A Generative Design of Timber Structures According to Eurocode

Development of a Parametric Model in Grasshopper

NICOLINA ANDRÉN JAKOBSSON

SIMON BOHMAN
A Generative Design of Timber Structures According to Eurocode
- Development of a Parametric Model in Grasshopper

Nicolina Andrén Jakobsson and Simon Bohman

Master Thesis, 2019
TRITA-ABE-MBT-19217
Abstract

The interest of timber structures has in recent years increased, primarily due to the environmental benefits of timber. This has created an increased demand for structural engineers with timber expertise. At the same time the concept of structural parametric design have become more popular. This new way of working with designs enables for architects and engineers to explore different geometries in early stages of a project. However, the combination of a parametric workflow and timber design have so far been limited due to the complexity of the material.

This thesis aims to create an parametric workflow within the visual programming environment Grasshopper. This enables analysis of structural design simultaneously with a cross sectional and topological optimization of timber structures. The structural analysis is performed with Karamba which is a plug-in tool to the Grasshopper environment. The design verification based on Eurocode EN-1995 have been manually scripted in python components. The parametric model have been applied to a case where the main bearing bearing of a glass roof is to be designed. Three different geometries have been evaluated with regard to cross sectional dimensions and geometrical shape.

The framework with a truss turned out to be a preferable design if only considering weight, deflection and utilization. The truss frame provides the lowest weight and the second smallest displacement. Furthermore, a comparison of the structural analysis and design have been performed with the FEM-program Robot. The compassion show similar results, increasing the reliability of the Grasshopper model and the results from this tool. It confirms it is possible to perform generative design of timber structures within the same interface.

The Grasshopper model is limited and can not handle all variations of 2D timber structures. The complexity and variation of such calculations in conjunction with the Eurocode have not been implemented during the time-span of this thesis. However, it is general within the limitations of the case study meaning a variety of frame geometries can be evaluated.

Keywords: Parametric design, Timber, Optimization, Generative Design, Glulam, Grasshopper, Python
Sammanfattning

Under de senaste åren har intresset för träbyggande ökat, främst på grund av de miljömässiga fördelarna. Detta har lett till en ökad efterfrågan på konstruktörer med kompetens inom området. Samtidigt har parametrisk design inom konstruktion blivit mer populärt. Det är ett nytt sätt att dimensionera konstruktioner och gör det möjligt för konstruktörer och arkitekter att utforska alternativa utföranden i tidiga skeden. Dock har kombinationen av parametrisk design och träkonstruktion hittills varit begränsad på grund av materialets komplexitet.


Modellen i Grasshopper är begränsad och kan inte hantera alla typer av 2D-konstruktioner i trä. Komplexiteten och variationen av sådana beräkningar i samband med Eurokoden har inte implementerats inom tidsramen för detta arbete. Det är emellertid generell inom ramen för fallstudien vilket betyder en rad olika typer av geometrier kan utvärderas.

Nyckelord: Parametrisk design, Trä, Optimering, Generativ Dimensionering, Limträ, Grasshopper, Python
Preface

This master thesis is the final work of the degree in Civil Engineering at KTH with a focus on house building and corresponds to 30 credits. The work was carried out at the company Tyréns who gave the proposal on the subject.

Our deepest gratitude to our supervisor and examiner Bert Norlin and his invaluable contribution to the structural courses at KTH. We very much appreciate his support throughout the project and the time he spent discussing the complexity of timber structures, thanks for always going above and beyond.

We also want to thank our former lecturer Anders Eriksson. His guidance regarding global stability was greatly appreciated.

Especially thanks to Johan Reissmüller at Tyréns for his assistance and his dedication for our work. His insights were truly helpful and the reason for achieving a generative design model.

Our thanks and appreciation to Tyréns and its employees for their shared knowledge and companionable work environment with interesting guest lectures and yummy breakfast buffets.

Finally, we want to take the opportunity to thank our family and friends for their support throughout our five years of studies.

Stockholm, June 2019

Nicolina Andrén Jakobsson and Simon Bohman
Authors

Nicolina Andrén Jakobsson and Simon Bohman
Structural engineering and bridges
KTH Royal Institute of Technology

Place for Project

Tyréns
Stockholm, Sweden

Examiner

Bert Norlin
KTH Royal Institute of Technology

Supervisor

Johan Reissmüller
Tyréns
Contents

Contents vii

List of Figures xii

List of Tables xiii

List of Symbols xvi

1 Introduction 1

1.1 Background ........................................ 1
1.2 Current Application .................................. 3
1.3 Aim and Scope ...................................... 3
1.4 Assumptions and Limitations ...................... 4
1.5 Used Software ...................................... 6

2 Parametric Design 7

2.1 In General .......................................... 7
2.2 Rhinoceros ......................................... 8
2.3 Grasshopper ........................................ 9
  2.3.1 Model Approach ................................ 9
  2.3.2 GhPython Component .......................... 11
2.4 Karamba 3D ........................................ 12
  2.4.1 Finite Element Method ....................... 12
  2.4.2 Model Assembly ................................ 13
  2.4.3 Analysis of Model ............................ 14
## CONTENTS

2.5 Octopus ......................................................... 14
2.6 Parametric Model ............................................... 15

### 3 Design of Timber Structures 17

3.1 Material Properties of Wood .................................. 17
3.2 Glulam .......................................................... 18
   3.2.1 Fire Safety ............................................... 19
3.3 Environmental Aspects .......................................... 20
3.4 Design According to Eurocode ................................ 21
   3.4.1 Design in Ultimate Limit State ......................... 22
   3.4.2 Design in Serviceability Limit State .................... 24
3.5 Connections and Supports ..................................... 24

### 4 Case Study 29

4.1 Sergelgatan .................................................... 29
4.2 Glass Roof .................................................... 30
   4.2.1 Steel Structure ........................................... 30
   4.2.2 Timber Structure ......................................... 31
4.3 Load Effects ................................................... 31
   4.3.1 Global Stability and Dynamic Analysis .................. 33
4.4 Model Geometry ............................................... 34
   4.4.1 Framework with Rectangular Cross Section .............. 36
   4.4.2 Framework with Single Tapered Beams .................... 37
   4.4.3 Framework with Truss .................................... 37
4.5 Optimization ................................................. 38
   4.5.1 Optimization of Framework with Rectangular Cross Section . 39
   4.5.2 Optimization of Framework with Single Tapered Beams .... 40
   4.5.3 Optimization of Framework with Truss .................... 40
4.6 Resulting Models .............................................. 42
   4.6.1 Framework with Rectangular Cross Section .............. 42
C Robot Model

C.1 Results

C.1.1 Framework with Truss Type I

C.2 Robot Calculation Report
List of Figures

1.1 Mannheim Multihalle and its hanging model. ................. 2
1.2 Workflow of used software. .................................. 6

2.1 A line created by Grasshopper components. ................. 9
2.2 Illustrates how elements are assigned an ID tag. ............ 10
2.3 Structure of cross section data. .............................. 10
2.4 Example of a customized GhPython component. ............ 11
2.5 Different FE-elements. From the left: beam element, mesh element and cubic element (Jean-Marc 2018). .................. 13
2.6 The Octopus interface, where the solution are presented in a 3D-graph and the objectives are shown at the bottom. ........ 15
2.7 Flowchart of the Grasshopper model. ....................... 16

3.1 Compression failure along the grain and perpendicular to the grain. The stress-strain diagrams displays the failure stress for each case (Kliger et al. 2016, Ch.2.4.1). ......................... 17
3.2 Tension failure along the grain. To the left it shows failure in lignin and to the right is shows failure in fibres (Kliger et al. 2016, Ch.2.4.1). 18
3.3 Two examples of common type connection for Glulam (Martinsson 2018). ............................................................ 25
3.4 Connection between struts and chord by recessed steel plates. Left figure shows the dowels and the right figure the recessed steel plates. 26
3.5 Finger jointed connection (Just, Piazza and Östman 2016, Ch.10.5). 26

4.1 Vision of Sergelgatan. ........................................... 29
4.2 Vision image of the steel frame. ............................... 31
4.3 Load actions on framework. ..................................... 32
4.4 Euler’s buckling factors (Steel Construction Manual 1987) ............... 35
4.5 Frame with rectangular cross sections ........................................... 36
4.6 Frame with tapered cross sections .................................................. 37
4.7 Draft of potential truss geometries .................................................. 38
4.8 Illustration of the bottom chord curvature ........................................ 41
4.9 Optimized cross sections and geometry for rectangular frame with and without strut ................................................................. 42
4.10 Optimized cross sections and geometry for tapered frame with and without a strut ................................................................. 44
4.11 Optimized geometry and cross sections for truss geometry I ............... 46
4.12 Optimized geometry and cross sections for truss geometry II ............. 47
4.13 The resulting structures based on their mass and maximum displacement ................................................................. 49
4.14 Maximum utilization for each member of the structure for Truss I ....... 50
4.15 Bending moment distribution. Note Karamba uses opposite signs convention compared to Robot ......................................................... 52
### List of Tables

| 4.1 | Design loads used in load combinations. | 32 |
| 4.2 | Global stability and dynamic requirements for the case study. | 34 |
| 4.3 | Buckling length factors for each part of the structure. | 35 |
| 4.4 | Objectives of the optimization. | 39 |
| 4.5 | Genomes used in optimization for the rectangular frame with and without strut. | 39 |
| 4.6 | Genomes used in optimization for tapered frame with strut and without strut. | 40 |
| 4.7 | Genomes used in optimization for frames with truss design. | 41 |
| 4.8 | Optimized genomes for rectangular frame with and without a strut. | 43 |
| 4.9 | Resulting objectives. | 43 |
| 4.10 | Optimized genomes for tapered frame with and without a strut. | 45 |
| 4.11 | Resulting objectives. | 45 |
| 4.12 | Optimized genomes used for truss design I and II. | 48 |
| 4.13 | Resulting objectives for the truss structures. | 48 |
| 4.14 | Section forces affecting the connection or support. | 51 |
| 4.15 | Section forces affecting the truss connection. Negative values mean compression. | 51 |
| 4.16 | Comparison of results from Karamba and Robot. The equations refer to the design equations in 3.4. \(^*\text{Note } k_h = 1.10 \text{ has been used in Robot.}\) | 53 |
| 4.17 | Natural frequency and global buckling factor of the first mode. | 53 |
List of Symbols

\( \gamma_d \) Partial coefficient for safety class
\( \gamma_G \) Partial coefficient for permanent load
\( \gamma_Q \) Partial coefficient for variable load
\( \psi \) Factor for combination value
\( \sigma_{c,0,d} \) Design compression stress along the grain
\( \sigma_{m,y,d} \) Design bending stress about the y-axis
\( \sigma_{m,z,d} \) Design bending stress about the z-axis
\( \sigma_{t,0,d} \) Design tension stress along the grain
\( \tau_d \) Design shear stress
\( \xi \) Reduction factor for permanent load
\( f_{c,0,d} \) Design compressive strength along the grain
\( f_{m,y,d} \) Design bending strength about the y-axis
\( f_{m,z,d} \) Design bending strength about the z-axis
\( f_{t,0,d} \) Design tension strength along the grain
\( f_{v,d} \) Design shear strength
\( g_{beam} \) Self weight from glulam beams
\( g_{roof} \) Self weight from glass roof and its installations
\( k_m \) Factor considering re-distribution of bending stresses in a cross section
\( k_{c,y} \) Instability factor
\( k_{c,z} \) Instability factor
\( k_{crit} \) Factor used for lateral buckling
\( k_{m,a} \) Factor reducing bending strength based on the tapered angle
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Q_{Ed,beam}$</td>
<td>Load combination for beam</td>
</tr>
<tr>
<td>$Q_{Ed, column}$</td>
<td>Load combination for column</td>
</tr>
<tr>
<td>$s$</td>
<td>Snow load</td>
</tr>
<tr>
<td>$w$</td>
<td>Wind load</td>
</tr>
<tr>
<td>$w_{creep}$</td>
<td>Creep deflection</td>
</tr>
<tr>
<td>$w_c$</td>
<td>Precamber</td>
</tr>
<tr>
<td>$w_{inst}$</td>
<td>Instantaneous deflection</td>
</tr>
<tr>
<td>$w_{net, fin}$</td>
<td>Net final deflection</td>
</tr>
<tr>
<td>$L$</td>
<td>Length of beam</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background

Parametric design involves designing parameters which in turn defines a model, in other words it is about designing relations in a model rather than designing an static model. This allows to explore new ways of design which is not possible through a conventional design process. However, it is not the only reason for choosing this work method. An advantageous use of parametric design is linking the parametric model to a structural analysis. One can then manipulate or change a design by its parameters and then quickly get a perception of how this would impact the structural behaviour.

Currently this method is available to implement for commonly used structural materials within the Grasshopper environment, but there is no built-in component for calculation of timber material. Timber has been on the up rise during the last years and the demand for timber structures has increased. The focus on environmentally friendly buildings and interest in natural building materials have both been contributing to the increased demand. Additionally, in 2018 the Swedish government published a document where they initiate a goal of sustainable constructions and where timber is an important contributing factor to reach that aim (Eriksson 2018).

The reason why timber has not been favoured before is the lack of knowledge in the building industry according to Johan Fröbel (2016) at Svenskt Trä. He argues the main reason is due to a misconception regarding the fire safety. Building materials such as timber, concrete and steel are in fact equal in the aspect of fire safety. It is of great importance to use proper building techniques to prevent fire spread. It was the lack of qualified engineering which caused a large number of city fires earlier in history. Up until 1994 there was a ban in Sweden for timber buildings with more than two floors. Even tough the ban was raised over twenty years ago, research and expertise is still lacking behind compared to other materials. Today there are developed methods for managing fire spread but wood remains a complex material. The structural design of timber differs from calculation of steel or concrete since wood is an orthotropic material. This means the material properties differ for each
direction; along the grains and perpendicular to the grains in radial and tangential directions. Considering these aspects of varying properties, the verification of the structural design is more complicated and further advancement to iterate a solution with generative design could be problematic and time consuming.

With new technology the design of a structure can be made more efficient with better utilization of the elements, lower weight and smaller dimensions. This will in turn result in less material consumption. This can be achieved by using the tools of parametric design through algorithms and manipulation of variables. However, parametric design is not a new innovation. It could be traced back in history to ancient buildings and structures such as the pyramids. It was during that time used by architects which often had knowledge in engineering and mathematics. One of the earlier methods to find complex forms and arches was to use a “hanging chain model”. A miniature scale model of the structure was made upside down to let the self-weight create pure tension shapes, which in reversed form create a pure compression shell that could be built in a larger scale (Philips 2010).

One of the most famous architects who used this method was Antoni Gaudí who designed La Sagrada de Familia in Barcelona, Spain. A more modern example and made out of timber is the Mannheim Multihalle in Germany. The architects Carlfield Mutschler and Winfried Langner designed this pavilion in a grid shell concept along with the architect Frei Otto, who had experience within the field. It resulted with a large curved timber grid shell construction that was and still is the largest structure of its kind ever made, see Figure 1.1.

![Mannheim Multihalle and its hanging model.](image1.jpg)

**Figure 1.1:** Mannheim Multihalle and its hanging model.

With parametric design new shapes can be achieved. This have created a new architecture style within contemporary avantgarde, called parametricism. As the design keeps evolving along with more advanced computations, the style continues to elaborate. Advanced structures which were impossible to create in the past are now possible, leading to endless variations of designs and increased utilization of materials, such as timber.
1.2 Current Application

The use of parametric design in structural engineering is not yet widely spread and whenever used it is often in complex projects with customized solutions for a specific project. It requires a lot of time to develop a parametric model, however it could still be useful in the end. Dr. Feng (2018) wrote about a few case studies of when both architects and engineers used parametric design in the early stages of projects in the book *Design and Analysis of Tall and Complex Structures*. Based on these case studies he concludes that by using a parametric model the design process becomes more manageable by enabling a easy way to handle small changes. Instead of manually redrawing a model changes can be made by modifying the input values which in turn redraws the model automatically.

In Sweden there is an interest in using parametric design within the building industry and a few companies are now investing in it and trying to apply it in projects. At Tyrens there is a team (FoU) focusing on how to spread the knowledge and implement the use of these techniques in an engineering work flow. The ambition with this investment is to create effective methods for employees to analyze complicated structures in detail, increasing the comprehension of structural and architectural qualities. One of the challenges is to establish knowledge among employees which requires resources, such as education and software licences. This does not only apply for those with expertise but for other disciplines too, as it would affect the collaboration. For example clients would need to have a comprehension for this kind of workflow to understand the opportunities and if it would be useful in their project. Creating a parametric model is time consuming but increases the flexibility. This means that it could in fact save time if there are many small adjustments that arises late in the project.

1.3 Aim and Scope

In recent years the method of parametric design has shown it is possible to bring architects and engineers closer in early stages of projects. Having a structural analysis follow the manipulation of a model offers easier communication and collaboration between architects and engineers. This give rise to new approaches for the design process and an opportunity to explore a variety of outcomes.

It is already possible to achieve a structural parametric model of a steel structure by using a plug-in tool to the software Grasshopper called Karamba 3D. However, this is not yet supported for timber due to its orthotropic features. Since timber structures now are on the uprising there is a new interest for a greater market and an inquiry for expertise.

There is a lack of competence in both timber structural engineering and in parametric design. Wherefore the goal of this master thesis is to create a parametric tool to make it more convenient to analyse, design and optimize timber structures. This is
implemented through a case study of a glazing roof structure with a timber frame located at Sergelgatan, Stockholm. However, the tool is aimed to be applicable to any kind of 2D timber structure. 3D structures is not considered since it is not necessary for the case study.

The structure should be able to carry its self-weight, load from roof installations, snow and wind load. How these forces act on each structural part is calculated through components in the used software. The structure is then checked against its structural capacity and instabilities. This design process is based on Eurocode 5: Design of timber structures EN 1995 1-1 and are built as script in Python components inside Grasshopper. The model is developed for a specific case, but is written with a general code which is applicable for other timber structures based on the same identification system. Furthermore, a verification of the validity for the built-in computation and design capacity is made by a comparison with a conventional FEM-software, Robot by AutoDesk.

In addition to the investigation of using tools for parametric design for timber material, problems regarding optimization is examined. There is a discussion of which parameter should be taken into consideration when optimizing a structure.

1.4 Assumptions and Limitations

In this master thesis the focus revolves around a case study consisting of a glass roof structure. Based on its prerequisites a parametric tool which is used to obtain a first draft of the shape and dimensions of structures is established. This tool can be used in early stages of a project to get a perception of possible designs for a framework of timber for any spans lengths and heights. The model only perform basic checks of the design, hence if the structure is planned to be constructed it is vital to proceed with a more profound analysis. A thorough dynamic analysis is not covered, instead a verification against defined constraints is performed. Also, the connections and supports is not a part of the optimization. Potential standard connections which would work with given structures are checked to withstand the forces acting on them. These features are possible to add in a further development of the parametric model. In addition, this report does not present a detailed description for all the used components in Grasshopper and Karamba, for further explanation of each component see associated manual (Preisinger 2012).

Since only a simplified analysis is presented in this report, some parts are excluded in the calculations and therefore the result does not include complete construction documents and drawings. These assumptions and limitations may affect the reliability of the model.

The assumptions related to the structural model for the case study are listed below.

- Lateral bracing in the glass roof is assumed to stabilize the beam and/or top chord depending on the type of geometry, i.e. it prevents the structure from lateral buckling perpendicular to plane.
1.4. ASSUMPTIONS AND LIMITATIONS

- Load from the glass roof is applied as an uniformly distributed load. In the actual case this load would be applied as point loads from the bracing.

- Wind load is only applied as an uniform constant load at one side of the framework since it is an open construction without walls on the other side.

- The connections between columns and the span configuration (beam or truss) are assumed to transfer all forces.

- The supports are assumed to be fixed to the ground.

The limitations related to the Grasshopper model are listed below.

- The model only operates on 2D structures.

- The model only handles solid rectangular cross-sections.

- The model only manages glued laminated timber of the types GL28c and GL30c, which are the two most commonly used. Other materials could be added if needed.

- Connections are not considered or optimized in the parametric model. It is however possible for the user to choose how the connections behave in terms of degrees of freedom and stiffness.

- The global stability analysis covers the computation of natural frequencies and global buckling factors. These parameters are constrained by predefined values in the optimization process.

- Shear deformation is not considered. This depends of the choice of using beam elements in the structural analysis which does not provide such calculations. Beam elements requires less computing capacity and are easier to handle when modelling simple structures consisting of columns, struts and beams compared to three-dimensional meshes.

- The geometry of a frame is developed for the case study but could be replaced with another geometry, based on the same indexing system described in Chapter 2.3.1.

- Only vital checks are verified in the model. The used chapters from the Eurocode EN 1995 with the Swedish annex are presented in Chapter 3.4.

- Calculation of fire resistance is not included in the model.

- In the model there is an option to choose which members to check for lateral and column stability. If the member is not chosen the design modules assume the member is braced. However, the FE-program Karamba still calculate them as none braced if the boundary conditions of the model is not changed.
1.5 Used Software

In this thesis several software are used together in a linked workflow. A 3D-modelling program called Rhinoceros 6 (Rhino 6) is used for visualization of the model. On top of Rhino 6 the graphical scripting interface called Grasshopper runs, which is the design environment where the parametrization of the model is created. The structural analysis and design is performed in the Grasshopper interface but through the plug-ins Karamba 3D and python scripts. The optimization loop runs through a plug-in called Octopus, which offers optimization of several parameters. The linkage between the software is presented in Figure 1.2 and a more thorough explanation of each application is explained in Chapter 2.

Figure 1.2: Workflow of used software.
Chapter 2

Parametric Design

2.1 In General

The word parametric origins from mathematics, where an equation is considered parametric if a change of an independent variable changes the outcome of that equation. In the realm of civil engineering and design Luigi Moretti, a pioneer of parametric design defined it as "the relation between the dimensions dependent upon various parameters" (Bucci 2002). In other words, parametric design consists of algorithms linked together to form geometries. An algorithm is an unambiguous recipe or description built on strict logic. In order to work, an algorithm need to have a distinct definition of its inputs. Therefore, the number of inputs and types need to be clearly defined (Tedeschi 2014).

Since it is the parameters which are designed and not the structure itself one is not limited to only evaluate one design, instead multiple designs may be evaluated. This approaches a more integrated way of working with designs, by merging different means of the design process closer together in early stages of a project. It allows an architect to explore possible structural outcomes and a structural engineer to explore shape finding.

Parametric modelling does not only enable opportunities for architects to advance their design but also for structural engineers to virtually analyze shapes and automate their computations. In conventional design, a designer creates a design of a model which the structural engineer has to adapt to. This initial model is limiting, and small changes can require redraft and time consuming modifications. It is an ineffective process which might lead to a disappointing outcome for both parts. The first draft may have an appealing design but might be too challenging and expensive to construct. The design process is restricted with this type of work method and does not allow further exploration of the design without creating additional work. Instead, parametric design can be used to optimize the design through computational strategies. However, it requires the designer to get involved in the sense of structural analysis since conventionally architectural design does not necessarily require technical knowledge. Often, it is not a demand within the disciplinary and
is usually not included in the education (Menges and Ahlquist 2011). Parametric design is a new way of working with designs which encourage new innovative shapes and designs. In turn this leads to greater proficiency requirements of designers in math and computer science, including skills in programming.

According to Menges and Ahlquist (2011) the lack of knowledge within the profession is not vital, instead they state “that the main challenge does not lie in mastering computational design techniques, but rather in acculturating a mode of computational design thinking”. Hence, the key does not only lay in education of programming language nevertheless it is about solving a design problem with a new mindset. In fact, it resembles with learning and be able to speak a new language, in this case programming language.

There are various software on the market which can be used for a computational design process. For the composed parametric model presented in Chapter 4, the 3D visualization application Rhino 6 with the programming interface Grasshopper were selected. Tyréns proposed using these on the account of the skill level among their employees. The choice were however based on the advantages of large accessibility to help forums and ability to customize new components by writing own scripts. These components are necessary for the design of timber structures since these functions does not yet exist within the Grasshopper interface. The script is written in Python code as own components in the visual programming software called Grasshopper, which outcome in turn is displayed in Rhino 6.

By using parametric design there is potential to create environments where structures are being optimized for individual locations considering its prerequisites. The challenge is to determine the weight distribution among the parameters since opinions on the goal might differ. This new approach of working with design emerge new encounters of whom should make these decisions.

2.2 Rhinoceros

Rhinoceros, also known as just Rhino, is a Computer Aided Design (CAD) software that visualizes geometry, either modelled directly or as a result of scripted code through plug-ins. It enables the creation of parametric relations between elements and can generate a variety of geometrical shapes.

Rhino is developed by an American company Robert McNeel & Associates and was first released in 1998. It is a modelling tool with high accuracy, that elaborates forms through so called NURBS, Non-Uniform Rational B-Splines. This means the shapes of curves and surfaces are generated by mathematical expressions of fixed points in the three-dimensional(3D) space. With these equations various amount of shapes can be created and compared, providing a flexible workflow with smooth transitions between points. Additionally, by using these mathematical representations of the shapes it requires less amount of storage space (Cheng 2008).

Rhino 6 is the latest version of the program and is used for the case study. The
2.3 GRASSHOPPER

reason why Rhino is chosen for this case is because it is user friendly and offers a variety of functions, one of them being the freedom to create own components through its plug-ins. This makes it possible to create and modify, and then analyze structures. The models can thereafter be exported in DWG-format to be used in other CAD or FEM software.

2.3 Grasshopper

Grasshopper is a graphical programming environment which is included in Rhino 6. It has been developed by David Rutten at Robert McNeel & Associates, the same company developing Rhinoceros. It consist of programming components, called nodes which are connected by wires to create scripts where data flows from left to right. It is therefore not possible to create loops but is suitable for generative modeling, in other words generate results and evaluate iterations to find optimal solutions. Since the models created in Grasshopper are always live, it makes it possible to manipulate the model and instantly see the response in Rhino 6 (Tedeschi 2014).

2.3.1 Model Approach

By coding a geometry the user can investigate different configurations easily and precise. The simplest example of this is a line connected by two points which coordinates can be manipulated through number sliders, seen in Figure 2.1. The two points are connected to the line component which creates a line from start point A to endpoint B. By manipulating any of the coordinates the length, placement and direction of the line can be changed. The line can therefore be said to be parametric.

Grasshopper gather multiple data in lists or trees. In a model it is important to have control over which kind of data types are used and in what format these are. A list is a set of data which is assigned to a certain index, while a tree structure

Figure 2.1: A line created by Grasshopper components.
may be seen as lists within lists where a list can branch out to different lists. Trees can be a powerful tool when handling large sets of data. By using tree structures the geometry can be divided into segments, enabling the code to run and produce reliable results for different geometries. This is applied in the model by structuring the data with an ID tag system to map all necessary input data to each element.

First, the different elements in the structure are assigned a specific ID tag describing which type of structural part the elements is associated to i.e. if it is a beam, column, strut or chord. An example of such a configuration is shown in Figure 2.2, where three lines are assigned to its associated ID tag. For instance, Column 1 consisting of one line is assigned \textit{C1} as ID tag since both of them are placed under list unit \{0\}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2_2.png}
\caption{Illustrates how elements are assigned an ID tag.}
\end{figure}

Each type of part is divided into a number of beam elements numbered 0-n, where n is the total number of elements in the structure. Every part will have length, cross-section parameters, ID tag and acting forces that are linked to the ID for that element. The different parameters can thereby be matched to each other. In Figure 2.3 it shows how material properties and cross-sections are assigned to the different structural parts.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2_3.png}
\caption{Structure of cross section data.}
\end{figure}
2.3.2 GhPython Component

In Grasshopper there are incorporated script components, one of them being GhPython with ability to execute data from Grasshopper components. When the assortment of built-in components in Grasshopper can not perform the desired operation, python components offers the user to write own functions (Nagy 2017).

Python is a high-level open source programming language that is widely spread and have support in a wide range of environments. The language is designed to handle complex executions with simple, comprehensible syntax. On the other hand it has a tendency to not be as efficient as other programming languages, but in general this is not a vital issue when using it in Grasshopper. Python script is used within several disciplines in science and therefore has a large support community, with many tutorials for beginners. By scripting custom python components the work flow can be redirected and complex operations ca be performed more efficient than with built in Grasshopper components.

In order to achieve an automated parametric tool for timber structure the design rules of Eurocode have been translated into user defined objects using the GhPython interpreter. Since the aim is to quickly estimate the structural behavior, the scripts covers Section 6 and Section 7.2 in *Eurocode EN 1995 Part 1-1*. These sections covers calculation in the ultimate limit state (ULS) and the deflection under serviceability limit state (SLS) conditions.

Section forces, stresses, strength properties and Element ID:s are set as inputs to the Python components in list formats, see Figure 2.4. The inquired parameters can be found through their matched list index and sorted based on their ID tags. The calculations will run item by item in for-loops with if-statements, which makes sure the correct cross section parameters are used in conjunction with the forces acting on that particular part.

![Figure 2.4: Example of a customized GhPython component.](image)

Further, in Figure 2.4 several outputs are shown. It is up to the user to choose which output to use by redirecting the connected wire. The picture shows the com-
ponent for verification of combined bending and axial compression or axial tension. The output view is consistent for the customized components, with a result list for all elements, a list of elements which exceeds their capacity and utilization ratio for all elements. For instance, by looking at element 7 it displays it is part of a column (C2), it is checked for combined bending and tension and the result of the verification (OK/NOT OK).

The python scripts, presented in Appendix A.1, are divided into three main sections. In the first section all inputs and outputs are described. In the second section modifications are made to the inputs and the outputs are defined. In the third section the computation runs element by element and their results are presented as lists. The outputs and the overall structure of the code is the same for all components for easier handling.

2.4 Karamba 3D

Karamba 3D is a Finite Element Analysis (FEA) plug-in to Grasshopper which make use of its graphical interface. Karamba is developed to be undemanding and to give an early understanding of structural responses with a seamless connection between geometry and analysis. A change in the geometry will give a live response in the analysis, compared to a conventional analysis which is done for a certain state. The short computing time of Karamba makes it possible to explore geometries both manually or in conjunction with optimization tools (Preisinger 2013).

2.4.1 Finite Element Method

If a structural problem is to complex to calculate by hand, computer calculations with finite element method (FEM) is often used. FEM is a general method which can be applied on almost any problem.

The method solves structural problems through several steps, where the first is to divide the structure into many small segments. There are three main types of elements a structure can be divided into; one dimensional rods or beams, two dimensional meshes or three dimensional cubic elements. These are shown in Figure 2.5. The accuracy of the solution depends on the number of elements. The more divisions the higher accuracy of the model. However, more elements will lead to longer computation time. On the perimeter of the elements there are nodes, for every node in the element there is an equation describing its place in the coordinate system. By relating displacements to stresses and further to potential energy a stiffness matrix for that element is created. This is done for every element in the structure.
The equation system of each element are combined together with the elements it shares nodes with, creating a system of equation which is linked together through the whole structure. When the FE-program solves this system of equations it will eventually solve the problem in one node, using this solution as a key to solve the rest of the system (Reddy 2005). While the method can be carried out by hand for simple problems the system of equation quickly expands with more complex problems making it impractical to solve by hand.

In Karamba, beam elements are defined according to the elastic beam theory, i.e. Bernoulli beams. Beam elements which are adjacent share one or several nodes making them dependent of each other. Each of these nodes consists of six degrees of freedom (DOF), enabling movement and rotation in every direction.

2.4.2 Model Assembly

An FE-model can be created with Karamba by converting the geometry built by Grasshopper components. The model is assembled through a Karamba component which require some input data for the model to be able to perform a structural analysis. The entities of the Assemble-component are defined by the following:

- **Beam Elements**
  Beam elements are created with the Line to Beam-component which translate line segments to connected beams. The beam elements are defined in a list, which is compatible with components in Grasshopper.

- **Support**
  Supports in the model can be defined at given nodes as completely fixed in some or all directions or as a pinned connections.

- **Load**
  Loads can be defined as uniformly distributed or as point loads which act in a direction global or local to elements. In the model, a python script calculates...
the design combination value for each load according to Eurocode EN1990-0 & EN1990-1 and Swedish standard BFS 2009:16 EKS 5.

- Cross-section
  Each structural part in the model can be assigned a specified cross-section. Properties such as type of cross-section, material, color, and dimensions are defined by the user.

- Material
  Material properties are defined by a component which offers selection of commonly used materials. Properties for timber need to be specified by the user.

- Joint
  The translations and rotations for a joint can be set to zero or be given a stiffness based on the connection type. Depending on if they are elastically restrained or not.

2.4.3 Analysis of Model

Karamba can perform both first and second order analysis of structures by computation of the assembled model. From the result one can view the analyzed model, observe deformation of the structure, or view distribution of stress and strain. The result for each element is also presented in lists, which can be sorted by the user to examine how each structural part is affected. There are also other Karamba components which can provide eigenfrequencies and buckling modes for the structure.

2.5 Octopus

Octopus is an evolution theory-based optimization plug-in to Grasshopper which uses genetic algorithms. Octopus is capable of multi-objective optimization, meaning the solution can satisfy more than one objective and can find optimums between several objectives.

A genetic algorithm tries to mimic natural selection as it happens in nature. A genetic algorithm in its basis consist of genes and objectives. The genes are the parameters which can change a system, while the objective are the results of that change. Example, a gene could be the thickness of a beam and the objective could be the deflection of the same beam. The deflection will not change unless the system changes, thus a variation in the gene have occurred.

During an optimization procedure with Octopus the objectives are minimized unless the user chooses to redefine the fitness value. It means Octopus strive to make the objectives as small as possible, i.e. head for results close to origo. If an optimal solution is to reach the number \(1\) the objective should be described as \(\text{abs}(1 - x)\) where \(x\) is the variable which should reach \(1\).
A number of genes creates a genome. The algorithm generates a large number of genomes and analyses how well they fulfill the objectives. The algorithm then keeps the best solutions (genomes) and discards the unfit solutions. The next generation of genomes are now based on the best from last generation. This process is repeated until the solution converges to a certain outcome or until it is stopped by the user. In Figure 2.6 the interface of Octopus can be seen. In the middle all current solutions are shown in the 3D graph and at the bottom part the objective with its respective axis are shown.

![Figure 2.6: The Octopus interface, where the solution are presented in a 3D-graph and the objectives are shown at the bottom.](image)

When performing any kind of optimization, the parameter to be optimized and which parameters that are allowed to be changed (genes) need to be decided. Depending on these decisions the result might vary substantially. It is not always trivial what parameters to optimize. For example, the objective to minimize displacement and utilization will lead to a structure where the mass increases to infinity. Because the objectives is highly dependent on the same parameter, mass. However, by choosing two contradicting objectives such as displacement and mass the optimum will be a compromise between the two.

## 2.6 Parametric Model

The optimization of a timber structure is done through a combination of a parameterized geometric model, finite element computation and genetic algorithms for an optimization process. The programs, methods and concepts described earlier in this
chapter are all a part of a generative design workflow, which is shown in Figure 2.7. The modules represent the different definitions in the model, the black arrows shows how the data flows and the red arrow the genes from the optimization process.

First, the inputs are defined by the user such as geometrical data, loads, boundary conditions etc. Based on the inputs, the geometry is established and the structural preconditions which are needed for the FE-calculations are applied. All geometrical and structural data are gathered by the assemble component.

From the assembly Karamba analyzes the model and extract the results, which are sorted and sent to the design module. Python components perform prerequisite checks according to Eurocode which are defined in Chapter 3.4. The results from the design module will run through the optimization module. A set of objectives decides what the model should strive for and will be induced by the genome. Chosen genes will generate new values as input and the whole computation will start over again. This creates an iterative process which will continue until it converge to a fulfilling result. The Grasshopper script for the developed parametric model can be found in Appendix B.1.

**Figure 2.7:** Flowchart of the Grasshopper model.
Chapter 3

Design of Timber Structures

3.1 Material Properties of Wood

Wood is an anisotropic and orthotropic material meaning it has different material properties in all three directions. Along the grains and perpendicular to the grain, in both transverse and radial direction. The material is composed by long fibre cells oriented in a mesh and bonded together by lignin, a natural adhesive included in the cell walls of plants (Kliger et al. 2016). The fibre mesh works like a pack of straws, tightly bond together. Consequently, the highest strength occurs along the grain of the material and the weakest occur perpendicular to the grain. The strength properties are therefore depending on which surface the forces are acting on, but it also depends on whether the force acts in compression, tension or shear.

Compression can act either along the grain or perpendicular to the grain, which is shown in Figure 3.1. Along the grain forces are taken by the strength of the stiff cellulose fibres. When the maximum capacity of the fibres is reached it will buckle. Perpendicular to the grain the fibres will be crushed which occurs at a much lower stress compared to the strength along the grain (Kliger et al. 2016, Ch.2.4).

Figure 3.1: Compression failure along the grain and perpendicular to the grain. The stress-strain diagrams displays the failure stress for each case (Kliger et al. 2016, Ch.2.4.1).
For tension strength parallel to the fibres there are two possible failure modes, both shown in Figure 3.2 below. Either the lignin fails or the fibres are pulled apart. The lignin will fail at much lower stresses in the perpendicular direction. Both failures are often brittle, why design of timber members in tension should be done cautiously.

![Figure 3.2: Tension failure along the grain. To the left it shows failure in lignin and to the right is shows failure in fibres (Kliger et al. 2016, Ch.2.4.1).](image)

Shear will occur radial or tangential in the longitudinal direction where the radial has the lower strength of the two. There is also shear perpendicular to the grain, where at failure the fibres roll over each other (Kliger et al. 2016 Ch.2).

Wood is very sensitive for variation of moisture content. It is because water is bound to the cell walls of cellulose which causes swelling. The greatest effect will be seen in the tangential direction which has almost twice the swelling compared to the radial direction. The effect in the longitude direction is in comparison low. Furthermore, the overall strength of wood decreases with a higher moisture content (Kliger et al. 2016 Ch.2).

Since wood is an organic material it will degrade over time if it is not treated in a correct way. If the structure is unprotected from weather and not treated it will become grey overtime due to the deterioration of the outermost lignin layer. However, this does not affect the strength of the structure. It is also important to take the right precautions against mold, rot and insects.

### 3.2 Glulam

Glued Laminated Timber is a refined timber material where two or several sawn boards of wood are glued together creating a very stiff structural component which can have a wide range of dimensions and lengths. Because of the multiple laminations, the impact of material imperfections affecting the performance such as twigs are low. Since the risk of several imperfections occurring at the same section is unlikely (Gross, Crocetti and Fröbel 2016, Ch.1). This leads to a smaller variation of the material strength, i.e. smaller reductions of its strength properties compared
3.2. GLULAM

to non refined timber. Glulam is therefore suitable for cases where heavy load or long spans are involved.

A glulam beam can have a wide range of dimensions. The maximum length of a glulam element in Sweden is about 30 m and is limited by both manufacturing and the transportation possibilities. The cross section of the element is dependent on the dimensions of the sawn boards which are glued together. The width of a board are seldom wider than 215 mm because of the sizes of the sawn boards. Larger widths are possible if several boards are glued together side by side. The height is usually a multiple of 45 mm, which is the standard height of a board and is limited to about 2 m due difficulties in manufacturing. Nevertheless, it can be overcome by gluing glulam beams on top of each other (Gross, Crocetti and Fröbel 2016 Ch.1).

There is a wide range of glulam classes depending of the required performance and in which climate it is used. Two examples are GL28h and GL30c, where ”GL” stands for glued laminated. The numbers ”28” and ”30” are the characteristic bending strength and the last letter represent the composition of the beam. An ”h” stands for homogeneous, meaning the whole section is composed of boards with the same strength properties. A ”c” means combined, where the uttermost boards are of the given strength class while the inner boards are weaker (Gross, Crocetti and Fröbel 2016, Ch.1). The combined beam is often an economical choice while the homogeneous beam have the greater performance.

3.2.1 Fire Safety

Previously directives from EU stated requirements for non-flammability but is now replaced with performance requirements. These include requirement for bearing capacity, integrity, compactness and isolation. This implicates a structure must maintain its bearing capacity at a certain exposure of fire. It also sets restrictions on structural parts such as walls and floors. The can not crack open for fire to spread and furthermore they must isolate the heat from dispersion (Östman 2016).

Naturally wood functions as fuel for fire, but it also possesses qualities which make timber construction still perform well under these conditions. When wood is exposed to fire, an outer layer of charcoal is formed which isolates from the heat. Right next to the charcoal a thin layer of pyrolyzing, also called as carbonization zone, is formed and plasticizes the wood, i.e. deforms when applied to constant loading (Just, Piazza and Östman 2016). This layer is only a few millimetres thick and will not affect the overall bearing capacity since the inner layer consist of unaffected wood.

SP Fire research in Norway performed tests on glulam construction and the result showed glulam is equally other building material regarding fire safety (Andersen 2017). It also has the advantage of a slow and predictable time lapse. The combustibility of a building material should therefore not be the main issue, but rather the functionality of the material in case of fire. Thus, it is a good reason to choose glulam as construction material, unlike what many people think.
3.3 Environmental Aspects

A growing tree absorbs carbon dioxide from the air which is included in constituents for the structural components. The process require energy which is absorbed from sunlight. These are the most important parts of the photosynthesis where oxygen is released back into the air. During its lifetime the tree will store carbon dioxide. When the tree decays the carbon will go back to the atmosphere and the circle is closed. This carbon dioxide is often referred to as biogenous carbon (Larsson et al. 2016) and as long as there is sustainable forestry (biogenous carbon in balance) wood may be seen as a close to carbon dioxide neutral material over its lifespan.

In 2016 KTH and IVL published an extensive environmental life cycle analysis of an apartment building with a timber frame structure with the foundation and elevator shafts in concrete (Larsson et al. 2016). The report makes a comparison with an apartment building made of concrete which is built under similar conditions. The two houses are theoretical modified in order to be comparable. The comparison shows that the carbon dioxide footprint during the construction phase (A4-5) is almost twice as large for the concrete structure. While the other phases during the lifespan of the buildings is almost identical. This indicates timber is as a climate friendly material in bigger projects and supports the claim that wood is important in the strive of a sustainable building industry.

Steel and timber are more similar than concrete and steel. Both steel and glulam can be used in the same kind of load bearing systems, wherefore they are interesting to compare. Steel is both stronger and a more homogeneous material than glulam. This make it possible to have slimmer constructions. However, the environmental impact from the production phase of steel is much higher than for timber. Manufacturers may, if they like, release an Environmental Product Declaration (EPD) which presents the result of a life cycle analysis in a short format. An EPD does not always include the whole life span of a product but rather parts of it (Boverket 2019). The company Martinsson have released an EPD for their glulam products that covers the extraction of the raw material until the packing of the finished product. The carbon dioxide equivalents released for each ton glulam is 39 kg CO\textsubscript{2}, neglecting the biogen bound carbon in the timber (Erlandsson 2015). For the same system limits SSAB have a EPD covering their structural steel where each ton steel emits 2490 kg CO\textsubscript{2} (Soininen 2016). The numbers can however not be compared directly because there are more aspects which is not included. One example is the lifespan of the product and what happens at the end of its lifespan. Both of these factors will have a great influence of the environmental impact of the product.

If the compression strength is taken into account the equivalent carbon dioxide footprint per compression strength is 2490/355 \approx 7.0 \text{ kg CO}_2 \text{ per MPa} for steel and for timber it is 39/24.5 \approx 1.6 \text{ kg CO}_2 \text{ per MPa}. This means the difference might not be as vital as one may think.
3.4 Design According to Eurocode

The Eurocode states a set of rules and guidelines to follow in order to achieve structures with predictable performance and which reaches a satisfying safety during its lifespan. The Eurocode contains ten different sections, covering different materials and types of structures. The Eurocode is written as Python code in the model. These script are found in Appendix A.1.

The countries within the European Union (EU) have some freedom to put their imprint on how to follow the codes in the specific country. It mostly revolves around the safety parameters, in Sweden these supplements are stated in the Swedish annex. The Section \textit{EN 1995 Design of timber structures} covers the design of civil engineering structures of timber. \textit{EN 1995} should be used in conjunction with \textit{EN 1990} and \textit{EN 1995}. The used sections from Eurocode are presented and described below.

- EN 1990 Eurocode 0: Basis of structural design
- EN 1991 Eurocode 1: Actions on structures
- EN 1995 Eurocode 5: Design of timber structures
  - Part 1-1: General – Common rules and rules for buildings
  - Part 1-2: Structural fire design
  - Part 1-3: Design of bridges

\textit{EN 1990} describes the basic rules and principles of structures. It sets the requirements regarding safety, durability and serviceability of civil engineering structures. It also describes the basis of design according to the Eurocode. This section defines the limit states of a structure, i.e. when the structure cannot longer reach the performance requirements. The limit states are divided into serviceability limit state (SLS) and the ultimate limit state (ULS). The SLS covers the functionality and user comfort of the structure, affected by deformations, vibration cracks and other factors. The ULS is the limit before the structure loses its structural ability. It handles failure, collapse, fatigue and the loss of equilibrium. The actions affecting a structure in the different states are determined through load combinations where the likelihood of different actions happening at the same time are weighted. In addition, \textit{EN 1990} works as a basis for structures with materials which are not covered in the other sections of the Eurocode.

\textit{EN 1991} contains descriptions of how the general expected action might affect a structure. It consists of four main parts divided into sub parts. Each part handles a specific type of action and how these are to be considered. As for example it concerns actions from snow, wind and traffic load, accidental actions from impact and explosion and also action load that can arise during construction.
EN 1995 present methods on design of timber structures and is divided into three parts. Part 1-1 applies to the general structural stability of load bearing structures in both SLS and ULS. The first chapters covers material parameters and safety factors, while chapter 5 to 9 applies to the structural design of structures. Part 1-2 considers structural fire safety and part 2 applies to bridges of timber materials.

### 3.4.1 Design in Ultimate Limit State

The ultimate limit state concerns the safety of the structure and the safety of people. Following checks are regarding the structural capacity.

The design of cross-sections subjected to stress in one principal direction. A member can be exposed to either axial compression or axial tension depending on the acting forces. In general columns and vertices are subjected to compression. Overall, the compressive strength is greater then the tensile strength for timber since it is affected by twigs and other disturbance of the fibers.

- **[EN 1995-1-6.1.2]** Tension parallel to the grain.
  \[ \sigma_{t,0.d} \leq f_{t,0.d} \quad (EC_5 6.1) \]

- **[EN 1995-1-6.1.4]** Compression parallel to the grain.
  \[ \sigma_{c,0.d} \leq f_{c,0.d} \quad (EC_5 6.2) \]

In the perpendicular direction timber has low stiffness and is sensitive to moisture, showing large deformations. An acting compressive force can occur in any part of the structure, however a common place is the contact surface between members. The capacity is reliant of the factor \( k_{c,90} \) which take in to account how the force is applied.

- **[EN 1995-1-6.1.5]** Compression perpendicular to the grain.
  \[ \sigma_{c,0.d} \leq k_{c,90} \cdot f_{c,0.d} \quad (EC_5 6.3) \]

For a two-axis bending, both design conditions needs to be fulfilled. The modification factor \( k_m \) allows for re-distribution of forces and inhomogeneities of the material for cross section. For rectangular sections the factor is set to 0.7, which is used in the model.

- **[EN 1995-1-6.1.6]** Bending.
  \[ \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \cdot \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 6.11) \]
  \[ k_m \cdot \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 6.12) \]
Shear stresses emerge whenever a beam is subjected to bending and will act along the length of the beam.

\[ \tau_d \leq f_{v,d} \quad (EC_5 \ 6.13) \]

Design of cross-sections subjected to combined stresses. For bending in combination with tension the design check is only the sum of previous checks. Regarding bending in combination with axial compression several situations arises. For the case of a beam with low slenderness the fracture will not occur due to buckling but for its axial strength. The relation between axial strength and capacity is calculated as squared because of the rising capacity when the material yields, i.e. becomes plastic instead of elastic.

\[ \frac{\sigma_{t,0,d}}{f_{t,0,d}} + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \cdot \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 \ 6.17) \]

\[ \frac{\sigma_{t,0,d}}{f_{t,0,d}} + k_m \cdot \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 \ 6.18) \]

Stability of columns are verified around the axis with the largest slenderness ratio. A large slenderness value increase the effects of buckling. The factors \( k_{c,y} \) and \( k_{c,z} \) consider the straightness of the element (0.1 for Glulam) and the slenderness of the element. In turn the slenderness is dependent of the critical load which is calculated through the Euler buckling cases. The buckling length factor \( \beta \) is for each element type manually assigned in the input section of the model.

\[ \left( \frac{\sigma_{c,0,d}}{f_{c,0,d}} \right)^2 + \frac{\sigma_{m,y,d}}{f_{m,y,d}} + k_m \cdot \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 \ 6.19) \]

\[ \left( \frac{\sigma_{c,0,d}}{f_{c,0,d}} \right)^2 + k_m \cdot \frac{\sigma_{m,y,d}}{f_{m,y,d}} + \frac{\sigma_{m,z,d}}{f_{m,z,d}} \leq 1 \quad (EC_5 \ 6.20) \]

The lateral torsion stability applies for buckling in twist and lateral direction. The factor \( k_{crit} \) accounts for the reduced bending strength due to lateral buckling and is dependent on the relative slenderness. Equation \( EC_5 \ 6.33 \) is used when only a bending force acts around the strong axis and the compressive axial force is zero.
Beams subjected to either bending or combined bending and compression.

\[ \sigma_{m.d} \leq k_{\text{crit}} \cdot f_{m.d} \]  
\( (EC_5 \ 6.33) \)

\[ \left( \frac{\sigma_{m.d}}{k_{\text{crit}} \cdot f_{m.d}} \right)^2 + \frac{\sigma_{c.d}}{k_{c.z} \cdot f_{c.0.d}} \leq 1 \]  
\( (EC_5 \ 6.35) \)

Timber beams and, glulam in particular can be designed as tapered beams. The design of members with varying cross-section is checked with equation \( EC_5 \ 6.38 \) below. The factor \( k_{m.a} \) reduces the bending capacity based on the tapered angle.

\[ \sigma_{m.a.d} \leq k_{m.a} \cdot f_{m.d} \]  
\( (EC_5 \ 6.38) \)

### 3.4.2 Design in Serviceability Limit State

The serviceability limit state consider the appearance, comfort and functionality of the structure during normal use. The performance requirements can vary for each specific case, depending on demand from the client or allowed deformation regarding its function e.g. the sensitivity of movement for glass fixed to structure. It is also important to distinguish between reversible and irreversible deformations.

One main issue when designing glulam structures is its tendency to obtain large deflection due to the material properties for timber. One must take into account to the surrounding environment and risk of exposure to moisture, since it has a large impact on the deflection of wooden based structures.

\[ \text{Net deflection of beam.} \]

\[ w_{\text{net,fin}} = w_{\text{inst}} + w_{\text{creep}} - w_c \]  
\( (EC_5 \ 7.2) \)

The net deflection is the resulting deflection of the instance deflection, creep deflection and the initial curvature if precamber is applied.

The limiting values for instance deflection of beams, where \( L \) is in meter, should be set between the interval:

\[ \frac{L}{300} \text{ to } \frac{L}{500} \]

### 3.5 Connections and Supports

When it comes to connections there are two extreme cases; completely fixed connections and jointed connections. In reality either of these truly fulfill their statements.
A jointed connection often have a small stiffness. And it is very difficult to create a connection that is completely rigid, especially for timber connections. A lot of steel and nails or bolts are required to approach a connection which can be assumed rigid. Consequently, most connections are somewhere in between the two extremes.

There are several different types of timber connections and supports. Common examples of connections and supports can be found in Martinsson’s catalogue of standard solutions (Martinsson 2018). An example of a connection between a column and a beam in glulam are shown in Figure 3.3a. Steel plates are from two sides connecting the two elements with plenty of large nails. This system is easily scalable by adding bigger plates and more nails. An example of a supposed fixed support is also presented by Martinsson and seen in Figure 3.3b. The fixed support is made possible through casting a bolt basket in concrete and connecting the column with massive steel plates with plenty of nails. The largest capacity of a support of this type will withstand a bending moment of $68 \, kN\, m$ about the strong axis and $20.5 \, kN$ in shear. But such a connection would require a pillar dimension of $630 \times 215 \, mm$ and an immovable surrounding foundation. The full tables of the discussed connection and support types is found in Appendix A.2.1. Where corresponding capacities and required member dimensions are found.

**Figure 3.3:** Two examples of common type connection for Glulam (Martinsson 2018).
A simple type of connection between struts and chord in a truss is shown in Figure 3.4. Steel plates are recessed in the glulam members and fastened by dowels. If a stronger connection is needed more plates and dowels can be added. An example of such a connection is calculated in Appendix A.2.2. The calculation is used in the case study to determine the feasibility of the resulting structures.

![Figure 3.4: Connection between struts and chord by recessed steel plates. Left figure shows the dowels and the right figure the recessed steel plates.](image)

The closest way of achieving a rigid connection is by using a finger jointed connection. Either between beams or with a separate middle piece, which are custom sawn and glued together. Figure 3.5 shows a finger jointed connection with a middle section. This type of connection is often used for single tapered beams in frame corners or when jointing together long glulam elements.

![Figure 3.5: Finger jointed connection (Just, Piazza and Östman 2016, Ch.10.5).](image)

Within the aim of this thesis the design of connections which will work within the parametric model is not included. To be able to include parametric connections which would work in conjunction with the parametric model is however possible but time consuming and would likely be a separate project. However, the degrees of freedom (DOF) of the supports and connections in the model can be set by the user.
When a DOF is locked it is hundred percent rigid and the stiffness is infinite, such a connection or support for timber is more or less impossible to achieve. An estimation of the stiffness required for a connection to be assumed rigid can be calculated by considering the elasticity, second order of inertia and the length of the member. Neither of these calculation are included in the parametric model since it is not a part of the aim of the model. However, the connections discussed above is used to verify that the results obtained from the case study in Chapter 4 is producible in the sense of connection and supports.
Chapter 4

Case Study

4.1 Sergelgatan

The five skyscrapers called Hötorgsskraporna are located in the southern part of Norrmalm in Stockholm. The nineteen-story buildings were built along with the transformation of Sergelcity in the 1950’s. The whole block is now in the upcoming for a transformation to make the area attractive again after being closed for many years. In the middle of the busiest part of the city, the terraces with public access would offer relaxation and an easy accessible meeting place. The terrace, called Hötorgsterrassen is just a minor part of the entire project which includes plans for a glass roof which is described further in the next section 4.2.

The ambition for the project is to adapt the area to today’s demand for high quality shopping, restaurants and events. It should offer varying services with streets with a natural flow in-between them. The result should not be permanent but rather open for new flexible solutions as society today is in constant change (Vasakronan 2019).

![Figure 4.1: Vision of Sergelgatan.](image)
4.2 Glass Roof

During the latest renovation of Hötorgsskraporna in the 1990’s there were plans to enclosure the space between the buildings with glass roofs. Two glass roofs were built but for several reasons mostly economical, the glass roof between house 1 and 2 was never built. Consequently, this glass roof became included in the new extensive plans for the area since it was already permitted in the detailed development plan.

The purpose of the glass roof is to detach the inner square from the busy surrounding city life. It will house one of the two main entrances for the office towers as well as an entrance from the next-door subway station. In the project description it states the vision of the glass roof is to create a light and characteristic climate protection. The structure should represent contemporary style with a uniqueness as well as making minimal impact for appearance of the buildings. The glazing should follow the division of the facade and the main lines of the houses (Vasakronan 2019).

The glass roof would consist of frames with bracing in between, which will take care of the forces along the length of the structure. The tall column on the left, next to building number two, would have a height of 12 meters, the shorter column would be 6 meters and the span between them would be 15 meters. Further, the top of the shorter column is situated two meters higher than the tall one creating a tilt of the beam connecting them, see Figure 4.2. The stability in the span direction is the main problem with this structure, since there is no bracing able to withstand the forces acting in the span direction. The lateral movement will be increased due to the tilted roof.

4.2.1 Steel Structure

The original roof structure is currently in its design phase. It is made of a steel frame with a hollow triangular cross section, where the beam and columns are tapered with its thickest corner in the intersection between the beam and the column, which is seen in Figure 4.2. The connections are butt welded for rigid corners in both ends and the supports are fixed to the ground. Each frame would weigh about 3000 kg and have a beam cross section height which varies between 400 mm to 600 mm. The six frames are connected with bracing along the passage between the buildings.

Steel is commonly used in frameworks for open hall constructions with long spans, since it can be made slim and can be assembled on site by welding. It is considered to have a close to infinite lifespan since it has a closed life cycle. The manufactured steel can after its use be melted and reused in new formations (Stålbyggnadsinstitutet 2017).
4.2.2 Timber Structure

The client for the project was interested in evaluating if timber could be used as the main structural material of the atrium. This idea was however neglected before analysis could begin because of the advancement and the timeline of the project.

Within the scope and limitations of this thesis the established parametric model is applied to this case. The parametric model will evaluate various timber geometries to find the best suited solution. The geometries consist of typical glulam configurations, i.e rectangular beams, tapered beams and a truss which is an assembly of smaller glulam beams.

4.3 Load Effects

For this project the acting loads are snow load, wind load, self weight and the weight of the glass roof and its installations. The snow load and the load from the glass roof is applied along the free span of the structure. In this case, the wind load only affects the shorter column as seen in Figure 4.3 since it is an open construction without resisting walls between the tallest columns(column to the left).
It is assumed there is no dynamic loads affecting the roof structure and therefore the wind load is defined as an equivalent static load. The values of each load is presented below in Table 4.1. Where the values for snow and wind load are set to standard table values based on the geographical zone, exposure and form factors. The self weights for this case consists of load from the roof installation and the load by the beams in the frame. The roof load is given by the weight of the glass construction and is set as a design value. The self weight of the beams varies with the geometry dimensions and is therefore described by an equation where $\gamma$ is the specific weight and $V_{structure}$ is the volume of the whole structure.

<table>
<thead>
<tr>
<th>Snow (s)</th>
<th>Wind (w)</th>
<th>Roof w. installation ($g_{roof}$)</th>
<th>Self weight ($g_{beam}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 [kN/m²]</td>
<td>0.2</td>
<td>1.1</td>
<td>$\gamma \cdot V_{structure}$</td>
</tr>
</tbody>
</table>

Table 4.1: Design loads used in load combinations.

At the site for the glass roof there is risk of serious injury wherefore the safety class is set to 3 according to BFS 2015:6, EKS10, which result with $\gamma_d = 1.00$ for the load combinations. The final line distributed loads applied on the structure are calculated by the load combinations defined in EN 1990-0 and can bee seen below. The combination from ULS respectively SLS which gives the largest value will be used in the analysis of the structure.

\[
\begin{align*}
Q_{Ed,beam} &= \gamma_d \cdot \gamma_G \cdot (g_{roof} + g_{beam}) + \gamma_d \cdot \gamma_Q \cdot \psi_0 \cdot s \\
(Q_{Ed,beam} &= \gamma_d \cdot \xi \cdot \gamma_G \cdot (g_{roof} + g_{beam}) + \gamma_d \cdot \gamma_Q \cdot s
\end{align*}
\]
4.3. LOAD EFFECTS

Load combination for load acting along right column in ULS.

\[ Q_{Ed.column} = \gamma_d \cdot \gamma_Q \cdot \psi_0 \cdot w \]  \hspace{1cm} (EC_0 \; 6.10a)
\[ Q_{Ed.column} = \gamma_d \cdot \gamma_Q \cdot w \]  \hspace{1cm} (EC_0 \; 6.10b)

Characteristic load combination in SLS.

\[ Q_{Ed.SLS.beam} = (g_{roof} + g_{beam}) + s \]  \hspace{1cm} (EC_0 \; 6.14a)
\[ Q_{Ed.SLS.column} = w \]  \hspace{1cm} (EC_0 \; 6.14a)

A limitation of the Grasshopper model is that no control of the fire safety is performed. Why the capacity of fire loading is not included in calculation of load action effects. However, since the framework for this case is protected with sprinklers in the roof installations it is assumed the structure is fairly prevented against fire spread.

4.3.1 Global Stability and Dynamic Analysis

The design of a structure can not solely be determined by local stability controls pursuant to the Eurocode. The global structural behaviour also need to be investigated since it has a great impact for the overall stability of the structure. The buckling load factor and the natural frequency is therefore a part of the design process in the Grasshopper model. Appropriate constraining values of these parameters are affected by type of load affects and other prerequisites such as the application and location of the construction. In this section a short description of each parameter is presented together with the maximum allowed value for the studied case. The constraining values for these parameters are presented in Table 4.2.

The global buckling load factor indicates the safety of the structure which expresses the relation between the yielding buckling load and applied loads under perfect conditions. Buckling is expected when the factor is less or equal to 1. The failures are often rapid and with devastating consequences. In the case studied the safety is established by accounting for imperfections. Wherefore the minimum limit for the buckling load factor is set to 4. The buckling load factors for each mode are calculated by the model from the linear second order analysis. The analysis is based on small deformations and the geometrical stiffness matrix of the elements. The analysis is only used to calculate the global buckling factor why this will be done along with the first order analysis.

The natural frequency is the eigenfrequency of the structure, i.e. the frequency where the structure will oscillate if no damping is present. The structure will deform in accordance with the shape of the corresponding eigenmode. The eigenfrequency has no load dependency in linear elastic systems, but depend on other characteristics such as mass and stiffness of the structure. The structure is in this case not subjected to any source of vibration except dynamic effects from the wind. Since there is a very low risk of higher frequencies occurring from humans or machines, the lower limit for the natural frequency can be set to 2 Hz.
CHAPTER 4. CASE STUDY

<table>
<thead>
<tr>
<th>Inequality constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buckling load factor</td>
</tr>
<tr>
<td>Eigenfrequency</td>
</tr>
</tbody>
</table>

Table 4.2: Global stability and dynamic requirements for the case study.

Both the buckling load factor and the natural frequency can be calculated with integrated Karamba components. If these factors are not within the restraints the mass, utilization and deflection will be increased to go beyond the reasonable outcomes. Resulting structures from the optimization process will therefore never exceed the determined constraining values. This is further described in Section 4.5.

4.4 Model Geometry

The parametric model is developed for the case study with approach and methods described in Chapter 2 and 3. The model in Grasshopper is used to evaluate three different types of geometries for the glass roof structure described in Section 4.2.

The geometrical shape is limited to a three part framework with two columns at each side and a middle part that spans between, which consists of either a beam or a truss. The length and angle of these parts can be modified by the user through number sliders with varied values. There are other parameters which offers more specific control over the geometry, these are described for each geometry in Section 4.4.1 - 4.4.3.

The geometries which will be evaluated are listed below and described more profoundly in the following Sections 4.4.1-4.4.3.

- Frame with rectangular cross sections, with or without a strut between the left column and the beam.
- Frame with tapered cross sections, with or without a strut between the left column and the beam.
- Columns connected by a truss.

The same boundary conditions are applied for the three variations of geometry. Both columns are assumed to be fixed to the ground in all directions. Such kind of support is difficult to achieve in practice and is discussed in Section 3.5. Further, the beam and top chord in each case are assumed to be stabilized by bracing perpendicular to the plane(y-direction) and all connections between members are assumed to be pinned if nothing else is stated. This means there is no elastically restraining force in the connections and they will be free to move and rotate. Both these assumption regarding supports and connections is due to the fact no computations regarding connection and supports are performed in the model.
The buckling length of each member is user defined by Euler’s buckling factors, seen in Figure 4.4. These factors apply for steel but would be slightly higher for timber due to imperfections in the material. The buckling length of the struts is assumed to be their real length and the buckling length of the chord is set to the distance between the bracing. The real buckling length of the chords is probably a bit shorter meaning the bracing length will be on the safe side. The columns and beam will have buckling lengths based on the Euler buckling cases related to their boundary conditions.

![Buckled shape of column shown by dashed line](image)

<table>
<thead>
<tr>
<th>Theoretical K value</th>
<th>Recommended design value K</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.65</td>
</tr>
<tr>
<td>0.7</td>
<td>0.80</td>
</tr>
<tr>
<td>1.0</td>
<td>1.2</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2.0</td>
<td>2.10</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>

**Table 4.3:** Buckling length factors for each part of the structure.

The analysis performed by Karamba will be carried out in accordance to first order effects, second order effects are considered by the Eurocode instability controls. All members are checked for buckling, as well as for lateral buckling except for the beam and top chord which are braced. The chosen buckling factor from Euler’s buckling cases for each member used in the optimization stage is presented in Table 4.3. Furthermore, a linear second order analysis is required to calculate the global buckling factor why this will be done along side the first order analysis. This result in longer computation time and a slower optimization process but it enable the consideration of global stability as described in Section 4.3.1.

For all three structures the glulam class GL30c is used since it is a commonly used class with a high strength capacity at a low cost. The design value of material properties is modified by a factor $k_{mod}$ which takes the load duration and moisture content into account. Based on the location the characteristic for service class 2
is valid and the load duration class is set to medium-term, which answers for load duration of 1 week to 6 months. The modification factor $k_{mod}$ is therefore set to 0.8.

These parameters are specified for this particular case but can be modified by the user by choosing values from a given list. The resultant design may therefore differ depending on the choices made by the user regarding location, glulam class, etc.

4.4.1 Framework with Rectangular Cross Section

The simplest structure to support the glass roof is a frame consisting of three parts with rectangular cross sections. The frame with an additional strut can be seen in Figure 4.5. The two columns are fixed to the ground at different heights, the left column is 12 m high and the right is 6 m high but with its top situated 2 m above the left column top. A beam of 15 m span between the columns and are assumed to be pinned to the columns, due to the height difference between the columns the beam is tilted.

A slightly modified version of the frame is also evaluated where a supporting strut is placed between the taller column and the beam to reduce the deflection. The placement of the strut is generated by parameters to move along the column and it is assumed to be connected with pinned joints in both ends.

Figure 4.5: Frame with rectangular cross sections.
4.4.2 Framework with Single Tapered Beams

The geometry is the same as for the frame with rectangular cross section but with single tapered elements oriented as seen in Figure 4.6. However if a supporting strut is chosen, this will have a uniform rectangular cross section. The connection between the columns and the beam is assumed to be rigid since such a connection is feasible to construct for this type of beams. The rigid connection transmits the forces in the beam by shear to the column. The supports are defined the same as previous structure, in other words as fixed support.

Figure 4.6: Frame with tapered cross sections.

4.4.3 Framework with Truss

In the following geometry the beam is replaced with a truss where all members are of glulam. The strut members are jointed by pinned connections to the top and bottom chord, which is shown in Figure 4.7. The bracing will act where the struts are connected to the top chord. Further, all other boundary conditions are the same as for the frame with rectangular cross-section (see Section 4.4.1). In addition to the overall dimension of the structure the truss is free to alter in height, top and bottom chord curvature, number of struts and dimension of cross sections. To accommodate the studied case, the height of the truss is limited to $3.5 \text{ m}$.

There are two variation of the truss where the connection is defined in two different ways. Either both top and bottom chord is connected to the top of the columns,
see Figure 4.7a. This leaves the truss hanging underneath the top chord. Another
definition of the truss geometry is when the top chord is connected to the top of
the columns and the bottom chord is connected along the length of the columns
depending on the chosen truss height, resulting in the appearance shown in Figure
4.7b.

(a) Truss with pinned connections (Truss I).
(b) Truss with bottom chord connected
to columns (Truss II).

Figure 4.7: Draft of potential truss geometries.

The truss is generated by parameters which control the height and number of struts.
The struts will always be placed symmetrically and evenly distributed along the
length of the top chord with vertices and diagonals. There will always be a vertical
strut placed centered at the mid-span and by increasing the number of segments,
diagonals with connected vertices will follow.

4.5 Optimization

The geometries have been evaluated independently with the same objectives but with
different genomes depending on the type of geometry. The variation of the genes
are decided by setting limits to the number sliders which controls the corresponding
parameter. For instance the dimensions for each cross section can be varied by a
number slider and is therefore a useful option to have as a gene when managing the
optimization process.

The objectives presented in Table 4.4 have been used as a fitness function for the
evaluated geometries. The mass represent the the total mass of timber in the struc-
ture, the maximum displacement is the largest displacement found in any node and
the maximum utilization is the highest utilization in any node. If the utilization
parameter exceeds the ratio of 1 or if the buckling coefficient alternatively the nat-
ural frequency of the structure exceeds its constraining values (see Section 4.3.1) the
parameter will be increased to a value outside the solution range. In that way the
unreliable solutions are sorted out.
A low mass will result in a large displacement while a small displacement will result in a large mass. The objectives are contradicting each other, and the result will be a trade off between these. A solution also needs to have a utilization ratio below or equal to one. The constraints for the displacement and utilization enables sorting the result to only show the relevant solutions. Based on these objectives the optimal solutions are the ones closest to the displacement threshold with the lowest mass and which satisfies the utilization ratio criterion.

### 4.5.1 Optimization of Framework with Rectangular Cross Section

The frame with rectangular cross sections is only optimized in cross sectional dimensions, meaning there will not be any topological changes to the structure. In the case with a strut there will be geometrical change for the placement of the strut, which will be set as a gene for the optimization operation. In Figure 4.5 the selected genes are displayed.

| Gene                        | Rectangular frame | Rectangular frame w. strut | Variation  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Width of columns and beam</td>
<td>×</td>
<td>×</td>
<td>10-21.5 [cm]</td>
</tr>
<tr>
<td>Height of columns</td>
<td>×</td>
<td>×</td>
<td>25-170 [cm]</td>
</tr>
<tr>
<td>Height of beam</td>
<td>×</td>
<td>×</td>
<td>25-170 [cm]</td>
</tr>
<tr>
<td>Width of strut</td>
<td>×</td>
<td></td>
<td>5-21 [cm]</td>
</tr>
<tr>
<td>Height of strut</td>
<td></td>
<td>×</td>
<td>10-95 [cm]</td>
</tr>
<tr>
<td>Placement of strut on column</td>
<td></td>
<td>×</td>
<td>segments [-]</td>
</tr>
<tr>
<td>Placement of strut on beam</td>
<td></td>
<td>×</td>
<td>segments [-]</td>
</tr>
</tbody>
</table>

**Table 4.5:** Genomes used in optimization for the rectangular frame with and without strut.

The genes for placement of the strut is an integer that is dependent on the number of divisions, i.e. segmentation of the column. It chooses the closest point to the input value, starting from the bottom of column until the second last node along the column. The same applies for the beam where the strut connection alters from the
second node on to the second last node. The values of the other genes corresponds to the cross section dimensions in centimeters. These genes are free to vary within their variation range.

4.5.2 Optimization of Framework with Single Tapered Beams

The tapered frame is optimized with the same approach as for the rectangular frame described in Section 4.9. The genes used in the optimization is listed in Table 4.6 together with the variation limits of each gene.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Tapered frame</th>
<th>Tapered frame w.strut</th>
<th>Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width of columns and beam</td>
<td>×</td>
<td>×</td>
<td>10-21.5 [cm]</td>
</tr>
<tr>
<td>Tapering of left column</td>
<td>×</td>
<td>×</td>
<td>0-10 [-]</td>
</tr>
<tr>
<td>Tapering of right column</td>
<td>×</td>
<td>×</td>
<td>0-10 [-]</td>
</tr>
<tr>
<td>Tapering of beam</td>
<td>×</td>
<td>×</td>
<td>0-10 [-]</td>
</tr>
<tr>
<td>Min. height of left column</td>
<td>×</td>
<td>×</td>
<td>25-100 [cm]</td>
</tr>
<tr>
<td>Min. height of right column</td>
<td>×</td>
<td>×</td>
<td>25-100 [cm]</td>
</tr>
<tr>
<td>Min. height of beam</td>
<td>×</td>
<td>×</td>
<td>25-150 [cm]</td>
</tr>
<tr>
<td>Width of strut</td>
<td></td>
<td>×</td>
<td>5-21 [cm]</td>
</tr>
<tr>
<td>Height of strut</td>
<td></td>
<td>×</td>
<td>10-95 [cm]</td>
</tr>
<tr>
<td>Placement of strut on column</td>
<td></td>
<td>×</td>
<td>no. segments [-]</td>
</tr>
<tr>
<td>Placement of strut on beam</td>
<td></td>
<td>×</td>
<td>no. segments [-]</td>
</tr>
</tbody>
</table>

Table 4.6: Genomes used in optimization for tapered frame with strut and without strut.

The placement of the strut works as described in previous Section 4.9. The genes described as tapering of the column and beam is the slope of a linear function which describes the tapering. The minimum height $h$ of the cross section for each structural part is defined by a number slider. Both $h$ and the tapering $t$ are genes that is part of the optimization and are inputs to the equation $H = h + t \cdot L$ where $H$ is the maximum cross sectional height and $L$ the total length of the member.

4.5.3 Optimization of Framework with Truss

The selected genes for the truss manipulates the geometry to a greater extent compared to the other two geometries. As before the cross section is optimized in width and height, where the width is the same for all members to decrease the number
of genes. With too many genes Octopus have trouble to converge. Both Truss I and Truss II share the same genes. Beyond the genes for the dimensions of cross sections, the topology will be optimized. Genes which adjust the curvature height of the bottom chord, curvature of the bottom chord and the number of segments (which controls number of struts) are included.

The curvature of the bottom chord is defined by a curve which starts in a point located on the left column. Next the line goes through the curvature height point and ends in a point on the right column. The curve is illustrated in Figure 4.8. The height point is the gene that controls the curvature height and is defined as the distance from mid-span of the top chord to that point (middle arrow). The column points are defined in a similar procedure as the strut for the frame structure (see Section 4.5.1). The closest point of the segmentation of the column is given by a number where zero is at the top of the column and a higher number gives a point further down (left and right arrow). The combination of the two parameters creates the final shape of the bottom chord.

![Figure 4.8: Illustration of the bottom chord curvature.](image)

The division of the chords corresponds to the number of vertical struts as $n - 1$ where $n$ is the number of segments. The number of segments can only be even numbers due to the geometry definition. Because of the symmetric formation the number of segments $n - 2$ is also equal to the number of diagonals in the truss. If the curvature height is smaller than the curvature of the bottom chord the number of diagonals for Truss II will be equal to the number of segment ($n$).

<table>
<thead>
<tr>
<th>Gene</th>
<th>Frame with truss</th>
<th>Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curvature of bottom chord</td>
<td>x</td>
<td>0.0-5.0 [m]</td>
</tr>
<tr>
<td>Curvature height</td>
<td>x</td>
<td>0.0-3.5 [m]</td>
</tr>
<tr>
<td>Number of segments</td>
<td>x</td>
<td>2-8 [-]</td>
</tr>
<tr>
<td>Width of all members</td>
<td>x</td>
<td>12-21.5 [cm]</td>
</tr>
<tr>
<td>Height of columns</td>
<td>x</td>
<td>15-100 [cm]</td>
</tr>
<tr>
<td>Height of struts</td>
<td>x</td>
<td>10-40 [cm]</td>
</tr>
<tr>
<td>Height of top chord</td>
<td>x</td>
<td>15-60 [cm]</td>
</tr>
<tr>
<td>Height of bottom chord</td>
<td>x</td>
<td>15-60 [cm]</td>
</tr>
</tbody>
</table>

Table 4.7: Genomes used in optimization for frames with truss design.
4.6 Resulting Models

In the following sections the resulting geometries after optimization are presented. All configuration of the resulting framework is within the limit for the global stability requirements, presented in Table 4.2. The bending moment distribution and the deformation for all the resulting geometries are shown in Appendix B.2.

4.6.1 Framework with Rectangular Cross Section

The optimized framework of rectangular cross sections with and without an added strut can be seen in Figure 4.9. The corresponding genes for both cases is presented in Table 4.8 and the mass and maximum displacement is given in Table 4.9.

![Frame with rectangular cross-section.](image1)
![Frame with rectangular cross-section and a strut.](image2)

**Figure 4.9:** Optimized cross sections and geometry for rectangular frame with and without strut.

The number of optimization genes differ for the two cases since the strut is dependent on several input variables. Despite their differences the cross sectional dimensions converge to similar results.

In both cases the columns and the beam are divided into 18 segments to create multiple calculation points and increase the accuracy of the computation. The points are located at the end of every segment which creates elements with sharing nodes. These node are also used to determine the placement of the strut where an equation prevents it from selecting the nodes at the ends of the column and beam. Otherwise undefined geometries would follow. The n:th node corresponds to the placement of the strut both along the column ant the beam. Hence the strut is generated by four genes, two for its cross sectional dimension and two to control the position.
This however result in a strut converging towards nodes closest to the connection between the column and the beam. The reason for this is the the additional moment in the column caused by the strut. A placement closer to the mid height of the column contributes to greater bending moment while a placement further along the beam would mean an increased weight and vulnerability against instabilities. Furthermore the angle of the strut also affects the additional moment. A smaller angle to the grain will minimize the horizontal force component, also lowering the additional moment. A small angle will increase the vertical force component which will be inflicted as additional compression in the beam.

The column is not laterally braced why it is more sensible to buckling than the frame without strut, and need a greater cross sectional height to cope with an increased bending moment. Since the resistance against bending moment is dependent of the elastic section modulus \( W_y \) and the resistance of deflection is depended on the second order of inertia \( I_y \). Both of these parameters depends mostly of the height of the structure, why the beam in both structures have massive heights. Consequently, the strut will minimize the moment it causes by converging to the corner between the column and beam.

<table>
<thead>
<tr>
<th>Gene</th>
<th>Rectangular frame</th>
<th>Rectangular frame with strut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width of columns and beam</td>
<td>21.1 [cm]</td>
<td>20.8 [cm]</td>
</tr>
<tr>
<td>Height of columns</td>
<td>53 [cm]</td>
<td>57 [cm]</td>
</tr>
<tr>
<td>Height of beam</td>
<td>133 [cm]</td>
<td>138 [cm]</td>
</tr>
<tr>
<td>Width of strut</td>
<td>-</td>
<td>20 [cm]</td>
</tr>
<tr>
<td>Height of strut</td>
<td>-</td>
<td>15 [cm]</td>
</tr>
<tr>
<td>Placement of strut on left column</td>
<td>-</td>
<td>16 [-]</td>
</tr>
<tr>
<td>Placement of strut on beam</td>
<td>-</td>
<td>1 [-]</td>
</tr>
</tbody>
</table>

**Table 4.8:** Optimized genomes for rectangular frame with and without a strut.

As explained above the geometry with a strut result in larger dimensions and consequently weighs about 280 \( kg \) more than the frame without a strut. The last mentioned is therefore a better solution. Besides, the frame with the strut have an unrealistic placement of the strut. If there within the model would be an option where the strut would disappear it probably would.

<table>
<thead>
<tr>
<th>Design</th>
<th>Mass [kg]</th>
<th>Displacement [mm]</th>
<th>Max Utz. [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular frame</td>
<td>2394.9</td>
<td>29.4</td>
<td>0.85</td>
</tr>
<tr>
<td>Rectangular frame with strut</td>
<td>2497.5</td>
<td>28.4</td>
<td>0.97</td>
</tr>
</tbody>
</table>

**Table 4.9:** Resulting objectives.
4.6.2 Framework with Single Tapered Beams

For the frame with tapered cross sections two resulting model is composed, one with a strut and one without. They can be seen in Figure 4.10 with their corresponding genes in Table 4.10 and their resulting objectives in Table 4.11.

![Diagram of frame with tapered cross-section and strut.](image)

**Figure 4.10:** Optimized cross sections and geometry for tapered frame with and without a strut.

Both structures have very similar cross sections. The width of the beam and columns are equal and the tapering of both columns are the same. The rigid connection between the left column and the beam will increase the bending moment in the column, resulting in an increased height of the left column.

As in the previous geometry with rectangular cross section, the strut will cause an additional bending moment in the left column. To accommodate for this the left column will have a slightly bigger minimum height at the bottom compared to the structure without the strut. Further, the minimum height of the beam is the same for both structures while the tapering is larger on the one without strut. This is because the forces can be taken by the strut instead of the beam.

The placement of the strut for this case differs compared to previous geometry with rectangular beams. As a result of the increased cross section height for the left column due to the rigid connection the strut is free to move. The bending moment caused by the strut is no longer the main contributing factor. Therefore several options are explored and result in a position around the middle of the left column and beam.
4.6. RESULTING MODELS

<table>
<thead>
<tr>
<th>Gene</th>
<th>Tapered frame</th>
<th>Tapered frame with strut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width of columns and beam</td>
<td>20 [cm]</td>
<td>20 [cm]</td>
</tr>
<tr>
<td>Min. height of left column</td>
<td>67 [cm]</td>
<td>71 [cm]</td>
</tr>
<tr>
<td>Tapering of left column</td>
<td>1.43 [-]</td>
<td>1.43 [-]</td>
</tr>
<tr>
<td>Min. height of right column</td>
<td>34 [cm]</td>
<td>34 [cm]</td>
</tr>
<tr>
<td>Tapering of right column</td>
<td>4.02 [-]</td>
<td>4.02 [-]</td>
</tr>
<tr>
<td>Min. height of beam</td>
<td>85 [cm]</td>
<td>85 [cm]</td>
</tr>
<tr>
<td>Tapering of beam</td>
<td>6.55 [-]</td>
<td>6.12 [-]</td>
</tr>
<tr>
<td>Width of strut</td>
<td>-</td>
<td>18 [cm]</td>
</tr>
<tr>
<td>Height of strut</td>
<td>-</td>
<td>31 [cm]</td>
</tr>
<tr>
<td>Placement of strut on left column</td>
<td>-</td>
<td>9 [-]</td>
</tr>
<tr>
<td>Placement of strut on beam</td>
<td>-</td>
<td>7 [-]</td>
</tr>
</tbody>
</table>

Table 4.10: Optimized genomes for tapered frame with and without a strut.

The framework with an additional strut is about 190 kg heavier due to extra mass from the strut and result in a larger deflection. The single tapered frame without a strut is therefore preferable in this sense.

<table>
<thead>
<tr>
<th>Design</th>
<th>Mass [kg]</th>
<th>Displacement [mm]</th>
<th>Max Utz. [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single tapered frame</td>
<td>2463.6</td>
<td>27.1</td>
<td>0.98</td>
</tr>
<tr>
<td>Single tapered with strut</td>
<td>2653.5</td>
<td>28.4</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 4.11: Resulting objectives.

4.6.3 Framework with Truss

The two different truss geometries of the structure is presented below. The resulting geometry is presented in the Figures 4.11 and 4.12 together with the resulting objectives in Table 4.13 and the genes corresponding the result is found in Table 4.12.

Truss Type I

The columns of the structure resembles the dimensions of the frame with rectangular cross section without a strut. This is because the pinned connections at the top of the column does not transfer any bending moment to the columns. The largest moment in the columns occurs at the support and is caused by the wind load.
The largest deflection appears slightly to the left of mid-span of the truss and the greatest bending moment will arise where the struts are connected to the top chord because of the bracing that acts at these points. To decrease the deflection and withstand the bending forces the truss will increase its height to achieve a larger second order of inertia and section modulus.

A large number of struts cause a more even distribution of the bending moment in the top chord and reduces the size of the maximums. The compression and tension forces are distributed among the struts. Although the bending moment is reduced, the weight of the structure increases with more number of struts. That is why the resulting model converge to three vertical struts. This is a compromise of relieving perpendicular compression on the top chord and reducing the weight of the structure.

**Truss Type II**

The difference between Truss I and Truss II is by what method the bottom chord is connected to the column. The top and bottom chord will create a equivalent force couple that causes an additional bending moment in the columns, resulting in an increased cross sectional height for these parts compared to Truss I.
While the optimization of Truss I converge to a state with few struts to decrease weight, Truss II converge towards a state with more struts to relieve the chords from its bending and compression forces. This result in slimmer chords with smaller heights. The top of the columns gains some extra stiffness due to the extra vertical struts connecting the top and bottom chords at the two ends of the truss.

**Results from Truss Types**

The results from the optimization of the two truss types are not significantly different regarding the dimensional sizes. The number of strut is the most influential factor for the mass and deflection of the system due to the impact on the bending stiffness of the truss.
<table>
<thead>
<tr>
<th>Gene</th>
<th>Truss I</th>
<th>Truss II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curvature of bottom chord</td>
<td>0.8 [-]</td>
<td>2.1 [-]</td>
</tr>
<tr>
<td>Curvature height</td>
<td>3.5 [-]</td>
<td>3.5 [-]</td>
</tr>
<tr>
<td>Number of segments</td>
<td>4 [-]</td>
<td>6 [-]</td>
</tr>
<tr>
<td>Width of all members</td>
<td>20.6 [cm]</td>
<td>19.9 [cm]</td>
</tr>
<tr>
<td>Height of columns</td>
<td>55 [cm]</td>
<td>60 [cm]</td>
</tr>
<tr>
<td>Height of struts</td>
<td>12 [cm]</td>
<td>15 [cm]</td>
</tr>
<tr>
<td>Height of top chord</td>
<td>24 [cm]</td>
<td>19 [cm]</td>
</tr>
<tr>
<td>Height of bottom chord</td>
<td>16 [cm]</td>
<td>15 [cm]</td>
</tr>
</tbody>
</table>

Table 4.12: Optimized genomes used for truss design I and II.

<table>
<thead>
<tr>
<th>Design</th>
<th>Mass [kg]</th>
<th>Displacement [mm]</th>
<th>Max Utz. [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truss frame I</td>
<td>1466.0</td>
<td>14.9</td>
<td>0.98</td>
</tr>
<tr>
<td>Truss frame II</td>
<td>1721.3</td>
<td>12.8</td>
<td>0.98</td>
</tr>
</tbody>
</table>

Table 4.13: Resulting objectives for the truss structures.

### 4.6.4 Summary of Result

The results for the six resulting models is plotted in Figure 4.13. It can be seen that all structures are below the deflection limit of 30 mm meaning all structures except the rectangular frame with strut (ill-formed for construction) could be further evaluated. That is if the mass is of no interest. The two truss configuration differentiate themselves from the others by having both a low weight and small deformations. While the other geometries are clustered around the same mass and displacement. Thus, the two options with a truss is preferable if a low mass is favourably.

The structure with the lowest mass is Truss I with a mass of 1460 kg and a maximum displacement of 14.9 mm. However, the geometry may be of higher importance than these parameters. The optimal structure becomes a subjective matter affected by the situational circumstances.
By further investigation of Truss I it appears the maximum utilization varies for each part in the structure. The variation is displayed in Figure 4.14. Hence, it could be misleading to only look at the maximum utilization of the structure.

The left column (C1) have a significant greater utilization ratio than the right column (C2). This is because they have the same dimensions but C1 is twice as long as C2, which increase the risk of buckling. By further studying the chart it shows the struts have the lowest overall ratio. The diagonal struts are barley making use of its strength in tension while the vertical struts (S1-S3) are somewhat sensible for buckling. Also, the top and bottom chord are well utilized, which can be traced to buckling in the top chord (T1) and to compression perpendicular to the grain in the bottom chord (T2).
Since connections or supports are not considered in the Grasshopper model a simplified verification against the standard connection discussed in Section 3.5 is presented below.

In Table 4.14 the forces acting on two connections are presented. The pinned connection between the column and the top chord must withstand 155 kN in compression and 0.3 kN in shear force. Given these forces and the dimensions of the column and top chord the pinned connection seen in Figure 3.3a is suitable. More specific the version named PB05, specified in Appendix A.2.1 is an appropriate connection for this case. It withstands 200 kN in vertical force and about 10 kN in the horizontal direction, for more details see the Appendix.

The support that is supposed to be fixed to the ground have an extensive normal force acting on the support but have a smaller bending moment and shear force. In combination with the dimensions of the column any version of the fixed support from Martinsson in Appendix A.2.1 is adequate. This is of course given that the concrete plate which the support is cast into can be assumed to be immovable. The support with the lowest resistance have a bending resistance of 36 kNm and a horizontal capacity of 10.5 kN and can manage the affecting forces.
Table 4.14: Section forces affecting the connection or support.

<table>
<thead>
<tr>
<th></th>
<th>Left support</th>
<th>Left column top</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal force $N$ [kN]</td>
<td>155.4</td>
<td>155.4</td>
</tr>
<tr>
<td>Shear force $V_z$ [kN]</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Bending moment $M_y$ [kNm]</td>
<td>1.4</td>
<td>0</td>
</tr>
</tbody>
</table>

The largest section forces in Truss I occurs at center of the truss where the three struts and the bottom chord meets. The forces acting in this intersection is presented in Table 4.15. The truss connection with recessed steel plates and dowels discussed in 3.5 and calculated in Appendix A.2.2 do manage these forces with the resulting member dimensions. It is assumed variations of this connection also will work at the other intersections in the truss.

Table 4.15: Section forces affecting the truss connection. Negative values mean compression.

<table>
<thead>
<tr>
<th></th>
<th>Truss detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal force in bottom chord $N_c$ [kN]</td>
<td>163</td>
</tr>
<tr>
<td>Normal force in diagonals $N_d$ [kN]</td>
<td>12</td>
</tr>
<tr>
<td>Normal force in vertical $N_v$ [kN]</td>
<td>-69</td>
</tr>
</tbody>
</table>

4.7 Model Verification with Robot

The use of a parametric design in conjunction with structural analysis and customized design components depends on if the model in fact provide reliable results. Consequently, the resulting model from Grasshopper is verified against a FEM analysis from Robot, a conventional structural design program. The optimized geometry of Truss I is exported to Robot where it is given the same material properties, cross sections, boundary conditions, element division and load combination. The bending moment distribution from both analysis can be seen in Figure 4.15. The moment diagrams have the same shapes and the values are close, indicating that the boundary conditions and loads are defined equally. The greatest difference is seen at the left support where the bending moment in Robot is about 1.5 kN larger than in Karamba. At the other calculation points the difference are of insignificant difference. A probable reason might be how to the two programs solves the equation matrix.
In Table 4.16 the resulting section forces and utilization ratios from the performed Eurocode controls are presented for three parts. The tall column, top chord and strut from Karamba are compared with the same elements from the Robot calculation. The section forces and the deflections show similar results. The major displacement difference is found at the top chord where the difference is 0.77 mm.

All utilization ratios from the Eurocode checks is not presented in the table. This is because the different methodology of calculation between the scripted design codes in Grasshopper and Robot. In Robot only the controls which will be limiting is calculated while the calculations in Grasshopper performs the controls the user selects. Which means the two calculations will not always perform the same controls.

For the column two common controls where performed. Shear and buckling (EC5 6.13, EC5 6.24). The models provide similar results for both these controls, except for a decimal difference from the buckling control. For the top chord the following controls is compared; shear (EC5 6.13), combined bending and compression (EC5 6.19) and buckling (EC5 6.23). The outcome does not differ substantially. However, the differences in the result from the combined bending and compression, and buckling can partly be explained by the $k_h$ factor which increases the bending resistance if a criterion in EN 1995-5 is fulfilled. This has not been used in the Grasshopper model. Finally, the strut is controlled for shear and buckling in both models. Where the first equation gives zero utilization due to the low shear force. While the buckling differs only by a decimal.

**Figure 4.15:** Bending moment distribution. *Note Karamba uses opposite signs convention compared to Robot.*
### Table 4.16: Comparison of results from Karamba and Robot. The equations refer to the design equations in 3.4. *Note $k_h = 1.10$ has been used in Robot.*

<table>
<thead>
<tr>
<th></th>
<th>Column (C1) All elements</th>
<th>Top chord Element 42</th>
<th>Strut Element 37</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Karamba</td>
<td>Robot</td>
<td>Karamba</td>
</tr>
<tr>
<td>$N$ [kN]</td>
<td>155.37</td>
<td>153.98</td>
<td>176.85</td>
</tr>
<tr>
<td>$V_z$ [kN]</td>
<td>0.12</td>
<td>0.10</td>
<td>38.74</td>
</tr>
<tr>
<td>$M_y$ [kNm]</td>
<td>1.39</td>
<td>1.22</td>
<td>27.44</td>
</tr>
<tr>
<td>$w_{inst}$ [mm]</td>
<td>2.17</td>
<td>2.00</td>
<td>14.02</td>
</tr>
</tbody>
</table>

Utilization ratios:

<table>
<thead>
<tr>
<th></th>
<th>$EC_5$ 6.13</th>
<th>0.00</th>
<th>0.00</th>
<th>0.78</th>
<th>0.78</th>
<th>0.00</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$EC_5$ 6.19</td>
<td>0.01</td>
<td>-</td>
<td>0.77</td>
<td>0.73*</td>
<td>0.03</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$EC_5$ 6.23</td>
<td>-</td>
<td>-</td>
<td>0.98</td>
<td>0.92*</td>
<td>0.46</td>
<td>0.45*</td>
</tr>
<tr>
<td></td>
<td>$EC_5$ 6.24</td>
<td>0.85</td>
<td>0.84</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$EC_5$ 6.33</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$EC_5$ 6.35</td>
<td>0.10</td>
<td>-</td>
<td>0.38</td>
<td>-</td>
<td>0.18</td>
<td>-</td>
</tr>
</tbody>
</table>

In the optimization process described in Section 4.5 there are assigned requirements for both the natural frequencies and global buckling factor. It is therefore of interest to examine if these factors are similar when calculated by Robot. In Table 4.17 the natural frequencies and the global buckling factor of the first mode is presented. Both calculations result in values which pass the minimum constraint values set for the studied case. The global buckling factor obtained from Robot is greater than the one from Karamba. The buckling analysis are in both models calculated linearly with second order effects and are based on the boundary conditions of the systems. Within this thesis it is not possible to further analyze the cause of this difference.

### Table 4.17: Natural frequency and global buckling factor of the first mode.

<table>
<thead>
<tr>
<th></th>
<th>Karamba</th>
<th>Robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural frequency [Hz]</td>
<td>2.42</td>
<td>2.40</td>
</tr>
<tr>
<td>Global buckling factor [-]</td>
<td>4.03</td>
<td>5.59</td>
</tr>
</tbody>
</table>
Chapter 5

Conclusion

5.1 Conclusive Summary

The main aim with this thesis and the Grasshopper model have been to create a model to perform FE-analysis, structural design and optimization, based on parametric and generative thinking. Further, any geometry specified in a predetermined approach should be possible to analyze. A general model is however difficult and complex to establish, even each computational part on its own are complicated to define.

As for now the Grasshopper model is not compatible for any 2D structural problem with timber. However, it do work in conjunction with the case study it is applied to, where different types of frame geometries are defined and evaluated. These geometries are defined with a parametric approach where all parameters are in relation. Hence, their dimensions can be modified freely within the 2D space and the computation will follow. The model is general for framework arrangements but would need further development to work on other types of structures.

To ensure the outcome of the model is accurate the results were compared with an analysis in Robot. The comparison showed the results to a great extent correspond to each other, which increase the reliability of the Grasshopper model. The differences that do exist can mostly be traced to different assumption about which controls to perform and the $k_h$ value that is discussed in Section 4.7.

The scripted design codes is reliant to how the geometry is composed. The geometry need to be structured in a predefined way based on indexation of the element groups. If this is not accomplished the python scripts will not be able to match section forces to the associated cross-section and element properties. The Eurocode manage a number of different boundary conditions which contributes to a complex calculation flow, especially when written in code. In a transformable structure these conditions may change, why the decision related to boundary conditions is up to the user to define.
The issues which emerge when performing optimization of any kind are discussed in previous chapters. An optimal structure is not an objective outcome but a result of one or several decisions concerning which parameters are important and which are not. The configuration of the optimization parameters must therefore be user defined. However, many of the possible parameters is prepared in the model for the user to choose from.

Furthermore, there are a lot of assumptions involved in the model. One vital assumption is the lack of consideration for the connections and supports. These calculations are not included in the model since it would become too complicated. A single connection consists of many variables and a large number of different connectors would be needed to find an optimal configuration. They would need to be applicable for each outcome of the changeable structure. A further analysis would therefore be necessary before final decision can be made about possible producibility.

Yet within the scope of the model, the model is applicable in early stages of projects why the limitations can be acceptable. Likewise the model can be considered somewhat general within the aforementioned constraints. In conclusion, the Grasshopper model confirms it is possible to use a generative approach for timber structural design within the same interface and produce reliable results.
Chapter 6

Discussion

6.1 Generative Design

The intention of the produced model was to be used as a tool for structural design and optimization of timber structures. The Grasshopper model consider design in the ultimate limit states, the displacement from serviceability limit state and a limited analysis of the global stability. They do not consider all structural factors which may affect a structure why the model is not quite general. It will have limitations as earlier described. Nevertheless it will be compatible for simple frameworks with multiple solutions.

It has been stated in the report that the Grasshopper model is inadequate regarding connections and supports. The calculation of these has to be done by hand after final geometry is determined. This may however affect the results since the resulting optimized solution perhaps is not appropriate to construct due to lack of connectivity possibilities. That is why it would be interesting to add these calculation in the Grasshopper model. There are several methods to implement computation of connections. A generative approach would be to calculate a connection for every iteration of the geometry. The connections would be based of the resulting section forces in the nodes between members, i.e. all connections would be parametric and generative. This would require extensive coding but more important the computation time would probably increase drastically, which limits the advantage of a generative model. A simpler solution would be to use predetermined connections. The user could choose among the variation of connection and decide at which node they should be located. The bearing capacity of the chosen connection would then be the limit of the allowed section force in each node.

The outcome of a structural optimization is dependent on how much the geometry is allowed to transform and which objectives that are chosen. The variation of the structure may often be limited by the design space while the objective is a more subjective problem. This raises the question of why and by whom objectives are chosen. Mass, cost, environmental impact, displacement and producibility is only a few examples of possible objectives to optimize. These may vary between the
client and contractor. Why it should be clear what factor should be of the greatest importance in the optimization process, since this most likely will affect the result.

It is difficult to say anything about time spent on developing the model compared to what the result would have become with conventional design by hand. Although if Karamba 3D would have supported anisotropic material, such as timber, it would have saved time in coding customized solutions. Such an implementation would open for a more convenient workflow for generative structural design of timber structures.

6.2 Comparison with Steel Structure

To compare the original steel frame with the truss type I geometry in a environmental perspective is not suitable. The two structures have not been developed in the same way and the steel structure have not been optimized in a generative manner. Further, the timber structure can not be assumed to be ready for manufacturing in its current state. Consequently it would not be appropriate to compare them even on a very basic scale.

An environmental comparison between two structures depends on many variables and aspects which need to be considered. The Environmental Product Declaration (EPD) presented in Section 3.3 is a simplified estimate of the environmental impact of the production of a material. It shows that glulam has a lower carbon dioxide emission than steel. However, the two EPD only states the data and is only comparable if the conditions of two structures are the same. Timber can be seen as a renewable resource with ability to plant new trees, while steel is accessed through iron from ground which has its limitation. However, steel has longer a lifespan than timber and can be melted and reused after demolition of a construction.

Other qualities which can be mention is the producibility. It is easy to make a corner of steel elements elastically restrained with butt welds. Welding is not possible for timber why it is hard to create connections with the same rigidity as butt welds for steel. A steel structure can be delivered to the construction site in pieces and later be jointed together. The glulam beam is preferably delivered in its final form, with pre-drilled boreholes and prepared connection. This is often necessary to prevent the glulam of being exposed to the natural elements. In the end it is up to the contractor to choose what material is suitable in their project.
6.3 Further Work

As previously mentioned the model in grasshopper is limited to be used as an estimation of the structural design of a simple framework in an early stage of a project. Wherefore it would be interesting to continue to develop this model to be used throughout the duration of the project. To achieve this, some details need to be studied. Suggestion of further research on this matter is therefore described below.

- An extensive dynamic analysis to account for other load events.
- Structural fire design according to Eurocode. An analysis of the fire resistance for the entire system and its weak parts.
- Implementation of timber connections in the model and include a generative computation of connections in the optimization process.
- Examine and make adjustments to compose a model which is reliable for 3D structures.
- An extensive comparison with an optimized timber structure and an optimized steel structure. Composed with the same preconditions and by using the same objectives in the optimization. A equitable comparison would be useful in future decision making of using either steel or timber in construction projects.
- Implement steel as a selection for beam parts in the model. Karamba already have material properties for steel classes included in components. This would increase the flexibility of the types of structures that can be analyzed.
Bibliography


Appendix A

Calculations
A.1 Python Scripts

6.1 Design of cross-sections subjected to stress in one principal direction

6.1.2 Tension parallel to grain
6.1.4 Compression parallel to grain

"""Design of cross-sections subjected to stress in one principal direction - Tension/Compression parallel to the grain
Inputs:
N: Normal force [kN]
A_crossec: Area cross-section [cm²]
strength_d: Design value of strength properties [MPa]
0 - f_md
1 - f_t0d
2 - f_t90d
3 - f_c0d
4 - f_c90d
5 - f_vd
6 - f_rd
Element: Element ID:s
Output:
Result: Result for each element from design according Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
σ_ct0d: Compressive stress(-) / Tensile stress(+) [MPa]
utz: Utilization ratio of each element"""

_author_ = "bohma", "nicol"
__version__ = "2019.04.10"
import rhinoscriptsyntax as rs

"""6.1 Design of cross-sections subjected to stress in one principal direction"""

"""6.1.2 Tension parallel to grain"""
"""6.1.4 Compression parallel to grain"""

f_c0d = strength_d[3]  #Design Compression capacity along the grain [MPa]
f_t0d = strength_d[1]  #Design Tension capacity along the grain [MPa]

N = [i * 10**-3 for i in N]  #Normal force [MN] instead of [kN]
A = [i * 10**-4 for i in A_crossec]  #Convert Area cross-section [m²] instead of [cm²]

Result = []  #Empty list output
Elements_NotOk = []  #Empty list output
σ_ct0d = []  #Empty list output
utz = []  #Empty list output

"""Calculates stresses with regard to quantity of normal forces fN"""
for i in N:
    index = fN.index(i)
sigma_ct0d = i/A[index]  #Axial -> Compression / Tension

"""Utilization ratio eq.6.1-6.2"""
usage = abs(sigma_ct0d)/f_c0d

# Tension
if sigma_ct0d > 0:
    if usage <= 1:
**6.1 Design of cross-sections subjected to stress in one principal direction**

**6.1.5 Compression perpendicular to grain**

"""Design of cross-sections subjected to stress in one principal direction - 6.1.5 Compression perpendicular to grain

**Inputs:**
- **N**: Normal force [kN]
- **A_crossec**: Area cross-section [cm²]
- **strength_d**: Design value of strength properties [MPa]
  - 0 - f_md
  - 1 - f_t0d
  - 2 - f_t90d
  - 3 - f_c0d
  - 4 - f_c90d
  - 5 - f_vd
  - 6 - f_rd
- **Element**: Element IDs

**Output:**
- **Result**: Result for each element from designing with Eurocode
- **Elements_NotOk**: Shows elements that are NOT OK according to design criterion
- **σ_c90d**: Perpendicular Compressive stress(-) [MPa]
  - If no perpendicular force then stress is 0 MPa
- **utz**: Utilization ratio of each element"

```python
f_c0d = strength_d[3]  # Design Compression capacity along the grain [MPa]
f_t0d = strength_d[1]  # Design Tension capacity along the grain [MPa]
```
f_c90d = strength_d[4]              #Design Compression capacity perpendicular to the
                                      #grain [MPa]

fN = [i * 10**-3 for i in N]        #Normal force [MN] instead of [kN]
A = [i * 10**-4 for i in A_crossec]  #Convert Area cross-section [m2] instead of [cm2]
b_list = [i * 10 for i in Width]
h_list = [i * 10 for i in Height]

Result = []                        #Empty list output
Elements_NotOk = []               #Empty list output
c_c90d = []                        #Empty list output
utz = []                           #Empty list output

""" Calculates stresses with regard to quantity of normal forces fN""
for i in fN:
    index = fN.index(i)
sigma_ct0d = i/A[index]            #Axial -> Compression / Tension

    # Tension along the to grain
    if sigma_ct0d > 0:
        usage = 0
        sigma_c90d = 0
        Ans = Element[index] + ": No compression force acting perpendicular to grain 

    # Compression
    else:
        if Element[index] in Compression90:
            #Compression perpendicular to grain
            """Utilization perpendicular to grain
            L = h_list[index]
b=[]
h=[]

            for j in fN:
                iteration = fN.index(j)
                #print index
                #print iteration
                if str(Element[iteration]) in str(Adjacent90):
                    b.extend([int(b_list[iteration])])
                    h.extend([int(h_list[iteration])])
                    b_min = sorted(b)[0]
                    h_min = sorted(h)[0]
                    #print b_min
                    #print h_min

                    if h_min <= 2.5*b_min:
                        #Effective length for internal supports that are not discrete
                        if SupportCondition == 1:                           #effective length
                            L_ef = L + h_min/3
                        elif SupportCondition == 2:
                            L_ef = L + h_min*2/3
                        elif SupportCondition == 3:
                            L_ef = 0.5*(L+L_s+h_min*2/3)  #For discrete
                            support. Calculation not supported in this code.
                        else:
                            see EN1995-1-1-6.1.5 (5)
L_ef = L  # If other option is chosen, L_ef is = L
k_c90 = (2.38 - L/250) * (L_ef/L)**0.5

else:
    k_c90 = (2.38 - L/250) * (1 + h_min/(6*L))

if Element[index] in MemberRot:
    # Calculates the horizontal/vertical component of the compressive force
    # temporary list to determine the highest usage
    # Note that this part of the code is highly "hard coded". If being used it should
    # be controlled that the calculations are accurate.
    print index
    for k in Angle:
        sigma_c90d = sigma_ct0d*sin(m.radians(k))
        usage.extend([abs(sigma_c90d)/(k_c90*f_c90d)])
        usage = sorted(usage)[0]

    # Utilization ratio for compression perpendicular to grain
    EN1995-1-1- (6.1.5) eq.6.3
    else:
        usage = abs(sigma_ct0d)/(k_c90*f_c90d)
        if usage <= 1:
            sigma_c90d = sigma_ct0d
            Ans = Element[index] + " : Compression OK  \sigma_{c0d} = "
        else:
            # Compression perpendicular to grain output
            Ans = Element[index] + " : Compression NOT OK  \sigma_{c0d} = "
            NotOk = "Element " + str(index) + ", Part " + Ans + str(sigma_c90d)[:8]
            Elements_NotOk.extend([NotOk])

    else:
        # Compression along the grain
        usage = 0
        sigma_c90d = 0
        Ans = Element[index] + " : No compression force acting perpendicular to grain"

    # Outputs
    display = "Element " + str(index) + " , Part " + Ans + str(sigma_c90d)[:8] + " MPa"
    Result.extend([display])
    \sigma_{c0d}.extend([sigma_c90d])
    utz.extend([usage])

6.1 Design of cross-sections subjected to stress in one principal direction
6.1.6 Bending

"""Design of cross-sections subjected to stress in one principal direction - Bending stresses about the principal axes
Inputs:
My: Maximum moment about y [kN/m]
Mz: Maximum moment about z [kN/m]
Wy: Resistance Moment about strong axis \([\text{cm}^3]\)
Wz: Resistance Moment about weak axis \([\text{cm}^3]\)
strength_d: Design value of strength properties \([\text{MPa}]\)

\[
\begin{align*}
0 & = f_{\text{md}} \\
1 & = f_{\text{t90d}} \\
2 & = f_{\text{c90d}} \\
3 & = f_{\text{c0d}} \\
4 & = f_{\text{vd}} \\
5 & = f_{\text{rd}}
\end{align*}
\]

Element: Element ID:s

Output:
Result: Result for each element from designing with Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
σ_myd: Resistance for pure bending around y-axis \([\text{MPa}]\)
σ_mzd: Resistance for pure bending around z-axis \([\text{MPa}]\)
utz_z: Utilization ratio of each element

__author__ = "bohma", "nicol"
__version__ = "2019.03.15"

import rhinoscriptsyntax as rs

"""6.1 Design of cross-sections subjected to stress in one principal direction"""
"""6.1.6 Bending""

f_myd = strength_d[0]  # Bending stiffness id asumed to be equal about
strong and week axis \([\text{MPa}]\)
f_mzd = strength_d[0]

M_yd = [i * 10**-3 for i in My]  # Maximum moment about y \([\text{MN/m}]\)
M_zd = [i * 10**-3 for i in Mz]  # Maximum moment about z \([\text{MN/m}]\)

Wy = [i * 10**-6 for i in Wy]  # Resistance Moment about strong axis \([\text{m}^3]\)
Wz = [i * 10**-6 for i in Wz]  # Resistance Moment about week axis \([\text{m}^3]\)

Result = []  # Empty list output
Elements_NotOk = []  # Empty list output
do_myd = []  # Empty output list
do_mzd = []  # Empty output list
utz_z = []  # Empty output list

""" Calculates stresses with regard to quantity of Moment around y \(M_{\text{yd}}\)"""
for i in M_yd:
    index = M_yd.index(i)
    sigma_myd = M_yd[index]/Wy[index]  # Bending stress y \([\text{MPa}]\)
    sigma_mzd = M_zd[index]/Wz[index]  # Bending stress z \([\text{MPa}]\)
    k_m = 0.7  # Cross-section parameter \([-\) 0.7 for
rectangular sections, 1.0 for others. From 6.1.6 (2)

"""Expression to satisfy"""

bending1 = abs(sigma_myd)/f_myd + k_m*abs(sigma_mzd)/f_mzd  # eq. 6.11
bending2 = k_m*(abs(sigma_myd)/f_myd) + abs(sigma_mzd)/f_mzd  # eq. 6.12

if bending1 <= 1 and bending2 <= 1:
    Ans = Element[index] + " : OK"
else:
    Ans = Element[index] + " : Not OK"
6.1 Design of cross-sections subjected to stress in one principal direction

6.1.7 Shear

"""Design of cross-sections subjected to stress in one principal direction - Shear stress parallel to grain

Inputs:
- Vz: Shear force Vz [kN]
- Vy: Shear force Vy [kN]
- A_crosssec: Area cross-section [cm2]
  - strength_d: Design value of strength properties [MPa]
    - 0 - f_md
    - 1 - f_t0d
    - 2 - f_t90d
    - 3 - f_c0d
    - 4 - f_c90d
    - 5 - f_vd
    - 6 - f_rd

Element: Element ID:s

Output:
- Result: Result for each element from designing with Eurocode
- Elements_NotOk: Shows elements that are NOT OK according to design criterion
  - τ_yd: Design shear stress y [MPa]
  - τ_zd: Design shear stress z [MPa]
  - utz_z: Utilization ratio of each element
  - utz_y: Utilization ratio of each element"

__author__ = "bohma", "nicol"
__version__ = "2019.04.10"

import rhinoscriptsyntax as rs

"""6.1 Design of cross-sections subjected to stress in one principal direction"
"""6.1.7 Shear

f_vd = strength_d[5] #Compression perpendicular to grains [MPa]
fVy = [i * 10**-3 for i in Vy] #Shear force in local y direction [MN] instead of [kN]
fVz = [i * 10**-3 for i in Vz] #Shear force in local z direction [MN] instead of [kN]
A = [i * 10**-4 for i in A_crosssec] #Convert Area cross-section [m2] instead of [cm2]

Result = [] #Empty list output
Elements_NotOk = [] #Empty list output
""" Calculates stresses with regard to quantity of shear forces V""

for i in fVz:
    index = fVz.index(i)
    tao_z = 1.5*i/A[index]  # Shear stress for rectangular sections [kPa]
    tao_y = 1.5*fVy[index]/A[index]  # Shear stress for rectangular sections [kPa]

usage_z = (abs(tao_z)/k_cr)/f_vd  
usage_y = (abs(tao_y)/k_cr)/f_vd

# Outputs
display = "Element " + str(index) + " Part " + Ans1 + " τ_zd = " + str(tao_z)[:-8]  
Result.extend([display])

tzd.extend([tao_z])  # Answers in MPa
tyd.extend([tao_y])  # Answers in MPa
utz_z.extend([usage_z])
utz_y.extend([usage_y])

6.2 Design of cross-sections subjected to combined stresses

6.2.3 Combined bending and axial tension
6.2.4 Combined bending and axial compression

"""Design of cross-sections subjected to combined stresses - Combined bending and axial Tension/Compression""

Inputs:
- σ_ct0d: Compression/tension resistance [MPa]
- σ_myd: Resistance for pure bending y-axis [MPa]
- σ_mzd: Resistance for pure bending z-axis [MPa]
- strength_d: Design value of strength properties [MPa]
  0 - f_md
  1 - f_t0d
```python
__author__ = "bohma", "nicol"
__version__ = "2019.03.15"

import rhinoscriptsyntax as rs
import math

"""6.2 Design of cross-sections subjected to combined stresses"""
"""6.2.3 Combined bending and axial tension"
"""6.2.4 Combined bending and axial compression"

f_myd = strength_d[0]
# Bending stiffness assumed to be equal about strong and
week axis [MPa]
f_mzd = strength_d[0]
f_c0d = strength_d[3]
# Design Compression capacity along the grain
f_t0d = strength_d[1]
# Design Tension capacity along the grain
k_m = 0.7
# Cross-section parameter [-] 0.7 for rectangular sections,
1.0 for others. From 6.1.6 (2)

Result = []
# Empty list output
Elements_NotOk = []
# Empty list output
utz = []
# Empty output list

""" Calculates stresses with regard to quantity of σ_c0d"
for i in σ_c0d:
    index = σ_c0d.index(i)
    sigma_myd = abs(σ_myd[index])
    sigma_mzd = abs(σ_mzd[index])
    if i > 0:
        """6.2.3 Combined bending and axial tension"
        type = Element[index] + ": Bending & tension "
        sigma_t0d = abs(i)
        comb1 = sigma_t0d/f_t0d + sigma_myd/f_myd + k_m*(sigma_mzd/f_mzd)  # eq. 6.17
        comb2 = sigma_t0d/f_t0d + k_m*(sigma_myd/f_myd) + sigma_mzd/f_mzd  # eq. 6.18
        comb = max(comb1,comb2)
        if comb1 <= 1 and comb2 <= 1:
            Ans = "OK"
        else:
            Ans = " NOT OK"
        NotOk = "Element " + str(index) + ", Part " + type + Ans + " utz = " + str(comb)[:4] + 
        Elements_NotOk.extend([NotOk])
```

2 - f\textsubscript{t90d}
3 - f\textsubscript{c0d}
4 - f\textsubscript{c90d}
5 - f\textsubscript{vd}
6 - f\textsubscript{rd}

Element: Element ID:s
Output:
Result: Result for each element from designing with Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
utz: Utilization ratio of each element"""
else:
    """6.2.4 Combined bending and axial compression"
    type = Element[index] + " : Bending & compression"
    sigma_c0d = abs(i)
    comb1 = (sigma_c0d/f_c0d)**2 + sigma_myd/f_myd + k_m*(sigma_mzd/f_mzd)
    comb2 = (sigma_c0d/f_c0d)**2 + k_m*(sigma_myd/f_myd) + sigma_mzd/f_mzd
    comb = max(comb1,comb2)
    if comb1 <= 1 and comb2 <= 1:
        Ans = " OK"
    else:
        Ans = " NOT OK"
        NotOk = "Element " + str(index) + ", Part " + type + Ans + "; utz = " + str(comb)[1:4]
        Elements_NotOk.extend([NotOk])

# Outputs
    display ="Element " + str(index) + ", Part " + type + Ans + str(comb1) + str(comb2)
    Result.extend([display])
    utz.extend([max(comb1,comb2)])

6.2 Stability of members

6.3.2 Columns subjected to either compression or combined compression and bending

"""Stability of members - Columns subjected to either compression or combined compression and bending
Inputs:
\( \sigma_{ct0d} \): Compression/tension resistance [MPa]
\( \sigma_{myd} \): Resistance for pure bending y-axis [MPa]
\( \sigma_{mzd} \): Resistance for pure bending z-axis [MPa]
\( A_{crossec} \): Area cross section [cm^2]
\( I_y \): Second moment of inertia around y-axis [cm^4]
\( I_z \): Second moment of inertia around z-axis [cm^4]
\( L \): Length of elements [m]
\( \beta \): Buckling multiplier [-]
\( \text{strength}_d \): Design value of strength properties [MPa]
\begin{align*}
0 & - f_{md} \\
1 & - f_{t0d} \\
2 & - f_{t90d} \\
3 & - f_{c0d} \\
4 & - f_{c90d} \\
5 & - f_{vd} \\
6 & - f_{rd}
\end{align*}
\( \text{strength}_k \): Characteristic strength properties [MPa]
\begin{align*}
0 & - f_{mk} \\
1 & - f_{t0k} \\
2 & - f_{t90k} \\
3 & - f_{ck} \\
4 & - f_{c90k} \\
5 & - f_{vk} \\
6 & - f_{rk}
\end{align*}
\( \text{stiffness}_k \): E-modulus [MPa]
\begin{align*}
0 & - E_{005} \\
1 & - E_{9005} \\
2 & - G_{05} \\
3 & - G_{0mean} \\
4 & - G_{90mean} \\
5 & - G_{mean}
\end{align*}
Element: Element ID:s
Element_check: Elements to check
Output:
    Result: Result for each element from designing with Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
utz: Utilization ratio of each element"

__author__ = "bohma", "nicol"
__version__ = "2019.04.11"

import math

"""6.3 Stability of members """
"""6.3.2 Columns subjected to either compression or combined compression and bending""

E_005 = stiffness_k[0]
f_c0k = strength_k[3] #Not used in lambda_rel
f_c0d = strength_d[3]
f_myd = strength_d[0]
f_mzd = strength_d[0]

A = [i * 10**-4 for i in A_crossec] #Area of cross-section [m^2]
L = #Length of elements [m]
I_y = [i * 10**-8 for i in Iy] #Second moment of inertia around y-axis [m^4]
I_z = [i * 10**-8 for i in Iz] #Second moment of inertia around z-axis [m^4]

Result = [] #Empty list output
Elements_NotOk = [] #Empty list output
utz = [] #Empty output list
ratioY = []
ratioZ = []

"""Buckling length """
#beta = 0.7                                  #The Euler bucklinglength depends on the
support condition,
#L_ef = [i * beta for i in L]                # free top-fixed = 2, pinned-pinned = 1,
pinned-fixed = 0.7, fixed-fixed = 0.5

"""Calculates stresses for choosen parts""
for i in Element_check:
    index = i
    sigma_ct0d = σ_ct0d[index]
sigma_myd = abs(σ_myd[index])
sigma_mzd = abs(σ_mzd[index])
    if σ_ct0d[index] >= 0:
        ans = "Element " + str(index) + ", Part " + Element[index] + ": Column
stability not needed, no compressive force acting on part."
#controls that a compression force is applied
Result.extend([ans])
    else:
        element = Element[i]
        #print element
        """Buckling length""
        if element[0] == "C": #For column
            b = beta[0]
        elif element[0] == "B": #For beam
b = beta[1]

**elif** element[0] == "S":  
    #for struts
    b = beta[2]

**elif** element[0] == "C":  
    #for top and bottom chord
    b = beta[3]

#elif element[0] == "[insert letter]": If more element types are added or
different beta for the same type of element are needed, these can be added here
#b = beta[4]

L_ef = L[index]*b
print(index, L_ef)

**"Critical euler load"**

# Characteristic value for E and f_c0
sigma_cry = (math.pi**2)*E_005*I_y[index]/(A[index]*L_ef**2)
# print("critical load ",sigma_cry, sigma_crz)

**"Relative slenderness ratio"**

lambda_rely = math.sqrt(f_c0k/sigma_cry)
lambda_relz = math.sqrt(f_c0k/sigma_crz)
# print("Slenderness ",lambda_rely,lambda_relz)  
# if smaller then 0.3
# no risk for buckling.

**"Reduction parameters"**

beta_c = 0.1  
# Straightness factor:
0.1 for lued laminated timber & LVL, 0.2 for solid timber.

k_y = 0.5*(1+beta_c*(lambda_rely-0.3)+lambda_rely**2)
k_z = 0.5*(1+beta_c*(lambda_relz-0.3)+lambda_relz**2)
print(index, k_y)

k_cy = 1/(k_y+math.sqrt(k_y**2-lambda_rely**2))
k_cz = 1/(k_z+math.sqrt(k_z**2-lambda_relz**2))
print(index, k_cy)

**"Expression to satisfy"**

km = 0.7  
# Cross-section parameter [-] 0.7 for rectangular sections, 1.0 for others.

eq1 = abs(sigma_cry)/(k_cy*f_c0d)+sigma_myd/(k_myd)+km*sigma_mzd/f_mzd
eq2 = abs(sigma_cry)/(k_cz*f_c0d)+km*sigma_myd/(k_myd)+sigma_mzd/f_mzd

eq_max = max(eq1, eq2)

crosection = eq1, eq2
display = "Element " + str(index) + ", Part " + Ans + " utz = " +

eq_max = max(eq1, eq2)

eq_max = max(eq1, eq2)

eq_max = max(eq1, eq2)

eq_max = max(eq1, eq2)

if eq_max <= 1:
    Ans = Element[index] + ": OK"
else:
    Ans = Element[index] + ": NOT OK"
eq = max(eq1, eq2)
NotOk = "Element " + str(index) + ", Part " + Ans + " utz = " +

eq = max(eq1, eq2)
result.extend([display])
print(result)

if eq_max <= 1:
    Ans = Element[index] + ": OK"
else:
    Ans = Element[index] + ": NOT OK"
eq = max(eq1, eq2)
result.extend([display])
print(result)

if eq_max <= 1:
    Ans = Element[index] + ": OK"
else:
    Ans = Element[index] + ": NOT OK"
eq = max(eq1, eq2)
result.extend([display])
print(result)
6.2 Stability of members

6.3.3 Beams subjected to either bending or combined bending and compression

```python
E_005 = stiffness_k[0]
f_md = strength_d[0]
f_c0d = strength_d[3]
f_t0d = strength_d[2]
f_t90d = strength_d[1]
f_c90d = strength_d[4]
I_z = [i * 10**-8 for i in Iz] #Second moment of inertia around z-axis [m^4]
I_tor = [i * 10**-8 for i in Itor] #Second moment of inertia torsion [m^4]
```
\[ W_y = [i \times 10^{-6} \text{ for } i \text{ in } Wy] \quad \# \text{Resistance Moment about strong axis [m}^3]\]

\[
\text{Result} = [] \quad \# \text{Empty list output}
\]

\[
\text{Elements_NotOk} = [] \quad \# \text{Empty list output}
\]

\[
\text{utz} = [] \quad \# \text{Empty output list}
\]

""" Calculates stresses for chosen parts"
"""

""" Calculates stresses with regard to quantity of \( \sigma_{myd} \)"

\[
\text{for } i \text{ in Element_check:}
\text{  index} = i
\text{  element} = \text{Element}[i]
\text{  sigma_cd} = \text{abs}(\sigma_{ct0d}[\text{index}])
\text{  sigma_md} = \text{abs}(\sigma_{myd}[\text{index}])
\text{  if } \text{sigma_md} == 0:
\text{    display} = \text{"Element " + str(index) + ", No bending moment"}
\text{    Result.extend([display])}
\text{  else:}
\text{    #Effective length Table 6.1}
\text{    L_eff} = \text{L[index]}*\text{ratio}
\text{    #Critical moment}
\text{    M_cr} = \text{math.pi*math.sqrt(E_005*I_z[index]*G_005*I_tor[index])/(L_eff*W_y[index])}
\text{    print} M_cr
\text{    #Relative slenderness}
\text{    lambda_relm} = \text{math.sqrt(f_mzk/M_cr)}
\text{    #Reduction parameters}
\text{    beta_c} = 0.1 \quad \# \text{straightness factor: 0.1 for glued laminated timber \\& LVL, 0.2 for solid timber.}
\text{    k_z} = 0.5*(1+beta_c*(lambda_relm-0.3)+lambda_relm**2)
\text{    k_cz} = 1/(k_z+math.sqrt(k_z**2-lambda_relm**2))
\text{    #Expression to satisfy}
\text{    if } lambda_relm <= 0.75:
\text{      k_crit} = 1
\text{    elif } lambda_relm <= 1.4:
\text{      k_crit} = 1.56-0.75*lambda_relm
\text{    else:}
\text{      k_crit} = 1/lambda_relm**2
\text{    if } \text{sigma_cd} == 0:
\text{      lim} = \text{k_crit}*f_md
\text{      if } \text{sigma_md} <= \text{lim:}
\text{        ans} = \text{Element[index]} + \text{"; OK"}
\text{      else:}
\text{        ans} = \text{Element[index]} + \text{"; Not OK"}
\text{        NotOk} = \text{"Element " + str(index) + ", Part " + ans + " utz = " + str(lim)[:4]}
\text{    else:}
\text{      Elements_NotOk.extend([NotOk])}
\]


```
lim = (sigma_md/(k_crit*f_mzd))**2+abs(sigma_cd)/(k_cz*f_c0d)

if lim <= 1:
    ans = Element[index] + " : OK"
else:
    ans = Element[index] + " : Not OK"
NotOk = "Element " + str(index) + ", Part " + ans + " utz = " + str(lim)[:4]
Elements_NotOk.extend([NotOk])

utz.extend([lim])
display = "Element " + str(index) + ", Part " + ans + " utz = " + str(lim)
Result.extend([display])
```

6.4 Design of cross-sections in members with varying cross-sections or curved shape

6.4.2 Single tapered beams

"""Design of cross-sections in members with varying cross-section or curved shape - Single tapered beams

Inputs:
My: Maximum moment about y [kN/m]
Wy: Resistance Moment about strong axis [cm^3]
tan_α: Angle of cross-section variation for single tapered beam
strength_d: Design value of strength properties [MPa]
  0 - f_md
  1 - f_t0d
  2 - f_t90d
  3 - f_c0d
  4 - f_c90d
  5 - f_vd
  6 - f_rd
Element: Element IDs

Output:
Result: Result for each element from designing with Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
utz: Utilization ratio of each element"

__author__ = "bohma", "nicol"
__version__ = "2019.04.11"

import math as m

"""6.4 Design of cross-sections in members with varying cross-section"
"""6.4.2 Single tapered beams"

Wy = [i * 10**-6 for i in Wy] #Resistance Moment about strong axis [m^3]
My_d = [i * 10**-3 for i in My] #Maximum moment about y [MN/m]
tan_alpha = tan_α

f_md = strength_d[0] #Bending stiffness id assumed to be equal about
f_vd = strength_d[5] #Design Shear capacity along the grains [MPa]
f_t90d = strength_d[2] #Design Tension capacity perpendicular to grains [MPa]
f_c90d = strength_d[4] #Design Compression capacity
```

6.4 Design of cross-sections in members with varying cross-sections or curved shape

6.4.2 Single tapered beams

"""Design of cross-sections in members with varying cross-section or curved shape - Single tapered beams

Inputs:
My: Maximum moment about y [kN/m]
Wy: Resistance Moment about strong axis [cm^3]
tan_α: Angle of cross-section variation for single tapered beam
strength_d: Design value of strength properties [MPa]
  0 - f_md
  1 - f_t0d
  2 - f_t90d
  3 - f_c0d
  4 - f_c90d
  5 - f_vd
  6 - f_rd
Element: Element IDs

Output:
Result: Result for each element from designing with Eurocode
Elements_NotOk: Shows elements that are NOT OK according to design criterion
utz: Utilization ratio of each element"

__author__ = "bohma", "nicol"
__version__ = "2019.04.11"

import math as m

"""6.4 Design of cross-sections in members with varying cross-section"
"""6.4.2 Single tapered beams"

Wy = [i * 10**-6 for i in Wy] #Resistance Moment about strong axis [m^3]
My_d = [i * 10**-3 for i in My] #Maximum moment about y [MN/m]
tan_alpha = tan_α

f_md = strength_d[0] #Bending stiffness id assumed to be equal about
f_vd = strength_d[5] #Design Shear capacity along the grains [MPa]
f_t90d = strength_d[2] #Design Tension capacity perpendicular to grains [MPa]
f_c90d = strength_d[4] #Design Compression capacity
""" Calculates stresses with regard to quantity of M_yd"
for i in M_yd:
    index = M_yd.index(i)
    #Bending stress for a single tapered beam [MPa] eq.6.37
    sigma_mad = M_yd[index]/Wy[index]
    if tan_alpha[index] == 0:
        ans = Element[index] + ": Part is not a single tapered beam"
    else:
        if sigma_mad > 0:
            #For tensile stresses parallel to the tapered edge eq. 6.39
            k_ma = 1/m.sqrt(1+(f_md*tan_alpha[index]/(0.75*f_vd))**2+(f_md*tan_alpha[index])**2/f_t90d)**2
            usage = sigma_mad/(k_ma*f_md)
            if usage <= 1:
                ans = Element[index] + ": Tension OK"
            else:
                NotOk = "Element " + str(index) + ", Part " + ans + str(usage)
                Elements_NotOk.extend([NotOk])
        else:
            #For compressive stresses parallel to the tapered edge eq. 6.40
            k_ma = 1/m.sqrt(1+(f_md*tan_alpha[index]/(1.5*f_vd))**2+(f_md*tan_alpha[index])**2/f_c90d)**2
            usage = abs(sigma_mad)/(k_ma*f_md)
            if usage <= 1:
                ans = Element[index] + ": Compression OK"
            else:
                NotOk = "Element " + str(index) + ", Part " + ans + str(usage)
                Elements_NotOk.extend([NotOk])

# Outputs
utz.extend([usage])
display = "Element " + str(index) + ", Part " + ans
Elements_Ok.extend([display])
A.2 Design of Connections
A.2.1 Table Values of Standard Connections

This page is extracted from Martinssons standard connection guide (Martinsson 2018).
A.2. DESIGN OF CONNECTIONS

Infästning balk till pelartopp

BYGGNADER/STOMMAR KONSTRUKTIONSLÖSNINGAR 3.3

PB01 (vfz)
Spikningsplät 3x120x500
Spikas 9+9+22

PB02 (lack)
Plåt 5x120x500
Skruvas 12+12

PB03 (vfz)
Stansad plät 5x140x500
Spikas 30+38

PB04 (vfz)
Stansad plåt 5x200x800
Spikas 48+58

PB05 (vfz)
Stansad plåt 5x280x800
Spikas 90+84

Kapacitet

Lasttyp: Kortvarig
Klimatklass: 1-2

Till spikning används ankarspik 50x4.
Spikarnas placering framgår av skiss och tabell. Spikarna slås i med början i de yttresta hålen i varje rad så att sprickbildning motverkas.
PB02 skruvas med WFR 6.0x60

This page is extracted from Martinsson's standard connection guide (Martinsson 2018).
A.2.2 Calculation of Truss Connection with Recessed Steel Plates

Calculation of a truss connection  
- Between three struts and the chord

This connection is based on a worst case scenario for an arc truss with a span of 15.2 m. The connection is located in the middle of the span, the other connections will have the same or similar design.

Steel: S235, same for dowels and plate

Thickness of steel plate  
\( t := 8 \text{ mm} \)

Dowel diameter  
\( d := 12 \text{ mm} \)

Diagonal member dimension  
\( h_d := 120 \text{ mm} \)
\( b_d := 215 \text{ mm} \)

Vertical member dimension  
\( h_v := 120 \text{ mm} \)
\( b_v := 215 \text{ mm} \)

Chord dimension  
\( h_c := 160 \text{ mm} \)
\( b_c := 215 \text{ mm} \)

Density of timber  
\( \rho_k := 390 \frac{\text{kg}}{\text{m}^3} \)

Characteristic tension capacity  
\( f_{t,0,k} := 19.5 \text{ MPa} \)

Characteristic shear capacity  
\( f_{v,k} := 3.5 \text{ MPa} \)

Number of steel plates  
\( n := 2 \)
\( k_{mod} := 0.8 \)
\( \gamma_{mc} := 1.3 \)

Normal force in each element

\( N_{D1,Rd} := 12 \text{ kN} \)
\( N_{D2,Rd} := 12 \text{ kN} \)
\( N_{V,Rd} := 69 \text{ kN} \)
\( N_{C,Rd} := 163 \text{ kN} \)

Embedment strength of timber - EN 1995-1-1 (8.32)

\[
f_{h,0,k} := 0.082 \left( 1 - 0.01 \frac{d}{\text{mm}} \right) \cdot \rho_k \cdot \frac{m^3}{kg} \cdot \text{MPa} = 28.142 \text{ MPa}
\]
Characteristic yield moment strength for each dowel - EN 1995-1-1 (8.30)

\[ M_{y,Rk} = 0.3 \cdot f_{uk} \cdot d^{2.6} \]

Smallest thickness for internal timber parts - Dimensionering av limträkonstruktioner, Table 13.8, pg.61

\[ t_2 = 1.15 \cdot 4 \cdot \sqrt{\frac{M_{y,Rk}}{f_{h,0,k} \cdot d}} = 79.065 \text{ mm} \]

Smallest thickness for external timber parts - Dimensionering av limträkonstruktioner, Table 13.8, pg.61

\[ t_1 = \sqrt{2} \cdot \sqrt{\frac{M_{y,Rk}}{f_{h,0,k} \cdot d}} = 24.308 \text{ mm} \]

Choses

\[ t_1 = 67 \text{ mm} \quad t_2 = 80 \text{ mm} \]

\[ t_2 \cdot (n - 1) + 2 \cdot t_1 = 214 \text{ mm} \quad \leq \quad b_d = 215 \text{ mm} \]

The characteristic load-carrying capacity per fastener - EN 1995-1-1 (8.7, j,k)

\[ \beta = \frac{f_{h,0,k}}{f_{h,0,k}} = 1 \]

Times two because of two shear planes per fastener

\[ F_{v,dowel,Rk} = 2 \cdot \min \left\{ 1.05 \cdot \frac{f_{h,0,k} \cdot t_1 \cdot d}{2 + \beta} \left( \frac{2 \cdot \beta \cdot (1 + \beta) + \frac{4 \cdot \beta \cdot (2 + \beta) \cdot M_{y,Rk}}{f_{h,0,k} \cdot d \cdot t_1^2} - \beta \right) \right\} = 18.825 \text{ kN} \]

The characteristic load-carrying capacity per steel plate - EN 1995-1-1 (8.9)

\[ F_{v,plate,Rk} = 2 \cdot \min \left\{ \frac{0.4 \cdot f_{h,0,k} \cdot t_1 \cdot d}{1.15 \cdot \sqrt{2} \cdot M_{y,Rk} \cdot f_{h,0,k} \cdot d} \right\} = 18.101 \text{ kN} \]

Total characteristic shear capacity for in connection per dowel

\[ R_k = n \cdot F_{v,dowel,Rk} + (n - 1) \cdot F_{v,plate,Rk} = 55.751 \text{ kN} \]

Total design shear capacity for in connection per dowel

\[ R_d = R_k \cdot \frac{k_{mod}}{\gamma_{mc}} = 34.308 \text{ kN} \]
Minimum distance for dowels in cross section - EN 1995-1-1 (Table 8.5)

\[ a_{1,\text{min}} = 5 \cdot d = 60 \text{ mm} \]
\[ a_{2,\text{min}} = 3 \cdot d = 36 \text{ mm} \]
\[ a_{3,\text{min}} = 7 \cdot d = 84 \text{ mm} \]
\[ a_{4,\text{min}} = 3 \cdot d = 36 \text{ mm} \]

Placement of dowels in connection

\[ a_1 = 100 \text{ mm} \]
\[ a_{C,2} = 44 \text{ mm} \]
\[ a_{V,2} = 36 \text{ mm} \]
\[ a_{D,2} = 36 \text{ mm} \]
\[ a_3 = 84 \text{ mm} \]
\[ a_4 = 36 \text{ mm} \]

Diagonals: \[ a_{D,2} \cdot 1 + a_4 \cdot 2 = 108 \text{ mm} \] \[ < \quad h_d = 120 \text{ mm} \]

Vertical: \[ a_{V,2} \cdot 1 + a_4 \cdot 2 = 108 \text{ mm} \] \[ < \quad h_v = 120 \text{ mm} \]

Chord: \[ a_{C,2} \cdot 2 + a_4 \cdot 2 = 160 \text{ mm} \] \[ < \quad h_c = 160 \text{ mm} \]

Number of rows

\[ n_{r,d} = 2 \quad n_{r,v} = 2 \quad n_{r,c} = 3 \]

Number of effective dowels per row - EN 1995-1-1 (8.34)

number of dowels in each row \( n_d = 2 \)

Diagonal 1 & 2

\[ n_{d,ef} = \min \left( n_d, n_d^{0.9} \cdot \frac{a_1}{13 \cdot d} \right) = 1.67 \]

Vertical

\[ n_{v,ef} = \min \left( n_d, n_d^{0.9} \cdot \frac{a_1}{13 \cdot d} \right) = 1.67 \]

Chord

\[ n_{c,ef} = \min \left( n_d, n_d^{0.9} \cdot \frac{a_1}{13 \cdot d} \right) = 1.67 \]
Load carrying capacity for each element

\[
\begin{align*}
\text{Diagonal 1 & 2} & \quad n_{r,d} \cdot n_{d,ef} \cdot R_d = 114.571 \text{ kN} \quad > \quad N_{D1,Rd} = 12 \text{ kN} \\
& \quad N_{D2,Rd} = 12 \text{ kN} \\
\text{Vertical} & \quad n_{r,v} \cdot n_{v,ef} \cdot R_d = 114.571 \text{ kN} \quad > \quad N_{V,Rd} = 69 \text{ kN} \\
\text{Chord} & \quad n_{r,c} \cdot n_{c,ef} \cdot R_d = 171.856 \text{ kN} \quad > \quad N_{C,Rd} = 163 \text{ kN}
\end{align*}
\]

**Blockskjuvning**

Net area of tension area

\[
\begin{align*}
\text{Diagonals} & \quad A_{D,net,t} := (a_{D,2} - d) \cdot (b_d - 2 \cdot t) = \left(4.776 \times 10^3\right) \text{ mm}^2 \\
\text{Tension chord} & \quad A_{C,net,t} := (a_{C,2} - d) \cdot (b_c - 2 \cdot t) = \left(6.368 \times 10^3\right) \text{ mm}^2
\end{align*}
\]

Net area of shear area

\[
\begin{align*}
\text{Diagonals} & \quad A_{D,net,s} := (a_1 + a_3 - 2 \cdot d) \cdot (b_d - 2 \cdot t) = \left(3.184 \times 10^4\right) \text{ mm}^2 \\
\text{Tension chord} & \quad A_{C,net,s} := (2 \cdot a_1 - d) \cdot (b_c - 2 \cdot t) = \left(3.741 \times 10^4\right) \text{ mm}^2
\end{align*}
\]

\[
\begin{align*}
F_{D,bs,RI} &= \max \left[ 1.5 \cdot A_{D,net,t} \cdot f_{t,0,k} \right. \\
& \quad 0.7 \cdot A_{D,net,s} \cdot f_{s,k} \left. \right] = 139.698 \text{ kN} \quad > \quad N_{D1,Rd} = 12 \text{ kN} \quad \text{OK} \\
F_{C,bs,RI} &= \max \left[ 1.5 \cdot A_{C,net,t} \cdot f_{t,0,k} \right. \\
& \quad 0.7 \cdot A_{C,net,s} \cdot f_{s,k} \left. \right] = 186.264 \text{ kN} \quad > \quad N_{C,Rd} = 163 \text{ kN} \quad \text{OK}
\end{align*}
\]

The proposed connection will manage the affecting section forces

According to *Dimensionering av limträkonstruktioner*, the total number of dowels should be increased by 10% even tough the conditions are satisfied. Alternatively the load are increased by 10%

\[
\begin{align*}
\text{Diagonal 1 & 2} & \quad n_{r,d} \cdot n_{d,ef} \cdot R_d = 114.571 \text{ kN} \quad > \quad N_{D1,Rd} \cdot 1.1 = 13.2 \text{ kN} \quad \text{OK} \\
& \quad N_{D2,Rd} \cdot 1.1 = 13.2 \text{ kN} \quad \text{OK} \\
\text{Vertical} & \quad n_{r,v} \cdot n_{v,ef} \cdot R_d = 114.571 \text{ kN} \quad > \quad N_{V,Rd} \cdot 1.1 = 75.9 \text{ kN} \quad \text{OK} \\
\text{Chord} & \quad n_{r,c} \cdot n_{c,ef} \cdot R_d = 171.856 \text{ kN} \quad > \quad N_{C,Rd} \cdot 1.1 = 179.3 \text{ kN} \quad \text{OK}
\end{align*}
\]

\[
\begin{align*}
F_{D,bs,RI} &= \max \left[ 1.5 \cdot A_{D,net,t} \cdot f_{t,0,k} \right. \\
& \quad 0.7 \cdot A_{D,net,s} \cdot f_{s,k} \left. \right] = 139.698 \text{ kN} \quad > \quad N_{D1,Rd} \cdot 1.1 = 13.2 \text{ kN} \quad \text{OK} \\
F_{C,bs,RI} &= \max \left[ 1.5 \cdot A_{C,net,t} \cdot f_{t,0,k} \right. \\
& \quad 0.7 \cdot A_{C,net,s} \cdot f_{s,k} \left. \right] = 186.264 \text{ kN} \quad > \quad N_{C,Rd} \cdot 1.1 = 179.3 \text{ kN} \quad \text{OK}
\end{align*}
\]
Appendix B

Grasshopper model
B.1 Grasshopper Canvas
B.2 Results

B.2.1 Framework with Rectangular Cross Section

Deformation of structure [mm]:

Moment distribution for structure [kNm]:

-581
-1.18

6.49
B.2.2 Framework with Rectangular cross Section and Strut

Deformation of structure [mm]:

Moment distribution for structure [kNm]:

-18.3

-29.2

-551.4

46.2
B.2.3 Framework with Single Tapered Cross Sections

Deformation of structure [mm]:

Moment distribution for structure [kNm]:

B.2.4 Framework with Single Tapered Cross Sections and Strut

Deformation of structure [mm]:

Moment distribution for structure [kNm]:
B.2.5 Framework with Truss Type I

Deformation of structure [mm]:

Moment distribution for structure [kNm]:

-22.3 27.4 19.9 27.4 -10.9 -11 -22.3 7.8 1.4
B.2.6 Framework with Truss Type II

Deformation of structure [mm]:

Moment distribution for structure [kNm]:
Appendix C

Robot Model
C.1 Results

C.1.1 Framework with Truss Type I

Deformation of structure [mm]:

Moment distribution for structure [kNm]:
C.2 Robot Calculation Report

TIMBER STRUCTURE CALCULATIONS

ANALYSIS TYPE: Member Verification

CODE GROUP:  
MEMBER: 15 Column_15  
POINT: 1  
COORDINATE: x = 0.00 L = 0.00 m

LOADS:  
Governing Load Case: 5 ULS2 Manual 1*1.00+3*1.36+4*0.49+2*1.21

MATERIAL  
GL30c  
gM = 1.25  
f m,0,k = 30.00 MPa  
f t,0,k = 19.50 MPa  
f c,0,k = 24.50 MPa  
E 0.05 = 10800.00 MPa

SECTION PARAMETERS: column  
ht=55.0 cm  
bf=20.6 cm  
tw=10.3 cm  
tf=10.3 cm

STRESSES  
ALLOWABLE STRESSES

Factors and additional parameters
kh = 1.10  
kh_y = 1.01  
kmod = 0.80  
Ksys = 1.00  
kcr = 0.67

ANCHORING PARAMETERS:

BUCKLING PARAMETERS:

LIMITATIONAL PARAMETERS:

VERIFICATION FORMULAS:

Section OK !!!
TIMBER STRUCTURE CALCULATIONS

ANALYSIS TYPE: Member Verification

CODE GROUP: MEMBERS: 18 Beam_18 POINT: 1 COORDINATE: x = 0.00 L = 0.00 m

LOADS: Governing Load Case: 5 ULS2 Manual 1*1.00+3*1.36+4*0.49+2*1.21

MATERIAL GL30c
\[ g_M = 1.25 \]
\[ f_{m,0,k} = 30.00 \text{ MPa} \]
\[ f_{t,0,k} = 19.50 \text{ MPa} \]
\[ f_{c,0,k} = 24.50 \text{ MPa} \]
\[ f_{v,k} = 3.50 \text{ MPa} \]
\[ f_{t,90,k} = 0.50 \text{ MPa} \]
\[ f_{c,90,k} = 2.50 \text{ MPa} \]
\[ E_{0,moyen} = 13000.00 \text{ MPa} \]
\[ E_{0,05} = 10800.00 \text{ MPa} \]
\[ G_{moyen} = 650.00 \text{ MPa} \]
Service class: 1
\[ \beta_c = 1.00 \]

SECTION PARAMETERS:

TopChord:
\[ h_t = 24.0 \text{ cm} \]
\[ b_f = 20.6 \text{ cm} \]
\[ t_w = 10.3 \text{ cm} \]
\[ t_f = 10.3 \text{ cm} \]
\[ A_y = 329.60 \text{ cm}^2 \]
\[ A_z = 329.60 \text{ cm}^2 \]
\[ A_x = 494.40 \text{ cm}^2 \]
\[ I_y = 23731.20 \text{ cm}^4 \]
\[ I_z = 17483.63 \text{ cm}^4 \]
\[ I_x = 32117.4 \text{ cm}^4 \]

STRESSES

\[ \sigma_{c,0,d} = \frac{N}{A_x} = 175.11/494.40 = 3.54 \text{ MPa} \]
\[ \sigma_{m,y,d} = \frac{M_Y}{W_y} = 27.70/1977.60 = 14.01 \text{ MPa} \]
\[ \tau_{z,d} = \frac{1.5*38.63}{494.40} = 1.17 \text{ MPa} \]

Factors and additional parameters
\[ k_h = 1.10 \]
\[ k_{h,y} = 1.10 \]
\[ k_{mod} = 0.80 \]
\[ K_{sys} = 1.00 \]
\[ k_{cr} = 0.67 \]

LATERAL BUCKLING PARAMETERS:
\[ l_{ef} = 3.40 \text{ m} \]
\[ \lambda_{rel m} = 0.34 \]
\[ \sigma_{cr} = 267.00 \text{ MPa} \]
\[ k_{crit} = 1.00 \]

BUCKLING PARAMETERS:

About Y axis:  

About Z axis:

VERIFICATION FORMULAS:
\[ (\sigma_{c,0,d}/f_{c,0,d})^2 + (\sigma_{m,y,d}/f_{m,y,d})^2 = 0.72 < 1.00 \quad (6.19) \]
\[ (\sigma_{m,y,d}/k_{crit}f_{m,y,d}) = 0.67 < 1.00 \quad (6.33) \]
\[ (\tau_{z,d}/k_{cr}f_{v,d}) = 0.78 < 1.00 \quad (6.13) \]

Section OK !!!
TIMBER STRUCTURE CALCULATIONS

ANALYSIS TYPE: Member Verification

CODE GROUP: Mesh Group
MEMBER: 12 Timber Beam_12
POINT: 2 COORDINATE: x = 0.50 L = 1.74 m

LOADS:
Governing Load Case: 5 ULS2 Manual 1*1.00+3*1.36+4*0.49+2*1.21

MATERIAL
GL30c
gM = 1.25
f m,0,k = 30.00 MPa
f t,0.k = 19.50 MPa
f c,0,k = 24.50 MPa
f v,k = 3.50 MPa
f t,90,k = 0.50 MPa
f c,90.k = 2.50 MPa
E 0,moyen = 13000.00 MPa
E 0,05 = 10800.00 MPa
G moyen = 650.00 MPa
Service class: 1
Beta c = 0.10

SECTION PARAMETERS:
Strut
ht=12.0 cm
bf=20.6 cm
Ay=164.80 cm2
Az=164.80 cm2
Ax=247.20 cm2
Iy=2966.40 cm4
Iz=8741.82 cm4
Ix=-2849.8 cm4
Wy=494.40 cm3
Wz=848.72 cm3

STRESSES

Factors and additional parameters
kh = 1.10
kh_y = 1.10
kmod = 0.80
Ksys = 1.00

LATERAL BUCKLING PARAMETERS:

LY = 3.48 m
Lambda Y = 100.35
Lambda,rel Y = 1.52
kY = 1.72
LFY = 3.48 m
kcy = 0.40

LZ = 3.48 m
Lambda Z = 58.46
Lambda,rel Z = 0.89
kz = 0.92
LFZ = 3.48 m
kcz = 0.85

VERIFICATION FORMULAS:
Sig_c,0,d/(kc,y*f c,0,d) + Sig_m,y,d/f m,y,d = 2.77/(0.40*15.68) + 0.03/21.12 = 0.45 < 1.00 (6.23)

Section OK !!!