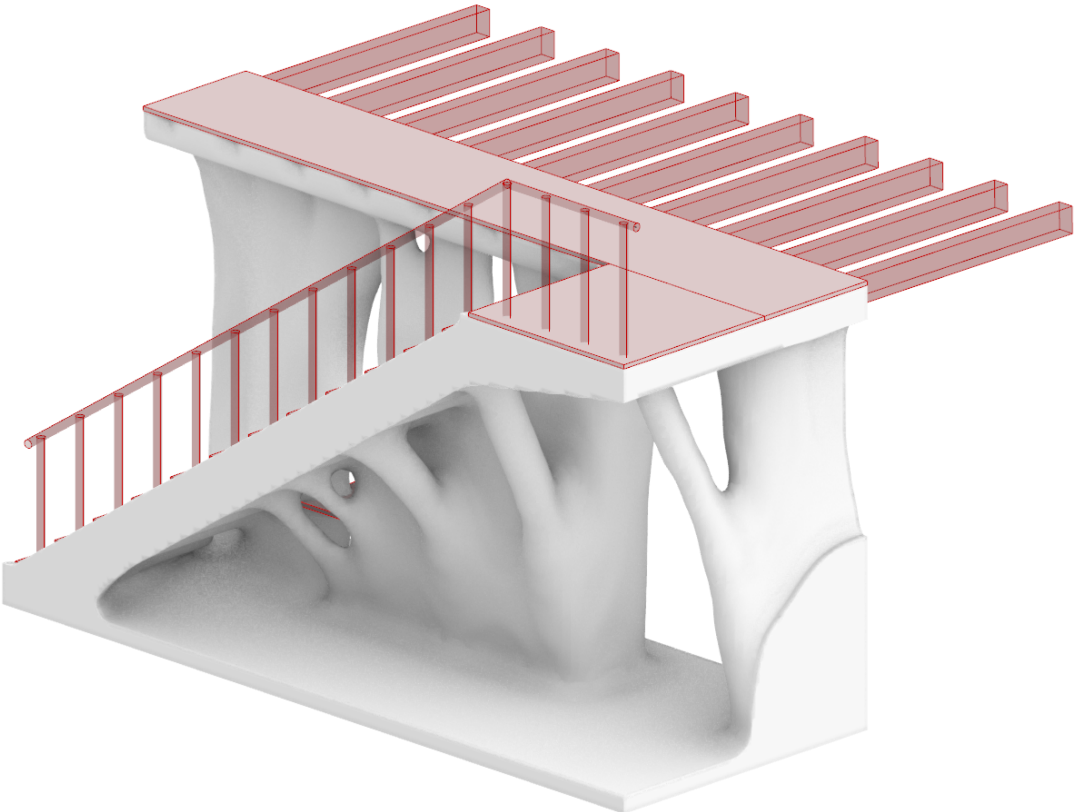


Advancing Sustainable Construction

Optimizing 3D-Printed Bio Fiber-Based Lightweight Building Elements

Degree project report in Structural Engineering and Building Technology

HERMAN EHRNBERG & SIMON WIKSTRÖM

DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Degree project report 2025

Advancing Sustainable Construction

Optimizing 3D-Printed Bio Fiber-Based Lightweight Building Elements

HERMAN EHRNBERG & SIMON WIKSTRÖM

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Architecture and Civil Engineering
Chalmers University of Technology
Gothenburg, Sweden 2025

Advancing Sustainable Construction
Optimizing 3D-Printed Bio Fiber-Based Lightweight Building Elements
HERMAN EHRNBERG & SIMON WIKSTRÖM

Supervisor: Robert Jockwer, Department of Architecture and Civil Engineering
Examiner: Robert Jockwer, Department of Architecture and Civil Engineering

Degree project report 2025
Department of Architecture and Civil Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Showcase Project 5, the optimized staircase.

Typeset in LaTeX
Gothenburg, Sweden 2025

Advancing Sustainable Construction
Optimizing 3D-Printed Bio Fiber-Based Lightweight Building Elements
HERMAN EHRNBERG & SIMON WIKSTRÖM
Department of Architecture and Civil Engineering
Chalmers University of Technology

Abstract

To address the continuous global population growth with the goal of simplifying the construction of affordable and environmentally friendly living spaces this thesis focuses on the impact of the building materials. In collaboration with an ongoing research project at Chalmers University of technology on cellulose-based Hybrid Composite Materials used in 3D-printing building elements a user-friendly design tool is created. The tool is created as a Grasshopper component for Rhinoceros 3D with the main goal of simplifying the use of the 3D-printable Hybrid Composite Material for the building sector. It takes its basis in Finite Element Analysis calculations in combination with Generative Design and Topological Optimisation as well as theory for 3D-Printing. From the research project access to the material as well as the 3D-printer is achieved making the collection of necessary data possible. When finalised, the program is used to showcase a few cases, both as 3-dimensional computer models as well as 3D-printed. At the end of the report the results as well as the different methods are discussed and examples of how to bring the subject even further, improvements for the program, as well as further ideas are debated.

Keywords: Hybrid Composite Material, Finite Element Method, Generative Design, Topology Optimisation, Grasshopper, 3D-Printing

Acknowledgements

We would like to express our sincere gratitude to the following individuals who have played a crucial role in the completion of this Master's thesis:
We are deeply thankful to our supervisor, Robert Jockwer, for their guidance, support, and invaluable insights throughout the research process. Their expertise and encouragement have been instrumental in shaping the trajectory of this thesis. Especially regarding questions about the behaviour of the researched material.
We would also like to acknowledge Jonas Lundberg for their collaboration and contributions to this research. Their insights and collaboration have enhanced the scope and depth of the study. Especially regarding questions about use of the 3D-Printer and the robot arm.
In conclusion, this thesis stands as a testament to the collective effort of those who have been part of our academic journey. Thank you all for your support and encouragement.

Herman Ehrnberg & Simon Wikström, Gothenburg, January 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BCM	Bio Composite Material
B-rep	Boundary Representation
CAD	Computer Aided Design
DOF	Degree Of Freedom
FEA	Finite Element Analysis
FEM	Finite Element Method
MPa	Mega Pascal

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

n	Index
i	Index

Sets

Ω	Domain of the Isoparametric space
N_e	A set of elements

Parameters

a	Hermitian Constant
b	Hermitian Constant
c	Hermitian Constant
d	Hermitian Constant
k	Dampening constant
P	Penalisation coefficient
S	Rate of change coefficient
Tol	Fitness tolerance
tU	Target utilisation value
w	Weights of the Gauss points
x	Positions of the Gauss points

Variables

D_e	Density of an element
E	Young's Modulus
F	Material Strength
G	Shear Modulus
gD	Sum of element densities
gU	Sum of element utilisation values
r	Radius
T	Fitness value threshold
u	Displacement of the nodes
v	Displacement of the nodes
v_i	Volume of an element
w	Displacement of the nodes
x	Global axis of the Cartesian space
y	Global axis of the Cartesian space
z	Global axis of the Cartesian space
ZS_e	Fitness value
η	Local axis of the Isoparametric space
ξ	Local axis of the Isoparametric space
ζ	Local axis of the Isoparametric space
σ	Normal Stresses
ρ_i	Density of an element
ϵ	Normal Strains
τ	Shear Stresses
γ	Shear Strains
ν	Poissons Ratio

Contents

List of Figures

1 Introduction

1.1 Background

As the global population continues to grow, there is an escalating demand for affordable and environmentally-friendly living spaces. While building standards have made progress in reducing the energy demand during the operational phase of structures, there remains a pressing need to focus on improving the building materials themselves. To address this challenge, an innovative production process utilizing environmentally-friendly building materials is essential. Engineered timber and wood-based products, integrated into 'Hybrid Composite Materials,' have the potential to revolutionize the construction industry and promote a sustainable built environment. These biobased materials offer numerous advantages, including eco-friendliness, lightweight properties, and the possibility of efficient 3D-printing, thus making them ideal candidates for construction.

To achieve this vision, an interdisciplinary research team from Chalmers University of Technology, combining expertise from the department of Chemistry and Chemical Engineering with the department of Architecture and Civil Engineering, is dedicated to optimizing the performance of the 'Hybrid Composite.' Their primary objective is to develop 3D-printed bio fiber-based lightweight building elements that are both structurally efficient and environmentally sustainable. The focus lies on investigating the structural properties of the material and understanding the impact of various parameters during its production process.

In the building sector today there are many different projects working with 3D-printable building materials. Most of which having focus on fully 3D-printable homes in order to simplify and speed up the building process to be able to keep up with the population growth. The projects does, however, in general not focus on the environmental impact and the most popular material used is concrete, which is a very suitable material for 3D-printing due to its characteristics, although not suitable when it comes to environmental impact. Other materials that are being used are Mortar, which is a material that is more flexible than concrete and therefore can be made less environmentally damaging and Soil, which is less flexible and durable but with almost no carbon footprint. Also Special Polymers as well as Recycled Plastic are used, which are great materials in flexibility and durability, but even though they are highly recyclable, they lack in environmental quality due to not being biological materials. There are a few projects focusing on the use of bio-based materials, such as wood, for example the BioHome3D developed by the University of Maine's Advanced Structures & Composites Center. These materials are great when it comes to durability, flexibility and environmental impact, however, since the materials are in a novel state not many published research papers on the area exist, especially when it comes to the structural properties of the materials.

1.2 Purpose

The purpose of this thesis project is to continue the discussion of environmentally friendly building materials and their construction methods by adding to the research of a Hybrid Composite material that can be used in 3D-printed building elements. There are high hopes that this discussion can bring forward further development on how to construct building elements in an efficient, simple and environmentally friendly way.

1.3 Goals

The main aim of this research project is to continue development of 3D-printed bio fiber-based lightweight building elements using the 'Hybrid Composite Materials', thereby promoting a circular economic model and fostering sustainability in the built environment. More specifically, the thesis aims to develop a material model for this innovative and novel material in order to create a parametric model for optimization of building elements. Put into more concrete terms the final goal of the thesis is a Grasshopper Plug-In Component with the purpose of optimizing a 3D-printed bio fiber-based lightweight building element structurally, by working parametrically from the specific boundary conditions of the element as input. Secondary goals of the thesis includes a finalized and printed building element, optimized for specific boundary conditions with the developed Grasshopper Plug-In Component, as well as additional flexibility incorporated in the Grasshopper Plug-In.

1.4 Limitations / Demarcations

This thesis mainly focuses on the research of 3D-printed hybrid composite materials as to be used in building elements, with the goal of creating a software to simplify the use of such a material and the hope of bringing the discussion on environmentally friendly materials in the building sector forward. The software is to be created for the use of architects and civil engineers to in a simple way create somewhat optimised building elements, both very generally and also in complex cases where calculations might be more difficult. The purpose of the software is to simplify optimised use of material, while at the same time giving possibility for a new construction method and the use of a new material.

The software takes its basis in the finite element method, which contains a few simplifications. All calculations will be performed on 3D hexahedral elements with 27

Gauss points and with 3 degrees of freedom regarding deformation, there will be no calculations performed on rotation. There will be possibility for both isotropic and orthotropic materials, however, the main focus will be on orthotropic materials. The software will be created in C# as several Grasshopper-components for Rhinoceros 3D.

The material properties used in the finite element analysis will be derived from tensile testing of samples of the material in different directions The properties not found in these tests will be assumed from the results with regards to the expected behaviour of the material. There will be no testing with different printer settings or with regards to moisture.

Comparison tests, for the reliability of the FEM-software, will be performed, however, only on stress-patterns and magnitudes. No comparison tests will be performed on deformations or on the function of the topological optimisation.

No 3D-printing will be performed with the 3D-printer or the material from the research project, however, small prototypes might be printed in plastic as a showcase of the software.

2
Theory

In the following section the theory supporting the different methods used in this thesis is presented. It is divided into several parts in the same way as the methods themselves.

2.1 The Bio Composite Material

The Bio Composite Material is material that is specifically looked into by the interdisciplinary research team and will from here on out be referred to as the BCM. The BCM is a Hybrid Composite material, which means that it consist of several materials blended together to create a new material with the preferred properties. The BCM researched in this project is a Woodcompound from Woodcomposite Sweden AB called "Woodcompound 3D S50 Flex K" and exist in the form of pellets that are loaded into the 3D-printer. The material is typically used for printing furniture, decorative items, interior products or windows, but should be apliable to almost any structure.

2.1.1 Material Properties

The BCM consist of a bio-based polymer matrix reinforced with wood-based materials and is generally suitable for replacing the polymers ABS, PET-G, PLA and PPGF. The bio-based polymers are renewable and ISCC+ certified and contains waste and residues from vegetable oil refining and cooking oil. The certification takes the entire value chain into account, which includes traceability to point of origin.

In order to find the material properties of the BCM, firstly a few assumptions about the material would needed to be made. These properties were necessary to find for creating the material model needed for the FEA. To understand which properties were required to measure in the experiments, it was necessary to create a greater understanding about the material first. The BCM is categorised as an orthotropic material, which is a class of anisotropic material for which three planes of symmetry exist and with the principal axes in rectangular cartesian coordinates (Zienkiewicz et al., 2005). Another orthotropic material, which can be used as a guide in this case is wood, and as the BCM consist of wood to at least 50% it can be safe to say that its properties should be somewhat similar. In wood the three general directions of the material are defined as Longitudinal, Tangential and Radial, depending following the direction of the timber grains. The BCM does not contain grains, but does however, as it is a 3D-printed material, contain a printing direction. As such the three directions of the BCM material properties will be defined as Parallel, Horizontally Perpendicular and Vertically Perpendicular according to the printing direction, or shortly as P, H and V.

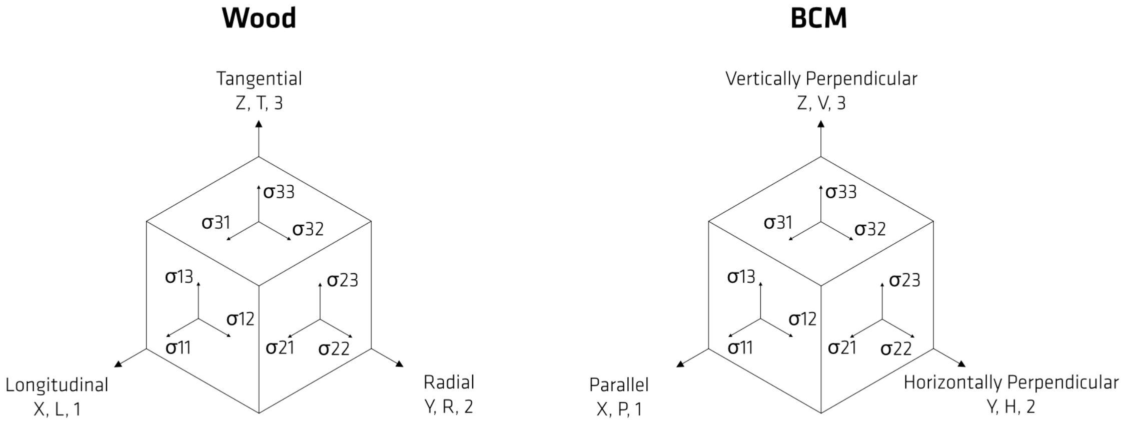


Figure 2.1: A comparison of the directions of the material properties of Wood and BCM.

In the FEA of this project, the material was analysed as a linear elastic material. As such, what was needed in the form of material properties was the elastic material parameters. For orthotropic materials, it is common to define these in terms of the materials Young's moduli, shear moduli and Poisson's ratios in the three different directions (Zienkiewicz et al., 2005). As the BCM is a 3D-printed material, it was also of interest to analyse the impact of printing speed, printing size and moisture content.

From the manufacturer of the pellets used in the printer a few material properties were provided, as well as optimal printing temperatures and maximum printing angles. According to this technical data sheet the material had a tensile strength of 30 MPa and a modulus of 2700 MPa, and could be printed at an angle of maximum 45 degrees in a temperature of 175 – 190 degrees Celsius, as can be seen in Figure 2.3.

2.1.2 Material Model

The material model of a material is a description of the relationship between its stress and strain in different directions, this chapter follows the method found in *"The Finite Element Method: Its Basis and Fundamentals"* (Zienkiewicz et al., 2005) to establish the structure of the material model for a 3-dimensional, solid, orthotropic material such as the BCM investigated in this thesis. In Finite Element Analysis, the stress-strain equations, also known as the constitutive relations, for a linear elastic material may be expressed by:

Typical properties and technical data	Standard	3D S50 Flex K	Unit
Wood content (weight)	-	50	%
Density	ISO 1183	1,10	g/cm³
MVR (220°C/5kg)	ISO 1133	-	cm³/10min
Tensile strength	ISO 527-2/50	30	MPa
Tensile modulus	ISO 527-2/2	2700	MPa
Strain at break	ISO 527-2/50	7,0	%
Flexural strength	ISO 178	43	MPa
Flexural modulus	ISO 178	2800	MPa
Charpy impact strength, 23°C	ISO 179/1eU	26	kJ/m2
Charpy notched impact, 23°C	ISO 179/1eA	-	kJ/m2
HDT A 1,8 MPa	ISO 75-2/A	-	°C

Figure 2.2: Typical properties and technical data of the BCM pellets.

3D printer settings

Typical printer parameters

Nozzle diameter :	2-10 mm
Melt density :	1,08 g/cm³
Extruder temperature	175°C-190°C
Bed temperature	130°C-140°C
Printing speed:	2500 – 3500 mm/min
Extrusion width:	30-50% higher than nozzle diameter
Maximum angles:	-45 – 45°
Layer cooling time:	> 1 min
Part removal:	Remove prior to bed cooling below 90°C
Surface quality:	Better surface quality is achieved when the pressure through the nozzle is high. Several ways to achieve this are possible including changing nozzle, increasing extrusion multiplier and increasing the layer width. Higher print speed also increases surface quality.

Figure 2.3: Technical data sheet for the 3D-printer settings.

$$\sigma = D(\epsilon - \epsilon_0) + \sigma_0 \tag{2.1}$$

$$\epsilon = D^{-1}(\sigma - \sigma_0) + \epsilon_0 \tag{2.2}$$

Where the D matrix is known as the elasticity matrix of moduli. The structure of this relationship varies between different types of materials, for anisotropic linear elastic materials the relationship can be written generally as:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} & D_{15} & D_{16} \\ D_{21} & D_{22} & D_{23} & D_{24} & D_{25} & D_{26} \\ D_{31} & D_{32} & D_{33} & D_{34} & D_{35} & D_{36} \\ D_{41} & D_{42} & D_{43} & D_{44} & D_{45} & D_{46} \\ D_{51} & D_{52} & D_{53} & D_{54} & D_{55} & D_{56} \\ D_{61} & D_{62} & D_{63} & D_{64} & D_{65} & D_{66} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} \tag{2.3}$$

For linear elastic materials the D matrix has to be symmetric, leading to

$$D_{xy} = D_{yx} \tag{2.4}$$

Orthotropic materials, such as the BCM is as earlier mentioned a class of anisotropic materials for which three planes of symmetry exist. For an orthotropic material the principal axes are also in rectangular cartesian coordinates and it is common to define the materials elastic parameters in terms of their Young's moduli, shear moduli and the Poisson's ratio.

For an orthotropic material where x, y, z are the three axes of material symmetry, the elastic strain-stress relations can be written as:

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{xy}}{E_y} & -\frac{\nu_{xz}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{yx}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{yz}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{zx}}{E_x} & -\frac{\nu_{zy}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} \tag{2.5}$$

Where E are elastic moduli, v are Poisson's ratios and G are elastic shear moduli. As the D matrix has to be symmetric, due to linear elasticity, this results in:

$$\frac{\nu_{ij}}{E_j} = \frac{\nu_{ji}}{E_i} \tag{2.6}$$

2.2 FEM 3D

To optimise a building material structurally in three dimensions a thorough analysis of the material with regards to its material properties is necessary. In this project the analysis was done according to the Finite Element Method in three dimensions. The majority of three dimensional FE are based on hexahedrons or tetrahedrons with 3 degrees of freedom per node and linear or quadratic shape functions (López Machado et al., 2022). In the case of BCM, it was modelled with hexahedral elements with 8 nodes per element, 3 DOFs per node and quadratic shape functions, meaning 3 nodes per edge. These quadratic shape functions are necessary as a minimum to capture the bending and shear effects of a 3D FEM (López Machado et al., 2022). The two main theories for beam formulation are Bernoulli's formulation and Timoshenko's Theory. As Bernoulli's formulation neglects the shear deformation while Timoshenko's Theory considers it, the later is more applicable in this case.

2.2.1 Isoparametric Formulation

The advantage of the Isoparametric formulation is that it is able to capture bending effects directly in the Isoparametric space, with only 8 nodes in a regular hexahedron (López Machado et al., 2022). The domain of the Isoparametric space can be seen in Equation 2.7, with its three local axes. To generate the shape functions for these axes the Hermitian polynomial to the third degree was selected for the interpolation.

$$\Omega = (-1, 1) \quad (\eta, \xi, \zeta) \tag{2.7}$$

$$P(\xi) = a + b\xi + c\xi^2 + d\xi^3 \tag{2.8}$$

Where the rotation at any point of P will be equal to the partial derivative and the constants a, b, c, d can be calculated, considering that the boundary conditions are taken in 2D for each edge as w_1, w_2, representing the displacement perpendicular to the edge axis respectively, according to:

$$\begin{bmatrix} \omega 1 \\ \omega 2 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{2.9}$$

According to this the DOFs of the hexahedron, as well as the DOFs of its edges can be visualized as Figures 2.4.
The shape functions for each edge can then be obtained by solving Equation 2.9 for constants a, b, c, d and then grouping w_1, w_2, and expanding the solution for each of the local axes. The final shape functions can be seen in Equations 2.10 and 2.11, where H_1 and H_2 represent displacement.

$$H_1(\xi) = \frac{\xi^3}{4} - \frac{3\xi}{4} + \frac{1}{2} \tag{2.10}$$

$$H_2(\xi) = -\frac{\xi^3}{4} + \frac{3\xi}{4} + \frac{1}{2} \tag{2.11}$$

Next, the shape functions of each node in the Isoparametric space are constructed. To do this, a superposition of the corresponding shape functions are proposed, resulting in the following equations for displacements, Equations 2.12-2.19.

$$NH_1(\eta, \xi, \zeta) = H_2(\xi)H_2(\eta)H_2(\zeta) \tag{2.12}$$

$$NH_2(\eta, \xi, \zeta) = H_2(\xi)H_1(\eta)H_2(\zeta) \tag{2.13}$$

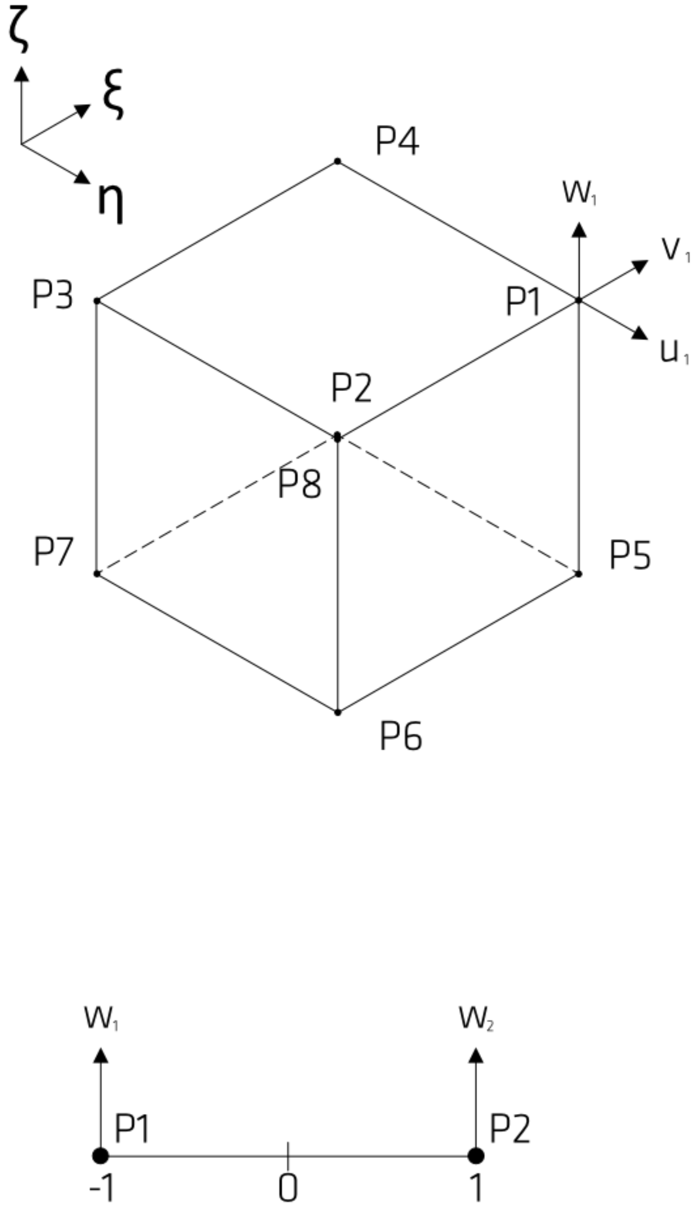


Figure 2.4: DOFs of the hexahedral elements and its edges.

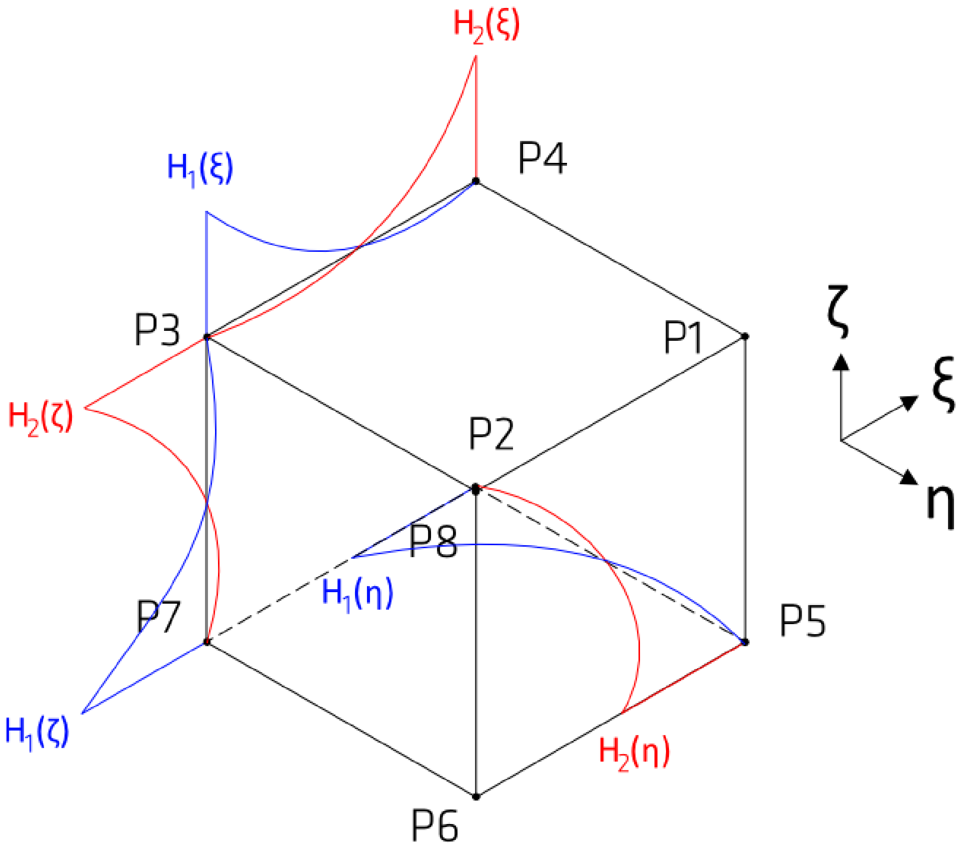


Figure 2.5: Nodal shape functions.

$$NH_3(\eta,\xi,\zeta) = H_1(\xi)H_1(\eta)H_2(\zeta) \tag{2.14}$$
$$NH_4(\eta,\xi,\zeta) = H_1(\xi)H_2(\eta)H_2(\zeta) \tag{2.15}$$
$$NH_5(\eta,\xi,\zeta) = H_2(\xi)H_2(\eta)H_1(\zeta) \tag{2.16}$$
$$NH_6(\eta,\xi,\zeta) = H_2(\xi)H_1(\eta)H_1(\zeta) \tag{2.17}$$
$$NH_7(\eta,\xi,\zeta) = H_1(\xi)H_1(\eta)H_1(\zeta) \tag{2.18}$$
$$NH_8(\eta,\xi,\zeta) = H_1(\xi)H_2(\eta)H_1(\zeta) \tag{2.19}$$

When the shape functions in the Isoparametric space has been obtained, a transformation to the Cartesian space is necessary. This transformation is done through the Jacobian matrix in Equation 2.20.

$$J = \begin{bmatrix} \frac{\partial x}{\partial \eta}(\eta,\xi,\zeta) & \frac{\partial y}{\partial \eta}(\eta,\xi,\zeta) & \frac{\partial z}{\partial \eta}(\eta,\xi,\zeta) \\ \frac{\partial x}{\partial \xi}(\eta,\xi,\zeta) & \frac{\partial y}{\partial \xi}(\eta,\xi,\zeta) & \frac{\partial z}{\partial \xi}(\eta,\xi,\zeta) \\ \frac{\partial x}{\partial \zeta}(\eta,\xi,\zeta) & \frac{\partial y}{\partial \zeta}(\eta,\xi,\zeta) & \frac{\partial z}{\partial \zeta}(\eta,\xi,\zeta) \end{bmatrix} \tag{2.20}$$

As can be seen in Equation 2.20, for the Jacobian matrix the partial derivatives of the shape functions expressed in Cartesian coordinates are necessary. The shape functions can be expressed in Cartesian coordinates according to Equation 2.21-2.23.

$$x = \sum_{i=1}^{n} NH_i(\eta,\xi,\zeta)x_i \tag{2.21}$$
$$y = \sum_{i=1}^{n} NH_i(\eta,\xi,\zeta)y_i \tag{2.22}$$
$$z = \sum_{i=1}^{n} NH_i(\eta,\xi,\zeta)z_i \tag{2.23}$$

And the transformation from Isoparametric to Cartesian coordinates as in Equation 2.24.

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{bmatrix} = (J^{-1})^T \begin{bmatrix} \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \zeta} \end{bmatrix} \tag{2.24}$$

2.2.2 Elasticity Parameters

With the system transformed from Isoparametric to Cartesian coordinates, the focus can move to Elasticity Parameters. The deformation tensor (matrix B) for an orthotropic material, based on the three degrees of freedom per node, is constructed from the partial derivatives of the shape functions in Cartesian coordinates, found through the Jacobian matrix, and can be seen in Equation 2.25.

$$\begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x}NH_n & 0 & 0 \\ 0 & \frac{\partial}{\partial y}NH_n & 0 \\ 0 & 0 & \frac{\partial}{\partial z}NH_n \\ \frac{\partial}{\partial y}NH_n & \frac{\partial}{\partial x}NH_n & 0 \\ 0 & \frac{\partial}{\partial z}NH_n & \frac{\partial}{\partial y}NH_n \\ \frac{\partial}{\partial z}NH_n & 0 & \frac{\partial}{\partial x}NH_n \end{bmatrix} \begin{bmatrix} u_n \\ v_n \\ w_n \end{bmatrix} \tag{2.25}$$

In Equation 2.26 the displacement matrix (matrix N) for one node is shown as a 3 by 3 matrix, for an entire element the dimension of the matrix would be 3 by 24.

$$N = \begin{bmatrix} NH_n & 0 & 0 \\ 0 & NH_n & 0 \\ 0 & 0 & NH_n \end{bmatrix} \tag{2.26}$$

The material properties are, as expressed earlier, shown in Equation 2.27, also called matrix D.

$$\begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{xy}}{E_y} & -\frac{\nu_{xz}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{yx}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{yz}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{zx}}{E_x} & -\frac{\nu_{zy}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{zx}} \end{bmatrix} \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{zx} \end{Bmatrix} \tag{2.27}$$

2.2.3 Elemental Stiffness and Forces Matrices

The formulated equation describing how to calculate the elemental stiffness as well as the elemental force matrix is called the strong formulation of the matrices and is shown in Equation 2.28-2.29. The strong formulation is, however, too complex to evaluated due to the transformation between Isoparametric and Cartesian coordinates. To be able to calculate these matrices in with the Finite Element Method they are instead simplified to a form called the weak formulation, which can be seen in Equation 2.30-2.31.

$$K_e = \int_\Omega B^T DBJd\Omega \tag{2.28}$$

$$F_e = \int_\Omega N^T q(x)Jd\Omega \tag{2.29}$$

$$K_e = K_e + \sum_{i=1}^{n} w_i B(x_i)^T DB(x_i)detJ \tag{2.30}$$

$$F_e = F_e + \sum_{i=1}^{n} w_i N(x_i)^T q(x)detJ \tag{2.31}$$

The weak formulation solves the integration problem of calculating the element stiffness matrix with numerical integration based on Gaussian method, with 3 integration (Gauss) points per edge of the element, resulting in 27 integration points. These points are obtained by the permutation of the values for position and weight

$$x = \left[-\sqrt{\tfrac{3}{5}}, \quad 0, \quad \sqrt{\tfrac{3}{5}}\right] \tag{2.32}$$

$$w = \left[\tfrac{5}{9}, \quad \tfrac{8}{9}, \quad \tfrac{5}{9}\right] \tag{2.33}$$

In the weak formulation q(x) is the Neumann condition or boundary conditions which represents the external forces on the element and n is the number of integration points.

2.2.4 Global Assembly

The global matrices **K** and **F** are obtained from the global assembly of Ke and Fe, where they are assembled according to the connectivity matrix making sure that each node is connected to the correct elements in the global space.

The system is then solved for the deformations according to Equation 2.34.

$$Ka = f \tag{2.34}$$

Where K is the global stiffness matrix, f the global force vector and a the global deformation vector. To solve the system for the deformations an iterative solver is used.

The deformations can then be used to find the global strain vector with use of the B-Matrix according to Equation 2.25 and the stresses in the different directions can then be found through the Material Model in Equation 2.27.

2.3 Parametric Model

To able to use the FEM defined in Section 2.2 for optimization a parametric model that enables iterative FEM-driven morphological structural optimization of structural elements is needed. This parametric model should work with a generative design method in collaboration with the FEM to iterate towards an optimized solution to the preset boundary conditions, and should result in a finalized model ready for 3D-printing.

2.3.1 Generative Design

Generative Design is a design process that uses computational method iteratively to reach the specific design outcomes (Urquhart et al., 2022). A generative design process starts with the initial constraints, which are established by the designer, that then engages with form-finding algorithms which are applied within a CAD environment. These form-finding algorithms establish the design options within the constraints, but the designer is, however, still required to apply knowledge, intuition and insight in the development of the design (Urquhart et al., 2022). The design process can be seen as a subset of Computational design with solutions that are more open ended, as the most advanced generative algorithms does not only use multiple forms of data to form the design, but also iterates through the process and converge on an optimized solution to several predefined constraints (Urquhart et al., 2022).

While the generative design process is fairly established in the building sector in the field of architecture, were it is employed by several high-profile architects such as Zaha Hadid and Thomas Heatherwick (Urquhart et al., 2022), in the combination with 3D-printing, it is almost non-existent. Although, there exist project in professional design practice where generative design have been used in combination with

3D-printing, for products such as footwear, parametric pattern and texture design (Urquhart et al., 2022). The great advantage with using Generative Design is that the algorithm has the ability to create variations in form with a greater range of options in a much shorter time than manually, at the same time it can be used with a specific goal in mind, for example, optimized material usage.

There exist several methods of optimization within Generative Design, this project will focus on topological optimization. Topological optimization of materials is a mathematical design method working generatively to change the material layout to satisfy certain design goals, such as for example displacement goals or loading constraints (Urquhart et al., 2022). The optimization specifies only the domain loads and the displacement constraints while the material can be freely assigned to the different elements of the design domain, following goal searching functions, making it perfectly suitable for combining with the Finite Element Method. An example of how this method can be used to optimize a bridge structurally can be seen in Figure 2.6.

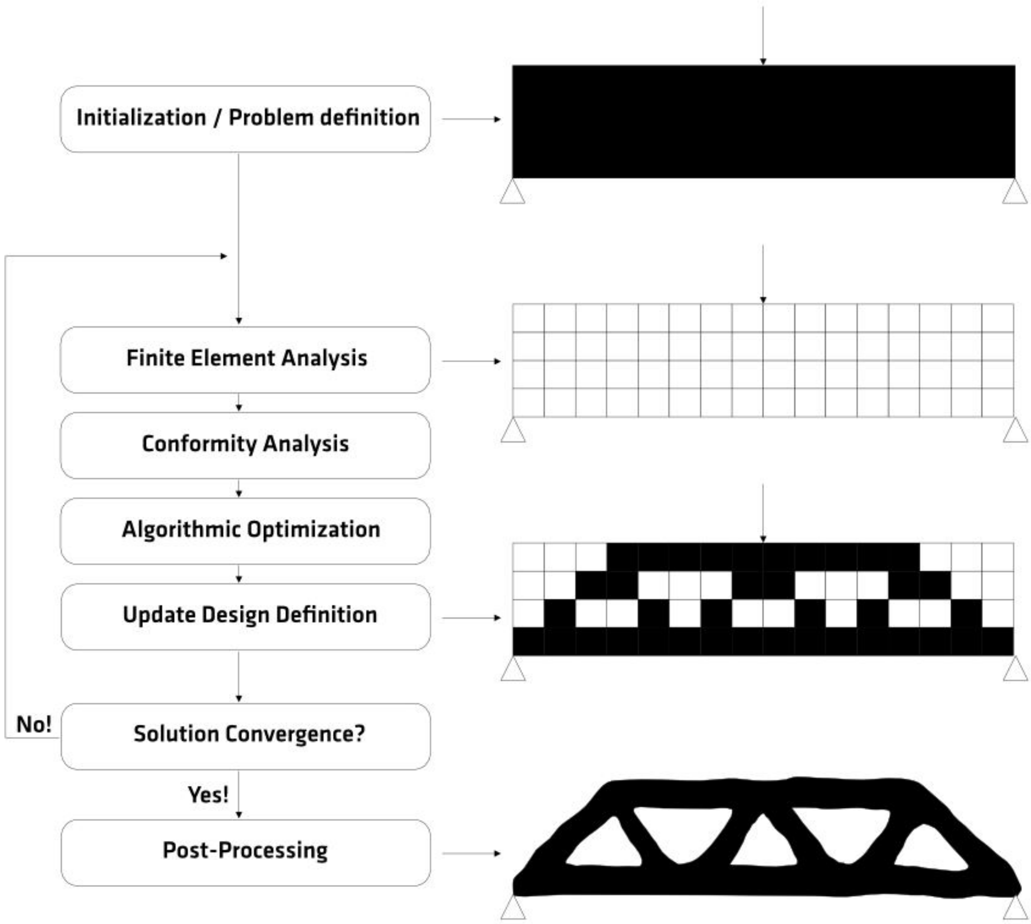


Figure 2.6: The Generative Design process with Topological Optimization for the material of a bridge.

When using the topological optimization, the method works following a structured layout. Firstly the predefined design domain is discretized into many smaller regions that are then each assigned a value matching their fictitious density (Lopar, 2020). As the densities can vary between fully void or fully solid, this value can vary between 0 and 1. As these smaller regions easily can be seen as elements building up a model it is fairly easy to incorporate this method with the Finite Element Method. After conducting a Finite Element Analysis on the model the different utilisation ratios of the elements can be found, and assigned as the elements density for the next iteration. By looking at the utilisation ratios of the elements the stress patterns of the solution can generally be used as a navigational tool. The elements most necessary for the structure are identified, as well as the elements least necessary. The highest ranking elements are kept for the next iteration and might be given a higher stiffness value, approaching but never surpassing the material capacity limits as to take more load, while the lowest ranking elements are either diminished or removed.

By controlling which elements are kept through the iterations as well as their stiffness the sequence can iterate generatively until convergence is achieved when the stiffness capacity of the elements are fully utilized and no more elements can be removed. Another parameter to look at for the convergence might be material usage, as the amount of material used might be of interest for the optimisation. The anticipated outcome is the emergence of a structure characterized by predominantly fully utilized elements.

2.3.2 Filters

An issue that might arise when using topology optimisation is that the stress patterns might not be very distinct in the early iterations, as the loads are not that high. This might result in the method removing or diminishing to many elements, and even elements that might be important later on. Another common issue is the occurrence of a checkerboard pattern in the design, which would make the design virtually impossible to manufacture (Lopar, 2020).

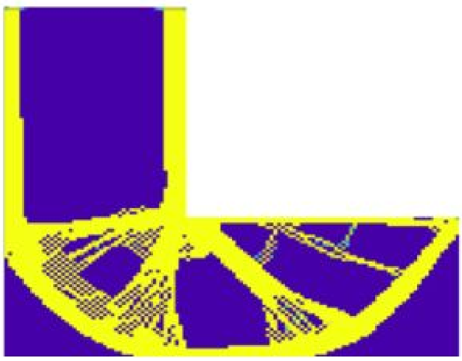


Figure 2.7: Checkerboard pattern (Lopar, 2020).

A solution that might be introduced to solve both of these issues are filters. A filter makes sure that the density values of all elements are influenced by the elements around them (Lopar 2020), working with different parameters to achieve the desired output. There are many different filters that might be used to achieve what is being searched for, which in this case is a concentration of utilisation ratios, to make sure that the design has legs to stand on. The filters usually works by using a radius of influence around the element, and taking influence from all elements that are inside the radius. The amount of influence received from an element depends on a weighing function that usually depends on the distance between the elements.

An example of a filter commonly used in topology optimizations is the density filter by Bruns and Tortorelli. This filter calculates the densities of the elements following Equation 2.35.

$$\rho_e = \frac{\sum_{i \in N_e} w(x_i) v_i \rho_i}{\sum_{i \in N_e} w(x_i) v_i} \tag{2.35}$$

This filter looks on all elements inside the set Ne, which can be decided using an influence radius. It also takes into account the volume of the element (v) and certain weighting functions (w) that can be adapted to specific needs.

A fairly common weighting function is the linear decaying function, also known as the cone-shaped function, which can be seen in Equation 2.36.

$$w(x_i) = r - |x_i - x_e| \tag{2.36}$$

This function finds the distance between the centres of elements i and e which makes it easily implemented.

A more complicated weighting function is the Gaussian distribution function, which can be seen in Equation 2.37.

$$w(x_i) = e^{-\frac{1}{2}(\frac{|x_i - x_e|}{R})^2} \tag{2.37}$$

When using this weighting function Ne may include all elements in the design, which makes it usable without any influence radius.

The main focus of many filters are to smooth out the result by blurring the values of the elements to create a more evenly distributed solution. Depending on the influence function as well as the radius the resulting design might have become to smooth, and areas with intermediate values, called grey zones, might have been introduced into the model. To diminish these zones, and to create sharper edges following the stress distribution, some sort of sharpening filter might be introduced. A method that takes sharpening to its extreme is called Heaviside projection.

When using Heaviside projection on the filtered densities they will be pushed to becoming either fully void or fully solid (Lopar, 2020). This will create a very sharp and clear model to be used for the next iteration.

2.3.3 Failure Criteria

To be able to compare the different elements of the model to each other, some sort of failure criteria should be used to define when an element is fully utilized and when it is unnecessary and should be removed. Today, several failure criteria exist and are being used for different materials, such as Von Mises, Maximum Stress Criteria and Maximum Strain Criteria, but as this thesis is working with a novel material, which criteria to be used will require some insight of the different criteria. In the program, different criteria might be used for different materials, for most homogenous ductile materials Von Mises yield criteria will be enough for analysis. However, for orthotropic materials such as wood and the BCM other criteria must be found.

Several of the usable criterias does however take their basis in the Von Mises yield criteria, which can be seen in Equation 2.38.

$$\sigma_v = \sqrt{\frac{1}{2}((\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2) + (\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)} \tag{2.38}$$

One failure criteria that has its roots in the Von Mises anisotropic yield condition is Hills yield criteria, which was a specialization of von Mises but adapted for the case of orthotropic materials. However, Hills criteria implies the same yield stress in both tension and compression for any orientation of the specimen (Malysko, 2011).

This issue could however be solved for orthotropic materials with varying yield strength in tension and compression by adding terms which are odd functions of σ_x, σ_Y and σ_Z as presented in Hoffmans generalisation of Hills limit condition (Hoffman, 1967), and can be seen in Equation 2.39.

$$C_1(\sigma_y-\sigma_z)^2+C_2(\sigma_z-\sigma_x)^2+C_3(\sigma_x-\sigma_y)^2+C_4\sigma_x+C_5\sigma_y+C_6\sigma_z+C_7\tau_{yz}^2+C_8\tau_{zx}^2+C_9\tau_{xy}^2 = 1 \tag{2.39}$$

Where the constants C1 – C9 are nine material parameters that are determined from nine basic strength data according to Equations 2.40-2.42.

$$C_1 = \frac{((F_{ty}F_{cy})^{-1} + (F_{tz}F_{cz})^{-1} - (F_{tx}F_{cx})^{-1})}{2} \tag{2.40}$$

$$C_4 = (F_{tx})^{-1} - (F_{cx})^{-1} \tag{2.41}$$

$$C_7 = (F_{syx})^{-2} \tag{2.42}$$

C2, C3, C5, C6, C8 and C9 can be found by permutation of x, y, z respectively.

Another interesting failure criteria, that as well is an interacting failure criteria, since all stress components are used simultaneously to determine whether failure has occurred or not, is the Tsai-Wu failure polynomial (Knight, 2006). The Tsai-Wu failure polynomial exist in both a 2D- and 3D-form with the 3D-form following Equation 2.43.

$$\phi = F_1\sigma_x + F_2\sigma_y + F_3\sigma_z + F_{11}\sigma_x^2 + F_{22}\sigma_y^2 + F_{33}\sigma_z^2 + 2F_{12}\sigma_x\sigma_y + 2F_{23}\sigma_y\sigma_z + 2F_{13}\sigma_x\sigma_z + F_{44}\tau_{zx}^2 + F_{55}\tau_{yz}^2 + F_{66}\tau_{xy}^2 \tag{2.43}$$

Where the terms F1-F66 are given by Equations 2.44-2.47.

$$F_1 = \frac{1}{F_{tx}} - \frac{1}{F_{cx}} \quad F_2 = \frac{1}{F_{ty}} - \frac{1}{F_{cy}} \quad F_3 = \frac{1}{F_{tz}} - \frac{1}{F_{cz}} \tag{2.44}$$

$$F_{11} = \frac{1}{F_{tx}F_{cx}} \quad F_{22} = \frac{1}{F_{ty}F_{cy}} \quad F_{33} = \frac{1}{F_{tz}F_{cz}} \tag{2.45}$$

$$F_{44} = \frac{1}{F_{szx}^2} \quad F_{55} = \frac{1}{F_{syz}^2} \quad F_{66} = \frac{1}{F_{syx}^2} \tag{2.46}$$

$$F_{12} = -\frac{1}{2\sqrt{F_{tx}F_{cx}F_{ty}F_{cy}}} \quad F_{13} = -\frac{1}{2\sqrt{F_{tx}F_{cx}F_{tz}F_{cz}}} \quad F_{12} = -\frac{1}{2\sqrt{F_{ty}F_{cy}F_{tz}F_{cz}}} \tag{2.47}$$

Failure in a certain material point is achieved when

$$\phi > 1 \tag{2.48}$$

Yet another failure criteria that might be interesting for the material in this project is the Hashin Failure Criteria. The Hashin Failure Criteria is an interacting failure criteria that uses more than a single stress component for evaluation of different failure modes (Knight, 2006). As it was originally developed to be used as a failure criteria for unidirectional polymeric composites, it might work well for the case of the BCM. Even though the Hashin Failure Criteria is an interacting failure criteria it is still divided into several failure modes, these are called Tensile fiber failure, Compressive fiber failure, Tensile matrix failure, Compressive matrix failure, Interlaminar normal tensile failure and Interlaminar normal compressive failure, and are presented in Equations 2.49-2.54.

$$(e_1^t)^2 = (\frac{\sigma_x}{F_{tx}})^2 + \frac{\tau_{xy}^2 + \tau_{zx}^2}{F_{syx}^2} \tag{2.49}$$

$$(e_1^c)^2 = (\frac{\sigma_x}{F_{cx}})^2 \tag{2.50}$$

$$(e_2^t)^2 = \frac{(\sigma_y + \sigma_z)^2}{F_{ty}^2} + \frac{\tau_{yz}^2 - \sigma_y\sigma z}{F_{syz}^2} + \frac{\tau_{xy}^2 + \tau_{zx}^2}{F_{syx}^2} \tag{2.51}$$

$$(e_2^c)^2 = ((\frac{F_{cy}}{2F_{syz}})^2 - 1)(\frac{\sigma_y + \sigma_z}{F_{cy}^2}) + \frac{(\sigma_y + \sigma_z)^2}{4F_{syz}^2} + \frac{\tau_{yz}^2 - \sigma_{y\sigma_z}}{F_{syz}^2} + \frac{\tau_{xy}^2 + \tau_{zx}^2}{F_{syx}^2} \tag{2.52}$$

$$(e_3^t)^2 = (\frac{\sigma_z}{F_{tz}})^2 \tag{2.53}$$

$$(e_3^c)^2 = (\frac{\sigma_z}{F_{cz}})^2 \tag{2.54}$$

Where if any failure criteria exceeding 1 equals failure in that mode.

When deciding what criteria to use for removing or diminishing elements in the generative design process it might also be interesting to look at the principal stresses. As the principal stresses are the stresses in a point broken down to the largest stress vectors that are normal stresses, this means that the principal stresses only exist when the stress vectors are rotated so that no shear stress exist in the point. Therefore they can be very useful in finding the stress pattern through the structure. If this pattern can be found and used to control which elements of the model to keep, it can be made sure that the model keeps elements following the pattern of the normal stresses throughout the entire structure.

2.4 3D-Printing

When creating a digital 3D-model with the main goal of being 3D-printable there are several things that needs to be taken into consideration. Firstly the 3D-printer itself has to be looked into, how does it work, what method does it use and what are its limitations. Secondly the material used has to be examined, to find out the necessary material data, as well as the limitations of the material. From this information several other decisions should become easier to make. Such as how to create the lattice structure of the material, as well as how to translate from the digital model to the 3D-printer.

In this project the printer used is mounted on a Robotic arm of the type *KUKA Quantec KR120* from *KUKA* with a reach of 2500mm and 6 different axes to be able to print elements in any sizes and from any directions needed. The printer itself consists of an extruder from *KFM Maskin i Sverige AB* of the type *ECO 3D 35*, with the different nozzle sizes varying from 2-12mm. The printer material, as earlier mentioned *Woodcompound 3D s50 Flex K*, has as well an impact on the 3D-printing due to its properties. According to the process instructions of the material the temperatures both of the extruder and the bed are of high importance for a good result, as well as the printing speed. The extrusion height becomes that of the nozzle diameter while the extrusion width becomes 30-50% higher than the nozzle diameter. The minimum and maximum printable angles are -45 and 45 degrees.

The Robotic arm is controlled by KUKAs own software, which exist as a component in Grasshopper. The input necessary are the paths for the printer to follow, as curves on planes. The planes needs to be sorted in the correct order, i.e. normally from the bottom up, for the printing process to work properly.

2.4.1 Lattice Structures

To create a printable structure a lot of focus lies on the inside structure of the elements or of the model as a whole. The material efficiency of this structure has a large effect on the overall material usage, and the distribution of material is also important for the printer itself. If material is printed to tight, and not with enough porosity, the heat inside the material risks becoming dangerously high, and a safety hazard. Due to these reasons the lattice structure is in itself a very important part of a 3D-printable model.

A lattice structure is typically a connected array of struts (Wu et al., 2019), and is the name of the structure filling the 3D-printed model, and thereby the carrying part of the material. However, in this project the material used for 3D-printing has a limitation of needing to be printed as surfaces, to be able to maintain its structural properties. This is due to the 3D-printer working in planes when printing, and therefore needing the surfaces of the lattices to interconnect the planes to each other. Therefore the lattice structure needs to be well thought through in order to create a working system in three dimensions.

Another important aspect of the material used is its orthotropic properties, meaning that its stiffness differ with different printing directions. This means that the lattice structure may have to be suitable for printing in several different directions as well as angles. Due to this, it could also be worth to consider the direction of the printed material, with regards to the directions of the stresses inside the model.

To create an as efficient lattice structure as possible it should also be adapted according to the resulting stress patterns from the Finite Element Analysis. Preferably there should be more materials, i.e. a tighter lattice structure where there are higher stress concentrations, and less material, i.e. a sparser lattice structure where there are lower. Additionally, it is also of importance for the structural capacity of the model to rotate the lattices in the right directions. In a structurally optimised model the lattice structure would follow the directions of the principal stresses, making full use of the material.

2.4.2 Translation for 3D-Printer

When the lattice structure is received it is in form of a 3D-mesh, constructed of vertices and faces. However, for the 3D-printer to be able to read the model, and print it, it is necessary to make a translation.

The movements of the printer are controlled by the movement of the robotic arm, which means that the 3D-model will need to be connected to the movement of the robotic arm. The KUKA robotic arms are controlled by the KUKA software, which includes a Grasshopper plug-in. The data necessary for the plug-in is in general the model divided into curves on planes, where the order of the planes are of highest importance as the printer will print them in their arranged order.
Therefore the resulting mesh from the Generative Design Process will need to be translated to curves on planes. The easiest way to do this is by finding the contours of the mesh in one, or several, specific directions and use these as paths for the printer. The distance between the contour planes needs to be adapted to the size of the printer nozzle, which should not be a problem as the size of the nozzle is known. The distance between the paths might however be a bit more tricky as the material

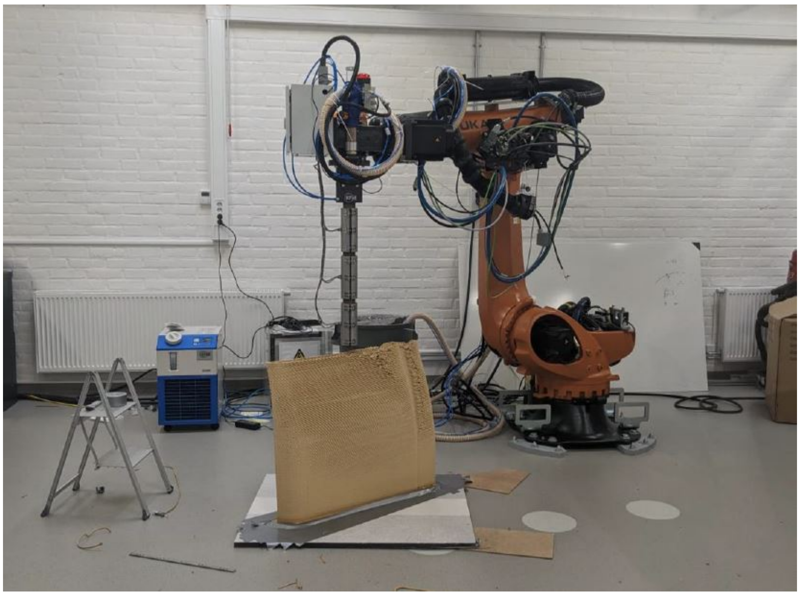


Figure 2.8: The Robot arm in action.

have a habit of floating out a bit and therefore taking up a bigger area than the size of the nozzle. This makes it hard to anticipate the distance necessary between the paths.

Another important matter when translating the mesh into curves on planes is the direction of the planes. The directions will have a large impact on the structural capacity of the model, especially as the material is orthotropic and have different strengths in different printing directions, but it also has a large impact on the printability of the model. If the model contains many protruding parts it might be smart to shift the directions of the planes for those parts, to make it easier to print them. This might, however, end with a rather complex puzzle of planes in the computer model, as the planes have to be in the correct order for them to be printed correctly.

3
Methods

The methods mainly focus on achieving the primary goal of the Grasshopper Plug-In, but also contains steps on additional development that might not be performed. The main method can be divided into a few steps:

Conduct Small-Scale Experiments: Carry out meticulous small-scale experiments to determine the material properties of the 3D-printed material while varying different parameters such as printing direction, loading direction, and moisture content. These experiments will help in understanding how the material behaves under different conditions.

Develop a Material Model: Construct a comprehensive material model from the results of the small-scale experiments that describes the strength and stiffness of the BCM in its orthotropic directions. This model will serve as a crucial foundation for the subsequent optimization process.

Create a Parametric Model for Optimization: Create a parametric model that enables iterative FEM-driven morphological structural optimization of structural elements. The objective of this method is to encourage the reinforcement of load-bearing domains, by the use of FEA in combination with Generative Design, thereby optimising the model and the material usage.

Translate the model for 3D-printing: Translate the resulting optimised model for the 3D-printer by changing the output data of the process into data suitable for the specific type of 3D-printing used for the relevant material. That means initially creating a structure that can be 3D-printed, and secondly assuring that the specific 3D-printer can read it and print it. In this part of the method a thorough understanding of both the material and the 3D-printer will be necessary for a successful result.

Create Grasshopper components: Create finalized Grasshopper components for the optimisation of the design. These components should be created alongside the earlier mentioned methods and used during the process for confirmation.

Showcase Different Use Cases: Demonstrate the applicability and versatility of the optimization tool by applying it to various use cases. These may include designing purely structural elements such as simple beams or columns, but also more complex, coherent structures, where the form-finding part of the program would really come into light.

Further development of the Parametric Model: When the earlier steps are finished, further development could be performed. Such as introducing flexibility into the parametric model, by allowing for morphological optimization that considers practical concerns like systemic compatibility, usability and general design concerns. This ensures that the resulting designs are not only structurally sound but also feasible for real-world construction. This will be implemented as weighted inputs, guiding the form-finding to include certain features, such as connectivity to other building elements, be the structural component of staircases or simply forcing a surface to be planar.

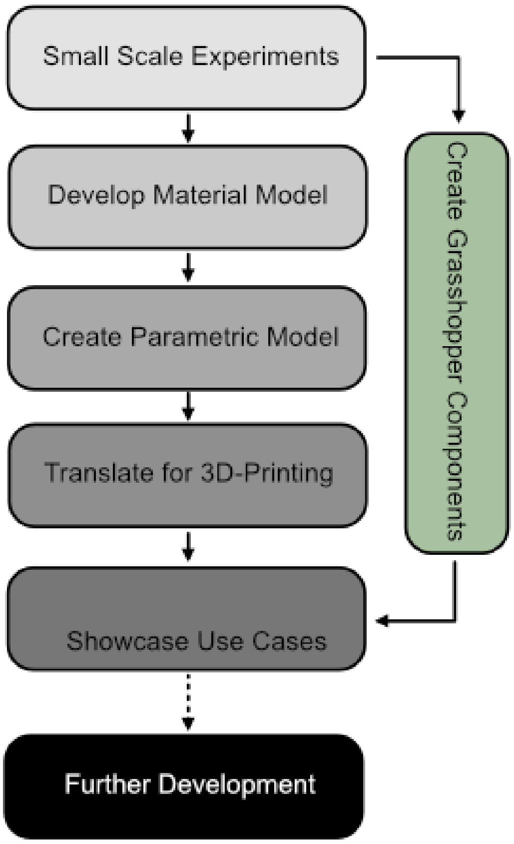


Figure 3.1: Flowchart of the general method.

3.1 Small Scale Experiments

Although material properties was received from the manufacturer of the BCM and this data could be used as a starting point for the project, small-scale experiments were still conducted to verify the properties of the printed material in a controlled environment. The initial experiments were performed as tensile strength tests by the interdisciplinary research team, to find the tensile strength as well as the modulus with different printing angles according to Figure 3.2.

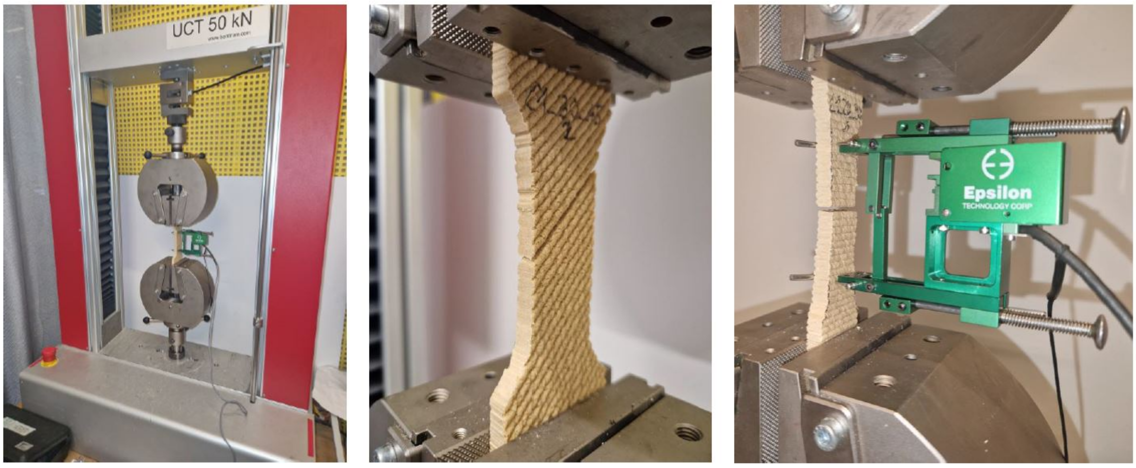


Figure 3.2: The set-up of the test for tensile strength.

The specimens that were tested were printed in three different directions to test the strength of the material with different printing directions. The printed directions of the material were 0 degrees angle, 45 degrees angle and 90 degrees angle, for the angle alpha as can be seen in Figure 3.3. Initially plans were made to perform more tests, such as for compression, shear, printing speed and moisture content. However, due to a lack of time, these tests had to be put on hold.

3.2 Implementing FEM

The Finite Element Method was used in a combination with the Generative Design as a method of finding the utilisation of different parts of the material. FEM was implemented according to the theory in Section 2.2 into several Grasshopper Components.

3.3 Grasshopper Components

As one of the goals of the thesis was to automate and simplify the use of BCM as a building material, an easy, accessible and well-integrated method to accomplish this was sought. Therefore it was decided to create the Parametric Model as a Grasshopper Component. Grasshopper is a Plug-In for Rhinoceros 3D built specifically for parametric design and is already well integrated as a tool in the building sector.

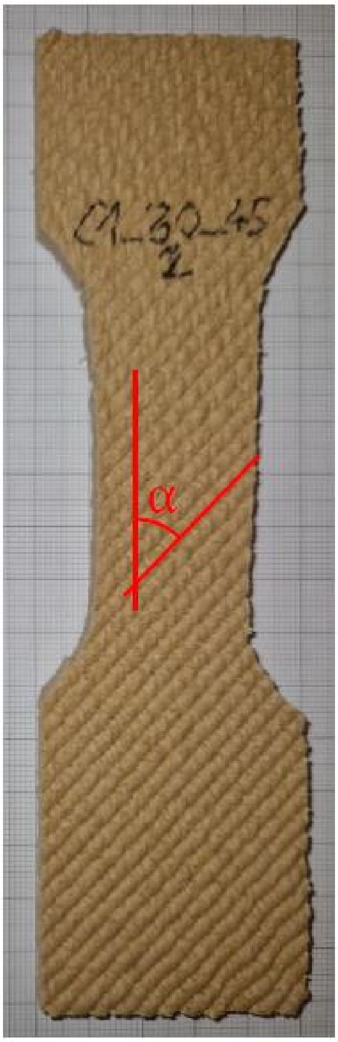


Figure 3.3: The small-scale experiments was performed for different angles of printing direction.

In Grasshopper the possibility of integrating code based components exist, meaning that the component can be written in any of the most common programming languages and easily incorporated into the program.

As the Parametric Model in this thesis was based on the Finite Element Method, the structure of the Grasshopper component had to follow a certain template. For simplicity the component was divided into several parts regarding the different steps of the FEM. The main idea was for these part to focus on the following areas:

Loads: The different loads of the system are incorporated, with the input of size, direction, position and type.

Supports: The different supports of the system are incorporated, with the input of position and type.

Material: The material properties are incorporated as Young's Modulus, Shear Modulus, Poisson's ratio and density are incorporated, as well as the type of material as Isotropic or Orthotropic.

Model: Creates the 3D-mesh built up from the different elements, with the input of the boundary conditions as well as element size.

Solver: The component that does all the hard work, solving the system with the Finite Element Method.

Iterative Solver: Solves the system in the same way as the normal solver, but has the possibility to iterate towards an optimized solution with Generative Design.

Visualization: A component for visualizing the results of the calculations.

3D-Printing: The final part of the component, with the job of transforming the optimized design from the Iterative Solver into a 3D-printable model.

3.4 C# Scripting

The Generative design process can be incorporated using a number of tools, in this thesis the tools being used is Grasshopper in combination with C# scripting. As Grasshopper is a naturally parametric tool with a friendly interface directly connected to a 3D-model in Rhinoceros 3D it suits the project perfectly. In combination with C#, that simplifies the iteration and computation process and is easily incorporated into Grasshopper, a generative design process can be established.

To write a Finite Element Analysis tool for Grasshopper in C# a clear plan over the necessary methods are needed. The code should be easy to follow and should focus on keeping the computation time as low as possible. The different methods will be divided over the several Grasshopper components presented above, dealing with different parts of the problem. The different components have different demands on them, and will be in need of different methods inside them as well as different types of component attributes. More specifically these components should be constructed according to the following system:

Load Component. A component dealing with the input of loads. This component should read a geometry for the location of the load, as well as a vector representing the size and direction of the load as input, and should convert these into load data that will be plugged into the Model Component.

Support Component. A component dealing with the input of boundary conditions. This component should read a geometry for the location of the support and should contain buttons where the user can specify in which directions the support is locked. It should provide a support data as output to be plugged into the Model Component.

Material Component. A component dealing with the input of material properties. This component should read the necessary Material Properties as input for custom materials and should have an easily accessible way for the user to choose material properties of already existing materials. It should also provide the possibility for the user to choose a region of the model to apply the material to, to be able to apply several materials to a model if necessary. The component should provide the necessary material data as output to be plugged into the Model Component.

Model Component. A component constructing the model. This component should take Load Data, Support Data, Material Data and Element Size as input and from this create an automated 3D-Mesh model with the necessary size to accommodate all the loads and supports. The model should be constructed of hexahedral elements with eight nodes each and with the loads and supports properly incorporated to the correct nodes. It should also provide the user with the possibility to insert a B-rep as a starting model if automation is not necessary. The component should provide all data necessary for FEM calculations as output to be plugged into the FEM-Solver Component

FEM-Solver Component. A component solving the problem through FEM. This component should read the previously mentioned Model Data as input and provide a Finite Element Analysis, solving the deformations and stresses for the model according to the theory presented in Section 2.2. It should provide new Model Data as output to be plugged into either the Visualize Component or the Post-Processing Component.

Iterative FEM-Solver Component. A component solving the problem generatively by iterating through FEM solutions. This component should work with the same input and calculations as the FEM-Solver Component, but should in its calculations work iteratively following the theory presented in Section 2.3 and provide outputs ordered in the different generations to be plugged into either the Visualize Component or the Post-Processing Component.

Visualize Component. A component for visualizing the results. This component should provide the user with the possibility to easily visualize the different possible results from the FEA, such as stresses in different directions and deformations. For the Iterative FEM-Solver Component it should also provide the possibility for the user to easily flip through results of the different generations.

*Post-Processing Component.*A component for post-processing and translation for the 3D-printer. This component should translate the final Model Data of the Iterative FEM-Solver Component into a 3D-printable model, by creating a lattice structure suitable for the resulting model and translating it for the 3D-printer.

3.5 Goal Searching Algorithms

The process used to find the optimised solutions in this project is a combined outcome of the theory presented in Section 2.3. The process is focused on an iterative FEM-solution that in combination with topology optimisation, filters and failure criteria should be able to find the optimised solution for any boundary conditions. These combined methods can be called Goal Searching Algorithms, with the common goal of finding the optimised solution.

The underlying principle of this proposed algorithm involves conducting a Finite Element Analysis across the entire domain in each iteration. By utilizing stress patterns as a navigational tool for the subsequent iteration, elements should be identified: those experiencing greater loads receive increased stiffness values, approaching but never surpassing the material capacity limits. Conversely, elements enduring minimal stress are either diminished or removed. This sequence iterates until the cumulative energy within each iteration within the system attains a predetermined threshold. The anticipated outcome is the emergence of a structure characterized by elements predominantly utilized close to the threshold.

For each iteration of the model, each element is assigned a fitness value with respect to its utilisation value gathered from the selected failure criteria, this may include a combination of failure mode or printing restrictions etc. A smoothing filter is then applied to all fitness-values such that each fitness value is a weighted average of neighbouring values. This ensures that deactivated elements may be activated for the next iteration if necessary, allowing for the design to grow into previously deactivated areas.

The presented method differs to more common implementations of topology optimisations, which generally seeks to lower the total volume to a certain volume threshold. Once the solution has reached that volume the total volume is adjusted to exactly retain that value. The presented method is based on the normal distribution of fitness values across the total domain, seeking to adjust the average utilisation of the design to match a desired utilisation threshold instead of a volume fraction. A normal distribution is calculated from all fitness-values and a threshold value is introduced which determines what elements should have their respective material density penalised or favoured.

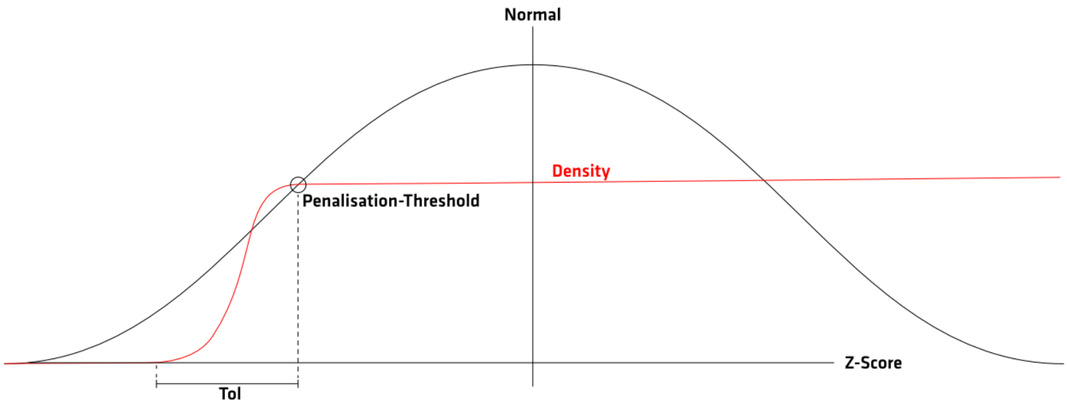


Figure 3.4: Normal distribution curve for penalisation.

The density (De) of an element with a fitness-value Z-Score (ZSe) below the threshold (T) is penalised by the penalisation coefficient (P) such that:

$$newD_e = \frac{D_e + P(a)}{2} \qquad a = ZS_e - T \tag{3.1}$$

Where newDe is the new element density value and P(a) varies according to Equation 3.2.

$$P(a) = \left\{ \begin{array}{c} 1 \quad when \rightarrow a \geq 0 \\ 0 \quad when \rightarrow |a| < T \\ \frac{cos(\frac{a\pi}{Tol})}{2} + 0.5 \quad when \rightarrow (a < 0) \wedge (|a| < Tol) \end{array} \right\} \tag{3.2}$$

The optimisation process aims to minimize the total mass change (E)

$$E = D_e - newD_e \tag{3.3}$$

When the strategy is applied to a cantilever beam, with an arbitrary static Z-Score threshold, the results become a density field that consist of primarily binary solid-void elements that has converged to a certain volume threshold defined by T, as can be seen in Figure 3.5.

(a) Iteration 1 **(b)** Iteration 5 **(c)** Iteration 10

Figure 3.5: Iterations of a cantilever beam with a static Z-Score threshold.

To introduce load and material-dependency the method changes the penalisation threshold according to:

$$T = T_0 + dT \tag{3.4}$$

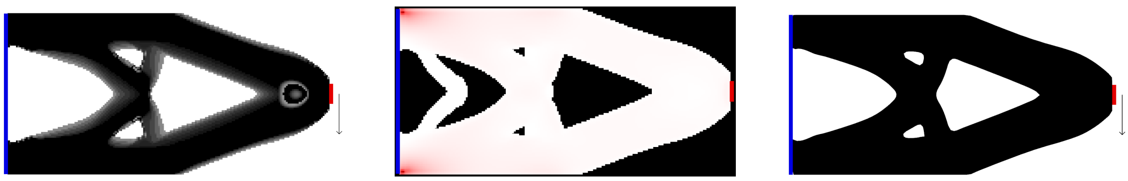
$$dT = dU * H(E, k) * S \tag{3.5}$$

$$dU = \frac{gU}{gD} - tU \tag{3.6}$$

$$H(x) = 1 + e^{(-2kx)^{-1}} \tag{3.7}$$

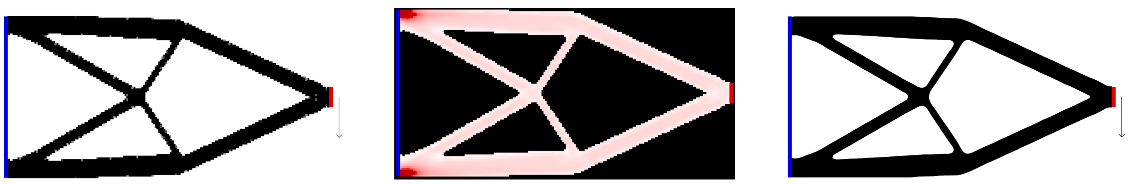
Where T0 is the previous iteration penalisation threshold, gU is the global sum of the element utilisation values, gD is the global sum of element densities and tU is the target utilisation value. S is a rate of change coefficient which determines how aggressive the target utilisation optimisation is allowed to be. H is a Heaviside function with the purpose of dampening the system if the energy, i.e. the total mass change, is high enough to risk numerical instability and lose the ability for convergence and k is here a dampening constant.

With the variable penalisation threshold implemented for the cantilever problem the method can iterate towards a better optimised solution, as can be seen in Figures 3.6 and 3.7.



(a) Iteration 15A **(b)** Iteration 15B **(c)** Iteration 15C

Figure 3.6: Iteration 15 of a cantilever beam with a dynamic Z-Score threshold.

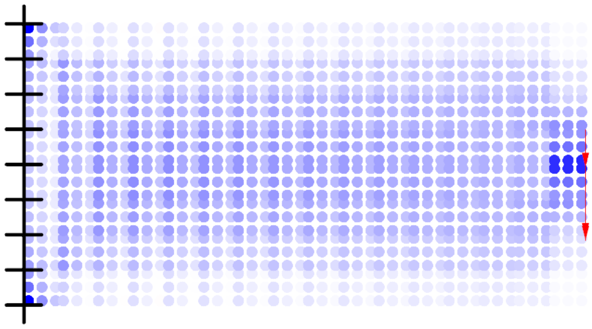


(a) Iteration 30A **(b)** Iteration 30B **(c)** Iteration 30C

Figure 3.7: Iteration 30 of a cantilever beam with a dynamic Z-Score threshold.

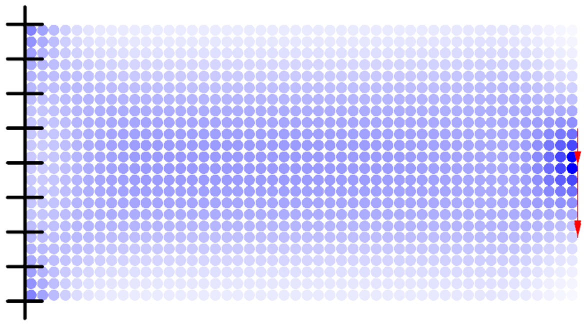
When using this method it is very clear that the most time consuming part is the FEM-calculations performed on the elements, if fewer elements are used, the calculations would be much quicker and more iterations could be performed in the same time frame. However, if fewer elements are used the resolution of the design would also become lower, and thereby the details of the resulting optimised solution. To be able to find a solution with a decent design resolution while still having a quite low model element resolution, i.e. a low number of elements, for quick iterations, the method would have to be updated.

An attempt to use the values at the integration points were used, using these stresses and densities as the design density field. The element model used, using 27 integration points with only 8 nodes and 3 degrees of freedom the element is over integrated, resulting in problematic shear behaviours. To overcome this problem an increased blurring radius was temporary tested. Moving forward, 8 integration points are used.



(a) Bad shear behavior

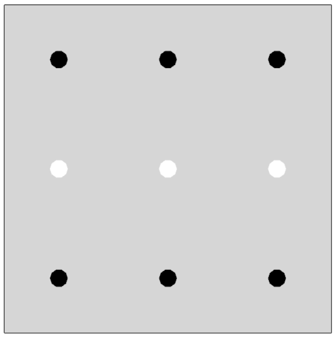
Figure 3.8: Bad shear behavior



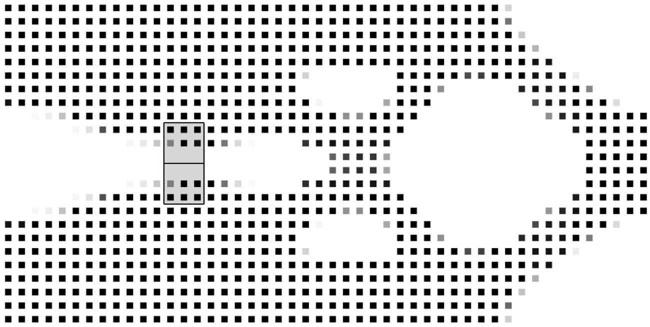
(a) Remedied using a very large blurring radius

Figure 3.9: Remedied using a very large blurring radius

However, issues arise that are related to how the increased design resolution is connected to the nodes, possibly resulting in designs within an element that would not in reality be able to transfer load in the desired directions.



(a) Row of zero-density values splitting the element



(b) Resulting design with problematic elements

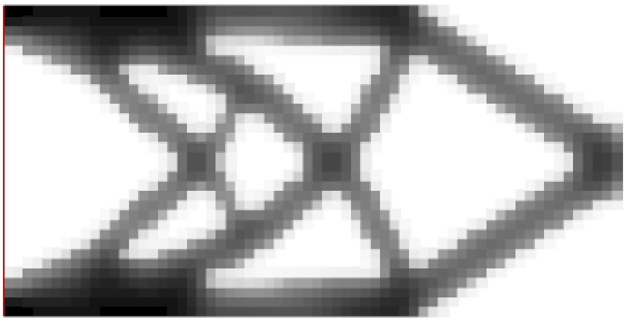
One way to help counteract this problem is to simply increase the radius during the smoothing phase, assuming the smoothing is a simple weighted average. However, this emphasizes issues related to values above the threshold in areas with high density contrast will be substantially lowered. This leads to the algorithm treating thin elements of the design, such as rods, as having a lower fitness value due to nearby empty elements, resulting in two major issues. Either the utilisation of thin elements exceed the desired utilisation, or thin elements are removed. A rudimentary solution is to introduce the density value as a parameter during the filtering phase, such that the density of each nearby data point affects its contribution to the resulting weighted average. This ensures that fitness values with a higher respective density relative to its neighbours has a higher priority during smoothing.

While increasing the radius is a solution, introducing directions to the smoothing algorithm results in a more precise manipulation of the desired outcome. The goal of such a smoothing functions is to allow for scaled smoothing along the principal stress directions, A few different weight functions were tested , the two most promising describe an ellipsoid, scaled along the principal directions and an interpolation between three different ellipsoids, each representing a sphere stretched along the planes. However, even implemented using the GPU, such operations increased calculation time considerably and were not pursued further.

A new approach was taken where instead of using the stresses at the Gauss-points as the design fitness values, the strains are smoothly projected to the design voxels where stresses are then calculated. The idea is to treat the design voxels as the definition for a smooth density field that the FEA elements can approximate. The density field is evaluated at the integration points of the FEA element, and the calculated strains define a strain field. To calculate stresses at an arbitrary point, both the density field and strain field are evaluated where the attained density guides the stiffness. This results in a stress field that can be evaluated to drive the optimisation process previously described. The primary benefit of this approach compared to previous methods is that a smoothing filter of the fitness values is no longer needed, thus no implicit treatment for thin elements is needed when the stresses are too small. Lower loads simply leads to a more porous solution.

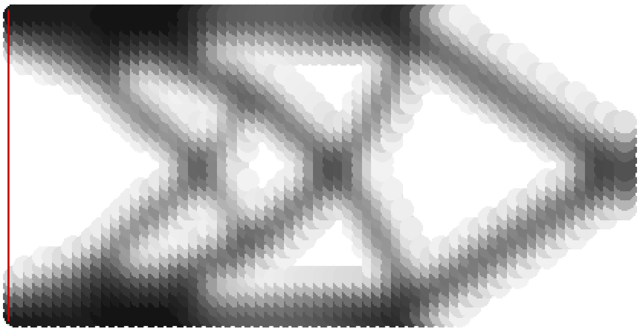
3.5.1 Utilization Criteria

Important for the goal searching algorithms is as mentioned also the failure criteria used to find the utilisation value of each element. Several different tests were conducted with the purpose of comparing the results for different failure criteria on the same model. As can be seen in Figures 3.8-3.9, the difference between using the Von Mises failure criteria in comparison with using the Principal stresses can lead to quite different results, however, both are viable optimised solutions, and thereby viable methods to use in the goal searching process.
It is important to note that several different failure criteria can be usable for the same load case, and that preferably several of them should be incorporated into the Grasshopper Components. Different failure criteria might be good for different materials.



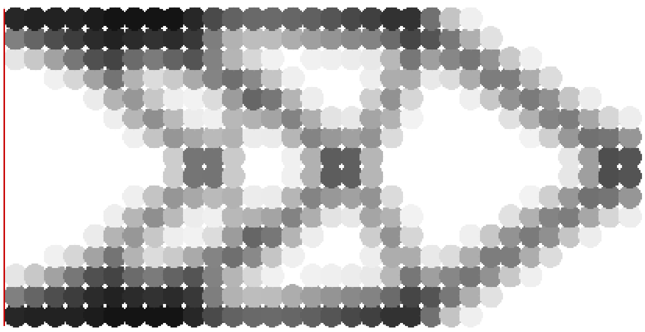
(c) Design values

Figure 3.10: Design density values



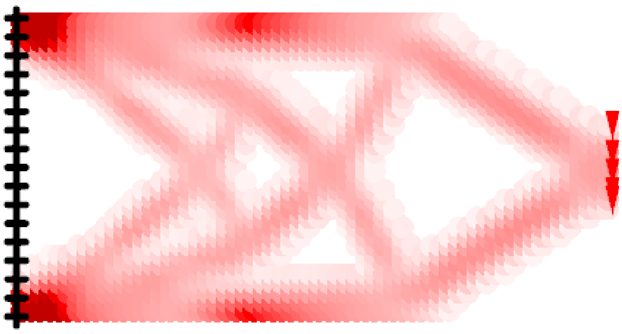
(a) Resulting density field

Figure 3.11: Resulting density field



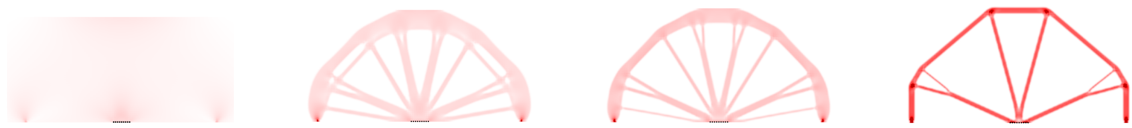
(a) Density values used for FEA

Figure 3.12: Density values used for FEA



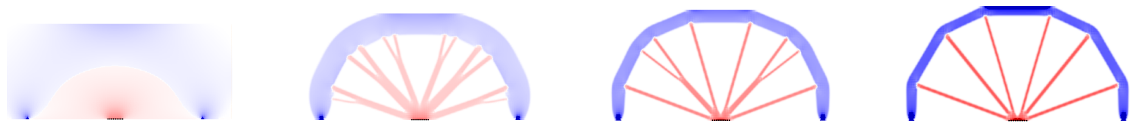
(a) Stress field displayed as Von Mises stresses

Figure 3.13: Stress field displayed as Von Mises stresses



(a) Iteration A (b) Iteration B (c) Iteration C (d) Iteration D

Figure 3.14: Using Von Mises failure criteria for utilisation.



(a) Iteration A (b) Iteration B (c) Iteration C (d) Iteration D

Figure 3.15: Using Principal Stress for utilisation.

3.6 Post Processing

When a final model has been created through the optimisation process it has to be post processed and prepared for 3D-Printing. This Post Processing consist of several important steps which can be quickly described as Meshing, Lattice Creation and Translation for 3D-Printing, and can be seen as simple visualizations in Figure 3.10.

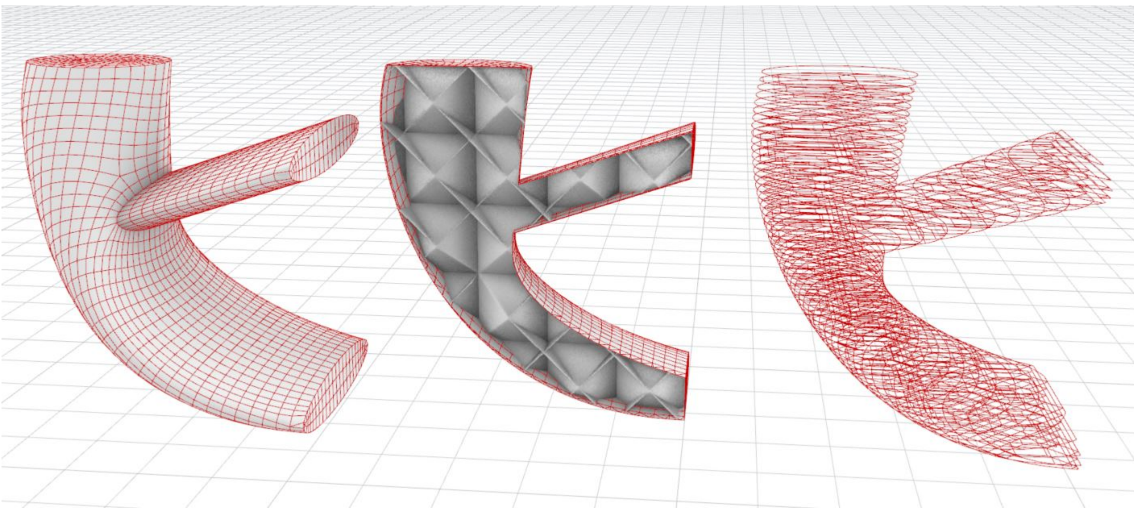
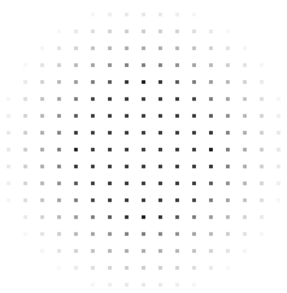


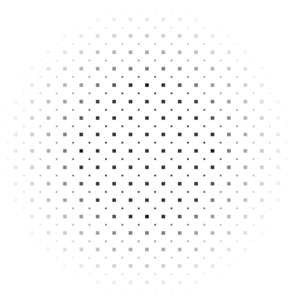
Figure 3.16: The three steps of Post Processing: Meshing, Lattice Creation and Translation for 3D-Printing.

3.6.1 Meshing

The goal of the meshing process is to be able to approximate an arbitrary density field into a mesh. The method initializes a 3D-grid of points over the entire design domain, each point is assigned a density value in the domain of 0 to 1. The strategy used utilizes the fact that the density field has values evenly spaced in 3D-space, treating them as voxels(pixels in space) with values. The initialized grid represents nodes connecting adjacent voxels, with values being an average of all voxel-values attached to the respective node. The values can also be a weighted average of known field values within a specified distance, thus eliminating the dependency of the field resolution.

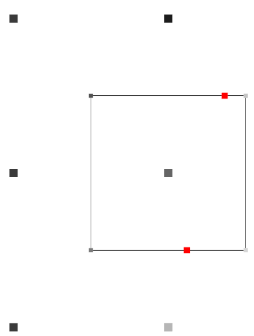


(a) Density field



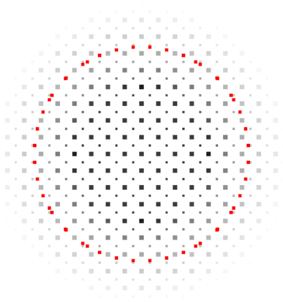
(b) Nodal Density values

With eight nodes and values for each voxel, the meshing procedure starts. Evaluating two nodes representing an edge of a voxel, the goal is to find the interpolated coordinate such that the interpolated value equals a certain clipping-threshold, indicating the point to cut the edge. When applied to all ten edges a set of points are determined, if the set has zero-length no edges were cut. Using these points, depending on how many points, a set of triangular and/or quad-faces are constructed. It is important to note that the voxel may represent more than one surface and a strategy to handle these rare cases is needed. What was implemented is based on the connectivity of the nodes, all nodes above the threshold are grouped into sub-groups such that each sub-group is connected by nodes above the threshold, resulting in each sub-group representing different disjoint faces.

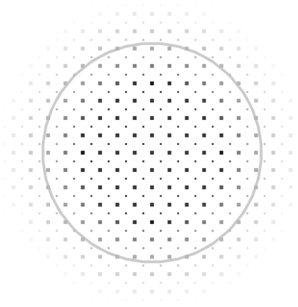


(a) Individual Voxel

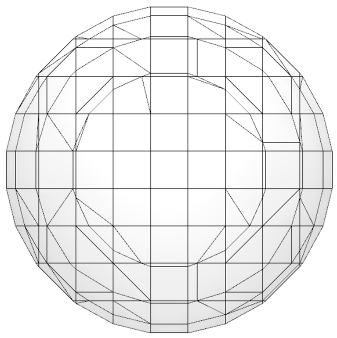
This strategy is applied to all edges for each voxel, resulting in a mesh that represent the density field at a certain density threshold. Cube marching greatly reduces the computation time of the procedure as it only evaluates voxels connected to a previously cut voxel. Finally, Grasshoppers Quad-Remesh is used to reduce the number of vertices and faces of the mesh and to give a smoother design. As this final step is relatively computationally time-consuming, it is not automatically applied.



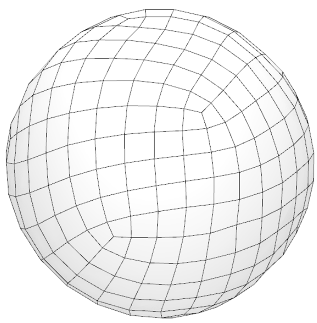
(a) Intersection points with a clipping threshold of 0.5



(b) Section of resulting mesh



(a) Mesh from intersection points



(b) The Quad Re-meshed result

The treatment of boundaries during all smoothing and interpolation procedures result in vastly different results. Using only values within the design domain favours boundary values, giving them a higher weight resulting in a solution that seemingly sticks to the boundary of the domain. One solution is to simply increase the domain but only do simulations within the original domain. The solution presented simply includes fake values outside the boundary, values containing a density of zero.

3.6.2 Lattice Structures

Given a density field as a solution, containing areas of intermediate density, ie values between 0 and 1, the goal is to find an appropiate representation of said field where all density values are either 1 or 0, solid or void.
One method is to subdivide the density field into a set of boxes, each representing a volume of a certain density. Defining a set of tile-able representations and mapping each representation to a density interval the design can be filled. Once filled, the box representation can be split and joined by the outer mesh.

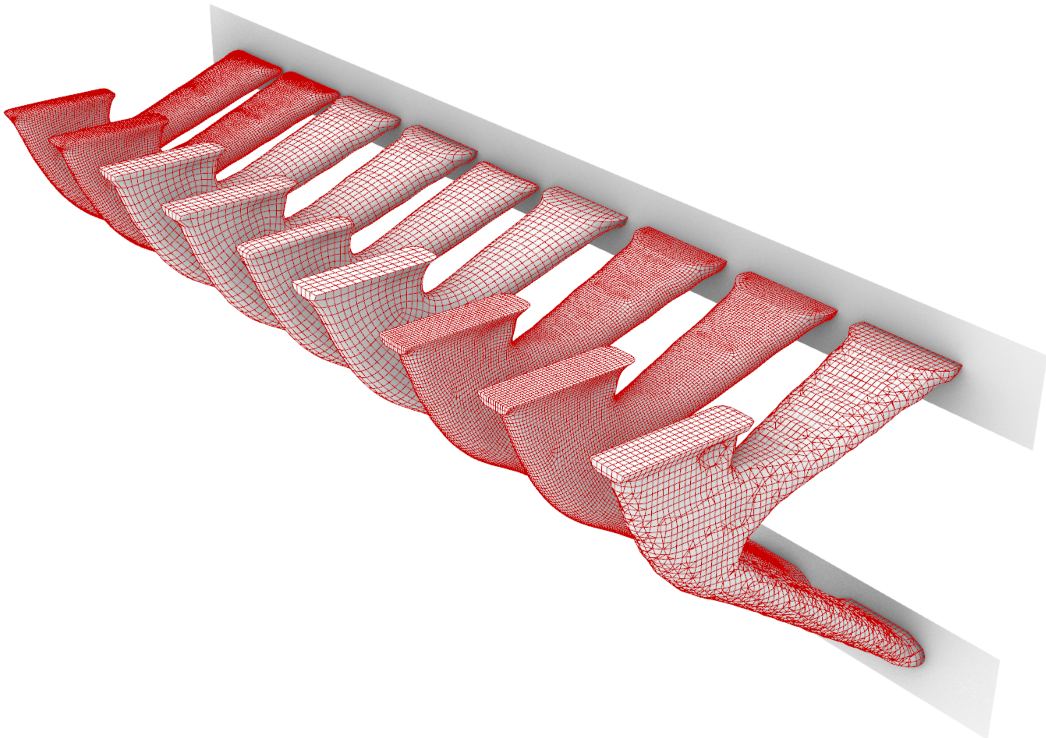
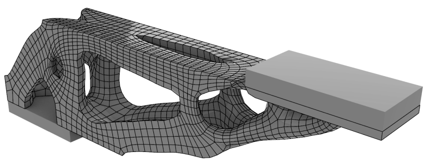
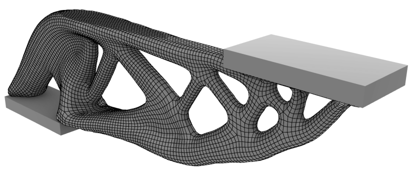


Figure 3.17: Comparison between different re-mesh settings. Right-most one is with no quad re-meshing



(a) No boundary treatment



(b) Treating the domain outside the design domain as void elements, except if the boundary is at a load or support surface

Figure 3.18: Different designs of the boundaries.

A simple example visualized below is to represent each box with pyramid-shaped structures, where four pyramids build up one element, with their tops meeting in the centre as can be seen in Figure 3.13. This structure creates surfaces that can transfer loads in the three main planes of the hexahedral, and connects the element with the elements around it.

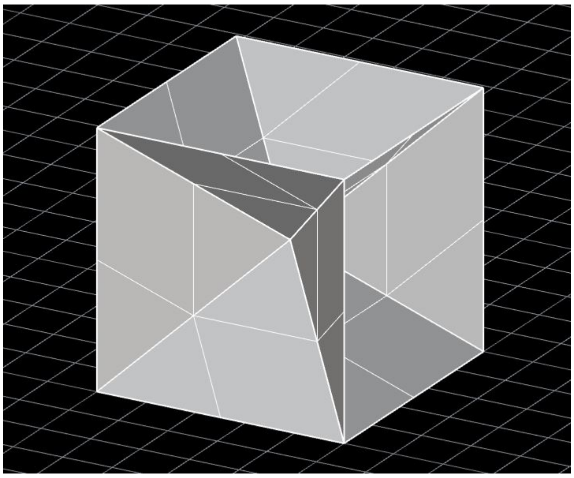
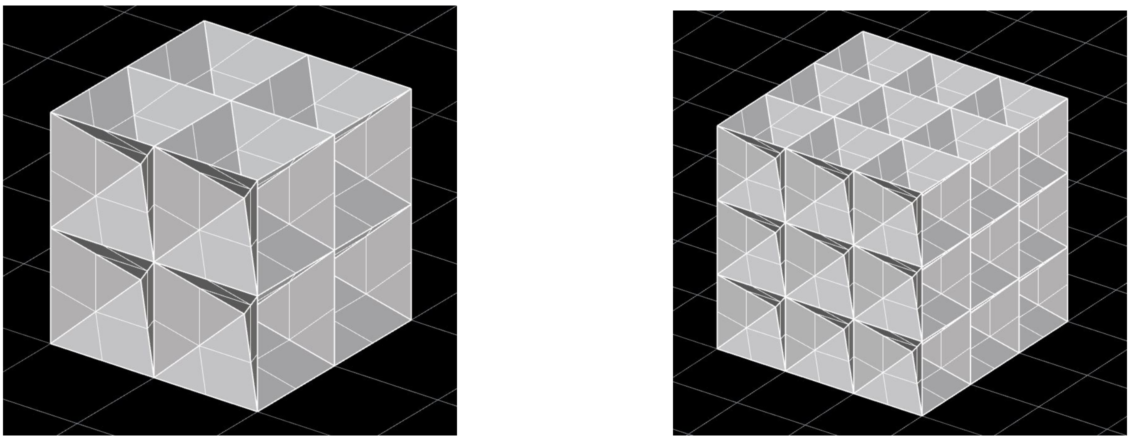


Figure 3.19: A single cell of a simple surface lattice structure.

Depending on the strength of the material as well as the size of the hexahedral elements the element can also be divided into more pyramid-structures as can be seen in Figure 3.14.



(a) Twice as dense form.

(b) Three times as dense form.

Figure 3.20: Denser forms of simple surface lattice structures.

This is a very simple way of filling a model with a working lattice structure, although it might not be the best solution. Firstly there might exist better solutions to fill the hexahedral elements, such as Schwarz Primitive, Gyroid, Schwarz Diamond and Neovius as presented in Figure 3.15 (Kladovasilakis et al., 2022). The advantage of these methods might be that they transfer loads more efficiently between the elements, especially if the loads are shifted from the Cartesian planes.

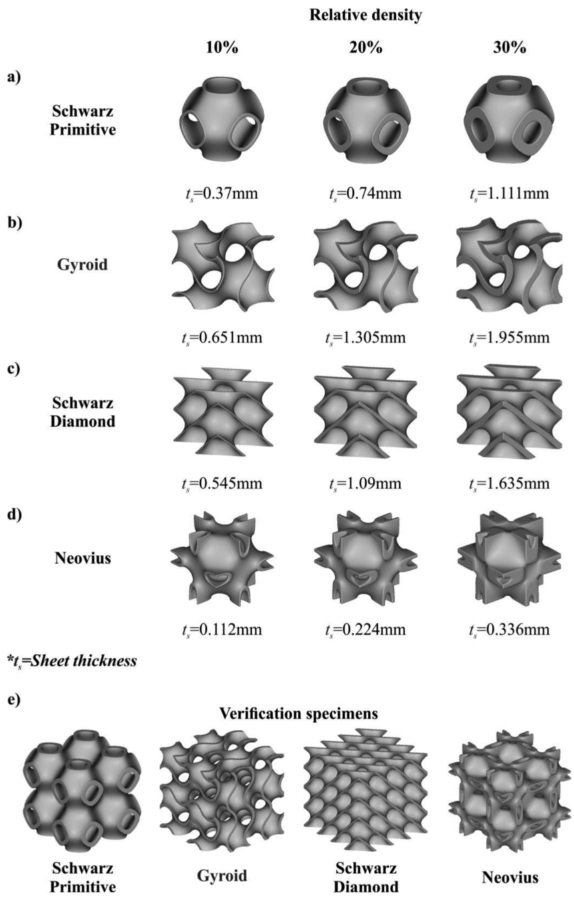


Figure 3.21: Examples of more complex surface lattice structures (Kladovasilakis et al., 2022)

If the focus is switched from trying to fill the hexahedral elements with a certain structure and instead trying to fill the entire model with lattices as one whole structure even more solutions exist. Through the use of Voronoi-Tesselation a lattice grid can be found that covers the entire model, and that takes the porosity of the different elements into consideration to create a varying grid throughout the structure (Lei, et al., 2020) & (Piros & Trautmann, 2023).

In the end, the optimal way to create a lattice structure in a model is probably to try and follow the principal stress pattern. This can in theory be done both by filling the entire model with a lattice structure following the stress patterns, or by arranging the elements after the stress patterns and then fill them with one of the structures suggested above.

An efficient method of how to create a lattice structure that follows the principal stress fields is presented in "Design and Optimization of Conforming Lattice Structures." (Wu et al., 2019), where the shape of the finite elements are rotated and moved through the process, being guided by the principal stress field, and used as lattice structure in the final stage of the process. To use this method, however, the generative design process has to be based on rotating and moving the elements. In this project the method did not involve changing any elements, only remove or diminish their stiffness, which ruled out this method of creating lattices. The method could although still be used as an inspiration for the lattice structure in this project.

Another method that might be used when creating a lattice structure is presented in *Homogenization-based topology optimization for high-resolution manufacturable microstructures* (Groen & Sigmund, 2017), although for 2-dimensional problems but the method should still be somewhat applicable for 3-dimensions as well. The method takes its basis in creating a function describing the principal stress curves in the three directions, and then uses certain filters to turn the functions into a hexahedral grid. This grid can then be filled and used as a lattice structure.

The third, and maybe most applicable method can be found in *Phasor noise for dehomogenisation in 2D multiscale topology optimisation* (Bærentzen et. al, 2023) which focuses on smooth transitions between shifted phasor kernals through interpolation. This method could be used as a basis for the interpolation between the vectors in the different nodes of the resulting model in this project.

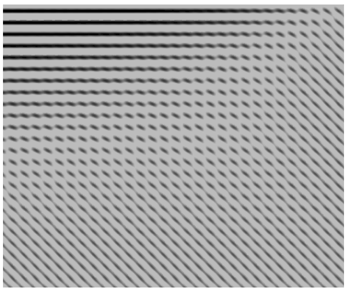
The method we propose is a simplified variation of the method proposed by Bærentzen applied to three dimensions. The idea is to represent the design field with a set of kernels. Each kernel is defined by a position and three vectors in 3D space, defined by the three principal stress directions at the position and three density values that are related to the utilisation of the material along each principal stress direction and an influence radius.
The design field that the kernels represent is a weighted average of the response of all nearby kernels. The response of each kernel is the maximum response each kernels direction. The response of each direction is defined as following:

$$R = Cos(Dot(dir, vec)/waveLength) \tag{3.8}$$

And its derivative as:

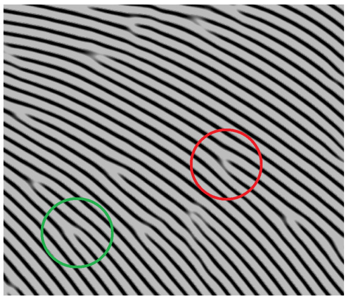
$$dR = -Sin(Dot(dir, vec)/waveLength) \tag{3.9}$$

Dir is the kernel direction and vec is the direction from the point to evaluate to the kernel position.
Each kernel uses the derivative of adjacent kernels to align itself with them, the goal is to align its influence with adjacent kernels such that the destructive interference of adjacent kernels is minimized. This alignment procedure is iterative and results in a phasor noise field aligned with the principal stress directions.



(a) Lattice A

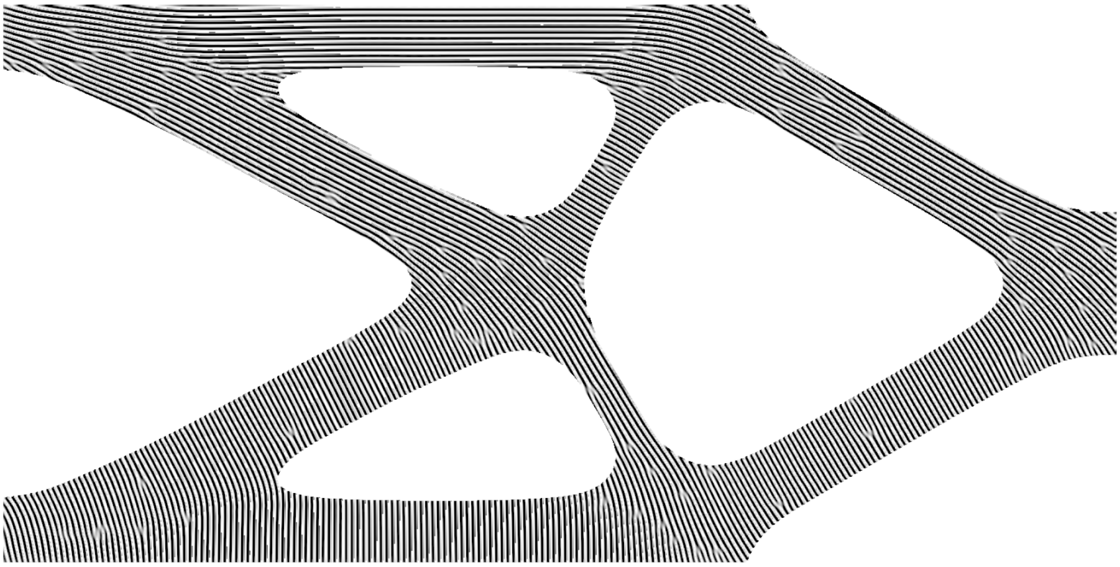
Figure 3.22: Kernel density field



(a) Lattice B

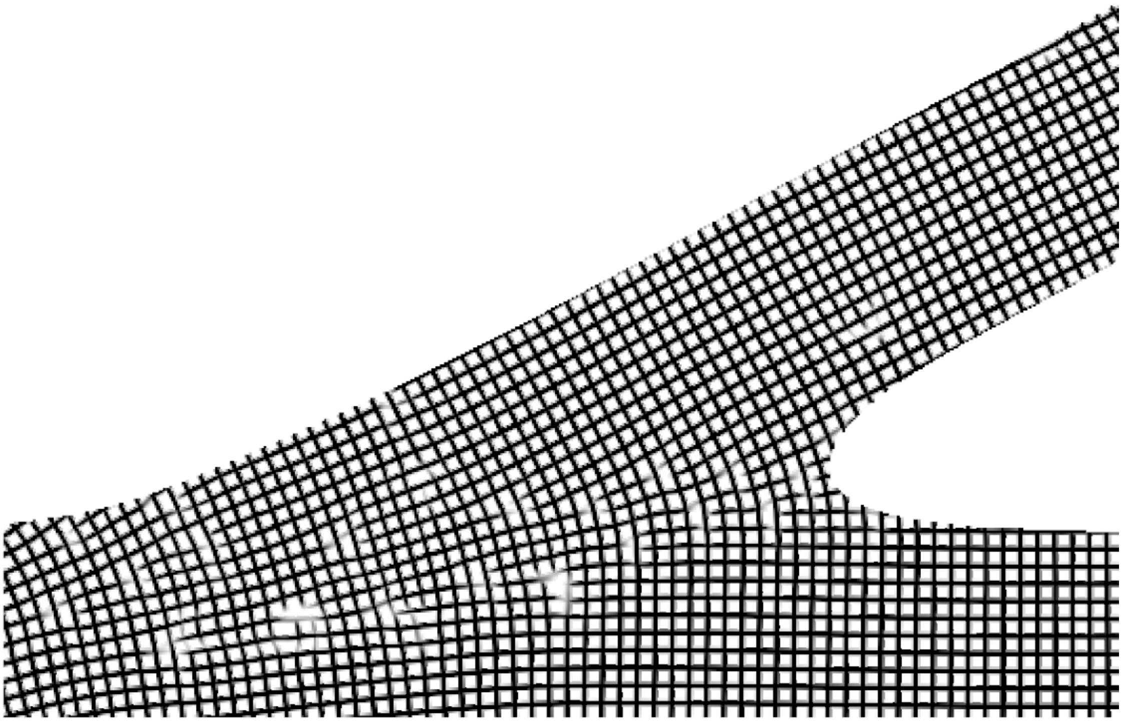
Figure 3.23: Aligned kernel density field

Given that the tensile capacity of the printed material is order of magnitudes stronger along the printing direction compared to perpendicular to it, using this field to guide the 3D-printing direction, using the printing layer thickness as the wave length, is a reasonable solution on how to fill solid volumes. More tests is needed in regards to what settings leads to the most uniform distribution of spacing. As bad spacing is highly localized it seems possible to adjust the settings such that there are no connected forks, to make sure there is enough space for the extruded material. Example of desired fork is marked in green and undesired marked in red.
Representing the Kernel density field with voxels is highly computationally expensive and thus a meshing strategy is proposed. The idea is to approximate important



(a) Lattice A

Figure 3.24: Filling the entire solution with solid density

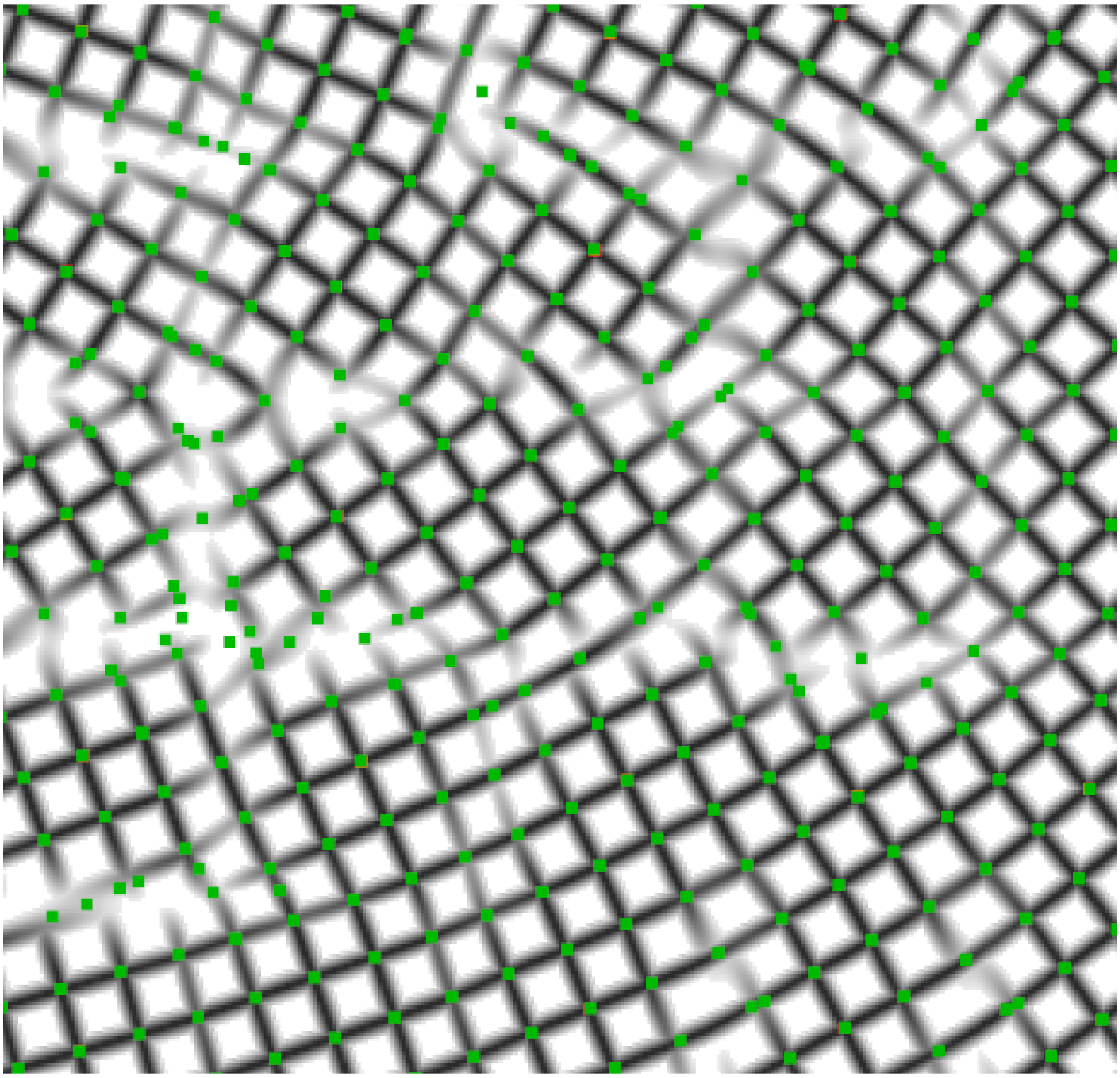


(a) Lattice A

Figure 3.25: Zoomed in on combining the response of both tension and compression

points, use the gradient of the field to move them to local maximums and then connect adjacent points into a mesh using the dot product of the vector to adjacent points and the principal stress direction.

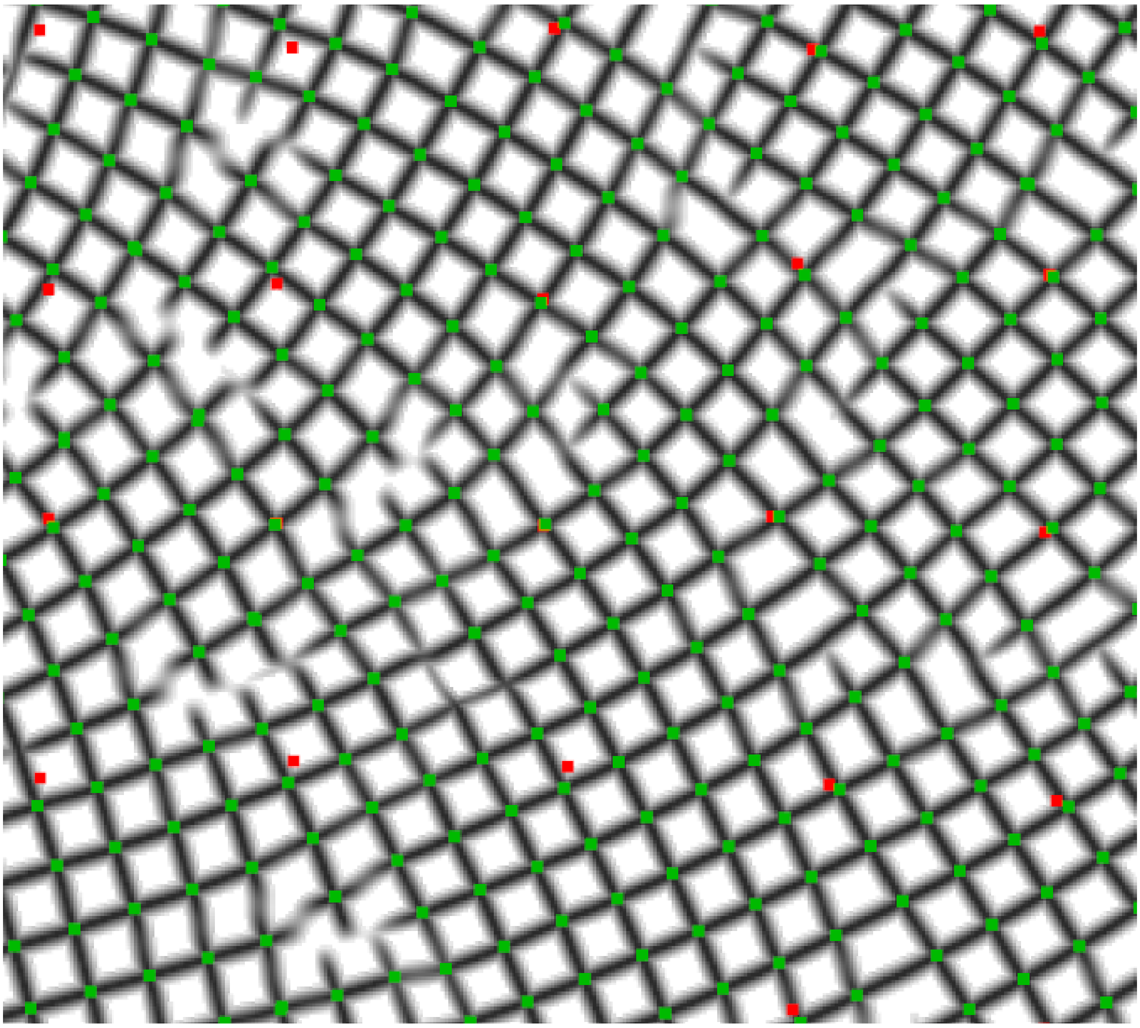
The initial test nodes are taken as the grid centered at the kernel position, with a spacing of the wave length. The points are then moved using the gradient of the kernel field with the goal of finding nodes that connect the cosine waves in all three direction.



(a) Lattice A

Figure 3.26: Initial approximated nodes, moved to find the local maxima of the Kernel field

Using these points a new set of kernels is created defining a field with a higher definition. During the alignment process all new kernels that are within a certain tolerance are merged. These points will later be used to build the mesh.



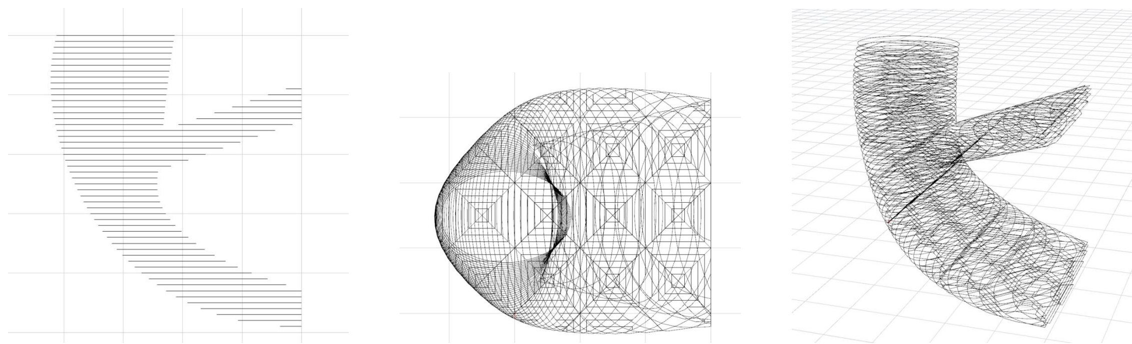
(a) Lattice A

Figure 3.27: Positions of new Kernels on the smaller scale

3.6.3 Translation for 3D-Printer

As mentioned in the theory section, the translation for the 3D-Printer can be very simple, as well as very complex depending on which level of optimization is sought. As the 3D-Printer works by following paths consisting of curves on planes, the simplest method to create a working geometry for the 3D-Printer is by working with contours.

The paths needed for the 3D-Printing can be easily found by contouring the finalized lattice structure, as well as the outer shell mesh, with a distance in the Z-direction fitting for the selected nozzle diameter. Thereby several horizontal planes, with path curves, will be created over the entire model, sorted from the bottom up. If, however, an even more optimized structure is sought, a more complex method would be necessary. It is in theory possible to rotate the planes used for the contour so that they are directed towards the principal stresses, and thereby using the optimized direction of the material.



(a) Contours from Side view **(b)** Contours from Top view **(c)** Contours from Perspective view

Figure 3.28: Printing paths of a model as contours sorted in planes.

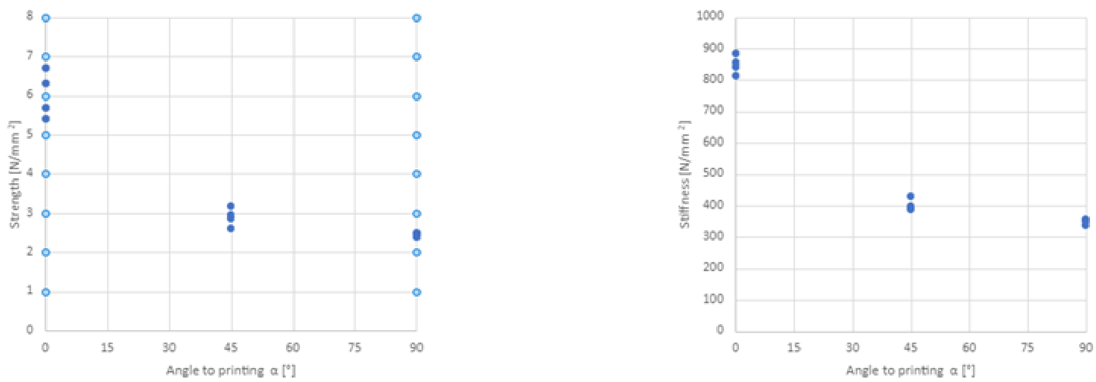
4

Results

4.1 Material Properties

The small-scale experiments were performed by the research group Lightweight Structures at Chalmers before the start of this thesis. The tests that were performed was simple tensile tests to find the tensile strength, as well as the tensile modulus, for different directions of printing the material. The tests were performed on several specimens with varying sizes and the results can be seen in Figures 4.1-4.2.

The original plan of the project was to perform tests to find all material properties necessary for the finite element analysis, however, as time ran out these tests were put on hold and in the end it was decided that the parameters would be found from the performed tests in combination with general assumptions.



(a) Strength depending on printing angle.

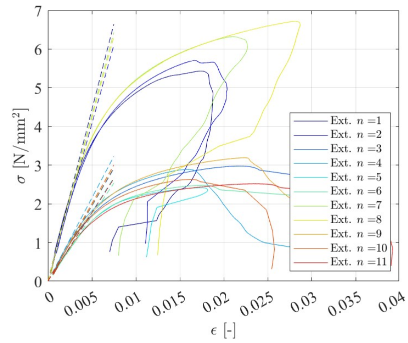
(b) Stiffness depending of printing angle.

Figure 4.1: Results from the testing of the material.

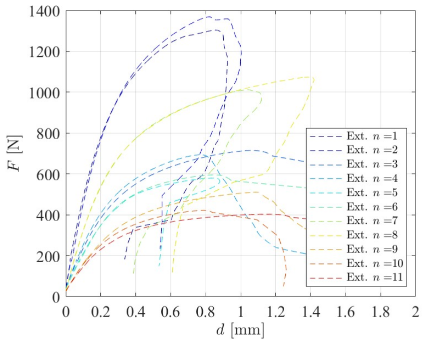
As the material tests were performed only in tension as well as only in one plane certain assumptions was made regarding the material data to find all properties necessary for the Finite Element Analysis. The results from the tests were for each angle of printing as follows:

$$E_0 = 850N/mm^2 \quad E_{45} = 400N/mm^2 \quad E_{90} = 350N/mm^2 \tag{4.1}$$

$$\sigma_0 = 6.0N/mm^2 \quad \sigma_{45} = 3.0N/mm^2 \quad \sigma_{90} = 2.5N/mm^2 \tag{4.2}$$



(a) Stress-strain relationship.



(b) Strength-thickness relationship.

Figure 4.2: Graphs from the testing of the material.

As the 3D-printer works by adding material in layers both in height and depth the x-direction of the material was assumed to be in 0 degrees, while both the y-direction and the z-direction were assumed to be in 90 degrees. This resulted in:

$$E_x = E_0 \quad E_y = E_{90} \quad E_z = E_{90} \tag{4.3}$$

$$F_{tx} = \sigma_0 \quad F_{ty} = \sigma_{90} \quad F_{tz} = \sigma_{90} \tag{4.4}$$

This is the data that was received from the tests, several assumptions then had to be made to find the rest of the material properties necessary for the finite element analysis. First it was assumed that the material would behave similar to wood and that the compressive strength of the material would be higher than the tensile strength in all directions, but with a different magnitude in the different directions. As can be seen in Equation 4.5.

$$F_{cx} = 2F_{tx} \quad F_{cy} = 5F_{ty} \quad F_{cz} = 5F_{tz} \tag{4.5}$$

Secondly the shear modulus in two planes were assumed to be equal to the tensile moduli for 45 degrees printing direction, while the last shear modulus was assumed to be rather small in comparison due to rolling shear, according to Equation 4.6.

$$G_{xy} = E_{45} \quad G_{yz} = 0.1E_{45} \quad G_{zx} = E_{45} \tag{4.6}$$

Thirdly the shear strengths of the material were assumed according to general timber properties as a certain factor of the compressive strengths, according to Equation 4.7.

$$F_{sxy} = \frac{3}{2}F_{cy} \quad F_{syz} = 0.5F_{cy} \quad F_{szx} = \frac{3}{2}F_{cz} \tag{4.7}$$

Lastly the Poisson's ratio of the material in the three different directions were assumed to be approximately the same as for wood, as can be seen in Equation 4.8.

$$\nu_{xy} = 0.4 \quad \nu_{yz} = 0.2 \quad \nu_{zx} = 0.4 \tag{4.8}$$

What can be noticed by the results is that they differ a lot from the values provided by the material manufacturer, which can be further discussed in Section 5.
The results from the small-scale experiments in combination with several assumptions gave all parameters necessary for building the material model. The model was built according to the theory in Section 2.1.2, and resulted in a matrix as follows:

$$D = \begin{bmatrix} \frac{1}{850} & -\frac{0.4}{350} & -\frac{0.4}{350} & 0 & 0 & 0 \\ -\frac{0.4}{850} & \frac{1}{350} & -\frac{0.2}{350} & 0 & 0 & 0 \\ -\frac{0.4}{850} & -\frac{0.2}{350} & \frac{1}{350} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{400} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{40} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{400} \end{bmatrix} \tag{4.9}$$

4.2 Resulting Components

The final components follows the initial plan but with a few tweaks to simplify usability. In this chapter the components will be presented with a thorough description of how they work, on a programming level, and how to use them.

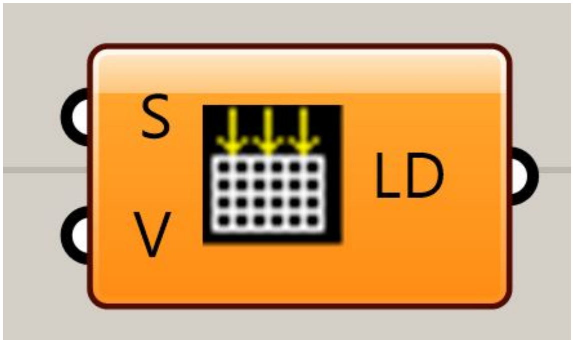


Figure 4.3: The Load Component

The Load Component takes as input a single planar surface and a vector in 3 dimensions. The surface symbolises the area that the load is applied to, while the vector describes the load itself, both its direction and size. If the user wants to create a point load it is possible by creating a surface small enough to only affect a single node in the Finite Element Analysis. The component takes this data and creates an internal class called LoadData, containing the surface and the vector. This class is set as output from the component. If several surfaces are used as input, the output is set to a list of LoadData classes.

The Support Component takes as input a single planar surface as well as Boolean values from its six different buttons. The surface symbolises the area that is restrained, while the buttons symbolises how it is restrained, and in which directions. With three buttons for displacement and three for rotation. The component takes this data and creates an internal class called SupportData, containing the surface as well as an array of Booleans describing the restrictions. The class is then set as

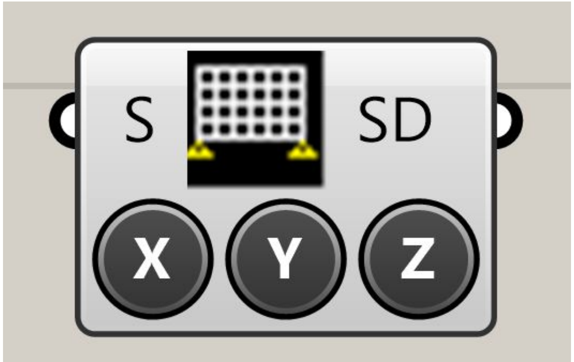
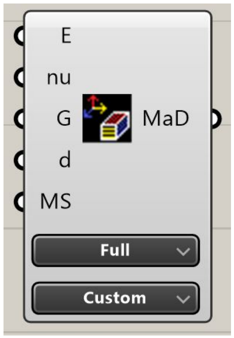


Figure 4.4: The Support Component

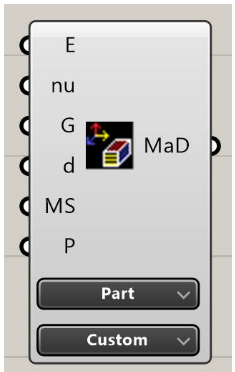
output from the component. If several surfaces are used as input, the output is set to a list of SupportData classes.

(a) Material Component Type A.

(b) Material Component Type B.



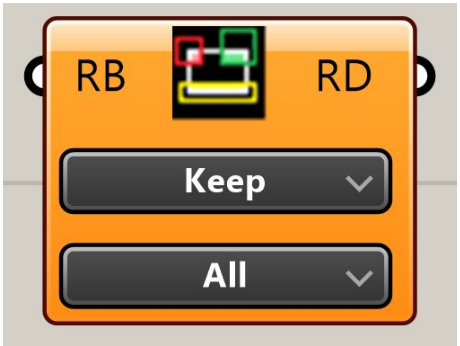
(c) Material Component Type C.



(d) Material Component Type D.

Figure 4.5: The Material Component

The Material Component works in several different ways, depending on the selections in its two drop-down menus. In the top menu the user can choose between Full model or Part of model, where Full model means that the material will be applied to the whole model, while Part of model means that the material will be applied to a certain region of the model. If Part is selected the component gains an extra input slot where the user can plug in a B-rep that contains the region for that material. The user can also choose between several different materials in the bottom menu, also having the ability to select a Custom material. If Custom is selected the component gains several new input slots where the user can plug in the necessary material data. Part and Custom can of course be selected at the same time, meaning that the Material Component has four different appearances. The component takes the data from its inputs and creates an internal class called MaterialData. This class contains information about the maximal and minimal stresses the material can take in form of intervals for the different directions, as well as two different methods for calculating the stiffness matrix D, one Isotropic and one Orthotropic. Depending on the length of the different input lists, e.g. if the Youngs Moduli contains three values or one etc, one of the methods is used to create the stiffness matrix. The class is then set as output data from the component. If Part is selected and several B-reps are plugged into the component, the output is set as a list of MaterialData classes.



(a) Region Component Type A.

(b) Region Component Type B.

Figure 4.6: The Region Component

The Region Component is a component that is only used for the iterative process, but as it is plugged into the model component it is explained in this section. The component works in several different ways, depending on the selections in its two drop-down menus. It takes as input a single B-rep that symbolises the region of the model that will be affected by the selections in the drop-down menus. In the top menu the user can choose between Keep and Remove, which, depending on the selection, forces the model to either keep the elements in the region, or remove them, through all iterations. In the bottom menu the user can choose between All or Custom. If All is selected, then all elements in the region will be either kept or removed, if Custom is selected, the component adds an input slot for a tolerance value between 0 and 1, that describes to what extent the elements are to be kept or removed. The components uses its input data to create an internal class called RegionData, that contains the B-rep symbolising the region as well as a value between -1 and 1 describing the importance of the elements in the region, with -1 being equal to “remove” and 1 being equal to “keep”. The class is then set as output from the component. If several B-reps are plugged into the component the output will be set as a list of RegionData.

The Model Component takes as input all the data from the earlier mentioned components, e.g. LoadData, SupportData, MaterialData and RegionData, as well as Element Size and Domain. The Element Size input describes the approximate size of the elements for the Finite Element Analysis, while the Domain sets the geometry of the model as a B-rep. If no data is plugged into the Domain the component

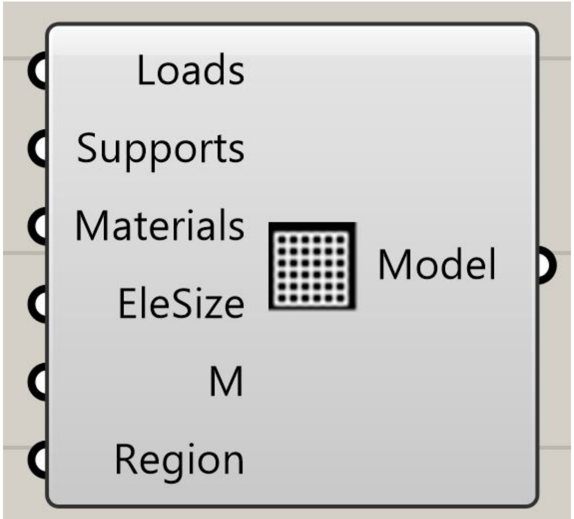


Figure 4.7: The Model Component

does, however, create a starting model from the input surfaces in LoadData and SupportData. The component uses the data from its input to generate an internal class called FemModel. The FemModel is constructed in another internal class called ModelBuilder where all the nodes and elements as well as their connectivity are constructed. The correct material data is added to the nodes as well as the loads and the constraints from the supports. The class FemModel contains many methods to make these calculations easy, as well as to make the data be easily accessible in the later Finite Element Analysis. These methods includes such as SetNodeCoordinate, SetNodeDisplacement and SetElementConnectivity but also methods as CalculateElementsPerNode, CalculateAdjacentElements and GetElementNodalPositions as well as AddMaterial, AddRegion and AddLoad. The idea is that all data of the model should be reachable from the FemModel class. The FemModel class is set as output from the component.

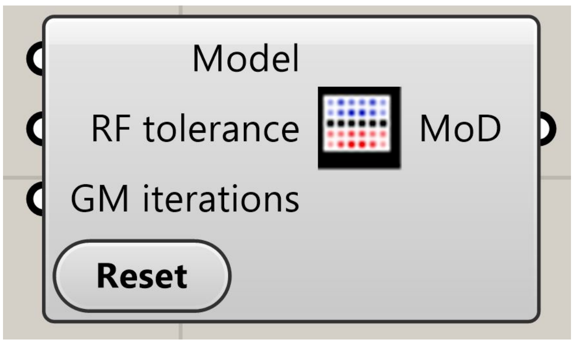


Figure 4.8: The FEM-Solver Component

The FEM-Solver Component takes as input the FemModel from the Model Component as well as an RF-tolerance and a number of GM-iterations. The RF-tolerance is used to control when to stop the solver, as the solver stops iterating when *tolerance* $< F - K * a$. If the tolerance is set to 0 the solver will automatically choose when to stop. The GM-iterations number sets the number of iterations of the GMRES-solver and is used to control the convergence behaviour of the solver. The component works by changing the FemModel class by running it through another internal class called FemSolver, which performs a Finite Element Analysis on the model, following the theory presented above, as well as two additional internal classes called DisplacementSolver and StrainStressSolver. It contains a reset button to easily enable the possibility to restart the calculations. The changed FemModel is set as output from the component, now containing all the data from the FEA, such as displacements, strains and stresses.

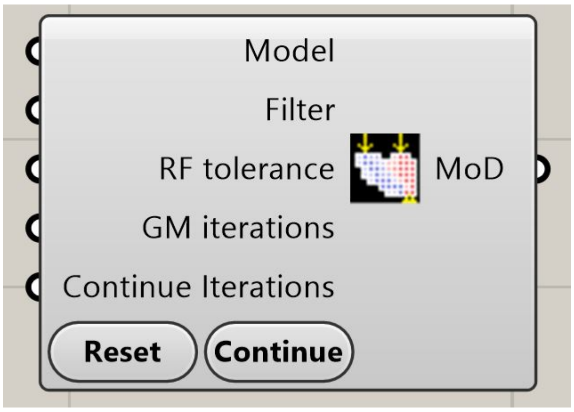
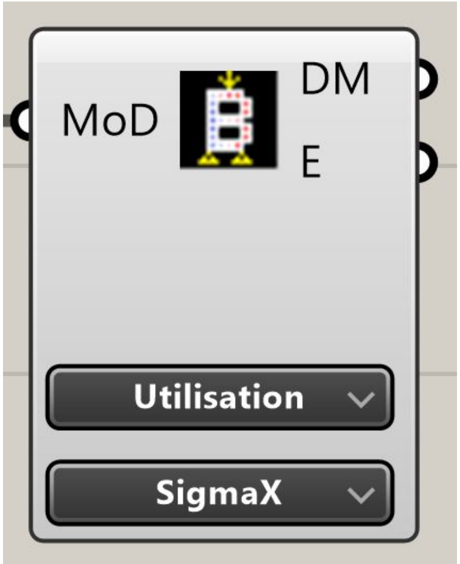


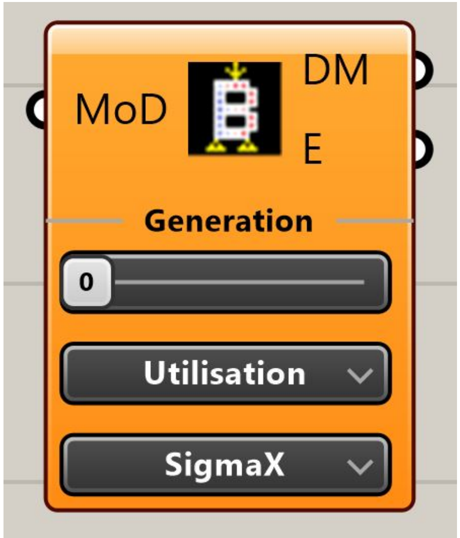
Figure 4.9: The Iterative FEM-Solver Component

The iterative FEM-Solver Component works in the exact same way as the regular FEM-Solver Component, with a few small differences. As this component uses Generative Design to iterate through the results and create an optimised levels it has to work with iterations. Therefore an additional input has been added, in combination with a button with direct relation to the number of iterations. The input called "Continue Iterations" takes the number of iterations for each step as an int. When the "Continue" button is pushed, the component will iterate this number of times. Additional inputs has been added as well, with one for output from the filter component as well as one for RF tolerance. The RF tolerance is a double which is equal to the accuracy of the FEM-solver, i.e. the residual force threshold as a percentage of the applied loads. The Model Data class with an updated model for each iteration is set as output from the component.

The Visualize Component takes the changed FemModel class from the FEM-Solver Component as input. Depending on the selected value of its slider, as well as the selections in its two drop-down menus, it can provide different outputs. The slider symbolises the different generations in the iterative process, and is as such only used when a FemModel with several iterations registered is plugged in. The drop down menus lets the user decide what data to visualize. The top menu contains the different alternatives Utilisation, Deformations, Principal Stress and Fitness, while the bottom menu contains a specification list for each of the top menu items. For Utilisation it contains SigmaX, SigmaY, SigmaZ, TauXY, TauYZ, TauZX, Von Mises,



(a) Visualize Component Type A.



(b) Visualize Component Type B.

Figure 4.10: The Visualize Component

Principal Stress Compression, Principal Stress Tension, Principal Stress Intermediate and Principal Stress Combined. For Deformations it contains UX, UY and UZ. For Principal Stress it contains Max-Min, Field and Intermediate and for Fitness it contains Element Von Mises, Blurred Element Von Mises, Nodal Von Mises and Blurred Nodal Von Mises. Depending on the selections a mesh that is coloured following the chosen result is set as output from the component.

The filter component is a component that is used to easily plug different filters into the iterative solver. The component was from the beginning created to provide a quick way to test between different filters when tweaking the component and its algorithms to work for several different cases. The component takes as input a Blurring Radius, a Sharpening Radius, a Sharpening Intensity, a number of Total Iterations, a number of Blurring Iterations and a number of Sharpening Iterations. It uses these values to create an internal class called Filter which creates a filter for the iterative process. The two drop-down buttons makes it possible to save and store several filters that can be easily switched between in the process. The Filter class is set as output from the component.

The Lattice Component works by taking the changed FemModel from the solver as an input, in combination with the shell mesh from the Visualize Component, and creating a lattice structure following the need of the model as well as the boundary of the mesh. It calls upon an internal class called LatticeData that creates a lattice structure as a mesh. The mesh is then set as output from the component.

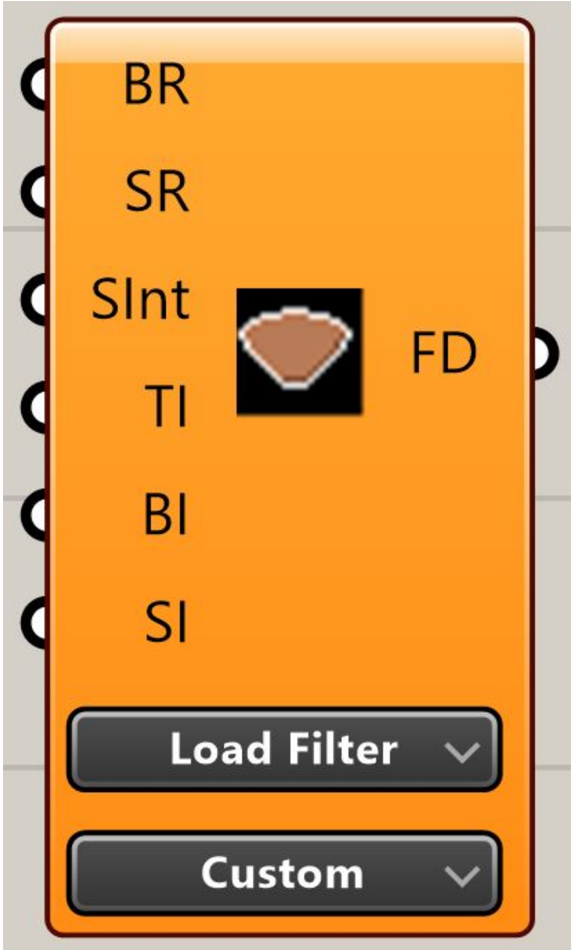


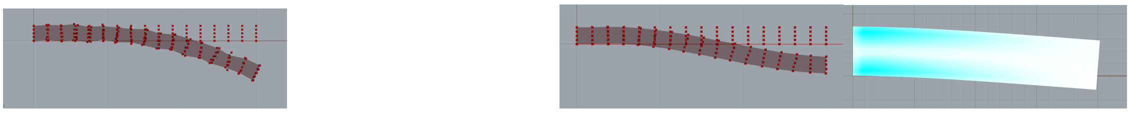
Figure 4.11: The Filter Component

Figure 4.12: The Lattice Component

4.3 Tests, Confirmations and Issues

During the process of implementing the theory into the components and creating the finalised software several tests were performed on the software itself. Some of these tests showed issues with the software that needed to be taken into consideration, while some were simply used as a confirmation of the functionality of the software and the components.

One issue that arose in early testing of the FEM component was an issue with rotation of the material. The issue with the rotations took its roots in the theory that was used as a reference in the start of the project. The theory was somewhat unclearly written in several ways, and the ways that the rotations were taken in considerations in could not be found in any other source. After struggling a while with several different methods of how to take the rotational degrees of freedom in consideration it was decided to, in order to move forward, simplify the solution and create a FEA-program that does not take the rotations into consideration. This simplification solved the issue with the rotations, but still provided a good and stable result that was not far off from any other sources that it was compared to.



(a) Test A **(b)** Test B **(c)** Test C

Figure 4.13: Bending tests

When adding loads to the model they were initially added directly onto the nodes that lay, within a certain tolerance, on the load surface, where an equal amount of load were added to each node. There were several discussions on how to correctly add the loads in a way that could also involve the possibility of the load surface cutting through the hexahedral elements in some way. In the earlier process of adding loads they could only be added exactly to the edge of the hexahedral elements, forcing the load surfaces to be orthogonal in one of the three Cartesian planes. After discussing and testing several different alternatives a solution was found. The solution was based on dividing the original load surface into a certain number of squares as a quite fine mesh. Every face of the mesh received a certain load depending on its area as well as the original load input on the load surface. The loads of the faces were then distributed to the nodes depending on which element they were positioned in, where in the element they were positioned as well as the shape functions of the edges of the element. In this way the loads of the surface could be distributed correctly even if it was cutting through several elements.

When the components for the Finite Element Analysis were nearing completion several comparisons were made with the software Abaqus in order to confirm the

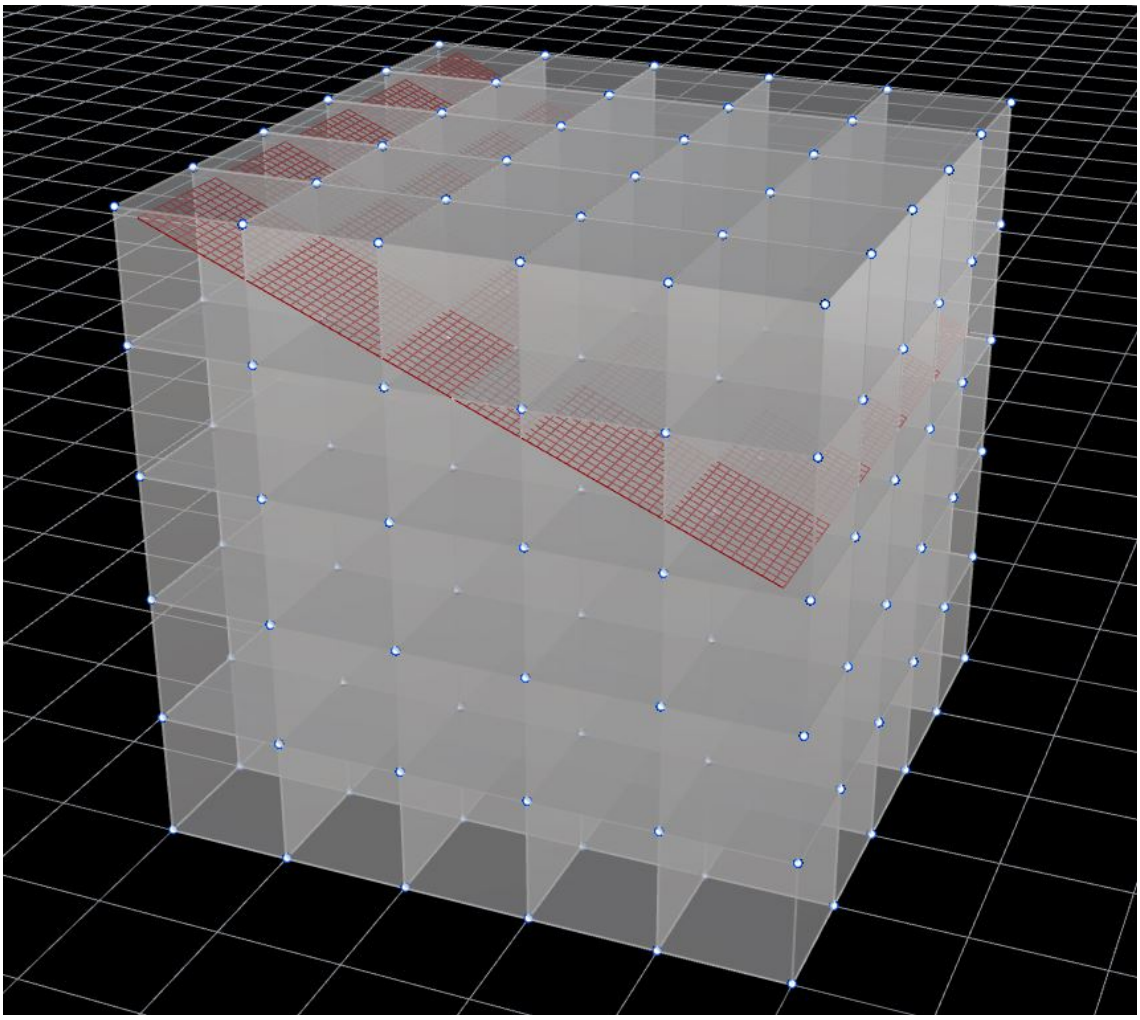
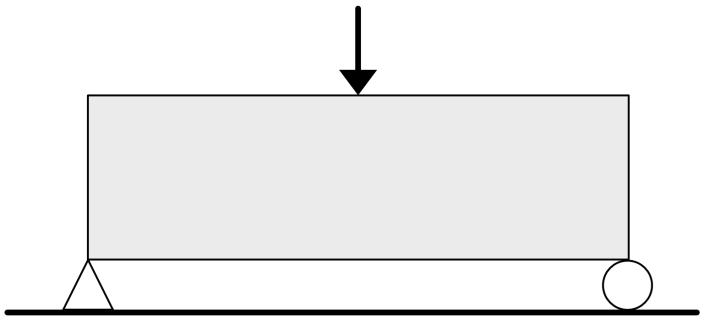
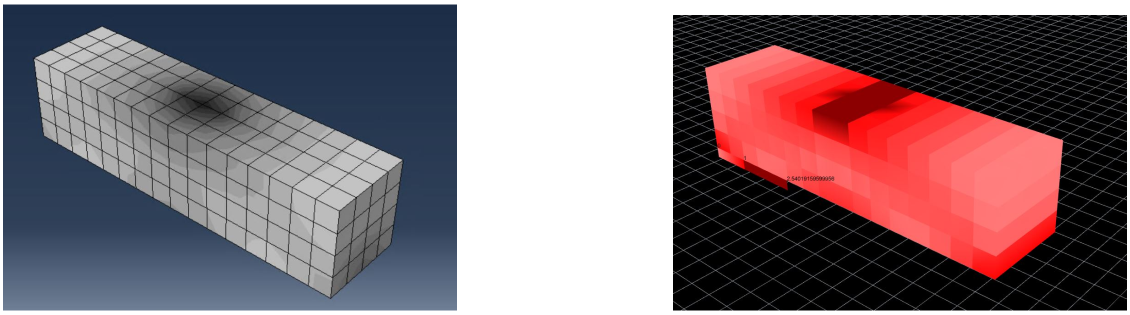


Figure 4.14: Adding Loads

calculations as well as the components useability in the generative design process. Two of the cases that were tested and compared where those of a regular beam, loaded with a point load in the middle, and a cantilever, loaded with a point load at the end.



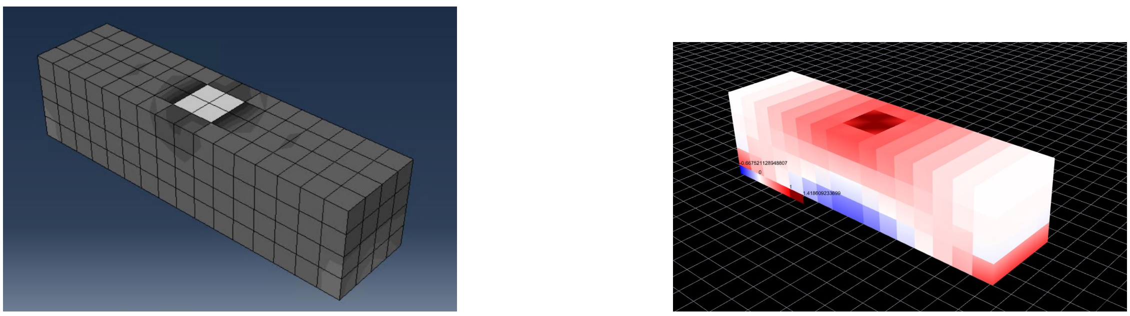
Comparison Test case A, Beam loaded with point load.



(a) Abaqus Beam Von Mises.

(b) Project Beam Von Mises

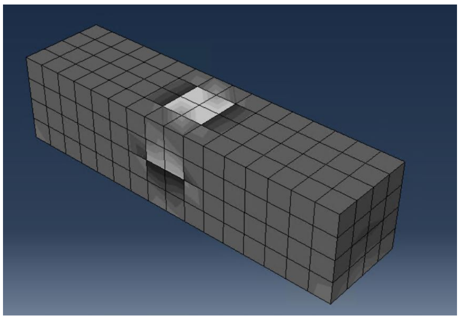
Figure 4.15: Abaqus Comparison Von Mises



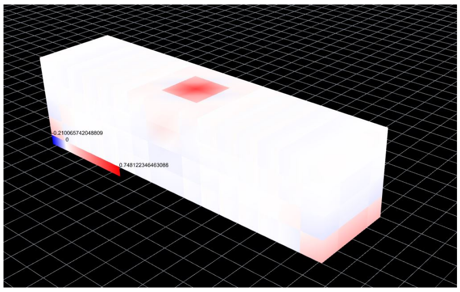
(a) Abaqus Beam Sigma X.

(b) Project Beam Sigma X

Figure 4.16: Abaqus Comparison Sigma X

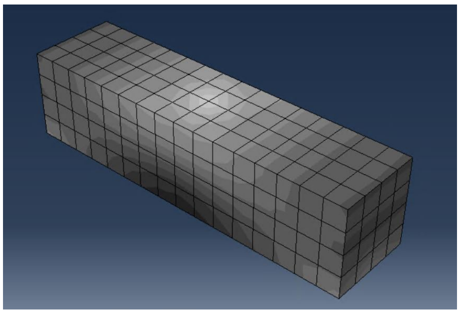


(a) Abaqus Beam Sigma Y.

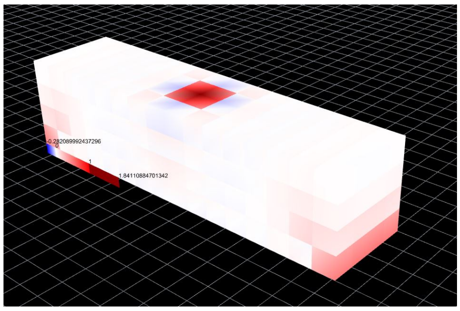


(b) Project Beam Sigma Y

Figure 4.17: Abaqus Comparison Sigma Y

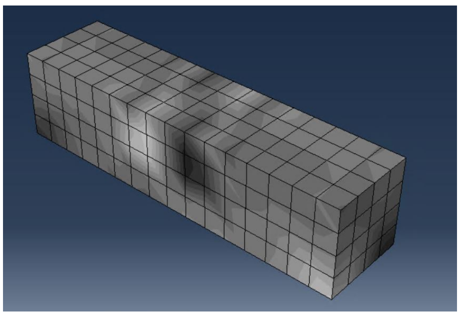


(a) Abaqus Beam Sigma Z.

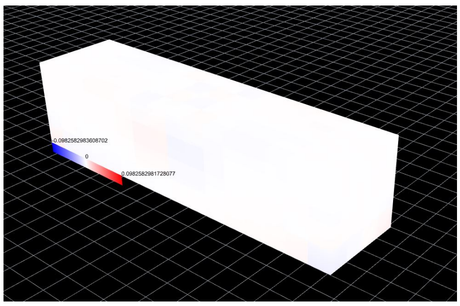


(b) Project Beam Sigma Z

Figure 4.18: Abaqus Comparison Sigma Z

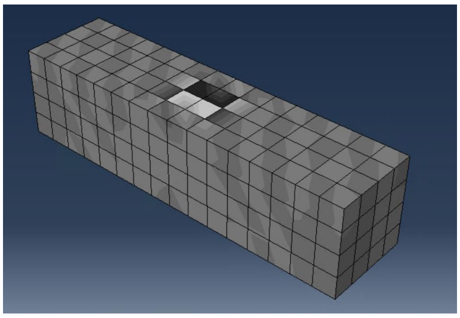


(a) Abaqus Beam Tau XY.

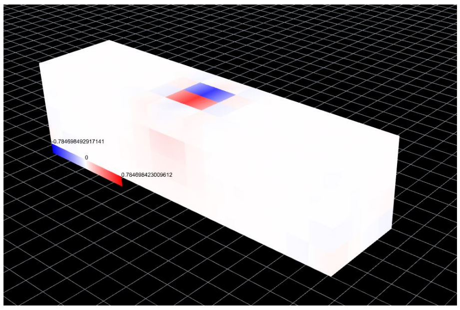


(b) Project Beam Tau XY

Figure 4.19: Abaqus Comparison Tau XY

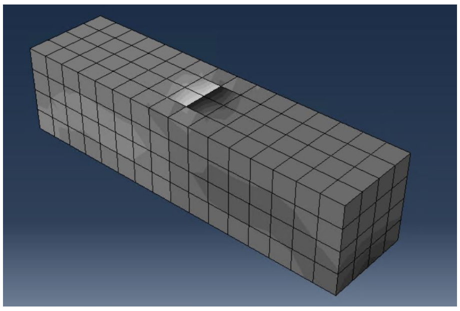


(a) Abaqus Beam Tau YZ.

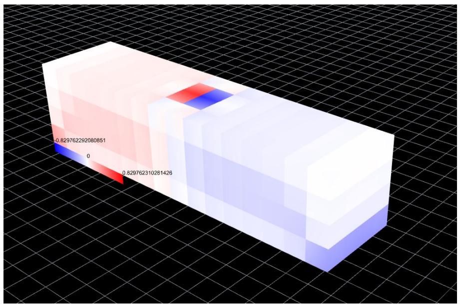


(b) Project Beam Tau YZ

Figure 4.20: Abaqus Comparison Tau YZ



(a) Abaqus Beam Tau ZX.



(b) Project Beam Tau ZX

Figure 4.21: Abaqus Comparison Tau ZX

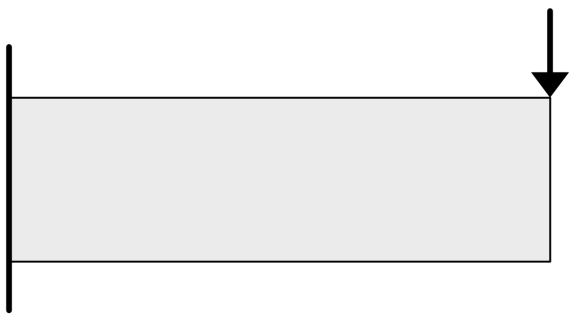
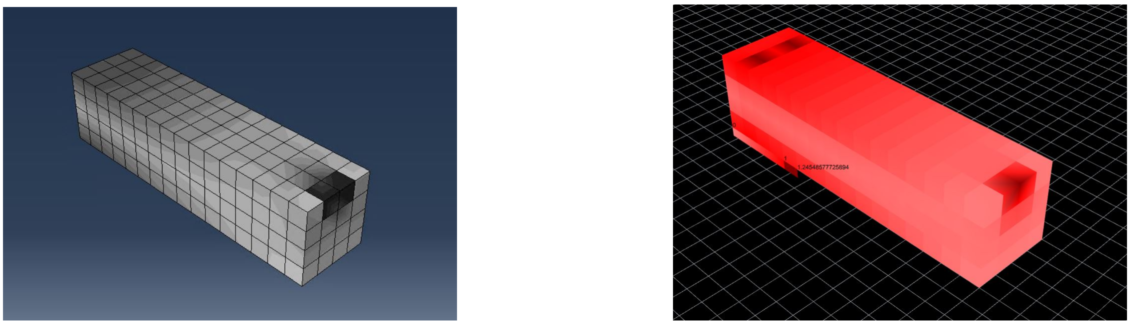


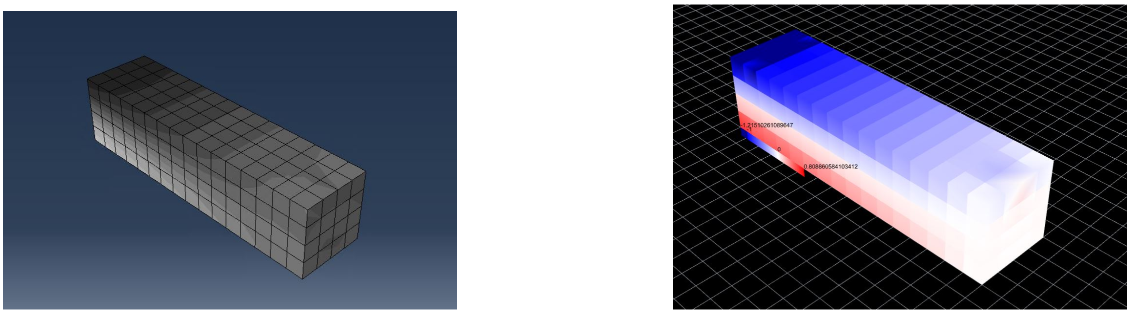
Figure 4.22: Comparison test case B, Cantilever loaded with point load



(a) Abaqus Cantilever Von Mises.

(b) Project Cantilever Von Mises

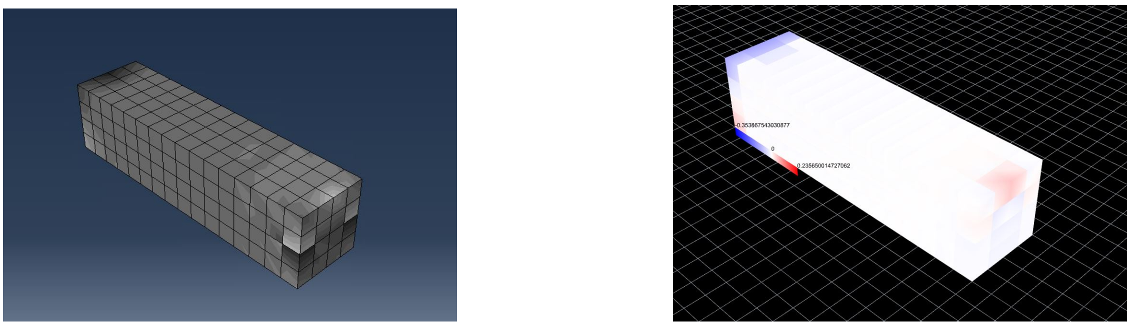
Figure 4.23: Abaqus Comparison Von Mises



(a) Abaqus Cantilever Sigma X.

(b) Project Cantilever Sigma X

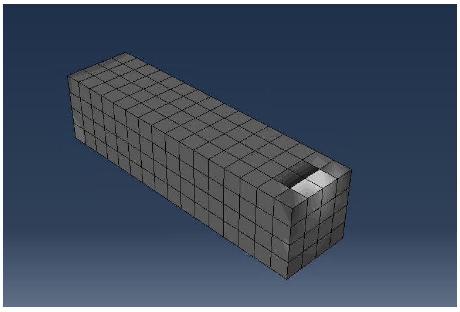
Figure 4.24: Abaqus Comparison Sigma X



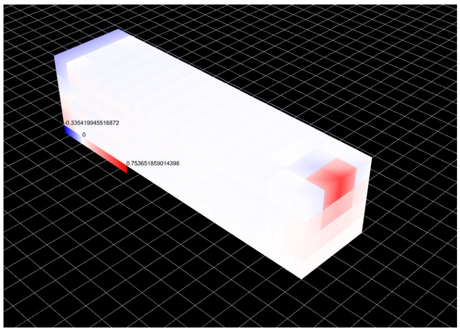
(a) Abaqus Cantilever Sigma Y.

(b) Project Cantilever Sigma Y

Figure 4.25: Abaqus Comparison Sigma Y

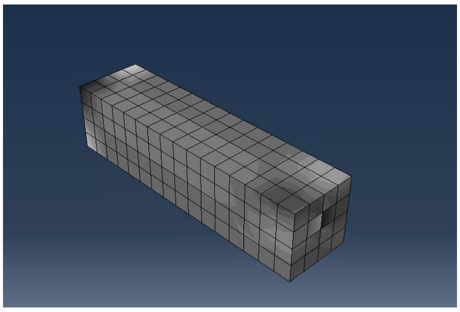


(a) Abaqus Cantilever Sigma Z.

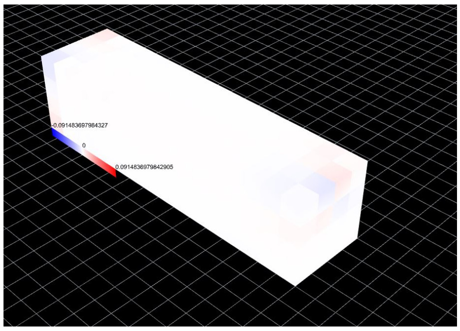


(b) Project Cantilever Sigma Z

Figure 4.26: Abaqus Comparison Sigma Z

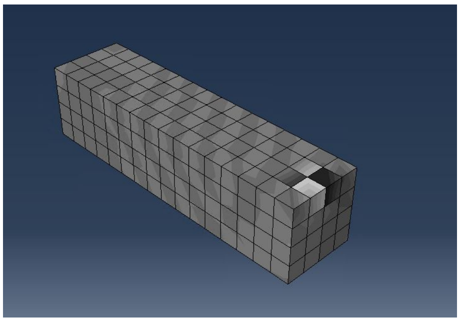


(a) Abaqus Cantilever Tau XY.

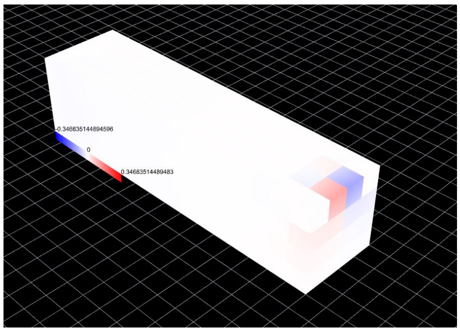


(b) Project Cantilever Tau XY

Figure 4.27: Abaqus Comparison Tau XY

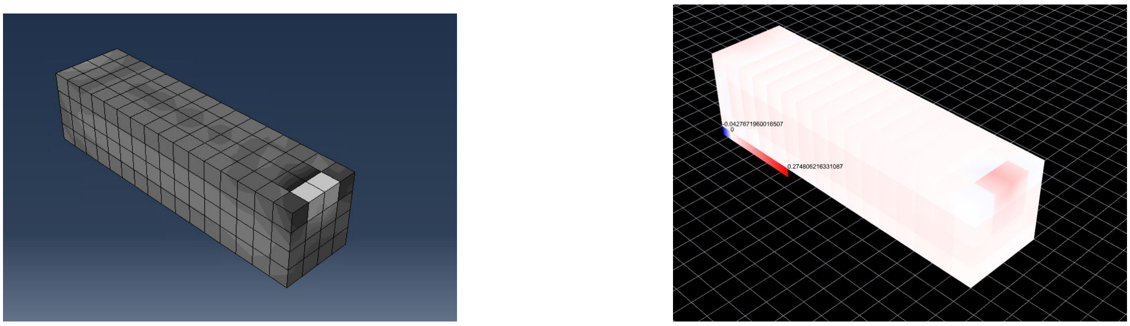


(a) Abaqus Cantilever Tau YZ.



(b) Project Cantilever Tau YZ

Figure 4.28: Abaqus Comparison Tau YZ



(a) Abaqus Cantilever Tau ZX.

(b) Project Cantilever Tau ZX

Figure 4.29: Abaqus Comparison Tau ZX

As can be seen from the comparison tests the results are very similar between the two programs. This means that the project components can simulate stress patterns in a correct way and can therefore be trusted to be used for the generative design process.

A few tests were also carried out with the focus on the capacity of the models to make sure that the utilisation ratios of the elements that the program resulted in coincided with the correct results from simple calculations. In Figure 4.30 the result of a single column that was used as a test object, can be seen.

The process of creating the Parametric Model begun as well with a lot of testing, mostly performed in 2D, i.e. in 3D but with the thickness of one element, to make sure that each test could be performed quickly enough for it to be viable for many fast and small changes. The goal of the process was to find a method for removing elements that would be suitable for all types of models. This included the use of filters as well as additional functions for choosing which elements to remove as well as which to add.

An early issue in this process was that the program removed to many elements, as well as elements in the wrong positions due to stress concentrations. This could result in the model cutting of its own legs, which after a few iterations more would result in the entire model being deleted since the program was not able to find a solution. This problem was solved with the use of correct filters, that could concentrate the stress patterns through the model and thereby safeguard the legs.

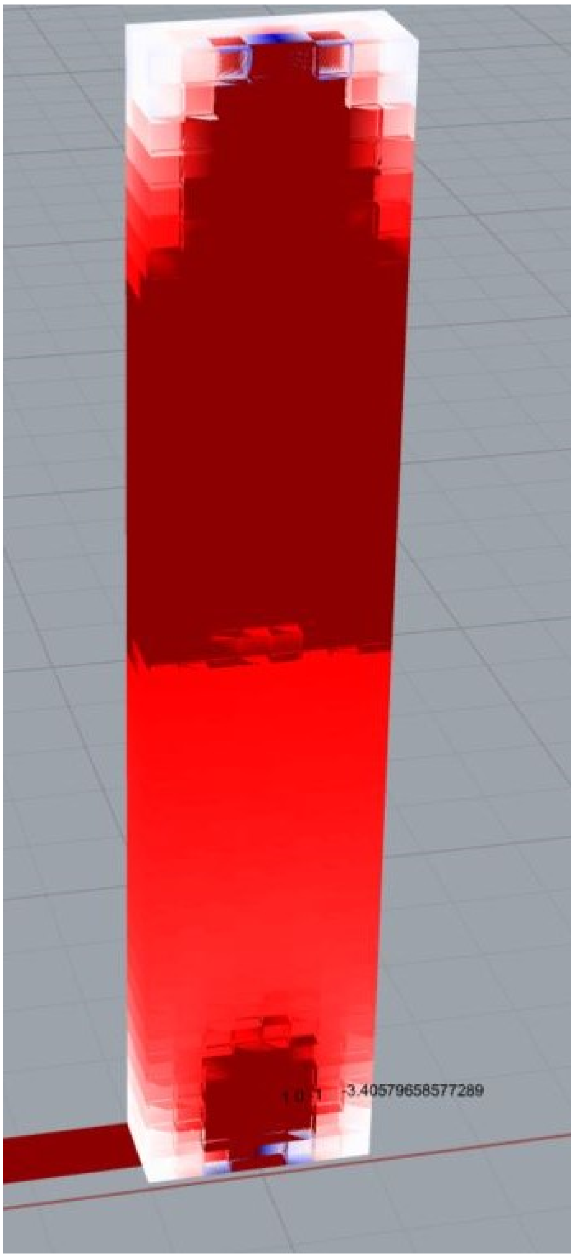


Figure 4.30: Capacity test of a single column.

(a) Iteration A

(b) Iteration B

(c) Iteration C

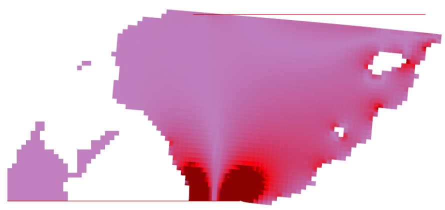
(d) Iteration D

(e) Iteration E

(f) Iteration F

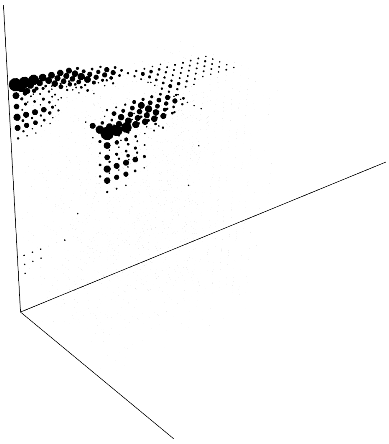
Figure 4.31: Early iterative tests of the generative design process, element have a density of either 1 or 0.

(a) Early iteration

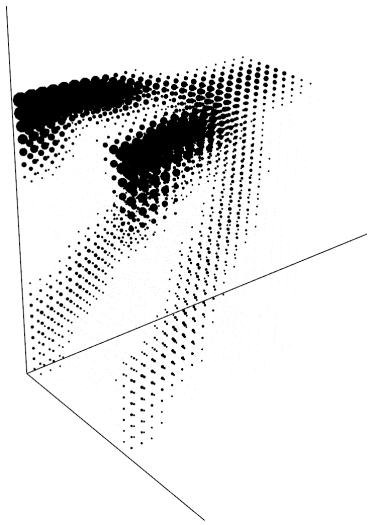


(b) Later iteration

Figure 4.32: The model cutting off its own leg.

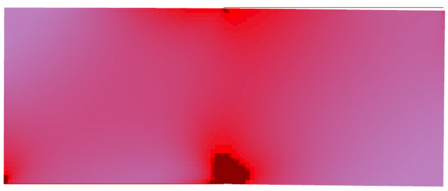


(a) No filter

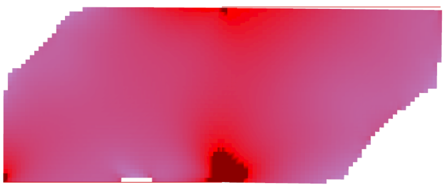


(b) With filter

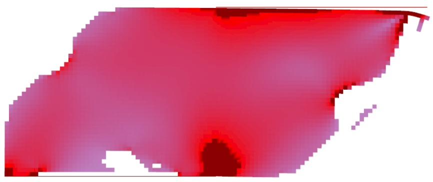
Figure 4.33: With the use of filters it is possible to concentrate the stress patterns, and thereby safeguard the legs of the model.



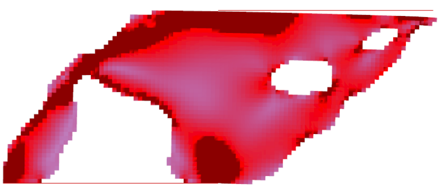
(a) Iteration A



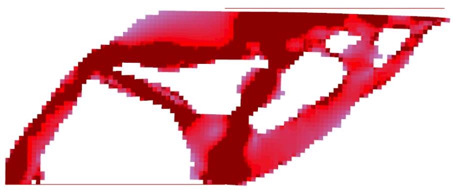
(b) Iteration B



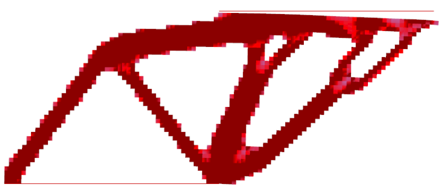
(c) Iteration C



(d) Iteration D



(e) Iteration E



(f) Iteration F

Figure 4.34: The use of filters create a more stable model.

4.4 Showcase Projects

To showcase the finalised Grasshopper components a few load cases were created as example project that was run through the program. Several simple projects were created to show the versatility of the program on simple load cases. These projects had the focus of a fast running time, with a rather coarse grid of elements, mainly to giver the idea of how the program could work as a quick design tool, with more approximate solutions.

After the simple showcase projects, another, more complex, showcase project was created. The purpose of this project was to show the programs possibility to solve complex load cases, even though it was created with a fast running time and a rather coarse grid of elements.

To show the detail level of the program, and its possibility to create smooth meshes in combination with complex solution another showcase project was created as well. Also this with a quite quick running time, but with a higher detail level, using the Region Component.

4.4.1 Project 1 - Bridge Beam

The first example project is a simple bridge beam with an evenly distributed load on top. As can be seen from the analysis the material used is stronger in compression than tension, thereby removing all elements in the tension zone while keeping an arch shape with struts for the compression loads.

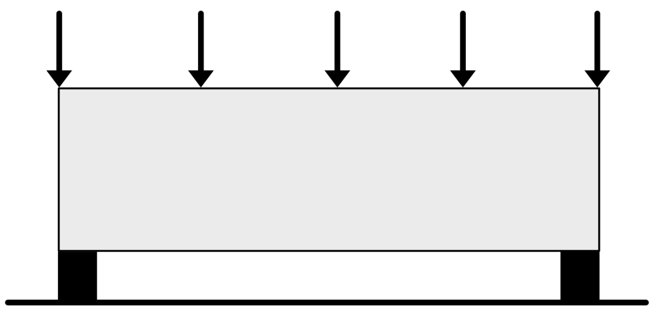


Figure 4.35: Project 1, Simple bridge beam with distributed load

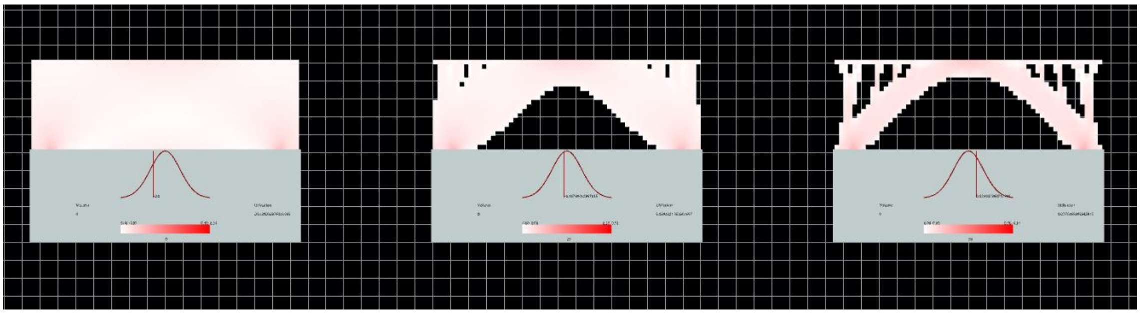


Figure 4.36: The iterative results from Project 1, showing iteration 0, 25 and 50.

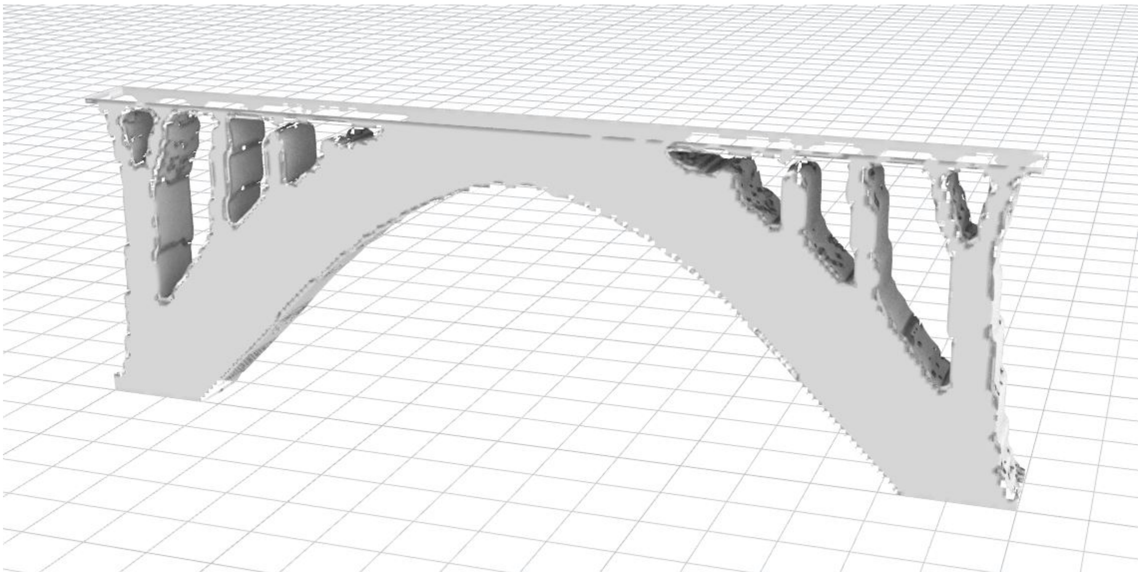


Figure 4.37: Refined mesh of the bridge project.

4.4.2 Project 2 - Cantilever Beam

The second example project is a simple cantilever with a downward force at the end. As can be seen from the analysis the model gets divided into several crossing compressive and tensile struts, that gets smaller for each iteration.

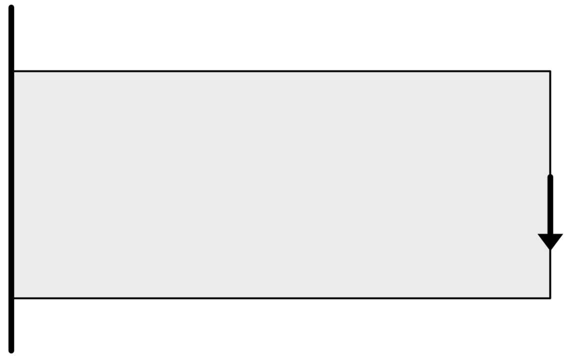


Figure 4.38: Project 2, Simple cantilever beam.

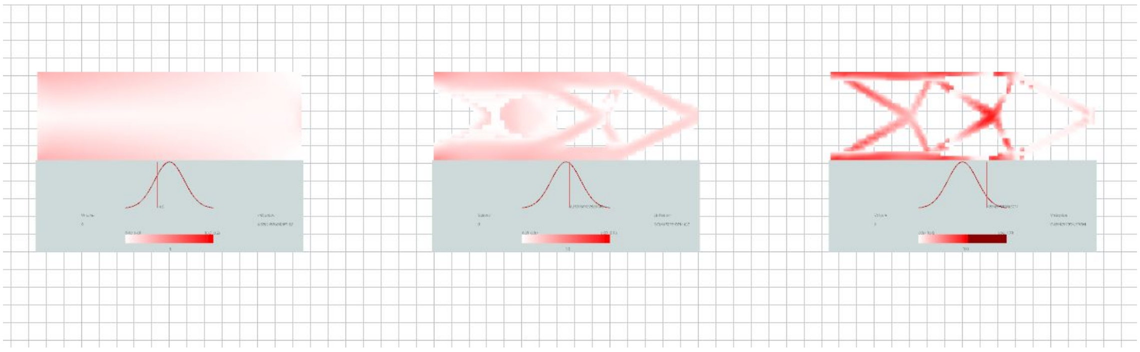


Figure 4.39: The iterative results from Project 2, showing iteration 0, 25 and 50.

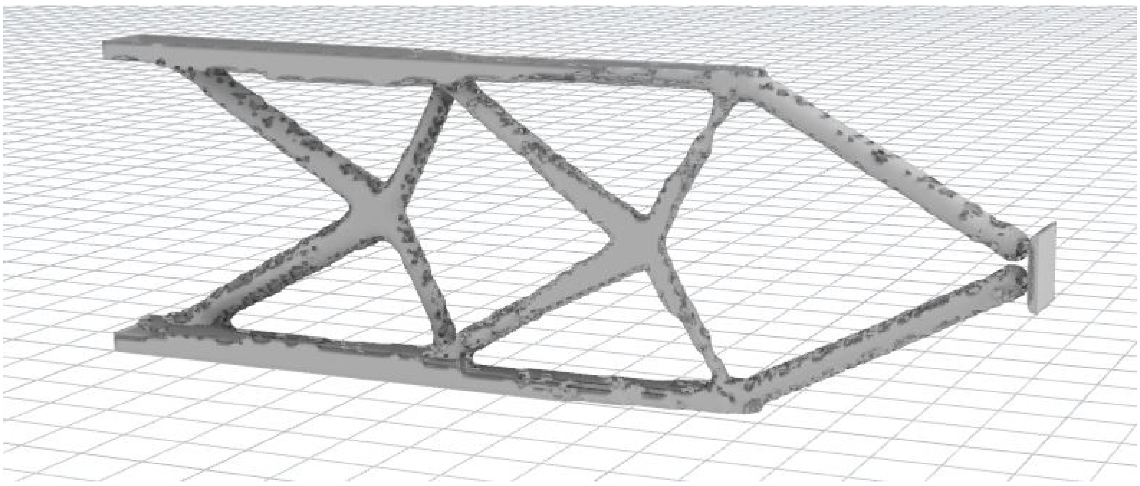


Figure 4.40: Refined mesh of the cantilever project.

4.4.3 Project 3 - Column

The third example project is a standing column with an evenly distributed load on top. As can be seen from the analysis the column becomes divided into several parts. It is clear that in this case the detail level of the grid of elements was maybe too coarse, and that some small tweaking with the filter component might be necessary for a more exact solution.

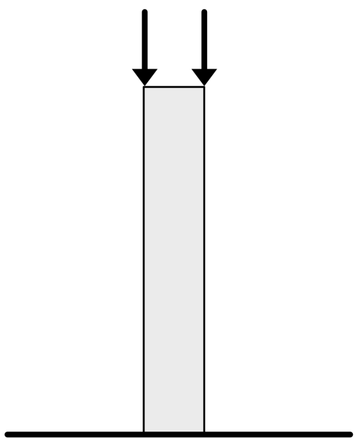


Figure 4.41: Project 3, Standing column

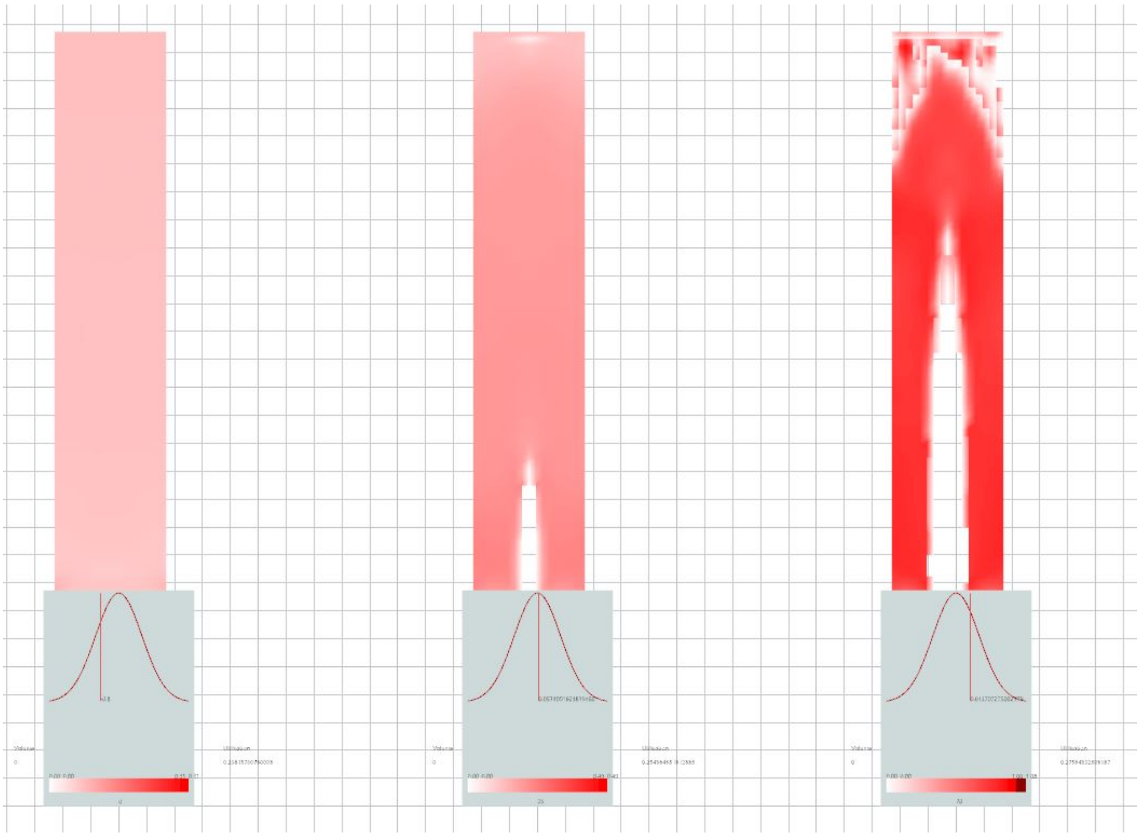


Figure 4.42: The iterative results from Project 3, showing iteration 0, 50 and 100.

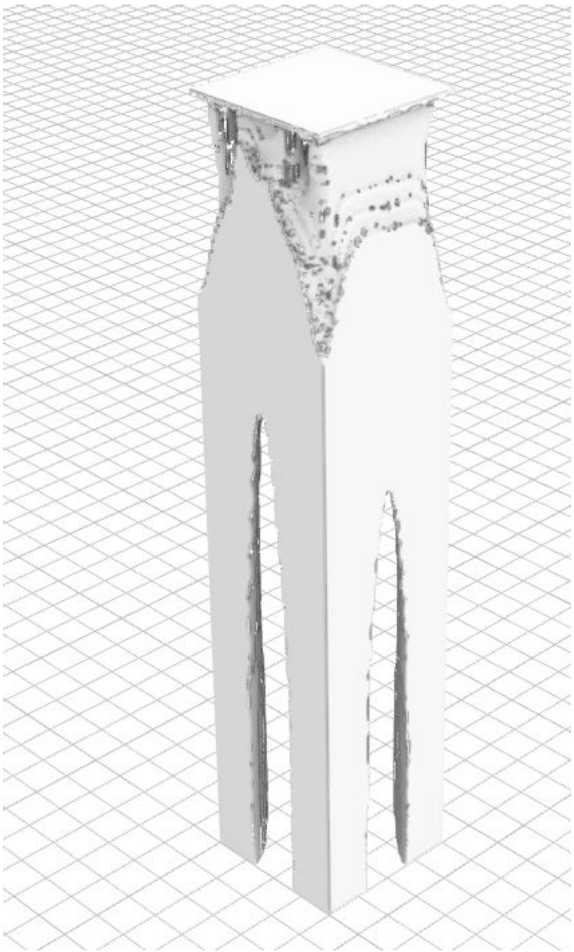


Figure 4.43: Refined mesh of the column project.

4.4.4 Project 4 Complex Load Case

To visualize a more complex load case as a showcase project, a structure was created out of the premises of a protruding roof from a gas- or bus-station or similar. The structure was allowed to use the building of the station as support, as well as certain areas on the ground, while being forced to protrude a certain area around the building. As can be seen in Figure 4.44.

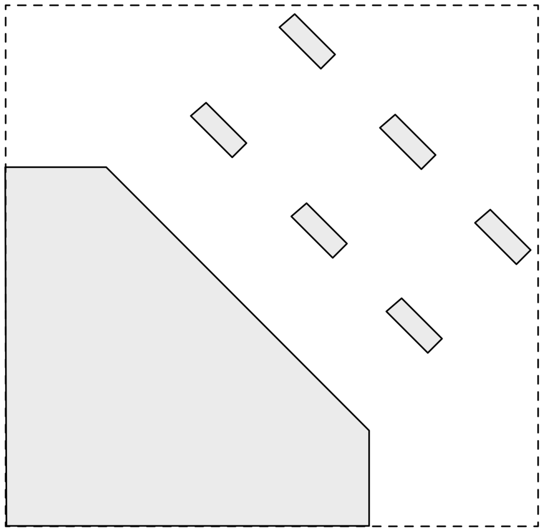


Figure 4.44: The ground plan of Project 4, the roof for a gas- or bus-station.

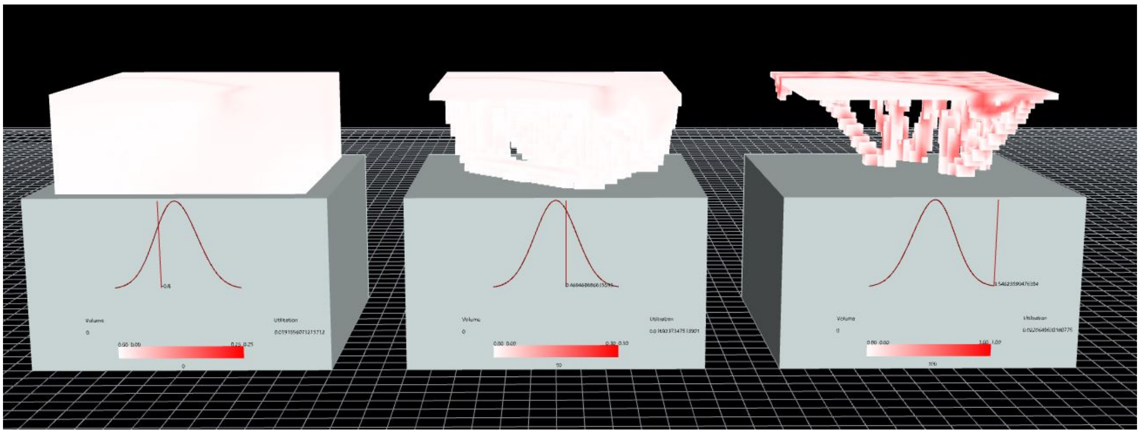
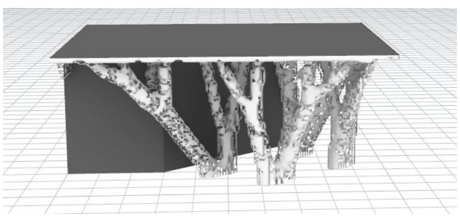
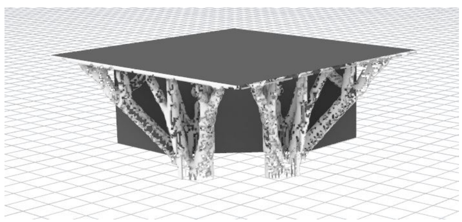


Figure 4.45: The iterative result from Project 4, showing iteration 0, 35 and 70.



(a) Perspective A



(b) Perspective B

Figure 4.46: Refined mesh of the Gas-Station project.

The iterations resulted in a protruding roof taking support to the ground through tree like structures. If this project was to be iterated with higher accuracy, a good thing to consider would be using the Regions Component for areas where cars are supposed to be able to drive.

4.4.5 Project 5 Detailed Complex Load Case

To show the possible detail level of the program another showcase project was created. This goal of the project was to create a staircase following a wall with a bench on the backside, using the Region Component for clearing the walking and sitting space and creating flat surfaces.

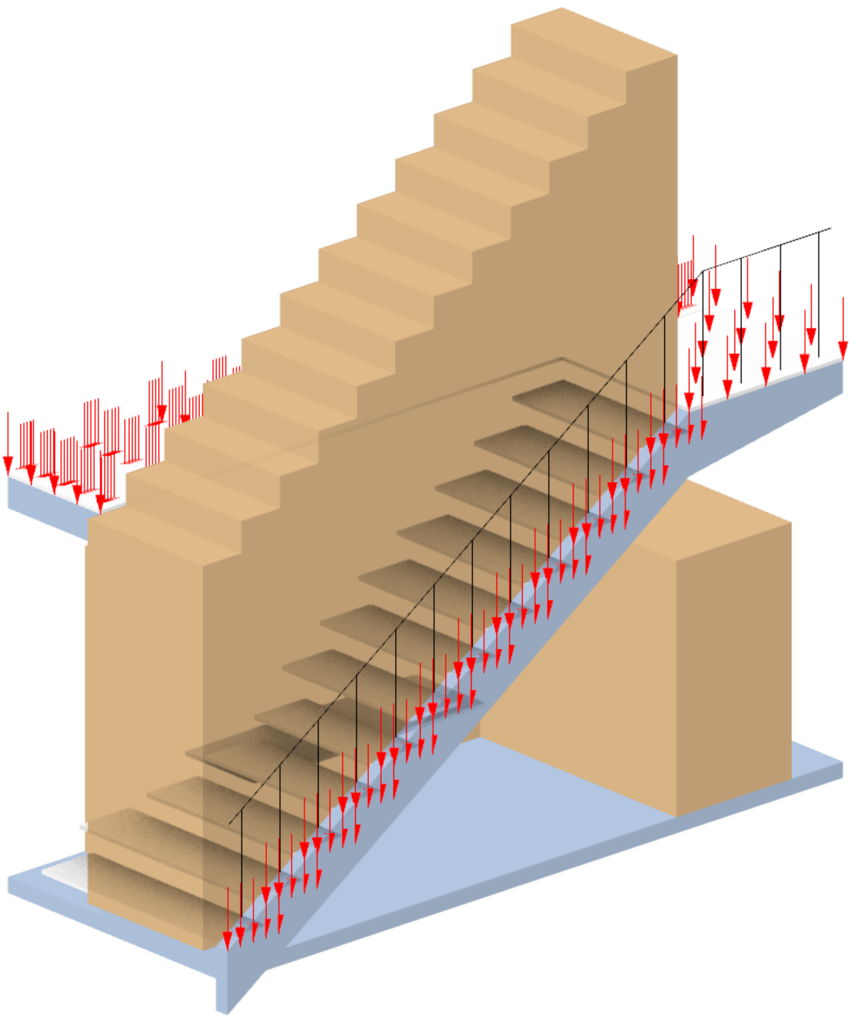
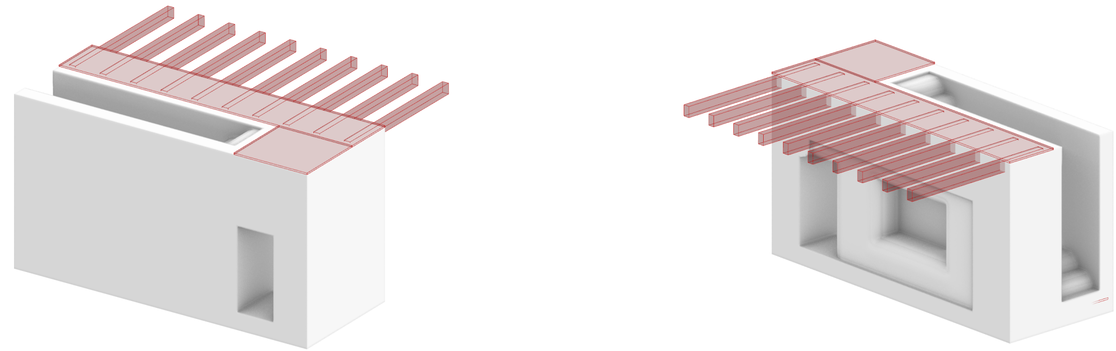


Figure 4.47: The boundary conditions for Project 5, with the yellow areas symbolising regions to remove and the blue areas regions to keep.

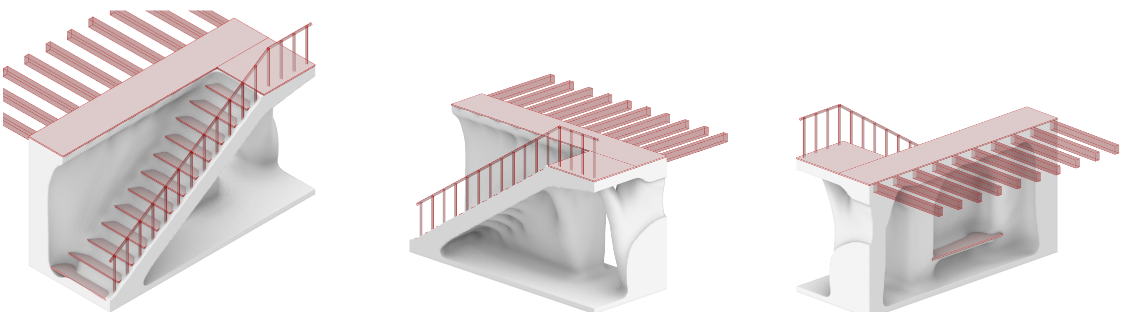
The project received a rather complex starting model, with several regions that were not allowed to be filled with elements and worked from this towards creating an optimised solution to carry the staircase.



(a) Perspective A **(b)** Perspective B

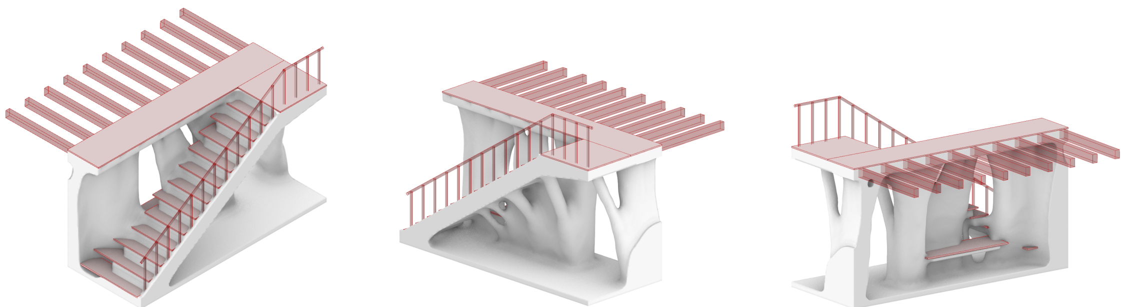
Figure 4.48: Starting model of Project 5.

This solution was found with two different methods with a different porosity of the material, with a higher porosity leading to thicker struts carrying the staircase, as can be seen in Figure 4.49-4.50.



(a) Perspective A **(b)** Perspective B **(c)** Perspective C

Figure 4.49: Iterations with a higher material porosity.



(a) Perspective A **(b)** Perspective B **(c)** Perspective C

Figure 4.50: Iterations with a lower material porosity.

4.5 3D-Printing

To showcase the use of the finalised grasshopper components one of the example projects from Section 4.4 was chosen to be 3D-printed. At the start of the project, the idea was to showcase a model printed in the hybrid composite material with the robot arm printer as well. However, due to lack of time as well as a fully booked printer, this idea had to be put on hold.

5
Conclusion

Having finished the project and the software with quite a few struggles on the way there are many thoughts circulating on what could have been done differently. Firstly the whole working process could have been changed in many different ways that might have given different end results. If, from the beginning of the project, more time had been put into setting up a distinct plan on the necessary steps of the project, several obstacles might have been found, and more easily overcome. If a more concrete plan had existed early on, the method could have been more literature focused in the beginning leading to many issues never arising. If, for example, the project would have had the focus on creating lattices following the principal stresses from the beginning, a whole other method might have been used, and the software would look very different. It can, however, be said that a more thorough literature study, focusing on the whole part of the project, early on, would have resulted in less issues along the way.

The project could also have had a different focus, which would have changed the outcome. In the beginning there were a few discussions about using another software for the whole project, or at least a part of it. Several suggestions were made to use the FEM-calculations or the topological optimisation of another software, to be more free to focus on the hybrid composite material, and its properties. However, the decision was made that the focus would be on creating a new software, with the goals of simplifying the use of the material in the building sector, rather than working with the material itself.

One thing, however, that was lost due to lack of time, and would have been a great addition to the project, would have been more tests for the material properties. As the project was performed, most of the properties had to be assumed from a few tests and several assumptions of the material behaviour. How much this method affects the outcome of the software is unknown, but it is definitely a point that could be improved. More tests that could have been performed to improve the feasibility of the software could have been more comparisons with known programs. For instance, several comparisons could have been made for both the FEM-part of the project as well as the part involving Generative Design. This could have increased the reliability of the project, and thereby its impact.

Other improvements that can be made, but haven't been due to lack of time, regards the preparations for printing. There were many discussions along the way regarding lattices, and their effect on the properties of the model. If more time would have been found, it could have been put into a method to create an optimised lattice structure following the stress patterns of the model. This optimisation could have been taken even further by implementing a method to rotate the material while printing. As the material have very varying properties in its three Cartesian directions, the printing direction is very important for the strength of the model. If the material direction could be rotated while printing, to optimise the material directions according to the stresses, an even more optimised shape could be found. This rotation of the material does, however, need to be included in the generative design process, as it has an impact on every iteration, which makes it complex and time consuming, and was therefore put on hold for the future.

If the project was to be developed further, these are some of the points that could have been important to look into. Especially the preparations for the printing, and the rotation of material, which could greatly increase the novelty of the software. Another thing that could be improved by further development and additional time would be the use of filtering algorithms in the iterative FEM-Solver. As the project was finished a decent filtering algorithm was found through a long time of testing different algorithms. However, even if the algorithm is decent for many different kinds of project it is not perfect for all, and there are many problems that would require the use of the filter component, with some tweaking, to find the optimised solution. If even more time would have been put into the project this might have resulted in an even better filtering algorithm, and thereby a more versatile software.

Even though the project could definitely be taken further, in many ways, it is safe to assume that in some ways the goals have been reached. A software has been developed that simplifies the use of the hybrid composite material in both simple and complex situations, and the discussion about using topological optimisation and generative design in the building sector has been brought further. The project strongly promotes the continued use of more environmentally friendly hybrid composite materials, and has provided a method of decreasing the amount of material used in the building sector. It has been proven to be usable both for optimising simple building elements such as beams, cantilevers and columns, as well as for complex structures and load cases. The software greatly simplifies the creation of a load bearing structure for these complex cases and could be usable for creating structures such as balconies and protruding roof-structures for bus-stations, gas-stations and similar. Hopefully, the method and discussion in this project will have a positive impact, no matter how small, on the environment through construction methods in the building sector.

Bibliography

[1] Bærentzen, J. A., Sigmund, O. & Woldseth, R. V. (2023). Phasor noise for dehomogenisation in 2D multiscale topology optimisation. *Computer Methods in Applied Mechanics and Engineering, 418 (2024).* https://doi.org/10.1016/j.cma.2023.116551

[2] Groen, J. P. & Sigmund, O. (2017). Homogenization-based tolopology optimization for high-resolution manufacturable microstructures. *Wiley - Special issue on advanced topology optimization.* 2018:113. 1148-1163. DOI: 10.1002/nme.5575

[3] Haakonsen, S. M., Dyvik, S. H., Luczkowski, M. & Rønnquist, A. (2022, June 14). *A grasshopper plugin for finite element analysis with solid elements and its application on Gridshell nodes.* MDPI. https://www.mdpi.com/2076-3417/12/12/6037

[4] Hamdan, S., Talib, Z. A., Rahman, Md. R., Ahmed, A. S. & Islam, Md. S. (2010). Dynamic young's modulus measurement of treated and post-treated tropical wood polymer composites (WPC). *BioResources*, *5*(1), 324–342. https://doi.org/10.15376/biores.5.1.324-342

[5] Hoffman, O. (1967). The Brittle Strength of Orthotropic Materials. *J. Composite Materials.* Vol. 1, p.200

[6] Kladovasilakis, N., Tsongas, K., Kostavelis, I., Tzovaras, D. & Tzetzis, D. (2022). Effective mechanical properties af additive manufactured triply periodic minimal surfaces: experimental and finite element study. *The International Journal of Advanced Manufacturing Technology (2022), 121:7169-7189.* https://doi.org/10.1007/s00170-022-09651-w

[7] Knight, N. (2006). User-Defined Material Model for Progressive Failure Analysis. *NASA*. NASA/CR-2006-214526

[8] Krüger, R. & Wagenführ, A. (2020). Comparison of methods for determining shear modulus of Wood. *European Journal of Wood and Wood Products*, *78*(6), 1087–1094. https://doi.org/10.1007/s00107-020-01565-2

[9] Lei, H-Y., Li, J-R., Xu, Z-J. & Wang, Q-H. (2020). Parametric design of Voronoi-based lattice porous structures. *Materials & Design,* Vol 191. https://doi.org/10.1016/j.matdes.2020.108607

[10] Lopar, S. (2020). Topology optimization with optimal spatially varying length scale (Master's thesis). Lund University, Lund. http://lup.lub.lu.se/student-papers/record/9032576

[11] López Machado, N. A., Vielma Pérez, J. C., López Machado, L. J. & Montesinos Machado, V. V. (2022, March 6). *An 8-nodes 3D hexahedral finite element and an 1d 2-nodes structural element for Timoshenko beams, both based on Hermitian intepolation, in Linear Range.* MDPI. https://www.mdpi.com/2227-7390/10/5/836

[12] Malysko, L. (2011). Orthotropic yield criteria in a material model for timber structures. *Mechanics and Control.* Vol. 30, No4, 2011.

[13] Piros, A & Trautmann, L. (2023). Creating interior support structures with Lightweight Voronoi Scaffold. *International Journal on Interactive Design and Manufacturing (IJIDeM) 17.* 93-101 https://doi.org/10.1007/s12008-022-01182-8

[14] Urquhart, L., Wodehouse, A., Loudon, B. & Fingland, C. (2022, April 6). *The application of generative algorithms in human-centered product development.* MDPI. https://www.mdpi.com/2076-3417/12/7/3682

[15] Wu, J., Wang, W. & Gao, X. (2019). Design and Optimization of Conforming Lattice Structures. *IEEE Transactions on Visualization and Computer Graphics,* PP(99):1-1. DOI: 10.1109/TVCG.2019.2938946

[16] Yoshihara, H. (2012). Off-axis young's modulus and off-axis shear modulus of wood measured by flexural vibration tests. *Holzforschung*, *66*(2), 207–213. https://doi.org/10.1515/hf.2011.118

[17] Yoshihara, H., Kubojima, Y., Nagaoka, K. & Ohta, M. (1998). Measurement of the shear modulus of wood by static bending tests. *Journal of Wood Science*, *44*(1), 15–20. https://doi.org/10.1007/bf00521869

[18] Zienkiewicz, O. C., Taylor, R. L. & Zuh, J. Z. (2005). *The Finite Element Method: Its Basis and Fundamentals* (Sixth Edition). Butterworth-Heinemann.

DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY