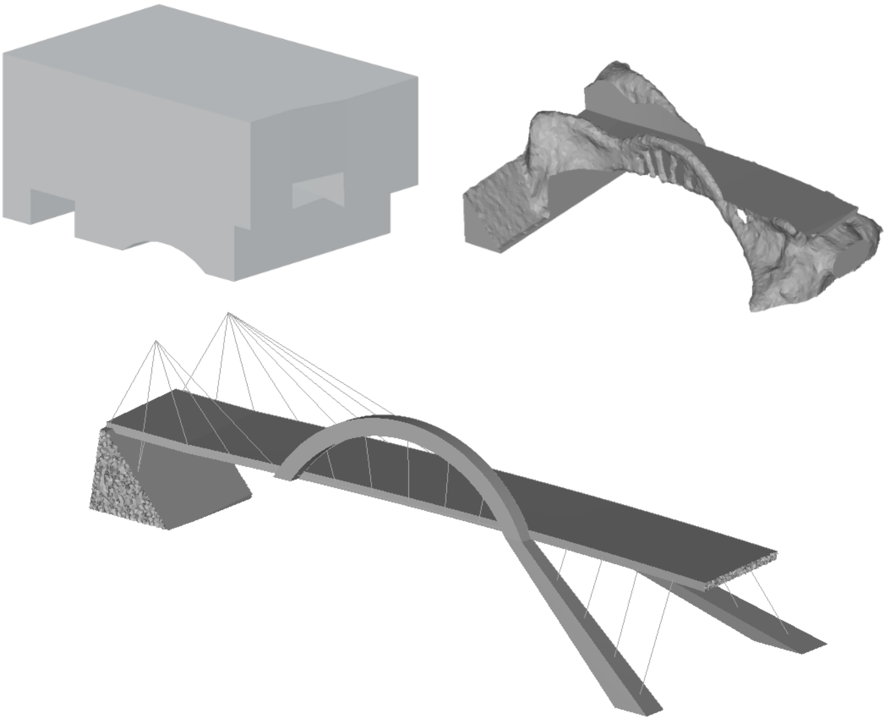
CHALMERS



Structural Form Optimisation

Methods of numerical optimisation and applications on civil engineering structures

Master of Science Thesis in the Master's Programme Structural Engineering and Building Performance Design

JOHANNES LUNDGREN
CHRISTOFFER PALMQVIST

Department of Civil and Environmental Engineering
Division of Structural Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2012
Master's Thesis 2012:97

MASTER'S THESIS 2012:97

Structural form optimisation

Methods of numerical optimisation and applications on civil engineering structures

Master of Science Thesis in the Master's Programme Structural Engineering and Building Performance Design

JOHANNES LUNDGREN

CHRISTOFFER PALMQVIST

Department of Civil and Environmental Engineering
Division of Structural Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2012

Structural Form Optimisation
Methods of numerical optimisation and applications on civil engineering structures

Master of Science Thesis in the Master's Programme Structural Engineering and Building Performance Design
JOHANNES LUNDGREN
CHRISTOFFER PALMQVIST

Examensarbete / Institutionen för bygg- och miljöteknik,
Chalmers tekniska högskola 2012:97

Department of Civil and Environmental Engineering
Division of Structural Engineering
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Design domain (left), optimised topology (right) and interpreted result (bottom) for bridge.

Department of Civil and Environmental Engineering Göteborg, Sweden 2012

Structural Form Optimisation
Methods of numerical optimisation and applications on civil engineering structures

Master of Science Thesis in the Master's Programme Structural Engineering and Building Performance Design

JOHANNES LUNDGREN
CHRISTOFFER PALMQVIST
Department of Civil and Environmental Engineering
Division of Structural Engineering
Chalmers University of Technology

ABSTRACT

The procedures for structural design used by civil engineers today often result in suboptimal solutions. However, in other industries optimisation is already part of the design process. In this thesis, numerical optimisation methods are investigated and applied on civil engineering structures, to find out if these methods can be part of the design procedures that brings forth the grand designs of tomorrow.

Several types of structural optimisation are investigated, including size, shape and topology optimisation methods. Most focus is put on the topology optimisation, where methods as SIMP, ESO and homogenisation are reviewed. The three optimisation types are applied on civil engineering structures, to investigate the usefulness of structural optimisation. Smaller structures are optimised, as well as a large-scale bridge.

The results presented in this report shows that structural optimisation is useful for structures, where simple linear material models are sufficient to represent the behaviour of the structure. For civil engineering structures, the optimisation methods have yet to be further developed for truly becoming useful in practice, mainly due to the lack of proper material models for the behaviour of concrete in the available optimisation softwares.

Key words: structural optimisation, topology optimisation, shape optimisation, size optimisation, SIMP, ESO, HyperWorks, Optistruct, Matlab

Optimering av konstruktioners form
Metoder för numerisk optimering och tillämpningar på konstruktionstekniska problem
Examensarbete inom Structural engineering and building performance design
JOHANNES LUNDGREN
CHRISTOFFER PALMQVIST
Institutionen för bygg- och miljöteknik
Chalmers tekniska högskola

SAMMANFATTNING

Dagens metoder för dimensionering av bärande konstruktioner inom byggindustrin leder ofta till suboptimala lösningar. I andra industrier, såsom flyg- och bilindustrin, är dock optimering redan än del av dimensioneringsprocessen. I den här rapporten undersöks metoder för numerisk optimering vilka också tillämpas på konstruktionstekniska problem för att undersöka om dessa optimeringsmetoder kan bli en del av nya metoder för dimensionering av bärande konstruktioner.

Flera optimeringstyper undersöks, såsom storlek-, form-, och topologioptimering. Stort fokus ligger på topologioptimering, där metoder som SIMP, ESO och homogeniseringsmetoden undersöks. De tre optimeringstyperna, storlek-, form-, och topologioptimering, appliceras på konstruktionstekniska problem för att undersöka huruvida dessa är användbara för optimering av konstruktioner inom byggindustrin. De appliceras på mindre problem, såväl som på en större bro.

Rapporten visar att optimering kan vara användbart, då enkla linjärelastiska materialmodeller är tillräckligt för att beskriva en strukturs verkliga beteende. För konstruktionstekniska problem inom byggindustrin behövs dock mer forskning innan de optimeringsmetoder som behandlas i den här rapporten kan komma till verklig användning. Framför allt beror detta på att det idag saknas modeller som på ett bra sätt beskriver betongens beteende, i de tillgängliga programvarorna.

Nyckelord: Optimering av konstruktioner, topologioptimering, formoptimiering storleksoptimering, SIMP, ESO, Hyperworks, Optistruct, Matlab

Contents

Preface

This report treats methods for numerical optimisation of structures. It has been carried out at the department of Civil and Environmental Engineering at Chalmers University of Technology. The work presented in this report is a part of the project TailorCrete, which Chalmers University of Technology is participating in. TailorCrete is a European research project, led by the Danish Institute of Technology, with the aim of developing new methods for manufacturing unique concrete structures in an industrialised way without the need of manual labour-intensive processes.

Many thanks go to our supervisor Mario Plos as well as our opponents Eyrún Gestsdóttir and Tómas Guðmundsson, for having assisted us in our work and contributed with their thoughts. We would also like to thank Altair Engineering for having provided us with licenses for HyperWorks.

Göteborg May 2012

Johannes Lundgren

Christoffer Palmqvist

1 Introduction

1.1 Background

The education of structural engineers is in structural design courses mostly about designing and analysing straight prismatic beams, columns and slabs. It is certainly easier to design and analyse such simple geometries and it can also be seen that it is very much appreciated by the contractors. In case design tasks is given involving curved or tapered beams the method is to initially follow some design rules, analyse the design and make changes if the design doesn't fulfil the requirements or fulfil them too well. This is an iterative, often time consuming process and as a consequence a design is normally accepted as long as it is above and reasonably close to the requirements. For a curved or tapered beam the accepted design is almost always suboptimal; there are so many parameters defining only the geometry thus finding the optimal design requires a large number of iterations. Just for a straight rectangular beam of reinforced concrete there is the height, width, steel area and steel position(s) but also concrete and steel strengths. In general, to obtain an optimised design it is necessary to use a computer aided optimisation technique.

When designing a structure the question should be: How should the load be transferred from point A to point B in the most efficient way? Not: Which size should the beam be? That is a difficult question for an engineer who is trained using rectangles and straight lines, but if the question can be put to a computer, it may also be able to give the answer. In this project we aim to expand our thinking using the aid of modern technology. As will be seen, optimised structures can have a quite complex geometry, at least if no constraints are set to keep it simple. This is certainly pushing the traditional production methods to their limits; the benefits of the optimised structures are easily eaten up by increased production costs. This is especially apparent for concrete structures where formwork and reinforcement already for simple geometries make up a big part of the cost. There are however much research in this area, see Williams et al. (2011) and Fall (2011) for some results of the on-going research project TailorCrete. It is an European collaboration working with new design tools, automatic formwork manufacturing, fibre reinforcements and self-compacting concrete; technologies that are needed for optimised concrete structures to compete in terms of cost and producibility.

1.2 Purpose

The purpose of this thesis is to survey the research field of structural form optimisation, specifically methods for how to numerically find the optimal form of a structure, and as a part of that present the underlying theory and mathematics in short terms. Different types and methods for optimisation are to be presented including a critical review where weaknesses and strengths are pointed out. The purpose is also to survey the possibilities of applications in the area of civil engineering by applying the methods to some simple illustrative examples. Comparisons between optimisation types, optimisation parameters and used tools are also to be made. One part of the survey of possible applications is to investigate the usefulness of structural optimisation of large scale civil engineering objects.

1.3 Method

The first theoretical part is conducted by a literature study of both textbooks and research papers. While there exist several methods with very different starting positions, focus is on common, well described methods and desirably with available implementations. The simple examples comparing the different optimisation methods and varying input parameters rely on available implementations, however modifications are done whenever needed, if possible. The final large scale problem is defined to resemble a road bridge with loads and geometrical constraints according to Swedish regulations. The structure is then optimised and developed in several steps using different numerical methods but also engineering judgment to compensate for conditions that are not possible or very difficult to include in the numerical optimisation.

2 Theory

2.1 Introduction to Structural Optimisation

In the beginning of the last century the Australian engineer A.G.M. Michell published his paper on structural optimization; *The Limits of Economy of Material in Frame-structures*, Michell (1904). Although this area of research was not given very much attention until the development of computers, Michell's work marked the beginning of it. Analytically he derived the optimum layout of some elementary frame structures but to optimise a real structure for arbitrary conditions numerical methods was necessary. The modern structural optimisation had to wait another half century and its evolution is described by Vanderplaats (1993). With the finite element analysis, traced back to the early 1940's, which opened up for rational assessment of proposed designs, it wasn't until in the early 1960's that L.A. Schmit combined the finite element analysis with nonlinear numerical optimization to what he then called structural synthesis, Schmit (1960). Yet only very simple structures were possible to be dealt with, like trusses and frames with a small number of members. The development of structural optimisation is of obvious reasons connected to the readily increasing power of the computer. From discrete structures the progress lead to the more general shape and topology optimisations of continuum structures.

Structural optimisation is today a wide concept and the outcome of a structural optimisation usually varies enormously due to a variety of possible constraints and optimisation aims. Especially if one are to include economical and aesthetical aspects it should be clear to any engineer that the stiffest structure may certainly not be the cheapest. An introduction to many of the used concepts and important applications of structural optimisation are given by Pedersen (2003). The aim of the optimisation is most often to minimise or maximise some kind of physical property of the structure, e.g. to minimise strain energy (equal to maximise global stiffness), minimise deflection of some chosen point or minimise the maximum stress, which are all load dependent properties. Other physical properties that can be used as optimisation objectives are volume, weight and perimeter area which (for small deformations) are all load independent properties. The constraints can be limiting the same properties as just listed, especially the global properties such as weight and strain energy is often used as constraints as this is computationally cheaper than constraining a local property such as the stress. This does not mean that the stress constraint is unimportant. Apart from the obvious possibility of avoiding the material becoming overstressed, using stress constraints may lead to different topological solutions whenever multiple load cases are present or the material response is different in tension and compression, Duysinx et al. (2008). Another type of constraint is allowable values for the design variables and the use of a spatial design domain wherein the solution is searched. In mathematical terms an optimisation can be expressed as

$$
\begin{aligned}
&min_x\, f(\boldsymbol{x}) \\
&subject\ to\!: \begin{cases} g(\boldsymbol{x}) < c \\ x_i \in \boldsymbol{\phi} \\ equilibrium\ condition \end{cases}
\end{aligned}
\tag{1}
$$

where $\boldsymbol{x}=[x_1\ x_2\ \dots\ x_n]$ are the design variables, $f(\boldsymbol{x})$ is the objective function that will be minimised, $g(\boldsymbol{x})$ is any response that is limited by the constraint c and $\boldsymbol{\phi}$ is the range of allowed values for the design variable. The optimised structure must also fulfil equilibrium conditions, this is usually provided by a finite element analysis which is also needed in order to calculate most responses used as objective function or being limited by constraints.

The design variables can be anything that defines the structure, Pedersen (2003). For a truss it can be the cross sectional areas of the members, the positions of the nodes or both. For a beam it can be the shapes of some of its sides and possible holes or whether each element in the finite element model should be represented as filled with material or not. For a layered composite material it can be the number of layers, thicknesses of layers, base material of each layer and for anisotropic materials the directions of the layers. Additionally, the design variables can be either continuous or discrete. Depending on the chosen variables and constraints the problem will be one of the three types: size, shape or topology optimisation. These are described in the following sections. Figure 1 shows a schematic overview over the different types and methods for structural optimisation treated in this report.

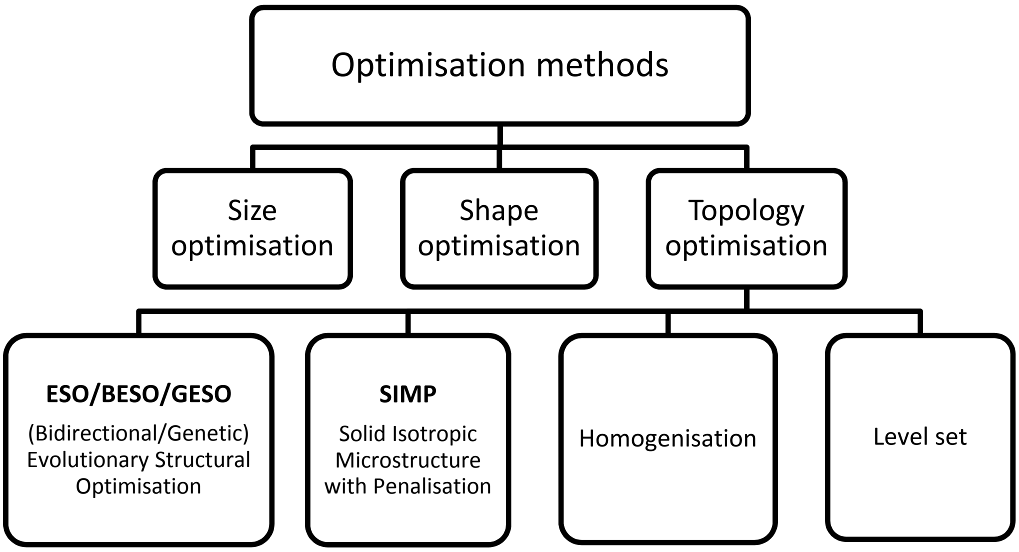


Figure 1 Schematic over different optimisation types and methods.

For a large structure, or for a model with a fine mesh, the amount of design variables is so large that it is practically impossible to solve the optimisation problem by trying every possible combination of the variables[1]. The problem is also highly non-linear so an iterative algorithm has to be used, Christensen and Klarbring (2008). Many solution methods have been developed to solve these large problems more efficiently, ranging from simple intuitive algorithms often trying to simulate a natural process, to advanced strictly mathematical algorithms. One simple method is to evaluate the current stress distribution of the structure and redistribute material to where it is most needed, i.e. from low stress to high stress areas. A more sophisticated mathematical method includes a sensitivity analysis in which the gradient of the objective function

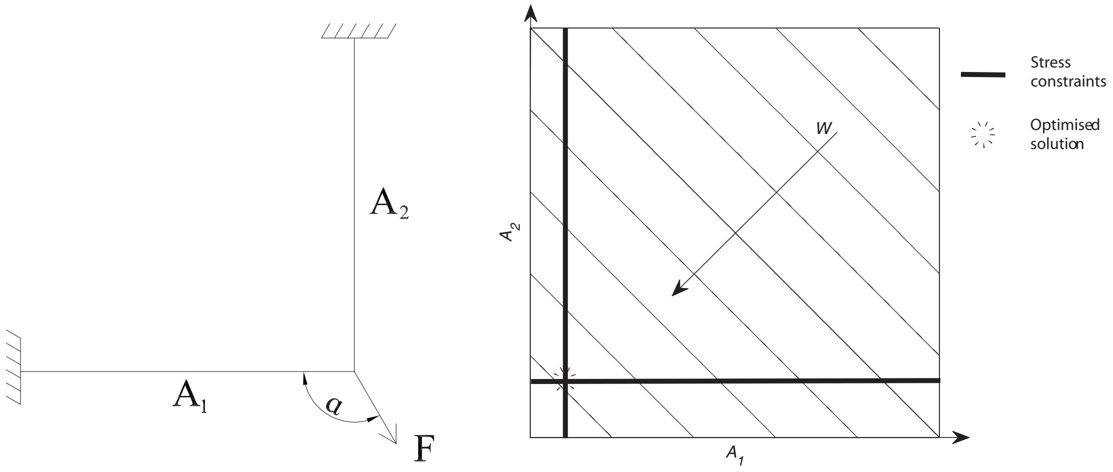
[1] For a 100 × 100 mesh with the discrete design variable $\mathbf{x};\ x_i \in \{0,1\}$, defining each element as either filled with material or not, the number of possible structures are 2^{10000}, each for which a FE analysis should be done.

over the design variables is calculated. The design is then updated based on the gradient, in direction as well as magnitude. A major problem for all optimisation problems is when the objective function is not truly convex downward (for minimisation), i.e. the function has not only a global minimum but also one or more local minima. A good algorithm must have the ability to see beyond, and don't stop at, such a false optimum.

2.2 Size optimisation

In size optimisation, or optimisation of discrete parameter systems as it is often called, the sizes of already defined structural members are varied to find the optimal design. This implies that before such an optimisation can be done, the structure to be analysed must be defined. A common way to perform the optimisation is to minimise the weight of the structure, under a stress constraint, Christensen and Klarbring (2008). Other constraints, such as constraints on stiffness or deformations, can be applied as well.

A typical application for the size optimisation is an already defined truss, where each bar in the truss is handled as one element in the analysis, i.e. there is no need to further mesh such a truss structure. The design variables consist of some sizing property of the structural elements. For the example with the truss, the design variables would typically be the diameters or the cross-sectional areas of the bars, as used by Christensen and Klarbring (2008). The variables can either be continuous or a discrete set of available bar profiles. If the structure is very small, let's say a truss with only two bars, the optimum can easily be calculated and the solution space visualised. By having the cross-sectional areas as the design variables and a limit on the allowable stress in the bars and the weight of the system as the objective function which is to be minimized, the objective function is a sloping plane with contour lines as in Figure 2. It is obvious for this simple example that the optimal solution is found when the sum of A_1 and A_2 is as small as possible, which is in at the crossing of the stress constraints.



(a) Example of truss for size optimisation.

(b) Solution space for the example truss.

Figure 2 Example of truss for size optimisation and its solution space.

For larger systems, optimising in the manner explained above is not practically possible, and more refined algorithms need to be used to find the optimum. Theoretically, it would of course be possible to find the optimum by comparing the results for all possible combinations of element sizes, but that would be computationally too time-consuming. Hager and Balling (1988) gives an example with 16 bars and 194 different possible sizes for the bars, which would give a total of $4 \cdot 10^{36}$ possible combinations. Even if a computer was able to try one million combinations per second, the computation time would still be several million years. Instead algorithms need to be used, which gradually converges towards an optimum. By doing this, the computation time for the example with 16 bars above would be a matter of minutes or seconds.

During the size optimisation, no elements are added or removed. This gives that the topology of the design is kept throughout the optimisation process. There is however the possibility, that the diameter, or area, of a bar gets so small, that it from an engineering point of view hardly exists.

2.3 Shape optimisation

In shape optimisation, the shape of structural members including eventual voids within them are optimised, Haftka and Grandhi (1986). No new connections or holes can appear and no existing members or voids can disappear. A shape optimisation can be used when no change in topology is needed or wanted, e.g. to refine the solution from a topology optimisation. It is especially good at eliminating high stress concentrations. Early work on this subject often used the position of the nodes as design variables to explicitly move the border. This option gives a high number of design variables which also can define complex shapes, as complex as is possible with the fineness of the border discretisation. Although only the positions of the border nodes are used as design variables also the interior mesh needs to be refined in order to keep it at a good quality. This is however sometimes not possible, especially at the boundaries which can start to oscillate, helped by the decreasing accuracy of the finite element analysis when the border elements get distorted, as seen in Figure 3. To overcome this problem the nodes at the border of the mesh should always be more than the design variables. This is accomplished by a parameterisation of the border; the parameters are used as design variables and a new mesh is generated in each iteration. Polynomials can be used with the coefficients as design variables, however the same problem occur when using high order polynomials. It has been found best to use splines (piecewise defined smooth function) build up by sections of low order polynomials. Other possible parameterisations are to linearly combine a set of shape functions or use piecewise linear functions.

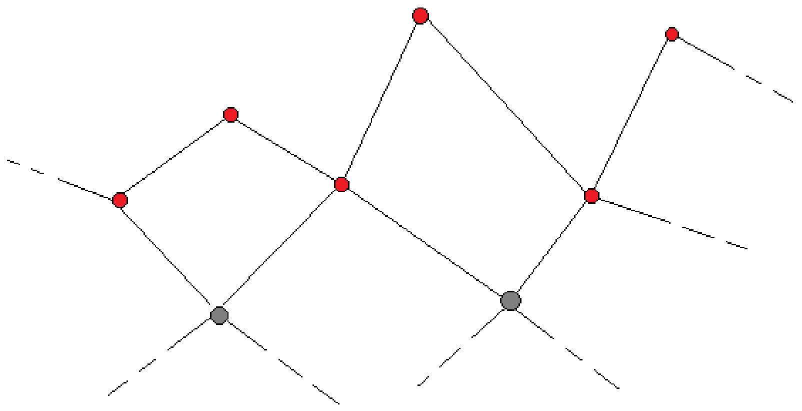


Figure 3 When the position of the boundary nodes (red) are used as design variables the boundary can easily start to oscillate and become distorted.

By using parameterised functions to define the shape there is now the problem of transforming this model into the finite element model for analysis, Wall et al. (2008). The finite element model will always be an approximation whenever smooth borders are used in the geometry model. This can be solved by using isogeometric analysis based on NURBS (non-uniform rational B-splines) in which the mesh is capable of exactly represent the modelled geometry.

One kind of shape optimisation is the topography optimisation mostly used to optimise embossing pattern for thin plates. By using the same approach as above the design variables control the out-of-plane deformation of a plate.

2.4 Topology optimisation

Topology optimisation is the most general type of structural optimisation, widening the problem to the study of the structures topology, i.e. not only to the size and shape of the members but also how the members are connected to each other. The methods used can be divided into those who operate on the state of each element in the finite element mesh, i.e. if it consists of material, void or an intermediate, and those who use lines or surfaces to define the material boundary. This section is focused on the former which are the methods most used. However, research is also done on the latter methods, of which one is described at the end of this section.

When optimising the topology of a structure it is natural to demand the solution to consist of clearly separated material and void, preferably with a material distribution possible to manufacture. This implies the use of a discrete variable for each element state, either material or void. On the other hand, using continuous variables open up for the use of efficient mathematical programming schemes, Bendsoe and Sigmund (2003). The homogenisation method and the SIMP method described below handle this in two similar ways: the former by allowing for composite materials consisting of infinite small voids and the latter by allowing for intermediate densities without any physical meaning. Grey areas that appear can be removed by applying a penalty to these intermediate states. It is still possible to use discrete variables, Beckers (1999), though these methods are so far inefficient or restricted to certain problem formulations, Bendsoe and Sigmund (2003).

The issue with the most general topology optimisation problem formulation is its lack of a true solution, Bendsoe and Sigmund (2003). This is due to the fact that refining the mesh, for a given problem, will always give rise to a topological different and better solution, at least when optimising the average stiffness, which is often the case. Instead of giving a better numerical approximation and a smoother boundary the structural members become smaller and smaller, a process that can continue indefinitely. Several techniques are available to avoid this problem, either by adding additional constraints or by applying some kind of filter which effectively sets a minimum length scale for the structural members. The additional constraints can be on the density gradients or the total perimeter. Another solution is to allow for intermediate states where, for a 2D problem, these can be interpreted as thickness variations in a plate. For a 3D problem, no such simple interpretation exists. The issue with a lack of solution is thus resolved either by adding constraints to the original problem, or relaxing it by allowing intermediate states.

Yet another issue is due to bad numerical approximation in the finite element method for elements connected at the corners, Bendsoe and Sigmund (2003). The stiffness of these connections are overrated and the resulting topology consists of alternating material and void on a per element period; known as the checkerboard pattern, Figure 4. The same techniques described above, that assure the existence of a solution, will generally also solve this issue.

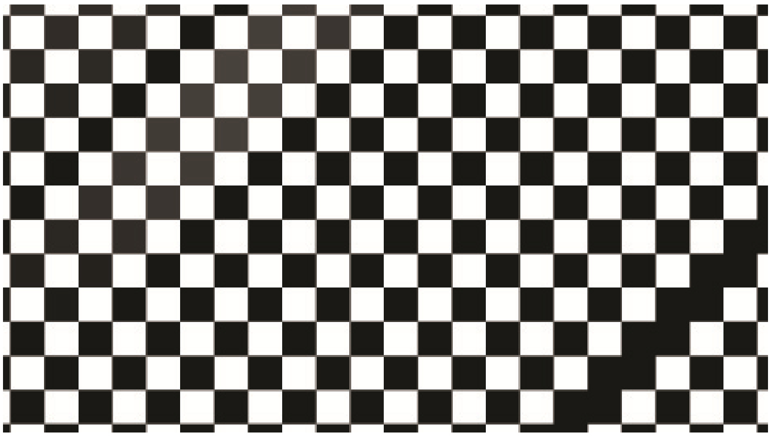


Figure 4 The checkerboard pattern.

2.4.1 Homogenisation

By relaxing the optimisation problem to include elements with intermediate densities it is possible to ensure the existence of a solution. The problem then is how to interpret this new material. The homogenisation method uses a composite material model, consisting of infinitely small but infinitely many unit cells consisting of material and void or two different materials, Bendsoe and Sigmund (2003). Although impossible to manufacture, as of today, the appearance of grey material is then not a problem from a physical point of view. This method has for ordinary engineering tasks been superseded by the SIMP method described below which disregard the problem of using unreal material as long as the result consists of mostly black and white.

The microstructure can be of any kind, but usually each cell is either layered or holds a quadratic or rectangular void, Bendsoe and Sigmund (2003). A so called ranked material uses two or more scales with one microstructure in each; each layer in the cell has another microstructure in itself. The design variables are defining the microstructure in each finite element and as the composite material is at least orthotropic, also the angular rotation of the microstructure becomes a design variable. From the design variables the stiffness tensor is calculated, which will be less than proportional to the effective density of the material except for all solid or all void elements. This is however not enough to drive the solution to all black and white, which is explained by the efficient use of material in the composite whilst ordinary homogenous material on the other hand, waste material in unstressed directions. Depending on the number of load cases and whether the design domain is two or three dimensional there exists an optimal microstructure which will give good results considering the objective function but also a lot of grey material. By using a suboptimal microstructure or excluding the rotation from the optimisation, grey material will be less efficient and the solution will contain less of it.

If analytical expressions for the stiffness tensor exist, as is the case for layered structures, the stiffness tensors are calculated when needed, Bendsoe and Sigmund (2003). For other microstructures a database of stiffness tensors first has to be generated, e.g. by a finite element analysis of the unit cell, one analysis for each combination of admissible design variables.

2.4.2 SIMP

Solid Isotropic Microstructure with Penalisation (SIMP) is the most widely used method for performing structural topology optimisation. It is used in almost all the available commercial optimisation softwares today, Rozvany (2009).

The design continuous variables in SIMP, $x \in [0,1]$, are one per element in the finite element discretisation and are commonly interpreted as a density ratio, Bendsoe and Sigmund (2003). This implies that before starting the optimisation process, a design domain, in which the final structure must be enclosed, is defined. At initialisation, this domain contains elements with an arbitrary density distribution. Elements with a full density are commonly represented as black elements whereas an element with zero density, which in any physical meaning would imply removal of the element, is represented by white. Intermediate densities are commonly represented by different shades of grey. This type of visualisation is used also in this text and works well for 2D problems, for 3D problems however, isosurfaces has to be used which doesn’t allow for the display of the entire density scale.

The element stiffness in the SIMP-model can be formulated as follows:

$$E_{ijkl}(x) = x^p E_{ijkl}^0,\ p > 1 \tag{2}$$

where E_{ijkl}^0 is a material property of the base material, typically the stiffness, p is the penalisation exponent and x is the design variable. By varying the penalisation it is defined how expensive grey material are. A penalisation of $p = 1$ means no penalisation and increasing the penalisation means intermediate densities get penalised i.e. the stiffness will no longer follow a linear relationship. This relationship between design variable and stiffness is showed in Figure 5. The reason for applying penalisation is to get a more black and white solution, which from an engineering point of view is preferable. In order to get a true black and white solution, a penalisation exponent greater than $p = 3$ is often required, Bendsoe and Sigmund (2003). It is however recommended, as pointed out by Rozvany (2009) to slowly increase the penalisation, starting from $p = 1$. The reason for this is to not rush the optimisation avoiding getting stuck at a local optimum. Despite the penalisation, grey material can be so much more efficient that it is often seen in the optimised result. In Equation 2, the variable p is an exponent but the p also appears as a factor when the expression is differentiated in the sensitivity analysis. In the literature p is referred to as both an exponent and a factor, in this text p is referred to as the penalisation exponent, or just the penalisation.

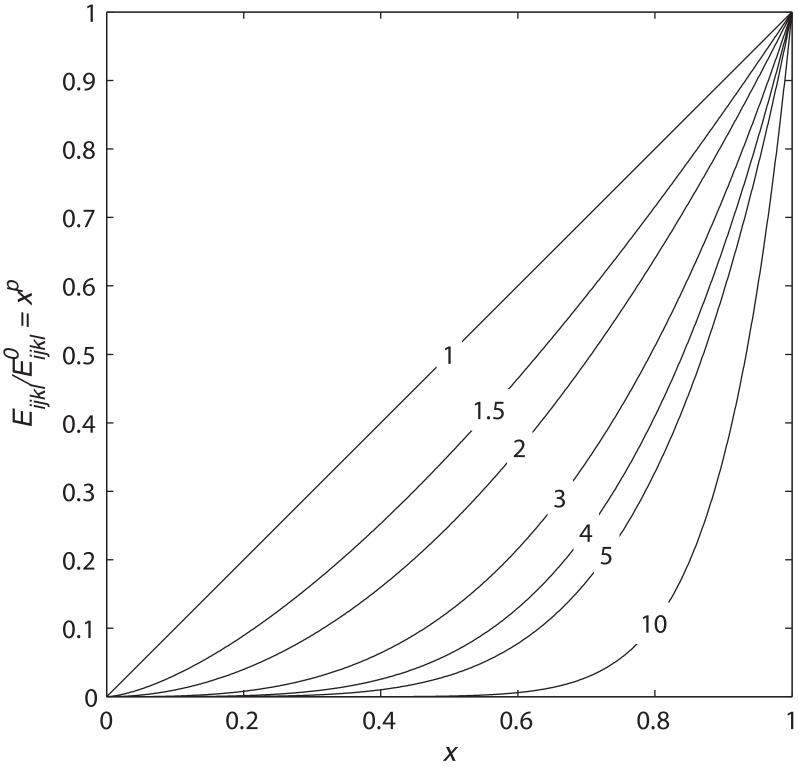


Figure 5 Intermediate densities decrease in efficiency for increased penalisations.

The weight is what is often constrained or used as objective function and for single material structures the weight is equivalent to the density scaled volume. The weight, W, is calculated by integrating the element densities $\rho(x)$ over the design domain **Ω**:

$$W = \int_{\Omega} \rho(x) d\Omega \qquad (3)$$

To update the density distribution a sensitivity analysis is performed during each iteration, as described in 2.1.

Although Rozvany (2009) states that it is likely that SIMP finds a true optimal solution, it has been shown by Stolpe and Svanberg (2001) that so is not always the case. They discovered that under certain conditions, SIMP would not reach the global optimum no matter how slowly the penalisation exponent was increased. The same authors also showed that SIMP, no matter how large the penalisation, sometimes fails to present a black and white solution.

2.4.3 ESO

One of the more widely spread methods for structural optimisation, is the Evolutionary Structural Optimization (ESO). ESO is based on the simple, and empirically based, principle that a structure evolves towards an optimum if the least stressed elements in each iteration are removed, Huang and Xie (2010). This implies, that before starting the analysis, an over-dimensioned structure and its boundary conditions must be specified.

Although stress originally was used as the criterion function, it has been shown that using the stress as criterion function results in structures with suboptimal stiffness. To correct this, the stress criterion was replaced by an elemental strain criterion, also called element compliance, by Chu et al. (1996).

To find the elements with the least element strains, a sensitivity analysis is performed where a sensitivity number is calculated for each element, i, according to the following equation

$$\alpha_i = \frac{1}{2}\boldsymbol{u}_i^T \boldsymbol{K}_i \boldsymbol{u}_i \;\; (i = 1, \dots n) \tag{4}$$

where $\boldsymbol{u_i}$ is the element displacement vector and $\boldsymbol{K}_i$ is the element stiffness matrix. After the sensitivity analysis, the elements are either fully removed or stays untouched based on the sensitivity numbers. No intermediate densities exist. This eliminates the need to penalize intermediate densities, which is necessary in other topology optimisation methods such as SIMP. ESO, in its original form, is also a one-way-method or a so called "hard-kill"-method. Elements can only be removed, no elements can be added. This renders the removal of elements a permanent action, since not even previously removed elements can be added again, Querin et al. (1998).

Another form of ESO is the additive evolutionary structural optimisation (AESO) method. Instead of removing elements with low sensitivity number or low stress as in ESO, elements are added next to the elements with high stress, Querin et al. (1998). This is, similarly to ESO, a one-way method. After an element is added, it cannot be removed. At least a minimum ground structure[2] is required from the beginning.

By combining the original ESO and AESO, Querin et al. (1998) presented the bi-directional evolutionary structural optimisation (BESO) method, where elements both can be added and removed. BESO also offers the choice to start the iterations from a minimum ground structure to which material mainly will be added, or to start with a maximum design domain from which material mainly will be removed. According to Querin et al. (1998), both starting points should end in the same result.

Genetic ESO (GESO) is a method presented by Liu et al. (2008), in which the concepts of survival-of-the-fittest and probability is introduced to the BESO-method. In GESO every element is given a binary string of arbitrary length, something that is not present in the BESO-method. At start, all digits in the binary string are 1 for all elements which means that all elements are filled with material. After each iteration

[2] A minimal ground structure is the smallest possible structure connecting the loads with the boundary conditions.

and sensitivity analysis, the genetic algorithms, based on the concepts mentioned above, are used to update the binary string. Depending on whether the elements is likely or not likely to be needed in the structure further into the iteration process, one bit in the string will be changed to 1 or 0 respectively. An element with one 0 is preliminary removed and the element is permanently removed when the binary string contains only 0. The probability that the element would have been useful later on in the iteration process is then very low.

The ESO method has been subject of criticism. This has mostly regarded the inefficiency of the method. Comparisons done by Rozvany (2009) shows that the method requires considerably more iterations to converge compared to other methods. The same author pointed out that the method in some cases also may not result in a globally optimal structure, but instead stops at a local optimum. Although the BESO-method is more likely to present a result near the global optimum, it is pointed out that even for BESO, the optimality of the solution cannot be guaranteed, Liu et al. (2008).

2.4.4 Level set

The methods for shape optimisation can also be used in topology optimisation if the boundary lines/surfaces are allowed to form new topologies. There is however the problem of re-parameterisation of the boundary when lines/surfaces merge, split, appears or disappears. Instead of defining a two dimensional geometry explicitly by its boundary lines a contour line of a surface in three dimensions, or equivalently an isoline of a two dimensional scalar field, can be used. As for a tree dimensional structure its boundary surface can be defined by an isosurface of a three dimensional scalar field. This way of defining the structure implicitly through a higher dimension function is used in the level set method.

3 Applications on simple examples

In this chapter examples are used to illustrate certain aspects of structural form optimisation and/or point out problems that are likely to occur. Effects of different optimisation parameters and optimisation types are investigated to show how the results are dependent on choices that have to be made. Table 1 contains all the examples shortly described.

Table 1 Optimisation examples and their purposes.

Section	Description
3.2	Size optimisation of a truss.
3.3	Shape and topology optimisation of a cantilever under stress constraint.
3.4	Topology optimisation of a cantilever under volume constraint.
3.5	Reliability of results from topology optimisations with minimisation of compliance as objective.
3.6	Topology optimisation of a short cantilever, comparing tools.
3.7	Penalisation and filtering variations in topology optimisation of a bridge.
3.8	Topology optimisation of a bridge including self-weight.
3.9	Topology optimisation of an S-shaped bridge in 3D.
3.10	Topology optimisation of a torsional support in 2D and 3D

A few notations that are used frequently deserve a proper definition. The compliance, Equation 5, or equally the strain energy, is often used as objective function. It is closely related to the inverse of the stiffness.

$$C = \boldsymbol{u^T F} = \boldsymbol{u^T K u} \tag{5}$$

In Equation 5 $\boldsymbol{u}$ is the nodal displacement vector, $\boldsymbol{F}$ is the nodal force vector and $\boldsymbol{K}$ the structural stiffness matrix. When results obtained from using a varying degree of penalisation are compared the compliance is at the end calculated without any penalisation, noted as $C_{\not{p}}$. This unpenalised compliance is calculated for the resulting structure but without penalisation for intermediate densities, i.e. a new stiffness matrix is constructed with $p = 1$ and a new displacement vector are calculated, both to be used in Equation 5.

A volume constraint $\bar{V}$ is set as a fraction of the total design domain volume. If the actual volume fraction of the solution differs from $\bar{V}$, it is denoted V. When using SIMP, the solution is very seldom all black and white. The amount of intermediate densities is expressed as an average density $\tilde{x}$ of all elements containing at least some material but not being prescribed to be material or void.

3.1 Tools

For the problems optimised in this report we have used both the commercial software HyperWorks and the Matlab program Toptimiser which is a modification of top88 by Andreassen et al. (2010). The main modifications are for efficiency, three-dimensional applications and ability to include self-weight. Introductions to HyperWorks and Toptimiser are given below.

3.1.1 HyperWorks

HyperWorks is a commercial program suite for computer-aided engineering (CAE), developed by Altair Engineering. The HyperWorks suite houses several programs, which are connected to one graphical user interface common for them all. The main programs in the suite are HyperMesh, a pre-processor used for drawing the model, assigning properties and making the mesh, and HyperView, a post-processor used for viewing and extracting results. The HyperWorks suite also contains other programs such as HyperGraph, which is used to make graphs from results, and HyperCrash, a program to simulate crashes of different kinds, e.g. car crashes.

HyperWorks can make use of various solvers to perform the analyses. The standard solver for finite element analyses in HyperWorks is RADIOSS, which is developed by Altair Engineering. Even though this is the standard solver for the program, the models created in HyperMesh can also be used with several third party solvers such as Abaqus and Ansys. One of the major features of HyperWorks is its ability to perform structural optimisations. This is done by the solver Optistruct, also developed by Altair Engineering. Optistruct can however only perform the optimisation algorithms and not the finite element analysis; it is thus required to use it together with RADIOSS to perform the optimisation tasks.

Optistruct has the ability to perform various kinds of structural optimisations. It can handle all the main optimisation types presented in this report, i.e. size, shape and topology optimisation. For the topology optimisation, the program is based on the SIMP method, Rozvany (2009), but how the SIMP method is applied in more detail is proprietary and is thus not publicly known. The SIMP method is described in Section 2.4.2. In Optistruct checkerboard control and minimum member size control can be applied, and it follows the method presented by Zhou et al. (2001).

In the structural optimisations done for this report, the programs HyperMesh and HyperView have been used together with the solvers RADIOSS and Optistruct. By using this constellation, the optimisations can be done with numerous different objectives and constraints. It is possible to run the optimisations with the objective to minimise compliance, volume or deflection among others. The same variables can also be used as constraints. It is also possible to add a stress constraint, and thus avoid optimised structures with extremely high stressed areas. Despite the wide choice of parameters to change, no appropriate method is available to model and optimise concrete structures. The option of setting different stress constraints in tension and compression does for example not exist, but is something what would greatly add to the applicability of the optimisation methods.

Convergence is achieved when the relative change in the objective function between two iterations fall below a given tolerance for 3 iterations in a row.

3.1.2 Toptimiser

The origin for this Matlab program was originally presented as topopt by Sigmund (2001). Topopt as well as Toptimiser uses the SIMP method together with an optimality criteria (OC) algorithm to update the densities. It does only optimise for minimum compliance under volume constraint, set as a volume fraction of the total design domain volume. The design domain is rectangular and discretized with a structured mesh using quadratic elements of equal size. The structured mesh with element and nodal numbering from top left to bottom right enable implementation of new problems, however not in the simplest way. Equal elements imply the use of only one element stiffness matrix with is scaled according to the density of each element. In topopt only one filter is available, filtering the sensitivities to set a minimum member size and therefore ensure the existence of a solution. As the magnitude of the density of the base material has no impact on the result, the design variables are directly interpreted as the densities as $\rho(x) = x = [0,1]$.

The program was updated to top88 by Andreassen et al. (2010), mainly for efficiency but also to include another filter option. Efficiency was mainly achieved by vectorising for-loops where possible, i.e. replacing for-loops with much faster vector operations. Matrices that had to be created with a for-loop were instead preallocated by reserving memory for the full matrix and consequently avoid the resizing of the matrix in the loop. The rectangular mesh implies that other than rectangular shapes for the design domain are specified by prescribing areas as void or material. Elements in these areas are however still included in the FE-analysis, as are elements in areas that are set to void during the optimisation. The response of these elements are however of no interest. By putting fixed supports at all nodes belonging to a zero density element leaving a reasonably thick layer or zero density elements with free nodes around as “cushions”, the accompanying degrees of freedom of the fixed nodes are excluded from the FE-analysis. The CPU-time per iteration will decrease as the optimisation proceeds and the total increase of efficiency depend mainly on the volume fraction. This modification will also decrease the memory usage, which is a critical issue as it sets a hard limit on the size of the model. However the decrease in memory usage in the first iteration, which has to be passed, will only be relative the amount of *prescribed* void area. Further decrease in memory usage will be of no practical benefit if the entire optimisation is run on the same computer. To be able to run large models a computer cluster with high memory capacity has been used for the first iterations and the process has been continued on an ordinary workstation when the memory usage has been low enough. Due to the difficulties in parallelising the FE-analysis and Matlab’s limitation to the use of no more than eight CPU cores, no increase in time efficiency was achieved by using a computer cluster.

The original code has also been extended from 2D to 3D, which is straightforward with only small changes in the main algorithm. Most different is the way the result is visualised, as pointed out already in 2.4.2. Only the sensitivity filter is available in the 3D extension, still using Matlab’s convolution function as suggested by Andreassen et al. (2010), now however the N-dimensional version. To calculate the element stiffness matrix for the cubic 8-nodes solid elements the CALFEM functions `hooke` and `soli8e` is used. The 2D and 3D versions are written as different programs; see the code in Appendix A-1 and Appendix A-4. The modified Matlab programs are in the text both referred to as Toptimiser. From a script defining each optimisation problem they are called as functions, `Toptimiser2D` or `Toptimiser3D`.

In structural engineering, the self-weight of a structure often make up for a large part of the total load. In contrast to ordinary applied loads the self-weight is design dependent and changes every iteration during the optimisation. This becomes a problem for the optimality criteria updater leading to divergence and oscillation of the design variables in large parts of the structure, thoroughly investigated by Bruyneel and Duysinx (2005). They show this problem using another updating scheme but we have encountered similar problems using optimality criteria. To overcome this problem they propose the use of a method developed by Svanberg (1987) called the method of moving asymptotes (MMA). A Matlab subroutine of MMA was provided by Svanberg and this has been implemented in Toptimiser. The subroutine includes the solution of a non-sparse system of equations with size n_{el}^2 during an inner loop which makes the optimisation process very slow. Self-weight problems have therefore been limited to low resolution 2D problems. Another problem using self-weight, also covered by Bruyneel and Duysinx (2005), appears in low-density regions where the power law used in SIMP gives an infinite weight to stiffness ratio. The effect is the formation of undesired members trying to control the otherwise uncontrolled deformations. A solution to this problem is the use of a linear penalisation for low density elements as in Figure 6.

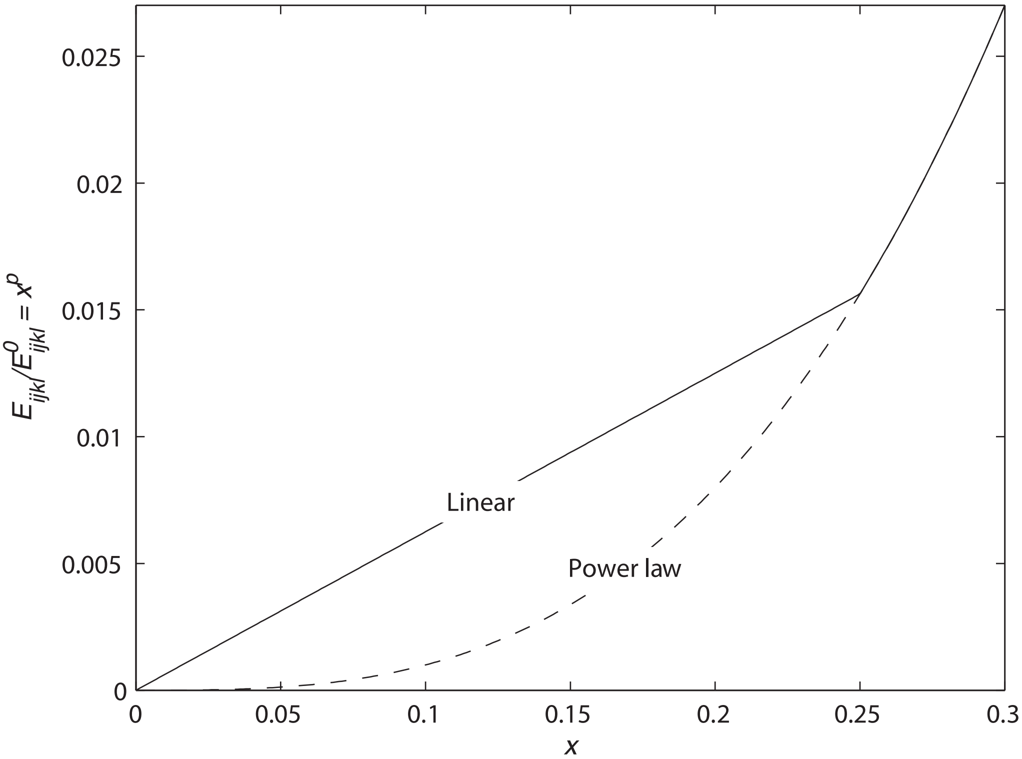


Figure 6 For gravity loaded structures linear penalisation is used for low densities (here x < 0.25).

Convergence is checked at the end of each iteration by comparing the element densities of the last and previous iterations. Convergence is said to be achieved when the maximum difference for all elements is smaller than a set tolerance.

3.2 Size optimisation of a truss

In this section, a truss is optimised using size optimisation. Since size optimisation only optimises sizes of discrete variables, and doesn't add or remove elements, the truss to be optimised must be defined prior to the optimisation. The truss optimised in this example is a truss with 21 bars, and can be seen in Figure 7. The continuous variables are chosen to be the cross-sectional areas of the bars. To decrease the number of design variables, the symmetry of the truss is used. This reduces the number of variables from 21 to 11.

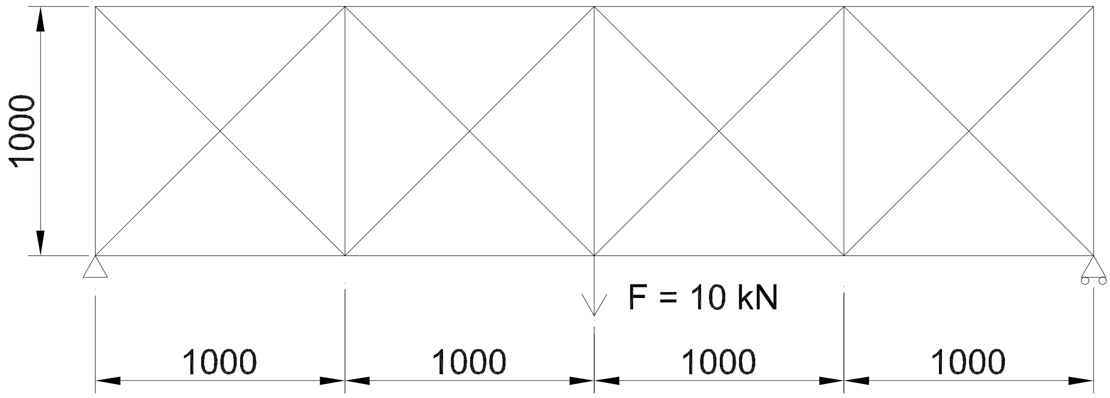


Figure 7 Drawing of the truss optimised by size optimisation. Dimensions are in mm.

The material parameters used for the model is those of steel, which gives a Young's modulus of 210 GPa and a Poisson's ratio of 0.3. Initially all the bars have the area 100 mm^2. During the optimisation the areas must be kept greater than 0 mm^2 but not more than 1000 mm^2.

The optimisation is done in two steps. In the first step, the truss is optimised with the objective to minimise the total volume of the structure. Deflection constraints are applied in the same point as the load. The deflection limit is chosen to be the deflection from a static analysis of the un-optimised truss. By doing this, the deflection is kept constant through the optimisation process which makes it possible to see how the same performance can be achieved with less material. After the first optimisation is complete, the bars with very low area are removed from the model. Keeping those bars would give a truss with unrealistically small members, as well as a truss that is unnecessarily complicated.

Step two is to redo the optimisation on the new truss, from which some elements have been removed. The parameters are basically the same, with the exception of an additional constraint on the stresses. The reason for not having the stress constraint in the first step is that a stress constraint in that case would prevent the areas of the bars from getting small. Even though the forces in the bars would be low, a very small area would still give rise to big stresses. The solution would because of that not have a distinct difference between the bars with big areas and those with small areas. A summary of the optimisation parameters used for these optimisations is shown in Table 2.

Table 2 Optimisation parameters used for size optimisation of truss.

	Step 1	Step 2
Optimisation objective	Minimise volume	Minimise volume
Optimisation constraints	Deflection limit (2.01 mm)	Deflection limit (2.01 mm) Stress limit (200 MPa)
Design variable limits	0 mm^2 < A ≤ 1000 mm^2	0 mm^2 < A ≤ 1000 mm^2

3.2.1 Results

The results from the size optimisations performed on the truss can be seen in Table 3. As seen in the table, the volume decreases with about 38 % after the two steps, compared with the values of the un-optimised truss. It can also be seen that the stress limit applied in the second step is not reached. It should be noted that the deflection for all cases is 2.01 mm, which is the same as the applied limit.

Table 3 Summary of results from size optimisation of truss.

	Initial values	After 1st step	After 2nd step
Total volume [mm^3]	$2.43 \cdot 10^6$	$1.56 \cdot 10^6$	$1.51 \cdot 10^6$
Max. stress [MPa]	70	53	100

The truss after the first step of the optimisation can be seen in Figure 8. The line weights are scaled according to the areas of the individual bars. It can clearly be seen that there is a distinct difference in area between the bars favoured by the optimisation and those not favoured. This makes it easy to choose the bars to be removed for the second step of the optimisation.

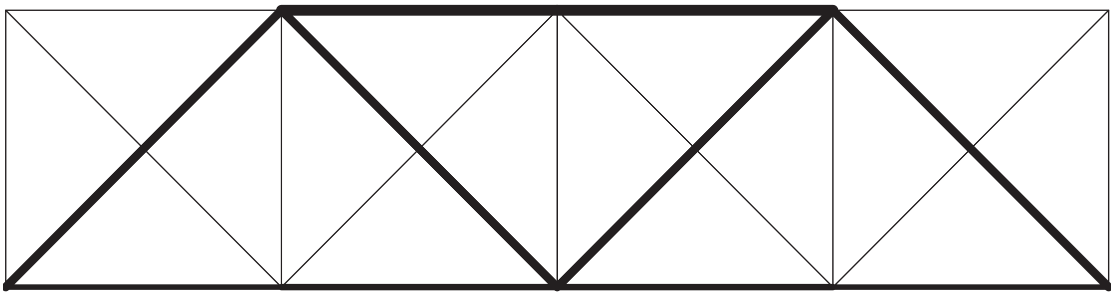


Figure 8 The truss after the first step of the optimisation.

The second optimisation, which is done with the bars with small areas removed, results in a truss that can be seen in Figure 9. Since the removed bars, even though they had small areas, helps to carry the load, the remaining bars have slightly bigger areas than they had before the second step. It can also be noted that the bars face a substantial increase in stresses, with almost 100 %, as can be seen in Table 2. The areas of the bars must be checked, so that no bar reaches the maximum allowed value of 1000 mm^2. For this truss the maximum area is 190 mm^2, which is well below the limit.

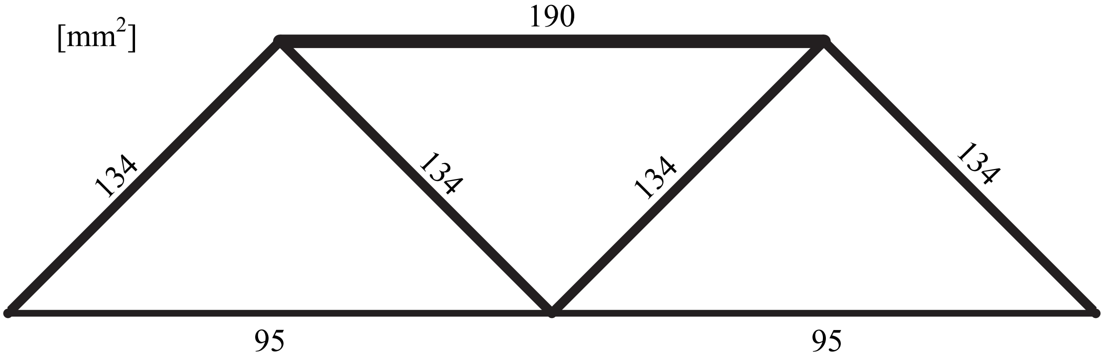


Figure 9 The final truss with the areas of the bars in mm^2, after both steps of the size optimisation.

3.2.2 Summary

The optimisation of a truss in this example shows how a more effective truss can be found, by the use of size optimisation. By using the two step procedure, as done in this example, a truss is achieved where both stresses and deflections are controlled to be within the allowable limits. It is seen that the volume of the structure, and thus the amount of steel needed to build it, can be greatly reduced.

The truss optimised in this example is quite simple, and a bigger truss with more loads introduces some problems. The major problem is that there might not be clear, which bars, and how many of them, to remove after step 1. This is due to that the areas are more widespread within the allowable range, so that there is no distinct difference between the most favoured bars and the less favoured ones. This means that even though the calculations and optimisations mainly are done by the computer, the method still required a skilled engineer to ensure a good quality result.

3.3 Shape and topology optimisation of a cantilever under stress constraint

In this example, a cantilever beam is optimised using both topology optimisation and shape optimisation, performed separately. The optimisations are done with a stress constraint. The tool used in this application is HyperWorks, because of its ability to handle stress constraints, something that can’t be done in Toptimiser. Topology and shape optimisation are two very different optimisation types. It is therefore of interest to see and compare what types of solutions the two methods are producing.

Before optimisation, the beam has a rectangular shape with a length of 3000 mm and a height of 300 mm, which can be seen in Figure 10. The beam is modelled with quadrilateral shell elements, which are given a thickness of 1 mm. The thickness is however not important regarding the final results. It will of course influence the magnitude of stresses and deflections, but not the shapes of the final structures. Since the structure is subject to bending, second order elements are used to get a better result, i.e. each element has an extra node per side, which gives each element 8 nodes in total. This increases the computational time, but not as much as if first order elements would be used in a mesh fine enough to get an equally accurate analysis. The mesh size used is 30 × 300 elements which gives a total of 9000 elements, a fine enough mesh for this optimisation.

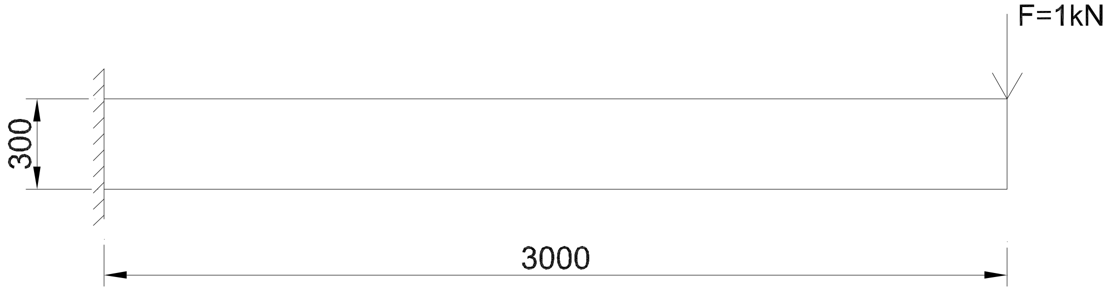


Figure 10 Design domain for cantilever beam, dimensions in mm.

The choice of material has no effect on the resulting shapes and topologies, as long as the material is homogenous and isotropic. To get reasonable stresses and deformations, the material parameters are however chosen to resemble those of steel. The Young's modulus is thus 210 GPa and the Poisson's ratio is 0.3.

The beam is loaded with a single 1 kN point load at the edge, as shown in Figure 10. This gives an initial maximum stress of 216.5 MPa at the corners of the fixed end of the beam, which is seen in Figure 11, showing the stress distribution in the beam before optimisation. All stresses calculated throughout this example are von Mises stresses.

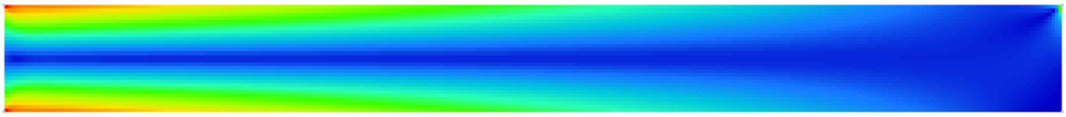


Figure 11 Stress distribution in beam before optimisation.

In both the shape and topology optimisations performed, the objective is to minimise the volume of the beam, which can be seen in Table 4. The only limit is the stress limitation, set to 220 MPa to be slightly higher than the highest stresses in the initial beam. No constraints are set on the deflection, compliance etc. This should give a highly stressed structure that utilizes the material in a quite effective way.

Table 4 Optimisation parameters for cantilever beam.

Parameter	Type
Optimisation objective	Minimise volume
Optimisation constraint	Stress limit (220 MPa)

3.3.1 Shape optimisation

The shape optimisation process is performed by first defining how the nodes are allowed to move. As mentioned in Section 2.3, it is preferable to have less design variables than nodes. In this application this is done by adding so called handles to the model. When the handles move, the mesh follows and the shape is changed, thus a parameterisation of the nodal movements is achieved which significantly decreases the number of design variables. The influence of one handle decreases linearly away from the handle and is zero at the neighbouring handles. Various combinations of handle points have been tested, but no significant differences in results have been found. The results in the report are based on the choice of three handles, evenly distributed along the bottom edge of the beam, which move in the vertical direction. The handle locations and possible shapes when moving one handle at the time are presented in Figure 12. In the handles are prescribed the maximum movement of the nodes, and to reach an optimal solution with regard to the shape, checks must be made to assure that the full movement is not reached during the optimisation. If that happens, it would suggest that the nodes should be moved even further to reach an optimal solution. The nodes can also move in the negative direction, although that is unlikely to occur in this example.

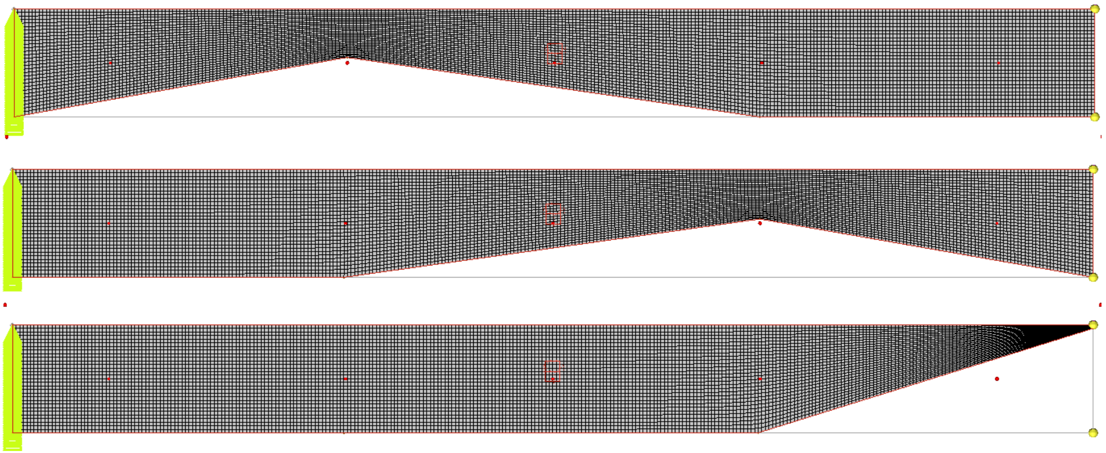


Figure 12 Shapes used in optimisation of cantilever beam

3.3.2 Topology optimisation

For the topology optimisation, parameters must be chosen to get a good black and white result. To achieve such a result in this example, checkerboarding is controlled and intermediate densities are penalised with a penalisation exponent of $p = 4$. Although checkerboarding is reduced by the use of second order elements, measures are taken to prevent it further, namely to introduce a limit on the minimum size of the structural members in the solution. The minimum dimension for this problem is set to 20 mm. That corresponds to 2 element lengths, which is the smallest possible dimension that can be applied. A summary of the parameters used are found in Table 5.

Table 5 Topology optimisation parameters for cantilever beam

Parameter	Value
Penalisation exponent	$p = 4$
Minimum member dimension	*MINDIM* = 20 mm / 2 elements

Several values of the penalisation exponent and minimum member dimension have been tested. The values used in the report, are the values that give the best performance, but still give a black and white solution without checkerboarding.

3.3.3 Results

In this example the focus is on differences between shape and topology optimisation, under a stress constraint. The results to be compared are the deflection directly under the load, the average stress and the volume. A summary of the results are presented below, in Table 6. Average stress is used as a measurement of the overall stress level of the structure. Since the optimisations are done with a stress constraint only, the stresses are expected to be high in large parts of the final structure, which then would give a high average stress.

Table 6 Results from optimisation of cantilever beam

	Initial values	Shape optimisation	Change	Topology optimisation	Change
Volume [mm^3]	$9.5 \cdot 10^5$	$5.8 \cdot 10^5$	**-36 %**	$4.40 \cdot 10^5$	**-51 %**
Avg. stress [MPa]	51.48	106.9	**108 %**	83.35	**62 %**
Deflection [mm]	-19.27	-43.33	**125 %**	-23.76	**19 %**

By looking at the result, it can be seen that even though the topology optimisation removes 51 % of the volume, compared to 36 % for the shape optimisation, it still out-performs the shape optimisation when it comes to deflection and average stresses. Especially remarkable is the deflection results, where the deflection for the shape optimisation increases with 125 % while the optimised structure from the topology optimisation only has 19 % larger deflection than the initial structure.

One thing to consider is not only the numerical results, but also the complexity of the resulting topology and shape. These can be seen in Figure 13 and Figure 14 below, where also the stress distribution is showed. The figures show clearly how the shape optimisation creates a simple structure without holes. The structure consists of a single structural member. The topology optimisation on the other hand results in a very complex structure, consisting of many members with different sizes.

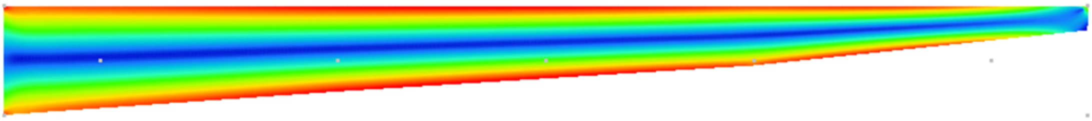


Figure 13 Resulting structure and stress distribution after shape optimisation.

The results from the shape optimisation shown in Figure 13, shows how the cantilever beam get narrower when moving out from the fixed end. This way the shape of the beam is similar to the shape of the moment distribution. It is seen clearly that further changes in shape is prevented by high stresses in the outmost fibres on the both sides of the beam. The result is a structure with evenly stressed edges along almost the entire beam. This is reflected in the high increase of average von Mises stresses in the structure, as seen in Table 6.

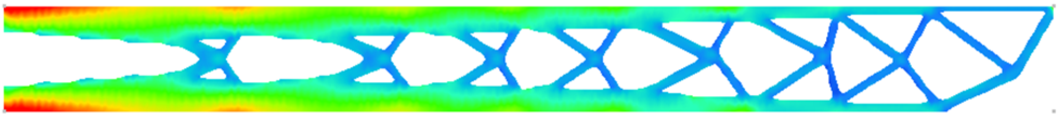


Figure 14 Resulting structure and stress distribution after topology optimisation.

The result from the topology optimisation is, as mentioned before, more complex than the result from the shape optimisation, but highly optimised. The structure consists of only 49 % of the initial material, but still manages to be stiff enough to only get a 12 % increase of the deflection. This indicates a very effective use of the material. This is also reflected in the, compared with the shape results, only moderate increase in average stresses.

The topology optimisation results could probably be optimised further. The topology optimisation method has the potential of developing fully stressed topologies, i.e. topologies where the stress of each member is equal to, or very close to the ultimate strength of the material. That this is not the case here partly depends on the mesh size and the minimum member dimension. Even though the dimension of less stressed members could have been reduced further, this was prevented by the minimum dimension control. If a finer mesh would have been used, smaller minimum dimensions could have been applied and thus also thinner members would be allowed.

3.3.4 Summary

The optimisations of the cantilever beam using both shape and topology optimisation methods with stress constraint show strengths and weaknesses of both the methods. The shape optimisation with stress constraint produces simple structures. If a single structural member is optimised, it will result in a single member as well, although with an optimised shape. This comes at a cost however, and that is average efficiency in material use and performance when it comes to deflections.

The topology optimisation on the other hand uses the material extremely efficiently and creates a stiff structure, which limits the increase in deflections despite a large decrease of material volume. This structure is however complex, with a lot of small members, which normally is a major disadvantage when it comes to manufacturing costs.

By using a stress constraint, the goal is to achieve a high utilization ratio of the material. Both methods show the potential of doing that, although with different results. Which method to use should be decided by the type of structure to be optimised, and the demands on that structure.

3.4 Topology optimisation of a cantilever under volume constraint

In this example, the same beam as in Section 3.3 is optimised, now with the optimisation objective to minimise the global compliance and enforcing a volume constraint. The volume constraint is chosen to be the volume fraction of the optimised structure in Section 3.3.2. By doing this optimisation, a comparison can be made between two topologies with the same amount of material but created under different constraints. This way it can be seen, if topology optimisation produces similar topologies regardless of which constraints are applied.

The parameters used during the optimisation can be seen in Table 7, and is chosen to be the same as in Section 3.3. Also the material parameters are chosen to be the same as in the previous section.

Table 7 Optimisation parameters for the cantilever beam.

Parameter	Value
Optimisation objective	Minimise compliance
Optimisation constraint	Volume fraction < 0.489
Penalisation	$p = 4$
Minimum member dimension	20 mm (2 elements)

3.4.1 Results

To see if different constraints during topology optimisation give different topologies, given that the volumes of both the results are the same, certain numerical results from the two results are compared. Furthermore the topologies are visually compared. The part of the results regarding optimisation with stress constraint is taken from Section 3.3.2. A summary of the results can be seen in Table 8.

Table 8 Summary of results for topology optimisation of cantilever with stress or volume constraint.

	Topology with stress constraint	Topology with vol. constraint	Difference
Volume	$4.40 \cdot 10^5$	$4.40 \cdot 10^5$	-
Avg. Stress	83.35	80.26	**-3.7 %**
Deflection	-23.76	-24.55	**3.3 %**
Compliance	$1.18 \cdot 10^4$	$1.22 \cdot 10^4$	**3.4 %**

The numerical results show no big differences between the two cases. The biggest difference can be seen in the average stress, where the stresses at the optimisation with volume constraint are 3.7 % smaller compared to those at the optimisation with stress constraint. This is however nothing strange, an optimisation with only a stress constraint should lead to high stressed structures.

One thing that is more noteworthy is that the compliance for the topology optimisation with volume constraint, where the objective was to minimise the compliance, is higher than for the optimisation with stress constraint, where the objective was to minimise volume.

The optimised topology from the optimisation with volume constraint can be seen in Figure 15. It is similar to the one from the optimisation with stress constraint, which can be seen in Figure 14, but with small differences. The most noticeable is the somewhat different topology in the area directly beneath the load. The complexity of the topology is also slightly higher for the topology created under volume constraint with 32 members between the outer flanges, compared to 29 for the topology with stress constraint.

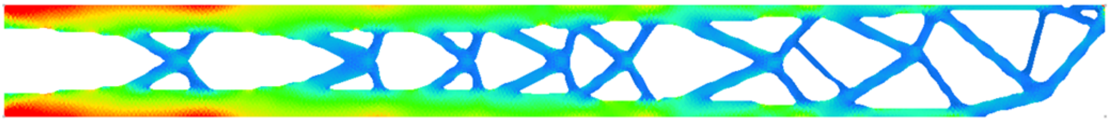


Figure 15 Optimised topology of cantilever beam with volume constraint. Colours represent stresses

3.4.2 Summary

The optimisation of the cantilever beam with volume constraint shows that similar topologies are produced by topology optimisation even though different objectives and constraints are applied. Only small differences are found in the numerical results. By visual inspection, the resulting topologies are also found to be similar. One strange anomaly is however found in the numerical results. The compliance is higher for the topology which was optimised to minimise the compliance compared to the topology optimised for minimum volume. This suggests that the stress constraint leads the optimisation along a better path to convergence than the volume constraint, which seems to end up in a local optimum, although one fairly close to the presumed global one.

3.5 Reliability of results from topology optimisations with minimisation of compliance as objective

A topology optimisation with the objective to minimise the compliance should lead to the topology with the smallest compliance for a given volume. The optimisation performed in Section 3.4 shows on a possibility that this is not always the case. It is therefore of interest to investigate if the issue with suboptimal results in Section 3.4 is an isolated event, or if it also can be produced for other structures as well. All optimisations in this section are done with HyperWorks, since it is in HyperWorks the issue has been detected.

In order to investigate this issue further, a number of structures are optimised. Two different optimisations are performed on the structures. One (optimisation 1) in which the objective of the optimisation is to minimise the volume of the resulting topology with a constraint on the allowable stresses. The volume of the resulting structure of that optimisation is then used as a volume constraint in the second optimisation (optimisation 2), where the objective is to minimise compliance. This gives two structures with identical volumes. A summary of the optimisation parameters used in all the optimisations can be found in Table 9.

Table 9 Optimisation parameters used for this section.

Parameter	Optimisation 1	Optimisation 2
Optimisation objective	Minimise volume	Minimise compliance
Optimisation constraint	Stress limit	Volume fraction constraint
Min. member dimension	4	
Penalisation	3	

The material parameters do not influence the comparability of the results, as long as the same parameters are used in both optimisations. To get familiar results, they are however chosen to be those of steel for all optimisations done in this section, i.e. a Young's modulus of 210 GPa and a Poisson's ratio of 0.3 are used.

The stress limits are however not based on the strength of steel. Instead they are chosen to be close to the maximum stresses in the design domain prior to optimisation, which are found through a simple static FE-analysis on the design domain. This is done in order to get reasonable topologies. If a too small stress limit is applied, the topologies become large and bulky. If the limit is set too high, the structural members instead can become very thin, even though a minimum member size is controlled.

3.5.1 Short cantilever

The short cantilever optimised has the dimensions as shown in Figure 16. The element size used is 1 mm, which gives a mesh size of 200 × 200 elements. The stress limit used is 50 MPa and the magnitude of the force applied is 100 N.

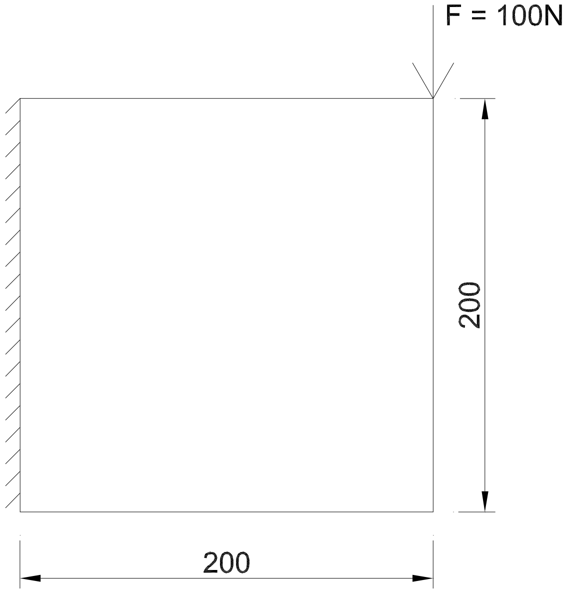


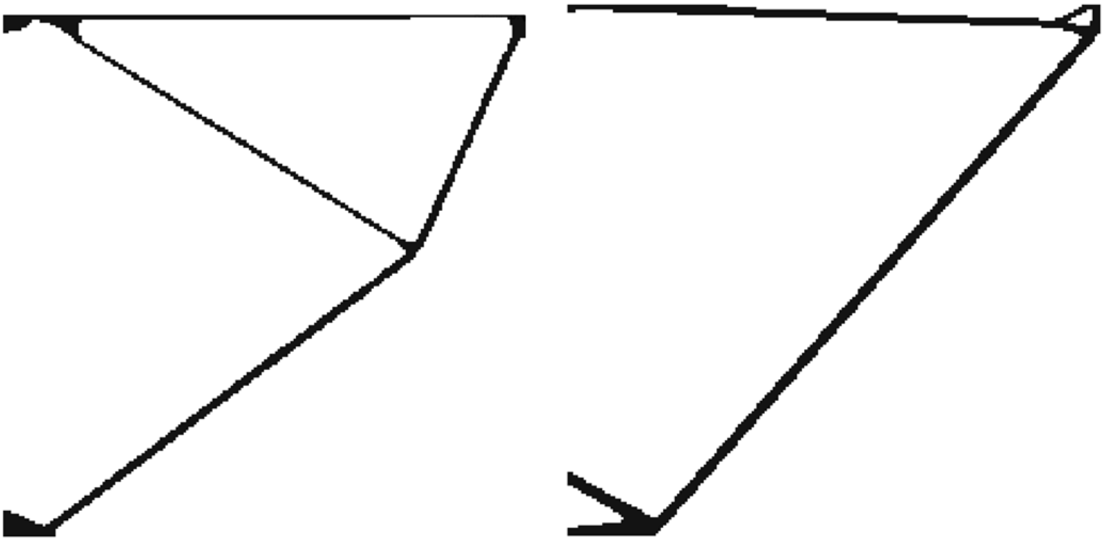
Figure 16 Design domain and load before optimisation, dimensions in mm.

As seen in Table 10 the two optimisations perform as they should. The optimisation with the objective to minimise the compliance does produce a topology with a lower compliance. The difference between the results are however small.

Table 10 Results from optimisation of short cantilever.

	Optimisation 1	Optimisation 2
Compliance	5.67	5.45
Volume fraction	0.0541	0.0541

The topologies created by the both optimisations are rather similar. Although the result from optimisation 1, seen in Figure 17a, contains one additional large structural member, optimisation 2 has a bit more complex structure at the lower boundary and under the load. Since no big differences are found neither in the numerical results nor in the topologies created, the two optimisations create equally good solutions for this problem.



(a) *Objective to minimise volume with stress constraint.*

(b) *Objective to minimise compliance with volume constraint.*

Figure 17 Resulting topologies from the optimisation 1 (a) and optimisation 2 (b).

3.5.2 Bridge-like structure

To catch the performance of a somewhat more complex case, the optimisations are performed on a design domain which resembles a bridge-like structure. The initial design domain can be seen in Figure 18, and as in the previous section the element size used is 1 mm. This gives a mesh size of 400 × 100 elements. The stress limit used in the example is 50 MPa and the load applied is a distributed load at mid-height with the magnitude 1 kN/m.

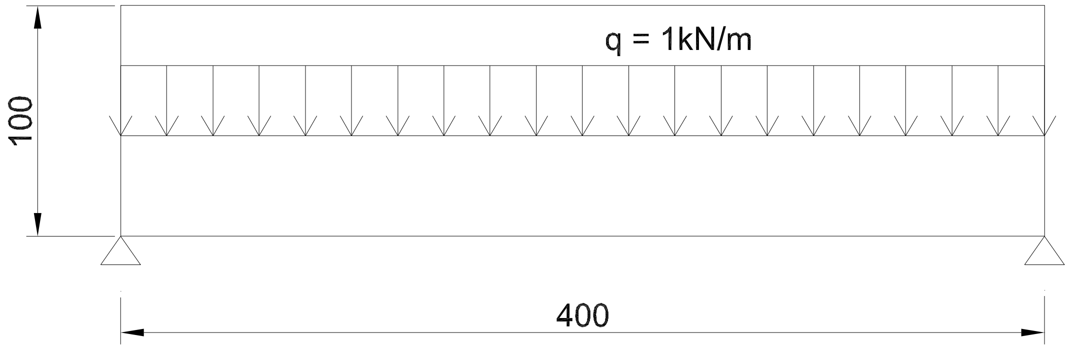


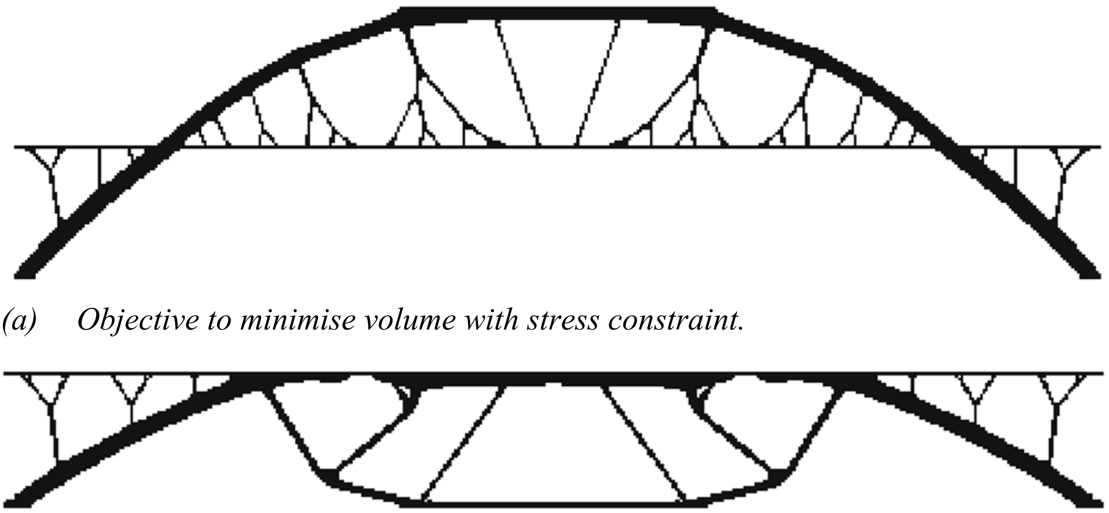
Figure 18 Design domain and distributed load for bridge-like structure before optimisation. Dimensions are in mm.

The results of the two optimisations are found in Table 11. It can be seen that surprisingly the topology from the optimisation where the volume was minimised with a stress constraint, out-performs the other topology by far with compliance less than a third. In other words, the compliance for the topology that was created to minimise the compliance is actually three times larger than for the topology that didn’t regard the compliance at all.

Table 11 Results from optimisation of bridge-like structure.

	Optimisation 1	Optimisation 2
Compliance	11.30	38.11
Volume fraction	0.0970	0.0970

The topologies created in the two optimisations differ from each other, seen in Figure 19. The topology created in optimisation 1, where the volume was minimised and stress was limited, consists mainly of a large arch with ties connecting the arch to the deck. The result from optimisation 2 has instead a truss beneath the deck. It also has fewer structural members than the topology from optimisation 1.



(a) Objective to minimise volume with stress constraint.

(b) Objective to minimise compliance with volume constraint.

Figure 19 Resulting topologies from optimisation 1 (a) and optimisation 2 (b).

It is clear that the arch shaped topology created from a minimisation of volume under a stress constraint performs substantially better when it comes to compliance compared to the topology created when compliance was minimised with a volume constraint.

3.5.3 Truss

The design domain to be optimised in this section, which is likely to produce a truss structure, can be seen in Figure 20. The element size is 1 mm, which gives a mesh size of 200 × 100 elements. The applied load has a magnitude of 100 N.

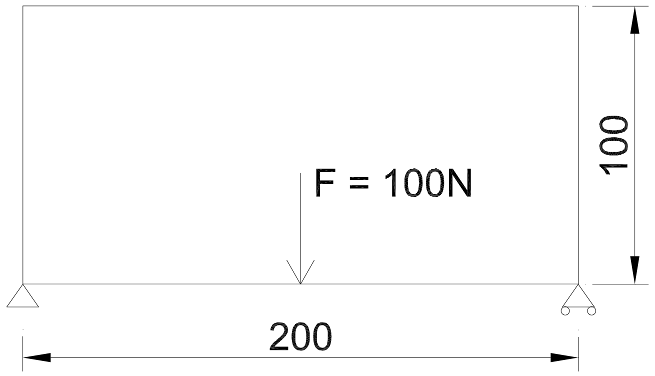


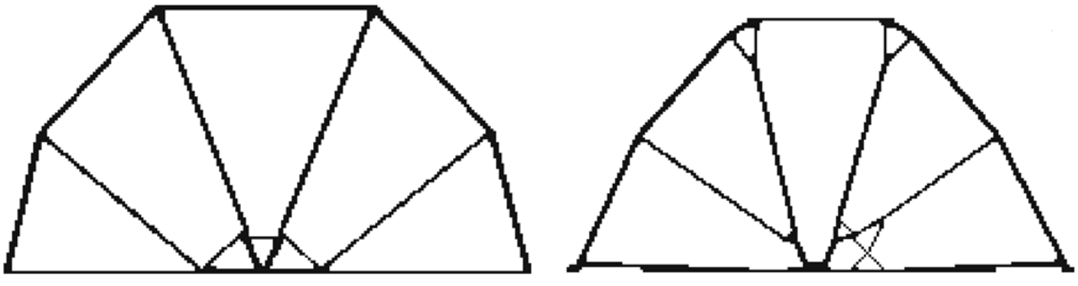
Figure 20 Design domain and load before optimisation. Dimensions are in mm.

The compliances from the resulting topologies of the two optimisations can be seen in Table 12. The compliance from optimisation 2 is about 18 % higher than the one from optimisation 1. That means that the compliance is again higher for the optimisation that was supposed to minimise the compliance.

Table 12 Results from optimisation of truss.

	Optimisation 1	Optimisation 2
Compliance	1.75	2.07
Volume fraction	9.33 %	9.33 %

The topologies created during the optimisation can be seen in Figure 21. The two topologies are rather similar, although the topology from optimisation 2 is not truly symmetrical. It is possible that this is caused by the unsymmetrical boundary conditions. Due to the topology being unsymmetrical, it also becomes somewhat more complex with more members.



(a) Objective to minimise volume with stress constraint.

(b) Objective to minimise compliance with volume constraint.

Figure 21 Resulting topologies from optimisation 1 (a) and optimisation 2 (b).

3.5.4 Summary

The compliances of the topologies created in this section are shown in Figure 22. In the chart are also the results from Section 3.3.2 and 3.4, where the analysis was performed in the same manner as in this section.

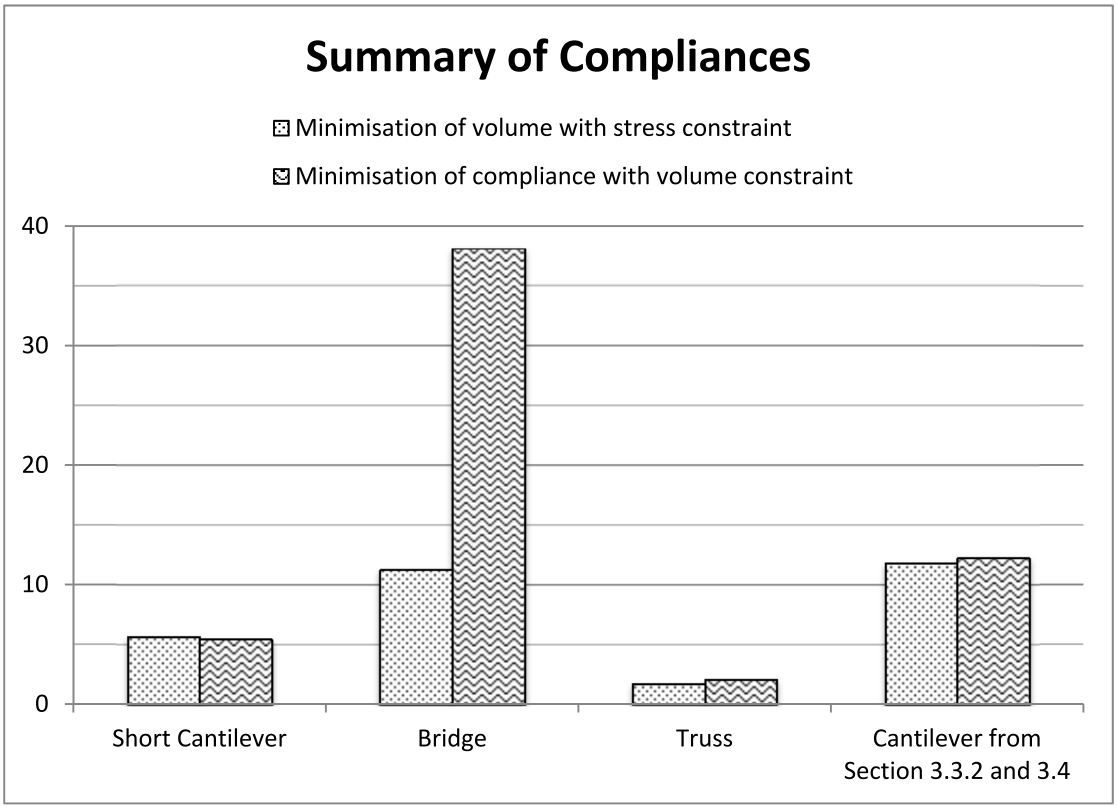


Figure 22 Summary of compliances from the topologies created in this section.

For the optimisations where the compliance is minimised and the volume constrained, a truly optimal solution means that no other topology with the same amount of material would have a lower compliance. The analyses in this section show that in 4 of 5 cases, the optimisation with the objective to minimise compliance fails to find a truly optimal solution. Instead a more optimal solution, with regard to compliance, is in those cases found when minimising the volume and limiting the stresses. This raises questions on the reliability of the results when performing optimisations, especially optimisations where compliance is minimised with the volume constrained.

3.6 Topology optimisation of a short cantilever, comparing tools

The purpose of the topology optimisations of the short cantilever performed in this example is to see if both programs used for this thesis produce similar topologies. This is done by comparing results from Toptimiser and HyperWorks. In order to make a comparison of the results relevant, the same problem is optimised in both programs and thus identical models are used in the two programs. The model described below is consequently valid for both Toptimiser and HyperWorks, i.e. the same geometry and the same parameters have been used as far as possible. What differs is mainly how the parameters are applied by the different programs, something that often may be impossible to change in the commercial software.

3.6.1 Model and parameters

The initial design domain to be optimised in this example can be seen in Figure 23. It is a simple quadratic shape, which enables the focus to be on differences between the resulting topologies rather than the topologies themselves. Boundary conditions are applied along the left boundary, where displacements are constrained in all directions. The structure is loaded with a single point load of 1 N, applied in the bottom-right corner. Appendix A-3 contains the script defining the cantilever problem for Toptimiser.

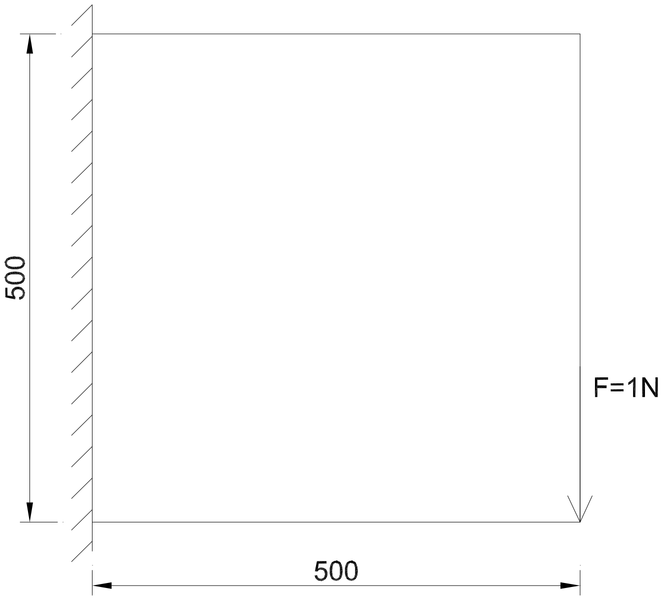


Figure 23 Model of cantilever with load and boundary conditions

The choice of material in the model does not affect the result, as long as the whole model is made from a single isotropic material. It is however important, since the results from HyperWorks and Toptimiser will be compared, that the same parameters are used in both programs. Otherwise stresses and deflections, for example, would not be comparable. Used in the model is an artificial material with a Young's modulus of 1 Pa and a Poisson's ratio of 0.3.

In order to get good results, a rather fine mesh is used. The mesh has the size 500×500 elements, which gives a total of 250 000 elements. The elements are square first order plane stress elements, i.e. four nodes per element. In Toptimiser this is the only available element type and each node has two degrees of freedom, the displacements in each two directions. Such elements are not explicitly featured in HyperWorks, where instead shell elements are used. The nodes of these elements have in addition to a third direction of displacement also two out of plane rotations, which gives a total of five degrees of freedom per node. With no out of plane loading these extra degrees of freedom will all be zero and the shell elements will behave as simple plane stress elements.

The objective of the optimisation for this problem is to minimise the global compliance of the model. This is done under a volume constraint, where the volume fraction is not allowed to differ from $\bar{V}$ = 0.15. By using this combination of objective and constraint, the optimisation should lead to the stiffest possible topology that uses 15 % of the initial material.

In HyperWorks convergence is checked against the relative change in objective function whereas Toptimiser uses the maximum absolute change in density for any element. To make the comparison more valid, Toptimiser is for this example only coded to use the same convergence criterion as HyperWorks. The tolerance for convergence is set to 0.5 %.

In order to get a good solution, with only black and white elements and without any checkerboard areas, intermediate densities are penalised and the member sizes are controlled. In HyperWorks the minimum member dimension is set to *MINDIM* = 10 elements and in Toptimiser the filter radius is set to r_{min} = 5 elements. The penalisation of intermediate densities is done with p = 5. How the penalisation is applied differs between the programs. For Toptimiser, the penalisation exponent is increased by 10 % each iteration when the maximum change of density for any element is lower than five times the global convergence criterion. In HyperWorks, it is not possible to control how the penalisation is applied. It is however obvious when looking at the iteration history, that the penalisation is not applied in full from the start, but gradually increases throughout the iteration process. This is also confirmed in the Optistruct User's guide, which states that the penalization increases stepwise during different stages of the iteration process. The parameter values are summarised in Table 13.

Table 13 Parameters used for optimisation of cantilever.

Parameter	Value
Optimisation objective	Minimise compliance
Optimisation constraint	$\bar{V}$ = 0.15
Penalisation	p = 5
Minimum dimension / filter radius	*MINDIM* = 10 elements / r_{min} = 5 elements

The compliance is not calculated the same way in both programs. In Toptimiser it is calculated according to Equation 5 while HyperWorks calculates it as $C = ½\, \boldsymbol{u}^T \boldsymbol{K} \boldsymbol{u}$. This means that in the results from HyperWorks, the compliance should be half of that from Toptimiser if an identical structure is analysed. To show this easier, a weighted result for HyperWorks has been created which simply is 2 times the output from the program. This applies however only on the compliance, and no other results such as deflection.

3.6.2 Results

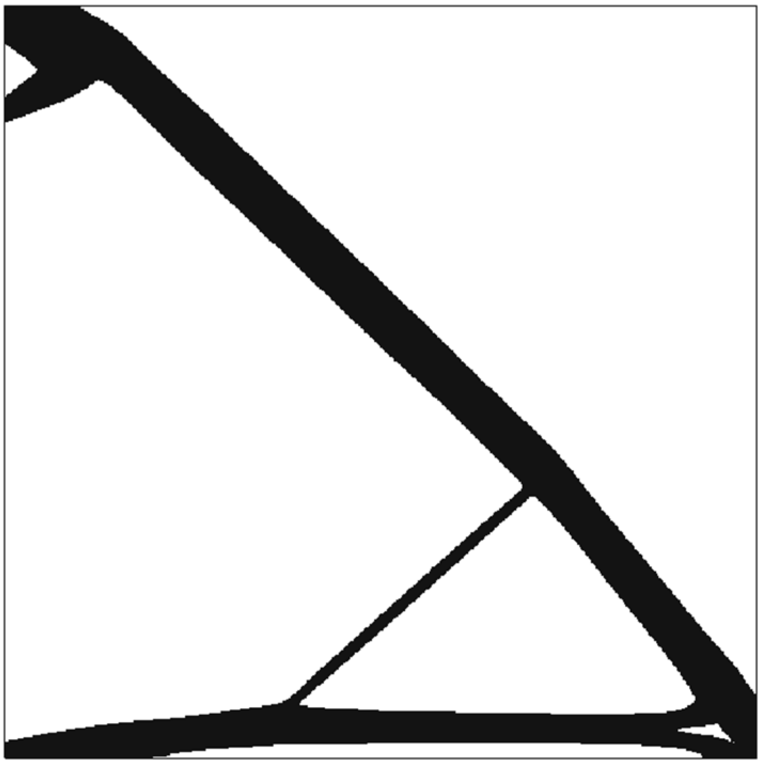
The results from the optimisation, from both HyperWorks and Toptimiser, are presented in Table 14 below. As explained in Section 3.1, both programs use the SIMP method to perform the topology optimisation. The implementation of the method is however not identical between the programs, and it is thus interesting to study if the two produces similar results.

Table 14 Summary of results from optimisation of cantilever. The compliance value for HyperWorks is the weighted value.

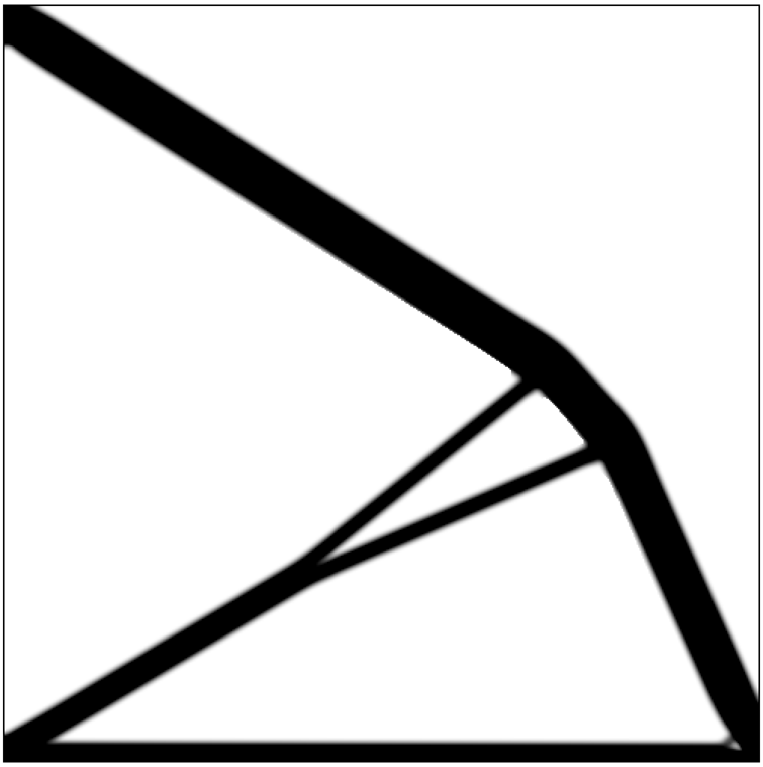
	HyperWorks	Toptimiser	Difference
Compliance	71.5	58.05	-19 %
Deflection	70.98	58.05	-18 %
Maximum Stress	2.3	1.85	-20 %
Number of iterations for convergence	55	1000 (no conv.)	-

The results show differences between the outputs from the two programs. The Matlab program Toptimiser produce values that are between 18 % and 20 % lower than those from HyperWorks, i.e. the topology from Toptimiser performs better and uses the material more effectively. Another thing to note is that even though the same maximum step size and the same convergence criterion have been used, the number of iterations required for the solution to converge differs greatly between the two programs. HyperWorks converges after 55 iterations while Toptimiser hasn't converged even after 1000 iterations. As will be seen later, Toptimiser has problem with convergence for penalisation exponents larger than $p = 3$. How the penalisation is applied can also have influence on these numbers. The time required for each iteration is however lower for Toptimiser, demonstrating the efficiency of the program that is, although simple, written in a high level programming language.

It is also of interest to compare the actual topologies produced by the two programs, to visually detect differences in the topology. The two optimised topologies are shown in Figure 24. As easily seen, the programs do not produce identical topologies, as could be assumed considering the numerical results. In general, Toptimiser produces a topology which is made of simple straight members with an even thickness, while the topology of HyperWorks features more curved members and also members with varying thickness. The varying thicknesses are mostly concentrated to the regions where the boundary conditions and the load are applied.



(a) Optimisation result from HyperWorks.



(b) Optimisation result from Toptimiser.

Figure 24 The optimised topologies from (a) HyperWorks and (b) Toptimiser. The borders are only to illustrate the initial design domain and are not part of the structures.

3.6.3 Summary

The topology optimisations of a cantilever performed in this example shows on differences between the two tools used for this thesis. It is shown that even though the same parameters are used, for the material as well as for the optimisation parameters, the programs do not produce identical topologies. This shows that the optimisation methods presented in Chapter 2 can be applied in many ways, and that it is not certain that two programs based on the same method, such as SIMP in this case, produces identical results.

When comparing the results from the two programs, it can be seen that Toptimiser in this case produces a solution that performs better in terms of compliance and deflection, and also a topology that is less complex and thus easier to realize. No further conclusions can however be drawn from this, since it is very possible that Toptimiser produces a less optimal solution for another problem.

3.7 Penalisation and filtering variations in topology optimisation of a bridge

The amount of penalty applied on intermediate densities and the chosen filtering technique with its parameters heavily affects the results of an optimisation. As discussed in Section 2.4 and 2.4.2 the discrete problem is in the SIMP-model relaxed by allowing continuous design variables. The areas of low-density material that now appear is removed by applying a penalisation to all intermediate densities which make them less efficient and therefore drives the optimisation towards a black and white solution. A filter can ensure a certain minimum member size, thus resolving the checkerboard problem, and may also be applied to remove grey material. In the following example different amount of penalisation and different filters with varying parameters are applied to the same problem visualising the differences of the results.

The problem can be interpreted as that of a bridge, loaded by a vertical distributed load on the top side of the design domain and with constrained displacement in both directions at the lower corners. The structure is optimised for minimum compliance with constraint on how much of the initial material that can be used. Only the resulting topology and stress distribution is of interest, not the magnitude of the deflections or stresses. Units on size of design domain and applied loads are therefore left out. The bridge has double the length as the height, but only one half is optimised due to the symmetry condition. The domain is discretized by a mesh of 125 × 125 4-node quadratic elements and the volume fraction, i.e. the amount of available material in relation to the design domain is set to $\overline{V} = 0.15$. The optimisation parameters are summarised in Table 15. The Matlab program Toptimiser is used, see Appendix A-2 for the script defining the bridge problem.

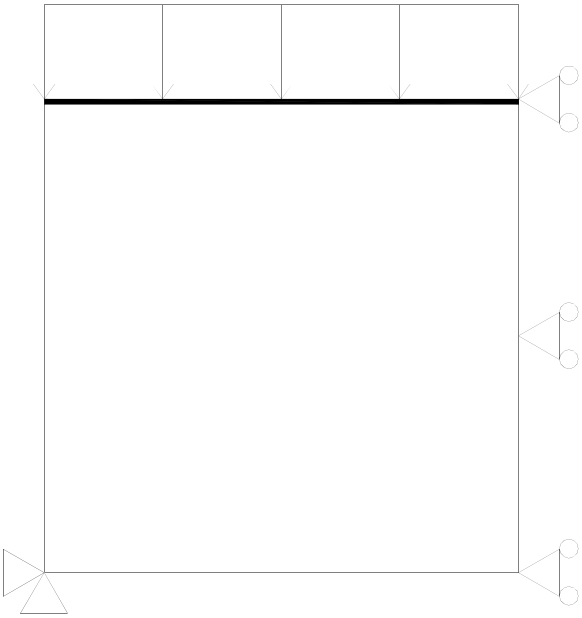


Figure 25 Design domain with boundary conditions and loading used for the bridge problem.

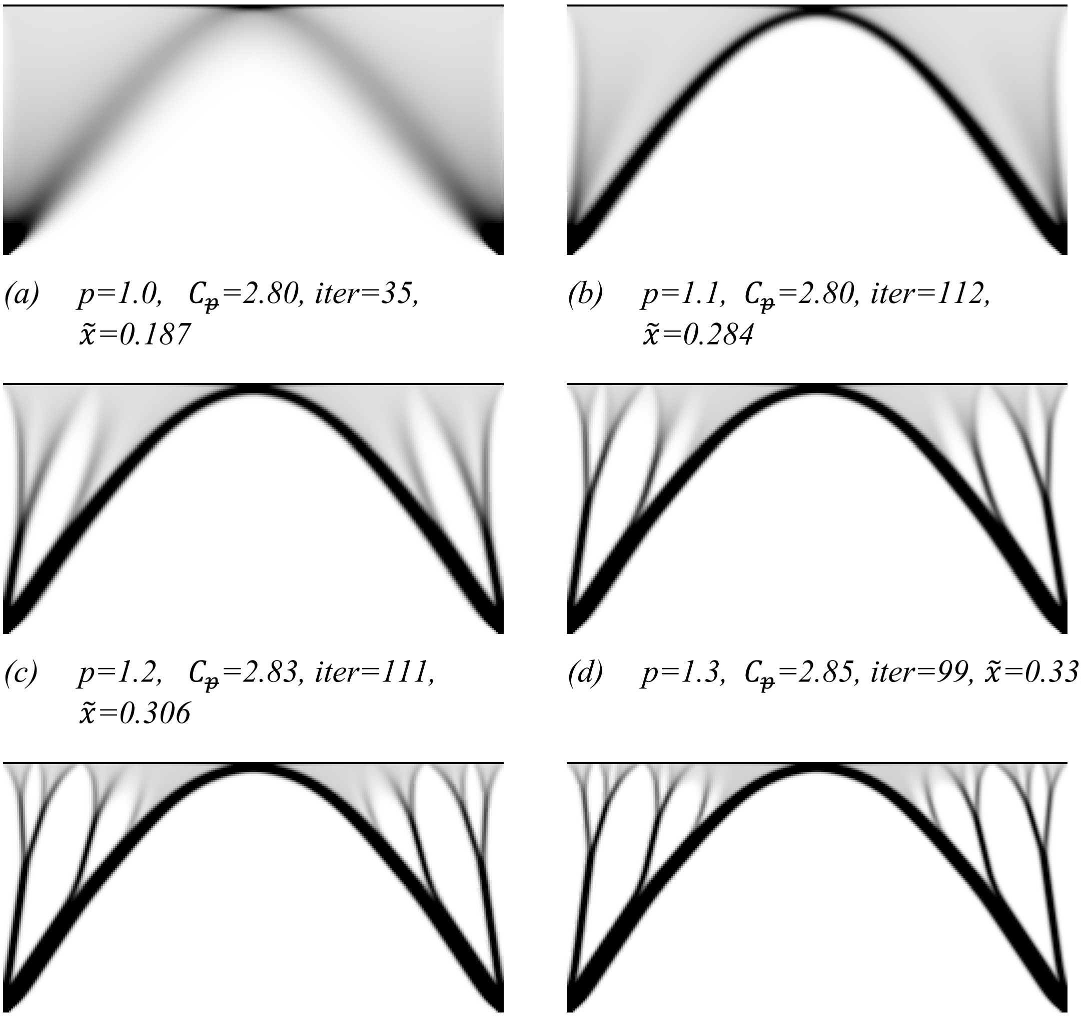
Table 15 Optimisation parameters common for all optimisations of the bridge.

Parameter	Value
Optimisation objective	Minimise compliance
Optimisation constraint	$\bar{V}$ = 0.15
Mesh size	125 × 125

3.7.1 Penalisation variation

Figure 26 shows the results from using a constant penalisation ranging from p = 1.0, meaning no penalisation, up to p = 10.0. A filter is used to ensure the minimum radius of any member not to be less than r_{min} = 1.5 element widths. The compliance, C, which is used as objective function is calculated for the resulting structure without penalisation, denoted $C_{\not p}$, to enable comparison between results. Up to a penalisation of p = 3.0 the optimisation was able to converge, with further increase however the solution started to oscillate, usually some thin members alternating being black or grey. It is observed that all solutions up to p = 5.0 contains an arch as the main load carrying member. With p = 6.0 and increasing, the arch is more and more interchanged for two struts. Up to p = 3.0 a continuous glomeration of low-density material into high-density, well defined members is observed, with no major change in the topology (except for none or very low penalisation where the area above the arch is mostly grey). The average density, $\tilde{x}$, is a measure on the amount of grey material and is calculated from all elements with a density larger than zero, excluding the prescribed black elements in the bridge deck. Already at $p = 2.5$ there is only a small amount of grey material left, and at $p = 10$ there is still a grey border around every member as a side effect of the filter used. Up to $p = 5.0$ the increase in compliance is moderate, from $p = 0$ to $p = 5$ the increase is just 13.6 %, see Figure 27.

The struts bringing the load down to the arch are tree-shaped due to the distributed load and the main arch gains thickness towards the supports and change direction where a strut is connected. It is very clear how the forces are brought to the support and the structure has a quite interesting organic expression.



(a) $p=1.0$, $C_{\not{p}}=2.80$, *iter*$=35$, $\tilde{x}=0.187$

(b) $p=1.1$, $C_{\not{p}}=2.80$, *iter*$=112$, $\tilde{x}=0.284$

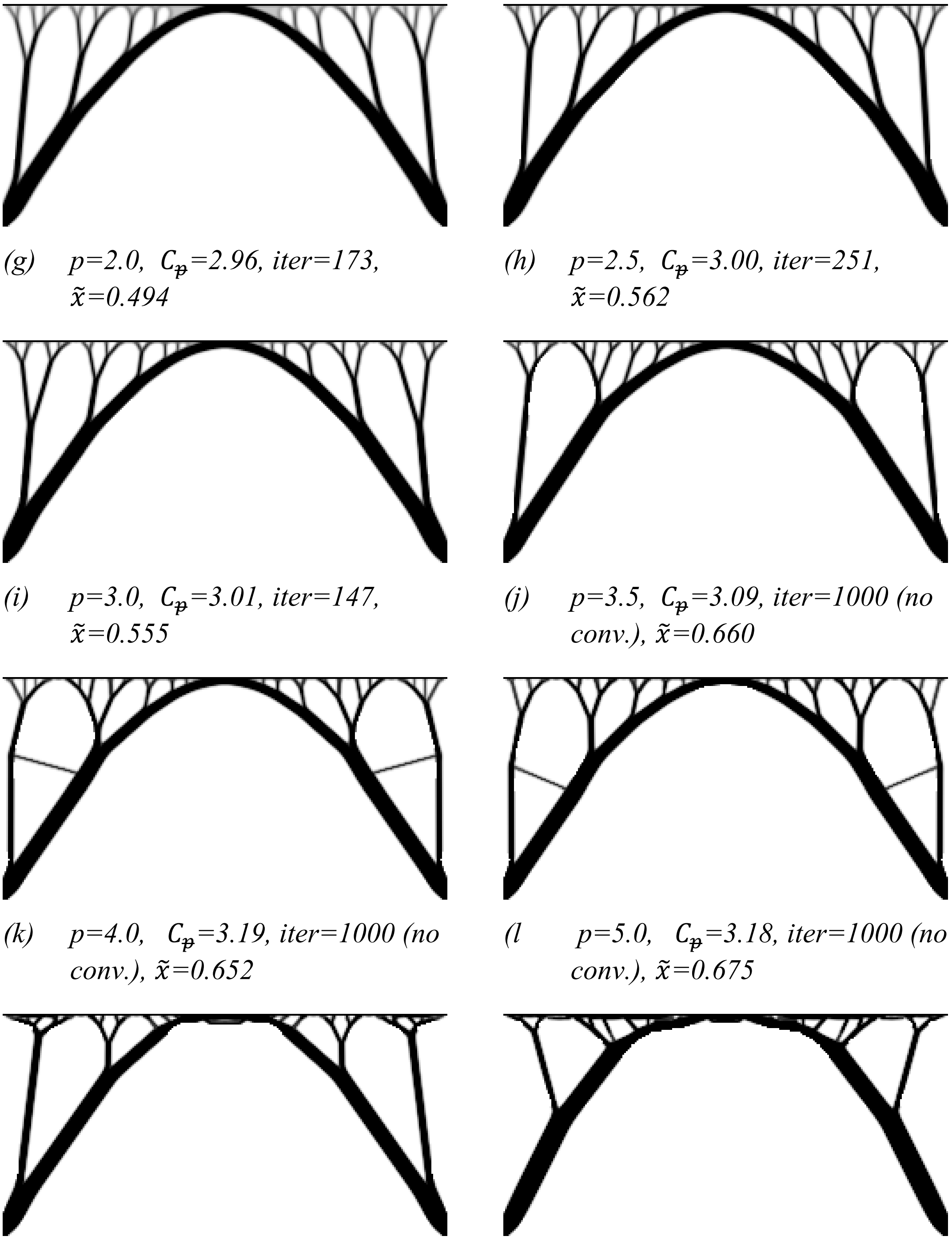
(c) $p=1.2$, $C_{\not{p}}=2.83$, *iter*$=111$, $\tilde{x}=0.306$

(d) $p=1.3$, $C_{\not{p}}=2.85$, *iter*$=99$, $\tilde{x}=0.339$

(e) $p=1.4$, $C_{\not{p}}=2.88$, *iter*$=109$, $\tilde{x}=0.377$

(f) $p=1.5$, $C_{\not{p}}= 2.89$, *iter*$=107$, $\tilde{x}=0.402$

Figure 26 *Optimised design for the bridge problem with increasing penalisation,* $p = [1,10]$*. The mesh is* 125×125 *with symmetry boundary conditions, a filtering radius of* $r_{min} = 1.5$ *is used.* $C_{\not{p}}$ *is the compliance calculated without penalisation for the resulting topology, iter is the number of required iterations for convergence and* $\tilde{x}$ *is the average of all non-zero design variable densities. Continued on next page.*



(g) p=2.0, $C_{\mathfrak{p}}$=2.96, iter=173, $\tilde{x}$=0.494

(h) p=2.5, $C_{\mathfrak{p}}$=3.00, iter=251, $\tilde{x}$=0.562

(i) p=3.0, $C_{\mathfrak{p}}$=3.01, iter=147, $\tilde{x}$=0.555

(j) p=3.5, $C_{\mathfrak{p}}$=3.09, iter=1000 (no conv.), $\tilde{x}$=0.660

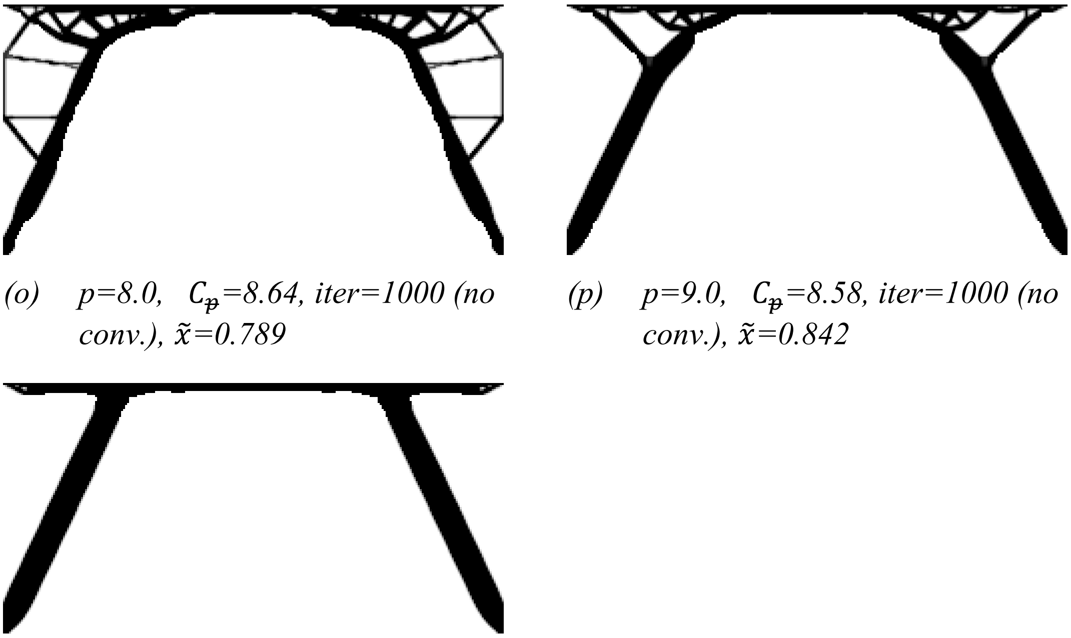
(k) p=4.0, $C_{\mathfrak{p}}$=3.19, iter=1000 (no conv.), $\tilde{x}$=0.652

(l p=5.0, $C_{\mathfrak{p}}$=3.18, iter=1000 (no conv.), $\tilde{x}$=0.675

(m) p=6.0, $C_{\mathfrak{p}}$=3.61, iter=1000 (no conv.), $\tilde{x}$=0.752

(n) p=7.0, $C_{\mathfrak{p}}$=4.43, iter=1000 (no conv.), $\tilde{x}$=0.800

Figure 26 Continued from last page. Optimised design for the bridge problem with increasing penalisation, p = [1,10]. The mesh is 125 × 125 with symmetry boundary conditions, a filtering radius of r_{min} = 1.5 is used. $C_{\mathfrak{p}}$ is the compliance calculated without penalisation for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities. Continued on next page.



(o) *p=8.0, $C_{\not{p}}$=8.64, iter=1000 (no conv.), $\tilde{x}$=0.789*

(p) *p=9.0, $C_{\not{p}}$=8.58, iter=1000 (no conv.), $\tilde{x}$=0.842*

(q) *p=10.0, $C_{\not{p}}$=38.56, iter=1000 (no conv.), $\tilde{x}$=0.888*

Figure 26 *Continued from last page. Optimised design for the bridge problem with increasing penalisation, p = [1,10]. The mesh is 125 × 125 with symmetry boundary conditions, a filtering radius of r_{min} = 1.5 is used. $C_{\not{p}}$ is the compliance calculated without penalisation for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities.*

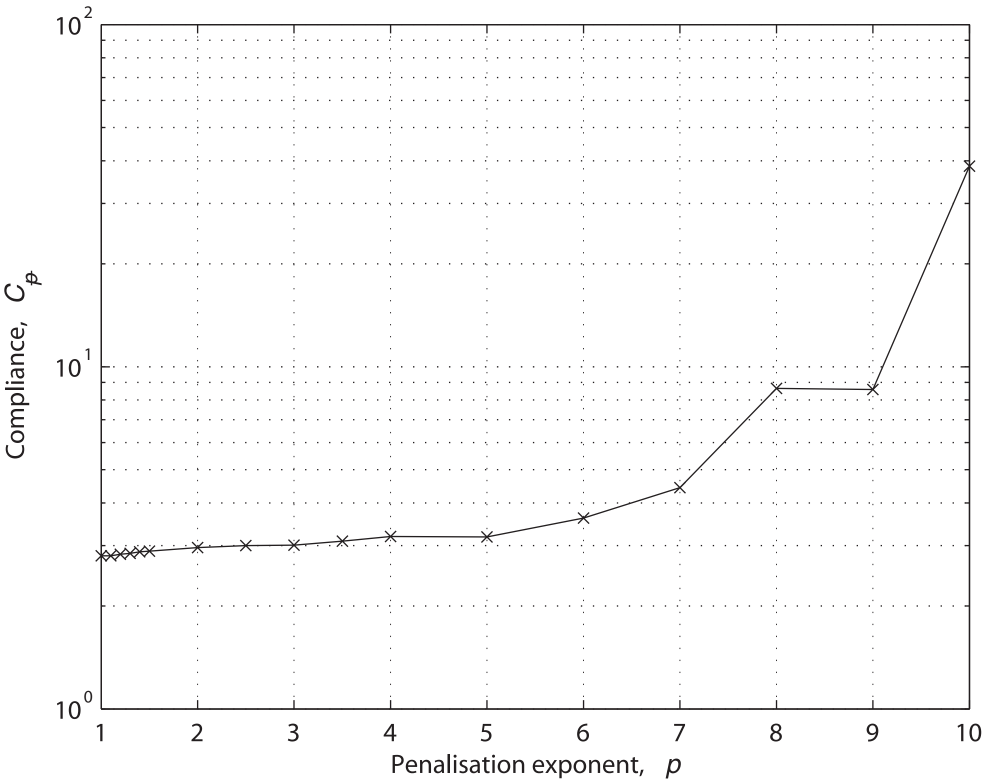
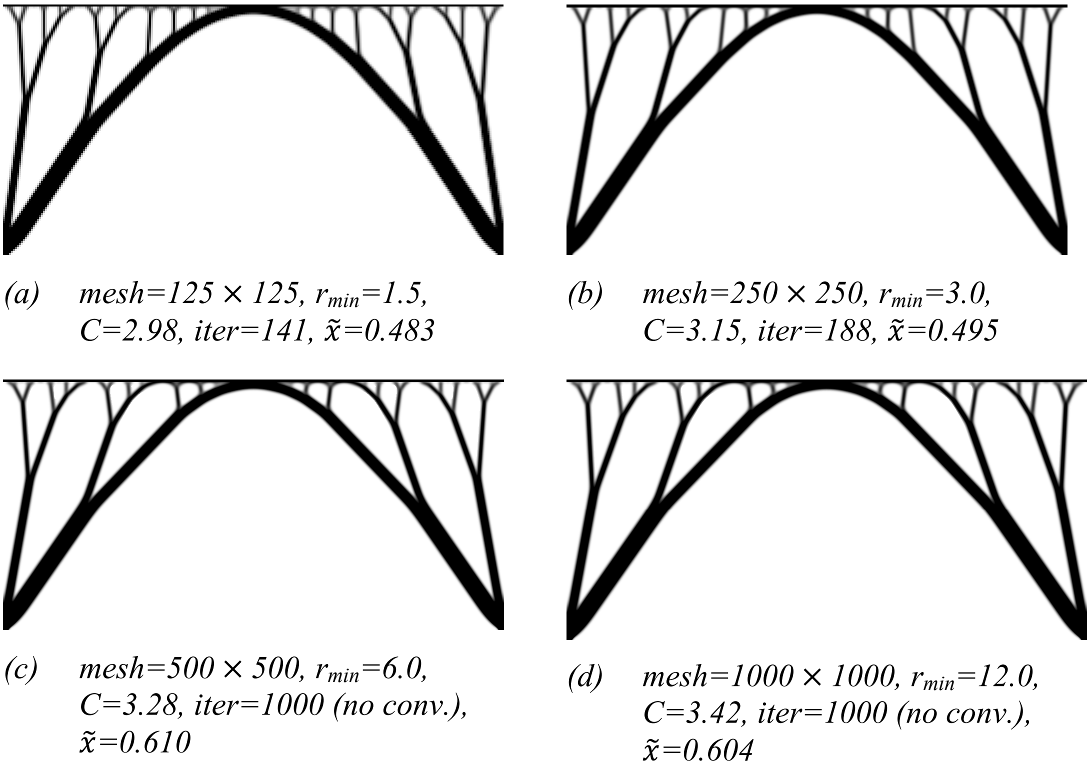


Figure 27 Compliance calculated without penalisation for optimisation result.

Instead of having a constant penalisation it is possible to start at a low value and increase as the optimisation proceeds. Figure 28a shows the results from such an optimisation where the penalisation is increased from $p = 1$ to $p = 3$ in increments of 10 % whenever the change of density in any element is 5 times the eventual convergence criterion of 0.01. The resulting topology is very similar to the low penalisation results except for the grey areas now having formed distinct members. The compliance has decreased with 1.0 % relative to the result using a constant $p = 3$ and the needed number of iterations has decreased with 4.1 %. In an attempt to incrementally increase the penalisation further, oscillations hinder it from exceeding $p = 3.45$. The result is then almost identical to the previous result, with a slightly increased compliance, as expected.

To illustrate that the results are not mesh-dependent, the same problem has been optimised with four different elements mesh sizes; for each mesh refinement the element size was halved and the filter radius measured in element widths was doubled. The results in Figure 28 have almost identical topologies whereas the increase in compliance is due to better accuracy of the finite element analysis. From Figure 29 it is evident that the optimisation strives towards members only in compression minimising bending in the bridge deck.



(a) *mesh=125 × 125, r_{min}=1.5, C=2.98, iter=141, $\tilde{x}$=0.483*

(b) *mesh=250 × 250, r_{min}=3.0, C=3.15, iter=188, $\tilde{x}$=0.495*

(c) *mesh=500 × 500, r_{min}=6.0, C=3.28, iter=1000 (no conv.), $\tilde{x}$=0.610*

(d) *mesh=1000 × 1000, r_{min}=12.0, C=3.42, iter=1000 (no conv.), $\tilde{x}$=0.604*

Figure 28 *Optimised design for the bridge problem for decreasing element size and accompanying increasing filter radius r_{min}. An increasing penalisation of p = [1,3] is used. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities.*

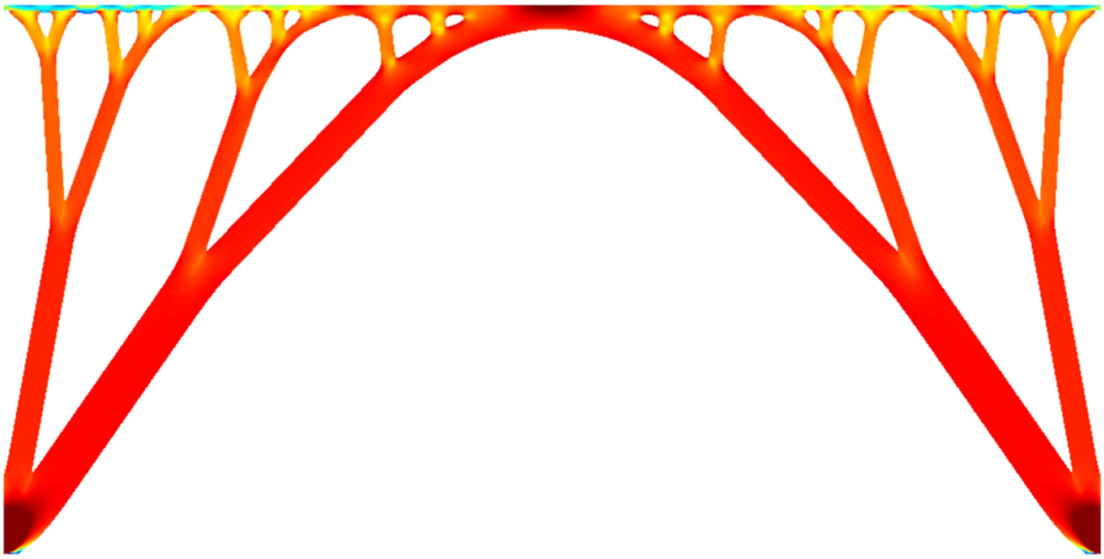


Figure 29 *Von Mises stresses of the solution in Figure 28c with the sign taken from the maximum principle stress, red to yellow being compression and cyan to blue being tension. Stress peaks at supports are cut off.*

So far focus has been on the numerical results of the objective function and other optimisation output from the optimisations. By instead focusing on the shapes that are produced, many interesting designs can be spotted. Figure 26(b) for example contains a large proportion of the material in diffuse grey areas but since the design variable can also be interpreted as the thickness of a plate, the grey areas are no longer a problem. Figure 30 visualise the same result but in plate thickness, while the plain stress assumption doesn't hold for this now three-dimensional structure it is still a design concept worth considering. Likewise the result in Figure 26(o) is interesting for its tensioned arches at the sides resisting the bending from the centre load and the results in Figure 26(p-q) are interesting for their simplicities.

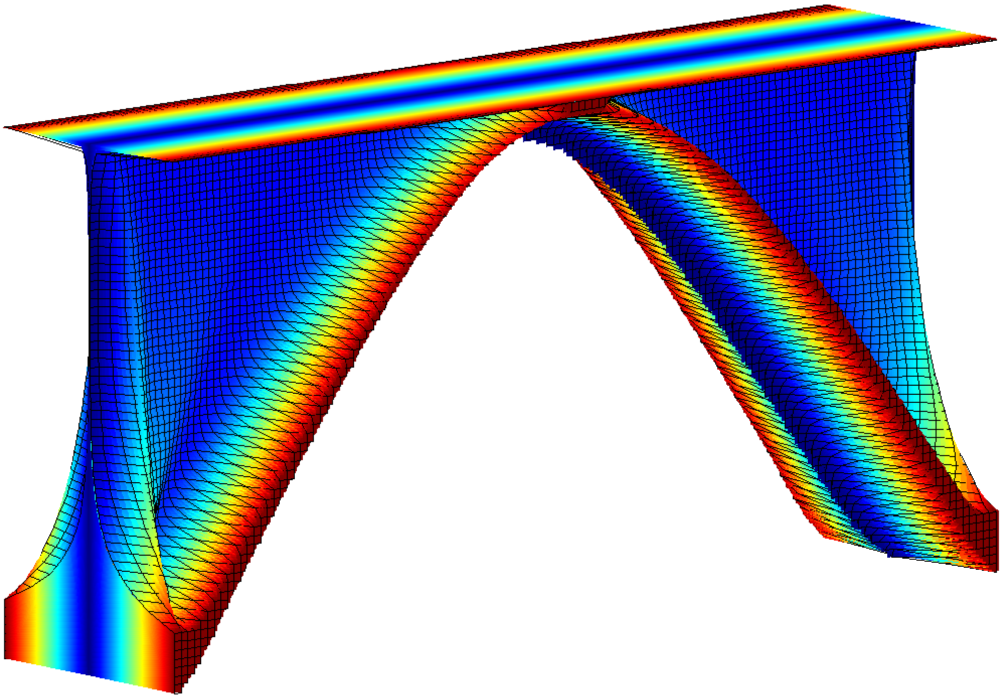


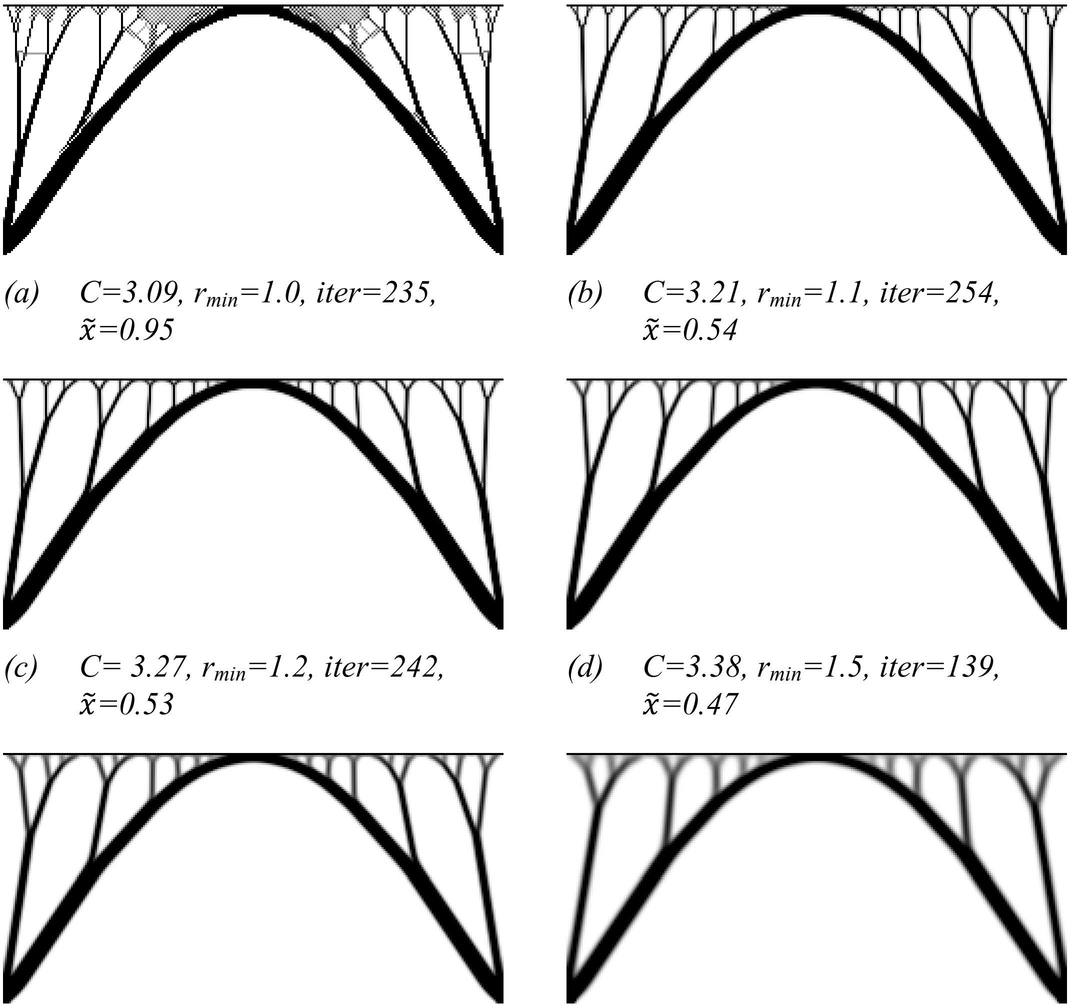
Figure 30 Densities from the 2D optimisation in Figure 26(b) interpreted as plate thickness, colour is distance from centre-plane.

3.7.2 Filtering variations

As the best result in the previous chapter was achieved by slowly increasing the penalisation, that setup will be used here when instead the filtering options are varied. Two filters are included in the original top88 code, one sensitivity filter and one density filter. Clear instructions are also given how two implement another two filters, an alternative density filter with increased performance and a black and white projection filter. In this chapter we study the effect of the two original filters and the projection filter, the results are compared in 3.7.2.4.

3.7.2.1 Sensitivity filter

This filter operates on the compliance sensitivities which in the density updating algorithm lead to prevention of occurrence of quick changes from black to white, effectively assuring a minimum radius for all members. The radius, $r_{min,}$ can be set as an option, setting it to anything between $r_{min} = 0$ and $r_{min} = 1$ element widths disables it and the result is in that case heavily affected by the checkerboard problem. The results from using a set of different filter radii can be seen in Figure 31. Increasing the radius to $r_{min} = 1.1$ eliminates the checkerboard problem but a grey border is introduced around every member. Increasing the minimum radius and the thinnest members start to vanish, although the main topology keeps the same up to around $r_{min} = 5.0$. Apart from the less defined member borders, the biggest changes are those of the positions of the connections between members.



(a) C=3.09, r_{min}=1.0, iter=235, $\tilde{x}$=0.95

(b) C=3.21, r_{min}=1.1, iter=254, $\tilde{x}$=0.54

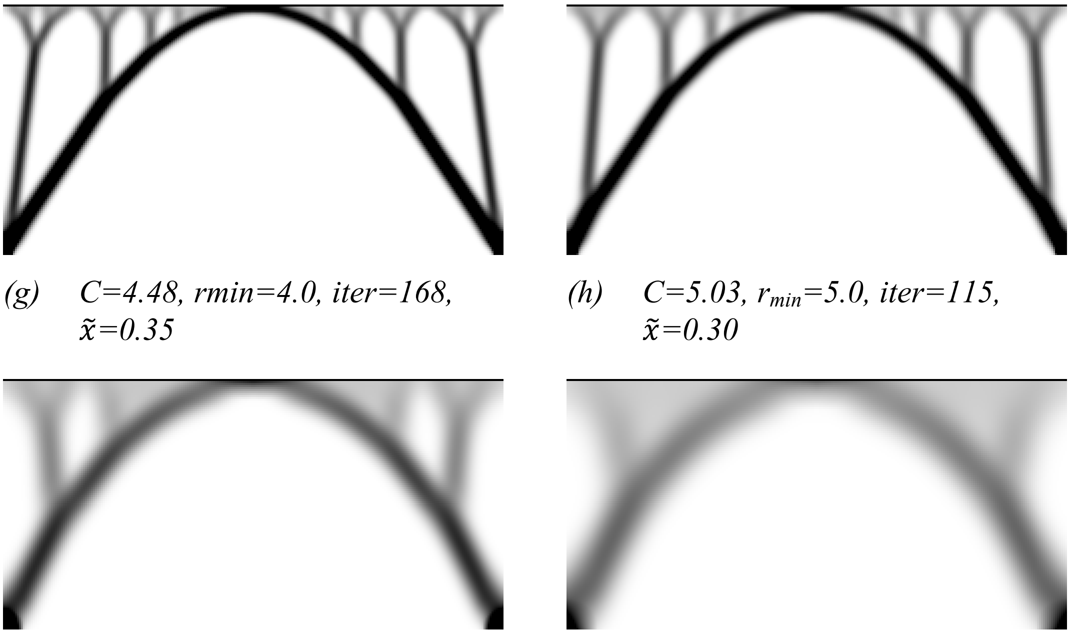
(c) C= 3.27, r_{min}=1.2, iter=242, $\tilde{x}$=0.53

(d) C=3.38, r_{min}=1.5, iter=139, $\tilde{x}$=0.47

(e) C=3.51, r_{min}=2.0, iter=183, $\tilde{x}$=0.47

(f) C=3.92, r_{min}=3.0, iter=173, $\tilde{x}$=0.39

Figure 31 Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15], using the sensitivity filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities. Continued on next page.



(g) C=4.48, rmin=4.0, iter=168, $\tilde{x}$=0.35

(h) C=5.03, r_{min}=5.0, iter=115, $\tilde{x}$=0.30

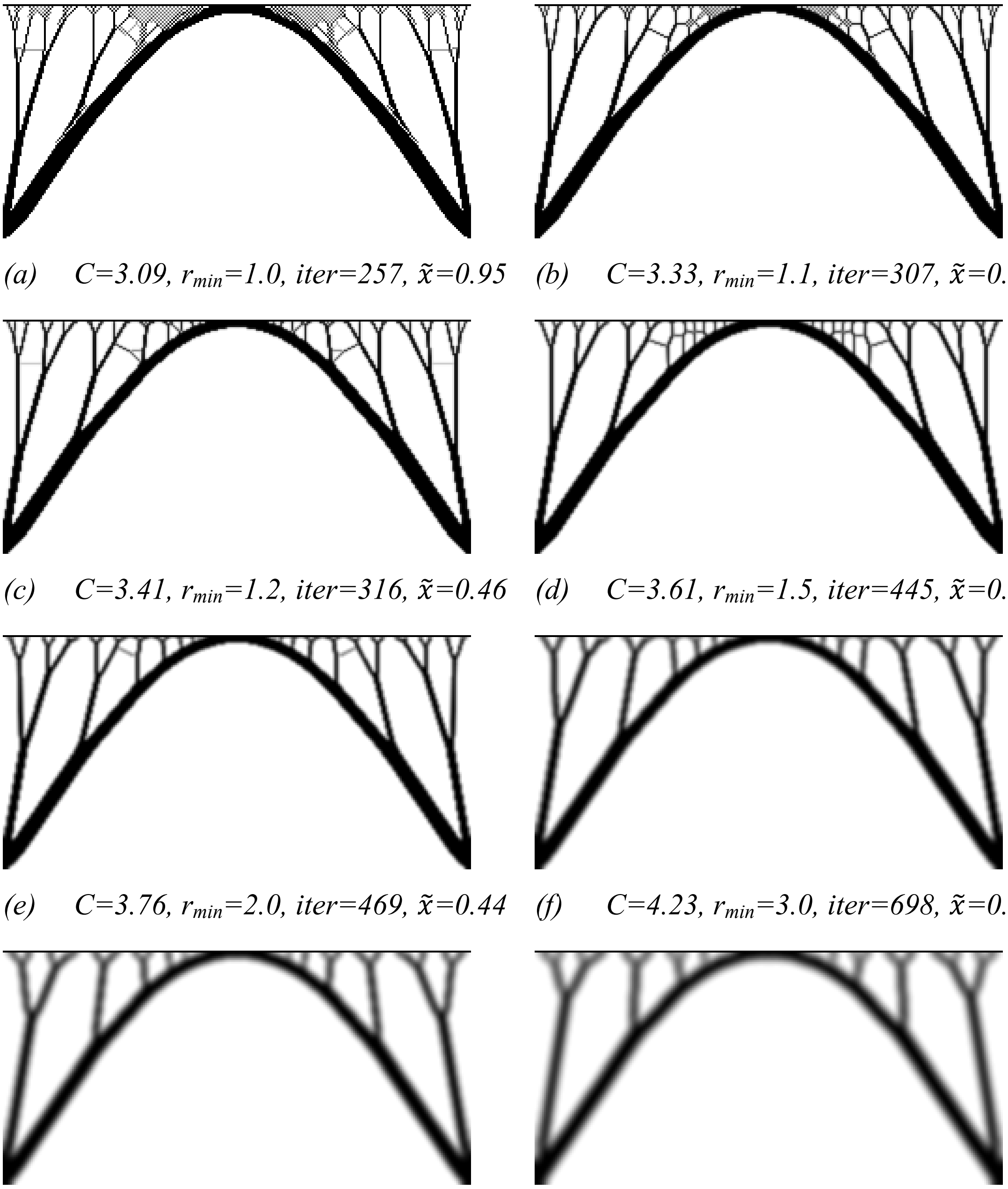
(i) C=9.20, r_{min}=10.0, iter=75, $\tilde{x}$=0.24

(j) C=14.59, r_{min}=15.0, iter=54, $\tilde{x}$=0.20

Figure 31 Continued from last page. Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15], using the sensitivity filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities.

3.7.2.2 Density filter

The second filter operates on the densities themselves, limiting the density gradient to prevent the formation of thinner members than specified by the minimum radius, r_{min}. The aim is therefore the same as with the sensitivity filter but the means and the results differ slightly, see Figure 32. With r_{min} = 1 the result is exactly the same as with the sensitivity filter. For a low filter radius of r_{min} = 1.1 to r_{min} = 1.5 the area that is last to glomerate into members forms a cell-like structure. For higher radii the result is very similar to the results using a sensitivity filter except for the struts being slightly bent at the forks and that a slightly higher compliance is obtained up to r_{min} = 4.0.



(a) *C=3.09, r_{min}=1.0, iter=257, $\tilde{x}$=0.95*

(b) *C=3.33, r_{min}=1.1, iter=307, $\tilde{x}$=0.46*

(c) *C=3.41, r_{min}=1.2, iter=316, $\tilde{x}$=0.46*

(d) *C=3.61, r_{min}=1.5, iter=445, $\tilde{x}$=0.45*

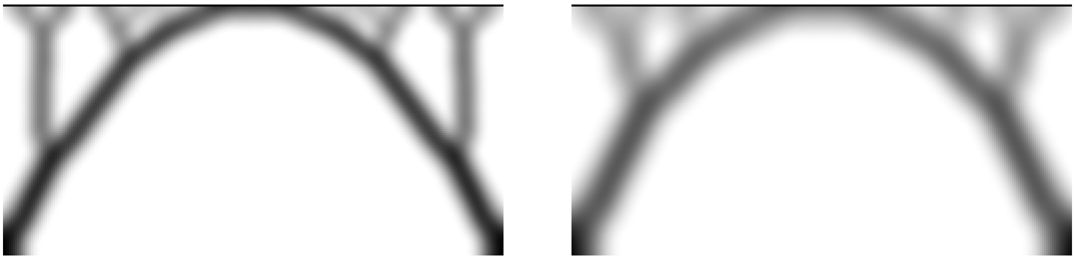
(e) *C=3.76, r_{min}=2.0, iter=469, $\tilde{x}$=0.44*

(f) *C=4.23, r_{min}=3.0, iter=698, $\tilde{x}$=0.39*

(g) *C=4.80, r_{min}=4.0, iter=732, $\tilde{x}$=0.35*

(h) *C=5.39, r_{min}=5.0, iter=1000 (no conv.), $\tilde{x}$=0.34*

Figure 32 *Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15] using the density filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities. Continued on next page.*



(i) *C=8.67, r_{min}=10.0, iter=1000 (no conv.), $\tilde{x}$=0.26*

(j) *C=12.14, r_{min}=15.0, iter=924, $\tilde{x}$=0.23*

Figure 32 Continued from last page. Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15] using the density filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities.

3.7.2.3 Heaviside projection filter

As implemented by Andreassen et al. (2010) the Heaviside projection filter is an extension to the density filter. The Heaviside step function is here defined as

$$H(x) = \begin{cases} 0 \; if \; x = 0 \\ 1 \; if \; x > 0 \end{cases} \tag{6}$$

where x in this case are the densities filtered by the density filter. $\sum H(x)$ is thus the number of elements with non-zero densities. To enable the use of the gradient based optimisation algorithm the Heaviside step function is made continuous by replacing it with a smooth approximation. To avoid numerical instability the approximation is gradually introduced as in Figure 33. If the penalty is increased during the optimisation the Heaviside projection filter extension is made active first after full penalisation is reached.

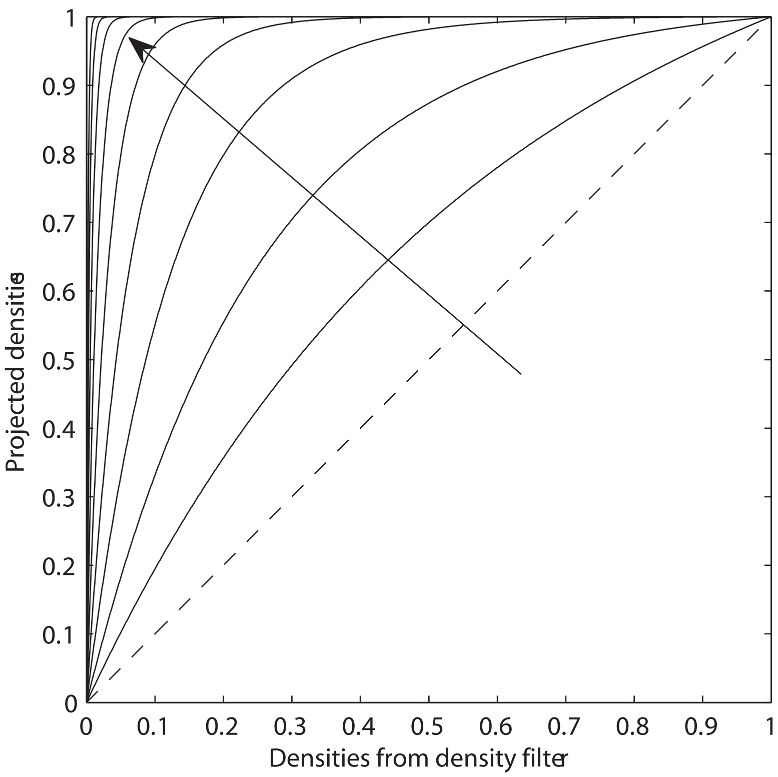
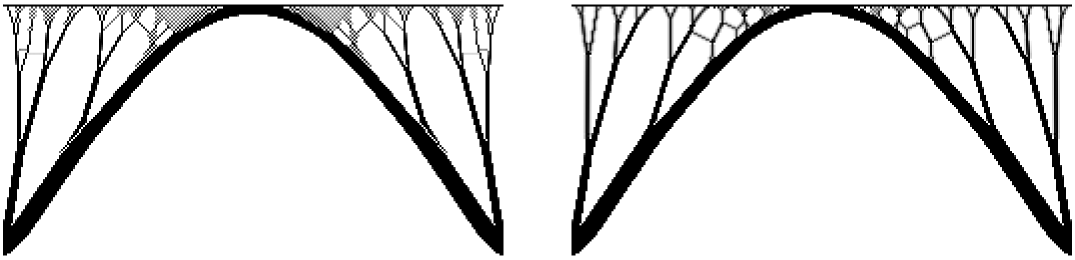


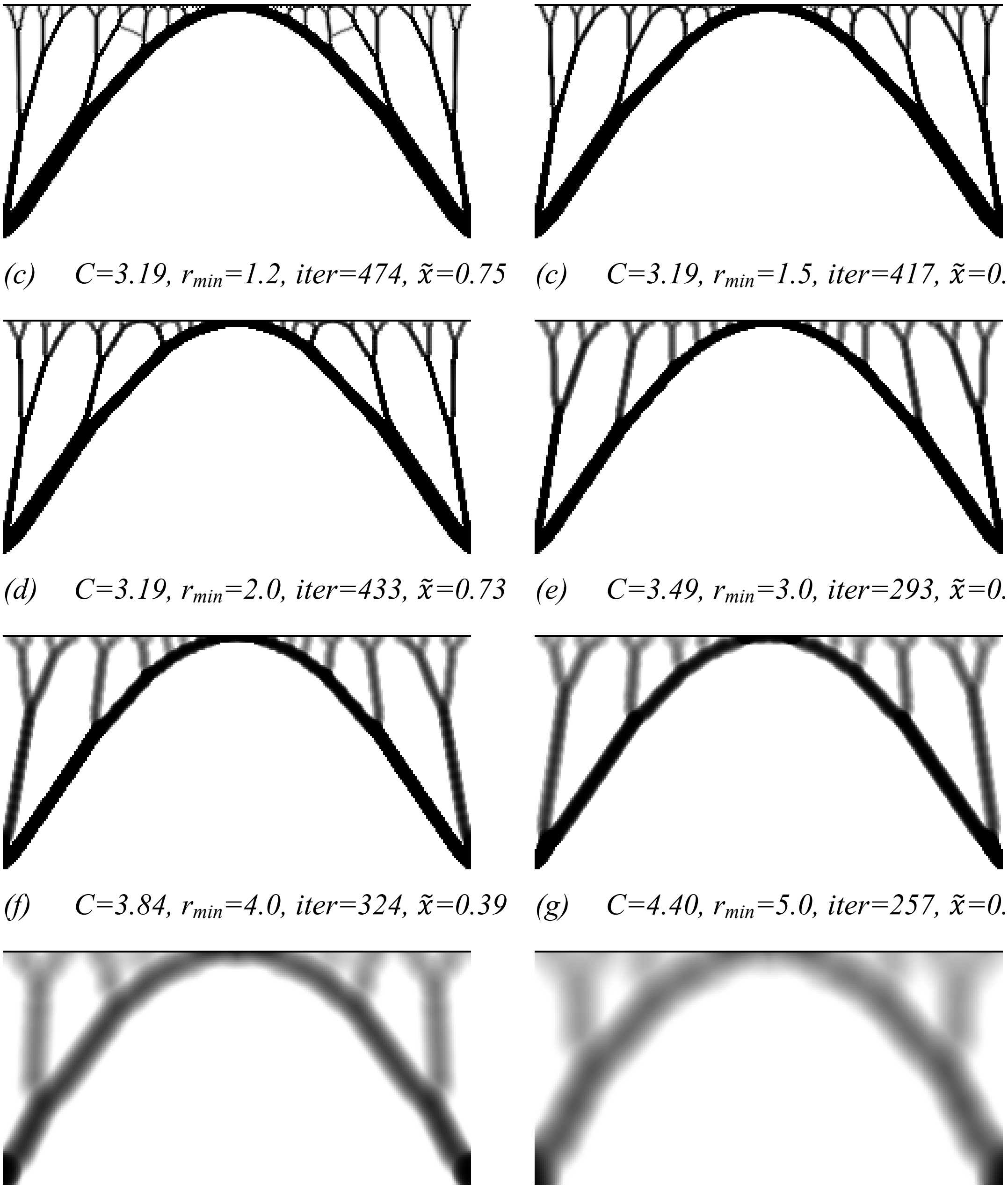
Figure 33 The Heaviside step function is gradually approximated in the direction of the arrow. The dashed line represents the starting point where the filter is inactive.

The results in Figure 34 show clear resemblance with the results from using only the density filter: the cell structure and the bent struts at the forks. The structures are generally more well-defined with less grey material, however the smallest members are in fact less well-defined. It is obvious that the thicker members resulting from a higher filter radius can't be all black due to the low volume constraint. This problem is more pronounced when using a distributed load or whenever the results include many thin members. Also relaxing the volume constraint will give better result.



(a) C=3.10, r_{min}=1.0, iter=433, $\tilde{x}$=0.95 *(b) C=3.19, r_{min}=1.1, iter=1000 (no conv.), $\tilde{x}$=0.80*

Figure 34 Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15], using the Heaviside projection filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities. Continued on next page.



(c) *C=3.19, r_{min}=1.2, iter=474, $\tilde{x}$=0.75*

(c) *C=3.19, r_{min}=1.5, iter=417, $\tilde{x}$=0.69*

(d) *C=3.19, r_{min}=2.0, iter=433, $\tilde{x}$=0.73*

(e) *C=3.49, r_{min}=3.0, iter=293, $\tilde{x}$=0.41*

(f) *C=3.84, r_{min}=4.0, iter=324, $\tilde{x}$=0.39*

(g) *C=4.40, r_{min}=5.0, iter=257, $\tilde{x}$=0.33*

(h) *C=8.29, r_{min}=10.0, iter=171, $\tilde{x}$=0.24*

(i) *C=12.23, r_{min}=15.0, iter=137, $\tilde{x}$=0.21*

Figure 34 *Continued from last page. Optimised design for the bridge problem with increasing minimum radius, r_{min} = [1,15], using the Heaviside projection filter. C is the compliance calculated for the resulting topology, iter is the number of required iterations for convergence and $\tilde{x}$ is the average of all non-zero design variable densities.*

3.7.2.4 Filter Comparison

The different filters generally produce very similar results, the Heaviside projection filter giving the smallest compliance for most filter radii partly because of the higher mean density and thus lower penalisation, see Figure 35-36. Except for small filter radii where the projection filter outperforms the other, the difference in mean density is too small to be of any interest. The largest differences in the results are found in the rates of convergence, an important property especially when dealing with large models. Also if the result from the topology optimisation should be manually simplified and maybe followed by a shape or optimisation, the efficiency of the filter is of greater importance than the precise accuracy of the result. As can be seen in Figure 37 the optimisation requires the most iterations when using the density filter and is for some radii not even able to converge within the maximum iterations limit of 1000. The fastest filter is the sensitivity filter, generally requiring less than half the number of iterations than the Heaviside projection filter.

Also in visual appearance do the filtering variations not produce results that differ that much from each other. They are mere a tool to avoid problems with checkerboarding and many very thin members.

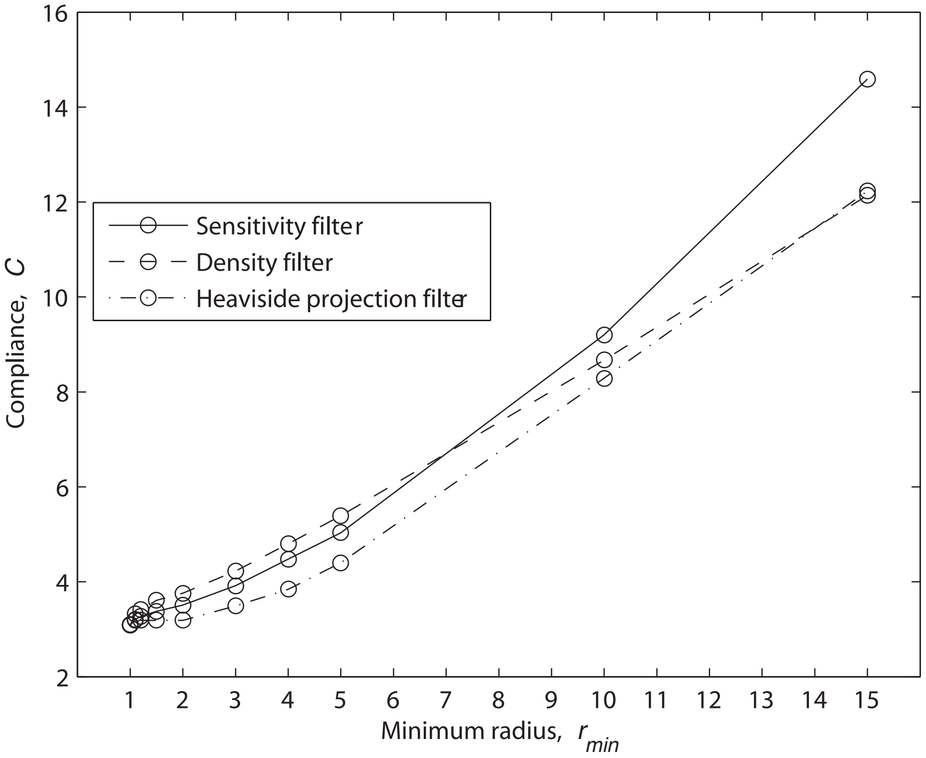


Figure 35 All filters produce results with increased compliance for increased minimum radius, mainly because of the increased amount of penalised grey material.

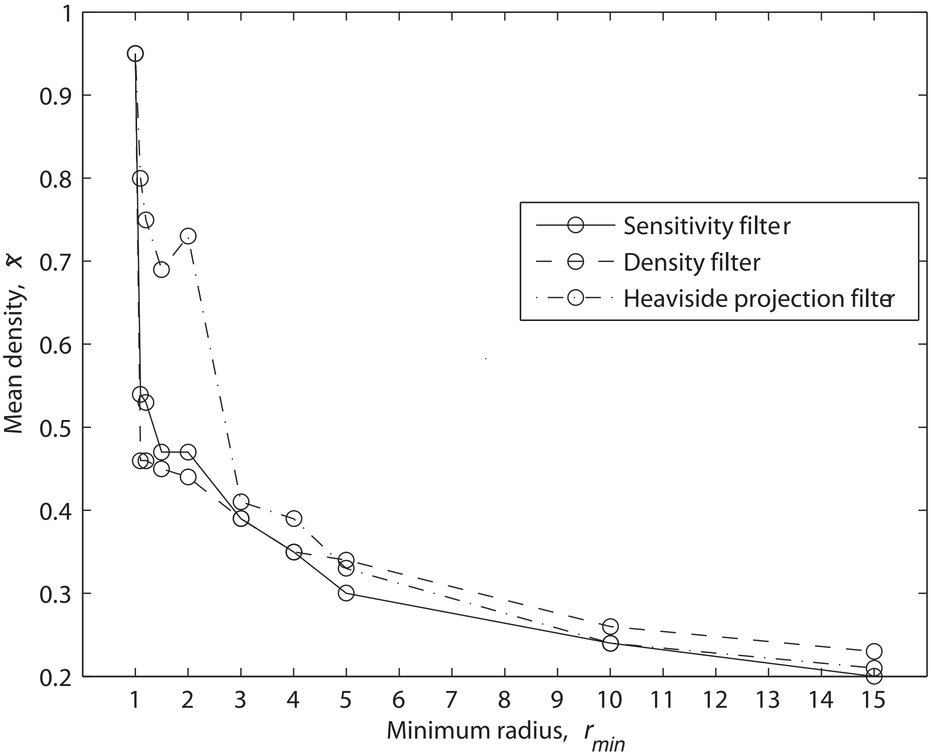


Figure 36 *The projection filter performs better than the other only for small filter radii.*

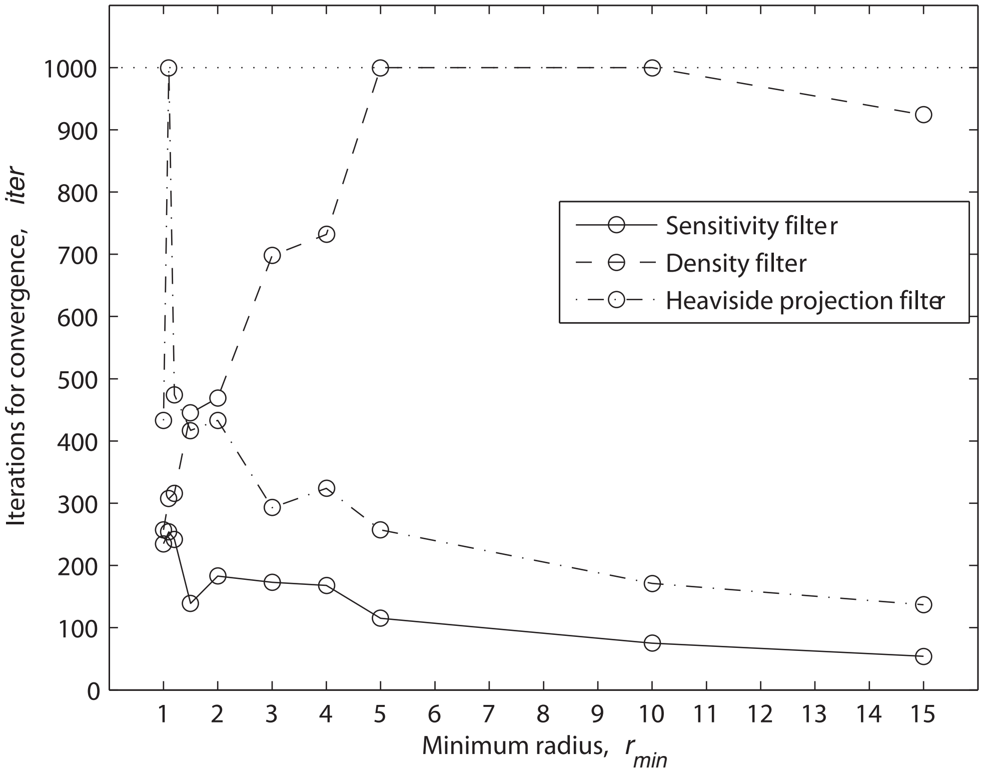


Figure 37 *The sensitivity filter shows excellent convergence properties, whereas the density filter often has problems to converge at all.*

3.8 Topology optimisation of a bridge including self-weight

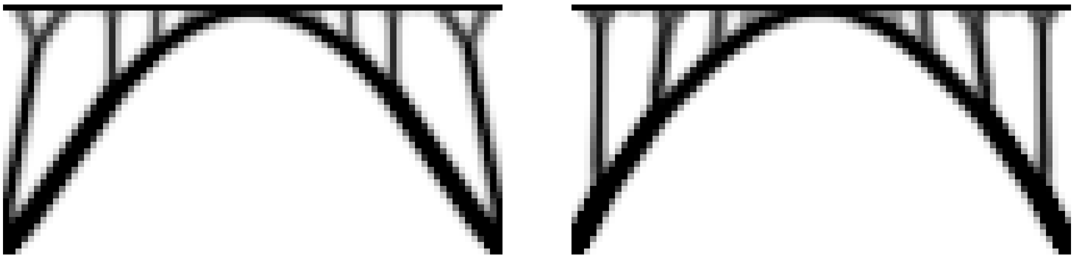
3.8.1 Arch bridge

In Section 3.7 the bridge problem was used to illustrate different penalisation and filtering options. Here self-weight and traffic load is applied to the same bridge problem. Refer to section 3.1.2 for more background on the self-weight problem as well as on the so called OC and MMA updating schemes. Due to the slow MMA subroutine that has to be used when self-weight is applied the mesh has been reduced to 40 × 40 elements. Throughout this section, real dimensions and material properties suitable for the bridge problem are used. The design domain is 50 × 50 m, i.e. every element is 1.25 × 1.25 m. The filter radius is set to r_{min} = 1.5 elements, which gave good results relatively fast in 3.7.2.1. MMA requires a higher penalisation to reach a near black and white solution; an increasing penalisation exponent of up to p = 5 is used. As mostly compressive stresses are expected unreinforced concrete is used as material with a density of 2700 kg/m^3, a Young's modulus of 30 GPa and a Poisson's ratio of 0.2. The optimisation parameters are summarised in Table 16.

Table 16 Optimisation parameters for the bridge with self-weight problem.

Parameter	Value
Optimisation objective	Minimise compliance
Optimisation constraint	$\bar{V}$ = 0.2 without self-weight $\bar{V}$ = 0.8 with self-weight
Penalisation	p = [1,3] for OC p = [1,5] for MMA
Mesh size	40 × 40

At first the problem is solved using both MMA and OC with traffic load only, see Figure 38 for the similar results. The penalisation when using OC is limited to 3 due to numerical instability. The MMA solution does have higher compliance but the difference is small. This is satisfactory as the next results cannot completely be derived to the change to use of MMA. Remark that even though the number of iterations is less when using MMA, the total time is almost five times longer.



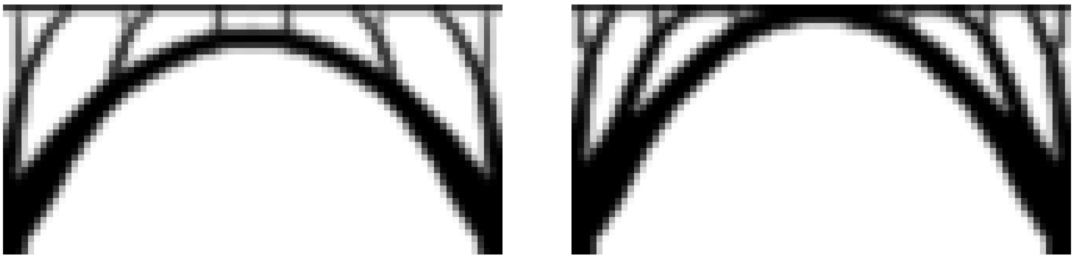
(a) OC. p=[1..3], $C_{\not{p}}$=7.67, iter=407, t=11.2 s, $\tilde{x}$=0.200

(b) MMA. p=[1..5], $C_{\not{p}}$=8.01, iter=260, t=52.4 s, $\tilde{x}$=0.201

Figure 38 Comparison between results obtained using either OC or MMA. The volume constraint is $\bar{V}$ = 0.2. The penalisation, p, needs to be higher for MMA to reach similar results. $C_{\not{p}}$ is the compliance calculated without penalisation for the resulting topology, iter is the number of required iterations for convergence, t is the total time and $\tilde{x}$ is the average of all non-zero design variable densities.

While the volume constraint for OC is stated as an equality, it is for MMA stated as an inequality meaning the volume is allowed to be less than the stated fraction. Without self-weight the structure with the largest volume is also the stiffest and best if optimised for minimum compliance. With self-weight applied a larger structural volume means greater stiffness but also more self-weight and possibly larger deformations, thus higher compliance. In this next optimisation only self-weight is applied and the volume constraint is set at $\bar{V}$ = 0.8 leaving for the optimisation process to determine also the optimal amount of material. When, as in this case, no non-structural load is applied, a few black prescribed elements or some initial density distribution is needed to get a solution other than the trivial all white. In this example the deck has this function.

The result in Figure 39a has a volume fraction of only V = 0.24 and the differences to previous results without self-weight are evident; the arc and struts gain thickness towards the base and have a curve between the connections. Applying a traffic load increases the volume fraction and give a topology that better supports the applied load, Figure 39b. Ideal a volume constraint equal to the resulting volume fraction of the first optimisation should be used but the process will then take another way steered by the constraint with a different, and worse, topology as a result.



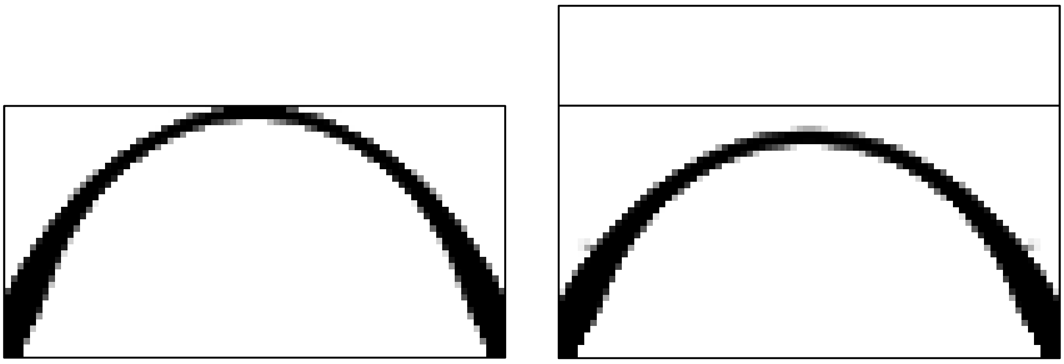
(a) q=0 kN/m t, V=0.243

(b) q=10 kN/m, V=0.291

Figure 39 Self-weight with and without traffic load, q. The volume constraints is $\bar{V}$ = 0.8.

3.8.2 Single arch

It is also interesting to look at the single arc without the deck. Bruyneel and Duysinx (2005) ensured a non-trivial solution for this problem by prescribing the top-mid elements to be black. By instead starting with a sufficiently dark flat density field and a design domain that is extended upwards, the height of the arc is no longer set a priori. The settings are the same as for the bridge problem except for the penalisation exponent only increasing to $p = 4$ which is enough for this problem. Interestingly enough the result in Figure 40 shows that the arch with the forced height uses less material and has a lower compliance. This means that the process is unable to find the global optimum solution but that additional constraints may help in finding it.



(a) Top-mid elements prescribed black. C=2.36, V=0.130

(b) No prescribed elements. C=2.51, V=0.135 (calculated for lower rectangle)

Figure 40 Only self-weight applied to rectangular design domain with and without prescribed height. The volume constraint is $\bar{V}$ = 0.8.

3.9 Topology optimisation of an S-shaped bridge in 3D

3.9.1 Model and parameters

Using the 3D version of Toptimiser this bridge is optimised for minimum compliance with volume constraint. The design domain with prescribed densities and boundary conditions is visualised in Figure 41. The problem is that of a road running parallel to another road, railway or stream, crossing the same at an angle. The shape of the bridge deck is sinusoidal in the plane and at constant elevation. Void volumes are prescribed both above the bridge deck and the crossed road. The bridge deck is the only volume prescribed to be full density and support is available at ground level at both sides of the bridge where the nodes are fixed in all three directions. The input parameters are collected in Table 17 and the script defining the problem for Toptimiser is found in Appendix A-5.

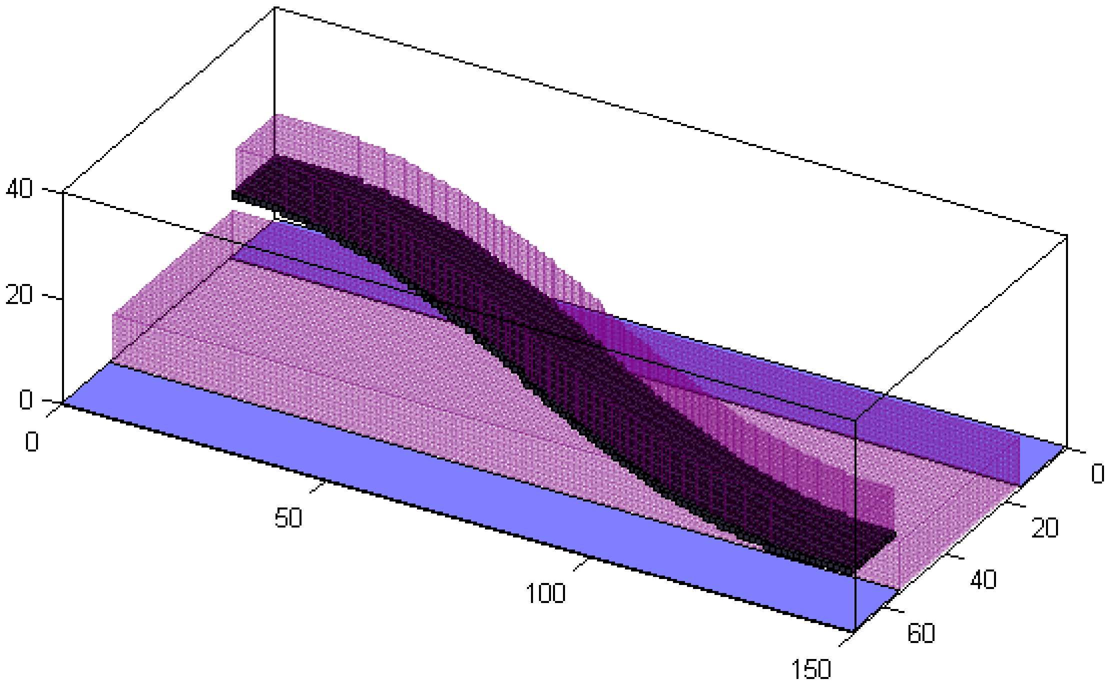


Figure 41 *Parallelepiped as design domain. Dark grey elements are prescribed full density, pink volumes are prescribed voids and blue areas are fixed in all directions. Scale is in meters.*

Table 17 *Parameters used for optimisation of the S-bridge.*

Parameter	Value
Optimisation objective	Minimise compliance
Optimisation constraint	$\bar{V} = 0.04$
Penalisation	$p = [1,3]$
Minimum radius	$r_{min} = 3.3$ elements (2.63 m)
Mesh size	136 × 64 × 36 elements

3.9.2 Results

The resulting topology is dominated by two arch-like structures in which the deck is hanging, see Figure 42. Horisontal and vertical projections are shown in Figure 43. At the corners of the design domain ordinary columns are placed along the edges of the deck and when the deck is about to turn in over the road the columns are instead placed at the deck's centreline. When no support is available under the bridge, structures are formed to lift the forces up to the arches that in turn transfer them down and to the sides. If self-weight had been applied during the optimisation the arches would have had a more curved shape instead of as now being completely straight between the connections. By varying the volume fraction and filter radius closely related alternatives can be generated. For less available material the two arches are tilted sideways and joined at one common foundation, saving on the groundwork. For a smaller filter radius the ties now being sheets separates into rods in a fan-like pattern. These modifications are also easily developed using engineering knowledge taking into account aspects the computer program didn't know about.

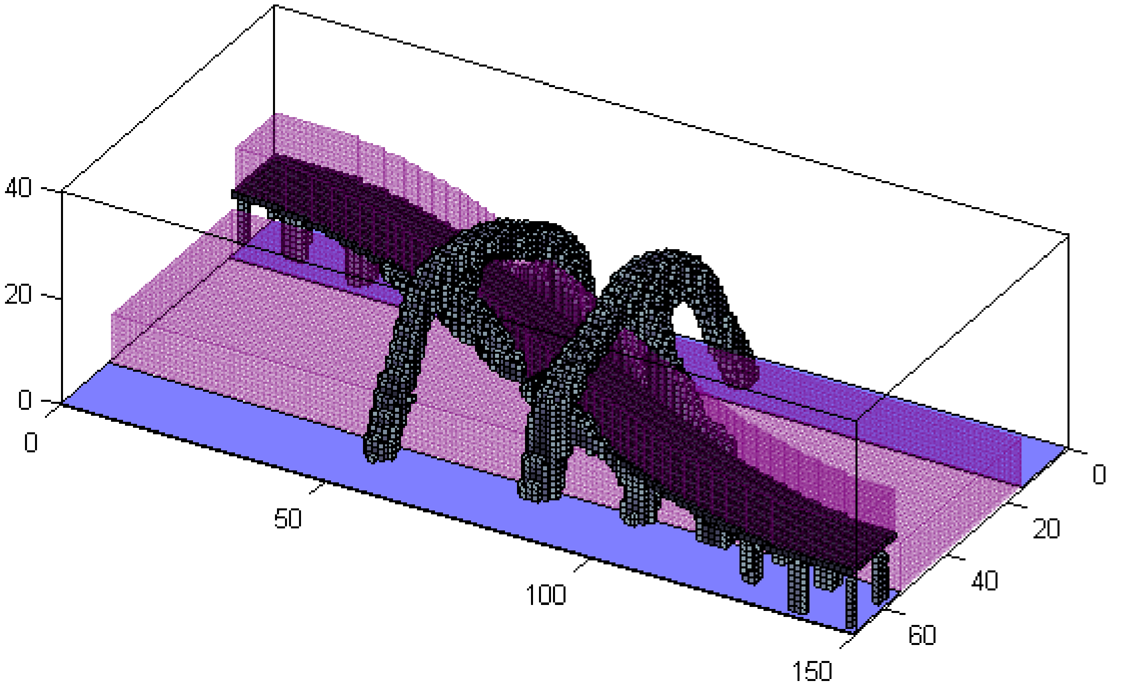


Figure 42 Result from topology optimisation. The greyscale shows the density of the elements, grey to black is 55-100 % density.

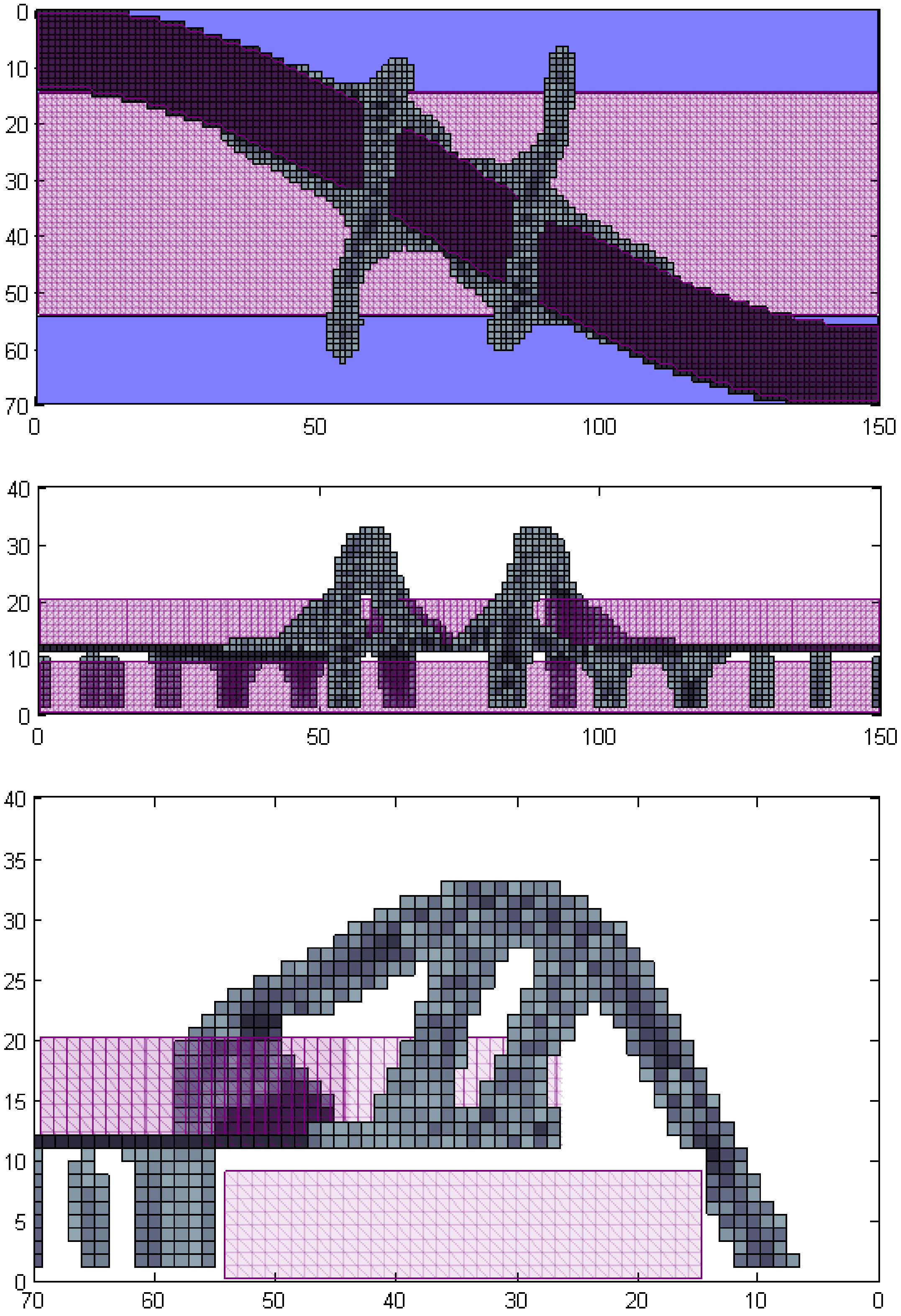
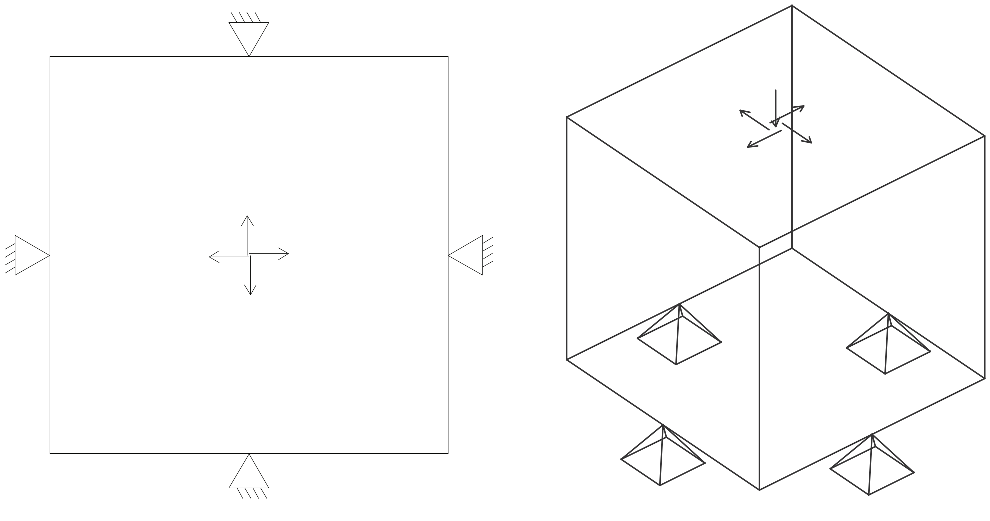


Figure 43 Vertical and horizontal views of the optimisation result. Bottom figure is looking from the right at half of the bridge. The greyscale shows the density of the elements, grey to black is 55-100 % density.

3.10 Topology optimisation of a torsional support in 2D and 3D

This topology optimisation is an example of how a 3D model can introduce unexpected effects not seen in the 2D counterpart. To begin with a square design domain is fixed at the mid-points on the four sides and a moment is acting at the centre created by two force-couples situated as close as possible to the centre. The model can be seen in Figure 44a and is very similar to an example worked out by Michell (1904). He fixed the structure in one point and applied one force in right angle to the line connecting the fixed point with the point of force application. Symmetry boundary conditions are used so that only one quarter of the original design domain is optimised; the nodes on the lines dividing the design domain in four squares are prevented to move in the directions of the lines. The optimised quarter is discretised into a 126 × 126 element mesh and the filter radius for the sensitivity filter is set to r_{min} = 1.1 elements. The volume is constrained to $\overline{V}$ = 0.2 and an increasing penalisation of p = [1,3] is used. A fine convergence tolerance of 0.001 gives a slower penalisation increase and the maximum number of iterations is set to 2000 to allow the optimisation to converge.

In the next step the design domain is extended upwards to become a cube with the fixed points at the bottom face's sides and the moment applied at the centre of the top side, see Figure 44b. The mesh is now reduced to 17 × 17 × 34 elements in each vertical quarter and a 2D optimisation with 17 × 17 elements is also done to compare with. With such a coarse mesh no advantage is made of an increasing penalisation; p = 3 is applied at once to reach convergence with fewer iterations. The volume constraint is for the 3D model lowered to $\overline{V}$ = 0.07 as a flat structure in 3D space, which is expected, occupy a small amount of the total volume. The optimisation parameters are summarised in Table 18. The scripts for the 2D and 3D optimisations are found in Appendix A-6 and Appendix A-7 respectively.



(a) 2D model. *(b) 3D model.*

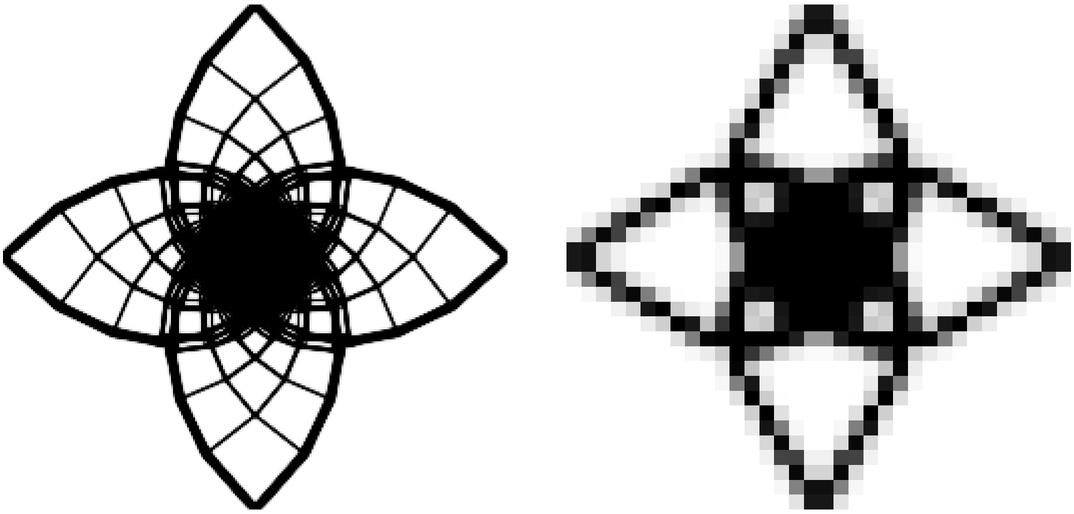
Figure 44 Design domain, boundary conditions and forces.

Table 18 Optimisation parameters for the torsional support problem.

Parameter	Value for 2D model	Value for 3D model
Optimisation objective	Minimise compliance	Minimise compliance
Optimisation constraint	$\bar{V} = 0.2$	$\bar{V} = 0.07$
Penalisation	$p = [1,3]$ and $p = 3$	$p = 3$
Minimum radius	$r_{min} = 1.1$ elements	$r_{min} = 1.1$ elements
Mesh size (each quarter)	126 × 126 and 17 × 17	17 × 17 × 34

3.10.1 Result of 2D optimisation

The result in Figure 45a is similar to that of the cantilever beam in section 3.3 and 3.4 due to the fact that the torsional support can be seen as four short cantilever beams joined at the supports. Each support point is connected to two equal spirals which enclose a set of inner spirals. The number of inner spirals is sensitive to the pace in which the penalisation is applied in addition to the volume constraint and filter radius. That is why a lower convergence tolerance was used. The result using the coarse mesh can be seen in Figure 45b. The outer spirals have become straight struts and ties as a result of the inner spirals having disappeared.



(a) Fine mesh, 126 × 126 elements in each quarter. *(b) Coarse mesh, 17 × 17 elements in each quarter.*

Figure 45 Optimal topology of the torsional support for two mesh sizes.

It should easily be noticed that if the moment is applied clockwise the spiral set rotating to the left outwards is in compression and the spiral set rotating to the right outwards is in tension. This is confirmed in Figure 46 which also shows the high stress concentration in the centre. The colour in Figure 46 represents the signed von

Mises stresses, i.e. the von Mises stresses with the sign taken from the largest principle stress in each element. The alternating blue and red pattern at the centre is only due to small fluctuations in the principle stresses, both are large but with opposite signs.

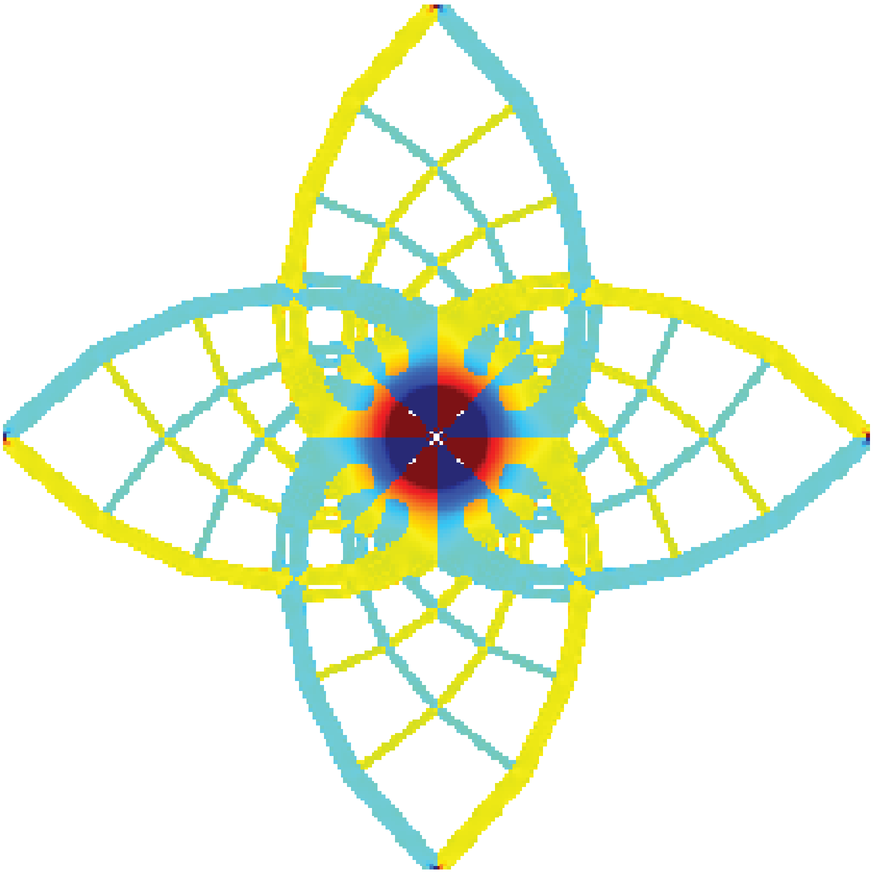


Figure 46 Von Mises stresses of the solution in Figure 45a with the sign taken from the maximum principle stress, red to yellow being tension and cyan to blue being compression. Stress peaks at supports and points of force application are cut off.

3.10.2 Result of 3D optimisation

Albeit not identical the result of the 3D optimisation in Figure 47 shows a clear resemblance with that of the 2D optimisation in Figure 45b. The longer distances from the moment to the supports enable the formation of more members, especially since the optimised structure brings the forces outwards to increase the internal level arms. This structure is by the look of it and by the fact that it is not optimised for it obviously not good at resisting vertical forces. Next, a force is therefore placed at the top centre node and the structure is again optimised with no other changes in the optimisation parameters. From having been independent of the direction of the moment the spiral set in tension has now almost completely disappeared, see Figure 48. The remaining spiral set has also straightened somewhat to better resist the vertical force. The explanation to the somewhat unexpected result lies in the vertical deformations. When the moment is applied to the optimised structure it will rotate and raise, a raise that is then counteracted by the vertical force. Also the rotations are discouraged since the moment and force will try to rotate the structure in opposite directions. If two load cases is used, the second with the moment rotating in the opposite direction or the force pulling upwards; the structure will again be independent of the direction, see Figure 49. The shape is now more of a cone where some remains of the spirals can be identified in the surface of the cone.

To reach a correct solution of this optimisation problem it is vital to know the magnitudes and directions of the moment and vertical force for the application at hand. It is also important to consider not only the global compliance, perhaps mainly the rotational stiffness is sought and the vertical stiffness is of lesser concern.

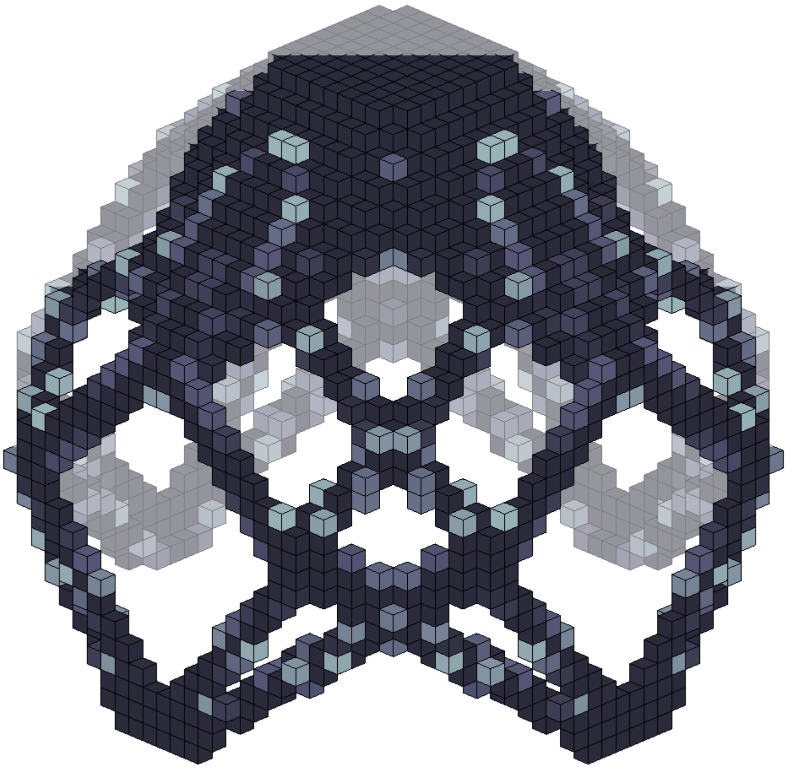


Figure 47 Without vertical force, the result is independent of the direction of the moment. A semi opaque plane divides the structure in two halves for better view.

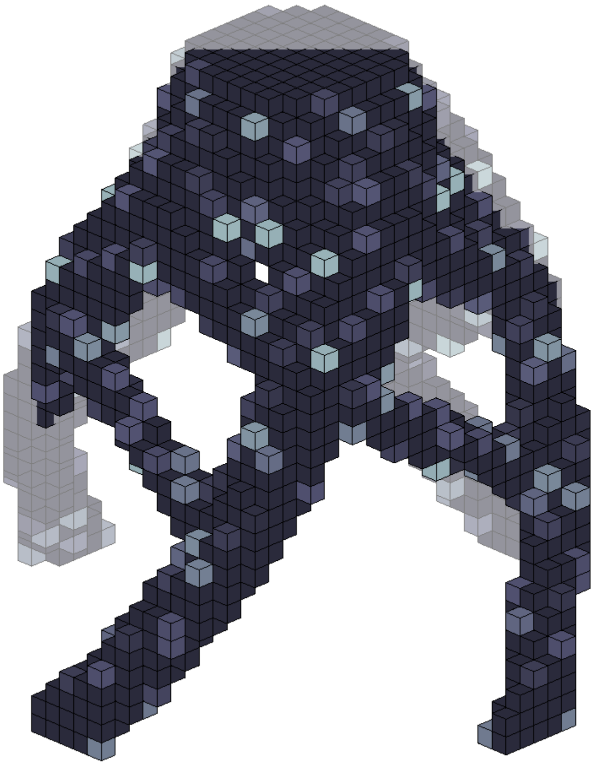


Figure 48 With vertical force, the tensioned spiral set has almost disappeared. A semi opaque plane divides the structure in two halves for better view.

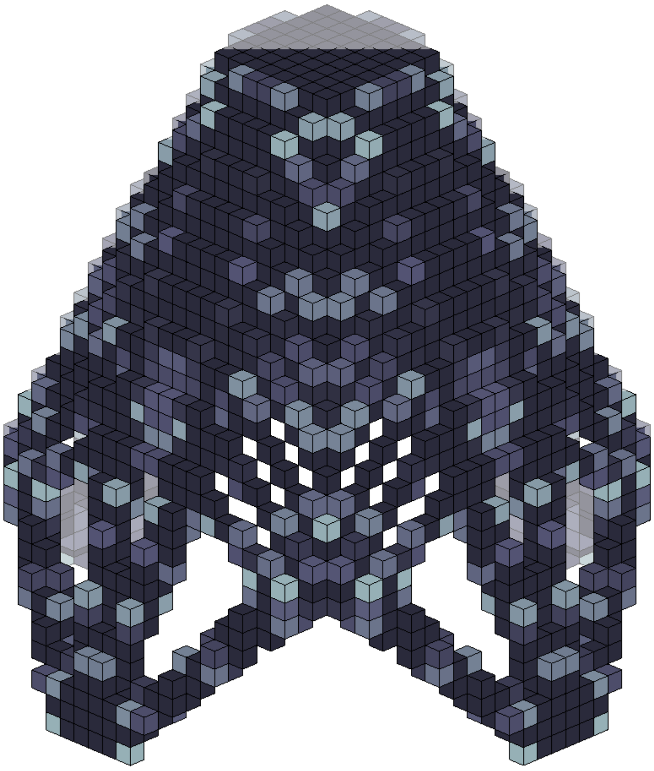


Figure 49 Vertical force and moment applied in both directions using two load cases. A semi opaque plane divides the structure in two halves for better view.

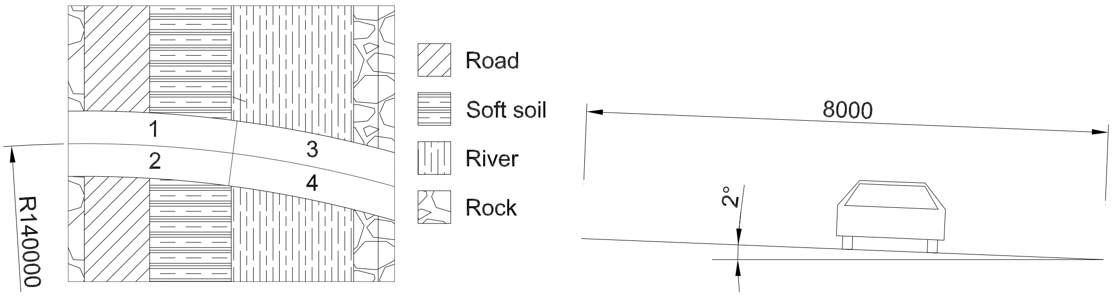
4 Multiple step optimisation of a bridge

In this chapter several previously studied aspects are put together, both size and shape as well as topology optimisation are applied consecutively to a realistic bridge problem. The geometry is not as simplified as earlier, real material properties are used and loads and constraints are set similar to current Swedish national regulations.

4.1 Geometry, loads and mesh

The bridge to be optimised is irregular in several ways. In order to make an optimisation of the bridge relevant and more interesting, a shape has been chosen that is not symmetrical and not straight. The bridge deck, i.e. the part of the bridge that has to be defined prior to the optimisation, is curved horizontally, and is set to slope. The roadway is also tilted somewhat towards the centre of the curve.

To get a realistic example, the geometry of the roadway is limited by regulations set by the Swedish Traffic Administration. This gives a radius of the curve which is 140 m, assuming the road has a maximum speed limit of 50 km/h and has good standard. The regulations also give a maximum slope of 6 % or 3.43° and a maximum tilt of 4 % or 2.29°. The width of the roadway is also chosen to be within the standard range. The geometry of the roadway of the bridge can be seen in Figure 50 and Figure 51.



(a) Plan view

(b) Section view

Figure 50 Plan and section view showing the geometrical demands on the bridge. Dimensions are in mm.

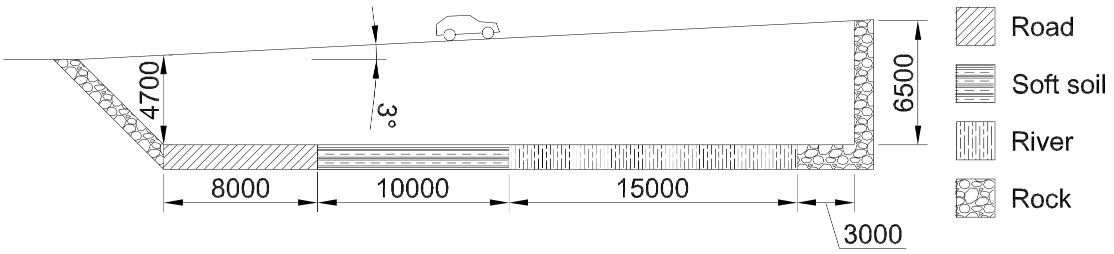


Figure 51 Elevation of the bridge with ground conditions. Dimensions are in mm.

The bridge is set to go from an embankment onto a cliff, spanning over a trafficked road and a small river. This can be seen in Figure 51. The distance between the road which the bridge passes over, and the bridge itself must be at least 4.7 m. The section between the road and the river is made of a soil with poor load carrying capacity, such as clay. The cliff is however assumed to be made of solid rock. These conditions generate an interesting scenario where the optimal topology isn't obvious, and where the optimisation process thus has a chance to show its potential.

The material parameters initially used are chosen to be similar to those of concrete. Young's modulus is set to 30 GPa and Poisson's ratio is 0.2. Since self-weight is included in the analysis, a density of 2500 kg/m^3 is used. The different soil types were supposed to be modelled by changing the stiffness of the elements. The soft, clay-like, soil between the road and the river was supposed to be modelled by heavily reducing the stiffness of the two bottom layers of the elements in that region. The thought was that by using such a low stiffness, it should be very expensive for the program to create supports in that area during the optimisation process i.e. supports would only be created in the soft parts if it was essential for the bridge. It was however found that this solution does not work in practice. Instead, the boundary conditions at the clay part are simply removed, making it impossible to place supports there at all. The embankment is not considered to be part of the optimisation problem, which is why stiffness there is set high, to allow for the creation of supports. The rock cliff is not created by changing stiffness, since it is at the edge of the design domain and is supposed to be absolutely stiff. The boundary conditions are thus set directly at the cliff face. The displacement is constrained in all directions for all boundary conditions used in this example.

The model is meshed with tetra-elements, i.e. triangular 3-dimensional elements with 4 faces and 4 nodes. The reason for the use of tetra-elements has to do with the meshing capabilities of HyperWorks, making tetra-elements the only reasonable choice. The tetras are allowed to differ in size and can be between 400 mm and 1000 mm. By allowing different sizes of the element, the program creates a better mesh with fewer bad elements. This element size gives a mesh with a total of 1078735 elements. Although a finer mesh would be preferable, the available computers limit the mesh size. The mesh is however fine enough to give acceptable topologies.

The loads applied are chosen to resemble the load of real traffic, although major simplifications are made to the standard load model. This has given one distributed load with the magnitude 10 kN/m^2, which is somewhat higher than the distributed load in the standard load model. No bogie loads are used, which makes the increased distributed load reasonable. The area, on which the load applies, is split in four parts for the topology optimisation, which is seen in Figure 50a. This is done to be able to create several load cases for which to optimise the bridge. 5 load cases are used in this example, 1 where the load is applied in all areas as well as self-weight, and 4 where the load is applied in one of the fields together with self-weight. Later on, during the size and shape optimisations, the areas and load cases are adjusted to fit in the spans of the structure.

4.2 Topology Optimisation

The first optimisation to be performed on the design domain, seen in Figure 52, is a topology optimisation. The optimisation is performed with the objective to minimise the global compliance, with a limit on the volume. This is to get an overall shape, which is optimal with regard to stiffness and gives a good topology to optimise further. An additional constraint on the deflections at some points along the length of the bridge is applied, to ensure that no excessive deflections occur. A summary of the used optimisation parameters is found in Table 19. The reason for not considering the stresses is that the optimisation of the bridge is done in several steps, and in this step the overall shape of the topology is the most important. Also, parts with too high stresses can be optimised in the later steps.

Table 19 Optimisation parameters used in the initial optimisation performed in this example.

Optimisation parameter	Value
Optimisation objective	Minimise global compliance
Optimisation constraint	Limit on volume (V < 7 % of initial volume) Deflection constraint (L/400)
Penalisation factor	4
Minimum dimension	1.2 m

The initial design domain can be seen in Figure 52. The design domain is essentially a box, where the space for the traffic of the bridge, the underlying road and some space above the river has been cut out. The height of the box is set to be high enough, so that the created topologies won't reach up to the top, but at the same time not unnecessarily high to avoid having unused elements, in order to keep the computation time down.

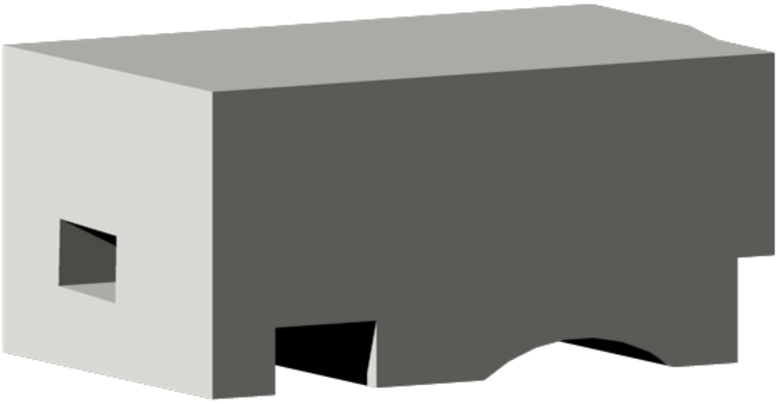


Figure 52 The initial design domain to be optimised.

4.2.1 Deflection constraint

Setting a constraint on the deflection as a fraction of the span length, as is done in most codes, is difficult since the span length is unknown prior to the optimisation. The method used here is to revise the deflection constraint depending on the span length of the last optimisation and then start a new optimisation until the span length converges, i.e. for the first optimisation the span length is assumed to be the full length of the bridge, then a optimisation is run, a topology is created and the deflection limit is adjusted to the span length of the new topology. Another problem is where to place the constraints. One possibility is of course to constrain all nodes in the deck. This will however severely increase the computation time and, to our experience, also make it difficult for the program to find a solution at all. Too few constrained nodes on the other hand may lead to a solution with large deflections in unconstrained parts of the deck. By choosing a handful of nodes, spread out over the length and width of the deck and setting the constraint relative to the length of the span from the previous optimisation, a good solution may be achieved in a reasonable amount of time.

4.2.2 Volume constraint

The volume constraint in this example is applied by choosing a maximum volume fraction, i.e. a percentage of the design domain, which is set as a limit on how much material the resulting topology can use. Choosing the right volume fraction is not a trivial task. While a higher volume fraction most often gives a stiffer topology, too high volume fraction can cause the topology to have big lumps of material at some points. Reducing the volume fraction reduces the risk for creation of lumps, but with a too low volume fraction no proper topology at all can be created.

To find the best value for the volume fraction the topology optimisation was performed several times with different values for the volume fraction. The chosen value, 7 % of the initial volume, was the one that gave the best compromise between well-defined topology and few big lumps.

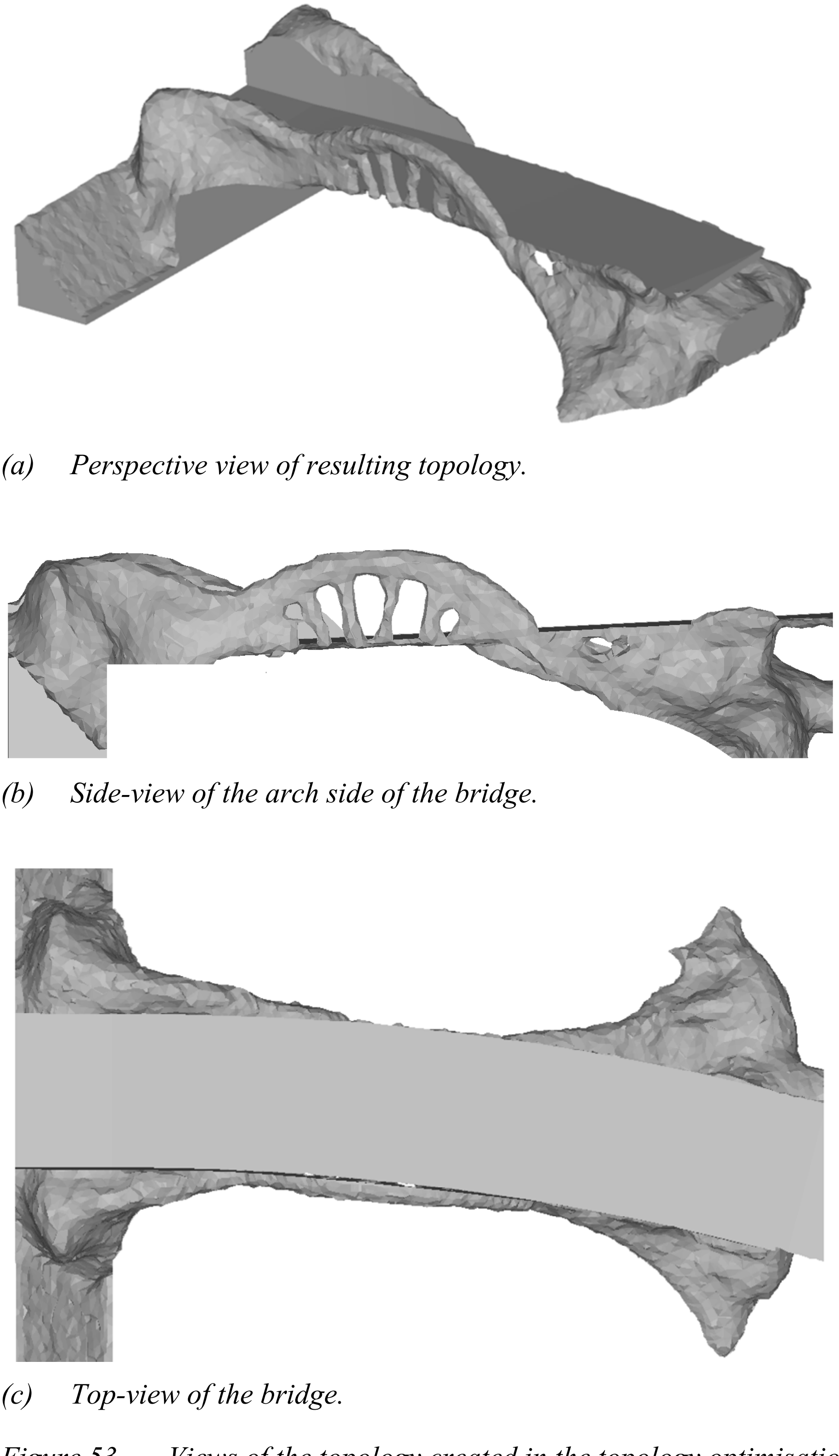
4.2.3 Results from topology optimisation

The topology optimisation resulted in a topology with several different types of structural elements, which is seen in Figure 53. The topology in the figure has been smoothed, in order to better show the shapes of the topology. The topology features an arch on one side of the bridge. The arch is centred on the bridge but the ends of the arch are not supported by the boundary conditions of the model and are therefore in need of additional support. This is achieved by two lumps of material on one side, which mainly are in tension, and an underlying compressed structure on the other side.

The maximum deflection, which is found at the centre of the deck widthwise and at the middle of the arch lengthwise, is 3.24 mm. This is far from the applied limit of 90 mm, showing that the deflection constraint in this case was not necessary. It can also be noted that an adjustment of the deflection limit, in order to make it better correspond to the actual span length, is not necessary. Such an adjustment would only slightly lower the limit, which would not affect the results.

The highest stresses are found at the supports at the side of the bridge close to the underlying road, i.e. the supports to the left in Figure 53. High stresses are also found where the arch connects to the roadway. The overall magnitude of the stresses is however low. The highest von Mises stress found in the structure is 11 MPa, while the stress in most of the structure doesn’t exceed 2 MPa.

Since the bridge will be optimised further, the numerical results are not of much importance. Instead it is the shape that plays the significant role. The shape is, as can be seen in Figure 53, very irregular. To perform the further optimisations, the structural members of the bridge need to be somewhat clearly defined. Obviously, this is not the case for the topology created by the topology optimisation. It would theoretically be possible to alter the resulting topology to having more well-defined members by manipulation of the mesh. This is however a task that is very time consuming, and an easier way, and maybe a way that gives a more reasonable solution, is to redraw the bridge from scratch, with the optimised topology as model.



(a) Perspective view of resulting topology.

(b) Side-view of the arch side of the bridge.

(c) Top-view of the bridge.

Figure 53 Views of the topology created in the topology optimisation.

4.3 Shape optimisation

The shape optimisation is done on the results from the topology optimisation. This can however not be done directly, since the topology is not very smooth, as previous explained in section 4.2.3. It is also too complex to make sense, with places with big lumps of material. To make a more simple structure, the topology created during the optimisation is interpreted and a new model is built around the topology. Some differences between them can be seen. The lumps at the embankment side of the bridge are replaced by a cable-stayed structure. This is due to the fact that big parts of the lumps are in tension, and a cable-stayed structure is an alternative that takes tensile forces well and that somewhat corresponds to the shape of the lumps.

On the other side of the bridge, the three slanted columns underneath the bridge are reduced to two, one on either side of the bridge. They are however still pointing outwards from the bridge, as in the topology created in the topology optimisation. The full interpretation of the topology is seen in Figure 54. The cables, the ties in the arch as well as the connections between the slanted columns and the deck are modelled with rod-elements, i.e. bar elements what only can take axial loads.

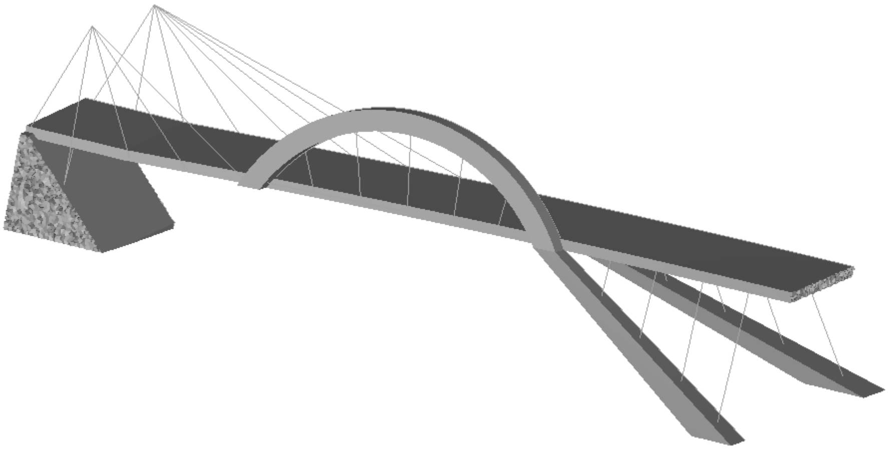


Figure 54 The interpreted bridge design from the topology optimisation result.

The shape optimisation is performed with the objective to minimise the global volume of the structure. Stress limitations are applied on the von Mises stresses, which is set to 20 MPa. This way, the volume of the structure will be reduced until the stresses reach the limit. An overview over the used optimisation parameters can be seen in Table 20.

For the shape optimisation, the load cases used have been changed. Instead of having five load cases as for the topology optimisation, three load cases are now used. This is due to that the topology of the bridge now is known. The load cases are now based on that the bridge has two spans: One consisting of the arch and the cable-stayed part, and the other of the part supported by the slanted columns. The new load cases are consequently either load in one of the spans or in both. Self-weight is also included in all load cases.

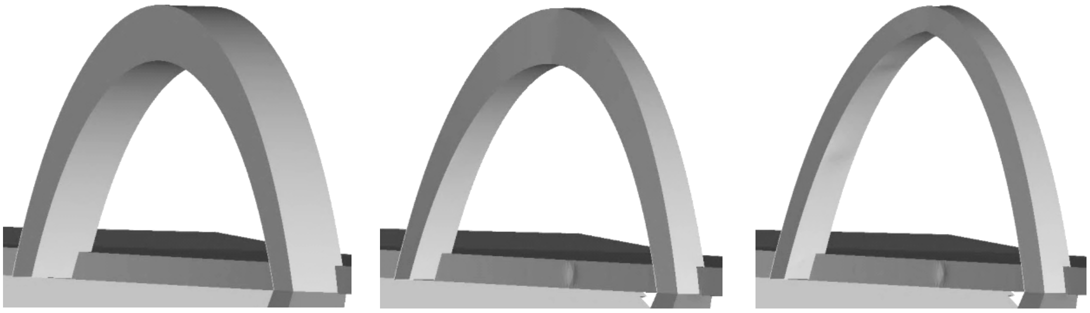
Table 20 Optimisation parameters used for the shape optimisation performed in this example.

Optimisation parameter	Value
Optimisation objective	Minimise volume
Optimisation constraint	Stress limit on von Mises stresses. (20 MPa)
Number of shapes/design variables	6

The shape optimisation is performed on three parts of the bridge: the arch, and the two slanted columns. These are the parts that are still made of solid elements and which are likely to benefit from a shape optimisation.

The shape optimisation is performed as in section 3.3.1, were shapes are predefined by the use of handles. The shapes are saved as node displacement, and the nodes are allowed to move from full to no displacement enabling the shapes to be applied to some degree.

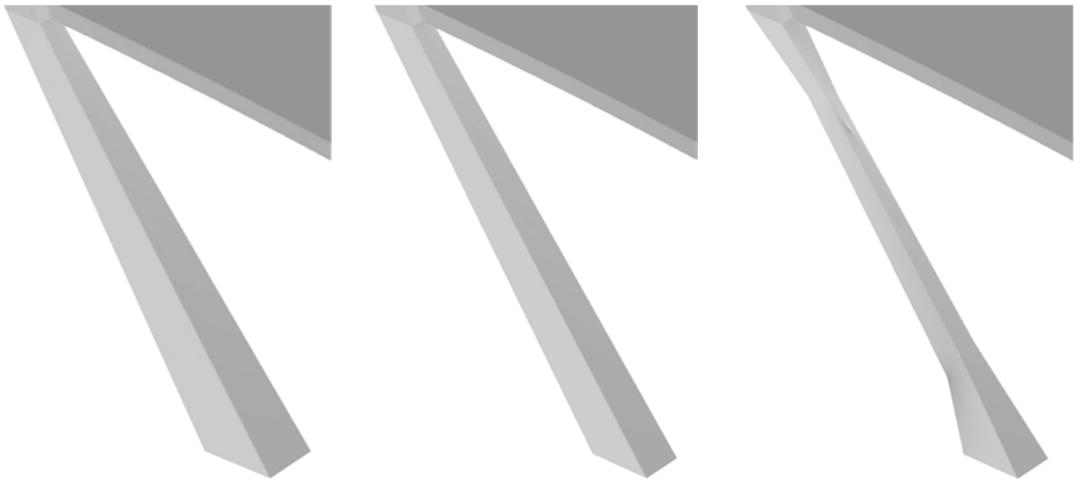
Two shapes are used for the arch, changing the horizontal and vertical widths of the arc. Since one of the slanted columns is directly attached to the arch, the change of shape of the arch also changes the top of the column. Possible shapes for the arch are seen in Figure 55.



(a) No shapes applied. *(b) One shape applied.* *(c) Both shapes applied.*

Figure 55 Shapes that can be applied to the arch during shape optimisation.

For the column two additional shapes can be applied: One that changes the dimensions of the support section, i.e. the bottom section where the boundary conditions are applied, and one that changes the dimensions of the mid part of the column. These can be seen in Figure 56. The shapes applied to the other column are very similar.



(a) No shapes applied. *(b) One shape applied.* *(c) Both shapes applied.*

Figure 56 Shapes applied to one of the supports.

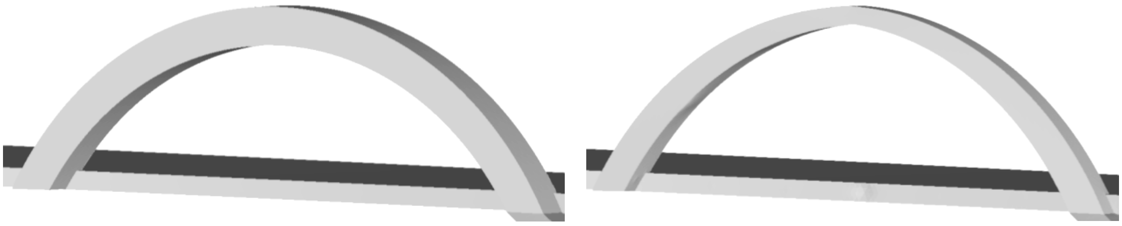
The limit on how much the shapes in this example can be applied, is set by the deformability of the elements. The shapes are chosen so that no further node movements can be achieved without distorting the elements too much when the shapes are fully applied. This is also the reason for the columns to have a bigger cross-section at the ends, where the elements are arranged in such a way that makes it harder to decrease the size of the section without distorting them.

4.3.1 Stress constraint

The stress constraint in this example is applied by defining a maximum limit on the von Mises-stresses. The limit is set to 20 MPa to be relevant for concrete in compression. Most of the members optimised in the shape optimisation in this example, such as the arch, are also assumed to be mostly in compression. The reason for introducing the stress constraint at this point of the optimisation is that the overall shape of the structure is now defined. It is thus left to optimise the individual members, rather than the whole structure. Since no regard has been taken to the stress earlier in the optimisation of the bridge, the stresses can be high, and it is reasonable at this point to try and make the members as small as possible without exceeding the stress limit.

4.3.2 Results from shape optimisation

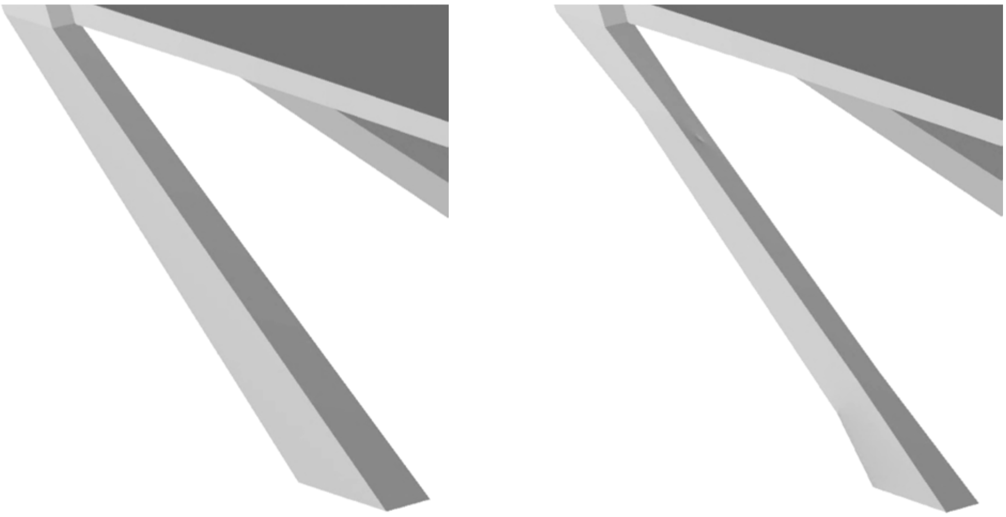
The results from the shape optimisation show that the dimensions could be reduced, without exceeding the stress limit. The arch and the slanted column attached to the arch both have substantial decrease in volume. The other support has less decrease in volume, but still a change in shape.



(a) Shape before optimisation. *(b) Shape after optimisation.*

Figure 57 Shapes of the arch before and after shape optimisation.

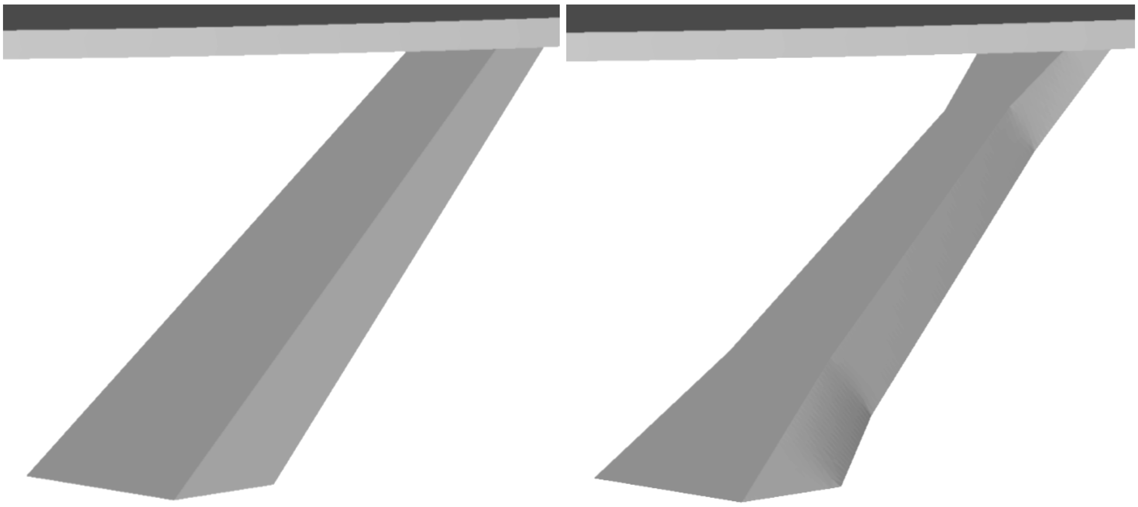
As can be read in Section 3.3.1, it is preferable if the nodes do not reach the limit on how much they are allowed to move during a shape optimisation. For the arch of the bridge optimised in this example, both the shapes do reach the limit. This implies that the change of shape, if allowed, would have continued and the size then would have been reduced further. As stated previously in this chapter, what limits shapes used for this example is the deformability of the elements. This means that it is not possible to allow for greater movement and greater morphing of the shape, since that would lead to elements with bad quality, and also a model that doesn't work. For this report, it is therefore sufficient to say, that the volume of the arch is reduced, but can probably be reduced a bit further. This is confirmed by the stresses, where the highest stress in the arch is 17 MPa, which leaves some margin up to the 20 MPa which is the limit.



(a) Shape before optimisation. *(b) Shape after optimisation.*

Figure 58 Shape of the support attached to the arch before and after shape optimisation.

The slanted column that is connected to the arch faces a similar situation as the arch. One of the two applied shapes reaches the limit. The shape that reaches the limit is the one that controls the dimensions of the bottom section of the slanted column, i.e. the area where the boundary conditions are applied. The shape that controls the dimensions of the mid-part of the column has not reached its limit. What prevents further decrease in dimensions there is instead the applied stress constraint. The maximum von Mises stress found in the column is 19.8 MPa, which is close to the limit of 20 MPa.



(a) Shape before optimisation. *(b) Shape after optimisation.*

Figure 59 Shape of the slanted column on the opposite side of the arch before and after the shape optimisation.

The slanted column on the opposite side of the arch has, as opposed to the other column and the arch, areas that are highly stressed already before the optimisation. This causes the change in shape to be less drastic than the other two. The change in volume is even smaller, since even though the width is reduced, the height is somewhat increased causing the volume for the optimised member to be only slightly lower than the un-optimised. Since the limits on how much deformation is allowed aren't reached, it implies that the stresses have reached the limit instead. This is confirmed when looking at the stresses, where the highest stress found in the column is 19.9 MPa. It should however be noted that only small parts of the column reaches the limit, and that big parts of the column is much lower stressed, with stresses between 5 MPa and 10 MPa.

4.4 Size optimisation

The size optimisation of the bridge in this example is performed a bit differently, compared to the size optimisation performed in Section 3.2. In that section, a structure with an abundance of bars is optimised, and thus some bars are removed during the optimisation. This is not the case for the bridge, where each bar is assumed to be needed since they correspond to the topology created during the topology optimisation. For this reason, only one step is used in the size optimisation, and that is the step where the volume of the cables and columns is minimised with a stress constraint applied. The used optimisation parameters can be seen in Table 21.

Table 21 Optimisation parameters for the size optimisation of the bridge

Optimisation parameter	Value
Optimisation objective	Minimise volume
Optimisation constraint	Limit on axial stress (200 MPa > σ > -30 MPa)
Design variables (No.)	Areas of the bars (30)
Limits for design variables	$1 \cdot 10^6$ mm^2 > A > 1 mm^2

The cables and columns are modelled with bar elements, and in total the structure contains 30 bars, which can be seen in Figure 54. 13 of the bars are expected to be in compression and can thus be interpreted as columns and 17 are expected to be in tension and can be interpreted as cables. The area of each bar is one design variable in the model, i.e. each bar is optimised independently. The volume to be minimised is however the total volume of all bars. In the optimisation process, the areas of the bars all start with the area 1 mm^2. During the optimisation process the areas are allowed to increase until the stress criteria is fulfilled. The maximum limit of $1 \cdot 10^6$ mm^2 is set since a limit is needed by the program. It is however set so that it is unlikely that any area of any bar area reaches it. For the size optimisation, the same three load cases as for the shape optimisation are used.

4.4.1 Stress constraint

The stress constraint in this example is set on the axial stress of the bars. Since this can give both compressive and tensile stresses, a limit must be set on both. For this example, the parts in compression are considered to be concrete and the parts in tension are considered as steel. This gives the stress limits seen in Table 21.

4.4.2 Results from size optimisation

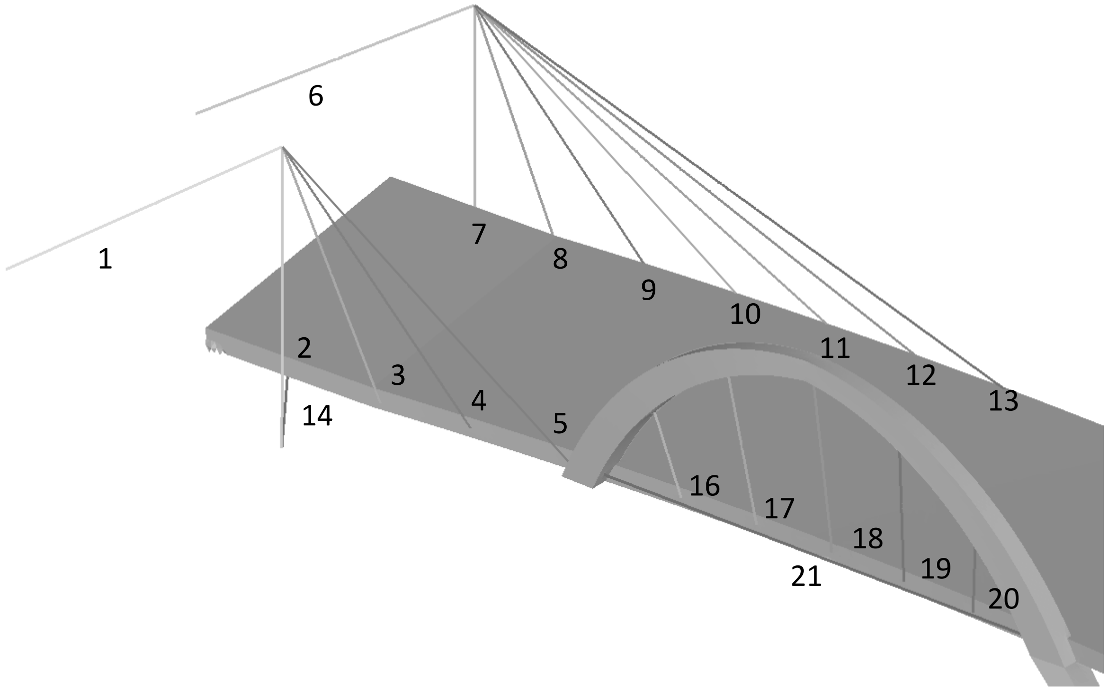
The results of the size optimisation of the cables and columns in the model of the bridge are presented below in Table 22. Which bar in the model that corresponds to which bar ID is seen in Figure 60.

The results show that while most bar areas have increased substantially from the starting value of 1 mm^2, some bars still haven't changed at all or only very little. By looking at the stresses, it can be understood why that is the case. As seen in Table 22, the bars that have areas close to 1 all have stresses at the limit or below. There is thus no reason for a bigger area.

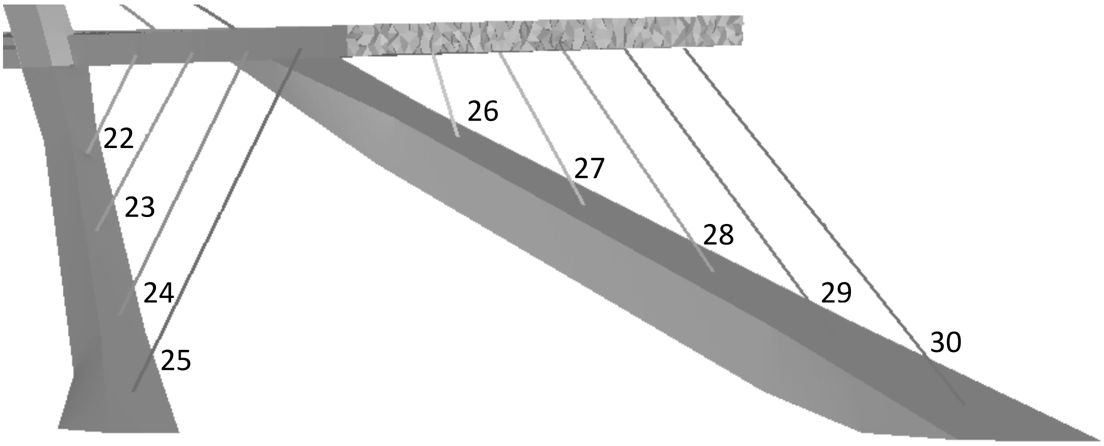
Table 22 Sizes of the bars after size optimisation

Bar ID	Area [mm^2]	Stress [MPa]	Bar ID	Area [mm^2]	Stress [MPa]	Bar ID	Area [mm^2]	Stress [MPa]
1	1366	200	11	1.076	114	21	1	9
2	12930	-30	12	7,848	28	22	2494	-18
3	5544	28	13	10860	-30	23	2494	-21
4	1.099	200	14	38840	-30	24	157	-18
5	1207	150	15	1	-30	25	1	-6
6	395.4	188	16	5.113	98	26	2494	26
7	46060	-16	17	917.6	113	27	5402	28
8	21870	40	18	1.022	200	28	6260	-30
9	1.006	199	19	529.2	187	29	2494	-24
10	1.005	196	20	875.6	118	30	2494	-9

The bars with biggest areas, bar 7 and bar 14, are both columns in compression. This is nothing strange, since the stress limit in compression is set lower than the limit for tensile stresses. The reason for this is that the parts in tension are assumed to be made of steel while the parts in compression are assumed to be made of concrete. Although many bar stresses are at the limit, there are bars with intermediate stresses and areas. Bar 3 is an example of such, where area is rather big, but the stress is low. The reason for this isn't obvious, but could be that a weaker bar there would cause an increase of stress in other bars. Noticeable is also that bar 13 is in compression, but is in the model supposed to be a cable in the cable stayed part of the bridge. One could therefore consider removing bar 13 from the model, since it is unreasonable to have a cable in compression.



(a) Bar IDs for the arch and the cable-stayed part of the bridge.



(b) Bar IDs for the other side of the bridge.

Figure 60 Bar IDs for the bars that have been optimised.

4.5 Summary

The optimisations performed in this chapter show how a structure, in this case a bridge, can be optimised in several steps. The bridge is first optimised by topology optimisation where an overall topology and shape of the structure is found. The topology created during the optimisation contains lumps of material and other irregularities, which makes it necessary to rework the model to be able to continue with the optimisations. This is done by making a new model, made out of simpler geometries, but with the overall shape as the topology from the topology optimisation.

The lumps found in the topology are e.g. replaced by a cable-stayed structure to capture the structural function of the lumps whilst making the structure simpler.

The new model is optimised further, by the use of shape optimisation. The three parts optimised are the arch, and the two slanted columns of the bridge. The results from the shape optimisation show a significant reduction of the volume of the optimised parts, while the stressed did not exceed the allowable limit.

The third step in the optimisation process of the bridge is a size optimisation of the cables and columns of the cable-stayed part and the columns connecting the deck with the slanted columns. The result from the size optimisation shows the sizes required for not exceeding the applied stress limit. It should however be noted that no regard is taken to instability phenomena such as buckling of the compressed parts; only stresses limit the sizes.

After these three optimisations are performed, the bridge ought to be a bridge with an overall design that is very effective in carrying the loads. The bridge should also be such that a high utilisation ratio is reached for most parts of the bridge. Although optimisation includes minimisation of the volume of the structure, it is both impossible and irrelevant to specify exactly how much the volume of the structure has decreased, since it started from an unreasonable big volume with the initial design domain as well as got completely redrawn prior to the shape optimisation. It is therefore sufficient to say, that even though the shape of the bridge might seem unconventional, the optimisation of it has shown that each method of optimisation has its possibilities and limitations and that a combination of many methods is needed to find an optimal solution of a more complex structure such as a bridge.

5 Discussion

From the applications in Chapter 3 it is evident that several obstacles has to be removed in order to reach a solution that is possible to manufacture and where all design aspects have been treated by using topology optimisation, especially for large three dimensional structures. From our experience topology optimisation of civil engineering structures is mostly of use in the early conceptual design phase and can with varying optimisation parameters produce a set of forms for the engineer or architect to continue with. The continuation may then include a shape or size optimisation but the amount of traditional design procedures needed should not be underestimated. An optimisation is usually limited to design aspects where a continuous and differentiable mathematical relationship exists, which excludes things like aesthetics, cost and working environment.

For a topology optimisation, there are many parameters that control how the final topology will be. Different values of for example the penalisation exponent and different techniques as values for the checkerboard-control methods all influences how the final topology will look like. This is shown in Section 3.7. Different objective and constraints also have big influences on the result. This of course raises the question, how to use the different parameters to achieve the most optimal result. Although some guidance has been presented in this report, for complex problems as the bridge in Chapter 4, it is still a difficult task to use the right combination of all available parameters to get the best solution. During the work with the bridge, it was found that several combinations of parameters gave very bad result or in some cases no results at all and even though a reasonable result was reached in the end, it is not in any way certain that there isn't a combination of parameters that would give a better solution.

Shape optimisation of structures, as performed in Section 3.3.1 and Section 4.3, is a more manual type of optimisation than e.g. topology optimisation, since the shapes of the structure is allowed to assume have to be manually defined. This means that there is more room for the person who performs the optimisation to control how the results will become. This gives good possibilities to force the optimisation towards a buildable solution, but the high level of manual input in the shape optimisation also makes it harder to find the most optimal solution. The results will always be the most optimal of the shapes defined, but it is very likely that a shape that has not been defined is better and thus the optimal solution is never found. It implies that the more shapes that are defined for the optimisation to use, the better the solution will be. However, the more shapes used, the more complex the solution. For example, in section 4.3 two shapes are used for each part that is optimised. This gives solutions with structural members that are simple and mostly straight. Adding more shapes would probably have given a more optimal solution, but with less simple geometries of the members. The number of shapes chosen is therefore a compromise between the optimality of the results and the simplicity of the shape of the optimised structure.

In this report a single material has been used for most optimisations while the ideal would be to let the optimisation chose the proper material for separate areas or volumes. To achieve a multi-material structure the only possibility in the tools we have used is to in advance define the materials for a partitioned design space. This could preferably be done after an initial single material optimisation which points out areas suitable for being a certain material. If the different materials were well separated, though still interacting as with a concrete arch carrying a bridge deck with steel tendons, this could be useful. The opposite being composite materials such as reinforced concrete where no predefinition of separate material partitions can be made both due to the reinforcing steel often being smaller than a feasible mesh size and the huge constraint this would imply on the optimisation. Specifically reinforced concrete is of high interest due to its massive use. There exists methods to optimise the steel and concrete composite by use of layered dual material elements both for linear, see Surit and Wethyavivorn (2011) and non-linear analysis, see Amir and Bogomolny (2010). However, the result must then be translated from the mere steel area needed at each position to a possible reinforcement layout.

Another problem concerns material models, only linear elastic and isotropic material models can be used in the available tools. Non-linear analysis would be useful for optimising structures for more advances behaviours, and taking into account behaviours such as cracking of concrete. It would however also greatly increase the computation time and probably need more advanced optimisation algorithms because of the additional non-linearity, but this is out of the scope of this work. The only non-linearity that is readily available for use in topology optimisation is geometrical non-linearity, vital for the optimisation for compliant mechanisms covered by Bendsoe and Sigmund (2003), but also needed for very slender structures and soft materials where large deformations may take place.

The mesh sizes of the models in this report have been chosen so that the mesh is fine enough to give acceptable results, but still give a computation time that is reasonable. Although a finer mesh has proven to give better, and more detailed, results for the topology optimisations, the fineness is ultimately limited by the available computer resources. For Matlab-based programs, the computer-cluster at Chalmers can be, and has been, used for some of the examples in this report. Use of the cluster is absolutely necessary for performing optimisation on bigger 3D models in Matlab, something that has proven to be very memory demanding. Approximately 54 GB of memory was for example required for the S-bridge problem in Section 3.9. A 2D problem with the same number of degrees of freedom only requires approximately 4 GB of memory. Although the cluster has proven to be very useful for the optimisation tasks performed by Matlab, it has not been possible to use the cluster for the problems optimised with HyperWorks. This is due to that the cluster doesn't support the file types used by the solver Optistruct. This makes the model sizes, and the fineness's of the meshes used in HyperWorks more limited.

For the shape optimisation the mesh size has proven to be of less importance. It has on the contrary been so that a coarser mesh makes it easier to define the shapes used in the optimisation. This is because a large element can tolerate larger node movements without getting too distorted. The distortion of elements is the biggest limitation when it comes to shape optimisation. The ideal would be to have the possibility to freely alter the shapes as seen fit, but in practice this is not possible. Too much morphing of the mesh creates bad quality elements. This is especially true for members with irregularities in the mesh, such as areas where different structural members connect to each other. Ironically, those are the places where shape optimisation can be of much use to reduce stress concentrations. If the program had the ability to refine the mesh after each change in shape, i.e. making the mesh independent from the geometrical parameterisation, the shape optimisation would most probably have been more useful.

6 Conclusions

6.1 Conclusions and recommendations

Structural form optimisation has proven to have both possibilities and limitations, as is shown in this report. One of the major limitations is that it as of today, in the available software, only is possible to use simple linear elastic material models. The optimisations are therefore primarily suited for applications and materials where such a simple material model is adequate to describe the real behaviour. This goes for all types of optimisation treated in this report, but especially for topology optimisation. That is because topology optimisations are the least constrained optimisation type and requires the most general algorithms. For more advanced materials such as composites, one being concrete, with pronounced anisotropic and non-linear behaviour, topology optimisation can today only give a hint of how the optimal topology would look like.

It is possible to sequentially apply the optimisation types in order to reach an optimal design, as is done in the report. This can however not be done in a simple way, much because of the result from the topology optimisation is not well-defined enough for further shape or size optimisations. This means that a new model has to be drawn as an interpretation of the result from the topology optimisation, which of course causes some loss of optimality of the solution.

If using the optimisation types treated in this report, one must be aware of each type's limitations and that each type of optimisation has its own purposes. Since only linear elastic material models can be used, which do not very accurately describe the behaviour of concrete, the accuracy of the solutions cannot be assured, especially not for topology optimisation. The recommendation is thus to only use topology optimisation in earl design stages. It could for example be an excellent tool for conceptual design, used to quickly find a promising overall shape of the structure. This is especially true for larger structures, where demands on computer hardware limit the mesh size effectively prohibiting the formation of thin members and therefor also the creation of detailed topologies. By varying the optimisation parameters such as the penalisation exponent and the minimum member dimension it is possible to obtain very different results. This can be very useful in a conceptual design stage where the results can give inspiration for a set of different promising solutions.

Size optimisation can be used, as shown in the report, as a simple tool for sizing of axially loaded members. Not to forget is however, that compressed members often becomes unreasonably small since the instability phenomena of compressed members isn't taken into account in the linear analysis. Shape optimisation is, as topology optimisation, limited by the material models. Shape optimisation can however be used e.g. for reducing stress concentrations in solid or welded steel parts where the stresses in the model well corresponds to the reality.

6.2 Future research

There are several areas only treated briefly in this report that would benefit from further research. One example is how to us the optimisation techniques to optimise reinforced concrete. Some research has been done, as mentioned earlier in this chapter, but the methods are still far from being incorporated in available software.

Another area of interest that could be seen as a continuation of this report, is the research of the gain of optimisation, i.e. how much can be won by optimising structures. It would for example be interesting to continue with the bridge optimised in Chapter 4, performing a proper dimensioning of the bridge and then compare the bridge with real world examples with similar conditions.

7 References

Amir O. and Bogomolny M. 2010. Topology Optimization for Conceptual Design of Reinforced Concrete Structures. *9th World Congress on Structural And Multidisciplinary Optimization.* Shizuoka, Japan.

Andreassen E., Clausen A., Schevenels M., Lazarov B. S. and Sigmund O. (2010): Efficient topolog optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization,* Vol. 43**,** 1-16.

Beckers M. (1999): Topology optimization using a dual method with discrete variables. *Structural Optimization,* Vol. 17**,** 14-24.

Bendsoe M. P. and Sigmund O. (2003): *Topology Optimization: Theory, Methods and Applications,* Springer-Verlag. Berlin, Germany,

Bruyneel M. and Duysinx P. (2005): Note on topology optimization of continuum structures including self-weight. *Structural and Multidisciplinary Optimization,* Vol. 29**,** 245-256.

Christensen P. W. and Klarbring A. 2008. An Introduction to Structural Optimization. Dordrecht: Springer Netherlands.

Chu D. N., Xie Y. M., Hira A. and Steven G. P. (1996): Evolutionary structural optimization for problems with stiffness constraints. *Finite Elements in Analysis and Design,* Vol. 21**,** 239-251.

Duysinx P., Van Miegroet L., Lemaire E., Brüls O. and Bruyneel M. (2008): Topology and generalized shape optimization: Why stress constraints are so important? *International Journal for Simulation and Multidisciplinary Design Optimization,* Vol. 4**,** 253–258.

Fall D. 2011. Reinforcement in Tailor-Made Concrete Structures. *Department of Civil and Environmental Engineering.* Göteborg: Chalmers.

Haftka R. T. and Grandhi R. V. (1986): Structural Shape Optimization - a Survey. *Computer Methods in Applied Mechanics and Engineering,* Vol. 57**,** 91-106.

Hager K. and Balling R. (1988): New Approach for Discrete Structural Optimization. *Journal of structural engineering,* Vol. 114**,** 1120-1134.

Huang X. D. and Xie Y. M. (2010): A further review of ESO type methods for topology optimization. *Structural and Multidisciplinary Optimization,* Vol. 41**,** 671-683.

Liu X., Yi W. J., Li Q. S. and Shen P. S. (2008): Genetic evolutionary structural optimization. *Journal of constructional steel research,* Vol. 64**,** 305-311.

Michell A. G. M. (1904): The limits of economy of material in frame-structures. *Philosophical Magazine,* Vol. 8**,** 589-597.

Pedersen P. 2003. Optimal Designs; Structures and Materials - Problems and Tools. Lyngby, Denmark.

Querin O. M., Steven G. P. and Xie Y. M. (1998): Evolutionary structural optimisation (ESO) using a bidirectional algorithm. *Engineering Computations,* Vol. 15**,** 1031-1048.

Rozvany G. I. N. (2009): A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization,* Vol. 37, 217-237.

Schmit L. A. 1960. Structural design by systematic synthesis. *Second ASCE Conference on Electronic Computation.* Pittsburgh, USA.

Sigmund O. (2001): A 99 line topoology optimization code written in Matlab. *Structural and Multidisciplinary Optimization,* Vol. 21, 120-127.

Stolpe M. and Svanberg K. (2001): On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization,* Vol. 21, 128-139.

Surit S. and Wethyavivorn B. (2011): Topology optimization of reinforced concrete beams by a spread-over reinforcement model with fixed grid mesh. *Songklanakarin Journal of Science and Technology,* Vol. 33, 95-100.

Svanberg K. (1987): The method of moving asymptotes - A new method for structural optimization. *International Journal for Numerical Methods in Engineering,* Vol. 24, 359-373.

Vanderplaats G. N. (1993): Thirty years of modern structural optimization. *Advances in engineering software,* Vol. 16, 81-88.

Williams N., Stehling H., Scheurer F., Oesterle S., Kohler M. and Gramazio F. (2011): A Case Study of a Collaborative Digital Workflow in the Design and Production of Formwork for 'Non-Standard' Concrete Structures. *International journal of architectural computing,* Vol. 9, 223-240.

Zhou M., Shyy Y. K. and Thomas H. L. (2001): Checkerboard and minimum member size control in topology optimization. *Structural and Multidisciplinary Optimization,* Vol. 21, 152-158.

Appendix A Matlab Code

Appendix A-1 Toptimiser 2D

```
%________             __________           _____
%___  __/_______________  /___(_)______.______(_)__________________
%__  /   _  __ \__  __ \  __/_  /__  __ `__ \_  /__  ___/  _ \_  ___/
%_  /    / /_/ /_  /_/ / /_ _  / _  / / / / /  / _(__  )/  __/  /
%/_/     \____/_  .___/\__/ /_/  /_/ /_/ /_//_/  /____/ \___//_/      [2D]
%              /_/
%2D topology optimiser for minimum compliance with volume constraint,
%Based on top (Sigmund, 2002) and top88 (Andreassen et al, 2010).
%Required variables in the problem script, saved to 'ProbScript'_in.mat:
% elemSize          Side length for the square elements.
% nx, ny            Number of elements in x and y direction.
% fixeddofs         Boundary conditions as a row vector of fixed degrees of
%                   freedom.
% F                 Force matrix with one column per load case.
% volfrac           Available material as ratio of design domain area.
% pen               Penalisation vector, for length(pen)=1 a constant
%                   penalisation is used. For length(pen)=3, pen(1) defines
%                   the starting value, pen(2) the increment factor each
%                   time  a converge critera is met and pen(3) the final
%                   penalisation.
% rmin              Filter radius, minimum radius for any structural
%                   member.
%Optional variables in the problem script:
% ft                Filter type. 1: Sensitivity filter, 2: Density filter,
%                   3: Density filter + Heaviside projextion filter.
% x                 Initial density distribution, (nx x ny).
% elWhite, elBlack  Matries (ndof x ndof) prescribing densities to white or
%                   black. 1 means prescribed density, 0 means variable.
% symm              Symmetry matrix used to display the entire structure,
%                   each  matrix element will display a flipped copy.
%                   1: flip up/down, 2: flip left/right, 3: no change,
%                   4: flip up/down and left/right.
%                   Examples:
%                     symm=[3 2] for single symmetry line in y-dir.
%                     symm=[1 4; 3 2] for double symmetry and only the
%                     lower left quarter is analysed.
%                   Symmetry boundary conditions still needs to be
%                   specified in fixeddofs.
% E, nu             Young's modulus and Poisson's ratio, def.: E=1, nu=0.3.
% rho               Material density, only used if g~=0.
% g                 Gravitaional acceleration. If g~=0 MMA will be used.
% opt               Design variable update scheme, 'oc' or 'mma'.
% move              Move limit for the density at each iteration, def.:0.2.
% maxit             Maximum number of iterations, def.: 1000.
% tol               Convergence tolerance, def.:1e-2.
% stress            Specify one load case for stress calculation.
% video             A video over the iteration process is created if this
%                   variable exist.
function []=top2D(ProbScript)
close all
load([ProbScript '_in'])
ndof=2*(ny+1)*(nx+1);
fprintf(['\nDegrees of freedom: %1.0i\nApproximate memory usage: %1.0f' ...
  'MB\n'],ndof,3.5e-3*ndof)
cont=input('Continue? (y/n)  ','s');
if cont == 'n', error('Aborted by user.'), end
%% Material Properties
if exist('E0','var') == 0, E0=1; end
Emin = E0/1e9;
if exist('nu','var') == 0, nu=0.3; end
%% Prepare finite element analysis
A11=[12  3 -6 -3;  3 12  3  0; -6  3 12 -3; -3  0 -3 12];
A12=[-6 -3  0  3; -3 -6 -3 -6;  0 -3 -6  3;  3 -6  3 -6];
B11=[-4  3 -2  9;  3 -4 -9  4; -2 -9 -4 -3;  9  4 -3 -4];
B12=[ 2 -3  4 -9; -3  2  9 -2;  4  9  2  3; -9 -2  3  2];
KE=1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
nodenrs=reshape(1:(1+nx)*(1+ny),1+ny,1+nx);
edofVec=reshape(2*nodenrs(1:end-1,1:end-1)+1,nx*ny,1);
edofMat=repmat(edofVec,1,8)+repmat([0 1 2*ny+[2 3 0 1] -2 -1],nx*ny,1);
alldofs = 1:ndof;
freedofs = setdiff(alldofs,fixeddofs);
```

```
%% Prepare filter
[dy,dx]=meshgrid(-ceil(rmin)+1:ceil(rmin)-1, -ceil(rmin)+1:ceil(rmin)-1);
h=max(0,rmin-sqrt(dx.^2+dy.^2));
Hs=conv2(ones(ny,nx),h,'same');
if exist('ft','var') == 0, ft=1; end
%% Display abort button
scrsz = get(0,'ScreenSize'); global abortfig
abortfig=figure('OuterPosition',[10 scrsz(4)-80 100 70],'MenuBar','none');
uicontrol('Style','Pushbutton','String','Abort and save','Units',...
  'Normalized','Position',[0.1 0.1 0.75 0.9],'Callback',@finnish);
%% Set default for missing parameters
if exist('elWhite','var') == 0, elWhite=zeros(ny,nx); end
if exist('elBlack','var') == 0, elBlack=zeros(ny,nx); end
if exist('symm','var') == 0, symm=3; end
if exist('g','var') == 0, g=0; end
if g~= 0 && exist('rho','var') == 0, rho=1; end
if exist('opt','var') == 0
  if g == 0, opt='oc'; end
  if g ~= 0, opt='mma'; end
end
if exist('maxit','var') == 0, maxit=1000; end
if exist('move','var') == 0, move=0.2; end
if exist('tol','var') == 0, tol=1e-2; end
if exist('stress','var') == 0, stress=0; end
%% Initialise iteration
volfracGlobal=volfrac*(nx*ny-sum(sum(elWhite==1)))/(nx*ny);
if exist('x','var') == 0
  x=volfracGlobal*ones(ny,nx); x(elBlack==1)=1; x(elWhite==1)=0;
end
if ft == 0 || ft == 1 || ft == 2
  xPhys=x;
  fixdist=rmin;
elseif ft == 3
  beta=1;
  xTilde=x;
  xPhys=1-exp(-beta*xTilde)+xTilde*exp(-beta);
  xPhys=volfracGlobal*(xPhys/mean(xPhys(:)));
  fixdist=2*rmin;
end
loop=0; loopbeta=0;
global abort, abort=0;
if exist('video','var')
  vidObj = VideoWriter([ProbScript '.avi'],'Uncompressed AVI');
  open(vidObj);
end
if length(pen)==1
  pen=[pen 1.0 pen];
end
p=pen(1);
%% Start iterations
while abort == 0
  tic
  loop=loop+1; loopbeta=loopbeta+1;
  %% Fix elements with zero density
  [dist,~]=bwdist(x,'chessboard');
  elFix=find(dist>fixdist);
  elFree=(1:nx*ny)'; elFree(elFix)=[];
  Nfixeddofs=union(fixeddofs,reshape(edofMat(elFix,:),1,8*length(elFix)));
  freedofs=setdiff(alldofs,Nfixeddofs);
  %% FE-analysis
  iK=reshape(kron(edofMat(elFree,:),ones(8,1))',64*length(elFree),1);
  jK=reshape(kron(edofMat(elFree,:),ones(1,8))',64*length(elFree),1);
  if g ~= 0
    xVarP=zeros(ny,nx);
    xVarP(xPhys>0.25)=xPhys(xPhys>0.25).^p;
    xVarP(xPhys<=0.25)=xPhys(xPhys<=0.25).*(0.25^(p-1));
    Fsw=zeros(ndof,1);
    for i=1:nx*ny
      Fsw(edofMat(i,2:2:end))=Fsw(edofMat(i,2:2:end))+...
        xPhys(i)*elSize^2*rho*g/4;
    end
    Fsw=repmat(Fsw,1,size(F,2));
    Ftot=F+Fsw;
  else
    xVarP=xPhys.^p; Ftot=F;
  end
  sK=reshape(KE(:)*(Emin+xVarP(elFree')*(E0-Emin)),64*length(elFree),1);
```

```
K=sparse(iK,jK,sK); K=(K+K')/2;
U=zeros(size(F));
U(freedofs,:)=K(freedofs,freedofs)\Ftot(freedofs,:);
%% Objective function and sensitivity analysis
c(loop)=0; dc=0;
for i=1:size(F,2)
  Ui=U(:,i);
  ce=reshape(sum((Ui(edofMat)*KE).*Ui(edofMat),2),ny,nx);
  c(loop)=c(loop)+sum(sum((Emin+xPhys.^p*(E0-Emin)).*ce));
  if g ~= 0
    cf=reshape(2*Ui(edofMat)*repmat([0; (elSize^2)/4*rho*g],4,1),ny,nx);
  else
    cf=0;
  end
  dc=dc+xPhys.*cf-p*(E0-Emin)*xPhys.^(p-1).*ce;
end
dv=ones(ny,nx)*(elSize^2);
%% Filtering/modification of sensitivities
if ft == 1
  dc=conv2(dc.*xPhys,h,'same')./Hs./max(1e-3,xPhys);
elseif ft == 2
  dc=conv2(dc./Hs,h,'same');
  dv=conv2(dv./Hs,h,'same');
elseif ft == 3
  dx=beta*exp(-beta*xTilde)+exp(-beta);
  dc=conv2(dc.*dx./Hs,h,'same');
  dv=conv2(dv.*dx./Hs,h,'same');
end
if strcmpi(opt,'OC')
  %% Optimality criteria update of design variables and physical densities
  cscale=1;
  l1=0; l2=1e9;
  while (l2-l1)/(l1+l2) > 1e-3
    lmid=0.5*(l2+l1);
    xnew=max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
    if ft == 1
      xPhys=xnew;
    elseif ft == 2
      xPhys=conv2(xnew,h,'same')./Hs;
    elseif ft == 3
      xTilde=conv2(xnew,h,'same')./Hs;
      xPhys=1-exp(-beta*xTilde)+xTilde*exp(-beta);
    end
    xPhys(elWhite==1)=0; xPhys(elBlack==1)=1;
    if sum(xPhys(:)) > volfracGlobal*nx*ny, l1=lmid; else l2=lmid; end
  end
elseif strcmpi(opt,'MMA')
  %% MMA Update of design variables and physical densities
  m=1; n=nx*ny; nfree=nx*ny-sum(elBlack(:)+elWhite(:));
  if loop > 2, xold2=xold1; else xold2=[]; end
  if loop > 1, xold1=xval; else xold1=[]; end
  xval=reshape(x,n,1); ID=find(xval+1);
  ID(elBlack==1)=[]; ID(elWhite==1)=[];
  xval(elBlack==1)=[]; xval(elWhite==1)=[];
  xmin=max(0,xval-move); xmax=min(1,xval+move);
  if loop == 1, cscale=100/c(loop); end
  f0val=c(loop)*cscale; df0dx=reshape(dc,n,1)*cscale;
  df0dx(elBlack==1)=[]; df0dx(elWhite==1)=[];
  fval=mean(xPhys(:))/volfrac-1;
  dfdx=ones(1,nfree)/n;
  if loop == 1, low=[]; upp=[]; end
  [xmma,~,~,~,~,~,~,~,~,low,upp]=mmasub(m,nfree,loop,xval,xmin,xmax,...
    xold1,xold2,f0val,df0dx,fval,dfdx,low,upp,1,zeros(m,1),1000*...
    ones(m,1),zeros(m,1));
  xnew=zeros(nx*ny,1); xnew(ID)=xmma; xnew=reshape(xnew,ny,nx);
  xnew(elBlack==1)=1; xnew(elWhite==1)=0;
  if ft == 0 || ft == 1
    xPhys=xnew;
  elseif ft == 2
    xPhys=conv2(xnew,h,'same')./Hs;
  elseif ft == 3
    xTilde=conv2(xnew,h,'same')./Hs;
    xPhys=1-exp(-beta*xTilde)+xTilde*exp(-beta);
  end
end
[change(loop),I]=max(abs(xnew(:)-x(:)));
change(loop)=change(loop)*sign(xnew(I)-x(I));
```

```
  x=xnew;
  %% Print results
  t(loop)=toc;
  fprintf(['It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f fdo:%6.0f t:%4i '...
    'p:%2.2f\n'],loop,c(loop),mean(xPhys(:)),change(loop),...
    length(freedofs),int32(t(loop)*1000),p);
  %% Plot densities
  figure(2), subplot(2,1,1), colormap(gray(256)); caxis([0 1])
  for i=1:size(symm,1)
    for j=1:size(symm,2)
      if symm(i,j)==4
        xSymm((i-1)*ny+1:i*ny,(j-1)*nx+1:j*nx)=rot90(xPhys,2);
      else
        xSymm((i-1)*ny+1:i*ny,(j-1)*nx+1:j*nx)=flipdim(xPhys,symm(i,j));
      end
      image(256*(1-xSymm));
    end
  end
  axis equal; axis off; axis([0 size(symm,2)*nx 0 size(symm,1)*ny]+0.5)
  if exist('video','var')
    [X,map]=gray2ind(1-xSymm,256);
    frame=im2frame(X,map);
    writeVideo(vidObj,frame);
  end
  %% Increase penalisation and beta-factor, plot and check for convergence
  if loop > 2
    if length(pen)==3 && abs(change(loop)) < 5*tol
      p=min(pen(2)*p,pen(3));
    end
    figure(2), subplot(2,1,2), cla
    ax=plotyy(1:loop,c,1:loop,abs(change),'semilogy'); hold on
    line([0 loop],[tol tol],'Parent',ax(2),'color','k','linestyle','--')
    line([0 loop],[5*tol 5*tol],'Parent',ax(2),'color','k','linestyle',':')
    hold off
    drawnow
  end
  if ft == 3 && beta < 512 && p == pen(3) && (loopbeta >= 10 || ...
      abs(change(loop)) <= 5*tol)
    beta=2*beta;
    fprintf('beta now is %3.0f\n',beta)
    loopbeta=0; change(loop)=1;
  end
  if (abs(change(loop)) < tol && p==pen(3)) || loop == maxit, abort=1; end
end
%% Compliance without penalisation
sK=reshape(KE(:)*(Emin+xPhys(elFree')*(E0-Emin)),64*length(elFree),1);
K=sparse(iK,jK,sK); K=(K+K')/2;
U=zeros(size(F));
U(freedofs,:)=K(freedofs,freedofs)\Ftot(freedofs,:);
cNoPen=0;
for i = 1:size(F,2)
  Ui=U(:,i);
  ce=reshape(sum((Ui(edofMat)*KE).*Ui(edofMat),2),ny,nx);
  cNoPen=cNoPen+sum(sum((Emin+xPhys*(E0-Emin)).*ce));
end
fprintf('Compliance without penalisation is: %1.3f\n',cNoPen)
%% Average density
meanx=mean(xPhys(xPhys(elBlack~=1)>0));
fprintf('Avarage density: %1.3f\n',meanx)
%% Stress calculation
if stress ~= 0
  ed=reshape(U(edofMat(:,:)',stress),8,nx*ny)';
  D=E0/(1-nu^2)*[1  nu 0;
    nu 1  0;
    0  0  (1-nu)/2];
  B=elSize/2*[-1  0  1  0  1  0 -1  0;
               0 -1  0 -1  0  1  0  1;
              -1 -1 -1  1  1  1  1 -1];
  ee=B*ed'; ee(:,xPhys<0.1)=0;
  ss=D*ee;
  %Principal stresses
  pss1=sum(ss(1:2,:))/2+sqrt((diff(ss(1:2,:))/2).^2+(ss(3,:)).^2);
  pss2=sum(ss(1:2,:))/2-sqrt((diff(ss(1:2,:))/2).^2+(ss(3,:)).^2);
  %von Mises stresses
  mis=sqrt((pss1-pss2).^2+pss2.^2+pss1.^2);
  %Signed von Mises stresses
  misS=sqrt((pss1-pss2).^2+pss2.^2+pss1.^2).*-sign(max(abs(pss1),...
    abs(pss2)).*sign(pss1+pss2));
  misM=mean(mis(mis~=0));
  %Plot normalised von Mises stresses
  misP=mis;
  misP(mis>2*misM)=2*misM;
  misP=misP/max(misP(:));
  figure(3), imagesc(reshape(misP,ny,nx))
  axis equal; axis off
  cm=colormap(jet(128)); colormap([1 1 1; cm(1:127,:)]), colorbar
  title('Normalised von Mises stresses')
  %Plot normalised signed von Mises stresses
  misSP=misS;
  misSP(misS<-2*misM)=-2*misM;
  misSP(misS>2*misM)=2*misM;
  misSP=misSP/max(misSP(:));
  misSP(misSP==0)=-1-1/256;
  figure(4), imagesc(reshape(misSP,ny,nx))
  axis equal; axis off;
  cm=colormap(jet(256)); colormap([1 1 1; cm(1:255,:)]), colorbar
  title('Normalised signed von Mises stresses')
end
%% Finnish plots
figure(2)
legend('Compliance','Max. change in density','Location','best')
%% Save results
save([ProbScript '_out'])
if exist('video','var')
  close(vidObj);
end
[X,map]=gray2ind(1-xSymm,256);
imwrite(X,map,[ProbScript '.png'],'png')
@finnish;
function finnish(~,~,~)
global abort abortfig, abort=1;
close(abortfig)
```

Appendix A-2 Script for the bridge problem

```
%--------------------------------------------------
%Script defining the bridge problem for Toptimiser2D.
%--------------------------------------------------
%Size of design domain
lx=1; ly=1;
elSize=0.008;
nx=round(lx/elSize);
ny=round(ly/elSize);
ndof=2*(ny+1)*(nx+1);
%Prescribed void or solid areas
elWhite=zeros(ny,nx);
elBlack=zeros(ny,nx); elBlack(1:max(1,round(0.01*ny)),:)=1;
%Boundary conditions
fixeddofs=[1:2:2*(ny+1) ndof-1 ndof];
symm=[2 3];
%Loads
top_nodes=2:(ny+1)*2:ndof;
F=sparse(top_nodes,ones(size(top_nodes)),[-1/2; ...
  -1*ones(length(top_nodes)-2,1); -1/2],ndof,1);
%Optimisation options
volfrac=0.15;
pen=[1 1.1 3];
rmin=1.5; %1/elemsize;
%Filter option
ft=1;
%Run optimisation
save('bridge_in')
Toptimiser2D('bridge')
```

Additional lines for self-weight loaded structure:

```
%Loads
g=9.82;
%Material
rho=2700;
```

Appendix A-3 Script for the cantilever problem

```
%------------------------------------------------------
%Script defining the cantilever problem for Toptimiser2D.
%------------------------------------------------------
% Size of design domain
l=500; h=500;
elSize=1;
nx=l/elSize;
ny=h/elSize;
ndof=2*(ny+1)*(nx+1);
% Boundary conditions
fixeddofs=1:2*(ny+1);
%Loads
F=sparse(ndof,1,-1,ndof,1); %Bottom
%Optimisation options
volfrac=0.15;
pen=[1 1.1 5];
rmin=5;
%Run optimisation
save('cantilever_in')
Toptimiser2D('cantilever')
```

Appendix A-4 Toptimiser 3D

```
%________            __________          _____
%___  __/______________  /___(_)______ _____(_)__________________
%__  /  _  __ \__  __ \  __/_  /__  __ `__ \_  /__  ___/  _ \_  ___/
%_  /   / /_/ /_  /_/ / /_ _  / _  / / / / /  / _(__  )/  __/  /
%/_/    \____/_  .___/\__/ /_/  /_/ /_/ /_//_/  /____/ \___//_/      [3D]
%             /_/
%3D topology optimiser for minimum compliance with volume constraint,
%Based on top (Sigmund, 2002) and top88 (Andreassen et al, 2010).
%Required variables in the problem script, saved to 'ProbScript'_in.mat:
% elSize            Side length for the qubic elements.
% nx, ny, nz        Number of elements in x-, y- and z-direction.
% fixeddofs         Boundary conditions as a row vector of fixed degrees of
%                   freedom.
% F                 Force matrix with one column per load case.
% volfrac           Available material as ratio of design domain volume.
% pen               Penalisation vector, for length(pen)=1 a constant
%                   penalisation is used. For length(pen)=3, pen(1) defines
%                   the starting value, pen(2) the increment factor each
%                   time  a converge critera is met and pen(3) the final
%                   enalisation.
% rmin              Filter radius, minimum radius for any structural
%                   member.
%Optional variables in the problem script:
% x                 Initial density distribution, (nx x ny x nz)
% elWhite, elBlack  Matries (nx x ny x nz) prescribing densities to white
%                   or black. 1 means prescribed density, 0 means variable.
% E, nu             Young's modulus and Poisson's ratio, def.: E=1, nu=0.3.
% move              Move limit for the density at each iteration, def.:0.2.
% maxit             Maximum number of iterations, def.: 1000.
% tol               Convergence tolerance, def.:1e-2.
% video             A video over the iteration process is created if this
%                   variable exist.
function []=Toptimiser3D(ProbScript)
load([ProbScript '_in'])
ndof=3*(ny+1)*(nx+1)*(nz+1);
fprintf(['\nDegrees of freedom: %1.0i\nApproximate memory usage: %1.0f' ...
  'MB\n'],ndof,0.025*ndof)
cont=input('Continue? (y/n)  ','s');
if cont == 'n', error('Aborted by user!'), end
%% Material Properties
if exist('E0','var') == 0, E0=1; end
Emin=E0/1e9;
if exist('nu','var') == 0, nu=0.3; end
%% Prepare finite element analysis
Ke=soli8e([0 1 1 0 0 1 1 0],[0 0 1 1 0 0 1 1],[0 0 0 0 1 1 1 1],2,...
  hooke(4,1,nu));
```

```
nodenrs=reshape(1:(1+nx)*(1+ny)*(1+nz),1+nz,1+ny,1+nx);
edofVec=reshape(3*nodenrs(1:end-1,1:end-1,1:end-1)-2,nx*ny*nz,1);
edofMat=repmat(edofVec,1,24)+repmat([0 1 2 3 4 5 3*(nz+1)+[3 4 5 0 1 2]...
  ((nz+1)*(ny+1)*3)+[0 1 2 3 4 5 3*(nz+1)+[3 4 5 0 1 2]]],nx*ny*nz,1);
alldofs=1:ndof;
%% Prepare filter
[dz,dy,dx]=meshgrid(-ceil(rmin)+1:ceil(rmin)-1,-ceil(rmin)+1:ceil(rmin)...
  -1,-ceil(rmin)+1:ceil(rmin)-1);
h=max(0,rmin-sqrt(dz.^2+dy.^2+dx.^2));
Hs=convn(ones(nz,ny,nx),h,'same');
%% Display abort button
scrsz = get(0,'ScreenSize'); global abortfig
abortfig=figure('OuterPosition',[10 scrsz(4)-80 100 70],'MenuBar','none');
uicontrol('Style','Pushbutton','String','Abort and save','Units',...
  'Normalized','Position',[0.1 0.1 0.75 0.9],'Callback',@finnish);
%% Initialise iteration
if exist('elWhite','var') == 0, elWhite=[]; end
if exist('elBlack','var') == 0, elBlack=[]; end
volfracGlobal=volfrac*(nx*ny*nz-sum(elWhite(:)==1)+sum(elBlack(:)==1))/...
  (nx*ny*nz);
if exist('x','var') == 0
    x=volfracGlobal*ones(nz,ny,nx); x(elBlack==1)=1; x(elWhite==1)=0;
end
loop=0; change=1;
if exist('maxit','var') == 0, maxit=1000; end
if exist('move','var') == 0, move=0.2; end
if exist('tol','var') == 0, tol=1e-2; end
global abort, abort=0;
if exist('video','var')
  vidObj=VideoWriter([ProbScript '.avi'],'Uncompressed AVI');
  open(vidObj);
end
if length(pen) == 1
  pen=[pen 1.0 pen];
end
p=pen(1);
[pz,py,px]=ndgrid(0.5:1:nz-0.5,0.5:1:ny-0.5,0.5:1:nx-0.5);%Plot coordinates
%% Start Iteration
while abort == 0
  tic
  loop=loop+1;
  %% Fix elements with zero density
  [dist,~]=bwdist(x,'chessboard');
  elFix=find(dist>rmin);
  elFree=(1:nx*ny*nz)'; elFree(elFix)=[];
  Nfixeddofs=union(fixeddofs,reshape(edofMat(elFix,:),1,24*length(elFix)));
  freedofs=setdiff(alldofs,Nfixeddofs);
  %% FE-analysis
  iK=reshape(kron(edofMat(elFree,:),ones(24,1))',576*length(elFree),1);
  jK=reshape(kron(edofMat(elFree,:),ones(1,24))',576*length(elFree),1);
  sK=reshape(Ke(:)*(Emin+x(elFree)'.^p*(E0-Emin)),576*length(elFree),1);
  K=sparse(iK,jK,sK); K=(K+K')/2;
  U=zeros(size(F));
  U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
  %% Objective function and sensitivity analysis
  c(loop)=0; dc=0;
  for i = 1:size(F,2)
    Ui=U(:,i);
    ce=reshape(sum((Ui(edofMat)*Ke).*Ui(edofMat),2),nz,ny,nx);
    c(loop)=c(loop)+sum(sum(sum((Emin+x.^p*(E0-Emin)).*ce)));
    dc=dc-p*(E0-Emin)*x.^(p-1).*ce;
  end
  dv=ones(nz,ny,nx);
  %% filtering/modification of sensitivities
  dc=convn(dc.*x,h,'same')./Hs./max(1e-3,x);
  %% Optimality criteria update of design variables
  l1=0; l2=1e9;
  while (l2-l1)/(l1+l2) > 1e-3
    lmid=0.5*(l2+l1);
    xnew=max(0,max(x-move,min(1,min(x+move,x.*sqrt(-min(0,dc)./dv/lmid)))));
    xnew(elWhite==1)=0; xnew(elBlack==1)=1;
    if sum(xnew(:)) > volfracGlobal*nx*ny*nz, l1=lmid; else l2=lmid; end
  end
  [change(loop),I]=max(abs(xnew(:)-x(:)));
  change(loop)=change(loop)*sign(xnew(I)-x(I));
  x=xnew;
  %% Print results
  fprintf(['It.:%4i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f fdo:%7i t:%3i '...
    'p:%5.2f\n'],loop,c(loop),mean(x(:)),change(loop),length(freedofs),...
    int16(toc),p);
  %% Increase penalisation
  if abs(change(loop)) < 5*tol
    p=min(pen(2)*p,pen(3));
  end
  %% Plot density iso-surface
  figure(3), cla
  p1=patch(isosurface(px,py,pz,x,0.5)); hold on
  p2=patch(isocaps(px,py,pz,x,0.5));
  set(p1,'FaceColor',[0.2 0.5 0.2],'EdgeColor','none');
  set(p2,'FaceColor',[0.2 0.5 0.2],'EdgeColor','none');
  light('Position',[0 ny -nz]);
  axis equal; axis([0 nx 0 ny 0 nz]); view(3); box on;
  set(gca,'ZDir','reverse','YDir','reverse')
  xlabel('x'), ylabel('y'), zlabel('z')
  set(gcf,'color','white');
  drawnow
  if exist('video','var') == 1, writeVideo(vidObj,getframe); end
  %% Plot and check for convergence
  if loop > 2
    figure(2), clf
    AX=plotyy(1:loop,c,1:loop,abs(change),'semilogy'); hold on
    line([0 loop],[tol tol],'Parent',AX(2),'color','k','linestyle','--')
    line([0 loop],[5*tol 5*tol],'Parent',AX(2),'color','k','linestyle',':')
  end
  if (abs(change(loop)) < tol || loop == maxit) && p == pen(3)
    abort=1;
  end
end
%% Compliance without penalisation
sK=reshape(Ke(:)*(Emin+x(elFree)'*(E0-Emin)),576*length(elFree),1);
K=sparse(iK,jK,sK); K=(K+K')/2; U=zeros(size(F));
U(freedofs,:)=K(freedofs,freedofs)\F(freedofs,:);
cNoPen=0;
for i=1:size(F,2)
  Ui=U(:,i);
  ce=reshape(sum((Ui(edofMat)*Ke).*Ui(edofMat),2),nz,ny,nx);
  cNoPen=cNoPen+sum(sum(sum((Emin+x*(E0-Emin)).*ce)));
end
fprintf('Compliance without penalisation is: %11.4f\n',cNoPen)
%% Amount of grey
Ngrey=length(find(0 < x & x < 1));
fprintf('Amount of grey elements: %1.2f%%\n',Ngrey/nx/ny/nz*100)
%% Better density plot
figure(3), cla
p1=patch(isosurface(px,py,pz,x,0.8)); hold on
p2=patch(isocaps(px,py,pz,x,0.8));
set(p1,'FaceColor',[0.2 0.5 0.2],'EdgeColor','none');
set(p2,'FaceColor',[0.2 0.5 0.2],'EdgeColor','none');
fixed=zeros(nx*ny*nz,1); fixed(elFix)=1; fixed=reshape(fixed,nz,ny,nx);
patch(isosurface(px,py,pz,x,0.5),'FaceAlpha',0.1,'EdgeAlpha',0.1);
patch(isocaps(px,py,pz,x,0.5),'FaceAlpha',0.1,'EdgeAlpha',0.1);
p3=patch(isosurface(px,py,pz,fixed,0.99),'FaceAlpha',0.1,'EdgeAlpha',0.1);
p4=patch(isocaps(px,py,pz,fixed,0.99),'FaceAlpha',0.1,'EdgeAlpha',0.1);
set(p3,'FaceColor',[1.0 0.0 1.0],'EdgeColor','none');
set(p4,'FaceColor',[1.0 0.0 1.0],'EdgeColor','none');
light('Position',[0 ny -nz]);
axis equal, axis([0 nx 0 ny 0 nz]); view(3); box on;
set(gca,'ZDir','reverse','YDir','reverse')
xlabel('x'), ylabel('y'), zlabel('z')
set(gcf,'color','white');
drawnow
%% Save results
save([ProbScript '_output'])
if exist('video','var') == 1, close(vidObj); end
frame=getframe(gcf);
[X,map]=frame2im(frame);
imwrite(X,[ProbScript '.png'],'png')
figure(2)
legend('Compliance','Max. change in density')
%% Plot bricks
h=elSize;
fac=[1 2 3 4; 2 6 7 3; 4 3 7 8; 1 5 8 4; 1 2 6 5; 5 6 7 8];
figure(4)
MAP=bone(2^8);
```

```
for i=1:nx % x-direction
  X=(i-1)*h;
  for k=1:nz % z-direction
    Z=nz*h-(k-1)*h;
    for j=1:ny % y-direction
      Y=(j-1)*h;
      if x(k,j,i) > 0.5 % 0.5 is the ISO value
        vert=[X Y Z; X Y+h Z; X+h Y+h Z; X+h Y Z; X Y Z+h;X Y+h Z+h;...
          X+h Y+h Z+h;X+h Y Z+h];
          patch('Faces',fac,'Vertices',vert,'FaceColor',...
            MAP(round((1-x(k,j,i))*2^8)+50,:))
        hold on;
      end
    end
  end
end
axis([0 h*nx 0 h*ny 0 h*nz]); axis equal; view(30,30); box on;
set(gca,'YDir','reverse')
@finnish;
function finnish(~,~,~)
global abort abortfig, abort=1;
close(abortfig)
```

Appendix A-5 Script for the S-bridge problem

```
%-----------------------------------------------
%Script defining the S-bridge problem for top3D.
%-----------------------------------------------
%% Size of design domain
lx=150; ly=70; lz=40;
elSize=1.1;
nx=round(lx/elSize); if rem(nx,2) ~= 0, nx=nx+1; end
ny=round(ly/elSize); if rem(ny,2) ~= 0, ny=ny+1; end
nz=round(lz/elSize);
ndof=3*(ny+1)*(nx+1)*(nz+1); nel=nx*ny*nz;
%% Prescribed void or solid areas
deckWidth=nx/10;
deckY=round(4/10*ny*-cos((1:nx)*(pi/nx))+ny*5/10);%y-coord. for deck centre
deckMat=zeros(ny,nx);
deckMat(deckY+(ny*(0:length(deckY)-1)))=1;%centre line as 'image'-matrix
deckMat=[deckMat(1:ny/2,:); rot90(deckMat(1:ny/2,:),2)];%ensure symmetry
deckDist=bwdist(deckMat);
deckMat(deckDist>deckWidth/2)=0;
deckMat(deckDist<=deckWidth/2)=1;%widen centre line
%Coordinates for deck elements
[by,bx]=find(deckMat==1); bz=round(nz-10/elSize)*ones(size(bx));
%Coordinates for open space above deck
w1x=(repmat(bx,round(10/elSize),1)); w1y=(repmat(by,round(10/elSize),1));
w1z=[]; for i=1:round(10/elSize), w1z=[w1z; bz-i]; end
%Coordinates for open space bellow bridge (highway)
w2y=repmat(round(deckY(1)+deckWidth/2+1):(ny-round(deckY(1)+...
deckWidth/2+1)),nx,1); w2y=w2y(:);
w2x=repmat(1:nx,1,length(unique(w2y)))';
w2z=repmat(max(bz)+1:nz,size(w2x),1); w2z=w2z(:);
w2y=repmat(w2y,nz-max(bz),1); w2x=repmat(w2x,nz-max(bz),1);
%Collect all coordinates for white elements
wz=[w1z; w2z]; wy=[w1y; w2y]; wx=[w1x; w2x];
%Matrices with black and white elements
elBlack=zeros(nz,ny,nx); elBlack(sub2ind([nz ny nx],bz,by,bx))=1;
elWhite=zeros(nz,ny,nx); elWhite(sub2ind([nz ny nx],wz,wy,wx))=1;
%% Boundary conditions
fixeddofs=[((nz+1)*3-2):(nz+1)*3:ndof ((nz+1)*3-1):(nz+1)*3:ndof ...
((nz+1)*3):(nz+1)*3:ndof];%entire bottom area
%% Loads
%--(Copied from main program, needed here to define F)
nodenrs=reshape(1:(1+nx)*(1+ny)*(1+nz),1+nz,1+ny,1+nx);
edofVec=reshape(3*nodenrs(1:end-1,1:end-1,1:end-1)-2,nx*ny*nz,1);
edofMat=repmat(edofVec,1,24)+repmat([0 1 2 3 4 5 3*(nz+1)+[3 4 5 0 1 2]...
((nz+1)*(ny+1)*3)+[0 1 2 3 4 5 3*(nz+1)+[3 4 5 0 1 2]]],nx*ny*nz,1);
%--
%Vectors containing element and dof numbering for deck elements
belVec=nz*ny*(bx-1)+nz*(by-1)+bz;
bdofVec=unique(reshape(edofMat(belVec,:),length(belVec)*24,1));
F=sparse(ndof,1); F(bdofVec(rem((bdofVec-1),3)==0))=-1;
```

```
%% Optimisation options
volfrac=0.04;
penal=[1.0 1.1 3.0];
rmin=3.3;
%% Numerical options
maxit=2000;
%% Run optimisation
save('sBridge_in')
Toptimiser3D('sBridge')
```

Appendix A-6 Script for the 2D torsional support

```
%-----------------------------------------------------------------
%Script defining the torsional support problem for Toptimiser 2D.
%-----------------------------------------------------------------
% Size of design domain
lx=0.5; ly=0.5;
elSize=0.004;
nx=round(lx/elSize); if rem(nx,2) ~= 0, nx=nx+1; end
ny=round(ly/elSize); if rem(ny,2) ~= 0, ny=ny+1; end
ndof=2*(ny+1)*(nx+1);
% Boundary conditions
fixeddofs=[2 1:(ny+1)*2:ndof ndof-2*ny:2:ndof ndof-1];
symm=[1 4; 3 2];
%Loads
load_nodes=[ndof-2*2*(ny+1)+2 ndof-2*ny+1];
F=sparse(load_nodes,ones(size(load_nodes)),[1 -1],ndof,1);
%Optimisation options
volfrac=0.2;
pen=[1 1.1 3];
rmin=1.1;
%Numerical options
tol=1e-3;
maxit=2000;
%Stress options
stress=1;
%Run optimisation
save('torSupp_in')
Toptimiser2D('torSupp)
```

Appendix A-7 Script for the 3D torsional support

```
%-----------------------------------------------------------------
%Script defining the torsional support problem for Toptimiser 3D.
%-----------------------------------------------------------------
%% Size of design domain
lx=1; ly=1; lz=1;
elSize=0.03;
nx=round(lx/elSize); if rem(nx,2) ~= 0, nx=nx+1; end
ny=round(ly/elSize); if rem(ny,2) ~= 0, ny=ny+1; end
nz=round(lz/elSize);
ndof=3*(ny+1)*(nx+1)*(nz+1); nel=nx*ny*nz;
% Boundary conditions
fixeddofs=[(3*(nz+1)*(ny/2+1)-2):(3*(nz+1)*(ny/2+1))...
  (3*(nz+1)*((ny+1)*(nx/2)+1)-2):(3*(nz+1)*((ny+1)*(nx/2)+1))...
  (3*(nz+1)*(ny+1)*(nx/2+1)-2):(3*(nz+1)*(ny+1)*(nx/2+1))...
  (ndof-3*(nz+1)*(ny/2)-2):(ndof-3*(nz+1)*(ny/2)-0)]; %All four sides
% Loads
F=sparse([3*(nz+1)*(ny+1)*(nx/2-1)+3*(nz+1)*ny/2+2 ...
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*(ny/2-1)+3 ...
 3*(nz+1)*(ny+1)*(nx/2+1)+3*(nz+1)*ny/2+2 ...
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*(ny/2+1)+3 ... %first direction
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*ny/2+1 ... %vertical
 3*(nz+1)*(ny+1)*(nx/2-1)+3*(nz+1)*ny/2+2 ...
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*(ny/2-1)+3 ...
 3*(nz+1)*(ny+1)*(nx/2+1)+3*(nz+1)*ny/2+2 ...
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*(ny/2+1)+3 ... %other direction
 3*(nz+1)*(ny+1)*nx/2+3*(nz+1)*ny/2+1 ... %vertical
 ],[ones(1,5) 2*ones(1,5)],[-1 1 1 -1 1.2  1 -1 -1 1 1.2],ndof,2);
% Optimisation options
volfrac=0.07;
pen=3.0;
```

```
rmin=1.1;
%Run optimisation
save('torSupp_in')
Toptimiser3D('torSupp')
```