

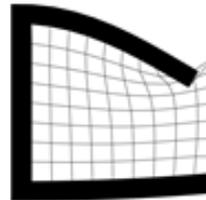
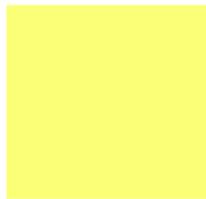


Department of Precision and Microsystems Engineering

Approximate geometric non-linear analysis in topology optimization

Jasper Hoevenaars

Report no : 2021.071
 Coach : Stijn Koppen
 Professor : Matthijs Langelaar
 Specialisation : EM
 Type of report : Master thesis
 Date : 1 October 2021



Approximate geometric non-linear analysis in topology optimization

A novel kind of structural analysis for topology optimization of finite-range compliant mechanisms

by

Jasper Hoevenaars

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday October 1, 2021 at 09:00.

Student number: 4487281
Project duration: June 16, 2020 – October 1, 2021
Thesis committee: Prof. dr. ir. M. Langelaar, TU Delft, supervisor
ir. S. Koppen, TU Delft, supervisor
Prof. dr. ir. F. v Keulen, TU Delft, committee member

This thesis is confidential and cannot be made public until March 31, 2022.

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This work is the final part in obtaining my master's degree at Precision and Microsystems Engineering at the Delft University of Technology. It was a journey in which an idea unveiled itself, which led to a new method being developed. This new method was implemented and studied over the course of a year (and a bit).

First and foremost, I would like to thank my daily supervisor Stijn for all the constructive discussions we have had. Stijn also helped me with all non-content-related things in a master thesis, guidance I definitely needed. Next, I would like to thank Matthijs, who has always been open and encouraging concerning my research. But most of all I would like to thank him for his patience in checking my work and providing feedback on all written aspects of this thesis.

Writing a thesis, which is a solo project for the most part, during the pandemic comes with extra disconnection from the world. Therefore, I would like to thank all my friends and family for keeping me with two feet on the ground during this sometimes wonderful and sometimes exhaustive process.

*Jasper Hoevenaars
Delft, October 2021*

Contents

1	Introduction	1
1.1	Optimizing compliant mechanisms	1
1.2	Research objective	4
2	Paper	5
2.1	Introduction	6
2.2	Method	9
2.3	Verification	13
2.4	Topology optimization formulation	17
2.5	Numerical examples	20
2.6	Discussion	31
2.7	Conclusion	34
3	Additional response functions	37
3.1	Derivatives with respect to the load factor.	37
3.2	Norm of the residual	37
3.3	Reaction forces	38
4	Discussion	41
4.1	Approximate analysis.	41
4.2	Response functions	41
5	Conclusion	43
A	Sensitivity analysis	44
A.1	Adjoint method	44
A.2	End-compliance	45
A.3	1-DOF	46
A.4	Residual.	47
A.5	Second derivative of the load curve	49
A.6	Compliance shape	50
A.7	Norm of reaction forces	51
A.8	Finite Difference check	52
B	Minimal working example	54
B.1	Load case.	54
B.2	Results	55
C	Geometric non-linear FEM	59
C.1	Analytical verification	59
C.2	Verification with Comsol	60
D	Scaling with Buckingham Pi theorem	62
E	Other solutions to the inverter problem	64

Introduction

The creation of mechanisms is a craft practiced by humans since the very first moment they set foot on earth. Whether it is to perform a task previously deemed impossible, or just to make life easier, creating a mechanism often offers a solution. Old mechanisms get replaced or improved and new ones are designed in the world on a continuous basis. To this day the optimization and design of new mechanisms is a big topic in the academic world. The need for newer, better or different machines seems like a never-ending quest.

In the high-tech industry mechanisms with high repeatability and precision are required to create ever more intricate parts. A certain class of mechanisms seems to meet those criteria, namely compliant mechanisms. These mechanisms are created out of one piece of continuous material, which is why they do not exhibit wear and tear and have no play in their joints (Howell et al., 2013). An added benefit is that there is no assembly needed to create these mechanisms. For these reasons, compliant mechanisms have great potential in the high-tech industry.

1.1. Optimizing compliant mechanisms

The design and optimization of compliant mechanisms is challenging due to the close relation between mechanism motion and stiffness within its structure. A popular design method for compliant mechanisms is structural optimization. There are three categories of structural optimization (Fig. 1.1):

- **Sizing optimization:** Here, the structure is already known, but some parameters of the structure are optimized. An example is a truss structure of which all locations and connection points of the rods are known, but the thicknesses of the beams are not. An optimization algorithm is then employed to find the optimal thicknesses for the beams in the structure.
- **Shape optimization:** In shape optimization, the topology, i.e. the distributions of holes in a structure, is known. Then, the optimization process finds the optimal size and locations for those

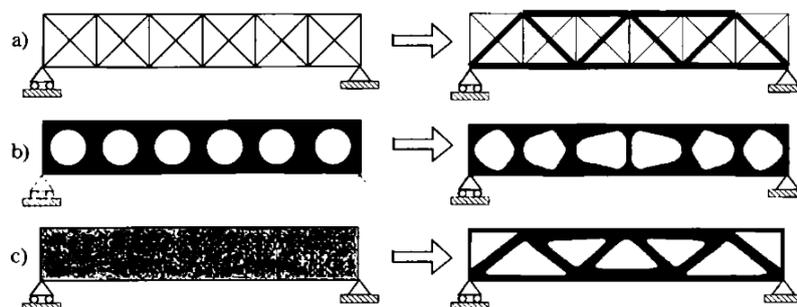


Figure 1.1: Three categories of structural optimization of a truss structure. a) Sizing optimization, b) shape optimization and c) topology optimization. The initial problems are shown on the left and the optimal solutions are shown on the right (Bendsøe et al., 2004).

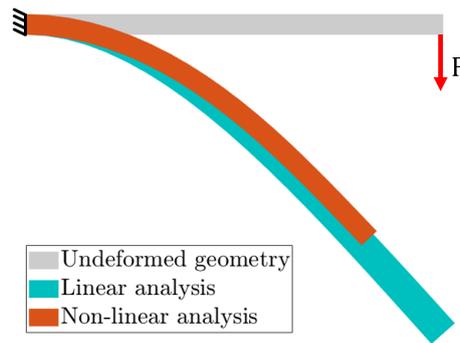


Figure 1.2: A beam that is fixed to the left with a tip load F on the right. As the beam deforms, the stiffness of the beam changes, due to geometric non-linear effects. This change of stiffness is not taken into account when linear analysis is used.

holes, without altering the topology.

- **Topology optimization:** In this method, nothing is known about the structure, except for the boundary conditions and the design space. Topology optimization is an iterative method, in which an optimizer looks for the best material distribution in a certain design space according to the given objective function and constraints. This means that the optimizer can add holes and solid material as it sees fit. Some other constraints or material models may be needed for a proper problem formulation, but it can be safely said that it is the most free form of structural optimization of the three.

The design freedom that topology optimization offers is what makes it, in the author's opinion, the most suited method for designing compliant mechanisms.

In order for topology optimization to create designs, a domain is discretized into elements. The densities of those elements are the design variables, for which the optimizer picks a value in each iteration. These design updates are based on the result of a structural and sensitivity analysis of the previous design. The elements in the design domain are thus filled with different densities, which together form a design.

Using topology optimization for compliant mechanism design comes with a few challenges. The type of structural analysis that should be used in topology optimization depends on the deflections that a design exhibits. Compliant mechanisms often show geometric non-linear behaviour, which is not caught well in linear analysis, a concept better explained in Fig. 1.2. The importance of taking geometric non-linearity into account when optimizing compliant mechanisms was well-described by, among others, Pedersen et al. (2001). Implementing non-linear analysis in topology optimization, however, comes with computational effort and stability problems.

Numerical instabilities

Non-linear analysis is done with an incremental iterative scheme. These schemes are prone to diverge when limit or bifurcation points appear. Linear analysis does not suffer from these instabilities, as linear analysis is not an iterative process. The high stiffness differences between the elements in topology optimization make the low stiffness areas prone to invert. This behaviour makes using non-linear analysis in topology optimization extra unstable.

Inversion of elements occurs when the local strains get so large that the element boundaries cross each other, changing the topology of the elements (Fig. 1.3). As Dijk et al. (2014) note, the Green-Lagrange strain measure, often used in non-linear topology optimization, is not physically meaningful for inverted elements. As an effect, no useful sensitivity information can be retrieved from these elements. The solution to the structural analysis is not realistic if elements are inverted, which creates a pass-through error in the optimization process.

To overcome this obstacle, three main approaches have been used: relaxing convergence criteria (Buhl et al., 2000; Pedersen et al., 2001), implementing different constitutive models (Wang et al., 2014; Wallin et al., 2018; Bluhm et al., 2021; Lahuerta et al., 2013) and changing the structure of the finite element analysis (Yoon et al., 2005; Dijk et al., 2014). Each of these methods has shown to be effective

under certain circumstances, some more successful than others. However, a general method against element inversion in most situations is yet to be found. It is still the case that element inversion is a problem in large deflection simulations.

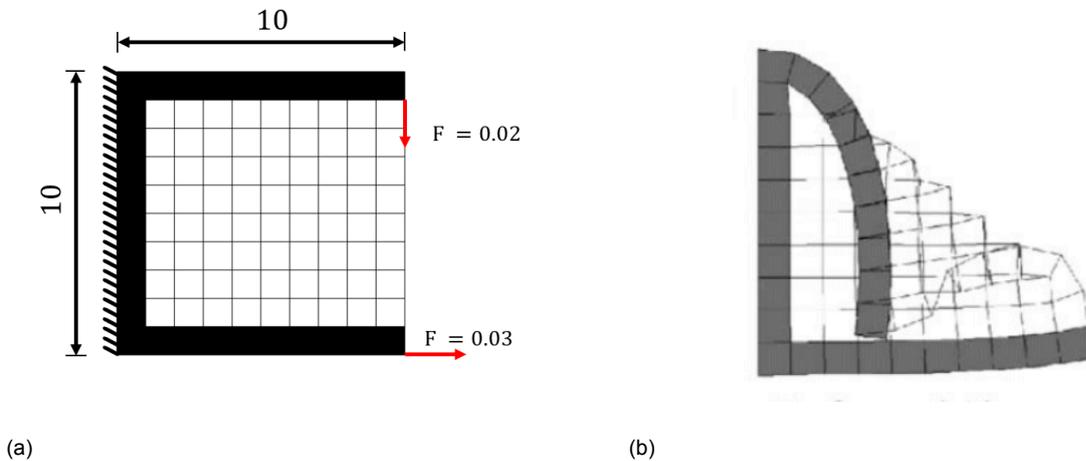


Figure 1.3: The non-linear analysis of the C-shape beam problem as described by Yoon et al. (2005). In a) a geometrical non-linear structure with a unit Young's modulus and a Poisson's ratio of 0.3, under two loads is shown. The void has a stiffness of 10^{-9} . b) shows the deformation field including inverted elements.

Computational effort

In topology optimization a large part of the computational effort lays in solving the linear systems of equations that occur in the structural analysis. Lazarov (2014) claims it accounts for up to 99% of the computational effort in the topology optimization process for linear analysis. Non-linear analysis needs to solve a linear system of equations in each iteration necessary to find a solution, making it computationally more expansive than linear analysis, in which only one linear system of equations needs to be solved. This makes reducing the computational effort used in structural analysis a priority.

There are many techniques to reduce the computational cost of structural analysis in topology optimization. Adaptive quadtree remeshing is a method that refines the element size in areas where strains are high and do the opposite in places that have low strain fluctuations in order to decrease the total amount of degrees of freedom (Maute et al., 1995). Model order reduction is a way to filter out redundant information in the finite element model (Gogu, 2015). In model order reduction, a dynamic model is built in which the solution vector is reduced in size until it has a minimal amount of variables, but still models the same input-output relationship as the original problem (Koutsovasilis et al., 2010). Reanalysis is a method that makes use of the fact that the stiffness matrix for a design is similar to the stiffness matrix of the design of a preceding iteration. This means that the displacement field of a previous iteration can be slightly modified to estimate the displacement field in the current iteration. By doing so, Amir (2015) managed to cut computational effort approximately in half. These techniques are quite common and the papers cited are far from the only ones using the techniques, proving their validity.

Although these methods all reduce the computational effort, there still is room for improvement. Overcoming the problems caused by inverted elements induces extra computational effort, making non-linear analysis extra unattractive. Linear analysis on the other hand, when used in topology optimization, does not create that perform as analyzed during the optimization process. Linear analysis thus creates inaccurate designs. A method that does not suffer from the stability and computational effort issues seen in non-linear analysis and is at the same time more accurate than linear analysis is desired.

1.2. Research objective

An interesting challenge is to create a method that extends the accuracy of the predicted range of motion (finite) of compliant mechanisms compared to linear analysis, with a relatively low computational effort compared to full geometric non-linear analysis. This thesis aims to find such a method.

The main body of this thesis is a paper that investigates a new form of structural analysis and uses it within topology optimization. After the paper, a chapter is devoted to lightly developed response function ideas, which could be used in combination with this form of structural analysis. A supplementary discussion and conclusion are added after that, followed by the appendices.

2

Paper

The paper on the next page is the main body of this thesis. The paper is self-contained with the exception of the Appendices, which can be found at the end of this thesis.

Approximated geometric non-linear analysis in topology optimization for compliant mechanisms

Jasper Hoevenaars

TU Delft, department of Precision and Microsystems Engineering (PME)

September 22, 2021

Abstract

Density-based topology optimization, when used to design structures that show geometrical non-linear behaviour, currently faces computational effort and stability issues. These issues are caused by the iterative method used in geometric non-linear structural analysis. On top of taking much computational effort to complete, this method encounters instabilities when analyzing low-density elements usually present in the design domain.

This study aims to bypass those issues by proposing approximate analysis in the topology optimization routine, which is an analysis based on an approximation of the geometrical non-linear load-deflection curve of a structure, constructed with equilibrium points close to its undeformed configuration.

To study the performance and the influence of the parameters that govern approximate analysis, three numerical examples are considered. These indicate that using approximate analysis in topology optimization leads to designs that perform over a finite range of motion, similar to when a non-linear analysis is used. The computational effort needed for approximate analysis is closer to the effort needed for linear analysis than non-linear analysis. A limitation of approximate analysis is that its results are only similar to non-linear analysis as long as the deflections stay in the mildly non-linear domain.

When concerning the topology optimization of compliant mechanisms that exhibit mildly ge-

ometric non-linear behaviour, we conclude that using approximate analysis is a stable and computationally efficient alternative to non-linear analysis.

1 Introduction

Mechanisms are mechanical devices designed to transfer a movement or force into an action desired by the user. Traditionally, mechanisms are built with rigid links, discrete joints and gears, some of which slide over each other to move. However, this sliding causes wear and tear and is only possible because there is a bit of play between the joints. This abrasion and play cause machines to be less precise and induces maintenance (Beek, 2015).

Compliant mechanisms (CMs) have the same purpose as traditional mechanisms but are monolithic. The movement of these devices is obtained by deforming the material of which they are made. Because the mechanism is built out of one piece, there is no play in its joints, there is no need to assemble different parts of the mechanism, and it does not exhibit abrasion. Because of these benefits, using CMs instead of traditional ones is very interesting for machines making repetitive or precise motions (Howell et al., 2013).

Due to the close relation between structure, stiffness and mechanism motion, manual design of CMs is challenging. A review by Gallego et al. (2010) shows various ways of designing CMs. A

popular design method is topology optimization due to its versatility and ability to generate novel concepts (Bendsøe et al., 2004). This method has a wide design freedom and has low requirements for an initial design, which makes it the choice for the present work. Topology optimization (TO) is a computational method that optimizes material layout in a discretized domain. It is an iterative process, as can be seen in Fig. 1. Each iteration, a structural analysis is performed on the current design to determine its deflection field. From this field, the performance of the design will be derived. Next, the sensitivity of the performance to design changes will be calculated. Finally, a new design will be proposed and a new iteration will start until a convergence criterion is met.

Which kind of structural analysis should be used during TO depends on the deformation of the design. Geometric non-linearity, which occurs when the stiffness of a structure changes as the structure deforms is illustrated in Fig. 2. Here stiffness $k[u]$ is a function of the deflection u . As long as the deflection increases monotonically with load f , the system is considered to be in the mildly non-linear domain. However, at deflection u_1 , maximum load f_{\max} has been reached and the stiffness of the structure becomes negative. This will be called the highly non-linear domain. If the range of motion of a structure is sufficiently small, its deflection can be approximated with linear analysis. However, using linear analysis for large deflections will result in a wrong prediction of its deflection. In CMs made of elastomers, also material non-linearity can become relevant as local strains become large. However, this paper assumes minor strains and a linear isotropic material model.

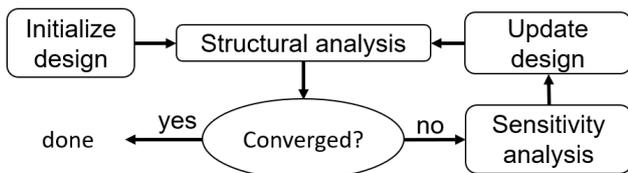


Figure 1: Flowchart of the topology optimization process.

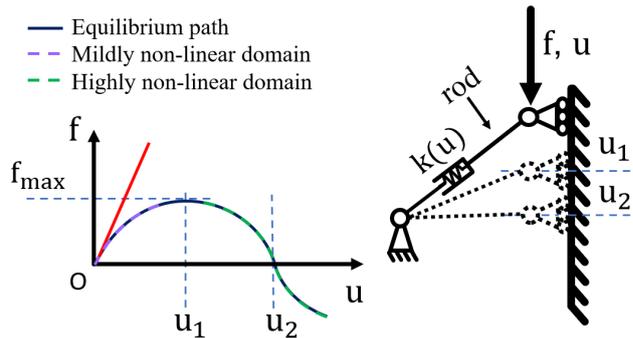


Figure 2: Right: A geometrically non-linear system. Left: Its schematic load-deflection curve, with in red the linear analysis of this system.

1.1 Challenges in topology optimization for compliant mechanisms

Taking geometric non-linearity into account during structural analysis comes with two main challenges. Firstly, for a given load case, a non-linear analysis is an iterative process in which an equilibrium state is sought between the external and internal forces, usually this process is done with the Newton-Raphson process (Borst et al., 2012). This process is computationally expensive and suffers from stability issues for larger deformations as the process can diverge. However, not taking geometric non-linearity into account can lead to designs that perform worse than predicted during the optimization process, as shown by, among others, Buhl et al. (2000).

In density-based topology optimization, to which this paper is limited, designs with a clear boundary between solid and void are preferred, as this is beneficial for production purposes. All elements should thus be either solid or void. Because it is impossible to do a structural analysis of elements with zero stiffness, generally a very low stiffness is assigned to these elements. This makes it possible to perform a structural analysis with a mesh including these elements without changing the solid structure's behaviour significantly. On the other hand, these so-called void elements are prone to invert, making it either impossible to find an equilibrium state for the

structure or adding extra computational effort to the analysis. These void elements thus induce extra numerical instability to geometrically non-linear analysis.

There have been many attempts to reduce computational effort in non-linear analysis and make the process more stable within topology optimization. The most notable methods to reduce computational effort are reduced-order modeling (Gogu, 2015), adaptive meshing (Maute et al., 1995) and reanalysis (Amir, 2015). For stability, the most critical methods rely on changing the constitutive relationship in the analysis ((Pedersen et al., 2001), (Lahuerta et al., 2013), (Bluhm et al., 2021)), ignore the inverted elements ((Buhl et al., 2000), (Saxena et al., 2001)), interpolate between a linear and a non-linear solution (Wang et al., 2014) or change the mathematical set up which describes the analysis (Yoon et al., 2005; Dijk et al., 2014). All these methods overcome element inversion for certain circumstances. However, a general approach that works in all cases has not yet been established.

There is a need for a topology optimization scheme, that does not suffer from the stability issues and computational effort seen when non-linear analysis is used. On top of that, this scheme should, in contrast to a scheme using linear analysis, still create accurate designs, i.e. that perform as analyzed during the optimization process. This would make topology optimization a more feasible and accessible method for applications with moderate displacements, such as CMs.

1.2 Research approach

The present work proposes an approximation of the geometric non-linear load curve in topology optimization. A linear analysis can be seen as a first-order approximation of the load-deflection curve around the undeformed configuration. A non-linear analysis is a more accurate approximation of the load-deflection curve, but only in the points where an equilibrium position has been found. A higher-order approximation

might capture the trend of the geometric non-linear load-deflection curve well enough that it is more useful in topology optimization compared to a first-order approximation (Fig. 3).

This work proposes such an approximation as an extrapolation built with a few equilibrium points close to the undeformed configuration. This approximation will be continuous in the magnitude of the load. A visual interpretation of this idea is given in Fig. 3, which shows the mildly non-linear domain of the curve in Fig. 2. Because these points will be chosen close to the undeformed state, only a few iterations are needed to calculate them, and the chance of instability is low. Using an approximation of the load-deflection curve, from now on called approximated analysis, may therefore be less computationally expensive and more stable than non-linear analysis, whilst still able to produce accurate designs.

There are some advantages and disadvantages of using such an approximation. For example, by not doing a complete geometric non-linear analysis of a particular load case, element inversion and high computational effort are potentially avoided. Extrapolations tend to give a worse prediction of non-linear behaviour the further the points of interest are from the point around which the extrapolation was built. Therefore, this method is not expected to apply to highly non-linear behaviour (Fig. 2). As the curve becomes more non-linear an increasing order of approximation is needed to describe it. Still, for mildly geometric non-linear behaviour, the approximation is expected to be close enough to the equilibrium path, such that when used in topology optimization, accurate designs are created.

The ultimate goal of this research is to establish to what degree this method complies with two criteria:

First, it should create sufficiently accurate designs, i.e. designs that perform close to the result obtained with a full geometrical non-linear analysis.

Second, it should consume substantially less computational effort than current optimization

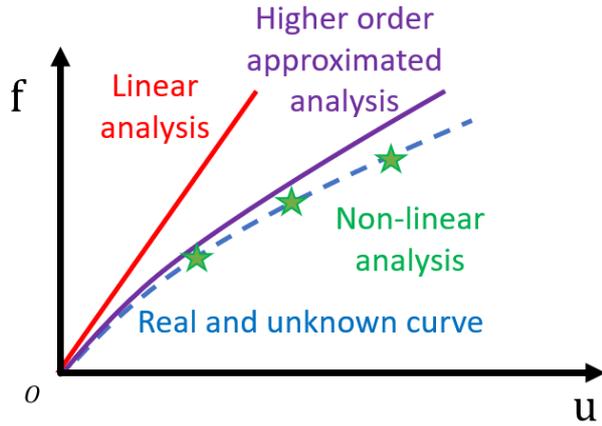


Figure 3: Load-displacement relations as predicted by different kinds of structural analysis.

schemes that use full non-linear analysis, i.e. the computational effort should be closer to that of a scheme using linear analysis than that of one that uses non-linear analysis.

This method will be tested and built with the following scope and boundary conditions in mind:

- The approximated analysis is compared to linear and non-linear analysis.
- The CMs this method produces are assumed to remain in mildly geometric non-linearity.
- For simplicity and characterization of the method, this paper studies 2D cases only.
- The material is assumed to behave linear isotropic.
- All applied forces will be independent from the design variables and deflection field, and boundary conditions will be constant.
- The structural analysis will be done with square quadrilateral elements with bilinear shape functions.
- The non-linear analysis will be limited to the Newton-Raphson method.

The following research questions will be answered to test how well the proposed method works in topology optimization:

1. How well does the approximated deflection curve correspond to the correct deflection curve?

2. What is the influence of the parameters that govern the approximated analysis?
3. Is the gain in computational effort with these approximations significant when compared to full non-linear analysis?
4. What are the implications of this method on the optimization routine regarding its mathematical formulation?
5. What is the range of applicability, in terms of design objectives and load cases that can be used with approximate analysis?

This work seeks answers to these questions in the following way: the next chapter contains the mathematical background of this method, its implementation in topology optimization and the implications on the sensitivity analysis. Next, three numerical examples will be introduced and investigated to study the performance and drawbacks of this method, compared to using linear and non-linear analysis. Finally, this thesis will conclude with an interpretation of the results and a discussion on the research questions.

2 Methods

In this section, first, non-linear and linear analysis are summarized. Approximated analysis is introduced next by deriving it from non-linear analysis, after which it is verified on a beam in bending. For simplicity and its implementation in TO, all structural analysis is performed with the finite element method (FEM). The beam in bending is analyzed with non-linear analysis, linear analysis and approximated analysis, of which the results are compared. After this proof of concept is delivered, topology optimization applications are considered.

2.1 Structural analysis

Any domain can be discretized into elements that are connected by nodes at the corners of the elements (Fig. 4). Each node has two degrees of freedom (DOF), in the x and y direction, respectively, on which an external force could be applied, all these nodal deflections and forces are

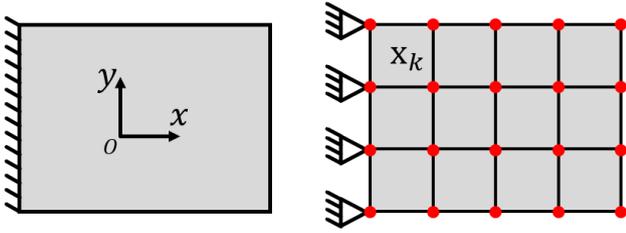


Figure 4: A discretization of a rectangular design domain. The red dots are the nodes and the squares the elements, each with a design variable x_k , stored together in \mathbf{x} . Note that the axis system has nothing to do with the design variables. x is merely an axis direction.

represented in vectors \mathbf{u} and \mathbf{f}^{ext} . All element density design variables are defined by \mathbf{x} .

The objective of doing structural analysis is to find the correct deflections \mathbf{u} for a certain external force vector \mathbf{f}^{ext} . All elements have a certain stiffness, which is dependent on the design \mathbf{x} and for any nonzero deflection field \mathbf{u} , the elements exert internal forces \mathbf{f}^{int} on the DOFs. To obtain the correct deflections \mathbf{u} the structure should be in equilibrium, i.e. the internal forces cancel out the external forces:

$$\mathbf{f}^{\text{ext}} = \mathbf{f}^{\text{int}}[\mathbf{u}, \mathbf{x}]. \quad (1)$$

If the structure is not in equilibrium, there are residual forces on the DOFs, defined as \mathbf{r} :

$$\mathbf{r} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}[\mathbf{u}, \mathbf{x}]. \quad (2)$$

To find an equilibrium position ($\mathbf{r} = 0$), an incremental iterative method is used, which follows the Newton-Raphson method for each increment. Each iteration a small step $\Delta\mathbf{u}$ is added to the total deflections \mathbf{u} , until convergence is met. This is done by building a first order Taylor approximation of Eq. (2) as a function of \mathbf{u} . For this approximation the derivative of the residual to the deflection field is needed:

$$\mathbf{K}_t[\mathbf{u}, \mathbf{x}] = -\frac{\partial \mathbf{r}}{\partial \mathbf{u}}. \quad (3)$$

Finding a solution to \mathbf{K}_t is usually by assembling a tangent stiffness matrix \mathbf{K}_t , instead of actual derivation of the residual. The process

of assembling such a stiffness matrix will not be covered here but can be found in any book on geometric non-linear FEM, like Borst et al. (2012). This matrix \mathbf{K}_t is used to calculate increment $\Delta\mathbf{u}$ in the following way:

$$\mathbf{K}_t[\mathbf{u}, \mathbf{x}]\Delta\mathbf{u} = \mathbf{r}[\mathbf{u}, \mathbf{x}]. \quad (4)$$

In practice the process is stopped if the residual is sufficiently small. To improve stability and convergence, it is common to subdivide the total load that needs to be analyzed into smaller load steps (Borst et al., 2012). One might also be interested in load points before the total load, so sub-dividing into smaller load steps is the way to go. A load step increases the load factor from one equilibrium position to another. In this thesis, the load factor is given by a scalar λ . If a force \mathbf{f}^{ext} is applied to the structure it can be represented by a multiplication between the maximum load $\bar{\mathbf{f}}^{\text{ext}}$ in the structural analysis and λ :

$$\mathbf{f}^{\text{ext}} = \lambda \bar{\mathbf{f}}^{\text{ext}}. \quad (5)$$

The process of finding the equilibrium position for different values of λ is visualized in Fig. 5, which shows the deflections of two arbitrary DOFs as a function of λ . For brevity, this form of structural analysis will be referred to as non-linear analysis.

After a certain point in the load curve of v in Fig. 5 a peak load has been reached. When this happens, the tangent stiffness matrix becomes indefinite, as the stiffness between DOF v and the other DOFs becomes zero. Important to note is that once this happens, the Newton-Raphson process overshoots the peak load as shown for v , after which the solution starts oscillating. There are other schemes than the Newton-Raphson scheme to overcome this problem, but those will not be covered in this work. The interested reader is referred to a book on the non-linear finite element method, like for example Borst et al. (2012).

Linear analysis is a more simplified version of non-linear analysis and can be derived from it. Eq. (4) is linearized in the undeformed configuration to accomplish this. In the undeformed

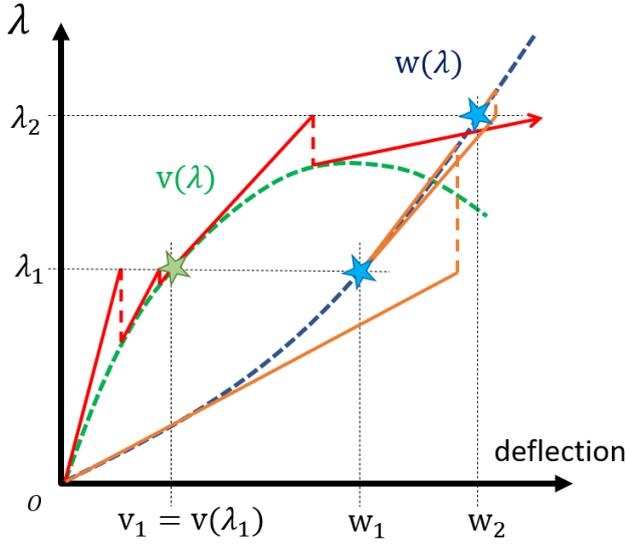


Figure 5: Visualization of the Newton-Raphson process for two arbitrary DOFs. The red and orange lines visualize the increment $\Delta \mathbf{u}$ each iteration, until equilibrium is found.

configuration $\mathbf{u} = \mathbf{0}$ and $\mathbf{f}^{\text{int}} = \mathbf{0}$, and since only one step will be taken $\Delta \mathbf{u} = \mathbf{u}$. \mathbf{K}_t is no longer dependent on the deformation and becomes a linear spring stiffness \mathbf{K} . Finding \mathbf{u} requires only one solution to a linear system of equations:

$$\mathbf{K}[\mathbf{x}]\mathbf{u} = \mathbf{f}^{\text{ext}}. \quad (6)$$

2.2 Approximation of the load-deflection curve

This research approximates deflection curves of structural analysis using a Taylor expansion. Taylor (1715) stated that any analytical function could be represented by a linear combination of all its derivatives:

$$\mathbf{u}[\lambda] = \mathbf{u}[\lambda_a] + \sum_{i=1}^{\infty} \frac{1}{i!} \frac{d^i \mathbf{u}[\lambda_a]}{d\lambda^i} (\lambda - \lambda_a)^i. \quad (7)$$

Eq. (7) considers the Taylor expansion of a vector valued function $\mathbf{u}[\lambda]$, which is continuous in scalar λ . In this function i is the order of derivative and λ_a is a point in λ around which the Taylor expansion is built. If this infinite sum

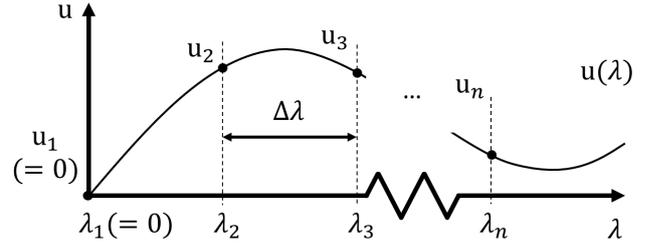


Figure 6: A grid of points for one of the indices of function \mathbf{u} , used to derive finite difference derivatives.

is truncated at a particular order of derivative, the remaining equation is called a Taylor approximation:

$$\tilde{\mathbf{u}}[\lambda] = \mathbf{u}[\lambda_a] + \sum_{i=1}^p \frac{(\lambda - \lambda_a)^i}{i!} \frac{d^i \mathbf{u}[\lambda_a]}{d\lambda^i}. \quad (8)$$

Here $\tilde{\mathbf{u}}$ is the approximation of \mathbf{u} built around $\mathbf{u}[\lambda_a]$ and p is the maximum order of derivative used in this approximation.

For the derivatives needed in this approximation, finite difference equations on equilibrium points $(\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$ on the load-deflection curve are used. These points will be called *foundation points* (FP), as the approximation $\tilde{\mathbf{u}}$ completely depends on those points. The foundation points are obtained by using a Newton-Raphson scheme.

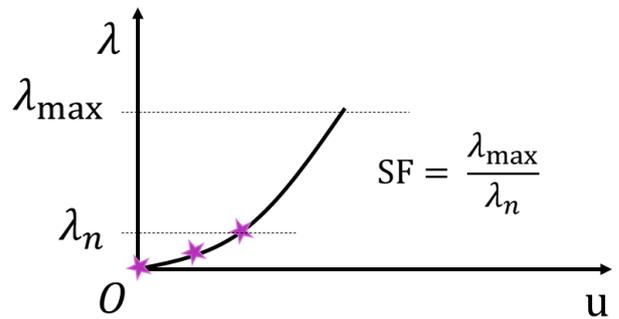


Figure 7: The concept of the scale factor visually explained. For a load case with a maximum load factor λ_{max} and three foundation points, λ_n is the load factor of the furthest foundation point.

The following systematic method is used for the finite difference equations (Chari et al., 2000). Consider Fig. 6, which shows n foundation points, each $\Delta\lambda$ apart, for one of the DOFs in $\mathbf{u}[\lambda]$. With n foundation points, up till order $n - 1$ derivative can be calculated. This derivative of order i , can be written as a linear combination of the foundation points, in which $\mathcal{O}(\Delta\lambda^i)$ is the truncation error:

$$\frac{d^i \mathbf{u}[\lambda_a]}{d\lambda^i} = \frac{1}{(\Delta\lambda)^i} \sum_{j=1}^n \alpha_{ij} \mathbf{u}[\lambda_j] + \mathcal{O}(\Delta\lambda^n). \quad (9)$$

In Eq. (9) there are n unknown variables $(\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in})$. For each foundation point j , one can write the function value of all other points as a Taylor expansion around point λ_j using Eq. (8). These n equations can now be substituted into Eq. (9), which yields a system of equations from which all unknown α 's can be solved (Chari et al., 2000).

Now the approximation of the deflection-load curve used in this thesis to perform approximate analysis can be defined. In this work, the Taylor approximation is built around the undeformed state of a structure ($\lambda_a = 0$), up to the order $n - 1$. Then Eq. (8) becomes:

$$\tilde{\mathbf{u}}[\lambda] = \sum_{i=1}^{n-1} \frac{1}{i!} \frac{d^i \mathbf{u}}{d\lambda^i} \lambda^i. \quad (10)$$

The finite difference equations from Eq. (9) are a linear combination of the foundation points, which when substituted in Eq. (10) transforms into a sum of a sum:

$$\tilde{\mathbf{u}}[\lambda] = \sum_{i=1}^{n-1} \frac{\lambda^i}{i!(\Delta\lambda)^i} \sum_{j=1}^n \alpha_{ij} \mathbf{u}_j. \quad (11)$$

The location of the foundation points are close to the undeformed configuration and are determined by the *scale factor* (SF). This scale factor is the ratio between the maximum load applied to a structure and the load of the furthest foundation point λ_n (Fig. 7), which can be written as a function of $\Delta\lambda$:

$$\text{SF} = \frac{\lambda_{\max}}{(n-1)\Delta\lambda}. \quad (12)$$

Rearranging Eq. (12), such that it is an expression for $\Delta\lambda$ as a function of SF and substitution in Eq. (11) yields:

$$\tilde{\mathbf{u}}[\lambda] = \sum_{i=1}^{n-1} \frac{1}{i!} \left(\frac{\lambda}{\lambda_{\max}} \right)^i ((n-1)\text{SF})^i \sum_{j=1}^n \alpha_{ij} \mathbf{u}_j. \quad (13)$$

To obtain the displacement field at the foundation points, a modified Newton-Raphson scheme is used. This scheme uses approximate analysis on the available foundation points for each load increment to make a first guess of the next \mathbf{u}_j . To clarify this process a small piece of pseudo-code is provided, which performs approximate analysis for a structural analysis problem. In this pseudo-code, it is assumed that there are two functions: NewtonRaphson and TaylorApprox. The input for NewtonRaphson is a load factor λ and $\mathbf{u}_{\text{start}}$, which is the deflection field at which the Newton-Raphson process starts. The output of NewtonRaphson is \mathbf{u} and \mathbf{f}^{ext} for the given λ . TaylorApprox takes in the distance $\Delta\lambda$ and a number of vectors, for which it returns a Taylor approximation up to one order lower than the number of vectors given.

Algorithm 1

A pseudo algorithm for the implementation of approximate analysis in this paper.

```

ustart  $\leftarrow$  0
u1  $\leftarrow$  0
f1ext  $\leftarrow$  0
 $\Delta\lambda \leftarrow \lambda_{\max} \div (\text{SF}(n-1))$ 
for  $j \in [2 \dots n]$  do
     $\lambda_j \leftarrow (j-1) \times \Delta\lambda$ 
    uj, fjext  $\leftarrow$  NewtonRaphson( $\lambda_j$ , ustart)
    ũ[ $\lambda$ ]  $\leftarrow$  TaylorApprox( $\Delta\lambda$ , [u1...uj])
    ustart  $\leftarrow$  ũ[ $j \times \Delta\lambda$ ]
end for
f̃ext[ $\lambda$ ]  $\leftarrow$  TaylorApprox( $\Delta\lambda$ , [f1ext...fnext])
return f̃ext[ $\lambda$ ], ũ[ $\lambda$ ]

```

2.3 Analytical parameter investigation

If Eq. (13) is studied more closely it can be concluded that deflection field $\tilde{\mathbf{u}}$ is a function of n , the foundation points, and the scale factor, continuous in load factor λ :

$$\tilde{\mathbf{u}} = \tilde{\mathbf{u}}[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n, n, \text{SF}, \lambda]. \quad (14)$$

The accuracy $\tilde{\mathbf{u}}$ is determined by the error propagation of two errors: A mathematical error and a numerical error. The mathematical error contains the truncation error in determining the derivatives with finite difference equations and the error of the Taylor approximation at value λ on the load curve. As the accuracy of $\tilde{\mathbf{u}}$ will be compared to non-linear analysis, this error is not covered here. However, an estimate of the mathematical error can be calculated with help of the Eq. (9) and Taylor's inequality as described in for example Stewart (2012). To see whether the numerical error affects $\tilde{\mathbf{u}}$ significantly, it is investigated further.

To keep things orderly, first Eq. (13) is rewritten:

$$\tilde{\mathbf{u}}[\lambda] = \sum_{j=1}^n q_j \mathbf{u}_j \quad (15)$$

with

$$q_j[\lambda] = \sum_{i=1}^{n-1} \frac{1}{i!} \left(\frac{\lambda}{\lambda_{\max}} \right)^i ((n-1)\text{SF})^i \alpha_{ij}. \quad (16)$$

To investigate the numerical error of $\tilde{\mathbf{u}}$, the problem is simplified to a 1 DOF example \tilde{u} dependent on foundation points u_j :

$$\tilde{u}[\lambda] = \sum_{j=1}^n q_j u_j. \quad (17)$$

If \tilde{u} is the sum of quantities, then the errors of those quantities add in quadrature to the error of \tilde{u} (Beek, 2015). That is, if there is a numerical error $\epsilon_{j,\text{num}}$ associated with u_j , the total error ϵ_{tot} of \tilde{u} in Eq. (17) becomes:

$$\epsilon_{\text{tot}}[\lambda] = \sqrt{\sum_{j=1}^n (q_j \epsilon_{j,\text{num}})^2}. \quad (18)$$

Now if $\epsilon_{j,\text{num}} = \epsilon_{\text{num}}$ for all foundation points, Eq. (18) can be written as

$$\epsilon_{\text{tot}}[\lambda] = M_\epsilon \epsilon_{\text{num}}, \quad (19)$$

with M_ϵ being the magnification factor of the numerical error:

$$M_\epsilon[n, \text{SF}] = \sqrt{\sum_{j=1}^n q_j^2}. \quad (20)$$

It can already be concluded that the error will be the largest at $\lambda = \lambda_{\max}$. To see whether this error is significant in comparison with the truncation error, it is calculated for the C-shape example in the next section.

3 Verification

The analysis approach described in the previous section has been implemented in MATLAB, as well as linear and non-linear analysis. Since the non-linear FEM code was self-written, there was need to validate whether it performed correctly. A few analytical tests were done, as well as a comparison with a case solved with Comsol. Those results can be found in Appendix C, which concludes that the code is working adequately.

To test whether approximated analysis can capture part of the geometric behaviour in structural analysis, a minimum working example shown in Appendix B is simulated using linear, non-linear and approximated analysis. This was a two DOF problem that showed that using approximate analysis might be a viable form of structural analysis.

Next, approximate analysis is tested on two simple finite element load cases. First, a simple beam in bending is analyzed to clarify the relation between the accuracy of approximate analysis and the load factor used in the problem. Next, the C-shape problem as proposed by Yoon et al. (2005) is solved using approximate

and non-linear analysis, to show the effect of approximate analysis on element inversion. In addition, the impact of the number of foundation points and scale factor on the accuracy of approximate analysis and the amount of iterations needed is investigated. For both tests, a Young's modulus of 1 kPa is used for solid material and a Poisson ratio of $\nu = 1/3$.

To measure computational effort the amount of linear solves $\#S$ is compared. To measure the accuracy of the different kinds of analysis, the error of the deflection in the x and y direction is compared to non-linear analysis in the following way, where u_x is the deflection of the node in the x direction according to the analysis used and $u_{x,NL}$ the deflection of the same DOF according to non-linear analysis:

$$\epsilon_x[\%] = \left(\frac{u_x}{u_{x,NL}} - 1 \right) \cdot 100\%. \quad (21)$$

The load case and dimensions for the simple beam in bending can be seen in Fig. 8. It is important to note that this example does not try to mimic a real-world beam in bending. On top of that, due to shear locking, non-linear analysis will find an incorrect solution. The idea behind this example is to show how close approximate analysis is to the solution found by non-linear analysis.

The results of this test can be seen in Fig. 8. This figure shows that, in comparison with linear analysis, the deflections according to approximate analysis are a lot closer to the deflections calculated with full non-linear analysis, as long as the deformation stays within the mildly non-linear domain. This is seen even more clearly in Fig. 9, which shows the relation between error in the deflection versus the load factor. After a load factor of $\lambda = 2$ has been reached, the error of approximate analysis increases rapidly.

Element inversion is unlikely to happen while calculating the foundation points, as the loads at the foundation points is low compared to the maximum load in the load case. But the deformation of void elements according to approximate analysis is still important to investigate. Elements that are highly deformed result

in bad sensitivities, which causes designs with unwanted features or causes divergence of the optimization. To investigate the deformation of void elements the C-shape problem as proposed by Yoon et al. (2005) is solved using approximate analysis. The problem consists of a solid C and a void inner section. The ratio in Young's modulus for void and solid elements is 10^{-9}

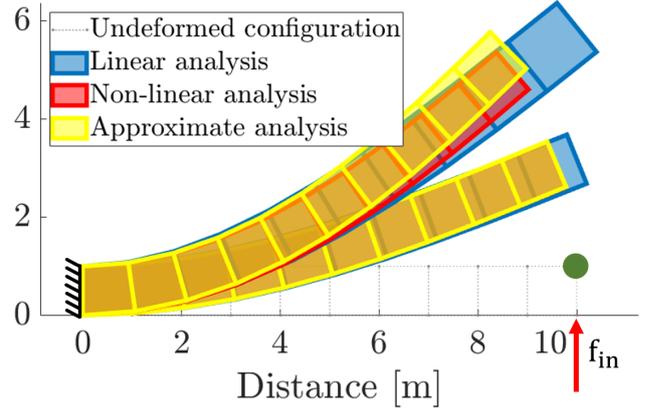


Figure 8: The bending of a horizontal beam according to linear, non-linear and approximate analysis with an applied load of $f_{in} = 1$ N, which corresponds to $\lambda = 1$. On the bottom three beams the load factor $\lambda = 1$, On the top three beams the load factor $\lambda = 2$. The green dot is the node on which the error analysis is done. The approximate analysis is done with $SF = 50$ and $n = 6$ foundation points.

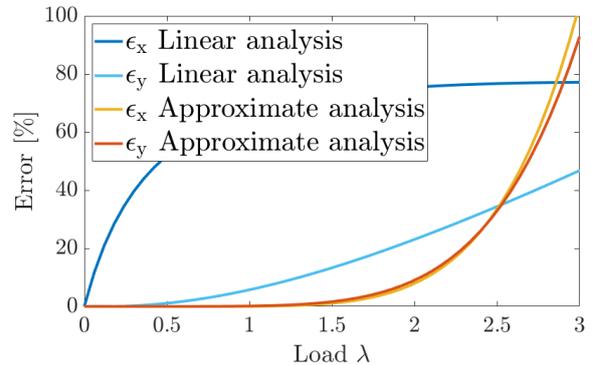


Figure 9: The error of the deflection on the green node in Fig. 8 for linear and approximate analysis, compared tot non-linear analysis.

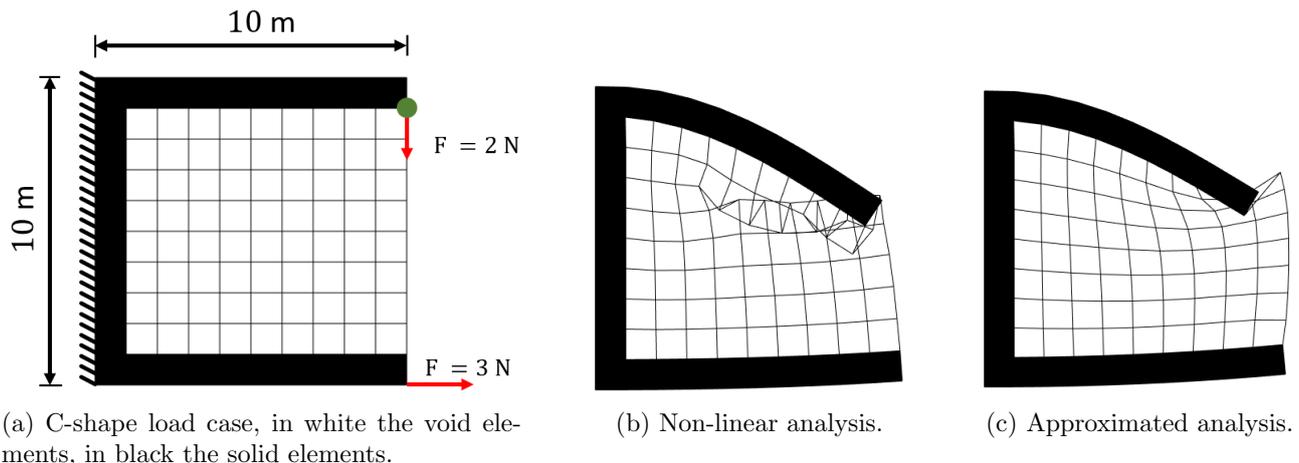


Figure 10: Comparing the solution to the C-shape problem (Yoon et al., 2005) for approximated analysis and non-linear analysis. Approximate analysis is done with $SF = 50$ and 4 foundation points.

Table 1: Error ϵ_x and ϵ_y in approximate analysis for 4, 5 and 6 foundation points and different load factors compared to linear and non-linear analysis.

	$\lambda = 0.5$		$\lambda = 1$		$\lambda = 1.5$	
Analysis	ϵ_x	ϵ_y	ϵ_x	ϵ_y	ϵ_x	ϵ_y
Linear	60.9	9.37	70.2	26.7	72.4	48.3
$n = 4$	4.71	1.18	18.0	11.7	33.0	43.4
$n = 5$	1.64	0.41	20.3	4.37	85.6	13.3
$n = 6$	0.54	0.01	5.28	6.26	12.7	48.8
	#N	#S	#N	#S	#N	#S
Linear	1	1	1	1	1	1
$n = 4$	3	7	3	8	3	9
$n = 5$	4	7	4	8	4	10
$n = 6$	5	8	5	9	5	10
NL	2	17	3	26	9	100

NL: non-linear analysis, Linear: linear analysis, #N: amount of load steps, #S: linear solves.

The load case and deformations can be seen in Fig. 10, for which the results are tabulated in Table 1. Just like the beam in bending, the quality of approximate analysis gets worse if the load factor in Fig. 10 increases. For $\lambda = 1$ the deformation of most elements in the C-shape problem stays within realistic values, whilst the solid part has a smaller error in the deflection than linear analysis. This is however only true for the mildly non-linear domain.

There is a trend in increasing accuracy with the number of foundation points n as shown in Table 1. This comes at the cost of an increased amount of linear solves. To make the comparison in linear solves fair to non-linear analysis, the lowest amount of load steps #N was used, such that the Newton-Raphson scheme did not diverge. For all load levels in this example, the amount of linear solves is significantly lower than for non-linear analysis.

The accuracy of approximate analysis as a function of the scale factor seems to converge to some value, as can be seen in Fig. 11. Scale factors below $SF = 2$ need significantly more linear solves than higher scale factors (the highest #S = 61, which was cut off the graph for readability). As the error converges the number of linear solves does as well for a minimum of #S = 6.

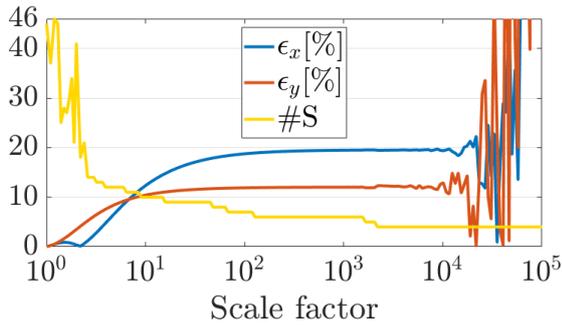


Figure 11: Scale factor sweep for the C-shape problem posed in Fig. 10a with 4 foundation points.

Concerning the numerical error of approximate analysis, Eq. (19) can be investigated, which shows that the numerical error is magnified by factor M_ϵ . A plot showing the relation between the magnification factor and scale factor can be seen in Fig. 12. This line grows exponentially, with the exponent being the maximum order of derivative used in constructing $\tilde{\mathbf{u}}$. For the SF used in Fig. 10c (SF = 50 and $n = 4$), the magnification factor $M_\epsilon = 2.44 \cdot 10^6$.

The value ϵ_{num} can be calculated with use of the tolerance setting $\text{TOL} = 1 \cdot 10^{-12}$ of the Newton-Raphson scheme used to obtain the foundation points \mathbf{u}_j , which converged if $\|\mathbf{r}\|/\|\mathbf{f}^{\text{ext}}\| \leq \text{TOL}$. Now if the error on $\|\mathbf{u}_j\|$ is assumed to be the same order of magnitude and that the numerical error on every DOF has the same value, ϵ_{num} can be calculated in the following way:

$$\epsilon_{\text{num}} \sqrt{\#\text{DOFs}} = \text{TOL}. \quad (22)$$

In the case of the C-shape problem with $\#\text{DOFs} = 121$ this error becomes $\epsilon_{\text{num}} = 1.1 \cdot 10^{-11}$. Substituting ϵ_{num} back into Eq. (19) delivers the absolute numerical error per DOF $\epsilon_{\text{tot}} = 2.68 \times 10^{-5}$ m. This error is insignificant in comparison with the mathematical error values seen in this example. However, it must be noted that for a SF = $1 \cdot 10^3$, the magnification factor is $2 \cdot 10^{10}$, which yields a total error of $\epsilon_{\text{tot}} = 0.22$ m. From that point on the numerical error becomes significant with respect to the

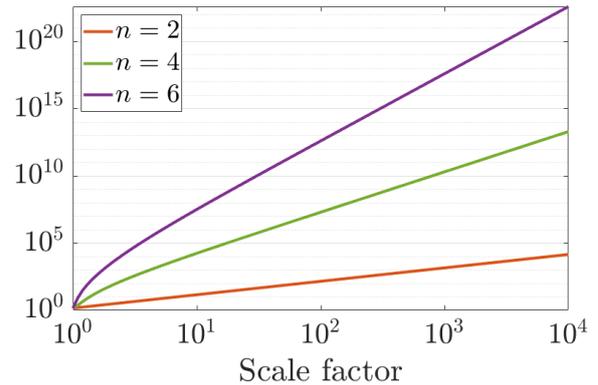


Figure 12: Magnification factor M_ϵ as a function of scale factor for the C-shape problem posed in Fig. 10a with different amounts of foundation points.

deflection of the tip, which might explain the unstable behaviour of the errors above SF = $1 \cdot 10^3$ in Fig. 11.

The amount of foundation points influences the numerical error in the way that the exponential growth of magnification factor M_ϵ is to the power $n - 1$. This means that numerical noise will occur for a lower scale factor than shown in Fig. 11 if the amount of foundation points is increased.

It must be noted that the estimation of ϵ_{num} is quite conservative, as the influence of the tangent stiffness matrix on ϵ_{num} is not accounted for and the assumption that the error is the same for all DOFs is not true. The error in the deflection of solid elements is probably lower as the accuracy of these DOFs influences the norm of the residual more than the accuracy of void deformations. Judging by the noise in Fig. 11, the error starts being significant at SF = $1 \cdot 10^4$, where the numerical error is 5% of the y -deflection. This transfers to an absolute error of $\epsilon_{\text{tot}} = 0.15$ m. Using Eq. (19) for this scale factor and total error, the value of $\epsilon_{\text{num}} = 7.45 \times 10^{-15}$ m, which might be a better estimate.

4 Topology optimization formulation

This section covers the general formulation used for all optimization routines in this paper.

4.1 General optimization setting

In density-based topology optimization the premise that the design domain is discretized into elements that all have a density between fully solid ($\bar{\rho}$) and void (ρ), which is dependent on their design variable x_k :

$$\rho_k = x_k \bar{\rho}. \quad (23)$$

The relation between density and stiffness of the element E_k (Young's modulus) is defined using SIMP interpolation (Bendsøe, 1989). Substituting Eq. (23) for density, the stiffness of the element can be written as a function of the design variables. Here \bar{E} is the Young's modulus of a solid element and p is the penalty factor:

$$E[x_k] = x_k^p \bar{E}. \quad (24)$$

For linear analysis this penalty factor will be $p = 3$ for all cases. For non-linear and approximate analysis of force-based structural problems, this factor will start at $p = 1$ and increase every iteration until a value of $p = 3$, such that no convergence issues of the Newton-Raphson scheme happen in the beginning of the optimization routine. The amount with which p increases is different per numerical example.

The tangential stiffness matrix and internal force vector are assembled out of the individual elemental stiffness matrices $\mathbf{K}_{t,e}$ and elemental internal force vector $\mathbf{f}_e^{\text{int}}$ (Borst et al., 2012). Because the elemental tangential stiffness matrix and internal force vectors are linear in the Young's modulus, they can be written as multiplication of the SIMP function and $\bar{\mathbf{K}}_{t,e}$ or $\bar{\mathbf{f}}_e^{\text{int}}$, the value both attain for a unit Young's modulus:

$$\mathbf{K}_{t,e}[\mathbf{u}, x_k] = E[x_k] \bar{\mathbf{K}}_{t,e}[\mathbf{u}], \quad (25)$$

$$\mathbf{f}_e^{\text{int}}[\mathbf{u}, x_k] = E[x_k] \bar{\mathbf{f}}_e^{\text{int}}[\mathbf{u}]. \quad (26)$$

Because the densities are continuous, a gradient-based optimizer can be used. In this research MMA (Svanberg, 1987) is used. This optimizer strives to minimize an objective f_0 , while conforming to all constraints g_l by optimizing the density field in the given domain:

$$\left. \begin{array}{l} \min_{\mathbf{x}} f_0[\mathbf{x}, \mathbf{u}], \\ \text{subject to } g_l[\mathbf{x}, \mathbf{u}] \leq 0, \\ \text{with } l = 1, \dots, m, \\ 0 < x_{\min} \leq x_k \leq 1 \end{array} \right\}. \quad (27)$$

In order to avoid checker boarding the density filtering method proposed in Andreassen et al. (2010) was used. The filter radius r_f will for all numerical examples be expressed in terms of the side length of the elements. For some examples, additionally, Heaviside projection is used. This is done with a slightly altered version of the projection scheme proposed in Wang et al. (2011). This scheme projects all density values above threshold η to one and all values below to zero. In order to keep the optimization problem continuous a smooth Heaviside function is built around η of which the steepness is defined by β . x_{\min} is added to make sure all values stay above the minimum density value:

$$\bar{x}_k = x_{\min} + (1 - x_{\min}) \cdot \frac{\tanh(\beta\eta) + \tanh(\beta(x_k - \eta))}{\tanh(\beta\eta) + \tanh(\beta(1 - \eta))}. \quad (28)$$

In all examples involving Heaviside projection, an initial value of $\beta = 1$ is used, after which it is multiplied with 1.05 each iteration until a maximum value β_{\max} is reached.

4.2 Sensitivity analysis

All sensitivity analyses in this thesis are performed using the adjoint method (Arora et al., 1979), which makes use of Lagrange multipliers. The general adjoint method for linear and non-linear analysis will not be covered here, but can be found in Appendix A. Instead, focus here is

on the adjoint method for a approximate analysis.

Say, response function f_0 depends on the approximate analysis of the load-deflection curve, $\tilde{\mathbf{u}}$, which is built with n foundation points.

$$\begin{aligned}\tilde{\mathbf{u}} &= \tilde{\mathbf{u}}[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n, \lambda], \\ f_0 &= f_0[\tilde{\mathbf{u}}, \mathbf{x}].\end{aligned}\quad (29)$$

Here $\mathbf{u}_1, \dots, \mathbf{u}_n$ are the foundation points and λ is the load factor. To obtain the sensitivities for response function f_0 , a Lagrangian is built, with n governing equations ($\mathbf{r}_j = 0$) at the foundation points \mathbf{u}_j . This is done by introducing a Lagrange multiplier $\boldsymbol{\mu}_j$ for each governing equation $\mathbf{r}_j = \mathbf{0}$:

$$f_{\mathcal{L}} = f_0 + \sum_{j=1}^n \boldsymbol{\mu}_j^T \mathbf{r}_j[\mathbf{u}_j, \mathbf{x}]. \quad (30)$$

Because of Eq. (29), the chain rule can be used to obtain the sensitivity of $f_{\mathcal{L}}$ to one of the design variables x_k via the foundation points:

$$\begin{aligned}\frac{df_{\mathcal{L}}[\tilde{\mathbf{u}}, \mathbf{x}_k]}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \sum_{j=1}^n \frac{\partial f_0}{\partial \tilde{\mathbf{u}}} \frac{d\tilde{\mathbf{u}}}{d\mathbf{u}_j} \frac{d\mathbf{u}_j}{dx_k} \\ &+ \sum_{j=1}^n \boldsymbol{\mu}_j^T \left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{u}_j} \frac{d\mathbf{u}_j}{dx_k} + \frac{\partial \mathbf{r}_j}{\partial x_k} \right).\end{aligned}\quad (31)$$

Combining the terms behind the summations and expanding \mathbf{r}_j yields:

$$\begin{aligned}\frac{df_{\mathcal{L}}}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \\ &\sum_{j=1}^n \left(\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} \frac{d\tilde{\mathbf{u}}}{d\mathbf{u}_j} \frac{d\mathbf{u}_j}{dx_k} + \boldsymbol{\mu}_j^T \left(\frac{\partial \mathbf{r}_j}{\partial \mathbf{u}_j} \frac{d\mathbf{u}_j}{dx_k} + \frac{\partial \mathbf{r}_j}{\partial x_k} \right) \right).\end{aligned}\quad \text{with} \quad (32)$$

Substituting Eq. (3) into Eq. (32) and collecting terms with state derivatives gives:

$$\begin{aligned}\frac{df_{\mathcal{L}}}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \\ &\sum_{j=1}^n \left(\left(\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} \frac{d\tilde{\mathbf{u}}}{d\mathbf{u}_j} - \boldsymbol{\mu}_j^T \mathbf{K}_{t,j} \right) \frac{d\mathbf{u}_j}{dx_k} + \boldsymbol{\mu}_j^T \frac{\partial \mathbf{r}_j}{\partial x_k} \right).\end{aligned}\quad (33)$$

Now to find sensitivities of any response function dependent on approximate analysis, we need to solve one adjoint equation for each $\boldsymbol{\mu}_j^T$:

$$\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} \frac{d\tilde{\mathbf{u}}}{d\mathbf{u}_j} - \boldsymbol{\mu}_j^T \mathbf{K}_{t,j} = 0, \quad (34)$$

$$\frac{df_{\mathcal{L}}}{dx_k} = \frac{\partial f_0}{\partial x_k} + \sum_{j=1}^n \boldsymbol{\mu}_j^T \frac{\partial \mathbf{r}_j}{\partial x_k}.$$

First the derivative of $\tilde{\mathbf{u}}$ with respect to \mathbf{u}_j needs to be calculated. This is done by deriving Eq. (15):

$$\frac{d\tilde{\mathbf{u}}}{d\mathbf{u}_j} = q_j \mathbf{I}, \quad (35)$$

with (repeated for clarity)

$$q_j[\lambda] = \sum_{i=1}^{n-1} \frac{1}{i!} \left(\frac{\lambda}{\lambda_{\max}} \right)^i ((n-1)\text{SF})^i \alpha_{ij}. \quad (36)$$

The partial derivative of the residual (Eq. (2)) yields:

$$\frac{\partial \mathbf{r}_j[\mathbf{x}, \mathbf{u}_j]}{\partial x_k} = - \frac{\partial \mathbf{f}_j^{\text{int}}[\mathbf{x}, \mathbf{u}_j]}{\partial x_k}. \quad (37)$$

After substitution of Eq. (35) and Eq. (37) into Eq. (34) two compact equations remain to represent the sensitivity of any $f_0[\mathbf{x}, \tilde{\mathbf{u}}]$:

$$\frac{df_0[\mathbf{x}, \tilde{\mathbf{u}}]}{dx_k} = \frac{\partial f_0}{\partial x_k} - \sum_{j=1}^n \boldsymbol{\mu}_j^T \frac{\partial \mathbf{f}_j^{\text{int}}}{\partial x_k} \quad (38)$$

$$\mathbf{K}_{t,j} \boldsymbol{\mu}_j = q_j \left(\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} \right)^T. \quad (39)$$

All the sensitivities needed for specific response quantities in the numerical examples can be found in Appendix A and will be skipped in the main text. If there are multiple response functions in an optimization routine, the Cholesky factorization of $\mathbf{K}_{t,j}$ can be reused for the different sensitivities, making the calculation of the sensitivities quite inexpensive.

4.2.1 Finite difference sensitivity analysis

To check the analytical sensitivities, a comparison finite difference sensitivities has been performed. The implementation can be found in Section A.8.

It is already explained in Section 2.3 that the numerical error obtained in calculating the foundation points is magnified by a quantity. This means that an analytical sensitivity that is dependent on $\tilde{\mathbf{u}}$ will also contain this amplified error.

When the numerical error value of the C-shape problem is considered (Section 3), the error of an arbitrary objective function can be estimated, in the case of this example end compliance, which is defined as:

$$f_0 = \mathbf{f}^{\text{ext}\text{T}} \mathbf{u}. \quad (40)$$

For the approximate analysis in Fig. 10c, $f_0 = 6.62\text{J}$. Because ϵ_{tot} is in the order of $\sim 2 \cdot 10^{-5}$ and \mathbf{f}^{ext} contains only 2 non-zero entries (one of $3N$ and one of $2N$) for the free DOFs, the error associated with f_0 will be in the order of $\sim 10^{-4}$. This is not a problem for the response function, as it is not significant in comparison to the value of f_0 . For the analytical sensitivities in this paper, which depend linearly on the response function, this was never a problem as well.

However, for finite difference sensitivity analysis it might be a problem. For finite difference analysis of response function f_0 , a small perturbation in x_k is applied to the design, after which approximate analysis is done to obtain

perturbed response function value f_{per} . Now to obtain sensitivities for a response function, f_{per} is subtracted from f_0 after which it is divided by x_k . Since f_{per} and f_0 are very close to each other, their difference in value might be smaller than the numerical error ϵ_{global} , causing instabilities.

This effect is better shown in Fig. 13. In this figure the sensitivities of response function $f_0 = \mathbf{f}^{\text{ext}\text{T}} \tilde{\mathbf{u}}$ are calculated with finite difference for different scale factors SF. While the deflection of the structure and value of the response functions stay similar throughout the parameter sweep, the value of the finite difference sensitivities start oscillating for higher scale factors (Fig. 13). This is probably caused by the scaling of the numerical noise described above. For smaller steps in Δx_k , the oscillations became more dramatic.

As can be seen in Eq. (19) the error is proportional to q_j , which scales with SF to the power of the highest order of derivative used in the approximate analysis. Therefore, to check the sensitivities for response functions based on approximate analysis, only low scale factors were used ($\text{SF} < 20$). On top of that it must be noted that analytical sensitivities for approximated analysis do not suffer from this difficulty.

4.3 Quantification of the measurements

To measure the accuracy of approximate analysis for the numerical examples a few measures are used. In the cantilever beam and inverter example, the error of the deflection of a certain node is measured using Eq. (21). For those examples the normalized residual is also used as a measure for accuracy. The normalized objective value is used as a measure for performance of the designs. The residual is normalized with the applied force and the objective value is normalized with the performance of the non-linearly-based design $f_{0,\text{NLD}}$, where $f_{0,\text{NL}}$ is the performance of a design according to non-linear analysis:

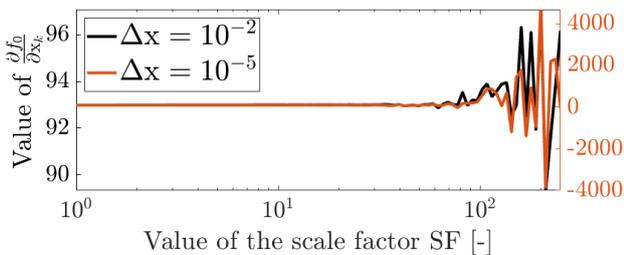


Figure 13: The value of the sensitivity for one element as a function of the scale factor.

$$r_n = \frac{\|\mathbf{r}\|}{f_{\text{in}}}, \quad (41)$$

$$f_n = \frac{f_{0,\text{NL}}}{f_{0,\text{NLD}}}. \quad (42)$$

To quantify the amount of computational effort, the average amount of linear solves per iteration $\#S_a$ is compared between different kinds of analysis, these also include linear solves needed for the sensitivity analysis. Additionally the accuracy in objective function calculation for the approximation-based designs is calculated by comparing them to the objective function value according to non-linear analysis:

$$\epsilon_f[\%] = \left(\frac{f_0}{f_{0,\text{NL}}} - 1 \right) \cdot 100\%. \quad (43)$$

5 Numerical examples

In this section three main numerical examples will be given. Firstly, the end-compliance minimization of a cantilever beam will be considered. Next an inverter design problem will be solved and lastly a flexure with constant stiffness will be designed. The last two examples consider compliant mechanisms, which is what this paper focuses on. The cantilever beam example is here as compliance minimization is an accessible objective within topology optimization.

5.1 Cantilever beam

Optimizing a cantilever beam is a common benchmark problem in topology optimization. The results are well known in literature for topology optimization routines using linear as well as non-linear analysis, which makes this design objective a good candidate to assess the performance of approximate analysis. This work considers a similar case as Buhl et al. (2000).

5.1.1 Objective formulation

The objective in this example is minimization of the end-compliance, with a constraint on the volume $V(\mathbf{x})$ used in the design domain. As can

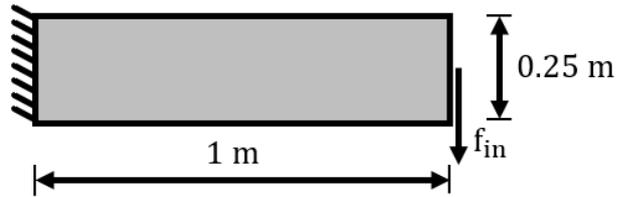


Figure 14: The load case used to optimize cantilever beams.

Table 2: Parameter for the optimization routine described in Section 5.1.1.

E:	3.5 GPa	# elements x :	80	r_f :	2
ν :	0.4	# elements y :	20	β :	-

be seen in Fig. 14, the left side of the domain is fixed, while at the middle of the right edge a force $f_{\text{in}} = 800$ N is applied. The parameters that govern this optimization problem can be found in Table 2. To the penalty factor (Eq. (24)) for non-linear and approximate analysis is increased with 0.05 each iteration until it reaches $p = 3$. The mathematical formulation for this problem is:

$$\left. \begin{array}{l} \min_{\mathbf{x}} f_0 = \mathbf{f}^{\text{extT}} \mathbf{u}, \\ \text{subject to } V(\mathbf{x}) \leq V^* \end{array} \right\}. \quad (44)$$

To quantify the performance and accuracy of approximate analysis for this design objective, a scale factor and foundation point sweep are performed. The approximation-based designs, i.e. designs created in a topology optimization routine using approximate analysis, are then compared to linearly and non-linearly based designs with the measures described in Section 4.3. The error in deflection in the y direction ϵ_y is measured on the node on which f_{in} is applied (Eq. (21)).

5.1.2 Results

A few things can be said about the designs produced by different kinds of analysis seen in Fig. 16. First of all, using linear analysis results in symmetric designs in the horizontal axis, while using non-linear analysis, the designs are

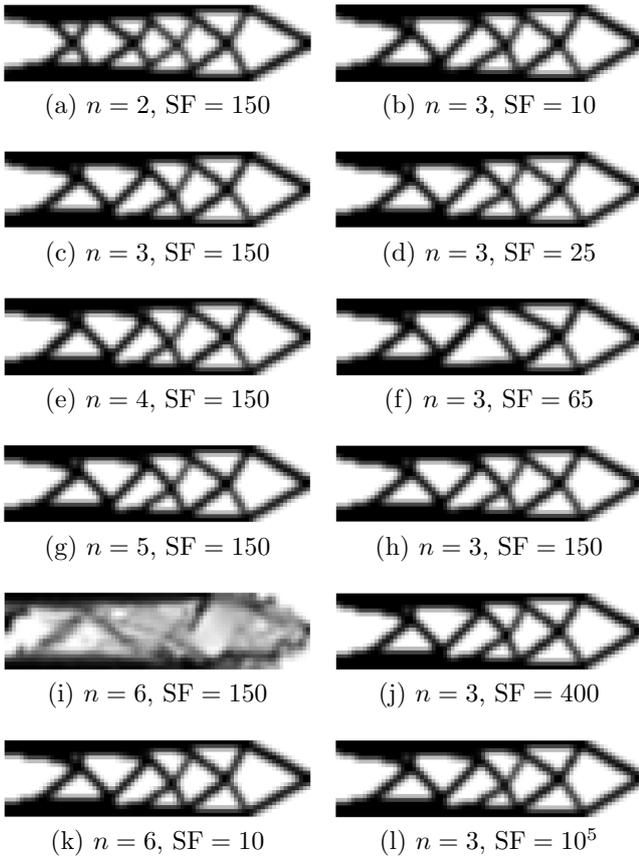


Figure 15: Cantilever designs for the foundation point and scale factor sweep. For $n = 3$, a scale factor $SF = 10^3$ and $SF = 10^4$ lead to the same topology as Fig. 15l

asymmetric. The approximation-based design is not symmetric in the horizontal axis, just like the non-linear one, indicating that geometric non-linearity is taken into account by the optimizer. For the impact of the amount of foundation points and the scale factor, the different designs can be seen in Fig. 15. For most combinations of n and SF the same local minimum is found, with only $(n = 2, SF = 150)$ and $(n = 3, SF = 65)$ being different, of which the last one is the same local minimum as the non-linearly based design.

Concerning the performance, one can take a look at Table 3, which shows the performance of all the designs. There is no clear trend in performance nor accuracy visible when the amount of foundation points of scale factor are varied. The

Table 3: Results for the optimization of cantilever beams for a scale factor and foundation point sweep, compared to the non-linearly based design.

n	SF	f_n	ϵ_f	ϵ_y	r_n	$\#S_a$
2	150	1.03	0.793	0.386	10.2	4
4	150	1.01	0.029	0.013	0.16	7.87
5	150	1.01	0.752	0.819	0.38	9
6	150	1.82	1191	1173	17e3	11.5
6	10	1.01	4e-4	0.001	5e-3	13
3	10	1.01	0.527	0.599	0.36	7
3	25	1.01	0.573	0.654	0.38	7
3	65	1.00	0.585	0.671	0.59	7
3	150	1.01	0.601	0.687	0.40	6.88
3	400	1.01	0.589	0.677	0.40	5
3	1000	1.01	0.6	0.686	0.40	5
	LIN	1.00	0.737	0.479	10.1	1
	NL	1.00	1.00	0	5.52e-13	5.956

NL: non-linearly based design, LIN: linearly based design.

designs perform slightly worse, objective value wise, than linearly and non-linearly based designs. The residual of the approximation-based designs is nearly two orders of magnitude lower than residual for the linearly based design. The amount of linear solves clearly increases with n and decrease with SF . In comparison to non-linear analysis, there is no gain in computational effort for this load case when using approximate analysis.

There are two outliers in the results. The first one being the design in Fig. 15i, for which the optimizer seized to converge. This probably has to do with the numerical error as described in Section 3, which tells us that high scale factors in combination with many foundation points results in highly magnified numerical errors. To overcome this problem with $n = 6$, the scale factor was lowered, leading to the other outlier Fig. 15k, which has an extremely low error for both the deflection and objective value and a

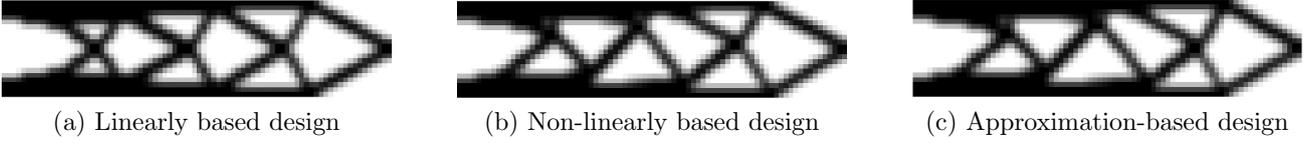


Figure 16: Comparing the optimization of a cantilever beam with for load-factor $\lambda = 1$. The approximate analysis was done with $n = 3$ and $SF = 65$

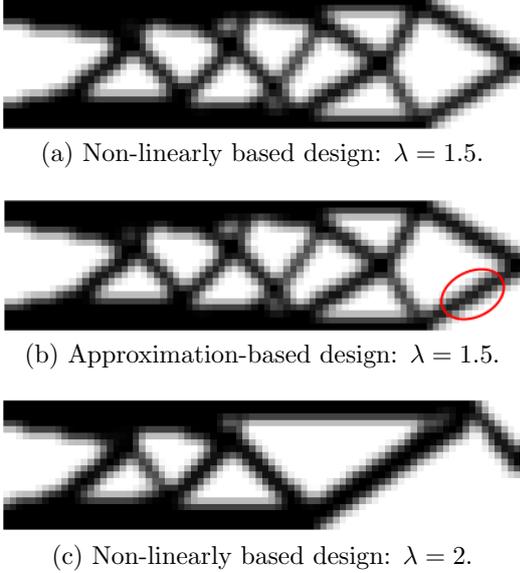


Figure 17: Non-linearly based designs for different λ values and an approximation-based design created with $n = 5$ and $SF = 50$.

normalized residual which was 4 orders of magnitude lower than for the linearly based design.

Table 4: Results for the optimization of cantilever beams for different values of lambda

λ	DesignR	f_n	ϵ_f	ϵ_y	r_n	$\#S_a$
1.5	Fig. 17a	1	0	0	4.80e-11	6.79
1.5	Fig. 16a	0.99	1.63	0.17	15.33	2
1.5	Fig. 17b	1	0.098	0.102	0.1	11
2	Fig. 17c	1	0	0	1.30e-10	149
2	Fig. 16a	1.044	2.813	0.457	20.8425	2

5.1.3 Increasing the load factor

Increasing the load factor of the design problem, makes the structure behave more non-linearly.

To investigate how large the load can be for approximate analysis to be accurate enough to result in well performing designs, the load factor λ was gradually increased. This worked out up until a value of $\lambda = 1.5$.

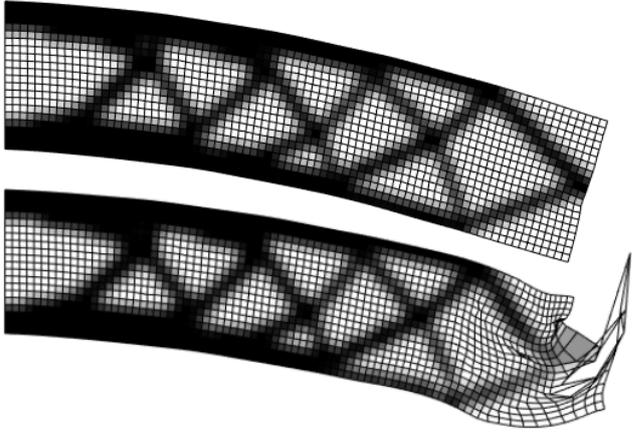


Figure 18: The deformed configuration according to approximate analysis at iteration 200 for $\lambda = 1.6$ (top) and after buckling effects occur at iteration 217 (bottom). With $n = 4$ and $SF = 150$

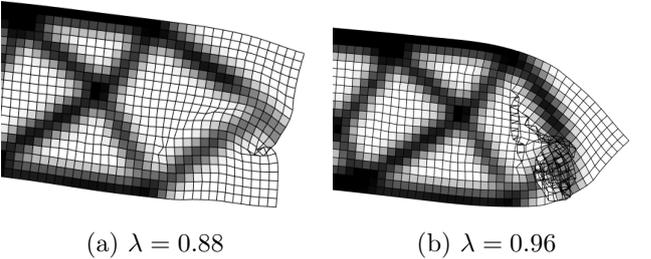


Figure 19: The design of Fig. 18 at 217 iterations, according to non-linear analysis for different load factors λ .

The designs and results for a load factor $\lambda = 1.5$ can be seen in Fig. 17 and Table 4. The

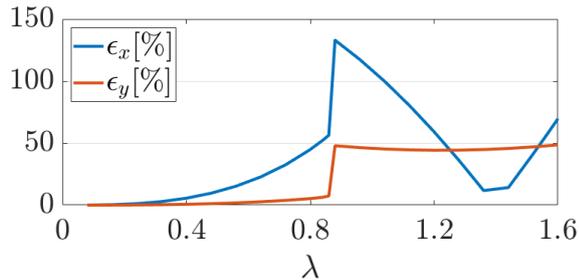


Figure 20: Error in the x and y direction of the deflection for the node at which f_{in} is applied as a function of the load factor.

local minimum found with approximate analysis is the same as the local minimum found with non-linear analysis with a similar performance and a lower error than the linearly based design. For a load of $\lambda = 1.6$ when using approximate analysis, after many design iterations in which the design is the one on top in Fig. 18, a process is started in which the beam with the red circle in Fig. 17b is eroded. This causes the beam to buckle, which can be seen in Fig. 19. If non-linear analysis is used in the topology optimization process, the buckled beam loses its load carrying capability and will disappear, after which a design arises as can be seen in Fig. 17c. This is a process well described in Buhl et al. (2000). In approximate analysis however, this buckling is not captured well as can be seen in the bottom part of Fig. 18, which is confirmed by Fig. 20. These extreme errors in void and tip deformation causes the design variables to diverge. Using approximate analysis thus does not lead to designs for load factors higher than $\lambda = 1.5$.

5.2 Inverter

This numerical example is the first one which considers compliant mechanism design. It is the main example to outline the performance of the approximated analysis within topology optimization. It is used to show the influence of the parameters that govern the approximated analysis, as well as its performance compared to using linear and non-linear analysis.

5.2.1 Design objective

One paper that has investigated the influence of non-linear analysis in inverter design is Pedersen et al. (2001), whose load case is used to investigate the influence of approximate analysis on inverter design. Their load case considers an inverter on the microscale. To make it possible to 3D print the designs in order to check the outcome in a later study, the problem load case and dimensions were scaled using the Buckingham π theorem, as explained in for example (White, 2016), to the specifications seen in Fig. 21 and Table 5. A more thorough explanation of this scaling can be found in Appendix D. The input is constrained at 0.833 mm by means of a spring, such that ratio between the maximum input and the inverter size is the same as in Pedersen et al. (2001). Due to symmetry only the top half is analyzed during structural analysis.

Table 5: Parameter for the optimization routine described in Section 5.2 and shown in Fig. 24

E :	3.5 GPa	# elements x :	200	r_f :	6
ν :	1/3	# elements y :	100	β :	10
k_{out} :	6.94 N m ⁻¹	f_{in} :	4.63 N	η :	0.3

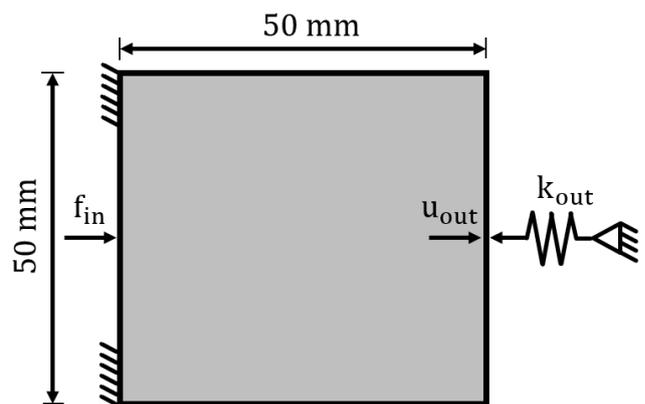


Figure 21: The boundary conditions and load case that all inverters are subjected to. The thickness of the design domain is 10 mm, the other parameters are tabulated in Table 5. This load case corresponds to $\lambda = 1$.

The objective in this optimization routine is the minimization of the (u_{out}) DOF, which

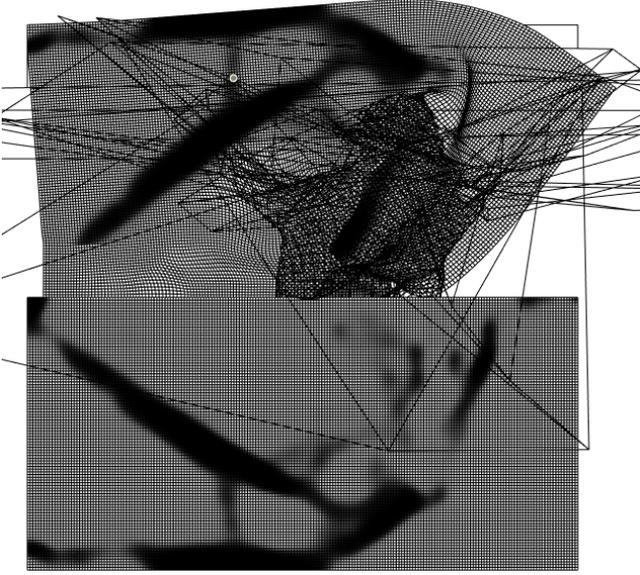


Figure 22: For this approximate analysis $n = 4$ and $SF = 50$ at iteration 58. The bottom half is the undeformed design.

equals a maximization of $(\mathbf{u}_{\text{out}})$ in the negative direction. To this end the deflection field \mathbf{u} is pre-multiplied with a vector \mathbf{l} , which is a vector containing only one 1 at the location of the degree of freedom of interest $(\mathbf{u}_{\text{out}})$:

$$f_0 = \mathbf{l}^T \mathbf{u}. \quad (45)$$

5.2.2 Initial observation

Because approximate analysis is an extrapolation, the optimizer seems to use that to its advantage. Not the actual objective is minimized but the deflection field of the foundation points is optimized to create a large value of \mathbf{u}_{out} , which is unlikely to be correct. This can be seen in Fig. 22, in which it is also shown that the approximation of the voids in the deflection field have expanded so much that they are bigger than the design domain. These unrealistic values cause the optimizer to create topologies with unwanted features like clouds or beams in useless locations. This behaviour is undesired and should be mitigated.

The norms of the residual could be used as a constraint to mitigate the large deflections. The residual is a measure for force imbalance

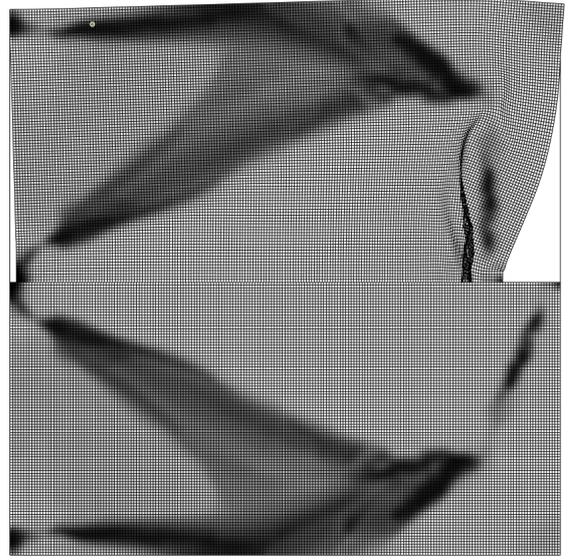


Figure 23: This is a design in which the norm of \mathbf{r}_s is constrained. For this approximate analysis $n = 4$ and $SF = 50$ at iteration 63. The bottom half is the undeformed design.

in a structural analysis and therefore indirectly a measure for accuracy of the displacement field. However, the value difference in $\mathbf{r}[\mathbf{u}, \mathbf{x}]$ on a DOF surrounded by void and one surrounded by material is large. Therefore this constraint would focus on shaping solid material, and most likely not on the deformations in the voids. A slight study to this effect can be found in Chapter 3 of this thesis.

To make sure the deformation of void elements stay realistic as well the values in the residual vector are normalized with respect to the stiffness between the DOFs. This is done by calculating the solid residual $\mathbf{r}_s[\mathbf{u}]$, which is the residual for a structure with the same deflections $\tilde{\mathbf{u}}$, but considering all elements as solid. This way, extreme deformations are penalized, such that the deformation for both void and solid elements stay within realistic bounds:

$$\|\mathbf{r}_s\| \leq r_{s,\text{max}}. \quad (46)$$

In Eq. (46), $r_{s,\text{max}}$ is the maximum value the norm of the solid residual of the design in the deformed state is allowed to attain. The constraint is effective at constraining excessive de-

formations, as can be concluded by comparing Fig. 22 with Fig. 23. Applying this constraint leads to large intermediate density areas in the design domain as can be seen in Fig. 23. To obtain designs without gray areas, the filter radius was increased to 6 element lengths and a Heaviside projection was introduced, for which the maximum value can be found in Table 5. To make the start of the optimization smooth, $2/3$'s are added to the penalty factor in Eq. (24) for non-linear and approximate analysis each iteration until it reaches $p = 3$.

5.2.3 Objective formulation

The constraint on the norm of the solid residual allows the optimizer to converge. This means that the objective can be formulated as follows,

$$\left. \begin{array}{l} \min_{\mathbf{x}} f_0 = \mathbf{I}^T \mathbf{u}, \\ \text{subject to } V(\mathbf{x}) \leq V^*, \\ \|\mathbf{r}_s\| \leq r_{s,\max} \end{array} \right\}, \quad (47)$$

in which the volume is constraint at $V^* = 20\%$ of the design domain. Finding a value for $r_{s,\max}$ is not trivial. The value should be high enough such that not all element deformations are constrained, but low enough such that

there is a low chance of extreme deformations. For all approximation-based inverter designs the value $r_{s,\max} = 1 \times 10^4 \text{ N}$ turned out to work adequately. It must be noted, that the $r_{0\max}$ constraint was only added to optimization routines with approximate analysis, as this constraint is not necessary for linear and non-linear analysis.

In this numerical example, a foundation point, as well as a scale factor sweep are performed. The best performing approximation-based design is discussed in detail. Performance and accuracy of the designs are compared in two ways as described in Section 4.3. The error in objective function evaluation also happens to be the error in deflection estimation in this case.

5.2.4 Results

Based on the deflection fields shown in Fig. 24, in which a linearly, non-linearly and approximation-based design are compared, a few things can be observed. The linearly based design overestimates its true performance as observed earlier by Pedersen et al. (2001). The non-linearly based design performs best of the three. The approximation-based design performs better than the linearly based design, but only 73 % as good as the non-linearly based de-

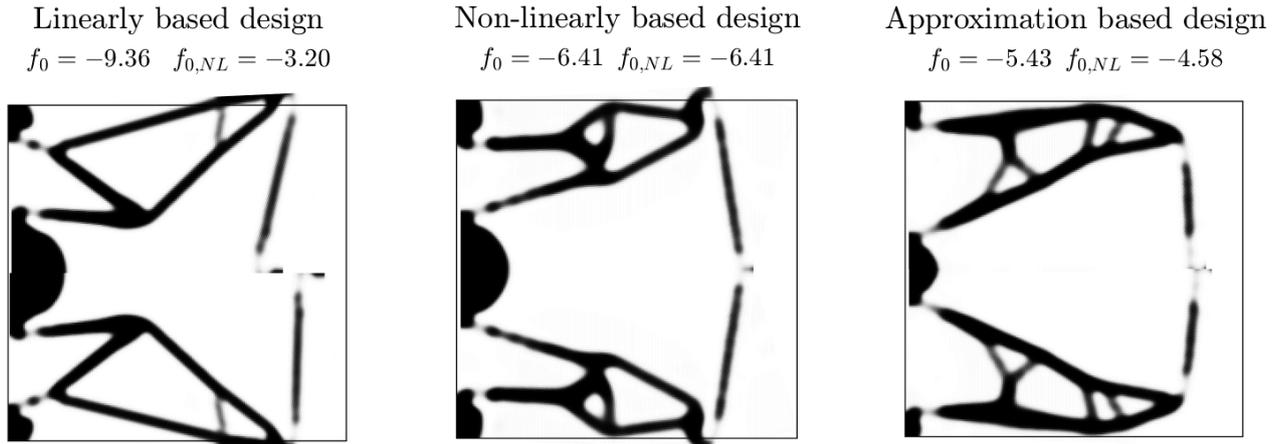


Figure 24: Comparison of the best performing solutions of three optimization routines with different structural analysis. For each design the top half is analyzed the same way as in its optimization routine, the bottom half is analyzed with non-linear analysis. The approximate analysis is done with $n = 6$ and $SF = 10$

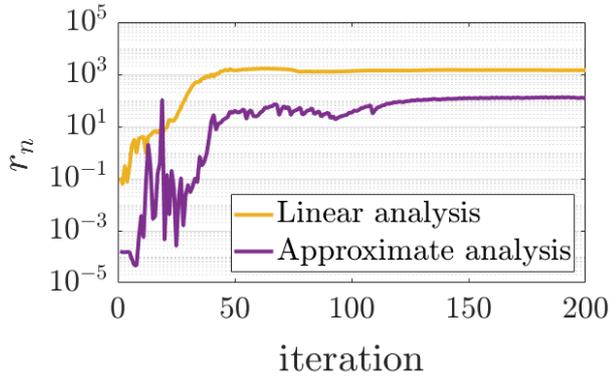


Figure 25: The normalized residual of the designs in Fig. 24 during design iterations on a logarithmic scale.

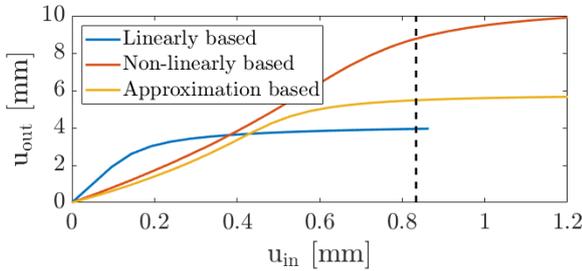


Figure 26: The performance of the differently optimized inverters for a range beyond the value for which they were optimized. All designs are analyzed using non-linear FEM. The dashed black line is the maximum deflection of u_{in} in the design load case.

sign. However, it must be noted that the approximate analysis of the approximation-based design has an error ϵ_f that is an order of magnitude lower than the error of linear analysis of the linearly based design, which makes approximate analysis more accurate than linear analysis. This accuracy is also reflected in the value of the normalized residual, seen in Fig. 25. As the design cycles go on, the normalized residual converges to some value, which for the linearly based design is more than an order of magnitude larger than the approximation-based design.

Fig. 26 shows how the inverters deflect under a displacement-based load case, while the design objective was force-based. Just like in Pedersen et al. (2001), it is clearly seen that the lin-

Table 6: Normalized performance of the inverter designs found in Fig. 24, Fig. 27 and Fig. 28. LIN and NL are the linearly and non-linearly based designs respectively.

n	SF	f_n	ϵ_f	r_n	$\#S_a$
2	10	0.46	165	145	8.52
3	10	0.52	132	23.7	12.2
4	10	0.61	87.1	8.68	15.7
5	10	0.72	38.1	10.7	19.6
6	10	0.73	20.0	12.2	18.3
7	20	0.68	104	66.1	17.2
5	20	0.66	50.7	16.4	17.5
5	40	0.66	56.7	11.6	17.8
5	80	0.65	61.1	10.6	15.6
5	150	0.63	64.1	13.9	14.9
5	250	0.27	54.3	6.42	13.0
LIN		0.50	193	128	2.00
NL		1.00	0.00	7.28×10^{-5}	138

NL: non-linearly based design, LIN: linearly based design.

early based design is optimized for infinitesimal deflection. The non-linearly and approximation-based designs are both optimized for finite range, which is reflected in the steepness of the input output deflection curve in Fig. 26.

Because the norm of the solid residual is constrained, designs with lower deformations overall are preferred, as the value of the solid residual is lower. This might be the reason that the approximation-based designs have the outside pivoting points on the beams connected to u_{out} closer to the center-line of the designs. This location of the pivoting points also result in a lower range of movement for u_{out} and thus lower deformations for all elements.

For different amounts of foundation points, the designs can be seen in Fig. 27. Design wise, for different amounts of foundation points, different local minima are found. Both the performance and the accuracy seem to increase with the amount of n as can be seen in Table 6. The

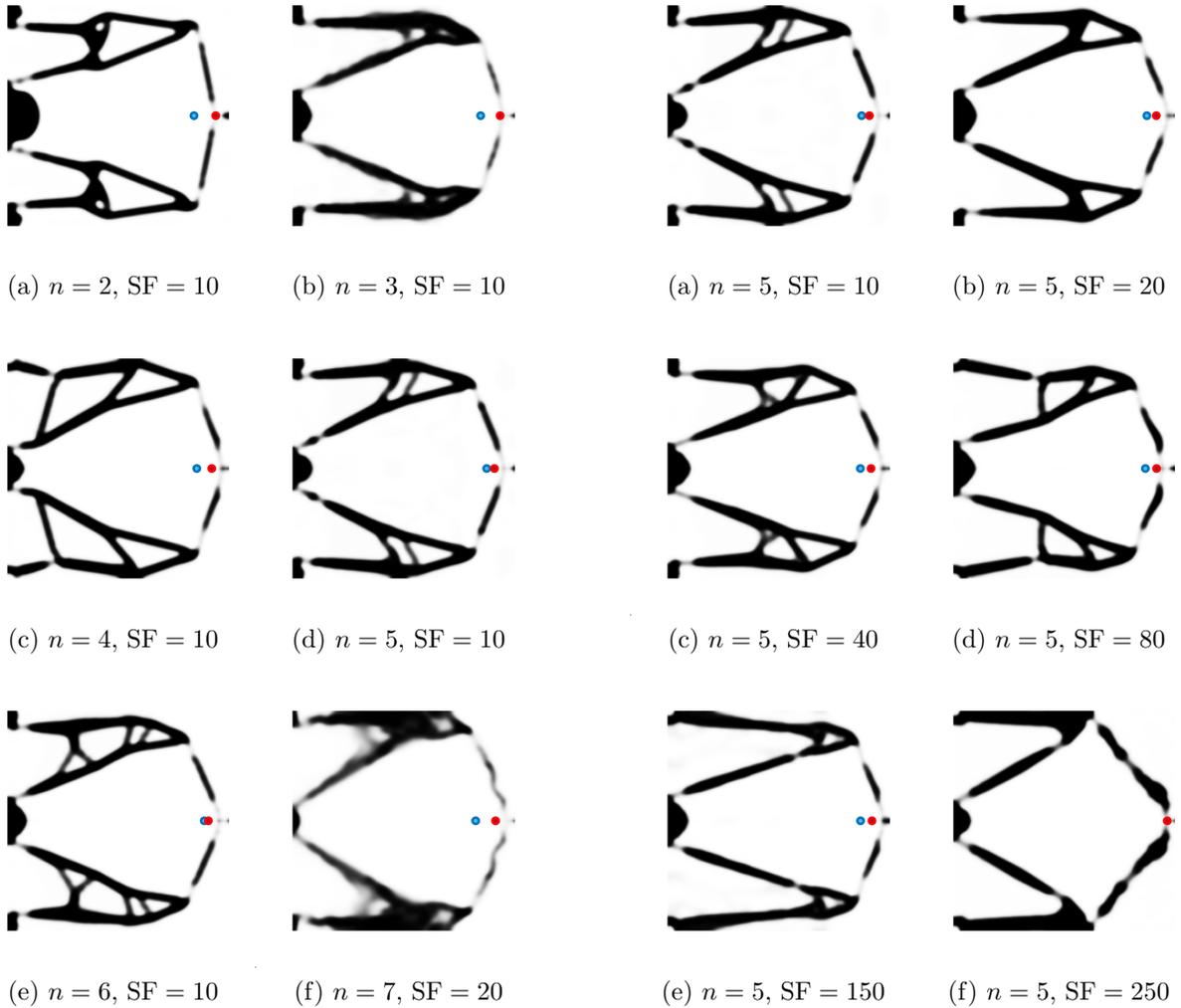


Figure 27: Results for the foundation point sweep. The blue dot is the deflection of \mathbf{u}_{out} according to approximate analysis and the red dot is the deflection according to non-linear analysis.

exception is the design made with 7 foundation points, for which the optimization routine was not very smooth. A design made with 8 foundation points ceased to converge. The norm of the residual also seems to decrease with the amount of foundation points, up until $n = 4$, after which it increases again.

For different scale factors, the designs can be seen in Fig. 28. The designs all perform differently, but there does not seem to be a trend in performance as a function of the scale factor. The optimizer merely finds another local

Figure 28: Results for the scale factor sweep. The blue dot is the deflection of \mathbf{u}_{out} according to approximate analysis and the red dot is the deflection according to non-linear analysis.

minimum. If 5 foundation points are used SFs higher than 250 the optimizer did not converge, which is probably caused by the high value of the numerical noise magnification factor M_ϵ as described in Section 4.2.1.

When computational effort of approximate analysis is concerned Table 6 also gives insight on the amount of linear solves for different combinations of SFs and n . Important to note is that all values in this table include the linear solves needed for the sensitivity analysis as well. More foundation points requires more linear solves to

obtain, as does a lower scale factor. For this problem it can be said that the computational effort to obtain approximation-based designs is an order of magnitude lower than for non-linearly based designs and an order of magnitude larger than for linearly based designs. Usually the solution \mathbf{u} of last iteration can be used as a starting point in non-linear analysis. In this research this was not done, as the inverted elements prohibited faster convergence when this is done. If a method to mitigate inverted elements is used, this could lead to increased performance of non-linear analysis.

Lastly, a note on the stability of this load case. The spring connected to the output DOF u_{out} constrains the movement of that DOF. A stronger spring will make the load case more stable as the value u_{out} can attain goes down. This is beneficial for approximate analysis, as there is less opportunity to misuse the extrapolation and the load case becomes less non-linear in general. In Appendix E, an optimization routine for the same design objective as this section has been performed with an output spring k_{out} that is 5 times stronger. Those results show that an optimizer using approximate analysis is capable of creating designs that perform 97 % as well as non-linearly based designs, with an error in the deflection of $u_{\text{out}} < 1\%$.

5.3 Flexure

An upcoming paper by Koppen et al. (2021) proposes to design flexures, i.e. compliant hinge elements, by making use of end compliance for different load cases. The main idea is to prescribe two displacement-based load cases, one for which the compliance is maximized and the other for which the compliance is constrained. This way the stiffness is maximized for deflection in one direction, while in another direction the design is flexible. As the paper considers only linear analysis, the designs are optimized for their properties in the undeformed configuration, and are only guaranteed to perform for small deflections. To extend this method for flexures experiencing larger deflections, we attempt here to optimize

a flexure to have a constant spring stiffness for a finite deflection.

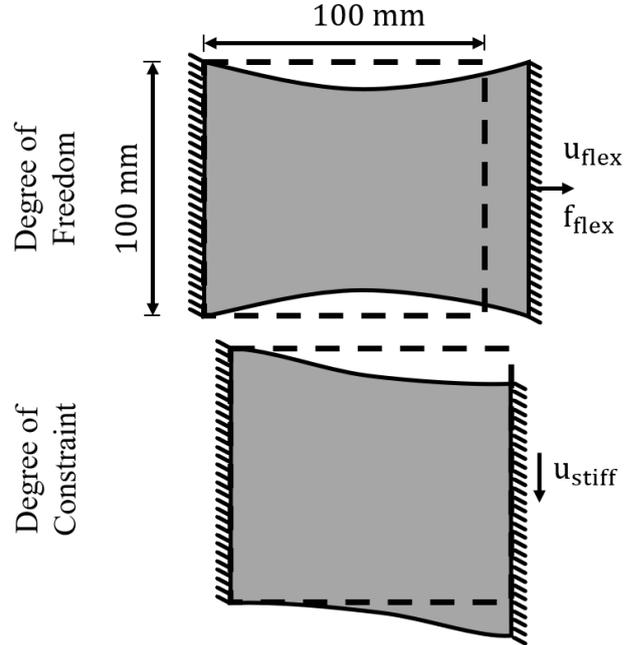


Figure 29: The load case used to optimize flexures, which have a thickness 20 mm. Note that this is a displacement-based load case where f_{flex} is a consequence of the applied displacement.

5.3.1 Objective formulation linear case

First, a flexure is designed with linear analysis in the optimization process. For this the design domain in Fig. 29 is discretized in a 100 x 100 element region. The material is chosen to be PLA (polylactic acid), for which the parameters can be found in Table 7. Other parameters for this optimization routine can be found there as well. For all kinds of analyses the penalty factor in Eq. (24) $p = 3$ from the start.

Table 7: Parameters for the optimization routine of the flexures

E:	3.5 GPa	# elements x :	100	r_f :	2.5
ν :	1/3	# elements y :	100	β :	50
u_{flex} :	10 mm	\mathcal{P} :	15	η :	0.5

Since linear theory is used, only the stiffness for infinitesimal deflections can be influenced. A

maximum value of compliance for the flex deflection ensures a maximum force needed to realize the deflection \mathbf{u}_{flex} . This is done with the following objective formulation:

$$\left. \begin{array}{l} \min_{\mathbf{x}} \quad f_0 = -\mathbf{f}_{\text{stiff}}^{\text{ext}\text{T}} \tilde{\mathbf{u}}_{\text{stiff}}, \\ \text{subject to} \quad \mathbf{f}_{\text{flex}}^{\text{ext}\text{T}} \tilde{\mathbf{u}}_{\text{flex}} \leq c_{\text{max}} \end{array} \right\}, \quad (48)$$

for which c_{max} is set to 100 J.

5.3.2 Objective formulation for approximate analysis

The spring stiffness of a flexure can be used in combination with a deflection to calculate the strain energy. In the non-linear setting, the spring stiffness $k[\mathbf{u}]$ is dependent on the deflection and therefore, to calculate the strain energy, an integral over the deflection should be performed:

$$\mathcal{E}_{\text{strain}} = \int_0^{\mathbf{u}_{\text{max}}} k[\mathbf{u}] \mathbf{u} d\mathbf{u}. \quad (49)$$

Now, if a flexure has a spring stiffness that is constant in, and thus independent of \mathbf{u} , the integral simplifies to:

$$\mathcal{E}_{\text{strain}} = \frac{1}{2} k \mathbf{u}^2. \quad (50)$$

When a structure deflects linearly, the compliance of the structure is equal to the strain energy multiplied with a factor two. Therefore, to optimize a flexure with a constant spring stiffness, the curve of the compliance as a function of the deflection should be optimized to be quadratic.

A quadratic shape of the compliance-deflection curve is achieved by maximizing a response function that is a measure for how well the compliance deflection curve approximates a quadratic curve. This measure is obtained by means of the in-product of two unit vectors, an idea from Maas (2021).

On load factor λ , \mathcal{P} logarithmically distributed precision points are chosen, such that there are more points for low values of the load

factor than for high values. At each precision point the compliance value is evaluated and stored in vector \mathbf{c}_λ . This vector is then pre-multiplied with a vector $\boldsymbol{\gamma}$, which contains the squared values of λ at all precision points. This objective function is then normalized with the norm of both vectors and will be called f_1 :

$$f_1 = \frac{\boldsymbol{\gamma}^{\text{T}} \mathbf{c}_\lambda}{\|\mathbf{c}_\lambda\| \|\boldsymbol{\gamma}\|}, \quad (51)$$

with

$$\mathbf{c}_\lambda = \begin{bmatrix} \mathbf{f}_1^{\text{T}} \tilde{\mathbf{u}}_1 \\ \mathbf{f}_2^{\text{T}} \tilde{\mathbf{u}}_2 \\ \vdots \\ \mathbf{f}_{\mathcal{P}}^{\text{T}} \tilde{\mathbf{u}}_{\mathcal{P}} \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \vdots \\ \lambda_{\mathcal{P}}^2 \end{bmatrix}. \quad (52)$$

This new response function f_1 is 1 when the two vectors $\boldsymbol{\gamma}$ and \mathbf{c}_λ are parallel to each other and 0 when they are orthogonal.

This measure for the shape of the compliance curve is now combined with the original objective formulation Eq. (48) in the following way:

$$\left. \begin{array}{l} \min_{\mathbf{x}} \quad f_0 = -f_1 f_2, \\ \text{subject to} \quad \mathbf{f}_{\text{flex}}^{\text{ext}\text{T}} \tilde{\mathbf{u}}_{\text{flex}} \leq c_{\text{max}} \end{array} \right\}, \quad (53)$$

with

$$f_2 = \mathbf{f}_{\text{stiff}}^{\text{ext}\text{T}} \tilde{\mathbf{u}}_{\text{stiff}}. \quad (54)$$

Because the objective is now a combination of two functions, the chain rule should be used for obtaining the sensitivities:

$$\frac{df_0}{dx_k} = \frac{df_1}{dx_k} f_2 + f_1 \frac{df_2}{dx_k}. \quad (55)$$

The full derivation the sensitivity analysis of function f_1 and f_2 can be found in Appendix A. This kind of objective formulation was found to produce structures with intermediate density elements. As this is not preferred, a Heaviside projection scheme is applied following the density filter. This ensures a design with little intermediate densities.

Table 8: Results for the optimization of flexures

n	SF	f_1 :	ϵ_f :	$\#S_a$
4	10	0.998	0.209	15.735
4	50	0.997	0.347	13.563
4	300	0.999	0.127	12.739
4	10^3	0.998	0.187	11
4	10^4	0.999	0.001	10.994
3	50	0.997	0.293	9.884
5	50	0.998	0.181	16.393
6	50	1.000	0.001	20.723
7	50	0.999	0.044	24.561
linearly based design		0.994	0.645	2

5.3.3 Results

To verify whether the approximation-based designs have a constant stiffness, they are compared to a linearly based design, optimized for the same flexibility. To do this comparison f_1 is calculated with non-linear analysis for all designs. At the same time the accuracy of approximate analysis was measured by comparing the value of f_1 according to approximate and non-linear analysis, by means of Eq. (43). As a measure for computational effort the average amount of linear solves $\#S_a$ is compared per design.

All designs can be seen in Fig. 30. Their corresponding results are tabulated in Table 8. The first thing to notice is how close all f_1 values are to one. Apparently this response function has a domain very close to 1, which might make it a not so ideal response function to evaluate the shape of the compliance load curve. The value for the linearly based design is the lowest and its error the largest. It must be noted that these numbers are hard to interpret as their differences are very small. Therefore, a load deflection curve was calculated for an approximation and linearly based design. This curve can be seen in Fig. 31, which shows a severe stiffening effect for the linearly based design, which is absent for the approximation-based design. The deflection corresponding to Fig. 31 can be seen

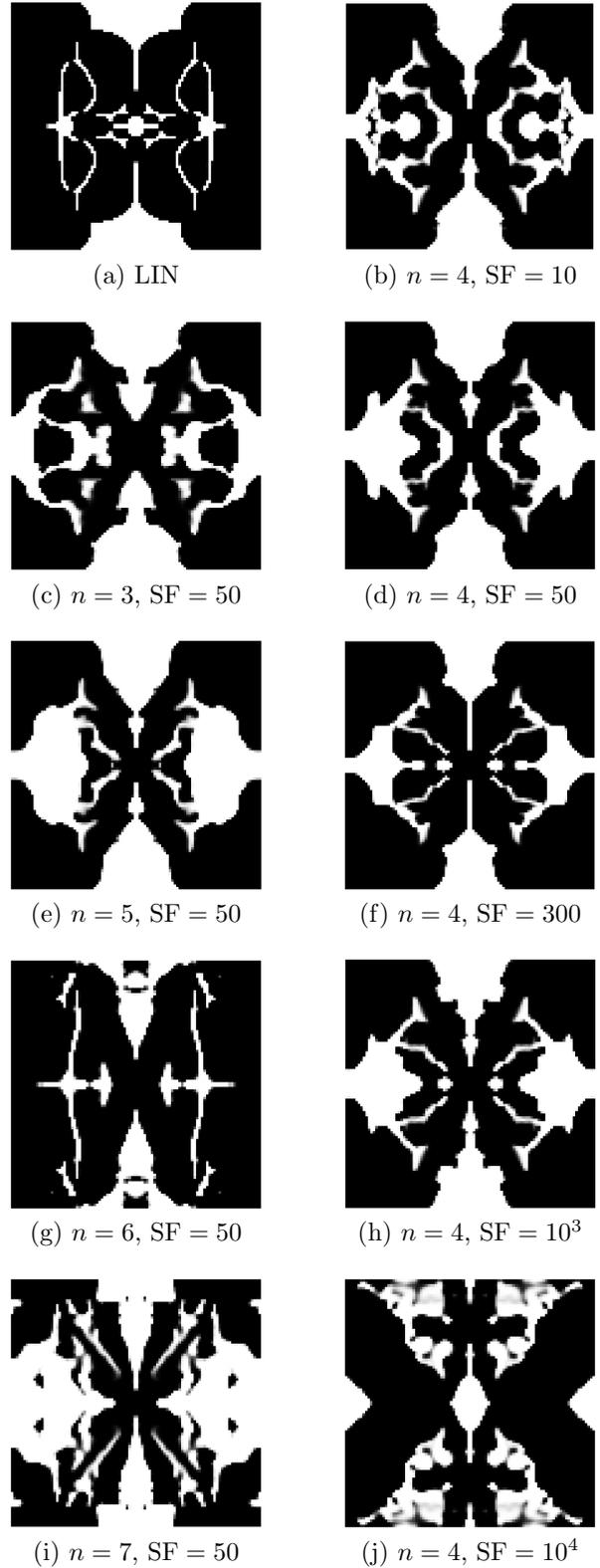


Figure 30: Flexures for a foundation point and scale factor sweep. LIN: linearly based design.

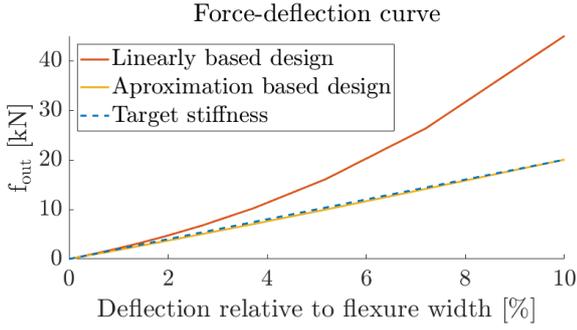


Figure 31: The performance of both flexure designs for the optimized deflection.

in Fig. 32.

Computational effort wise, using approximate analysis is up to one order of magnitude more expensive than linear analysis. This has also to do with the fact that in the approximate analysis one extra response function is added, for which there are as many extra linear solves as foundation points minus one and the fact that the sensitivities for compliance are self adjoint for linear analysis. Doing a non-linear analysis however would require finding 15 equilibrium points on the deflection load curve, and an additional 15 linear solves when concerning the sensitivity analysis for all precision points. The non-linear evaluation in Fig. 32 should give an indication of the computational effort. When the sensitivity analysis is included the amount of linear solves comes down to $\#S = 94$. This was deemed too much computational effort and was therefore not used as a comparison.

For this design objective, the scale factor and foundation point sweep resulted in many different local minima found by the optimizer. There is no clear trend in the quality of the designs in the scale factor sweep, only less computational effort is noticed for higher scale factors. More foundation points lead to better performing designs which have less of an error when evaluating f_1 . However, this comes at a higher computational cost. Outliers are the design for $n = 7$ and the one with $SF = 10^4$, which were both the maximum for which the optimizer was able to converge. The designs are a bit messy, which

probably comes due to the magnified numerical error.

Because the load cases in this design objective are displacement-based, there is less opportunity for the optimizer to misuse approximate analysis to create unrealistically performing designs. This has a few added benefits. In this case there is no need for response function to mitigate misuse as seen in the inverter load case. On top of that, the optimizer is still able to converge for very high scale factors, as the numerical noise has less of an impact on the convergence.

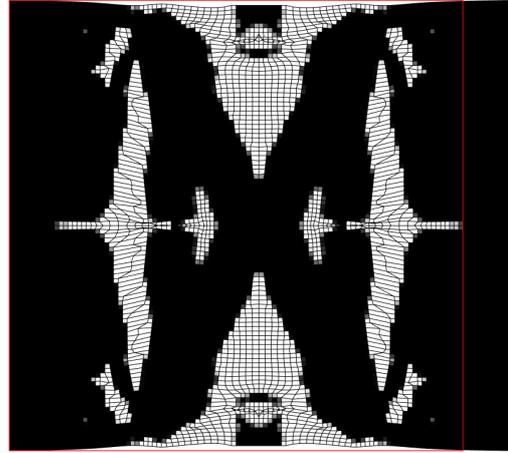


Figure 32: The deflection of the design in Fig. 30g. The red box indicates the original size of the flexure. The non-linear analysis for this flexure took $\#S = 79$.

6 Discussion

This work presents a novel way of structural analysis, used in topology optimization for compliant mechanisms that exhibit geometrical non-linear behaviour. Instead of a full non-linear analysis, the load-deflection curve is approximated by means of a Taylor approximation around the origin. To build this Taylor approximation, a few equilibrium points, close to the origin are calculated. The response functions are then defined in terms of the design variables and the approximated load-deflection curve.

This section will first discuss approximate analysis in general, based on the verification of

the method done in Section 3 and the results of the numerical examples obtained in Section 5, after which future research is discussed.

6.1 Characteristics of approximate analysis

To test how this approximated load-deflection curve performs in a topology optimization scheme, a few benchmark problems were studied. The three numerical examples contain different load cases and response functions in order to explore the behaviour of approximate analysis. The research questions stated in the introductions will now be discussed.

Regarding questions 1 and 2, the accuracy of approximate analysis depends on the non-linearity of the load case and the number of foundation points and value of the scale factor. The simple beam in bending and C-shape problem in Section 3 show that for larger loads, approximate analysis becomes less accurate. But then in the optimizations of a cantilever beam, the error in accuracy is less than 1% up until buckling effects occur. The buckling effects induce high errors for certain DOFs, causing divergence of the optimizer. From this, we can conclude that the accuracy of approximate analysis depends not so much on the severity of geometric non-linear effects in general, but more on the non-linearity of the load deflection curve of the individual DOFs.

For all cases where an error in deflection was measured it was noticed that the error decreased if the number of foundation points increased. For the scale factor, the story is a bit more nuanced, as the C-shape problem shows that the error in deflection converges for higher scale factors. This means that increasing the scale factor leads to a worse accuracy of approximate analysis, but only up until the error has converged.

When compared to linear analysis, approximate analysis has the potential to be much more accurate. This was seen in the accuracy of approximate analysis in the beam in bending and the c-shape problem in Section 3, as well as in the inverter example. This potential is only available in the mildly non-linear domain. The

verification examples showed that when the load is increased, there is a point, where the error of approximate analysis crosses the error made by linear analysis. The inverter example showed that using approximate analysis in topology optimization leads to designs that are more accurate than when linear analysis is used. The error of the output DOF was decreased up to an order of magnitude in comparison to linear analysis. This is because geometrical non-linearities are taken into account in approximate analysis.

Regarding question 3: the computational effort of linear, approximate and non-linear analysis are all an order of magnitude apart for the inverter load case. For the optimization of a cantilever beam, the amounts of solves were similar for non-linear and approximate analysis, up until buckling effects occur. At that point, approximate analysis diverges, while the amount of solves needed for non-linear analysis increases tenfold. For the optimization of a flexure, the number of linear solves needed in non-linear analysis can only be guessed, but as described in Section 5.3, would also be close to an order of magnitude more than for approximate analysis. In absolute measures, this means that approximate analysis is much closer to linear analysis, computational effort wise, than to non-linear analysis. This makes it appealing for design problems where computational capacity is limited. If more precision is needed, approximate analysis could be used to create an initial design, after which non-linear analysis takes over in the optimization routine.

The scale factor and amount of foundation points influence the number of linear solves needed to complete the optimization scheme and the quality of the approximation. More foundation points and lower scale factors lead to more accurate designs and higher computational costs. The more geometric non-linear behaviour is present in a load-deflection curve, the higher the order of derivative should be in order to approximate the curve well, which must be taken into account when deciding on the number of foundation points. In the end, the choice of n and SF comes down to a trade-off between qual-

ity in approximation and computational effort.

Regarding question 4, the implementation and complexity of the approximated load-deflection curve, the following can be said: The mathematical description of a topology optimization routine with approximate analysis is slightly more complex than one with full non-linear analysis. However, because the approximation is based on foundation points that are found with full non-linear analysis, all existing mathematical solutions for non-linear analysis can be used to integrate approximate analysis in topology optimization. As long as response function formulations are defined as functions of the load-deflection curve and quantities that depend on that curve, they can be translated into the foundation points. On top of that, the approximated load-deflection curve is continuous in the load factor, making it possible to define new kinds of response functions containing the load factor, like for example the derivative of $\tilde{\mathbf{u}}$ to λ .

Considering question 5, there are for now only three design objectives to compare. The three design objectives differ in two main points: the load cases and response functions. The first two examples both consider a force-based load case. Force-based load cases have more of a tendency for the DOFs to diverge. This was seen in both examples, where the error increases rapidly when the load crossed a certain value. The cantilever beam and the in inverter differ in response function. Because in the inverter example 1 DOF is maximized, there is more opportunity for the optimization process to misuse the error produced by approximate analysis. When this misuse happens, a way of mitigated is necessary, which in the present work was done by constraining $\|\mathbf{r}_0\|$. For the cantilever beam, the misuse was less of a problem, as the response function has no benefit by maximizing the error made by approximate analysis. The cantilever design objective only diverged if the load case induced buckling effects in the structure, showcasing that approximate analysis does not catch such non-linearities well.

It must be noted that the intention was

to use approximate analysis for design objectives that consider compliant mechanisms in the mildly non-linear domain. The optimization of the flexure is the perfect example that does justice to approximate analysis. The load cases are displacement-based, which makes the structural analysis of the structure a more stable process in general, and the response functions have no gain by misuse of the error produced by approximate analysis. The flexure produced with approximate analysis is clearly designed for a finite range of motion, showing a linear relation in the load-deflection curve. Using this kind of design objective is perfectly suited for approximate analysis.

A final word will be devoted to numerical stability. The inversion of elements seen in non-linear analysis and the occasional divergence of the incremental iterative schemes used are quite a hurdle when implementing non-linear analysis in topology optimization. The loads seen when determining the foundation points, granted scale factors are not close to 1, are so low that inversion of elements and other bifurcation points are rarely seen. This makes approximate analysis a much more stable process.

6.2 Future research

When considering the approximation of the structural behaviour in this work, a few things could be interesting to investigate further.

- The current code makes use of double-precision variables, it might be interesting to use variables with a higher precision to make it possible to make approximations based on more foundation points. Of course, the convergence criterion for the Newton-Raphson scheme should then also be set to a tighter tolerance. This would come at a higher computational effort.
- The application point of the approximated load curve could be chosen differently. For example, the approximation could be built halfway the full load, such that the approx-

imation is a better representation of the load-deflection curve.

- To avoid multiple linear solves, reanalysis could be used to find the foundation points, needed to build the approximated curve.
- It might be interesting to determine the number of foundation points on the flow, by evaluating the accuracy of the load-deflection curve continuously while building the approximation, this could be done by means of the norm of the residual for example.

Now from a more general perspective, a few other things might be interesting to research:

- The Taylor approximation is now based on finite difference equations in \mathbf{u} , but could also be made with finite difference equations in \mathbf{K}_t . This way the order of approximation can be one higher than the number of foundation points. Doing so would require some extra thought on what this would mean for the sensitivities.
- Other ways of approximating the load-deflection curve might be interesting. Taylor approximations are very general, an approximation better suited for structural analysis might perform better. A polynomial fit or function specifically designed for structural analysis for example.
- Newton-Raphson makes a first-order Taylor expansion, using approximate analysis by means of a modified Newton-Raphson scheme might accelerate the process of non-linear analysis.

7 Conclusion

In topology optimization for compliant mechanisms with finite range, geometric non-linear deflections are important to take into account. Instead of doing a full non-linear analysis, this re-

search opts to approximate the non-linear load-deflection curve, based on equilibrium points close to the undeformed configuration. This so-called approximate analysis is integrated into the topology optimization process via these equilibrium points, making it possible to use the available mathematical formulas for non-linear analysis.

The numerical examples indicate that when approximate analysis is used in topology optimization, designs can be created that outperform designs made with linear analysis. Depending on the load case and response functions, the result can be as good as when non-linear analysis is used. The computational effort is reduced an order of magnitude compared to non-linear analysis, with the added benefit of numerical stability in obtaining the equilibrium points.

The results for the optimization of an inverter are promising and show the added benefit of approximate analysis for compliant mechanism design. The most interesting result is the one for the optimization of a flexure, which showed that it is possible to efficiently optimize for a desired or undesired linear behaviour of the designs.

In practice compliant mechanisms show geometric non-linear behaviour and taking that into account is an absolute design requirement. Approximate analysis has shown that it can attain sufficient accuracy to create designs exhibiting mildly non-linear behaviour.

This new point of view on structural analysis within topology optimization opens a lot of doors for new research. The hope of the authors of this paper is that this method makes including geometric non-linearity into a topology optimization routine more accessible and thus will induce interesting new topology optimization schemes.

References

- Amir, O. (Jan. 2015). “Revisiting approximate reanalysis in topology optimization: on the advantages of recycled preconditioning in a minimum weight procedure”. In: *Structural and Multidisciplinary Optimization* 51.1, pp. 41–57. ISSN: 16151488. DOI: [10.1007/s00158-014-1098-7](https://doi.org/10.1007/s00158-014-1098-7).
- Andreassen, E., A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund (Nov. 2010). “Efficient topology optimization in MATLAB using 88 lines of code”. In: *Structural and Multidisciplinary Optimization* 2010 43:1 43.1, pp. 1–16. ISSN: 1615-1488. DOI: [10.1007/S00158-010-0594-7](https://doi.org/10.1007/S00158-010-0594-7).
- Arora, J. S. and E. J. Haug (Sept. 1979). “Methods of Design Sensitivity Analysis in Structural Optimization”. In: *AIAA Journal* 17.9, pp. 970–974. ISSN: 0001-1452. DOI: [10.2514/3.61260](https://doi.org/10.2514/3.61260).
- Beek, A. (2015). *Advanced engineering design : lifetime performance and reliability*. Sixth edit. Delft: TU Delft. ISBN: 9789081040617.
- Bendsøe, M. P. (Dec. 1989). “Optimal shape design as a material distribution problem”. In: *Structural Optimization* 1.4, pp. 193–202. ISSN: 09344373. DOI: [10.1007/BF01650949](https://doi.org/10.1007/BF01650949).
- Bendsøe, M. P. and O. Sigmund (2004). *Topology Optimization*. Springer Berlin Heidelberg. DOI: [10.1007/978-3-662-05086-6](https://doi.org/10.1007/978-3-662-05086-6).
- Bluhm, G. L., O. Sigmund, and K. Poulios (Mar. 2021). “Internal contact modeling for finite strain topology optimization”. In: *Computational Mechanics* 2021 67:4 67.4, pp. 1099–1114. ISSN: 1432-0924. DOI: [10.1007/S00466-021-01974-X](https://doi.org/10.1007/S00466-021-01974-X).
- Borst, R. d., M. A. Crisfield, J. J. Remmers, and C. V. Verhoosel (Sept. 2012). *Non-Linear FEA Analysis of Solids and Structures*. Wiley. ISBN: 978-0-470-66644-9.
- Buhl, T., C. B. W. Pedersen, and O. Sigmund (2000). “Stiffness design of geometrically nonlinear structures using topology optimization”. In: *Structural and Multidisciplinary Optimization* 19.2, pp. 93–104. ISSN: 1615147X. DOI: [10.1007/s001580050089](https://doi.org/10.1007/s001580050089).
- Chari, M. and S. Salon (2000). *Numerical Methods in Electromagnetism*. Elsevier, pp. 105–141. DOI: [10.1016/b978-0-12-615760-4.x5000-7](https://doi.org/10.1016/b978-0-12-615760-4.x5000-7).
- Dijk, N. P. van, M. Langelaar, and F. van Keulen (Aug. 2014). “Element deformation scaling for robust geometrically nonlinear analyses in topology optimization”. In: *Structural and Multidisciplinary Optimization* 50.4, pp. 537–560. ISSN: 16151488. DOI: [10.1007/s00158-014-1145-4](https://doi.org/10.1007/s00158-014-1145-4).
- Gallego, J. A. and J. Herder (July 2010). “Synthesis Methods in Compliant Mechanisms: An Overview”. In: *Proceedings of the ASME Design Engineering Technical Conference* 7, pp. 193–214. DOI: [10.1115/DETC2009-86845](https://doi.org/10.1115/DETC2009-86845).
- Gogu, C. (Jan. 2015). “Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction”. In: *International Journal for Numerical Methods in Engineering* 101.4, pp. 281–304. ISSN: 1097-0207. DOI: [10.1002/NME.4797](https://doi.org/10.1002/NME.4797).
- Howell, L. L., S. P. Magleby, and B. M. Olsen (Feb. 2013). *Handbook of Compliant Mechanisms*. Ed. by L. L. Howell, S. P. Magleby, and B. M. Olsen. Wiley, pp. 7–14. ISBN: 9781119953456. DOI: [10.1002/9781118516485](https://doi.org/10.1002/9781118516485).
- Koppen, S., M. Langelaar, and F. Keulen (2021). “A simple and versatile formulation for topology optimization of flexures”. In: *Mechanism and Machine Theory* (In review).
- Lahuerta, R. D., E. T. Simões, E. M. Campello, P. M. Pimenta, and E. C. Silva (2013). “Towards the stabilization of the low density elements in topology optimization with large deformation”. In: *Computational Mechanics* 52.4, pp. 779–797. ISSN: 01787675. DOI: [10.1007/s00466-013-0843-x](https://doi.org/10.1007/s00466-013-0843-x).
- Maas, J. (2021). “The topology optimization of a compliant variable-camber morphing wing”. PhD thesis. TU Delft.

- Maute, K. and E. Ramm (Oct. 1995). “Adaptive topology optimization”. In: *Structural optimization 1995 10:2* 10.2, pp. 100–112. ISSN: 1615-1488. DOI: [10.1007/BF01743537](https://doi.org/10.1007/BF01743537).
- Pedersen, C., T. Buhl, and O. Sigmund (Apr. 2001). “Topology synthesis of large-displacement compliant mechanisms”. In: *International Journal for Numerical Methods in Engineering* 50.12, pp. 2683–2705. ISSN: 00295981. DOI: [10.1002/nme.148](https://doi.org/10.1002/nme.148).
- Saxena, A. and G. K. Ananthasuresh (2001). “Topology synthesis of compliant mechanisms for nonlinear force–deflection and curved path specifications”. In: *Journal of Mechanical Design, Transactions of the ASME* 123.1, pp. 33–42. ISSN: 10500472. DOI: [10.1115/1.1333096](https://doi.org/10.1115/1.1333096).
- Stewart, J. (2012). *Early Transcendentals Metric International Version*. Brooks/Cole, pp. 755–757. ISBN: 9780538498876.
- Svanberg, K. (Feb. 1987). “The method of moving asymptotes—a new method for structural optimization”. In: *International Journal for Numerical Methods in Engineering* 24.2, pp. 359–373. ISSN: 10970207. DOI: [10.1002/nme.1620240207](https://doi.org/10.1002/nme.1620240207).
- Taylor, B. (1715). *Methodus incrementorum directa & inversa*. Londini.
- Wang, F., B. Lazarov, and O. Sigmund (2011). “On projection methods, convergence and robust formulations in topology optimization”. In: *Structural and Multidisciplinary Optimization* 43.6, pp. 767–784. ISSN: 1615147X. DOI: [10.1007/s00158-010-0602-y](https://doi.org/10.1007/s00158-010-0602-y).
- Wang, F., B. S. Lazarov, O. Sigmund, and J. S. Jensen (July 2014). “Interpolation scheme for fictitious domain techniques and topology optimization of finite strain elastic problems”. In: *Computer Methods in Applied Mechanics and Engineering* 276, pp. 453–472. ISSN: 00457825. DOI: [10.1016/j.cma.2014.03.021](https://doi.org/10.1016/j.cma.2014.03.021).
- White, F. m. (2016). *Fluid Mechanics*. Mc Graw Hill Education, pp. 266–273. ISBN: 978-9-814-72017-5.
- Yoon, G. H. and Y. Y. Kim (Apr. 2005). “Element connectivity parameterization for topology optimization of geometrically nonlinear structures”. In: *International Journal of Solids and Structures* 42.7, pp. 1983–2009. ISSN: 00207683. DOI: [10.1016/j.ijsolstr.2004.09.005](https://doi.org/10.1016/j.ijsolstr.2004.09.005).

3

Additional response functions

There are some extra response functions, for which there was not enough time to describe them in detail, but which are interesting to discuss. Therefore, this chapter will show what the idea behind those response functions is and what meaningful research has been done already. Three extra response functions are interesting enough to note here: the second derivative with respect to the load curve, the norm of the residual and the minimization of reaction forces in a structure.

3.1. Derivatives with respect to the load factor

As approximate analysis is a function of λ , it allows for response functions to be continuous in λ as well. This property could be used in response functions. One example is using derivatives of $\tilde{\mathbf{u}}$ in λ as response functions, for example:

$$f_0 = \frac{d^2 \tilde{\mathbf{u}}}{d\lambda^2} \ell, \quad (3.1)$$

which is only possible because $\tilde{\mathbf{u}}[\lambda]$ is a continuous function of λ . The sensitivity analysis for this response can be found in Appendix A.

This kind of response function might be interesting when a linear relation between input and output is desired. One could, for example, in the inverter design problem, minimize all derivatives except the first one to obtain a linear relationship between the input force and output displacement. If a linear relation between input displacement and output displacement is desired, the load case should just simply be changed to be displacement-based.

3.2. Norm of the residual

The norm of the residual for a certain deformation of a structure is a measure for force imbalance in that structure. This measure is also an indirect measure for how accurate the deflection field is for a given load. It might be interesting to use this as a response function for deflection fields calculated by linear and approximate analysis. A clearer explanation of this response function is seen in Fig. 3.1.

In this figure, λ is the load factor and λ_{\max} is the maximum load factor for the current structural analysis. The deflection fields \mathbf{u}_1 and $\tilde{\mathbf{u}}$ are the linear and the approximate analysis of the real deflection field (the deflection field for which the system is in equilibrium). \mathbf{r} is the residual for a certain deflection field \mathbf{u} and density \mathbf{x} , for example:

$$\begin{aligned} \mathbf{r}_{\text{lin}} &= \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}_{\text{lin}}, \mathbf{x}), \\ \mathbf{r}_{\text{app}} &= \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\tilde{\mathbf{u}}, \mathbf{x}), \end{aligned} \quad (3.2)$$

where \mathbf{f}^{ext} and \mathbf{f}^{int} are the external and internal force vectors respectively. The sensitivities for this response function can also be found in Appendix A.

These responses lead to large gray areas as can be seen in Fig. 3.2. This is probably caused by the fact that the internal forces are highest in elements with a high stiffness. This causes the residual

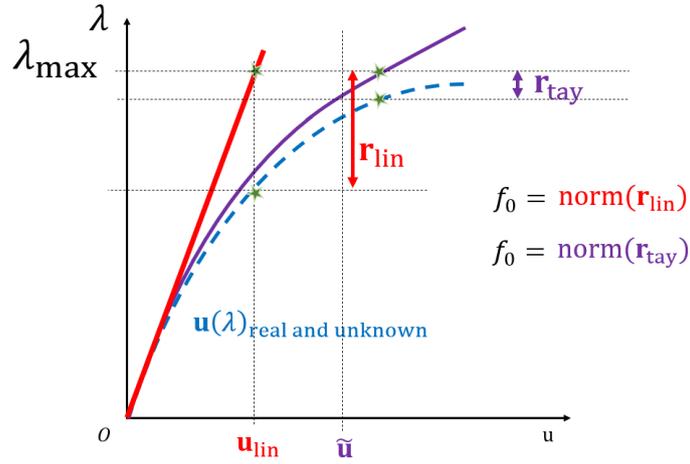


Figure 3.1: Two response functions for linear and approximate analysis, that concern the norm of the residual.

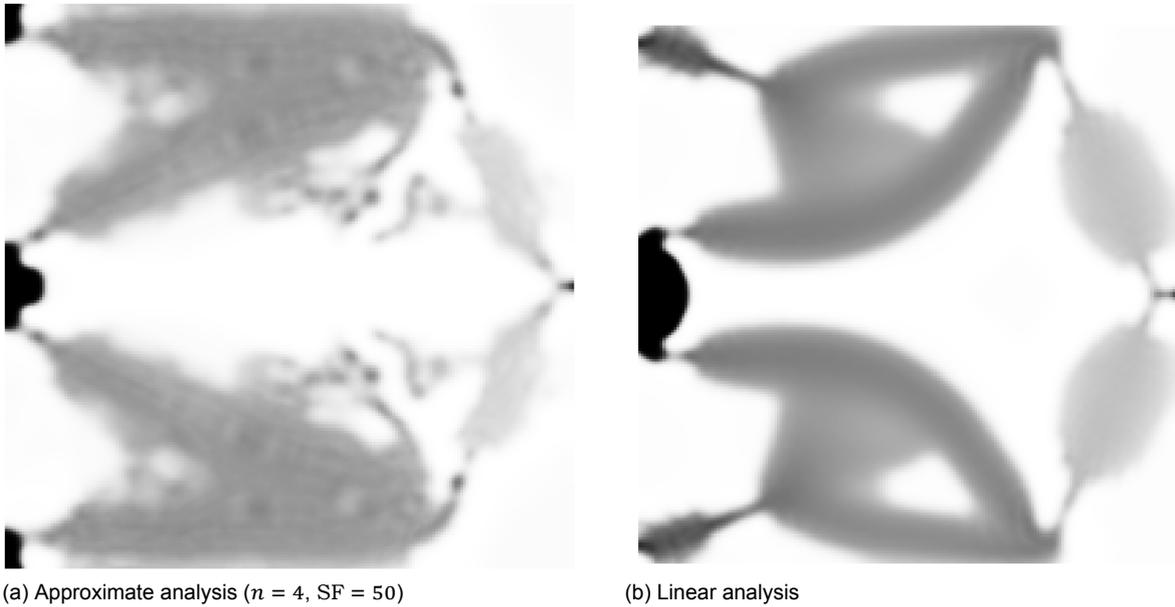


Figure 3.2: Iteration 85 of the inverter optimization process for two optimization processes, with a constraint on the norm of the residual for a maximum of 10 N. The design objective and settings are the same as in Section 5.2 of the paper, except for the output spring, which now has a value of $k_{out} = 13.8 \text{ mN m}^{-1}$. These designs did not cross the constraint boundary.

to be larger in the DOFs of that element. Decreasing the stiffness in these elements leads to a lower value of the residual. Large gray beams are thus created to obtain a high objective value while keeping the residual low. A Heaviside projection scheme was added to try and solve the problem, but did not give promising results. Another solution to mitigate this effect was not found.

3.3. Reaction forces

From Koppen et al. (2021) it is known that, in flexure design, a load case like in Fig. 3.3 will result in two horizontal beams. This allows for a downward motion and high stiffness in the stiff direction. This is sub-optimal, as the right edge of the design domain moves inward, as the flexure deflects.

One idea to mitigate this inward motion is by constraining the horizontal reaction forces on the right edge of this flexure. The main idea is that if there are no horizontal forces on the right edge, which is constrained in the horizontal direction, it will move straight down in an unconstrained situation.

To fulfill this idea, the reaction force at certain DOFs can be used in a response function. In this

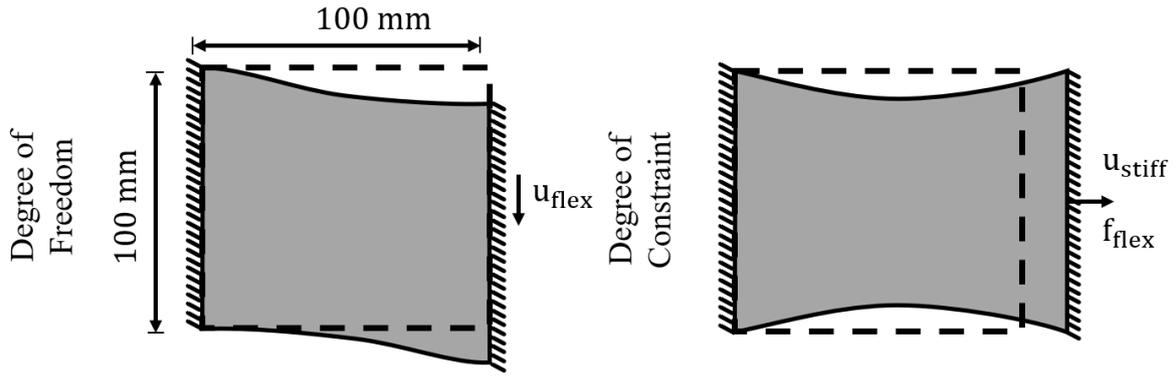


Figure 3.3: The load case for a horizontal moving flexure.

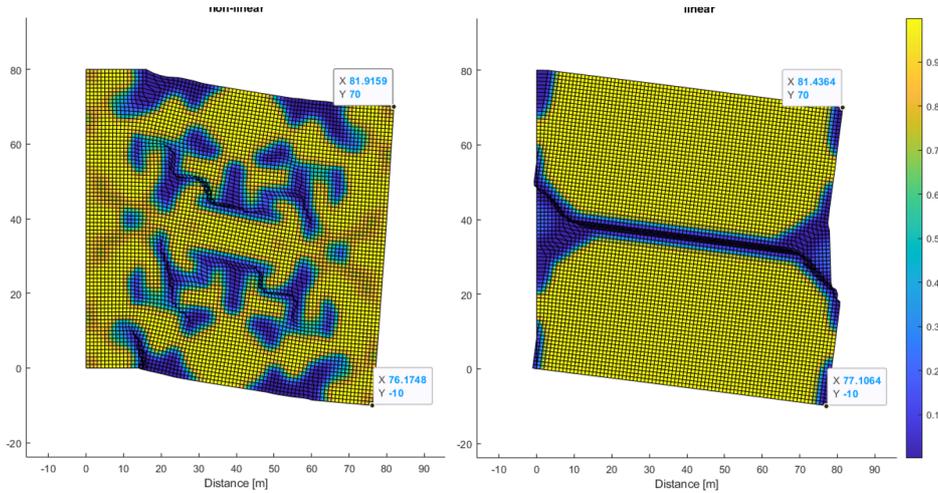


Figure 3.4: Deflection field of a flexure design in which the free direction is downward. The left flexure is optimized by minimizing the horizontal reaction forces, the right one is the classic formulation. The right edge of the flexure is not vertical in the deflected form and thus does not work adequately.

small experiment, it was opted to use the norm of the vector containing the horizontal reaction forces at DOFs of preference, such that some reaction forces can be excluded. To obtain this response function, first, a clear definition of reaction forces should be given, in this case for linear analysis:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad (3.3)$$

which can be partitioned, in free f and prescribed p DOFs:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fp} \\ \mathbf{K}_{pf} & \mathbf{K}_{pp} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_p \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_p \end{bmatrix}, \quad (3.4)$$

from which the following relations can be derived:

$$\mathbf{K}_{ff}\mathbf{u}_f = \mathbf{f}_f - \mathbf{K}_{fp}\mathbf{u}_p, \quad (3.5)$$

and

$$\mathbf{f}_p = \mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p. \quad (3.6)$$

From these, a response function can be constructed to fulfill the requirements described above. In this response function, \mathbf{A} is a diagonal matrix, with ones on the DOFs that need to be included in the analysis:

$$f_{\text{react}} = \sqrt{\mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}. \quad (3.7)$$

This response function is a measure of the reaction forces in the design domain. The matrix \mathbf{A} allows for picking the preferred DOFs, which makes it possible to only measure the horizontal reaction forces on the right edge of the design domain. The sensitivity analysis of this response function is quite complex and can be found in Appendix A.

The implementation of this response function in an optimization routine happens as follows:

$$\left. \begin{array}{l} \min_{\mathbf{x}} \quad f_0 = -\mathbf{f}_{\text{stiff}}^{\text{ext}\top} \tilde{\mathbf{u}}_{\text{stiff}}, \\ \text{subject to} \quad \mathbf{f}_{\text{flex}}^{\text{ext}\top} \tilde{\mathbf{u}}_{\text{flex}} \leq C_{\text{max}} \\ \quad \quad \quad f_{\text{react}} \leq f_{\text{react,max}} \end{array} \right\}, \quad (3.8)$$

in which $f_{\text{react,max}}$ was set to a low value as to minimize the reaction forces.

This did not work out but did give a nice topology as can be seen in the left figure of Fig. 3.4. This is probably caused by the fact that the reaction forces are not equal to 0, but a very low value. If someone were to extend this research, it is advised to minimize those reaction forces.

4

Discussion

This chapter is supplementary to the discussion in the main paper and discusses approximate analysis from a broader perspective. First, the method will be discussed, after which some response functions will be discussed as well.

4.1. Approximate analysis

In this thesis, an approximation is made on basis of equilibrium points on the load curve. It might be interesting to build this approximation in a different way. The mathematics in the finite element method deliberately skip terms with second-order information in finding the general equations for a Newton update. If there is a way to calculate this second-order derivative in the initial configuration, a second-order step $\Delta \mathbf{u}$ could be taken, which might accelerate the Newton process. It might also make it possible to do an approximate analysis without obtaining equilibrium points with an incremental iterative scheme.

Material non-linearities might also be interesting to take into account. If the material model is continuous in the elemental strain, part of it could be captured in the foundation points. If that is not enough, the non-linear material equations could be used to make an approximation of the load-deflection curve instead of a Taylor expansion. This could be interesting for structures where there are material non-linearities and no large deflections.

A general trend in all examples is that the mathematical error that the approximate analysis produces gets smaller for decreasing scale factors. This is not in line with the mathematical error of the finite difference derivatives of the load curve, as that error gets larger for smaller scale factors. We can conclude from this phenomenon that the mathematical error produced by approximate analysis is in favor of the equilibrium path for lower scale factors.

It is peculiar for numerical errors to appear in topology optimization. In this work, these errors were first encountered when obtaining finite difference sensitivities. This led to the belief that the analytical sensitivities were wrong for a long time, but when the numerical error was discovered it inspired deeper research on that subject. The errors are not a problem in optimization as there are more than enough SFs on which responses and analytical sensitivities do not suffer from the errors. When building new response functions based on approximate analysis, one should take note these errors exist, when checking the sensitivities.

4.2. Response functions

Constraining $\|\mathbf{r}_s\|$ has shown to be effective in mitigating the effects of unrealistically deformed elements. It must be noted that this constraint only indirectly steers the imbalance in the solid parts of a structure to a more realistic solution and is thus more focused on the extreme deformations of void elements. As this function does not allow for strains to become large, it indirectly constrains the range of motion of the designs obtained with this method. This might explain the difference in the performance of approximated analysis and non-linear analysis for inverter designs.

For all cases that were compared on basis of the residual, it was shown that the force imbalance is at least an order of magnitude less when compared to linear analysis. The norm of the residual

is only indirectly a measure for accuracy of the analysis. That is why all results were compared to a non-linear analyzed solution as well. If there is another way to measure the accuracy of approximate or linear analysis, without doing a non-linear analysis, one might use that as a response function to make more accurate designs. Using the norm of $\|\mathbf{x}\|$ showed to be effective in constraining the residual but introduced gray beams. This is probably caused by the fact that the residual is high valued in the solid material. A combination of this and the $\|\mathbf{x}_0\|$ constraint could be effective in constraining extreme deformations, while at the same time also constraining the force imbalance in the structure. Tests performed by the author of this paper in combining those two response functions did not give promising results yet.

The hinges produced by the density filter in combination with Heaviside projection in the inverter example are not realistically manufacturable. A method to improve the manufacturability of the designs is using a robust formulation, which guarantees minimum feature size in the hinges. Using robust formulations, however, leads to designs that underperformed in comparison to non-robustly optimized designs. This is probably caused by the thicker hinges, which constrain the range of motion of the designs. Therefore, the effects of approximate analysis on topology optimization are less visible. The focus of this paper was to describe approximate analysis as thoroughly as possible and thus it was opted to not use robust formulation.

The flexure design objective showed values for f_0 that were very close to each other. The objective for constant spring stiffness could be improved by altering the response function, such that the difference in the value of f_1 and f_2 are closer to each other. Multiplying f_1 with a high valued scalar would not work as this only leads to a higher value of f_0 . The value of response function f_1 showed to be very close to 1. Therefore, it might be interesting to subtract a substantial amount from this factor, such that the value difference is more pronounced, for example, $f_3 = f_1 - 0.95$. This way, better performing flexures could be produced, as the optimizer now focuses more on the shape of the compliance-load curve.

All in all, each new method comes with its own difficulties. The difficulty that is most notable in approximate analysis is the tendency of the optimizer to misuse the approximation to its benefit. Constraining the optimizer to not misuse this approximation is quite a task and using the norm of the solid residual showed great potential. There are probably better ways of constraining this misbehavior shown by the optimizer, however, they are yet to be discovered. In this research it was found that the combination of design objective and load case can be beneficial for approximate analysis, such that misuse of approximate analysis is avoided. Choosing the design objectives in a similar fashion as the flexure case in Section 2.5 yields a more stable optimization process.

5

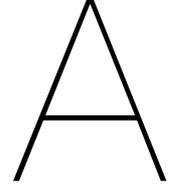
Conclusion

In this work, a method was created to use topology optimization for the design of compliant mechanisms with a finite range. The method consists of a new kind of structural analysis, approximate analysis, in which an extrapolation was built on the basis of a few equilibrium points close to the undeformed configuration of a structure.

This first research done on this type of approximate analysis within topology optimization shows promising results. Concerning the structural analysis in topology optimization, this method combines a higher accuracy of the designs compared to linear analysis with a lower computational burden compared to non-linear analysis. For compliant mechanism design in the real world, taking geometrical non-linearity into account is important and linear analysis does not meet the requirements. Approximate analysis has shown that it can attain sufficient accuracy to create designs exhibiting mildly non-linear behaviour.

Outside of compliant mechanism design, approximate analysis might also be useful. In structural analysis, it might be used to accelerate the Newton-Raphson process, something that is done already to a certain extent with modified Newton schemes. In the analysis of structures exhibiting material non-linearity, approximate analysis might be used as a computationally efficient alternative to non-linear analysis. Approximate analysis could be applied to other physical phenomena where non-linearity plays a role as well.

Approximate analysis showed great potential for the optimization of finite-range compliant mechanisms. The hope of the author is that the implementation and adoption of this method by other researchers may lead to new insights about this way of approximating load-deflection curves.



Sensitivity analysis

This appendix contains all sensitivity analyses performed in this thesis. All sensitivities are obtained using the adjoint method, for which a recap will be given.

A.1. Adjoint method

The adjoint method (Arora et al., 1979) is slightly different for linear, non-linear and approximate analysis, which is why they are all three stated here.

Adjoint method for linear analysis

Any response function analyzed with linear FEM can be objected to the adjoint method in the following way:

$$\begin{aligned} f_0 &= f_0(\mathbf{u}, \mathbf{x}), \\ f_L &= f_0 + \boldsymbol{\mu}^T(\mathbf{K}\mathbf{u} - \mathbf{f}^{\text{ext}}), \end{aligned} \quad (\text{A.1})$$

$$\frac{df_L}{dx_k} = \frac{\partial f_0}{\partial x_k} + \frac{\partial f_0}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx_k} + \boldsymbol{\mu}^T \left(\frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u} + \mathbf{K} \frac{d\mathbf{u}}{dx_k} \right), \quad (\text{A.2})$$

$$\frac{df_L}{dx_k} = \frac{\partial f_0}{\partial x_k} + \left(\frac{\partial f_0}{\partial \mathbf{u}} + \boldsymbol{\mu}^T \mathbf{K} \right) \frac{d\mathbf{u}}{dx_k} + \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}. \quad (\text{A.3})$$

If linear FEM is used, this equation simplifies to the following equation, as first shown in Arora et al. (1979), then called the "State-space" method:

$$\begin{aligned} \frac{df_L}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}, \\ \boldsymbol{\mu} &= -\mathbf{K}^{-1} \left(\frac{\partial f_0}{\partial \mathbf{u}} \right)^T. \end{aligned} \quad (\text{A.4})$$

All that needs to be done, is to find a solution to the adjoint vector $\boldsymbol{\mu}$ and substitute it into the $\frac{df_L}{dx_k}$ to get the sensitivities, which are the same as those of the original problem $\frac{\partial f_0}{\partial x_k}$.

Adjoint method for non-linear analysis

Any objective function f_0 can be adjointed with a Lagrange multiplier and a constraint. In this case we add the equilibrium equation for the FEM, as described in Eq. (2), to the objective, which looks like the following:

$$\begin{aligned}\mathbf{r} &= \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}, \mathbf{x}), \\ f_L &= f_0 + \boldsymbol{\mu}^T \mathbf{r}.\end{aligned}\tag{A.5}$$

Here f_L is the Lagrangian of function f_0 to which it is also equal, since \mathbf{r} is equal to zero. This formulation can then be differentiated with respect to the design variables, which yields Eq. (A.6):

$$\begin{aligned}\frac{df_L}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \frac{\partial f_0}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx_k} + \boldsymbol{\mu}^T \left(\frac{\partial \mathbf{r}}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx_k} + \frac{\partial \mathbf{r}}{\partial x_k} \right), \\ \frac{\partial \mathbf{r}}{\partial \mathbf{u}} &= -\mathbf{K}_t,\end{aligned}\tag{A.6}$$

$$\frac{df_L}{dx_k} = \frac{\partial f_0}{\partial x_k} + \left(\frac{\partial f_0}{\partial \mathbf{u}} - \boldsymbol{\mu}^T \mathbf{K}_t \right) \frac{d\mathbf{u}}{dx_k} + \boldsymbol{\mu}^T \frac{\partial \mathbf{r}}{\partial x_k}.$$

Because \mathbf{r} is equal to zero at equilibrium, $\boldsymbol{\mu}$ is free to choose. Now if $\boldsymbol{\mu}$ is chosen such that $\left(\frac{\partial f_0}{\partial \mathbf{u}} - \boldsymbol{\mu}^T \mathbf{K}_t \right) = \mathbf{0}$, the general form of the adjoint formulation arises (Buhl et al., 2000):

$$\begin{aligned}\frac{df_L}{dx_k} &= \frac{\partial f_0}{\partial x_k} + \boldsymbol{\mu}^T \frac{\partial \mathbf{r}}{\partial x_k}, \\ \boldsymbol{\mu} &= \mathbf{K}_t^{-1} \left(\frac{\partial f_0}{\partial \mathbf{u}} \right)^T.\end{aligned}\tag{A.7}$$

Adjoint method for Taylor approximation

The full derivation for the adjoint method for the Taylor approximation is already described in the thesis. The final equation was:

$$\begin{aligned}\boldsymbol{\mu}_i &= q_i \mathbf{K}_{t,i}^{-1} \left(\frac{\partial f_0}{\partial \mathbf{u}} \right)^T, \\ \frac{df_0}{dx_k} &= \frac{\partial f_0}{\partial x_k} - \sum_{i=1}^n \boldsymbol{\mu}_i^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}.\end{aligned}\tag{A.8}$$

A.2. End-compliance

End-compliance of linear analysis

To determine the end-compliance for the linear analysis Eq. (A.4) for the linear adjoint method can be adjusted with $\frac{\partial f_0}{\partial \mathbf{u}} = \mathbf{f}^{\text{ext}T}$ and $\frac{\partial f_0}{\partial x_k} = 0$. This gives:

$$\begin{aligned}\frac{df_L}{dx_k} &= \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}, \\ \boldsymbol{\mu} &= -\mathbf{K}^{-1} (\mathbf{f}^{\text{ext}}).\end{aligned}\tag{A.9}$$

End-compliance of non-linear analysis

The same can be done for the end-compliance for non-linear analysis. Eq. (A.7) can be rewritten with $\frac{\partial f_0}{\partial \mathbf{u}} = \mathbf{f}^{\text{ext}\top}$ and $\frac{\partial f_0}{\partial x_k} = 0$. Then:

$$\frac{df_L}{dx_k} = \boldsymbol{\mu}^\top \frac{\partial \mathbf{r}}{\partial x_k}, \quad (\text{A.10})$$

$$\boldsymbol{\mu} = \mathbf{K}_t^{-1}(\mathbf{f}^{\text{ext}}).$$

End-compliance of Taylor approximation

Again the same substitution can be used for the Taylor approximation end-compliance with $\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} = \mathbf{f}^{\text{ext}\top}$ and $\frac{\partial f_0}{\partial x_k} = 0$ in Eq. (A.8). This results in:

$$\frac{df_0}{dx_k} = - \sum_{i=1}^n \boldsymbol{\mu}_i^\top \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}, \quad (\text{A.11})$$

$$\boldsymbol{\mu}_i = q_i \mathbf{K}_{t,i}^{-1} \mathbf{f}^{\text{ext}}.$$

A.3. 1-DOF

1-DOF for linear analysis

For the 1-DOF response function, the linear sensitivity analysis can be done filling in $\frac{\partial f_0}{\partial x_k} = 0$ and $\frac{\partial f_0}{\partial \mathbf{u}} = \mathbf{l}^\top$ in Eq. (A.4). This results in:

$$\frac{df_L}{dx_k} = \boldsymbol{\mu}^\top \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}, \quad (\text{A.12})$$

$$\boldsymbol{\mu} = -\mathbf{K}^{-1} \mathbf{l}.$$

1-DOF for non-linear analysis

The non-linear sensitivity analysis for the 1-DOF response function can be performed using $\frac{\partial f_0}{\partial x_k} = 0$ and $\frac{\partial f_0}{\partial \mathbf{u}} = \mathbf{l}^\top$ in Eq. (A.7).

$$\frac{df_L}{dx_k} = \boldsymbol{\mu}^\top \frac{\partial \mathbf{r}}{\partial x_k}, \quad (\text{A.13})$$

$$\boldsymbol{\mu} = \mathbf{K}_t^{-1} \mathbf{l}.$$

1-DOF for Taylor approximation

This sensitivity analysis using the Taylor approximation for this response function can be done using the substitutions $\frac{\partial f_0}{\partial x_k} = 0$ and $\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} = \mathbf{l}^\top$ in Eq. (A.8):

$$\frac{df_0}{dx_k} = - \sum_{i=1}^n \boldsymbol{\mu}_i^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}, \quad (\text{A.14})$$

$$\boldsymbol{\mu}_i = q_i \mathbf{K}_{t,i}^{-1} \mathbf{l}.$$

A.4. Residual

The norms of two residuals are used in this thesis. The procedure of obtaining the sensitivities is the same, with only the residual itself and the tangent stiffness matrix being different. The derivation for the normal residual will be given. For the solid residual \mathbf{r}_s is used in the derivation instead of \mathbf{r} , and instead of $\mathbf{K}_{t,\bar{\mathbf{u}}}$, the tangent matrix of the same deflection, but all elements are solid $\mathbf{K}_{st,\bar{\mathbf{u}}}$ is used. Note that the tangent stiffness matrix and residual are not different for the foundation points.

Residual for linear analysis

It might be interesting to use the residual of \mathbf{u}_{lin} in response function, as it is an indicator for non-linearity. A good scalar representation for this non-linearity would be the norm of the residual:

$$f_0 = \sqrt{\mathbf{r}^T \mathbf{r}}. \quad (\text{A.15})$$

To be clear, if \mathbf{K} is the stiffness matrix of the undeformed configuration, with which a linear analysis is done, then:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{K}^{-1} \mathbf{f}^{\text{ext}}, \\ \mathbf{r} &= \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\mathbf{u}_{\text{lin}}, \mathbf{x}). \end{aligned} \quad (\text{A.16})$$

For readability subscript lin in \mathbf{r}_{lin} is dropped and \mathbf{u}_{lin} is shortened to \mathbf{u}_1 . \mathbf{u}_1 is now the deformed configuration for a linear analysis. In this configuration, the system has a tangent stiffness matrix \mathbf{K}_t and a residual \mathbf{r} . Now set up the adjoint equation for the norm of the residual.

$$f_0 = \sqrt{\mathbf{r}^T \mathbf{r}} + \boldsymbol{\mu}^T (\mathbf{K} \mathbf{u}_1 - \mathbf{f}^{\text{ext}}), \quad (\text{A.17})$$

$$\frac{df_0}{dx_k} = \frac{\partial f_0}{\partial x_k} + \frac{\partial f_0}{\partial \mathbf{u}_1} \frac{d\mathbf{u}_1}{dx_k} + \boldsymbol{\mu}^T \left(\mathbf{K} \frac{d\mathbf{u}_1}{dx_k} + \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}_1 \right), \quad (\text{A.18})$$

$$\frac{df_0}{dx_k} = \frac{\partial f_0}{\partial x_k} + \left(\frac{\partial f_0}{\partial \mathbf{u}_1} + \boldsymbol{\mu}^T \mathbf{K} \right) \frac{d\mathbf{u}_1}{dx_k} + \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}_1.$$

The values in Eq. (A.18) can be substituted by:

$$\frac{\partial \mathbf{r}}{\partial x_k} = - \frac{\partial \mathbf{f}^{\text{int}}}{\partial x_k}, \quad \frac{\partial \mathbf{r}}{\partial \mathbf{u}_1} = -\mathbf{K}_t. \quad (\text{A.19})$$

$$\frac{\partial f_0}{\partial x_k} = \frac{1}{2\sqrt{\mathbf{r}^T \mathbf{r}}} \frac{\partial (\mathbf{r}^T \mathbf{r})}{\partial x_k}, \quad (\text{A.20})$$

$$\frac{\partial f_0}{\partial x_k} = - \frac{\mathbf{r}^T \frac{\partial \mathbf{f}^{\text{int}}}{\partial x_k}}{\sqrt{\mathbf{r}^T \mathbf{r}}}, \quad (\text{A.21})$$

$$\frac{\partial f_0}{\partial \mathbf{u}_1} = \frac{1}{2\sqrt{\mathbf{r}^T \mathbf{r}}} \frac{\partial (\mathbf{r}^T \mathbf{r})}{\partial \mathbf{u}_1}, \quad \frac{\partial f_0}{\partial \mathbf{u}_1} = - \frac{\mathbf{K}_t \mathbf{r}}{\sqrt{\mathbf{r}^T \mathbf{r}}}.$$

Now Eq. (A.21) are substituted in the adjoint formulation A.18 to obtain the sensitivities:

$$\frac{df_0}{dx_k} = -\frac{\mathbf{r}^T \frac{\partial \mathbf{f}^{\text{int}}}{\partial x_k}}{\sqrt{\mathbf{r}^T \mathbf{r}}} + \left(-\frac{\mathbf{K}_t \mathbf{r}}{\sqrt{\mathbf{r}^T \mathbf{r}}} + \boldsymbol{\mu}^T \mathbf{K} \right) \frac{d\mathbf{u}_1}{dx_k} - \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}_1, \quad (\text{A.22})$$

$$\boldsymbol{\mu} = \mathbf{K}^{-1} \frac{\mathbf{K}_t \mathbf{r}}{\sqrt{\mathbf{r}^T \mathbf{r}}},$$

(A.23)

$$\frac{df_0}{dx_k} = -\frac{\mathbf{r}^T \frac{\partial \mathbf{f}^{\text{int}}}{\partial x_k}}{\sqrt{\mathbf{r}^T \mathbf{r}}} - \boldsymbol{\mu}^T \frac{\partial \mathbf{K}}{\partial x_k} \mathbf{u}_1.$$

These equations were successfully validated with finite difference sensitivities.

Residual for Taylor approximation

The following objective function:

$$f_0 = \sqrt{\mathbf{r}^T \mathbf{r}}, \quad (\text{A.24})$$

$$\mathbf{r} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}}(\tilde{\mathbf{u}}, \mathbf{x}).$$

is a bit different from the norm of the residual for linear analysis. Sensitivities are obtained by filling in Eq. (A.8) (duplicated here for clarity):

$$\boldsymbol{\mu}_i = q_i \mathbf{K}_{t,i}^{-1} \frac{\partial f_0}{\partial \tilde{\mathbf{u}}}, \quad (\text{A.25})$$

$$\frac{df_0}{dx_k} = \frac{\partial f_0}{\partial x_k} - \sum_{i=1}^n \boldsymbol{\mu}_i^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}.$$

Using the same approach as with the linear analysis results in:

$$\frac{\partial \mathbf{r}}{\partial x_k} = -\frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}, \quad \frac{\partial \mathbf{r}}{\partial \tilde{\mathbf{u}}} = -\mathbf{K}_{t,\tilde{\mathbf{u}}}, \quad (\text{A.26})$$

$$\frac{\partial \mathbf{r}_i}{\partial x_k} = -\frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}, \quad \frac{\partial \mathbf{r}_i}{\partial \tilde{\mathbf{u}}} = -\mathbf{K}_{t,i}.$$

$$\frac{\partial f_0}{\partial x_k} = \frac{1}{2\sqrt{\mathbf{r}^T \mathbf{r}}} \frac{\partial (\mathbf{r}^T \mathbf{r})}{\partial x_k}, \quad (\text{A.27}) \quad \frac{\partial f_0}{\partial x_k} = -\frac{\mathbf{r}^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}}{\sqrt{\mathbf{r}^T \mathbf{r}}}, \quad (\text{A.28})$$

$$\frac{\partial f_0}{\partial \tilde{\mathbf{u}}} = \frac{1}{2\sqrt{\mathbf{r}^T \mathbf{r}}} \frac{\partial (\mathbf{r}^T \mathbf{r})}{\partial \tilde{\mathbf{u}}}, \quad \frac{\partial f_0}{\partial \tilde{\mathbf{u}}} = -\frac{\mathbf{K}_{t,\tilde{\mathbf{u}}} \mathbf{r}}{\sqrt{\mathbf{r}^T \mathbf{r}}}.$$

This yields:

$$\boldsymbol{\mu}_i = -q_i \mathbf{K}_{t,i}^{-1} \frac{\mathbf{K}_{t,\mathbf{u}} \mathbf{r}}{\sqrt{\mathbf{r}^T \mathbf{r}}},$$

(A.29)

$$\frac{df_0}{dx_k} = -\frac{\mathbf{r}^T \frac{\partial \mathbf{f}_{\mathbf{u}}^{\text{int}}}{\partial x_k}}{\sqrt{\mathbf{r}^T \mathbf{r}}} - \sum_{i=1}^n \boldsymbol{\mu}_i^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k}.$$

The equations were tested with the finite difference and were found to be correct.

A.5. Second derivative of the load curve

With approximate analysis, more information on the deflection path of a structure is known, in the sense that the deflection is continuous in the load factor λ , $\mathbf{u}[\lambda]$. For example, the second derivative in the load factor on any point on the load curve can be measured. If used, sensitivities are needed, where ℓ is a vector with one 1 on the DOF interest:

$$f_0 = \frac{d^2 \mathbf{u}}{d\lambda^2} \ell. \quad (\text{A.30})$$

From Eq. (11) it is known that:

$$\frac{d^2 \mathbf{u}}{d\lambda^2} = \alpha_1 \mathbf{u}_1 + \alpha_2 \mathbf{u}_2 + \dots \alpha_n \mathbf{u}_n, \quad (\text{A.31})$$

$$\frac{df_0}{d\mathbf{u}_i} = \alpha_i \ell. \quad (\text{A.32})$$

Function ($f_0 = f_0(\mathbf{u}_1, \dots \mathbf{u}_n)$), is only dependent on the foundation points, which on their turn dependent on \mathbf{x} . So the function is only indirectly dependent on \mathbf{x} :

$$\frac{df_0}{dx_k} = \sum_{i=1}^n \left(\frac{df_0}{d\mathbf{u}_i} \frac{d\mathbf{u}_i}{dx_k} + \boldsymbol{\mu}_i^T \frac{d\mathbf{r}_i}{dx_k} \right),$$

$$\frac{dg_i}{dx_k} = \frac{df_0}{d\mathbf{u}_i} \frac{d\mathbf{u}_i}{dx_k} + \boldsymbol{\mu}_i^T \frac{d\mathbf{r}_i}{dx_k}, \quad (\text{A.33})$$

$$\frac{dg_i}{dx_k} = \frac{df_0}{d\mathbf{u}_i} \frac{d\mathbf{u}_i}{dx_k} + \boldsymbol{\mu}_i^T \left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{u}_i} \frac{d\mathbf{u}_i}{dx_k} + \frac{\partial \mathbf{r}_i}{\partial x_k} \right),$$

$$\frac{dg_i}{dx_k} = \left(\frac{df_0}{d\mathbf{u}_i} + \boldsymbol{\mu}_i^T \frac{\partial \mathbf{r}_i}{\partial \mathbf{u}_i} \right) \frac{d\mathbf{u}_i}{dx_k} + \boldsymbol{\mu}_i^T \frac{\partial \mathbf{r}_i}{\partial x_k}.$$

Now filling in Eq. (A.31) finally results in:

$$\boldsymbol{\mu}_i^T = \alpha_i \mathbf{K}_t^{-1}(\mathbf{u}_i) \ell,$$

$$\frac{df_0}{dx_k} = \sum_{i=1}^n \left(\boldsymbol{\mu}_i^T \frac{\partial \mathbf{f}_i^{\text{int}}}{\partial x_k} \right). \quad (\text{A.34})$$

A.6. Compliance shape

How well the shape of the compliance approximates a quadratic curve in the load factor λ can be defined as follows:

$$f_0 = \frac{f_n}{f_d} = \frac{\boldsymbol{\gamma}^T \mathbf{c}_\lambda}{\|\mathbf{c}_\lambda\| \|\boldsymbol{\gamma}\|}, \quad (\text{A.35})$$

$$f_d = \|\boldsymbol{\gamma}\| \sqrt{\mathbf{c}_\lambda^T \mathbf{c}_\lambda}, \quad (\text{A.36})$$

$$f_n = \boldsymbol{\gamma}^T \mathbf{c}_\lambda. \quad (\text{A.37})$$

First we define a vector with the compliances at different steps in λ and a vector with the quadratic values of λ and norm 1:

$$\mathbf{c}_\lambda = \begin{bmatrix} \mathbf{f}_1^T \mathbf{u}_1 \\ \mathbf{f}_2^T \mathbf{u}_2 \\ \vdots \\ \mathbf{f}_p^T \mathbf{u}_p \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} \lambda_1^2 \\ \lambda_2^2 \\ \vdots \\ \lambda_p^2 \end{bmatrix}. \quad (\text{A.38})$$

Gamma is thus not dependent on the design variables or deflection field.

$$\frac{df_d}{dx_k} = \frac{\|\boldsymbol{\gamma}\|}{2\sqrt{\mathbf{c}_\lambda^T \mathbf{c}_\lambda}} \frac{\partial \mathbf{c}_\lambda^T \mathbf{c}_\lambda}{\partial \mathbf{u}_i} \frac{d\mathbf{u}_i}{dx_k}, \quad (\text{A.39})$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \vdots \\ \mathbf{f}_{pp} \end{bmatrix}, \quad \mathbf{q} = \text{diag} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{pp} \end{bmatrix}, \quad (\text{A.40})$$

$$\frac{\partial \mathbf{c}_\lambda}{\partial \mathbf{u}_i} = \begin{bmatrix} \mathbf{f}_1 q_1 \\ \mathbf{f}_2 q_2 \\ \vdots \\ \mathbf{f}_{pp} q_{pp} \end{bmatrix}, \quad \frac{\partial \mathbf{c}_\lambda}{\partial \mathbf{u}_i} = \mathbf{qF}, \quad (\text{A.41})$$

$$\frac{\partial \mathbf{c}_\lambda^T \mathbf{c}_\lambda}{\partial \mathbf{u}_i} = [(\mathbf{qF})^T \mathbf{c}_\lambda]^T + \mathbf{c}_\lambda^T \mathbf{qF}, \quad (\text{A.42})$$

$$\frac{\partial \mathbf{c}_\lambda^T \mathbf{c}_\lambda}{\partial \mathbf{u}_i} = 2\mathbf{c}_\lambda^T \mathbf{qF}, \quad (\text{A.43})$$

$$\frac{df_d}{dx_k} = \frac{\|\boldsymbol{\gamma}\|}{\|\mathbf{c}_\lambda\|} \mathbf{c}_\lambda^T \mathbf{qF} \frac{d\mathbf{u}_i}{dx_k}. \quad (\text{A.44})$$

Now for the numerator, using the same methods as for the denominator:

$$\frac{df_n}{dx_k} = \boldsymbol{\gamma}^T \mathbf{qF} \frac{d\mathbf{u}_i}{dx_k}. \quad (\text{A.45})$$

Putting everything together gives:

$$\frac{df_0}{dx_k} = \frac{f_d \frac{df_n}{dx_k} - f_n \frac{df_d}{dx_k}}{f_d^2}, \quad (\text{A.46})$$

$$\frac{df_0}{dx_k} = \frac{1}{\|\mathbf{c}_\lambda\|^2 \|\boldsymbol{\gamma}\|} (\|\mathbf{c}_\lambda\| \hat{\boldsymbol{\gamma}}^T \mathbf{qF} - \hat{\boldsymbol{\gamma}}^T \mathbf{c}_\lambda \frac{\mathbf{c}_\lambda^T}{\|\mathbf{c}_\lambda\|} \mathbf{qF}) \frac{d\mathbf{u}_i}{dx_k}, \quad (\text{A.47})$$

$$\frac{df_0}{dx_k} = \frac{\|\mathbf{c}_\lambda\| \hat{\boldsymbol{\gamma}}^T - \hat{\boldsymbol{\gamma}}^T \mathbf{c}_\lambda \frac{\mathbf{c}_\lambda^T}{\|\mathbf{c}_\lambda\|}}{\|\mathbf{c}_\lambda\|^2 \|\boldsymbol{\gamma}\|} \mathbf{qF} \frac{d\mathbf{u}_i}{dx_k}. \quad (\text{A.48})$$

This sensitivity was checked with a finite difference sensitivity and found to be accurate.

A.7. Norm of reaction forces

The reaction force at certain DOFs could be used in a response function. In this thesis, the norm of the vector containing the reaction forces at DOFs of preference was chosen. The DOFs of preference are chosen such that some reaction forces can be excluded. To obtain this response function, first, a clear definition of reaction forces should be given, in this case for linear analysis. The equilibrium equations for linear analysis are:

$$\mathbf{K}\mathbf{u} = \mathbf{f}, \quad (\text{A.49})$$

and can be partitioned, in free f and prescribed p DOFs:

$$\begin{bmatrix} \mathbf{K}_{ff} & \mathbf{K}_{fp} \\ \mathbf{K}_{pf} & \mathbf{K}_{pp} \end{bmatrix} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_p \end{bmatrix} = \begin{bmatrix} \mathbf{f}_f \\ \mathbf{f}_p \end{bmatrix}, \quad (\text{A.50})$$

from which the following relations can be derived:

$$\mathbf{K}_{ff}\mathbf{u}_f = \mathbf{f}_f - \mathbf{K}_{fp}\mathbf{u}_p, \quad (\text{A.51})$$

and

$$\mathbf{f}_p = \mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p. \quad (\text{A.52})$$

From these, a response function can be constructed to fulfill the requirements described above. In this response function, \mathbf{A} is a diagonal matrix, with ones on the DOFs that need to be included in the analysis:

$$f_0 = \sqrt{\mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}. \quad (\text{A.53})$$

This function is augmented with equilibrium Eq. (A.51) to

$$f_L = \sqrt{\mathbf{f}_p^T \mathbf{A} \mathbf{f}_p} + \mu^T (\mathbf{K}_{ff}\mathbf{u}_f - \mathbf{f}_f + \mathbf{K}_{fp}\mathbf{u}_p). \quad (\text{A.54})$$

Derivation to x_k yields:

$$\frac{df_L}{dx_k} = \frac{\partial f_0}{\partial x_k} + \frac{\partial f_0}{\partial \mathbf{u}_f} \frac{d\mathbf{u}_f}{dx_k} + \mu^T \left(\frac{\partial \mathbf{K}_{ff}}{\partial x_k} \mathbf{u}_f + \mathbf{K}_{ff} \frac{d\mathbf{u}_f}{dx_k} + \frac{\partial \mathbf{K}_{fp}}{\partial x_k} \mathbf{u}_p \right). \quad (\text{A.55})$$

This equation contains a lot of terms, which will be derived separately:

$$\frac{\partial f_0}{\partial x_k} = \frac{1}{2 \cdot \sqrt{\mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}} \frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial x_k} = \frac{1}{2f_0} \frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial x_k}. \quad (\text{A.56})$$

which can be solved by substituting Eq. (A.52):

$$\frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial x_k} = \frac{\partial}{\partial x_k} ((\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p)^T \mathbf{A} (\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p)). \quad (\text{A.57})$$

Solving the derivatives:

$$\begin{aligned} \frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial x_k} &= \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right)^T \mathbf{A} (\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p) \\ &\quad + (\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p)^T \mathbf{A} \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right), \end{aligned} \quad (\text{A.58})$$

$$\frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial x_k} = 2(\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p)^T \mathbf{A} \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right). \quad (\text{A.59})$$

Back substitution in Eq. (A.56) yields:

$$\frac{\partial f_0}{\partial x_k} = \frac{1}{f_0} (\mathbf{K}_{pf}\mathbf{u}_f + \mathbf{K}_{pp}\mathbf{u}_p)^T \mathbf{A} \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right). \quad (\text{A.60})$$

Now only $\partial f_0/\partial \mathbf{u}_f$ is unknown, but can easily be found by the fact that the start is done the same way as in Eq. (A.56):

$$\frac{\partial f_0}{\partial \mathbf{u}_f} = \frac{1}{2 \cdot \sqrt{\mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}} \frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial \mathbf{u}_f} = \frac{1}{2f_0} \frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial \mathbf{u}_f}, \quad (\text{A.61})$$

which consists of:

$$\frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial \mathbf{u}_f} = \frac{\partial}{\partial \mathbf{u}_f} ((\mathbf{K}_{pf} \mathbf{u}_f + \mathbf{K}_{pp} \mathbf{u}_p)^T \mathbf{A} (\mathbf{K}_{pf} \mathbf{u}_f + \mathbf{K}_{pp} \mathbf{u}_p)), \quad (\text{A.62})$$

$$\frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial \mathbf{u}_f} = \mathbf{K}_{pf}^T \mathbf{A} (\mathbf{K}_{pf} \mathbf{u}_f + \mathbf{K}_{pp} \mathbf{u}_p) + (\mathbf{K}_{pf} \mathbf{u}_f + \mathbf{K}_{pp} \mathbf{u}_p)^T \mathbf{A} \mathbf{K}_{pf}, \quad (\text{A.63})$$

$$\frac{\partial \mathbf{f}_p^T \mathbf{A} \mathbf{f}_p}{\partial \mathbf{u}_f} = \mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p + \mathbf{f}_p^T \mathbf{A} \mathbf{K}_{pf} = 2\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p. \quad (\text{A.64})$$

This result is back substituted in Eq. (A.61) and yields:

$$\frac{\partial f_0}{\partial \mathbf{u}_f} = \frac{\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p}{f_0}. \quad (\text{A.65})$$

Now substitution of Eq. (A.60) and Eq. (A.65) into Eq. (A.55) yields:

$$\begin{aligned} \frac{df_L}{dx_k} &= \frac{1}{f_0} (\mathbf{K}_{pf} \mathbf{u}_f + \mathbf{K}_{pp} \mathbf{u}_p)^T \mathbf{A} \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right) \\ &+ \frac{\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p}{f_0} \frac{d\mathbf{u}_f}{dx_k} + \boldsymbol{\mu}^T \left(\frac{\partial \mathbf{K}_{ff}}{\partial x_k} \mathbf{u}_f + \mathbf{K}_{ff} \frac{d\mathbf{u}_f}{dx_k} + \frac{\partial \mathbf{K}_{fp}}{\partial x_k} \mathbf{u}_p \right), \end{aligned} \quad (\text{A.66})$$

$$\begin{aligned} \frac{df_L}{dx_k} &= \frac{1}{f_0} \mathbf{f}_p^T \mathbf{A} \left(\frac{\partial \mathbf{K}_{pf}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{pp}}{\partial x_k} \mathbf{u}_p \right) + \boldsymbol{\mu}^T \left(\frac{\partial \mathbf{K}_{ff}}{\partial x_k} \mathbf{u}_f + \frac{\partial \mathbf{K}_{fp}}{\partial x_k} \mathbf{u}_p \right) \\ &+ \left(\frac{\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p}{f_0} + \boldsymbol{\mu}^T \mathbf{K}_{ff} \right) \frac{d\mathbf{u}_f}{dx_k}, \end{aligned} \quad (\text{A.67})$$

$$\frac{df_L}{dx_k} = \left[\boldsymbol{\mu}^T \frac{\mathbf{f}_p^T \mathbf{A}}{f_0} \right] \frac{\partial \mathbf{K}}{\partial x_k} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_p \end{bmatrix} + \left(\frac{\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p}{f_0} + \boldsymbol{\mu}^T \mathbf{K}_{ff} \right) \frac{d\mathbf{u}_f}{dx_k}. \quad (\text{A.68})$$

All equations needed to obtain the sensitivities are now complete:

$$\frac{df_0}{dx_k} = \left[\boldsymbol{\mu}^T \frac{\mathbf{f}_p^T \mathbf{A}}{f_0} \right] \frac{\partial \mathbf{K}}{\partial x_k} \begin{bmatrix} \mathbf{u}_f \\ \mathbf{u}_p \end{bmatrix}, \quad (\text{A.69})$$

with

$$\mathbf{K}_{ff} \boldsymbol{\mu} = -\frac{\mathbf{K}_{fp} \mathbf{A} \mathbf{f}_p}{f_0}. \quad (\text{A.70})$$

This sensitivity was successfully validated with a finite difference check.

A.8. Finite Difference check

To check analytical sensitivities, a check file was written. To the check file, a self-chosen design and load case are supplied. This means that at any moment in a topology optimization routine, the process can be stopped and the design and load case can be fed into the sensitivity checker. A finite-difference analysis is done within the check file for every response function used in this thesis. The finite difference sensitivities are calculated for one element, by perturbing the density of that element by a small Δx . After that, an analytical analysis is done for all elements. Only the analytical sensitivity of the element of interest is compared to check the correctness of the analytical sensitivity. For example, for end-compliance the sensitivity check would look like the following, where the first row is the function value and the second and third the finite difference and analytical values, respectively:

	Linear analysis	Non-linear analysis	Approximate analysis
End compliance	2061.493	3114.804	3096.483
Finite difference	11.04235	19.18782	19.41857
Adjoint method	11.04126	19.18619	19.41677

Table A.1: Typical results from the sensitivity analysis check. In this case, end-compliance was considered.

B

Minimal working example

This appendix considers the solutions to the minimum working example as described in the paper.

B.1. Load case

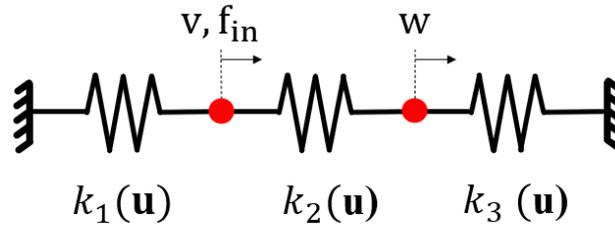


Figure B.1: The three spring model with two DOFs, located at the red dots in the horizontal direction. This model is used to demonstrate procedures to obtain deflection values $\mathbf{u} = [v, w]^T$ as a consequence of input force f_{in} on the left DOF. The stiffnesses of the springs depend on the deflections of the DOFs, represented by \mathbf{u} , and are thus geometrically non-linear.

Consider the geometrically non-linear spring model shown in Fig. B.1. This model has three zero-length springs, which have different stiffnesses $k_i(\mathbf{u})$. Spring one and three are fixed on one side and connected to spring two on the other side. The connections between the springs are called nodes 1 and 2. This means this system has two degrees of freedom (DOFs) in the horizontal direction, located at those nodes. The internal forces these springs exert on the free dDOFs f_{k_i} can be calculated in the following way, from which their stiffness k_i can be derived:

$$\begin{aligned} f_{k_i} &= f_{k_i}[\mathbf{u}], \\ k_i &= \frac{df_{k_i}^{int}}{dl}. \end{aligned} \tag{B.1}$$

Where l is the length of the springs and can be written as a function of the deflection of the DOFs, for example the length of spring two can be written as $l_2 = v - w$. The springs can exert internal forces on the DOFs. For example, the internal force on DOF v is the sum of the forces produced by spring one and two:

$$f_v^{int} = f_{k_1} - f_{k_2}, \tag{B.2}$$

$$f_w^{int} = f_{k_2} - f_{k_3}. \tag{B.3}$$

On each DOF an external force can be applied. In this example, only on the first DOF a force is applied. The objective is now to find the correct deflections \mathbf{u} for the corresponding external forces. To

do this the Newton-Raphson method is used, which is explained in Section 2.2. The most important equation will be copied here:

$$\mathbf{K}_t[\mathbf{u}]\Delta\mathbf{u} = \mathbf{r}[\mathbf{u}], \quad (\text{B.4})$$

with

$$\mathbf{r}[\mathbf{u}] = \mathbf{f}^{\text{int}}[\mathbf{u}] - \mathbf{f}^{\text{ext}}. \quad (\text{B.5})$$

The \mathbf{K}_t -matrix is constructed by differentiating Eq. (B.2) and Eq. (B.3) to DOFs u and v , and assembled into a linear system of equations, in which all variables of Eq. (B.4) are evaluated:

$$\begin{aligned} \mathbf{K}_{t,1} &= \begin{bmatrix} k_1[\mathbf{u}] + k_2[\mathbf{u}] & -k_2[\mathbf{u}] \\ -k_2[\mathbf{u}] & k_2[\mathbf{u}] + k_3[\mathbf{u}] \end{bmatrix}, \\ \mathbf{f}^{\text{ext}} &= \begin{bmatrix} f_{\text{in}} \\ 0 \end{bmatrix}, \\ \mathbf{f}^{\text{int}}[\mathbf{u}] &= \begin{bmatrix} f_v^{\text{int}}[\mathbf{u}] \\ f_w^{\text{int}}[\mathbf{u}] \end{bmatrix}, \\ \Delta\mathbf{u} &= \begin{bmatrix} \Delta v \\ \Delta w \end{bmatrix}, \end{aligned} \quad (\text{B.6})$$

with which a step $\Delta\mathbf{u}$ can be calculated as:

$$\Delta\mathbf{u} = \mathbf{K}_{t,1}^{-1}\mathbf{r}[\mathbf{u}]. \quad (\text{B.7})$$

Now a new value for \mathbf{u} is found and again a step $\Delta\mathbf{u}$ is made. The correct deflections are found if the system is in equilibrium, which occurs when the external forces equal the internal forces produced by the spring stiffnesses.

In this example the following internal force functions were used. The odd formulation for spring one is because this is the analytical function for the problem in Appendix B.1 for a specific length and stiffness:

$$f_1^{\text{int}} = \frac{((v - 1/2) \cdot (3.36 \cdot 10^{15} \cdot (v - 1/2)^2 - 3.36 \cdot 10^{16} \cdot ((v - 1/2)^2 + 100)^{1/2} + 3.36 \cdot 10^{17}))}{(1 \cdot 10^{11} \cdot ((v - 1/2)^2 + 100))}, \quad (\text{B.8})$$

$$f_2^{\text{int}} = 4(v - w), \quad (\text{B.9})$$

$$f_3^{\text{int}} = 4(-w). \quad (\text{B.10})$$

Note that springs 2 and 3 are regular linear springs. To demonstrate approximate analysis the number of foundation points and the scale factor SF were varied.

B.2. Results

As the figures on the following pages indicate, more foundation points lead to better accuracy, just like lower scale factors. There is a significant difference with linear analysis, which indicates that using approximate analysis is a viable option. In Fig. B.8 the approximation shows a misfit and diverges from the equilibrium path directly after a small load, this might be because of a numerical error in the calculation of one of the points, but it is a perfect showcase of the danger in using higher-order approximations.

Figure B.2: Force-deflection relations for the minimal working example with 2 foundation points and scale factor 2 using various analyses.

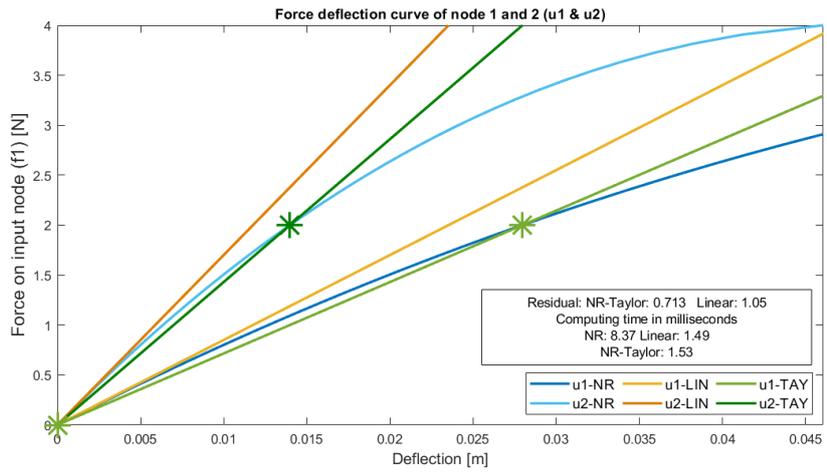


Figure B.3: Force-deflection relations for the minimal working example with 2 foundation points and scale factor 10 using various analyses.

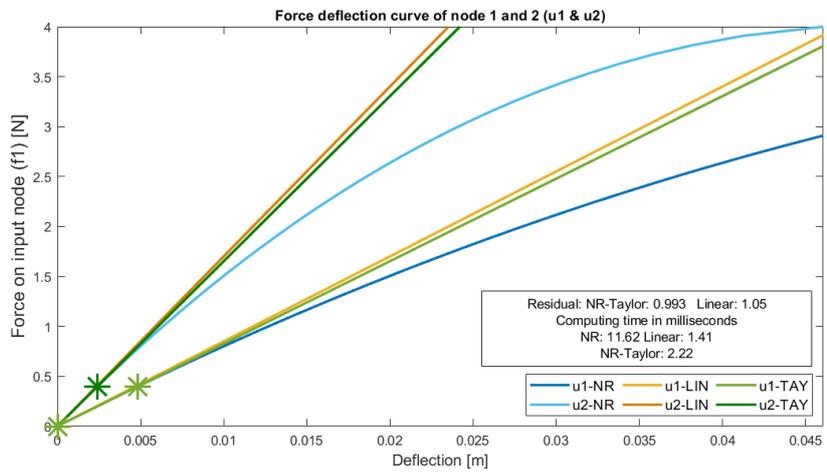


Figure B.4: Force-deflection relations for the minimal working example with 2 foundation points and scale factor 100 using various analyses.

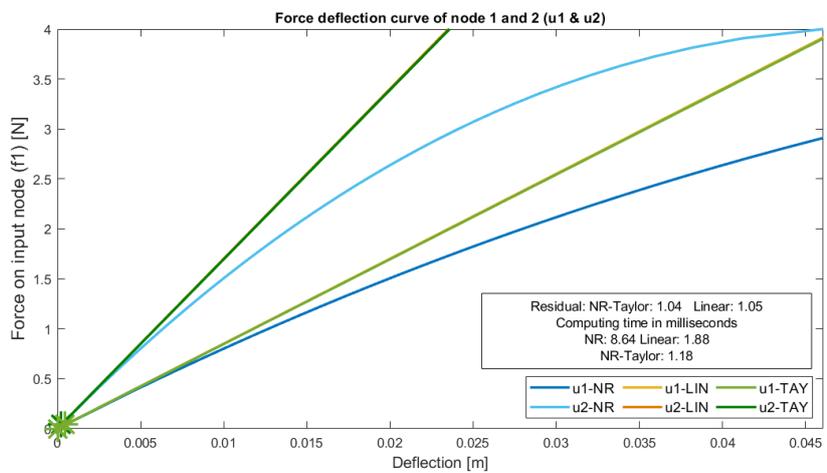


Figure B.5: Force-deflection relations for the minimal working example with 3 foundation points and scale factor 2 using various analyses.

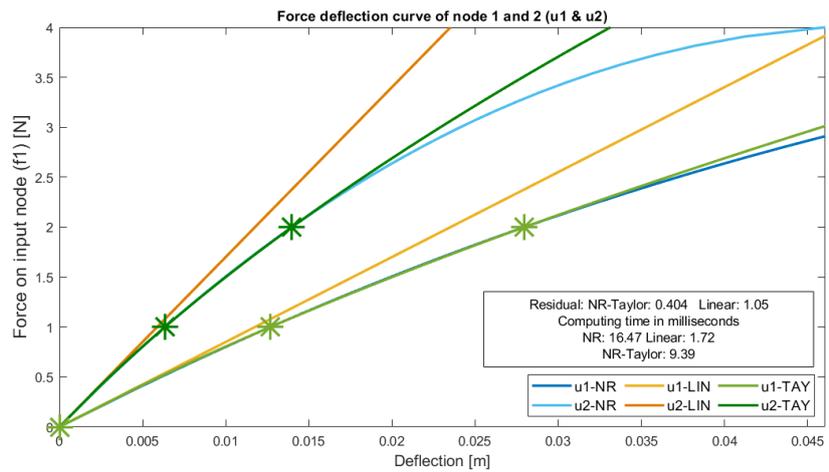


Figure B.6: Force-deflection relations for the minimal working example with 3 foundation points and scale factor 10 using various analyses.

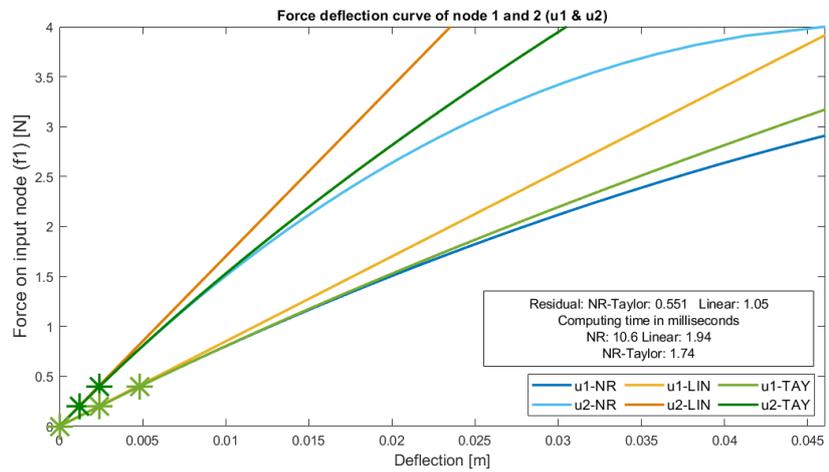


Figure B.7: Force-deflection relations for the minimal working example with 3 foundation points and scale factor 100 using various analyses.

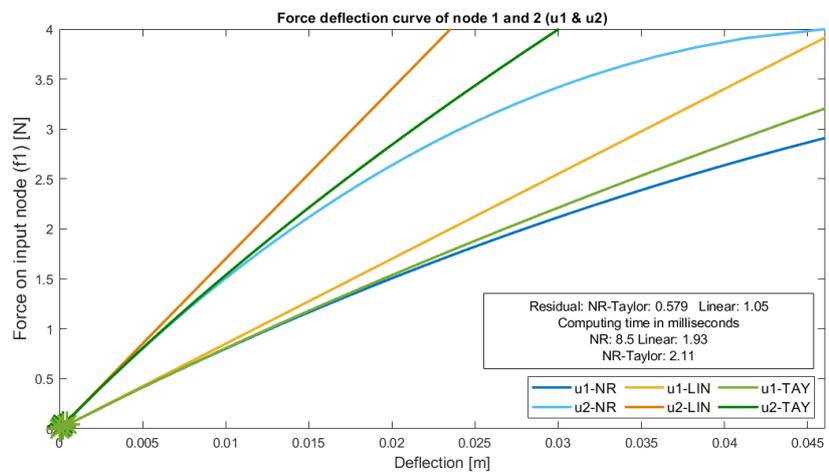


Figure B.8: Force-deflection relations for the minimal working example with 5 foundation points and scale factor 20 using various analyses.

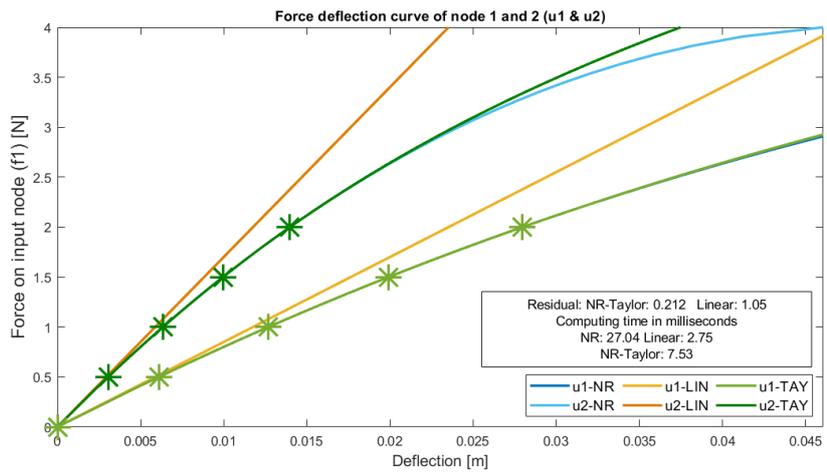


Figure B.9: Force-deflection relations for the minimal working example with 5 foundation points and scale factor 30 using various analyses.

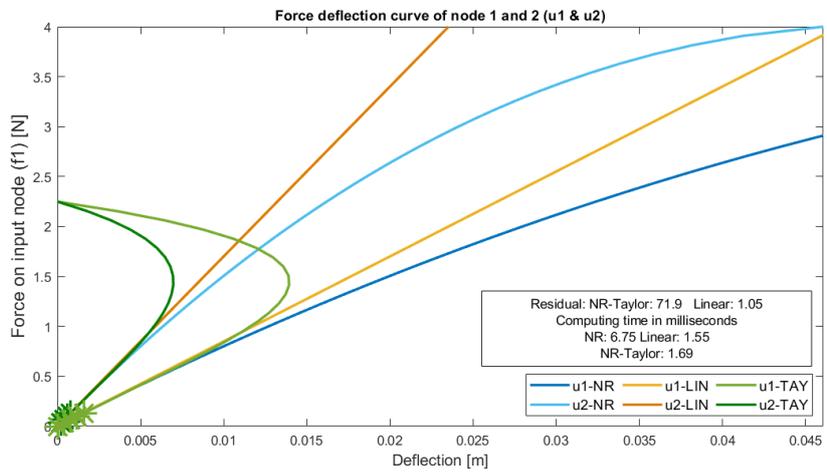
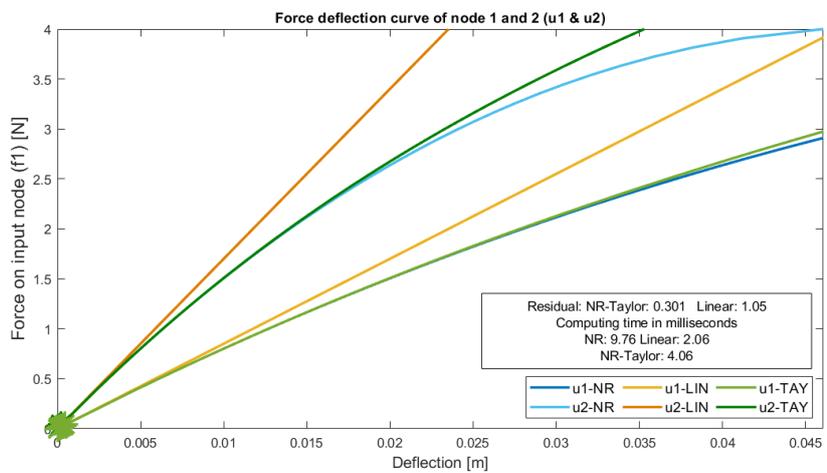
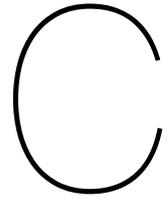


Figure B.10: Force-deflection relations for the minimal working example with 5 foundation points and scale factor 100 using various analyses.





Geometric non-linear FEM

Both analytical tests and comparisons with commercial finite element software were performed to assess the accuracy of the self-written FEM code.

C.1. Analytical verification

The first analytical test performed on the finite element code concerns the calculation of stresses a beam under tension. The beam has dimensions width 5 m, height = 4 m and thickness = 1 m. The material has a Young's modulus of $E = 1 \text{ Pa}$ and a Poisson ratio $\nu = 1/3$ in a plane strain constitutive relation. The bottom of the beam is fixed in the vertical direction and the left edge is fixed in the horizontal direction. A distributed load of $F = 1 \text{ N}$ is applied to the right edge. As can be seen in Fig. C.1, due to the tension, the beam has narrowed to a height of 3.74 m. A simple calculation of the stress is now possible in combination with the surface area $A = \text{height} \times \text{thickness}$ normal to the force:

$$\sigma_{xx} = \frac{F}{A} = \frac{1\text{N}}{3.74\text{m}^2} = 0.26739\text{Pa}. \quad (\text{C.1})$$

The analytical solution of the horizontal stress is equal to the solution found by the finite element code (figure C.1).

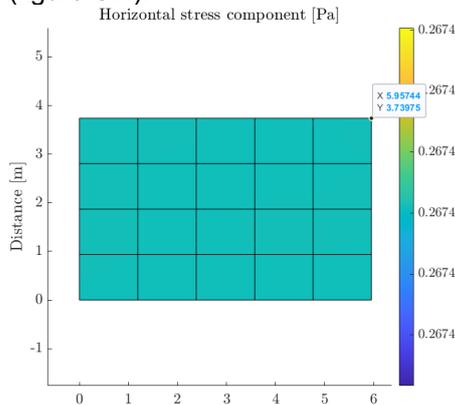


Figure C.1: The stretching of a vertical beam with the corresponding stress distribution.

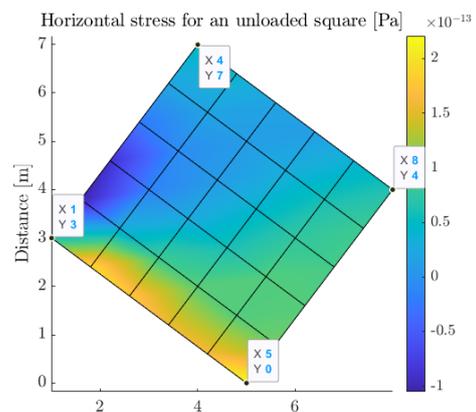


Figure C.2: The rotation of a square with the corresponding stress distribution

The second test was to rotate a structure such there was no residual stress within the structure and the deformation was only a rigid body mode. To do this, a structure with dimensions width 5 m, height = 5 m and thickness = 1 m was considered. The load case is now a vertical displacement of the bottom left corner, while the bottom right corner is fixed in both degrees of freedom. As can be seen in Fig. C.2, the rotation of the square according to the finite element calculation is equal to what is to be expected

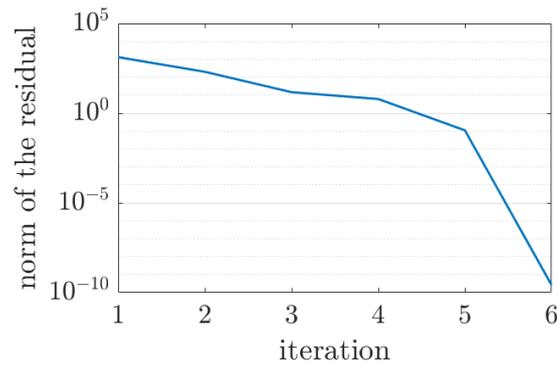


Figure C.3: Convergence of the structure and load case in Fig. C.2

when calculated analytically, and the structure is stress-free up until the convergence tolerance. The convergence per iteration is shown in Fig. C.3, which shows quadratic convergence for the Beam rotate test.

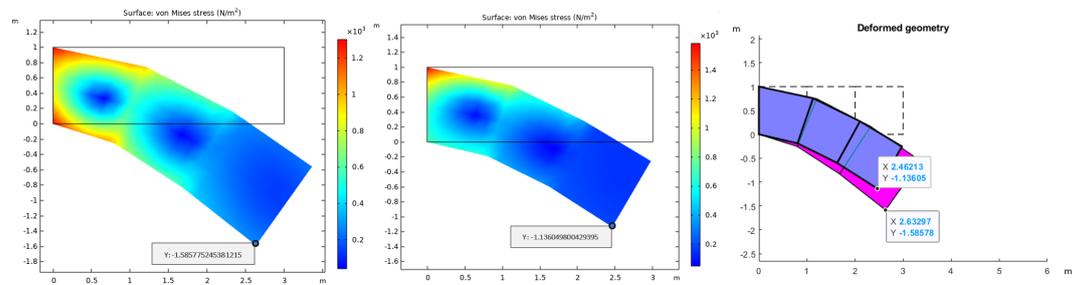
C.2. Verification with Comsol

A few tests were conducted with both the self-written MATLAB script and a Comsol model with the same parameters in order to check whether the MATLAB code is working correctly. Three cases were checked, each with a different amount of elements. All three were force-based. The following features were used for the three tests using three, twenty, and two-thousand elements.

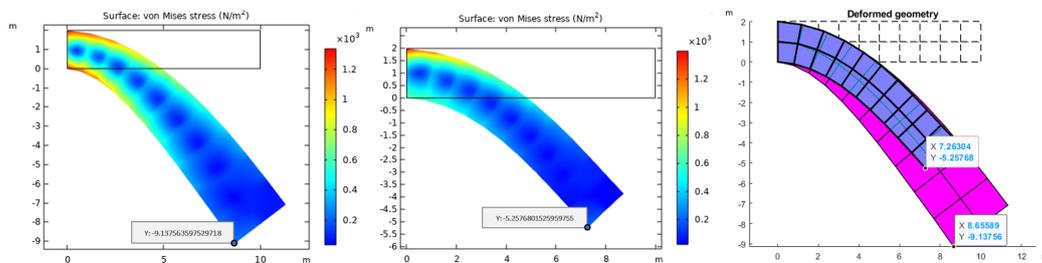
- Element edge size is 1 m
- $E = 5 \text{ kPa}$
- $\nu = 0.3$
- Linear shape functions
- Four integration points

In Fig. C.4, one can see linear and non-linear analyses of the cantilever beam produced in Comsol (respectively a, d, g and b, e, h) and in the author's code (c, f, i). The pink beams in the results of the author's code represent the linear analysis, while the purple beams correspond to the non-linear analysis. The Comsol solves provide similar behaviour and corresponding values for displacements fields compared to the author's results.

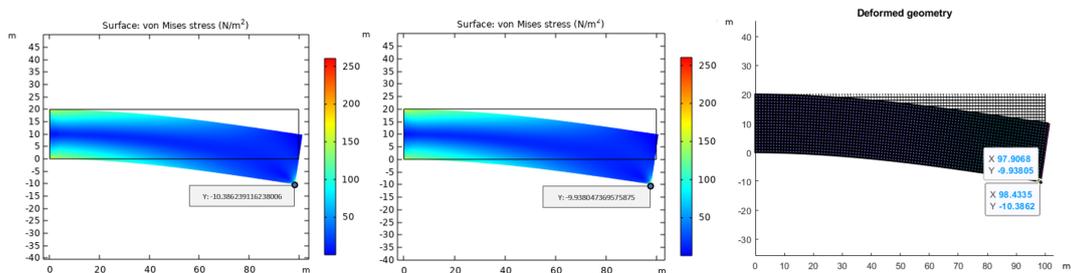
The deflections of the cantilever beams were the same for both the MATLAB FEM package as well as the Comsol model. The results can be seen in Fig. C.4. The stiffness matrices that Comsol uses are of a different size, even though the shape functions are set to bilinear and the quads have 4 nodes each. For example, a beam with 8 nodes should have (for linear shape functions) a stiffness matrix of 16×16 . Comsol produces a 28×28 matrix, with very different values in them. Nevertheless, the solution obtained with these different linear systems is identical with the author's code up to the numerical tolerance. From these results it can be concluded that the written code works and provides the correct deflections for a given load case.



(a) Linear analysis (b) Non-linear analysis (c) Pink: Linear analysis, Purple: Non-linear analysis

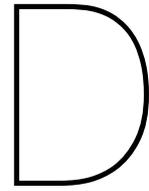


(d) Linear analysis (e) Non-linear analysis (f) Pink: Linear analysis, Purple: Non-linear analysis



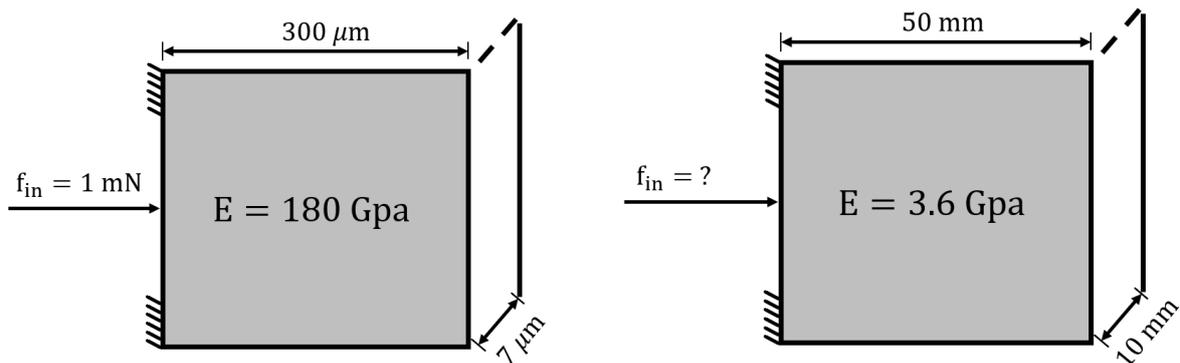
(g) Linear analysis (h) Non-linear analysis (i) Pink: Linear analysis, Purple: Non-linear analysis

Figure C.4: Deformed beam geometries were obtained using linear and non-linear analysis in Comsol Multiphysics in the first two columns. Solutions created by the self-written code in the third column, for three elements (respectively a,b,c), for twenty elements (respectively d, e, f), and for two-thousand elements (respectively g, h, i). All self-calculated deflections are in accordance with the values Comsol calculated.



Scaling with Buckingham Pi theorem

The Buckingham Pi method is a key theory in dimensional analysis. The method, as described in for example White (2016), describes a way of scaling physical phenomena. In the inverter design objective, we are interested in a design domain that is 3D printable, but conforms to the same relative displacement relationships as the problem in Pedersen et al. (2001). This scaling is done by keeping so-called pi groups equal across the different dimensions.



(a) The dimensions used in the design objective of Pedersen et al. (2001).

(b) The dimensions used in the design objective of Section 2.5.

Figure D.1: The difference in dimensions for a similar inverter design objective.

The design domain for the inverter problem is governed by 4 variables, assuming that the design domain is square. These are the Young's modulus E , force f_{in} , side length L and thickness t , which make use of 2 base units, m and N. The pi theorem states that this problem is therefore described by $4 - 2 = 2$ pi groups (White, 2016). The design domain that is solved in this paper has a different surface area to thickness ratio than Pedersen et al. (2001), and therefore it will not be possible to keep these two pi groups equal for the different design domains. To still solve a similar problem a pi group Π , containing all variables is created in the following way. A solution is to be found for a, b, c and d , in

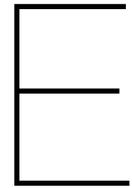
$$\Pi = E^a f_{in}^b L^c t^d, \quad (D.1)$$

such that Π is dimensionless. A process to do this is described in White (2016), which for this problem yields:

$$\Pi = \frac{ELt}{f_{in}}. \quad (D.2)$$

If the variables in Fig. D.1a are filled in in Eq. (D.2), the value that is found is $\Pi = 378 \cdot 10^3$. Now this value can be used to determine the value of f_{in} in Fig. D.1b if all other variables are filled in for Eq. (D.2). This yields a force of $f_{in} = 4.76$ N.

There are two springs attached to the design domain. The first one is attached to the DOF on which force f_{in} is applied, to constrain the input displacement and the second one is attached to the DOF on which the output displacement is measured, to mimic the work piece stiffness. The ratio between these springs happens to be $5/4 \cdot 10^{-3}$ according to Pedersen et al. (2001). The maximum displacement was set to $5/300$ of the design domain, which for the case in this paper would mean $5/300 \cdot 50 = 0.83$ mm. To constrain the input displacement for this length and an input force of 4.76 N, a spring is needed of 5.55 kN m^{-1} . Keeping the spring ratio constant, yields an output spring of stiffness $k_{out} = 6.94 \text{ N m}^{-1}$.



Other solutions to the inverter problem

As described in the discussion section, the load case and design objective combination for the inverter numerical example can result in instabilities. Approximate analysis is based on points very close to the undeformed configuration. It is beneficial for the optimizer to create structures with high values of orders derivatives of u_{out} to λ near the undeformed configuration, because the approximation of u_{out} will be a higher value. This happens to be the objective of the design problem, causing the optimizer to create such structures.

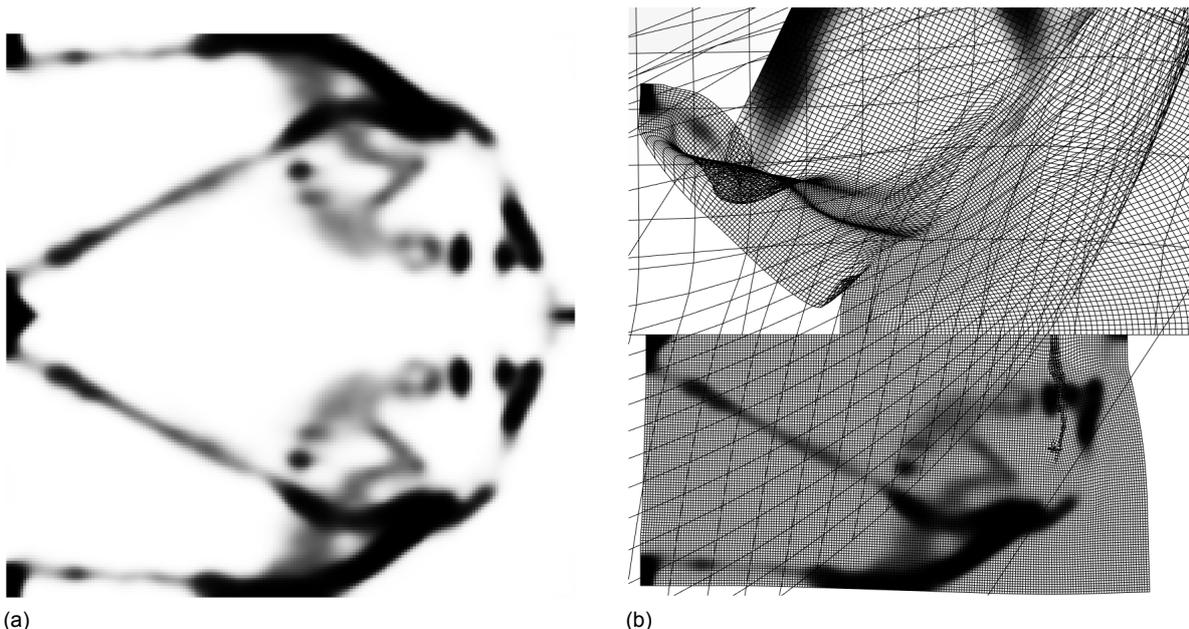


Figure E.1: Structure created by the optimizer without the $\|\mathbf{r}_0\|$ constraint, in a) the topology and in b) the deflections according to approximate analysis (Top) and non-linear analysis (Bottom).

One way of making the design problem more stable is to increase the value of the spring connected to u_{out} . This spring prevents excessive deformations of u_{out} as there is already an intrinsic spring stiffness connected to the DOF. On top of that, the rotations of the beams of inverters will be lower, as there is more force needed to overcome the spring force at u_{out} . This is beneficial for approximate analysis.

To try and compare this approach, the exact same load case, mesh and materials are used as in the inverter section. Only the spring attached to the degree of freedom u_{out} is increased fivefold to $k_{out} = 34.7 \text{ N m}^{-1}$. First, the optimization routine was tried without the solid residual constraint. These designs did not converge, as the optimizer still made designs that had unrealistic deformations, which lead to

weird designs with unwanted features. This can be seen in Fig. E.1, which shows such a design at 47 iterations and the approximate deflections compared to non-linear deflections. Therefore the same solid residual constraint was added as in the inverter section. The approximate analysis was done with $n = 5$ and $SF = 50$.

	f_n	ϵ_f	r_n	$\#S_a$
Linearly based design	0.69	63.1	542.3	2.0
Non-linearly based design	1	0	9.00E-03	187.0
Approximation-based design	0.97	0.61	16.12	14.9

Table E.1: Performance for the designs in Fig. E.2.

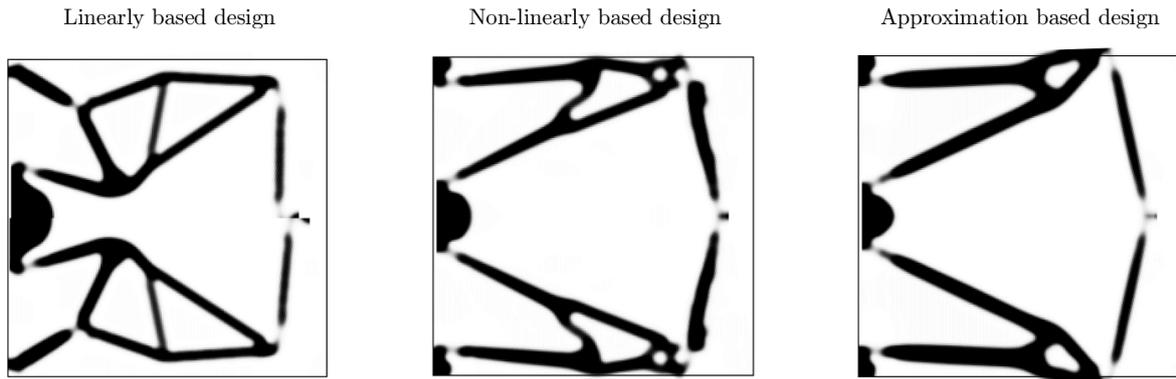


Figure E.2: A comparison with the best performing solutions of three optimization routines with different structural analyses. For each design the top half is analyzed the same way as in its optimization routine, the bottom half is analyzed with non-linear analysis.

As can be seen in the Table E.1, which shows the results for the designs in Fig. E.2, the approximation-based design is on par with the non-linearly-based design, performance-wise. The error in the objective estimation of approximate analysis, when compared to the non-linear analysis of the approximation-based design, is 0.61%, which also happens to be the error of the deflection at u_{out} . That is a more than 200 times more accurate approximation of u_{out} than linear analysis does for its design.

Another note must be made about the deformation field of the void area according to approximate analysis. This can be seen more clearly in Fig. E.3. The lack of inverted or highly deformed voids for approximate analysis is beneficial for the design updates, as such elements create wrong sensitivities (Dijk et al., 2014). Fig. E.4 shows even more deformed elements, which can be explained by the lack of any response function mitigating these inverted elements.

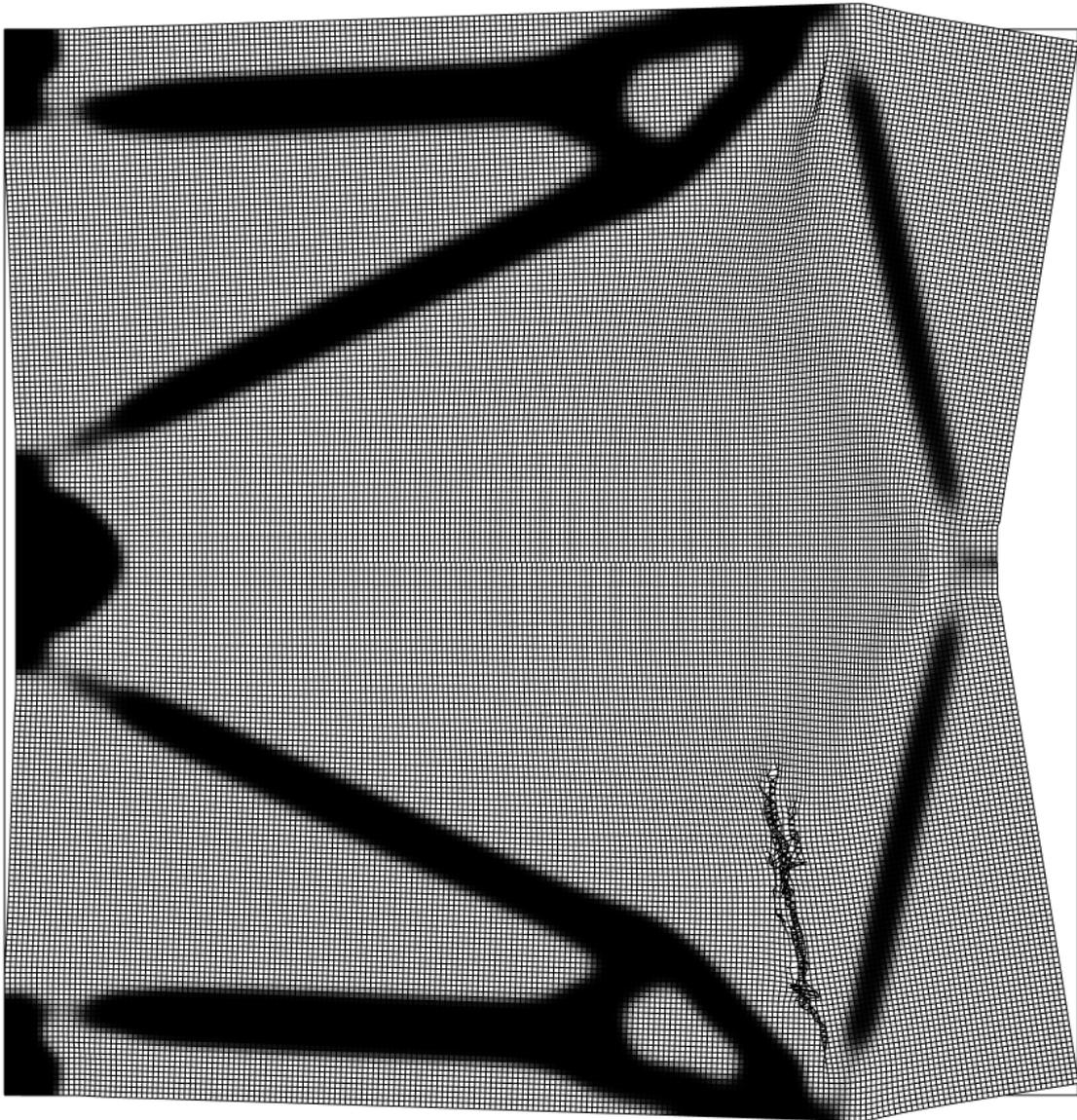


Figure E.3: A bigger plot of the deformations of the approximation-based design. The top half is analyzed with approximate analysis, the bottom half is analyzed with non-linear analysis.

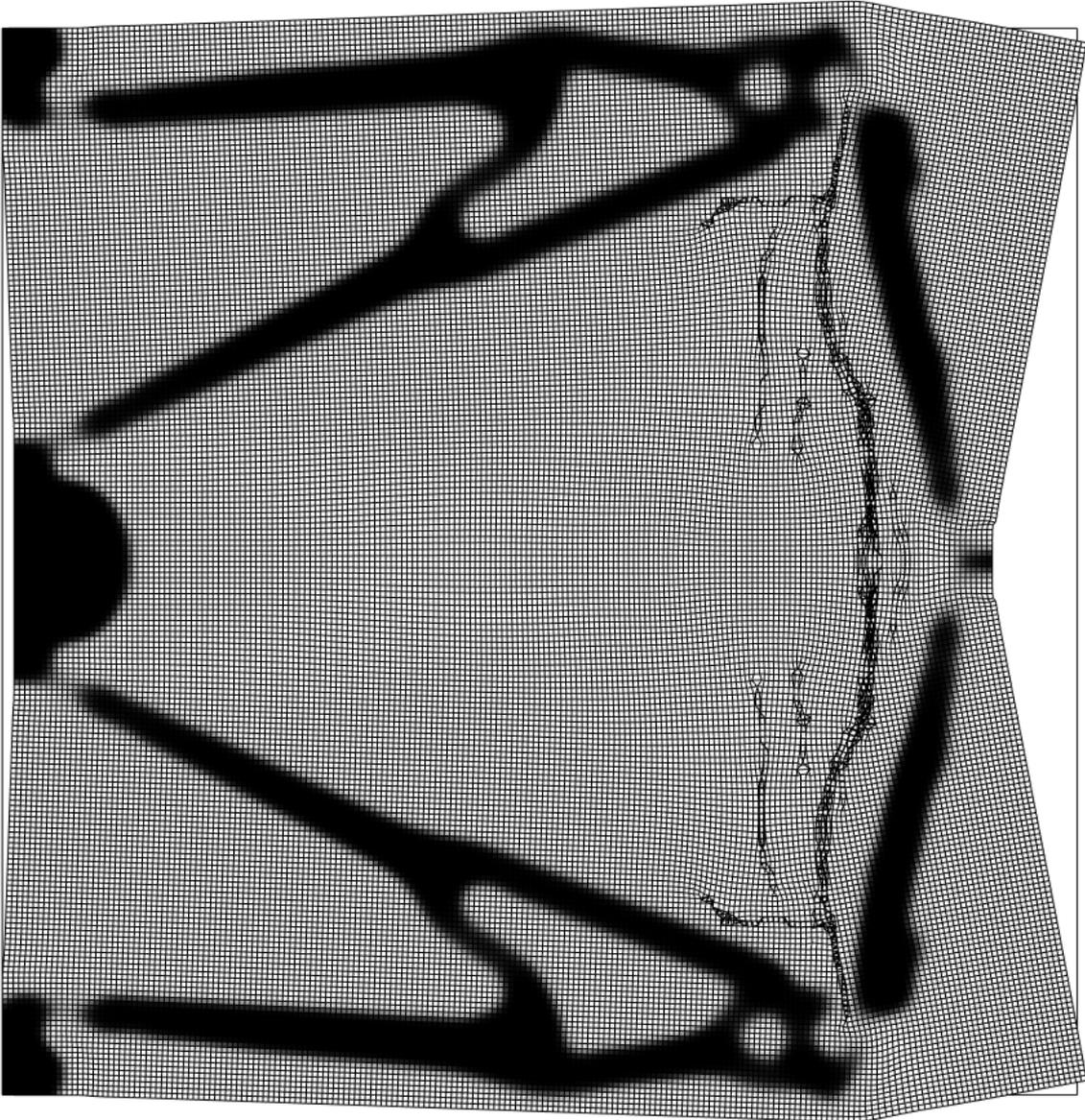


Figure E.4: A bigger plot of the deformations of the non-linearly based design, analyzed with non-linear analysis.

Bibliography

- Amir, O. (Jan. 2015). "Revisiting approximate reanalysis in topology optimization: on the advantages of recycled preconditioning in a minimum weight procedure". In: *Structural and Multidisciplinary Optimization* 51.1, pp. 41–57. DOI: 10.1007/s00158-014-1098-7.
- Arora, J. S. and E. J. Haug (Sept. 1979). "Methods of Design Sensitivity Analysis in Structural Optimization". In: *AIAA Journal* 17.9, pp. 970–974. DOI: 10.2514/3.61260.
- Bendsøe, M. P. and O. Sigmund (2004). *Topology Optimization*. Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-05086-6.
- Bluhm, G. L., O. Sigmund, and K. Poullos (Mar. 2021). "Internal contact modeling for finite strain topology optimization". In: *Computational Mechanics* 2021 67:4 67.4, pp. 1099–1114. DOI: 10.1007/s00466-021-01974-x.
- Buhl, T., C. B. W. Pedersen, and O. Sigmund (2000). "Stiffness design of geometrically nonlinear structures using topology optimization". In: *Structural and Multidisciplinary Optimization* 19.2, pp. 93–104. DOI: 10.1007/s001580050089.
- Dijk, N. P. van, M. Langelaar, and F. van Keulen (Aug. 2014). "Element deformation scaling for robust geometrically nonlinear analyses in topology optimization". In: *Structural and Multidisciplinary Optimization* 50.4, pp. 537–560. DOI: 10.1007/s00158-014-1145-4.
- Gogu, C. (Jan. 2015). "Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction". In: *International Journal for Numerical Methods in Engineering* 101.4, pp. 281–304. DOI: 10.1002/NME.4797.
- Howell, L. L., S. P. Magleby, and B. M. Olsen (Feb. 2013). *Handbook of Compliant Mechanisms*. Ed. by L. L. Howell, S. P. Magleby, and B. M. Olsen. Wiley, pp. 7–14. DOI: 10.1002/9781118516485.
- Koppen, S., M. Langelaar, and F. Keulen (2021). "A simple and versatile formulation for topology optimization of flexures". In: *Mechanism and Machine Theory*(In review).
- Koutsovasilis, P. and M. Beitel Schmidt (Mar. 2010). "Model order reduction of finite element models: improved component mode synthesis". In: *Mathematical and Computer Modelling of Dynamical Systems* 16.1, pp. 57–73. DOI: 10.1080/13873951003590214.
- Lahuerta, R. D., E. T. Simões, E. M. Campello, P. M. Pimenta, and E. C. Silva (2013). "Towards the stabilization of the low density elements in topology optimization with large deformation". In: *Computational Mechanics* 52.4, pp. 779–797. DOI: 10.1007/s00466-013-0843-x.
- Lazarov, B. S. (2014). "Topology optimization using multiscale finite element method for high-contrast media". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8353 LNCS. Springer Verlag, pp. 339–346. DOI: 10.1007/978-3-662-43880-0{_ }38.
- Maute, K. and E. Ramm (Oct. 1995). "Adaptive topology optimization". In: *Structural optimization* 1995 10:2 10.2, pp. 100–112. DOI: 10.1007/BF01743537.
- Pedersen, C., T. Buhl, and O. Sigmund (Apr. 2001). "Topology synthesis of large-displacement compliant mechanisms". In: *International Journal for Numerical Methods in Engineering* 50.12, pp. 2683–2705. DOI: 10.1002/nme.148.
- Wallin, M., N. Ivarsson, and D. Tortorelli (Mar. 2018). "Stiffness optimization of non-linear elastic structures". In: *Computer Methods in Applied Mechanics and Engineering* 330, pp. 292–307. DOI: 10.1016/j.cma.2017.11.004.
- Wang, F., B. S. Lazarov, O. Sigmund, and J. S. Jensen (July 2014). "Interpolation scheme for fictitious domain techniques and topology optimization of finite strain elastic problems". In: *Computer Methods in Applied Mechanics and Engineering* 276, pp. 453–472. DOI: 10.1016/j.cma.2014.03.021.
- White, F. m. (2016). *Fluid Mechanics*. Mc Graw Hill Education, pp. 266–273.
- Yoon, G. H. and Y. Y. Kim (Apr. 2005). "Element connectivity parameterization for topology optimization of geometrically nonlinear structures". In: *International Journal of Solids and Structures* 42.7, pp. 1983–2009. DOI: 10.1016/j.ijsolstr.2004.09.005.

