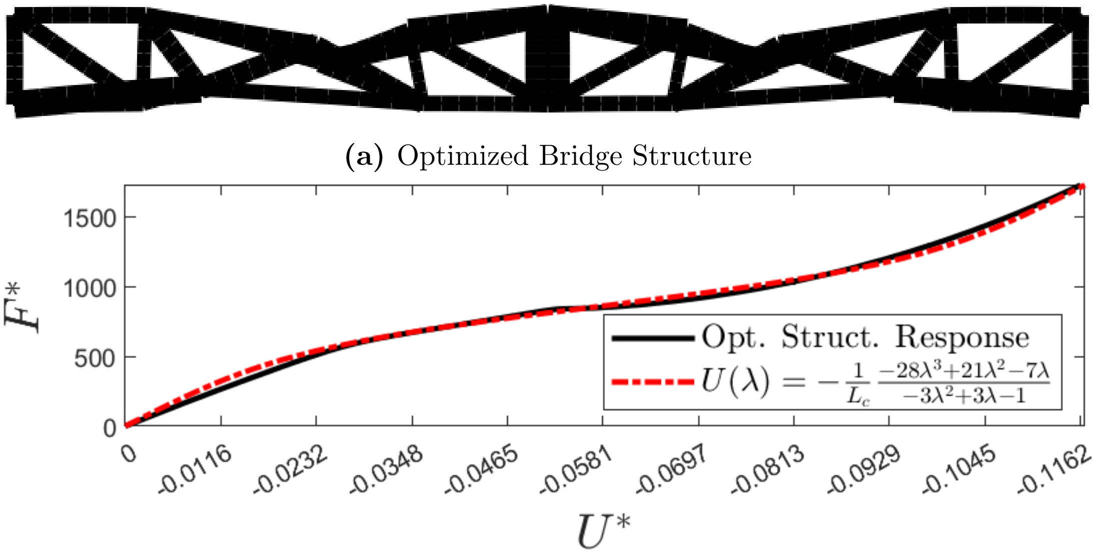
(c) $\lambda = 0.5$

(d) $\lambda = 0.75$

(e) $\lambda = 1.00$

Figure 6.12. Motion of a Bridge following stiffening force-displacement path

Whereas the bridge shown in fig. 6.11 exhibits a continuing stiffen response, as second bridge shown in fig. 6.13 was created that initially softens then stiffens, resulting in a cubic-like response.

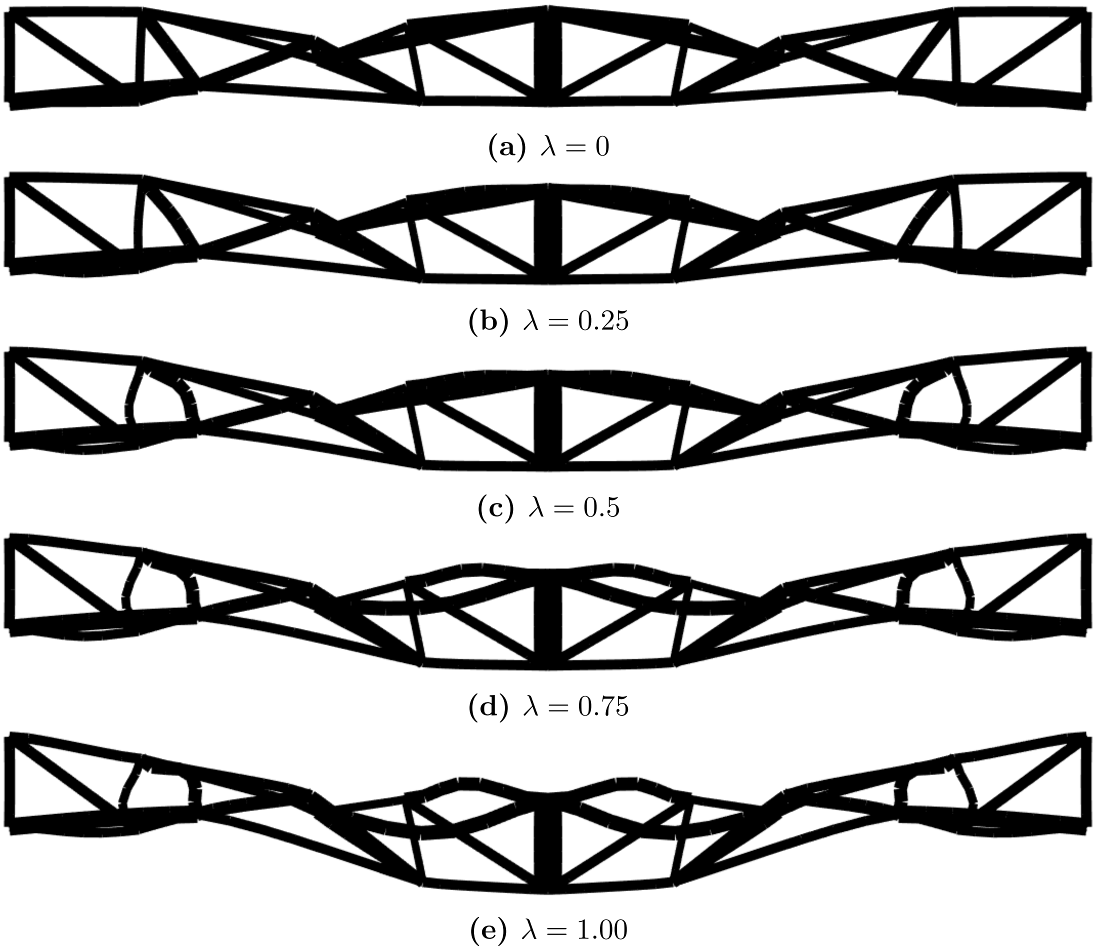


(a) Optimized Bridge Structure

(b) Force-displacement curve for fig. 6.13a

Figure 6.13. Optimized Bridge with a softening-stiffening force-displacement path

The motion of the bridge shown in fig. 6.13 is displayed in fig. 6.14. In the unloaded state, several beams are in close proximity and intersect when the structure is loaded. Physically, this is not possible and necessitates the inclusion of internal-contact in analysis to prevent such occurrences; however, this is beyond the scope of the present work and is discussed later in chapter 7.



(a) $\lambda = 0$

(b) $\lambda = 0.25$

(c) $\lambda = 0.5$

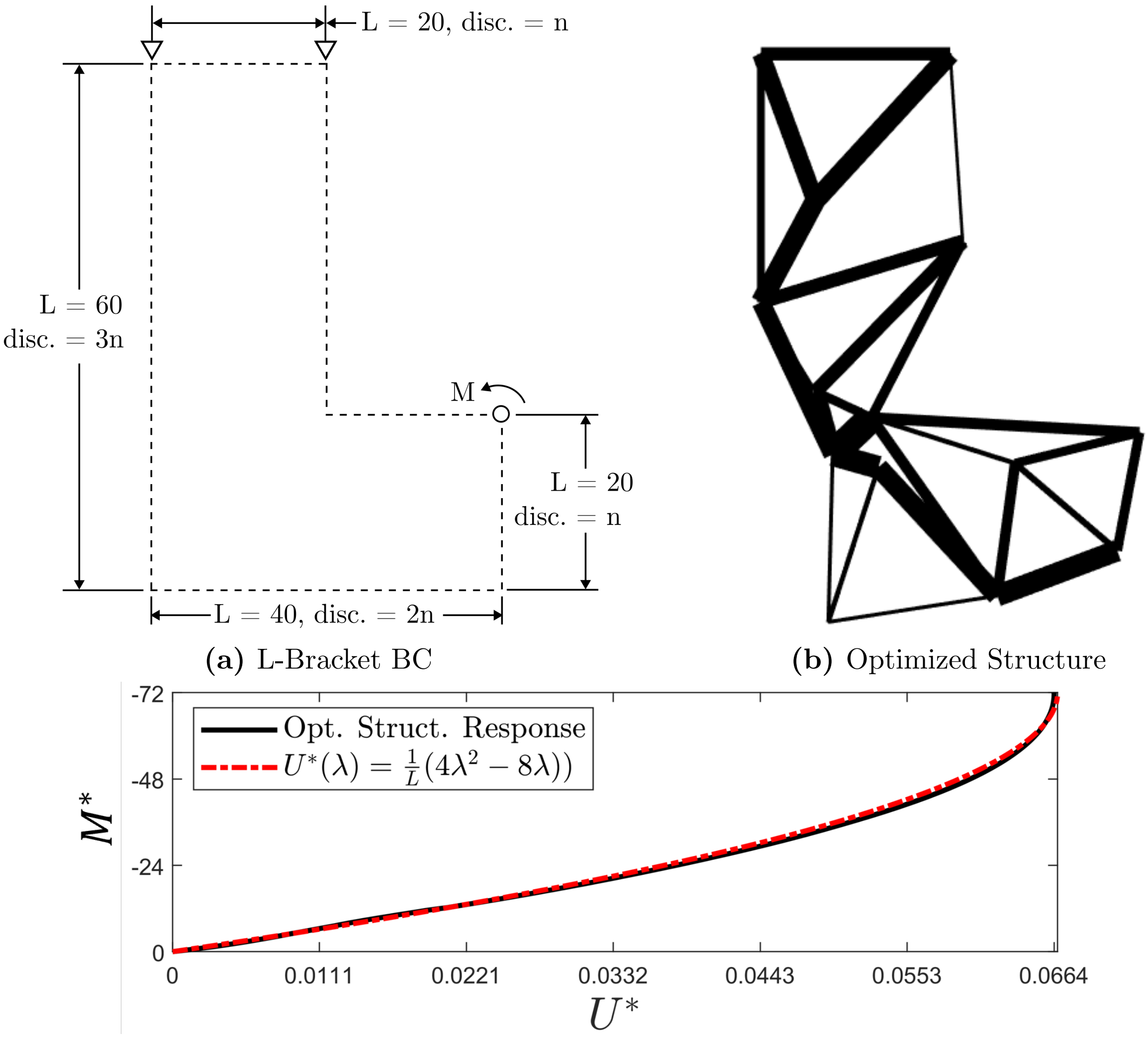
(d) $\lambda = 0.75$

(e) $\lambda = 1.00$

Figure 6.14. Motion of a Bridge following softening-stiffening force-displacement path

6.3 L-Bracket with concentrated Moment

Similar to the process used in linear elements; to show the capability of the method this section will investigate a non-rectangular domain in the shape of an L. This time however the concentrated load is replaced with a concentrated moment, resulting in the boundary conditions shown in fig. 6.15a. To optimize the structure, the target curve was given as $U(\lambda) = -5(\lambda)^2 + 16(\lambda)$, resulting in the optimized structure and its force displacement curve may be seen in figs. 6.15 and 6.15b respectfully.



(a) L-Bracket BC

(b) Optimized Structure

(c) Moment-displacement curve for fig. 6.15b

Figure 6.15. Optimized L-Bracket following a target force-displacement path

The motion of this L-Bracket is viewable in fig. 6.16. Similar to the second bridge example (fig. 6.13) this structure also possess beams that pass through one another, this structure also has beams that intersect.

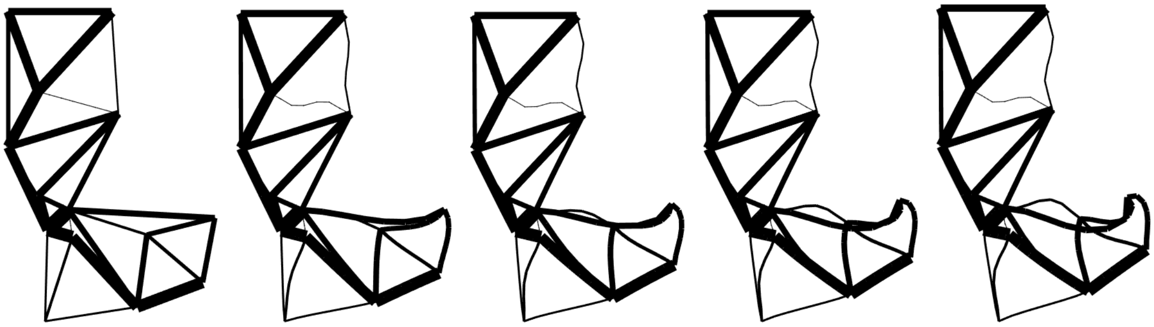

(a) $\lambda = 0$ **(b)** $\lambda = 0.25$ **(c)** $\lambda = 0.5$ **(d)** $\lambda = 0.75$ **(e)** $\lambda = 1.00$

Figure 6.16. Motion of the L-Bracket

6.4 Nonlinear Pincer Mechanism

This section will produce a pincer mechanism using the boundary condition shown in fig. 6.17. The pincer investigated in the nonlinear case features a wider jaw than the pincer examined in the linear case and replaces the pin conditions with a fixed boundary condition. This section will explore two types of pincer problems and provide two examples of each. The first case will match two simple curves, while the second case will explore a more complex curve.

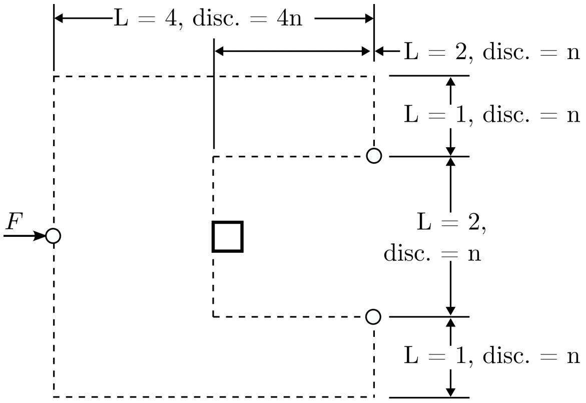
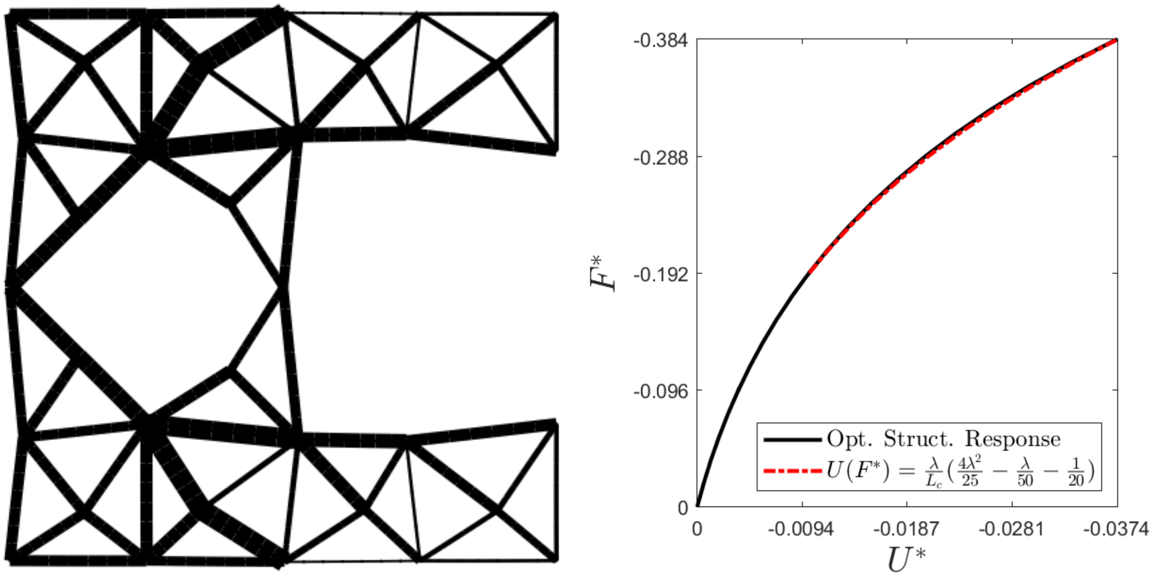


Figure 6.17. Nonlinear Pincer Boundary Conditions

The first structure will follow a softening target curve given by $U(\lambda) = \frac{1}{L}(0.16\lambda^3 -$

$0.02\lambda^2 - 0.05\lambda)$. The optimized structure and its force displacement curve may be seen in figs. 6.22 and 6.22a respectfully.

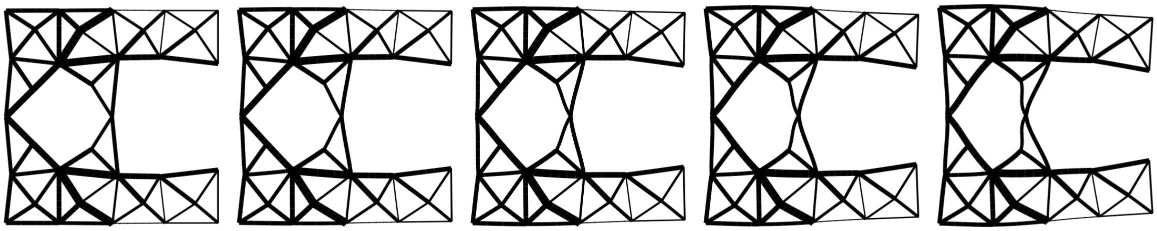


(a) Opt. Struct.

(b) Force-displacement curve for fig. 6.18a

Figure 6.18. Optimized Pincer following a softening force-displacement path

The motion of this structure may be seen in fig. 6.19



(a) $\lambda = 0$

(b) $\lambda = 0.25$

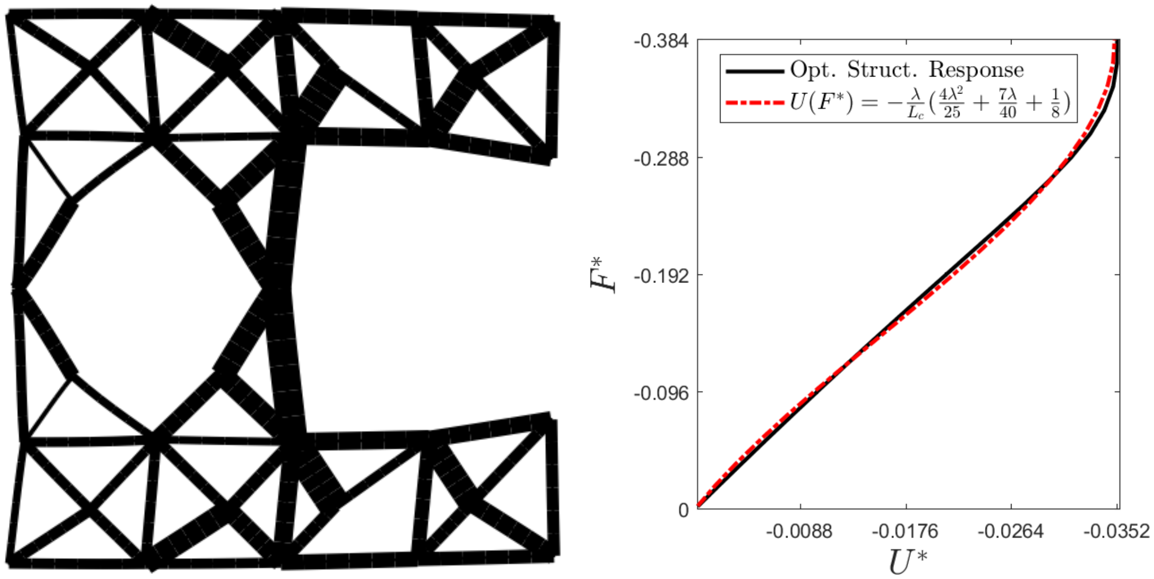
(c) $\lambda = 0.5$

(d) $\lambda = 0.75$

(e) $\lambda = 1.00$

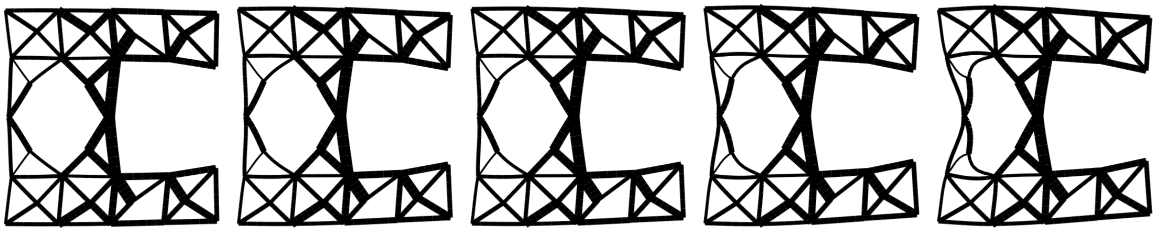
Figure 6.19. Motion of the Softening Pincer

A second pincer possesses an initially linear loading behavior, followed by a rapidly stiffening response. To produce the structure, the target curve of $U(\lambda) = \frac{1}{L}(0.16\lambda^3 + 0.175\lambda^2 + 0.125\lambda)$ was used. The optimized structure and its force-displacement curve may be seen in figs. 6.20 and 6.20a respectfully, with its motion depicted in fig. 6.21.



(a) Opt. Struct. **(b)** Force-displacement curve for fig. 6.18a

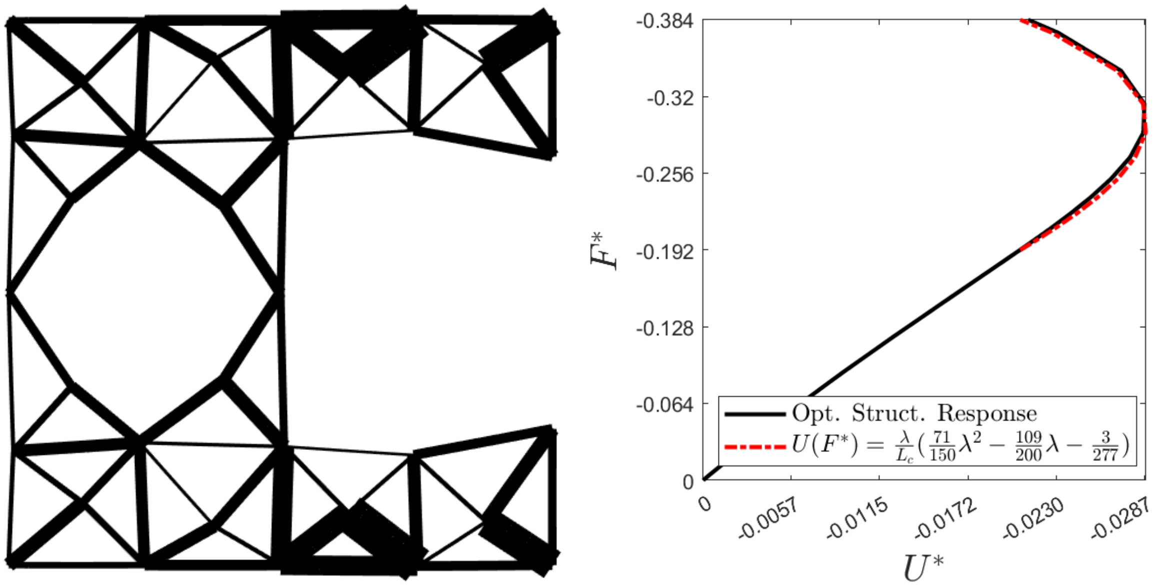
Figure 6.20. Optimized Pincer following a linear-stiffening force-displacement path



(a) $\lambda = 0$ **(b)** $\lambda = 0.25$ **(c)** $\lambda = 0.5$ **(d)** $\lambda = 0.75$ **(e)** $\lambda = 1.00$

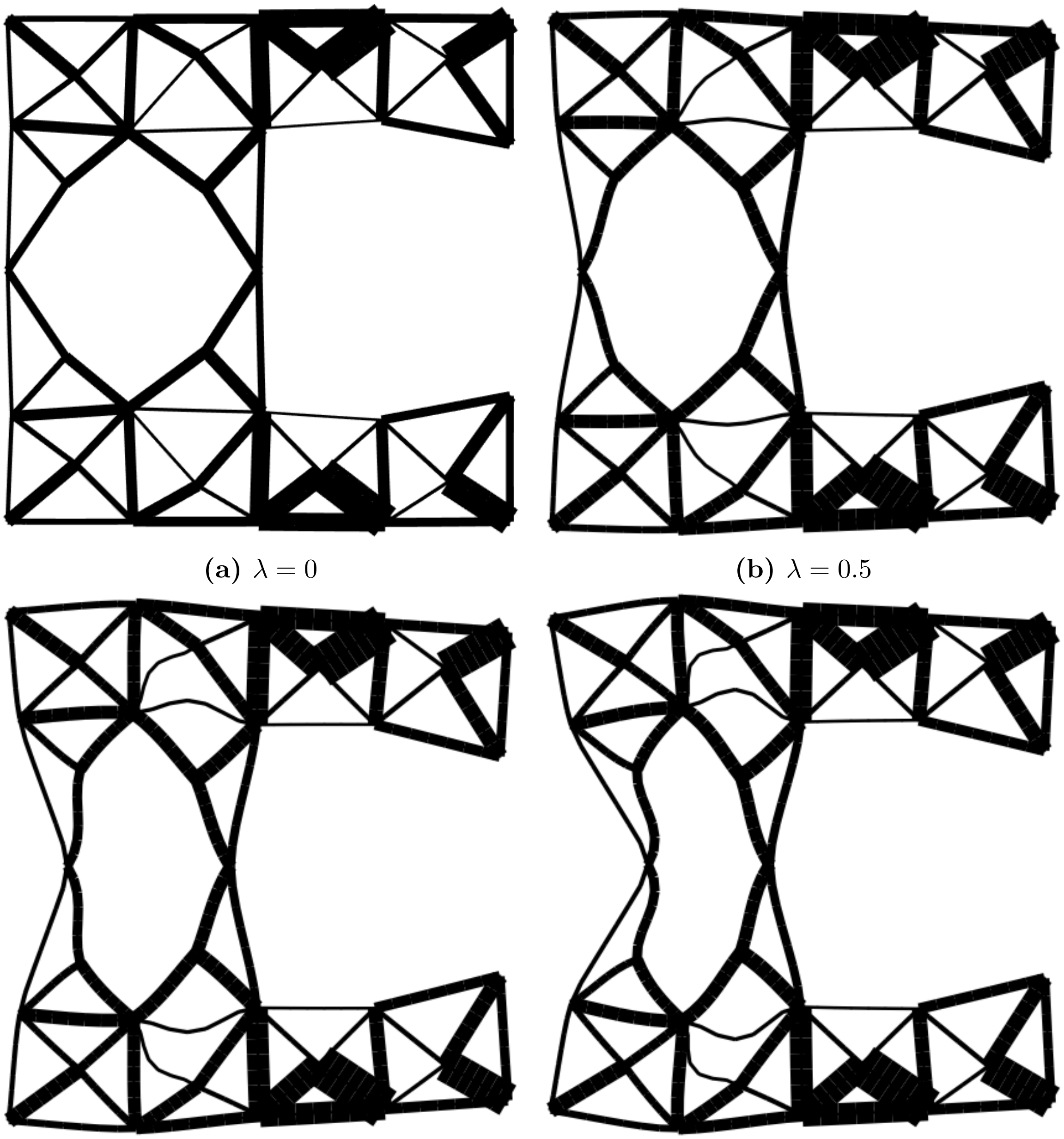
Figure 6.21. Motion of the linear-stiffening Pincer

The second set of optimization problems for the gripper will initially move the tips together, only to reverse direction once a threshold has been met. Practically, this behavior can be seen as a safety feature; where instead of the tips continuing to close after a target force has been met, they instead retract, preventing the gripper from grasping an inserted object too tightly. To produce this behavior, the target curve is defined by: $U(\lambda) = \frac{1}{L}(0.4733(\lambda)^3 - 0.545(\lambda)^2 - 0.01083(\lambda) - 1)$. As this optimization is more interested in producing this retraction behavior, only displacements at half and full load were optimized. The optimized structure and its force displacement curve may be seen in figs. 6.22 and 6.22a respectfully, with its motion shown in fig. 6.23.



(a) Opt. Struct. **(b)** Force-displacement curve for fig. 6.18a

Figure 6.22. Optimized pincer that retracts once a load threshold has been exceeded



(a) $\lambda = 0$

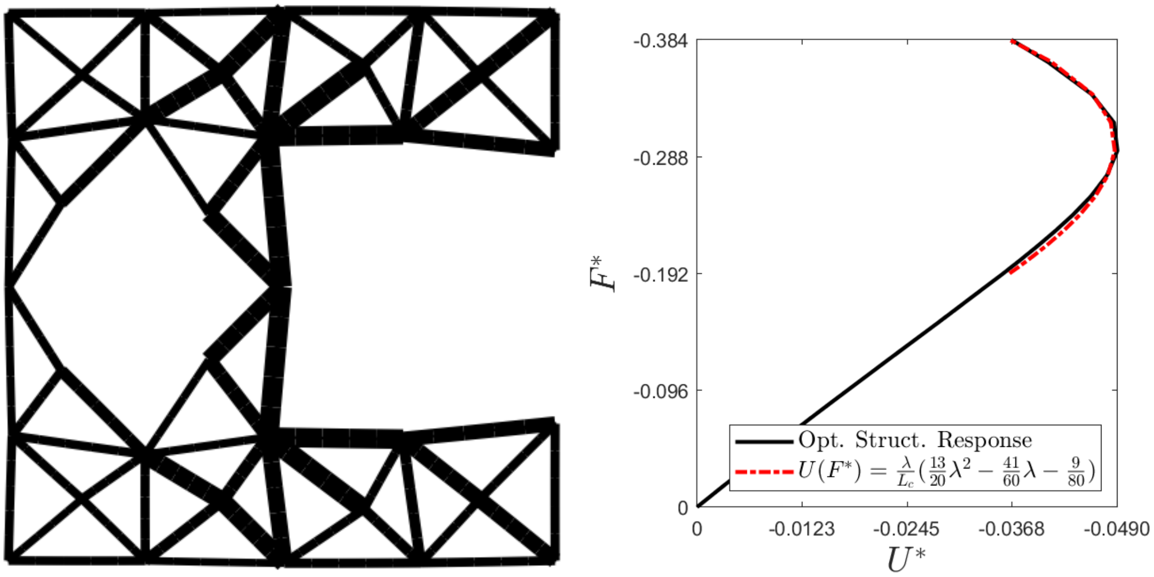
(b) $\lambda = 0.5$

(c) $\lambda = 0.75$

(d) $\lambda = 1.00$

Figure 6.23. Motion of the Retraction Pincer

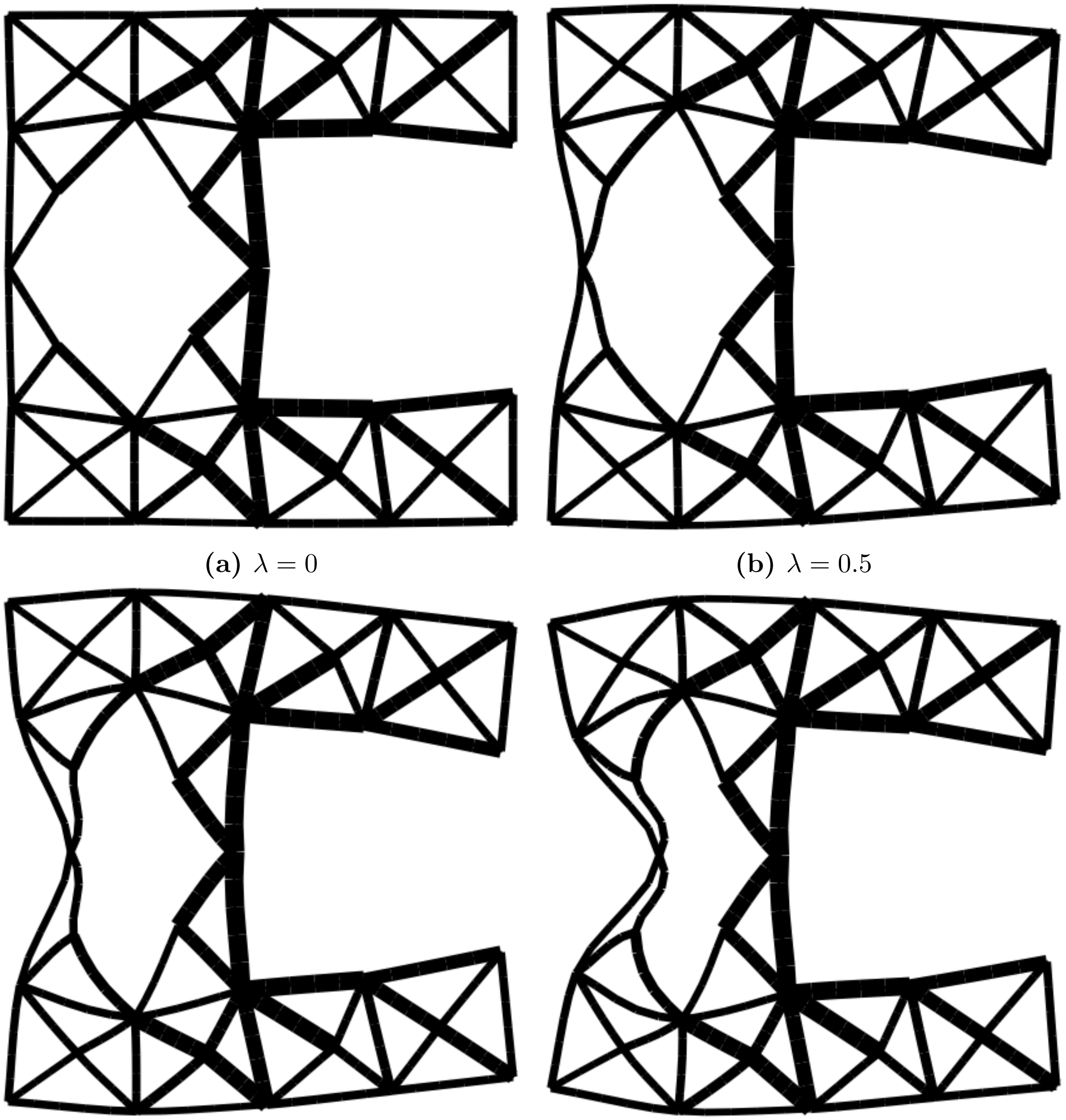
A second-retraction pincer was produced, this time with more displacement defined by: $U(\lambda) = \frac{1}{L}(0.4733(\lambda)^3 - 0.545(\lambda)^2 - 0.01083(\lambda) - 1)$. The resulting optimized structure and its force-displacement curve may be seen in figs. 6.24 and 6.24a, with its motion depicted in fig. 6.25.



(a) Opt. Struct.

(b) Force-displacement curve for fig. 6.18a

Figure 6.24. Optimized Pincer with a larger Retraction



(a) $\lambda = 0$

(b) $\lambda = 0.5$

(c) $\lambda = 0.75$

(d) $\lambda = 1.00$

Figure 6.25. Motion of the Pincer with a larger Retraction

This clamp concludes the nonlinear problems investigated in the work, and is the last example presented for both the linear and nonlinear topology optimization problems. The following section concludes the manuscript by providing a summary of the work and plans for future work.

Chapter 7

Final Thoughts

7.1 Conclusion

This work introduces a novel approach to Topology Optimization using a morphing beam network. The method constructs a network of beams and optimizes each beam's width and node positions simultaneously to achieve a desired design objective. By leveraging beam elements, the method allows for the use of coarse meshes constructed with beam elements, significantly reducing both finite element simulation and optimization computational costs while retaining the result fidelity when compared to existing methods. This efficiency enables exploration of more computationally demanding analyses, such as nonlinear deformations as demonstrated in this study.

Chapter 2 discusses the method used to construct the beam network considered in this work. Construction of the network begins by first covering a domain with an orthogonal grid of nodes. Each node is then connected to its immediate neighbors, with an additional node inserted at the point of intersection.

To optimize the network, an analysis of the network needs to be performed, which was described in chapter 3. Two different methods were investigated in this work: the first considered only linear-elastic behavior, while the second considered nonlinear effects occurring from element deformations.

With the topological and analysis methods defined, the last step in defining the optimization process is the optimization formulation, which was detailed in chapter 4. This method defines objective functions, gradients, constraints, and algorithms used to optimize the network.

Once the proposed method was defined, it is important to verify the method's ability to reproduce existing results. Chapter 5 demonstrated the proposed method's capability by obtaining comparable results to those achieved by an existing solid element solution, but with a fraction of the computational cost. With the proposed method's ability to reproduce existing results established, the final goal in this work was to increase the fidelity of the physics considered and optimize structures that have user-defined mechanical-response behavior. The results of this effort were successful and are shown in chapter 6.

In conclusion, this work has introduced a novel method for topology optimization utilizing a morphing beam network, demonstrating significant computational efficiency without loss of fidelity compared to traditional methods. By constructing and optimizing a network of beams, the method effectively reduces computational costs while approximating the topology of a solid domain. The successful outcomes presented in the work underscore the viability and advantages of this approach in advancing the field of topology optimization.

7.2 Future Work

This work has established a foundational method for topology optimization using a morphing beam network. As outlined in chapter 1, topology optimization can be categorized into three main components: topology representation, analysis type, and optimization method. Building upon this framework, several avenues for future research and development are identified:

1. Topology Construction

   (a) The network construction considered in this paper constructed elements as straight connections between its two nodes. Extending the current method to include curved beams, or splines, instead could potentially improve the design space and allow for more organic and efficient structural layouts.

   (b) Similarly, the cross-sectional area was held constant across each element. However, optimizing the cross-sectional areas can further optimize structural performance and efficiency of the resulting structures.

   (c) While the extension to 3D is straightforward, it should be pursued to broaden the applicability of the method.

2. Analysis Method

   (a) This paper only considered mechanical behavior. Expanding the morphing beam network to tackle other types of engineering problems such as vibration, heat transfer, and fluid dynamics would enhance the method's versatility and applicability across different disciplines while retaining decreases in computational cost.

(b) The analysis method explored in this work consists of the stiffness definition using Co-rotational formulation and a Newton-Raphson solution algorithm. There are many alternative methods for obtaining both the stiffness behavior and solving the nonlinear system. One of interest are Riks, or arc-length algorithms, which can more appropriately model snap-through which can allow for optimizing structures that possess those abilities.

(c) When undergoing large deformations, some element pass-through one another. Physically this is not possible. Therefore internal contact should be included in the model to prevent this behavior from happening.

3. Optimization

   (a) To prevent overlapping nodes which would result in a zero-length, nodes are constrained to lie within their starting voronoi cells (section 4.1.3). While this method is shown to be successful in reproducing the results obtained from solid elements, ideally the nodes should be free to navigate the domain. This would require two algorithms. One to detect overlapping elements and insert new nodes dynamically, and one to merge nodes that become coincident. The later algorithm was investigated during construction of the method, but produced results that varied greatly from existing solid element solutions.

   (b) The node constraint definition for clipping, eq. (4.11), is computationally expensive. All domains investigated in this work were rectangular in construction, thus the rotation method described in eq. (4.13) was sufficient. But for domains that feature curved boundaries, likely faster methods should be investigated to retain the computational advances.

   (c) The optimization method used in this paper was gradient descent, but higher-order methods should be investigated which either may speed up the solution time or produce better results.

(d) With regards of taking higher order derivatives; is an interesting event that happens after taking the fourth derivative of $\mathbf{K}$ w.r.t. $\mathbf{z}$. The terms of regarding $\mathbf{w}$ are zero, yet node positions continue to be nonzero. Optimization methods that pursue this high of differentiation a rare to put it mildly, but this behavior is interesting enough to be worthy of investigation.

These avenues represent opportunities for advancing the proposed methodology and expanding its capabilities to address more complex design challenges. By pursuing these directions, future research can further solidify the method's position as a versatile and efficient tool for topology optimization across various engineering disciplines.

Bibliography

[1] James K. Guest et al. “Structural topology optimization: MOVING BEYOND LINEAR elastic design objectives”. In: *20th Analysis and Computation Specialty Conference* (2012). DOI: 10.1061/9780784412374.022.

[2] Talib Dbouk. “A review about the engineering design of optimal heat transfer systems using topology optimization”. In: *Applied Thermal Engineering* 112 (2017), pp. 841–854.

[3] Sajjad Zargham et al. “Topology optimization: a review for structural designs under vibration problems”. In: *Structural and Multidisciplinary Optimization* 53 (2016). contains automotive aero, pp. 1157–1177.

[4] Andrew Montalbano, Georges M. Fadel, and Gang Li. “Design for energy absorption using snap-through bistable metamaterials”. In: *Mechanics Based Design of Structures and Machines* 51.3 (2021), pp. 1368–1386. DOI: 10.1080/15397734.2020.1867167.

[5] Singiresu S Rao. *Engineering optimization: theory and practice*. John Wiley & Sons, 2019.

[6] Erik Andreassen et al. “Efficient topology optimization in MATLAB USING 88 lines of code”. In: *Structural and Multidisciplinary Optimization* 43.1 (2010), pp. 1–16. DOI: 10.1007/s00158-010-0594-7.

[7] Kai Liu and Andrés Tovar. “An efficient 3D topology optimization code written in Matlab”. In: *Structural and multidisciplinary optimization* 50 (2014), pp. 1175–1196.

[8] Tomás Zegard and Glaucio H Paulino. “GRAND—Ground structure based topology optimization for arbitrary 2D domains using MATLAB”. In: *Structural and Multidisciplinary Optimization* 50 (2014), pp. 861–882.

[9] Tyler E Bruns. “Topology optimization of convection-dominated, steady-state heat transfer problems”. In: *International Journal of Heat and Mass Transfer* 50.15-16 (2007), pp. 2859–2873.

[10] Pedro Gomes and Rafael Palacios. “Aerostructural topology optimization using high fidelity modeling”. In: *Structural and Multidisciplinary Optimization* 65.5 (2022), p. 137.

[11] Srinivas Vasista and Liyong Tong. “Topology-optimized design and testing of a pressure-driven morphing-aerofoil trailing-edge structure”. In: *AIAA journal* 51.8 (2013), pp. 1898–1907.

[12] Y Maeda et al. “Structural topology optimization of vibrating structures with specified eigenfrequencies and eigenmode shapes”. In: *International Journal for Numerical Methods in Engineering* 67.5 (2006), pp. 597–628.

[13] Lidan Zhang, Yi Zhang, and Fred Van Keulen. “Topology optimization of geometrically nonlinear structures using reduced-order modeling”. en. In: *Computer Methods in Applied Mechanics and Engineering* 416 (Nov. 2023), p. 116371. ISSN: 00457825. DOI: 10.1016/j.cma.2023.116371. URL: https://linkinghub.elsevier.com/retrieve/pii/S0045782523004954.

[14] AA Polynkin, F Van Keulen, and VV Toropov. “Optimization of geometrically nonlinear thin-walled structures using the multipoint approximation method”. In: *Structural optimization* 9 (1995), pp. 105–116.

[15] Zachary Satterfield et al. “Unit cell synthesis for design of materials with targeted nonlinear deformation response”. In: *Journal of Mechanical Design* 139.12 (2017), p. 121401.

[16] Gore Lukas Bluhm, Ole Sigmund, and Konstantinos Poulios. “Inverse design of mechanical springs with tailored nonlinear elastic response utilizing internal contact”. In: *International Journal of Non-Linear Mechanics* 157 (2023), p. 104552.

[17] Navid Changizi et al. “Topology optimization of structural frames based on a nonlinear Timoshenko beam finite element considering full interaction”. In: *International Journal for Numerical Methods in Engineering* 123.19 (2022), pp. 4562–4585. DOI: 10.1002/nme.7046.

[18] Nigel J.W. Morris, Pradeep Kumar Jayaraman, and Adrian Butscher. “BeNTO: Beam Network Topology Optimization”. en. In: *Computer-Aided Design* 156 (Mar. 2023), p. 103439. ISSN: 00104485. DOI: 10.1016/j.cad.2022.103439. URL: https://linkinghub.elsevier.com/retrieve/pii/S0010448522001725.

[19] Alireza Asadpoure and Lorenzo Valdevit. “Topology optimization of lightweight periodic lattices under simultaneous compressive and shear stiffness constraints”. en. In: *International Journal of Solids and Structures* 60–61 (May 2015), pp. 1–16. ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2015.01.016. URL: https://linkinghub.elsevier.com/retrieve/pii/S0020768315000268.

[20] Mariusz Pyrz. “Evolutionary algorithm integrating stress heuristics for truss optimization”. In: *Optimization and Engineering* 5.1 (2004), pp. 45–57. DOI: 10.1023/b:opte.0000013634.36489.32.

[21] Meisam Abdi, Ian Ashcroft, and Ricky Wildman. “Topology optimization of geometrically nonlinear structures using an evolutionary optimization method”. en. In: *Engineering Optimization* 50.11 (Nov. 2018), pp. 1850–1870. ISSN: 0305-215X, 1029-0273. DOI: 10.1080/0305215X.2017.1418864. URL: https://www.tandfonline.com/doi/full/10.1080/0305215X.2017.1418864.

[22] Jinyong Joo and Sridhar Kota. “Topological Synthesis of Compliant Mechanisms Using Nonlinear Beam Elements”. en. In: *Mechanics Based Design of Structures and Machines* 32.1 (Dec. 2004), pp. 17–38. ISSN: 1539-7734, 1539-7742. DOI: 10.1081/SME-120026588. URL: http://www.tandfonline.com/doi/abs/10.1081/SME-120026588.

[23] Teerapol Techasen et al. “Simultaneous topology, shape, and size optimization of trusses, taking account of uncertainties using multi-objective evolutionary algorithms”. In: *Engineering with Computers* 35 (2019), pp. 721–740.

[24] Anthony George Maldon Michell. “LVIII. The limits of economy of material in frame-structures”. In: *Philosophical Magazine Series 1* 8 (1904), pp. 589–597. URL: https://api.semanticscholar.org/CorpusID:120415256.

[25] OEIS Foundation Inc. *Entry A083374*. URL: https://oeis.org/A083374.

[26] OEIS Foundation Inc. *Entry A002943*. URL: https://oeis.org/A002943.

[27] OEIS Foundation Inc. *Entry A046092*. URL: https://oeis.org/A046092.

[28] William McGuire, Richard H Gallagher, and Ronald D Ziemian. *Matrix structural analysis*. 2000.

[29] Rafael Lopez Rangel. “EDUCATIONAL TOOL FOR STRUCTURAL ANALYSIS OF PLANE FRAME MODELS WITH GEOMETRIC NONLINEARITY”. en. MESTRE EM ENGENHARIA CIVIL. Rio de Janeiro, Brazil: PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, Sept. 2019. DOI: 10.17771/PUCRio.acad.47858. URL: http://www.maxwell.vrac.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=47858@2.

[30] Nadim Diab. “Adapting Ant Colony for Topology Optimization of Compliant Mechanisms With Variable Material Density”. In: *ASME International Mechanical Engineering Congress and Exposition*. Vol. 58462. American Society of Mechanical Engineers. 2017, V011T15A007.

[31] René de Borst. *Nonlinear finite element analysis of solids and structures*. Wiley, 2012.

[32] 1989. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/bmk/default.htm?startat=ch04s06.html.

[33] Carlos A Almeida et al. “Geometric nonlinear analyses of functionally graded beams using a tailored Lagrangian formulation”. In: *Mechanics Research Communications* 38.8 (2011), pp. 553–559.

[34] Nam H. Kim et al. "Generalized Optimality Criteria Method for Topology Optimization". en. In: *Applied Sciences* 11.7 (2021), p. 3175. ISSN: 2076-3417. DOI: 10.3390/app11073175. URL: https://www.mdpi.com/2076-3417/11/7/3175.

[35] 1989. URL: https://classes.engineering.wustl.edu/2009/spring/mase5513/abaqus/docs/v6.6/books/bmk/ch04s06anf41.html.