

# Structural optimization for effective strut-and-tie models

Design of support crossbeams in single girder  
concrete bridges

Master of Science Thesis in the Master Program Structural Engineering and  
Building Technology

SIMON NILSSON  
PETTER ÖHMAN



MASTER'S THESIS ACEX30-19-77

# Structural optimization for effective strut-and-tie models

Design of support crossbeams in single girder concrete bridges

SIMON NILSSON  
PETTER ÖHMAN



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Architecture and Civil Engineering  
*Division of Structural Engineering*  
Concrete Structures  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2019

Structural optimization for effective strut-and-tie models  
Design of support crossbeams in single girder concrete bridges

SIMON NILSSON  
PETTER ÖHMAN

© SIMON NILSSON, PETTER ÖHMAN, 2019.

Supervisors: Max Fredriksson, Inhouse Tech  
Alexandre Mathern, Department of Architecture and Civil Engineering  
Mario Plos, Department of Civil and Environmental Engineering  
Examiner: Mario Plos, Department of Civil and Environmental Engineering

Department of Architecture and Civil Engineering  
Division of Structural Engineering  
Concrete Structures  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: 3D structural optimized crossbeam, optimized using BRIGADE/Plus Version 6.2-5 and Python 2.7, studied as a case study in the thesis.

Department of Architecture and Civil Engineering  
Gothenburg, Sweden 2019

Structural optimization for effective strut-and-tie models  
Design of support crossbeams in single girder concrete bridges  
SIMON NILSSON  
PETTER ÖHMAN  
Department of Architecture and Civil Engineering  
Chalmers University of Technology

## Abstract

The strut and tie (ST) method is a method often used by structural engineers when designing reinforced concrete members with nonlinear stress flows. In bridge design, crossbeams for support by abutments is a typical example of a structure where the ST method often is applied. Experience from engineering practice indicates that the ST models used might be over-conservative, leading to an oversized and costly reinforcement design. Therefore, there is a need to investigate if the method and its applications can be improved and optimized.

The purpose of this master thesis was to investigate if structural optimization can be used to find more effective ST models and thereby understand how today's ST method can be improved. This was achieved by performing a case study on a crossbeam in a bridge, previously designed by the civil engineering company Inhouse Tech AB.

The structural optimization method used in this thesis was a bi-directional evolutionary structural optimization (BESO). It was applied on the crossbeam using both a python-script, that was implemented in the Abaqus 6.14-2 extension BRIGADE/Plus version 6.2-5, and a MATLAB-script. Both scripts are based on existing scripts, written by different authors, and implemented with a few modifications. For BRIGADE/Plus version 6.2-5, both 2D and 3D models were built, while the MATLAB-script only performed the optimization for the 2D case. The ST models obtained, based on the finite element (FE) analyses, were then assessed following the guidelines provided by Eurocode 2. These ST models was finally compared to a "conventional" ST analysis performed on the same crossbeam.

Several comparisons were made between different FE-analyses and hand calculations to see which ST model led to the optimal way to design the reinforcement. It was found that in the models constructed with inclined ties the total reinforcement amount could be reduced.

Keywords: strut, tie, BESO, structural optimization, BRIGADE, Abaqus, Eurocode, crossbeam, finite element modelling, reinforcement design

Structural optimization for effective strut-and-tie models  
Design of support crossbeams in single girder concrete bridges  
SIMON NILSSON  
PETTER ÖHMAN  
Institution för arkitektur- och samhällsbyggnadsteknik  
Chalmers tekniska högskola

## Sammanfattning

Strut- and tie- (ST-) metoden används ofta av konstruktörer, vid dimensionering av armering i konstruktionselement med icke-linjära spänningsflöden. Vid brokonstruktion är tvärbalkar vid stöd ett typiskt exempel på konstruktionselement på vilket ST-metoden ofta tillämpas. Yrkeserfarenheter visar att de ST-modeller som används kan vara överkonservativa, vilket leder till en överdimensionerad och dyr armeringsutformning. Därför är det nödvändigt att undersöka om metoden och dess applikationerna kan förbättras och optimeras.

Syftet med detta examensarbete var att undersöka om strukturoptimering kan användas för att hitta mer effektiva ST-modeller och därigenom förstå hur dagens ST-metod kan förbättras. Detta uppnåddes genom att genomföra en fallstudie på en tvärbalk i en bro, tidigare konstruerad av konsultbolaget Inhouse Tech AB.

Strukturoptimeringsmetoden som användes i detta examensarbete var en så kallad "Bi-directional evolutionary structural optimization" (BESO). Den applicerades på tvärbalken genom att använda dels ett python-script, som implementerades i BRIGADE/Plus version 6.2-5, och dels ett MATLAB-script. Båda scripten är baserade på befintliga script, skrivna av olika författare, och implementerade med några få modifieringar. För BRIGADE/Plus version 6.2-5 byggdes både 2D- och 3D-modeller, medan MATLAB-scriptet endast utförde optimering för 2D-fallet. De erhållna ST-modellerna, som baserades på FE-analyserna, utvärderades sedan enligt riktlinjerna från Eurocode 2. Slutligen jämfördes dessa ST-modeller med en "konventionell" ST-analys som utfördes på samma tvärbalk.

Flera jämförelser gjordes mellan olika FE-analyser och handberäkningar för att se vilken ST-modell som ger den optimala armeringsutformningen. Det visade sig att den totala mängden armering kunde minskas för ST modeller med lutande dragband.

Nyckelord: fackverksmodell, trycksträva, dragband, BESO, strukturoptimering, BRIGADE, Abaqus, Eurocode, tvärbalk, FE-modellering, dimensionering av armering



# Preface

This master thesis was conducted from January 2019 to June 2019, at the civil engineering company Inhouse Tech AB in Gothenburg. It is the final and conclusive work of the master of science program Structural Engineering and Building Technology at Chalmers University of Technology.

We would like to extend our gratitude to our supervisors Max Fredriksson at Inhouse Tech AB, Alexandre Mathern at Chalmers University of Technology and Mario Plos at Chalmers University of Technology for the support through the project and to all the constructive feedback we have received.

We would also like to thank all the people working at Inhouse Tech AB for their hospitality and for taking their time discussing problems that arose during the project.

Finally we would like to thank our opponents Michaela Henriksson and Anna Rosberg for their constructive feedback.

Simon Nilsson, Petter Öhman, Gothenburg, June 2019



# Contents

<b>Abstract</b>	<b>xii</b>
<b>Sammanfattning</b>	<b>xii</b>
<b>Preface</b>	<b>xii</b>
<b>Abbreviations</b>	<b>xiii</b>
<b>Notations</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Aim and objectives . . . . .	2
1.3 Limitations . . . . .	2
1.4 Method . . . . .	3
<b>2 Theory</b>	<b>4</b>
2.1 Discontinuity regions . . . . .	4
2.2 The Strut and tie method . . . . .	5
2.2.1 The load path method . . . . .	6
2.2.1.1 The development of a load path . . . . .	6
2.2.1.2 Rules for the load path method . . . . .	6
2.2.2 ST models developed from FE analyses . . . . .	7
2.2.3 ST method design according to Eurocode . . . . .	7
2.2.3.1 Angles between struts and ties . . . . .	7
2.2.3.2 Struts . . . . .	8
2.2.3.3 Ties . . . . .	9
2.2.3.4 Nodes . . . . .	9
2.2.4 Truss models for high beams . . . . .	10
2.2.4.1 Truss model with vertical ties . . . . .	11
2.2.4.2 Truss model with inclined ties . . . . .	11
2.3 Finite element method . . . . .	11
2.3.1 The FEM in general . . . . .	11
2.3.2 FEM implementation in BRIGADE/Plus . . . . .	12
2.4 Structural optimization . . . . .	12
2.4.1 Mathematical form of structural optimization . . . . .	14
2.4.2 Different types of structural optimization . . . . .	16

2.4.3	Bi-directional evolutionary structural optimization . . . . .	17
2.4.3.1	Optimization algorithm . . . . .	18
2.4.3.2	Mesh dependency and convergence . . . . .	20
2.5	Python . . . . .	20
2.6	MATLAB and CALFEM . . . . .	21
<b>3</b>	<b>Case study and applied methods</b>	<b>22</b>
3.1	Description of the studied beam . . . . .	23
3.2	Analytical development of Model A and B . . . . .	24
3.2.1	Development of Model A . . . . .	24
3.2.2	Development of Model B . . . . .	25
3.2.3	Verification of Model A using CALFEM . . . . .	26
3.3	Development of Model C - 1 using structural optimization in MATLAB	26
3.3.1	Model C - 1 setup . . . . .	26
3.3.2	BESO implementation using MATLAB . . . . .	27
3.4	Development of Model C - 2 and Model C - 3 using structural opti- mization in BRIGADE/Plus . . . . .	28
3.4.1	Model C - 2 setup . . . . .	28
3.4.2	Model C - 3 setup . . . . .	29
3.4.3	BESO implementation using Python . . . . .	31
3.4.3.1	Saving the filter map . . . . .	33
3.4.3.2	Producing viewports . . . . .	34
3.4.3.3	Verification of the python code . . . . .	34
3.5	Development of Model C based on the sub-models . . . . .	35
3.5.1	Translation of the results from the sub-models . . . . .	35
3.5.2	The resulting Model C . . . . .	36
3.6	Reinforcement design based on Model A, B and C . . . . .	37
<b>4</b>	<b>Results</b>	<b>39</b>
4.1	Results from Model A . . . . .	39
4.1.1	Forces in struts and ties from Model A . . . . .	39
4.1.2	Verification of hand calculations for Model A using CALFEM	40
4.1.3	Check of critical nodes for Model A . . . . .	40
4.1.4	Reinforcement design for Model A . . . . .	41
4.2	Results from Model B . . . . .	42
4.2.1	Forces in struts and ties for Model B . . . . .	42
4.2.2	Check of critical nodes for Model B . . . . .	42
4.2.3	Reinforcement design for Model B . . . . .	43
4.3	Results from the sub models on which Model C are based . . . . .	44
4.3.1	Model C - 1 . . . . .	44
4.3.2	Model C - 2 . . . . .	46
4.3.3	Model C - 3 . . . . .	47
4.4	Results from Model C . . . . .	50
4.4.1	Forces in struts and ties from Model C . . . . .	50
4.4.2	Check of critical nodes for Model C . . . . .	50
4.4.3	Reinforcement design for Model C . . . . .	51

<b>5</b>	<b>Discussion</b>	<b>52</b>
5.1	Model A . . . . .	52
5.2	Model B . . . . .	53
5.3	Model C . . . . .	53
5.3.1	Model C - 1 . . . . .	54
5.3.2	Model C - 2 . . . . .	54
5.3.3	Model C - 3 . . . . .	54
5.3.4	Convergence of the sub-models . . . . .	55
5.4	Comparison between the models . . . . .	56
<b>6</b>	<b>Conclusion</b>	<b>59</b>
<b>7</b>	<b>Further work</b>	<b>61</b>
	<b>References</b>	<b>62</b>
	<b>Appendix</b>	<b>I</b>
<b>A</b>	<b>Hand calculations</b>	<b>I</b>
A.1	Hand calculations on Model A . . . . .	I
A.2	Hand calculations on Model B . . . . .	X
A.3	Hand calculations on Model C . . . . .	XIX

# Abbreviations

2D	Two dimensional
3D	Three dimensional
BESO	Bi-directional evolutionary structural optimization
D-regions	Discontinuity regions
ESO	Evolutionary structural optimization
FE	Finite element
FEA	Finite element analysis
FEM	Finite element method
IHT	Inhouse Tech, civil engineering company
m-script	MATLAB script
Model A	The ST model obtained by using the ST method, as applied by Inhouse Tech AB
Model B	The ST model obtained by using a different ST approach, where the ties are placed with an inclination instead of vertical, as in Model A
Model C	The ST model are developed using structural optimization based on different FE analyses
PDE	Partial differential equations
SIMP	Solid isotropic material penalization
SLS	Service limit state
ST	Strut and tie
ULS	Ultimate limit state

# Notations

## Roman upper case letters

$A_s$	Cross-sectional area of reinforcement
$C$	Objective function in BESO or compressive force in strut
$E_0$	Initial young's modulus
$E_e$	Artificial young's modulus
$ER$	Evolutionary ratio
$F$	Force vector
$\mathbf{K}(\mathbf{x})$	Stiffness matrix
$N$	Number of FE:s studied in homogenization
$P$	External force in ties
$R$	Reaction force
<b>SO</b>	General form of structural optimization
$T$	Tensile force in tie
$V$	Volume of design domain
$V_f$	Volume fraction
$V^{k+1}$	Volume evolution
$\bar{V}_s$	Upper volume limit in homogenization

## Roman lower case letters

$a_e$	Width of a void in homogenization
$b_e$	Height of a void in homogenization
$d$	Extension of a stressed region describing the extension of a disturbed region
$f$	Objective function in general structural optimization
$f_{cd}$	Design compressive strength of concrete
$f_{ck}$	Characteristic compressive strength of concrete
$f_{yd}$	Design yield strength of reinforcement steel
$h$	Highest part of a D-region
$k_0$	Solid element stiffness matrix
$k_1$	National determined parameter governing CCC-nodes
$k_2$	National determined parameter governing CCT-nodes
$k_3$	National determined parameter governing CTT-nodes
$p$	Penalization factor
$r_{min}$	Filter radius

$u$	Displacement vector
$u_e$	Elemental displacement vector
$v_e$	Elemental volume
$w_i$	Weight factor used in structural optimization
$x_e$	Design variable in BESO
$x$	State variable in general structural optimization
$y$	Design variable in general structural optimization

## Greek letters

$\alpha$	Deviation angle for stresses under concentrated forces
$\alpha_{cc}$	Coefficient considering long term effects
$\alpha_{th}$	Sensitivity threshold
$\hat{\alpha}$	Mesh independent element sensitivity
$\gamma_c$	Partial factor for concrete in ordinary situations
$\nu'$	National determined parameter
$\Pi$	Potential energy in homogenization
$\theta_1$	Angle for strut meeting a single tie
$\theta_2$	Angle for a strut between two perpendicular ties
$\theta_e$	Orientation of of the element in homogenization
$\rho_e$	Artificial element density
$\sigma_{Rd,max}$	Maximal compressive stress that a strut can carry

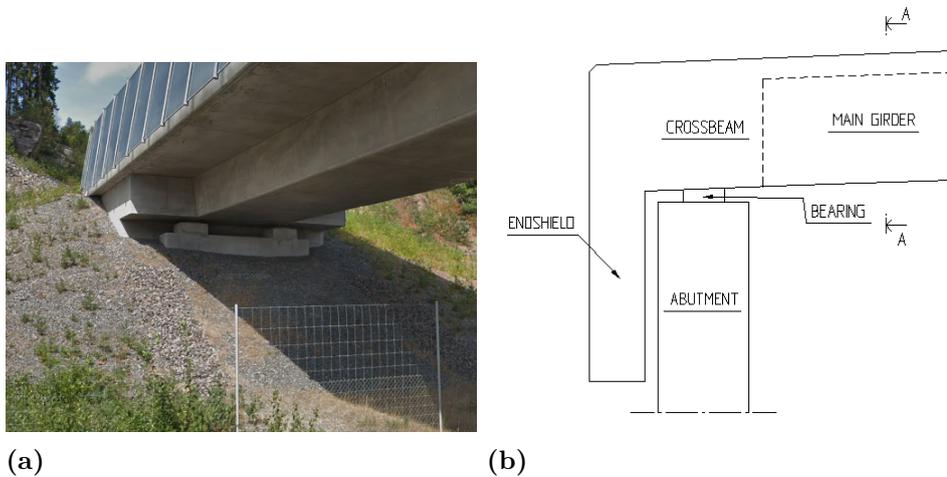


# 1

## Introduction

### 1.1 Background

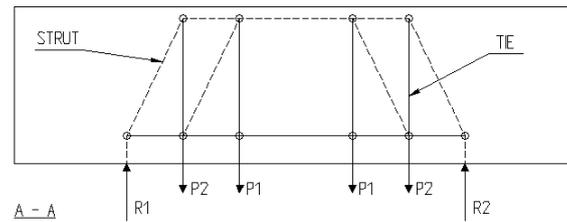
The Strut and tie (ST) method can be used to simulate the stress flow for discontinuity regions (D-regions) in reinforced concrete structures (Engström 2011). This method is commonly used in today's engineering practice. Experience indicates that the ST method is over-conservative<sup>1</sup>, particularly in cases with three dimensional flow of forces (Mathern et al. 2017). One such case is crossbeams in bridge structures. Figure 1.1a shows an example of how a bridge end support can be built up and in Figure 1.1b an describing sketch of a elevation view is visualized. The load from the superstructure is transferred to the abutments by a crossbeam supported on bearings.



**Figure 1.1:** Visualization of the part of bridge studied in this thesis (a) Picture of an end support in an existing bridge (b) Sketch of the assembly of the end support

Section A-A from Figure 1.1b, visualized in Figure 1.2, presents one example of how the ST method can be implemented on the crossbeam. The procedure from this stage is to calculate the forces in the ties and then design the reinforcement according to the capacity required.

<sup>1</sup>Max Fredriksson, Inhouse Tech AB



**Figure 1.2:** Example of how the ST method can be implemented on crossbeams

An over-conservative method leads to an oversized and costly reinforcement design. Therefore, there is a reason to investigate if the ST method can be improved and optimized, to be more efficient with respect to economy, environment and design and construction efficiency. One way to find a more effective ST model can be to use structural optimization.

The idea with structural optimization is to get a better understanding of how the forces flow within the structural element. Instead of guessing how the forces flow in loaded members, this optimization method can be used to get a more accurate picture, and then the ST model can be developed according to the simulated flow of forces.

## 1.2 Aim and objectives

The aim of this master thesis was to investigate the potential in using structural optimization to find effective ST models.

The main objectives of the study were to:

- Perform a literature study regarding ST modelling and structural optimization.
- Perform a case study where the knowledge obtained in the literature study is applied on a crossbeam in an existing bridge.
- From the case study obtain a ST model using the ST method and perform calculations following the guidelines given by Eurocode.
- From the case study obtain a ST model using structural optimization and perform calculations following the guidelines given by Eurocode.
- Compare the obtained ST models to see if the ST model obtained from structural optimization is more efficient than the ST model obtained from the conventional ST method.

## 1.3 Limitations

- The crossbeam presented in the case study, with geometry and load cases, was obtained from documentation at Inhouse Tech AB through supervisor Max Fredriksson.
- Only the reinforcement design of the crossbeam was investigated, even though the ST method can be applied for other members of the bridge as well.

- The difference in response due to different concrete and steel qualities was not investigated.
- The European standard (*Eurocode 2* 2008) was used during this project.
- Only design methods for the ultimate limit state (ULS) were studied during this project.

## 1.4 Method

Literature studies were performed during the first half of the project. They provided deeper understanding regarding design of D-regions, the ST method and how to interpret Eurocode. They also provided new knowledge regarding different structural optimization methods, on which much of this thesis is based on. Some literature was provided by the research group (Concrete Structures) but most of it was found by searches through the Chalmers library database and on the internet.

By using the ST method according to *Eurocode 2* (2008)<sup>2</sup> with vertical stirrups a ST model was developed. This ST model served as a basis for comparison with the ST models obtained from the FE analyses, on which structural optimization was applied.

FE analyses were conducted to obtain more efficient ST models. In those analyses, structural optimization theory was applied. A plugin script was developed, based on the python script created by Zuo & Xie (2015), to be executed in the FE program BRIGADE/Plus version 6.2-5. In addition, a similar routine was implemented in MATLAB, to verify the ST models obtained by the BRIGADE/Plus simulations. This MATLAB-script (m-script) was written by Xia et al. (2016) and some minor changes were done to make it suitable for this thesis purpose. The scripts performed structural topology optimization and was used to obtain effective load paths, which could be translated into ST models.

The ST models, obtained from the structural topology optimization FE analyses, was checked to the requirements in *Eurocode 2* (2008) and compared to the ST model obtained from common ST methodology. Thus, it was possible to compare the methods currently used in engineering practice to the developed optimized ST models obtained by the FE analyses.

---

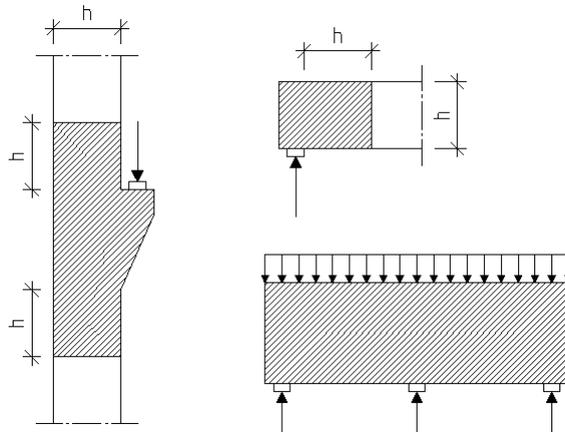
<sup>2</sup>as applied at Inhouse Tech AB

# 2

## Theory

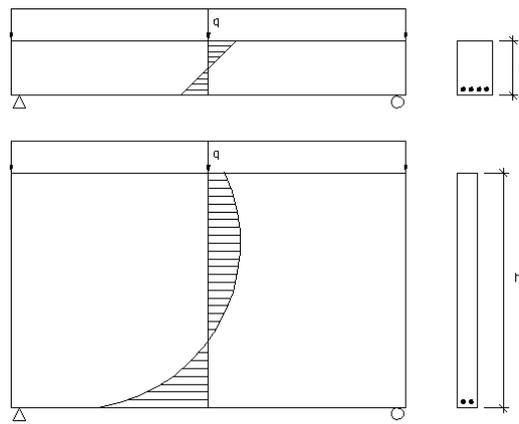
### 2.1 Discontinuity regions

As a result of geometric and/or static discontinuities, regular Bernoulli beam theory can not be applied to certain regions of (concrete) structures, appearing in e.g. deep beams and corbels, Engström (2011). These regions are called disturbed- or discontinuity-regions, further referred to as D-regions. A D-region is characterized by a non-linear stress distribution over the cross-section. These regions can be described by Saint-Venant's principle of body subjected to a system of forces in equilibrium. Saint-Venant stresses will appear as local effects in the body and will, according to *Eurocode 2* (2008), be extended to  $d \approx h$  where  $h$  is the highest part of the subjected section (see Figure 2.1).



**Figure 2.1:** Examples of different D-regions, based on Engström (2011)

According to *Eurocode 2* (2008), a beam should be treated as a deep beam if the span length is shorter than three times the overall section depth. For a deep beam, the stress and strain distribution becomes non-linear (see Figure 2.2) and the whole beam can be considered as a D-region.



**Figure 2.2:** Normal strain distribution in a beam (top) vs. deep beam (bottom), based on Engström (2011)

## 2.2 The Strut and tie method

The strut and tie method (ST) is a lower bound approach used to simulate non-linear stress fields in D-regions (described in section 2.1) in ultimate limit state (ULS) in reinforced concrete. The ST method is based on the theory of utilizing the compressive strength of concrete and adding reinforcement in tensile zones of the structure. It is recommended that the ST models is based on the stress field under linear elastic stress field (Engström 2011). The reason for this is that:

- The deformation capacity of concrete is limited by its ability of plastic rotation.
- The requirements under SLS-loading should be fulfilled.

The ST method is suitable for reinforcement design in an early stage since the inputs needed are few (Dahl 2018); geometry, load cases and concrete stiffness is enough to get an appropriate model. The inputs needs to be combined using the following assumptions:

- The stress field should be in equilibrium with the current load case.
- No regions should be subjected to stresses above their plastic capacity.
- The material should be assumed to be ideal plastic.

To develop an appropriate ST model based on linear elastic stress field, there are two main approaches:

- To use the load path method (Section 2.2.1).
- To perform FE analyses from which the stress trajectories or principal stresses can be used to develop an ST model (Section 2.2.2).

### 2.2.1 The load path method

The load path method can be used to simulate and simplify the stress field in deep beams and other D-regions. This method is based on inserting single load path that corresponds to the flow of the stresses (Engström 2011).

For a simply supported deep beam, with uniformly distributed load, the shear force is zero at mid-span. In this section, the load is divided so that half of the load goes to each support. This division of the loads creates a unique stress field which then is translated to a unique load path from the applied load to the support. There are steps and rules to be followed in the load path method and this procedure will enable the development of a ST model.

#### 2.2.1.1 The development of a load path

To achieve appropriate shape of the load path, Engström (2011) recommends to perform the following steps:

1. Identify the proportion of the load that goes to each support and where the load dividers are located.
2. Sketch the stress field.
3. Use the resultant of the stress field in each section to sketch a smoothly curved load path.
4. If the load path is not able to characterize the shape of the stress field, the load path is over-simplistic. Then the stress field has to be divided into more parts.
5. Identify the transverse forces and their locations that is required to change the direction of the load path in the D-region.

The simply supported beam studied in this thesis is considered as a deep beam where the whole beam is seen as a D-region. In such model, it is not possible to sketch a single smoothly curved load path, from the load to the support, due to the angle limitations stated in *Eurocode 2* (2008). As step 4 in the list above suggest, the stress field has to be divided into more parts. This is done by using truss models (see section 2.2.4).

#### 2.2.1.2 Rules for the load path method

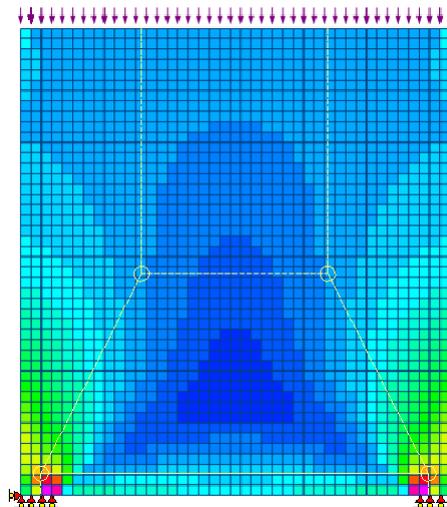
There are rules and limitations to follow when using the load path method. Engström (2011) describes them as:

- In each section, the load path should represent the resultant of the stress field which is simulated.
- Load paths can not cross each other.
- The load path should start in the same direction as the applied load or support reaction at the boundaries of the discontinuity region.
- In the vicinity of a concentrated force, the load path should have a sharp bend.

- The load path should have a soft bend where it needs to change direction away from a concentrated force.

### 2.2.2 ST models developed from FE analyses

By creating a FE model of the structure and performing linear FEA, principal stress trajectories can be obtained to give an understanding of how the forces flow within the structure (Engström 2011). The trajectories can then be translated into struts and ties to establish a ST model. By translating the stress flow in 2.3, a ST model can be developed.



**Figure 2.3:** Stress flow in a deep beam subjected to a distributed load acting on the top of the beam

### 2.2.3 ST method design according to Eurocode

In *Eurocode 2* (2008), guidelines is presented to how the ST method should be applied and calculated. Additional guidelines are presented in Engström (2011) that governs the recommendations of angles between the components of the ST model.

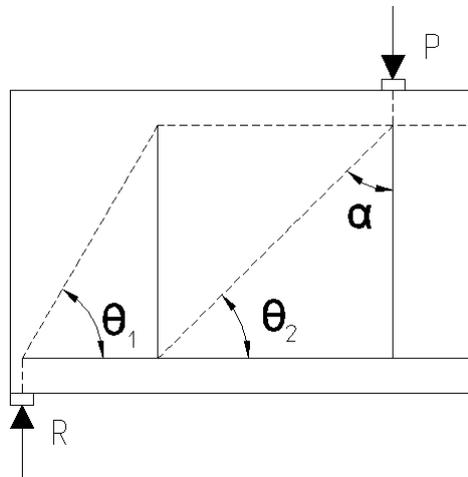
#### 2.2.3.1 Angles between struts and ties

The angles between struts and ties should be taken into account during the design of the ST model (Mathern & Chantelot 2010). There are two reasons for this:

- Strain compatibility problems and a high need for plastic redistribution can arise if the deviation angle at concentrated forces is inappropriate.
- Strain compatibility problems can also arise if the angle between struts and ties is too small.

The recommendations are summarized below, along with Figure 2.4, and are taken from Engström (2011), who are citing Schäfer (1999) in fib, Bulletin 2. In Figure

2.4, and throughout this report, the struts and ties are represented by dashed lines and solid lines respectively.



**Figure 2.4:** Angles to be taken into account in strut and tie design

- The angle  $\theta_1$ , for a strut meeting a single tie, should not be smaller than  $45^\circ$ . An angle of at least  $60^\circ$  is preferable.
- The angle  $\theta_2$ , for a strut between two perpendicular ties, should not be smaller than  $30^\circ$ . An angle of at least  $45^\circ$  is preferable.
- The deviation angle  $\alpha$ , for stresses under concentrated forces, should not exceed  $45^\circ$ . A reasonable choice is  $\alpha = 30^\circ$ .

In addition to the rules regarding angle limitation, there is a rule saying that concentrated forces in D-regions should not be carried by concentrated struts across wide elements (Engström 2011).

### 2.2.3.2 Struts

The multi axial state of stress, that a compressive strut experiences, influences the stress carrying capacity of the strut (Hendy C. R. & Smith D. A. 2007). A transverse compressive stress will affect the capacity in a positive manner while a transverse tensile stress will affect the capacity negatively. *Eurocode 2* (2008) gives two simplified and conservative limits for the maximal compressive stress,  $\sigma_{Rd,max}$ , that a strut can carry.

**In case of no transverse stress or transverse compressive stress**

$$\sigma_{Rd,max} = f_{cd} \quad (2.1)$$

Here,  $f_{cd} = \alpha_{cc} \frac{f_{ek}}{\gamma_c}$  with  $\alpha_{cc} = 0.85$  and  $\gamma_c = 1.5$ . This expression will rarely be possible to use since transverse tensile stress easily can occur.

**In case of transverse tensile stress (cracked concrete)**

$$\sigma_{Rd,max} = 0.6\nu' f_{cd} \quad (2.2)$$

Here,  $f_{cd} = \frac{f_{ck}}{\gamma_c}$  since it is recommended that the factor  $\alpha_{cc}$  has the same value as for shear design, i.e.  $\alpha_{cc} = 1$ .  $\nu'$  is a parameter that is nationally determined and is recommended to be  $\nu' = 1 - \frac{f_{ck}}{250}$  with  $f_{ck}$  in MPa.

For compressive struts affected by transverse tensile stress, different limits for different situations can be applied. These different situations can be quite difficult to distinguish. However, Equation 2.2 can be considered conservative for all those situations and was used in this thesis.

### 2.2.3.3 Ties

In a concrete structure, the forces in the tensile ties are carried by reinforcing steel. At the ULS, the reinforcement steel may be used up to its design yield strength (Hendy C. R. & Smith D. A. 2007). The center of gravity of the reinforcement bars should have the same position as the forces in the tensile ties. The required reinforcement cross-section area is given in Equation 2.3.

$$A_s = \frac{T}{f_{yd}} \quad (2.3)$$

Here,  $T$  is the tensile force in the tie and  $f_{yd}$  is the design yield strength of the reinforcement steel.

The ties must also be adequately anchored at the nodes. For further reading regarding anchorage of ties, the reader is referred to Engström (2011).

### 2.2.3.4 Nodes

The intersections between struts and ties are known as nodes (Hendy C. R. & Smith D. A. 2007). A node can be seen as a volume of concrete and it is the geometry of connecting struts, ties and external forces that determines the dimensions of the node.

There are two types of nodes (Engström 2011). One of those is the distributed node, which appears where distributed stress fields meet. They are never the critical nodes, so there is no need to check those. The critical nodes that need to be checked are the concentrated nodes, which appear where concentrated forces act.

The concentrated nodes should be designed with regard to the applied stresses at the edges of the nodes (Engström 2011). There are several aspects that affect these applied stresses, e.g. dimensions of loading plates and bearings. The concentrated nodes are divided into three different types, CCC-node, CCT-node and CTT-node, where C stands for compression strut and T for tensile tie. The maximum allowable stress,  $\sigma_{Rd,max}$ , for each type of node is given below, according to Hendy C. R. & Smith D. A. (2007).

**CCC-node**

$$\sigma_{Rd,max} = k_1 \nu' f_{cd} \quad (2.4)$$

This node describes a node where all intersecting members are struts. The nationally determined parameter  $k_1$  is recommended to be 1.0 and  $\nu'$  is defined in section 2.2.3.2. This type of node does not typically occur in a truss model of a simply supported beam with distributed load. It can occur at the compression faces of frame corners with closing moment and at internal beam supports.

**CCT-node**

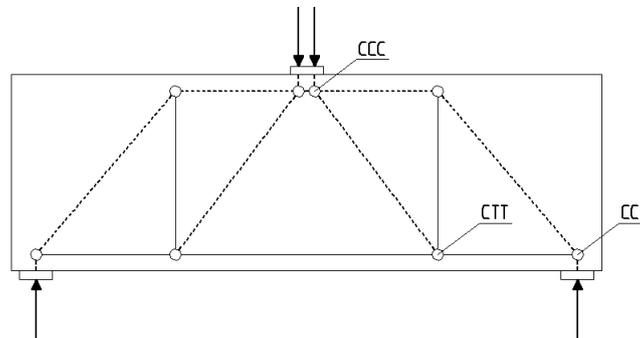
$$\sigma_{Rd,max} = k_2 \nu' f_{cd} \quad (2.5)$$

This node describes a node where one intersecting members is a tie and the others are struts. The nationally determined parameter  $k_2$  is recommended to be 0.85. This type of node can occur at e.g. end supports and in deep beams.

**CTT-node**

$$\sigma_{Rd,max} = k_3 \nu' f_{cd} \quad (2.6)$$

This node describes a node where two (or more) intersecting members are ties and the others are struts. The nationally determined parameter  $k_3$  is recommended to be 0.75.



**Figure 2.5:** Examples of node types, in a deep beam subjected to concentrated loads

## 2.2.4 Truss models for high beams

To apply the ST method, a truss model for the structure or D-region, must be developed. The truss model must fulfill the requirements on struts, ties, nodes and angles between struts and ties given in section 2.2.

In cases where the angles between the strut and the tie do not meet the requirements stated in section 2.2.3.1, a simple strut and tie model can not be used (Engström 2011). In such cases, a single load path can not be drawn from the load to the

support, so another solutions have to be used. One way is to use a regular type of the ST method, known as "truss model".

#### **2.2.4.1 Truss model with vertical ties**

The idea with such model is to transfer the shear force, from the applied load to the support, by series of inclined compressive struts and vertical tensile ties (Engström 2011). This enables distribution of stress fields so that the angle requirements can be reached. The vertical ties will then act and have the same function as the shear force reinforcement in a beam.

#### **2.2.4.2 Truss model with inclined ties**

For the construction worker, the handling of vertical stirrups are more convenient than the handling of inclined stirrups<sup>1</sup>. This is one reason to why the use of inclined stirrups is uncommon in design. However, inclined stirrups can be more effective when it comes to carrying tension and thus reduce the needed amount of reinforcing steel.

## **2.3 Finite element method**

The finite element method (FEM) can be used to solve the often complex differential equations that are encountered in engineering mechanics (Saabye Ottosen & Petersson 1992). In this section, the basics of the method is covered along with the implementation of the method in BRIGADE/Plus.

### **2.3.1 The FEM in general**

The FEM is a numerical approach that leads to approximative solutions to general differential equations. The differential equations, that describe a physical phenomena, are assumed to hold over a certain region (Saabye Ottosen & Petersson 1992). In the FEM, this certain region is divided into smaller sub regions (finite elements) for which the approximations are carried out in turn. This is a characteristic feature of the FEM which means that one approximation is not sought for the whole region. It is rather multiple approximations that are sought for the smaller sub regions.

As an example, let us assume that we have a problem where the variable varies in a non-linear manner over the entire region. Then, for the approximations over each finite element, it may be acceptable to assume that the variable varies in a linear manner (Saabye Ottosen & Petersson 1992). When the behaviour of each element has been determined, the elements can be patched together following specific rules. Eventually, the behaviour of the entire region can be approximated.

---

<sup>1</sup>Rikard Migell, Inhouse Tech AB

In section 2.4, a method known as structural optimization will be described. This method can for instance be used to find optimal solutions for structural problems and are based on the finite element method.

### 2.3.2 FEM implementation in BRIGADE/Plus

Over the past 20 years the rapid increase of computational power have made finite element (FE) simulations a powerful tool in structural engineering practice (Plevris & Tsiatas 2018). With the use of FE simulation programs such as e.g. BRIGADE/Plus, a wide range of different simulations can be performed. These simulations can e.g. consist of static analyses, performed to find the load in ULS to non-linear analyses, which in turn are used to investigate the development of cracks in reinforced concrete (*BRIGADE/Plus 6.2-5* 2019).

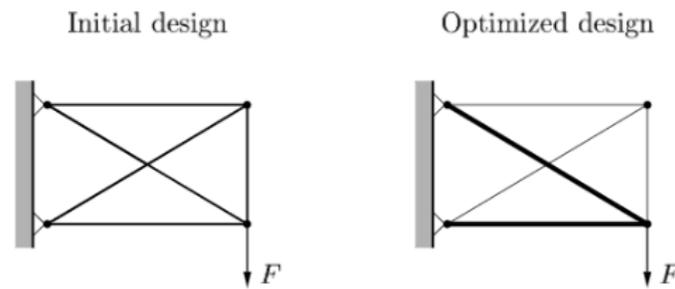
In this thesis, BRIGADE/Plus version 6.2-5 was used to perform Bi-directional evolutionary structural optimization (BESO). BRIGADE/Plus version 6.2-5 is an extension of Abaqus 6.14-2 and is customized to be used for structural engineering purposes. BRIGADE/Plus consist of a toolbox with predefined loads and load combinations adapted to a wide range of design codes such as e.g. Eurocode (*BRIGADE/Plus 6.2-5* 2019).

## 2.4 Structural optimization

For a given structural problem, structural optimization can be applied to find an optimal solution to a given structural problem (Christiansen & Klarbring 2009). An optimal solution can be defined by different types of optimums. One can be to make a structure as light as possible, another to make a structure as stiff as possible or to increase a structure's resistance to buckling. In order to get well defined optimums, different types of constraints need to be applied to the given optimization problem. Commonly used constraints in structural optimization are stresses, displacements and volumes. The measure of structural performance can be defined as weight, stiffness, critical load, stress, displacements and geometry. These measures can then be used to define the objective function which is to be optimized towards its minimum or maximum.

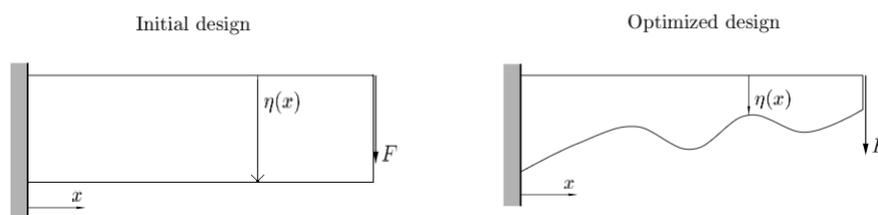
Structural optimization is commonly divided into three methods: size/volume-, shape- and topology optimization. According to Christiansen & Klarbring (2009) the methods are based on:

- *Size/volume optimization*: The size or volume of members is changed to find the most effective structure for a predefined given problem, this is visualized in Figure 2.6.



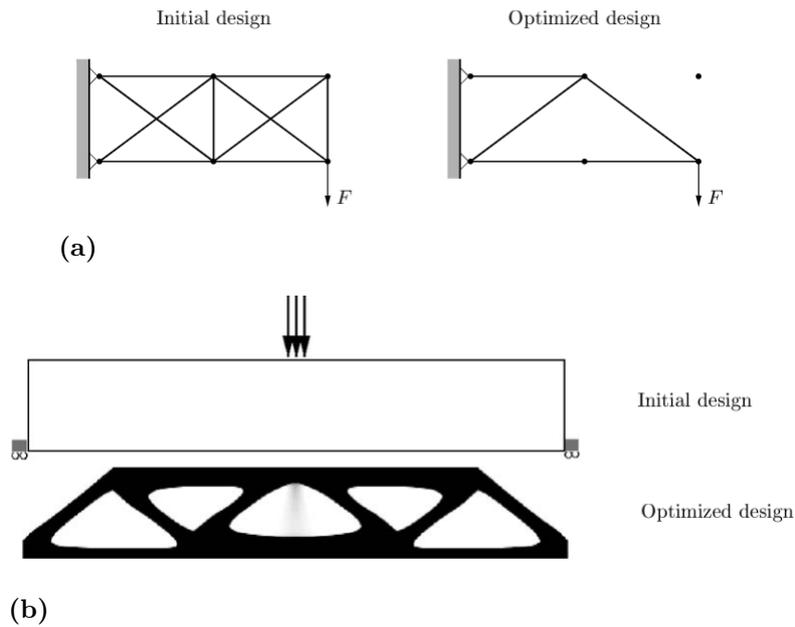
**Figure 2.6:** Size optimization for a truss system, adapted from (Christiansen & Klarbring 2009). Reprinted with permission.

- *Shape optimization:* A design domain is described as sets of differential equations. By integrating over the design domain a structure can be defined. The optimization can accordingly be defined by choosing the design domain for the differential equations as effective as possible. No change of connectivity and no new boundaries can be created. An example of shape optimization is presented in Figure 2.7.



**Figure 2.7:** Shape optimization of a structure defined by the function  $\eta(x)$ , adapted from (Christiansen & Klarbring 2009). Reprinted with permission.

- *Topology optimization:* The topology of the structure is allowed to change. This is done by removing or adding elements within the design domain of the studied structure. This is made by allowing element areas or volume to be zero or remaining fixed to a predefined size. Two types of topology optimization is visualized in Figure 2.8.



**Figure 2.8:** Two types of topology optimization, adapted from (Christiansen & Klarbring 2009). (a) Topology optimization of a truss system. (b) Topology optimization by removing material from the structure. Reprinted with permission.

### 2.4.1 Mathematical form of structural optimization

In Christiansen & Klarbring (2009) it is stated that for all structural optimization problems there is an *objective function*  $f$  that is used to classify the design and return a number to indicate progression of the optimization.  $f$  is usually chosen so that a smaller number indicates a better optimization. Some examples of what  $f$  can be measures of are effective stress, weight or displacements.

The *state variable*  $x$  describes a function or a vector that can be changed during optimization. In the common case  $x$  describes geometry or the choice of material. The *design variable*  $y$  is the output for a given  $x$  and describes the response of a structure in form of displacement, stress, strain or force (Christiansen & Klarbring 2009).

By combining the objective function, state variable and design variable, the general form of structural optimization (SO) can be stated as:

$$\text{SO} \begin{cases} \text{minimize} & f(x, y) \text{ with respect to } x \text{ and } y \\ \text{subject to} & \begin{cases} \text{behavioral constraints on } y \\ \text{design constraints on } x \\ \text{equilibrium constraint} \end{cases} \end{cases}$$

For a case with more than one objective function, so-called *multiple criteria-* or *vector-*optimization problems, the minimization problem is stated as followed:

$$\text{minimize}(f_1(x, y), f_2(x, y), \dots, f_l(x, y)) \quad (2.7)$$

where  $l$  corresponds to the number of objective functions subjected to the same constraints as described for **SO**.

Since general structural optimization is not made for the same set of  $x$  and  $y$ , there is no common case of solving the optimization for all  $f_i$ . To deal with these type of problems *Pareto optimality* is sought for. *Pareto optimality* means that no other design satisfies the objectives better for  $x^*$  and  $y^*$  and no other  $x$  and  $y$  fulfill the constraints(Christiansen & Klarbring 2009):

$$f_i(x, y) \leq f_i(x^*, y^*) \text{ for all } i = 1, \dots, l \quad (2.8)$$

$$f_i(x, y) < f_i(x^*, y^*) \text{ for at least one } i \in i = 1, \dots, l \quad (2.9)$$

One common way of finding Pareto optimal point is by introducing a weight factor  $w_i \geq 0$  with  $i=1, \dots, l$  which satisfies  $\sum_{i=1}^l w_i = 1$  and yields the scalar objective function:

$$\sum_{i=1}^l w_i * f_i(x, y) \quad (2.10)$$

The minimizing of Equation 2.10 under the constraints acting on **SO** can now be solved as a standard optimization problem with a Pareto optimal solution. It should be noted that not all Pareto optimal points can be obtained by using Equation 2.10.

The objective function  $\mathbf{f}$  is constrained in **SO** by three type of constraints, *behavioral-*, *design-* and *equilibrium constraints*(Christiansen & Klarbring 2009). Behavioral constraints are constraints subjected to the state variable  $y$ . These constraints can be expressed as  $g(y) \leq 0$  where  $g$  is a function describing e.g. displacements in a chosen direction. Design constraints governs the design variable  $x$  in a similar way as behavioral constraints. These constraints can be combined to give a common expression of  $g$ . Equilibrium constraints are based on the form:

$$\mathbf{K}(x)\mathbf{u} = \mathbf{F}(x) \quad (2.11)$$

where  $\mathbf{K}$  is the stiffness matrix,  $\mathbf{u}$  the displacement vector and  $\mathbf{F}$  is the force vector. Here  $\mathbf{u}$  takes the place of the state variable  $y$ . For continuum problems, the equilibrium constraints consists of partial differential equations (PDE). Treating the design variables,  $x$  and  $y$ , independently and provided that  $\mathbf{K}$  can be inverted 2.11 can be rewritten as,  $\mathbf{u}(x) = \mathbf{K}^{-1}(x)\mathbf{F}(x)$ . Thus, the equilibrium constraint can be left out and **SO** rewritten as (Christiansen & Klarbring 2009):

$$\mathbf{SO} \begin{cases} \min & f(x, \mathbf{u}(x)) \\ \text{subject to} & g(x, \mathbf{u}(x)) \leq 0 \end{cases} \quad (2.12)$$

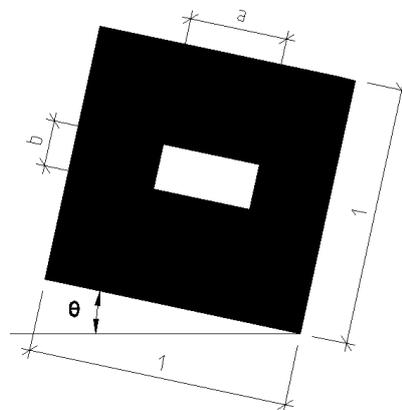
This is called the nested formulation and can be used to solve structural optimization problems numerically. This is performed by deriving  $f$  and  $g$  with respect to the design variable  $x$ .

### 2.4.2 Different types of structural optimization

In engineering practice today, the ST method is a well defined rational method described in national design codes such as e.g. Eurocode. Using the guidelines and rules described in chapter 2.2 it is up to each structural engineer's engineering judgment to choose a ST model that satisfies the problem at hand. In cases with complex geometry and/or loading conditions this becomes a trial and error procedure to find a model that satisfies the codes. This model is sufficient to carry the load but is not assured to be an optima. According to Shobeiri & Ahmadi-Nedushan (2017) topology optimization can be used as an alternative approach to find optimal models which can be translated into a ST model. In Qing Quan Liang et al. (2000) and Osvaldo M et al. (2017) several optimization procedures are presented.

The **Homogenization method in Topology Optimization** is described in Osvaldo M et al. (2017) as a method based on introducing micro voids within the elements of a given structure. These voids are then used as the design parameters in the optimization process. The optimization process is based on removing and adding material to expand or decrease the micro voids to create porous and solid regions. Therefore, the homogenization process can be seen as a type of Topology Optimization which uses the micro voids to find optimal material parameters.

When using the Homogenization method in FE analysis the micro void within the elements is defined by a width  $a$  and a height  $b$  with an orientation  $\theta$  (see Figure 2.9). The optimization process is then performed to find the maximal potential energy for the studied structure. This iterative process is visualized in Equation 2.13, where  $\Pi$  is the potential energy,  $N$  is the number of FE:s studied,  $\bar{V}_s$  is the upper volume limit,  $a_e$  is the width of the void,  $b_e$  is the height of the void,  $\theta_e$  is the orientation and  $v_e$  is the elemental volume (Osvaldo M et al. 2017).



**Figure 2.9:** Sketch describing the Homogenization method

$$\text{SO} \left\{ \begin{array}{l} \text{Maximize:} \quad \Pi(u) \\ \text{Subject to:} \quad \sum_{e=1}^N (1 - a_e b_e) v_e - \bar{V}_s = 0 \\ \text{and} \quad \begin{cases} a_e - 1 \leq 0 \\ -a_e \leq 0 \\ b_e - 1 \leq 0 \\ a_e, b_e, \theta_e = 1, 2, \dots, N \end{cases} \end{array} \right. \quad (2.13)$$

Developed from the theory of homogenization, is the method of **Solid Isotropic Material Penalization**, further referred to as the **SIMP**-method. This method is, according to Osvaldo M et al. (2017), the most renowned and most used in commercial software today. In this method an artificial element density  $\rho_e$  with a value within the range  $0 < \rho_{min} \leq \rho_e \leq 1$  is introduced. Multiplying the volume of the element,  $v_e$ , with the artificial element density produces the actual volume of the design domain as described in Equation 2.14.

$$V = \sum_{e=1}^N v_e \rho_e \quad (2.14)$$

A penalization factor,  $p$ , is introduced, which is applied to  $\rho_e$  when transforming the initial young's modulus  $E_0$  to artificial element young's modulus  $E_e$  as described in Equation 2.15.

$$E_e = \rho_e^p E_0 \quad (2.15)$$

The penalization factor  $p$  transforms the problem into a 0–1 problem. 0 is removed while 1 is kept for further iterations and to obtain a true optimization design value of  $p > 3$ . The iterative SIMP process can thus be formulated as(Osvaldo M et al. 2017):

$$\text{SO} \left\{ \begin{array}{l} \text{Maximize:} \quad c(p_e) \mathbf{F}^T u \\ \text{Subject to:} \quad \sum_{e=1}^N \rho_e^p u = \mathbf{F} \\ \text{and} \quad \begin{cases} \sum_{e=1}^N v_e \rho_e \leq \bar{V}_s \\ 0 < \rho_{min} \leq \rho_e \leq 1 \\ e = 1, 2, \dots, N \\ p = 1, 2, \dots, p_{max} \\ p_{max} > 3 \end{cases} \end{array} \right. \quad (2.16)$$

### 2.4.3 Bi-directional evolutionary structural optimization

The topology optimization routine that will be used in this thesis is the bi-directional evolutionary structural optimization method, further referred to as the BESO-

method (Shobeiri & Ahmadi-Nedushan 2017). The BESO-method is a direct development of the evolutionary structural optimization method (ESO-method).

The ESO-method is based on removing ineffective materials from a structure in order to find an optimal design (Shobeiri & Ahmadi-Nedushan 2017). In FE-design it can be translated to removing ineffective meshed elements while retaining the effective elements. Since ESO removes elements based on initial criteria, the later stages of the optimization can be affected by removing elements in early stages which can effect the final solution in a negative way. Through ESO, the improved

BESO-method has been developed (Shobeiri & Ahmadi-Nedushan 2017). BESO is based on the same theory of removing ineffective cells but also adding elements next to areas with high stresses. The BESO-method is more effective in terms of computer efficiency in the iteration process and robustness of the final model than the ESO-method.

### 2.4.3.1 Optimization algorithm

By applying the BESO-method to the general optimization problem, the **SO** can, according to Zuo & Xie (2015), be stated as described in Equation 2.17.

$$\mathbf{SO} \begin{cases} \text{minimize} & C(\mathbf{X}) = \mathbf{F}^T \mathbf{U} = \mathbf{U}^T \mathbf{K} \mathbf{U} \\ \text{subject to} & \begin{cases} \mathbf{X} = \{x_e\}, x_e = 1 \text{ or } x_{min} \quad \forall e = 1, \dots, N \\ \mathbf{F} = \mathbf{K} \mathbf{U} \\ V(\mathbf{X}) = \sum_X x_e v_e = V^* \end{cases} \end{cases} \quad (2.17)$$

The BESO-method is a gradient based method where the design variable, also known as the element sensitivity,  $\alpha_e$  is updated by differentiating the objective function  $C$  with respect to the design variable  $x_e$ , described by Equation 2.18 (Zuo & Xie 2015).

$$\alpha_e = \frac{\partial C}{\partial x_e} \quad (2.18)$$

The element sensitivity can be adapted from the SIMP-method using the material model defined in Equation 2.19.

$$\alpha_e = -p x_e^{p-1} \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e = \frac{-p}{x_e} x_e^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e = -p \frac{E_e}{x_e} \quad (2.19)$$

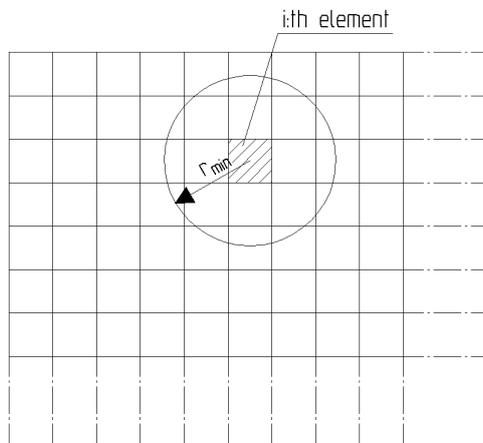
Where  $p$  is the penalty factor, that according to Equation 2.16, should be larger than 3.  $\mathbf{u}_e^T$  is the element displacement vector and  $\mathbf{k}_0$  corresponds to the solid element stiffness matrix with ( $x_e = 1$ ).  $x^p \mathbf{u}_e^T \mathbf{k}_0 \mathbf{u}_e$  corresponds to the element strain energy  $E_e$  which can be obtained using finite element analysis (FEA).

In order to obtain a mesh independent solution, the element sensitivity is updated according to Equation 2.20 with weight function  $w(r_{ej})$  according to Equation 2.21:

$$\hat{\alpha} = \frac{\sum_j w(r_{ej})\alpha_j}{\sum_j w(r_{ej})} = \sum_j \frac{w(r_{ej})}{\sum_j w(r_{ej})}\alpha_j = \sum_j \eta_j \alpha_j \quad (2.20)$$

$$w(r_{ej}) = \max(0, r_{min} - r_{ej}) \quad (2.21)$$

where the filter radius,  $r_{min}$ , is the smallest radius from the center of the  $i$ :th element to the surrounding elements (see Figure 2.10). Using the filter radius, a filter map can be constructed which is used to calculate the sensitivities, i.e. the influence of the surrounding elements on the  $i$ :th element.



**Figure 2.10:** Description of the input  $r_{min}$  that is used to create the filter map

Convergence is reached by averaging the sensitivities calculated in Equation 2.20. This is done by averaging the current iteration with the previous iteration according to Equation 2.22.

$$\tilde{\alpha} = \frac{\hat{\alpha}_e^k + \hat{\alpha}_e^{k-1}}{2} \quad (2.22)$$

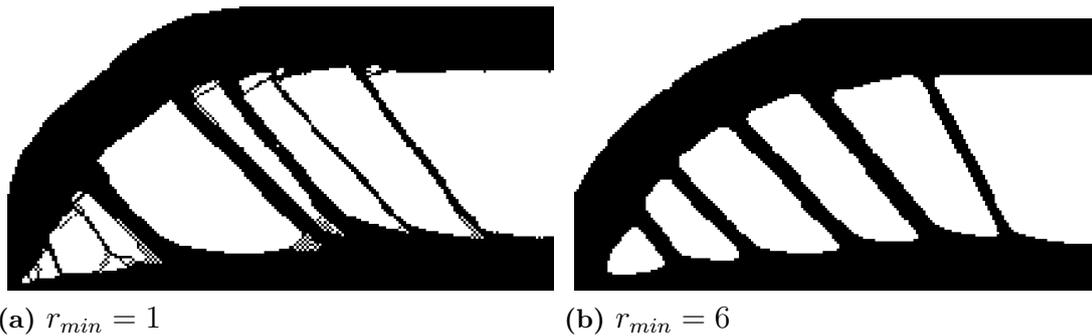
Starting with a full design the target volume for the next iteration is calculated using the evolutionary ratio, ER, as described in Equation 2.23 (Zuo & Xie 2015). By using the target volume, a threshold sensitivity  $\alpha^{th}$  is computed. This threshold is used to determine which of the elements that should be assigned solid ( $\alpha_i > \alpha_{th}$ ) and void ( $\alpha_i \leq \alpha_{th}$ ) element properties. This corresponds to assigning elements with element sensitivities larger than the threshold with the design variable  $x_{max} = 1$  and accordingly smaller with  $x_{min} = 0.001$ . This value  $x$  for the elements then governs the assignment of solid or void properties to the elements. Two types of approaches can be adopted hard- or softkill BESO. In hardkill BESO the stiffness of the void elements are set to zero while in softkill BESO the stiffness of the element are assigned a low stiffness in order to obtain consistency in the optimization.

$$V^{k+1} = V^k(1 \pm ER) \quad (2.23)$$

### 2.4.3.2 Mesh dependency and convergence

The BESO-method is mesh-dependent and this can cause problems for the final result (Xia et al. 2016). A more efficient topology design can always be obtained as long as new holes are introduced in the design. This effect can be seen as a numerical instability and there is a relation between a finer mesh and an increasing number of holes. However, mesh-independent solutions can be obtained using BESO with perimeter control, see Equation 2.20 and 2.21 in section 2.4.3.1.

For a new design problem, it is not a trivial task to predict the value for the perimeter constraint (Xia et al. 2016). The effect of using a too small value on the filter radius,  $r_{min}$  can be seen in Figure 2.11. The filter radius for the results shown in 2.11a and 2.11b are set to 1 respectively 6 times the element length, and is the only input that differs in the two analyses. Figure 2.11a clearly shows a chess pattern that is undesired for the final design. According to Huang & Xie (2010) the filter radius is commonly chosen as 2-3 times the size of the largest element.



**Figure 2.11:** Comparison between two analyses with different values on the filter radius. The two figures are extracted from FE analyses performed in MATLAB during this master thesis.

Even though the filter scheme, described with 2.20 and 2.21, is used, there can be problems with convergence for the topology and the objective function (Xia et al. 2016).

## 2.5 Python

Python is a popular and commonly used programming language. It was created and released by Guido van Rossum in 1991 and later developed by the Python Software Foundation. The main emphasis of the Python language is its readability and expressing its syntax in fewer lines of code (Saraswat n.d.).

## 2.6 MATLAB and CALFEM

MATLAB is a computer program and program language developed by MathWorks. Its primary area of use is to perform mathematical and technical calculations. The program language is widely known in engineering practice.

During MATLAB's existence, a vast number of toolboxes has been developed. One of those is CALFEM which is a toolbox containing finite element applications (Austrell et al. 2004).

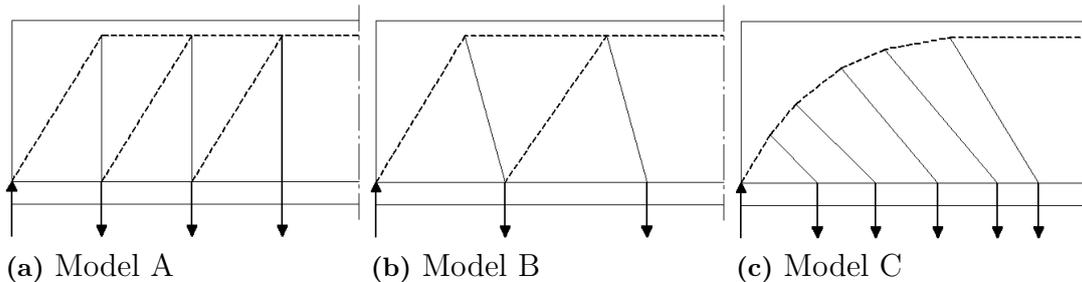
# 3

## Case study and applied methods

This chapter will present a case study of improving a crossbeam in an existing bridge designed by Inhouse Tech AB. Firstly, the crossbeam studied is introduced describing the dimensions, loads and other properties of the beam. Secondly, the two analytical methods, in developing ST models, are described. Thereafter, the implementation of structural optimization, using MATLAB and BRIGADE/Plus, to find effective ST models, are described. Finally, the translation from MATLAB and BRIGADE/Plus results to ST models are presented.

For the reader to be able to follow this report more easily, names have been assigned to the different ST models:

- **Model A** - The ST model obtained by using the ST method, as applied by Inhouse Tech AB.
- **Model B** - The ST model obtained by using a different ST approach, where the ties are placed with an inclination instead of vertical, as in Model A.
- **Model C** - The ST model are developed using structural optimization based on different FE analyses.



**Figure 3.1:** Names of the different ST models

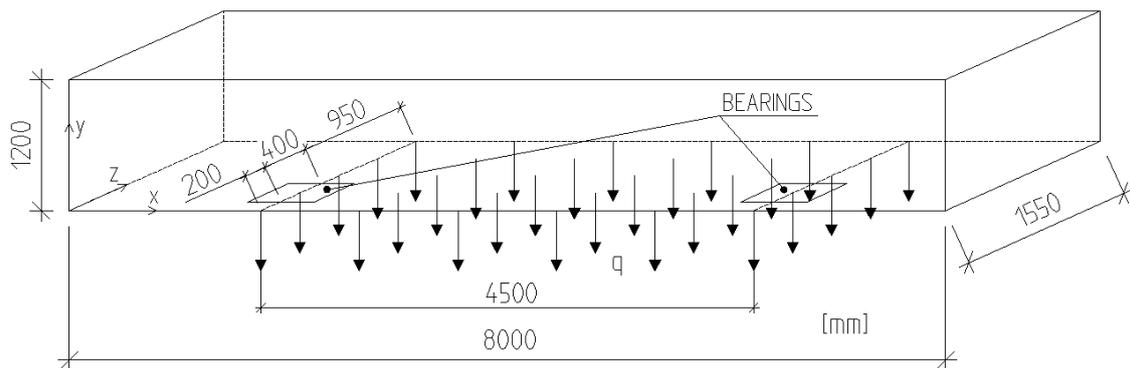
The development of Model C was carried out by optimizing finite element models. These FE models were created in MATLAB and BRIGADE/Plus. The sub-models studied in order to obtain Model C was named as followed:

- **Model C - 1:** 2D model created in MATLAB.
- **Model C - 2:** 2D model created in BRIGADE/Plus.
- **Model C - 3:** 3D model created in BRIGADE/Plus.

Model C - 1 is described in section 3.3 while Model C - 2 and Model C - 3 are described in section 3.4.

### 3.1 Description of the studied beam

For the case study, a crossbeam from an existing Swedish road bridge was chosen. The bridge was originally designed by Inhouse Tech AB and is located in the middle parts of Sweden, just north of the lake Vättern. The geometry and dimensions of the crossbeam can be seen in Figure 3.2. In this figure, the coordinate system that is used in this thesis, is also visualized. Note that only the span length of 4500 mm between the supports is included for the analyses of the developed ST models.



**Figure 3.2:** The geometry and dimensions of the beam studied in this thesis

There are several advantages to using this beam. Drawings and calculations from the existing bridge are available, revealing how it was designed and allowing for comparisons. It also provides the opportunity to have an open dialogue with the designer of the bridge. In this way, calculations and assumptions, made in this thesis, can be verified with experiences from today's engineering practice.

As can be seen in Figure 3.2, the beam studied is not a deep beam according to *Eurocode 2* (2008), since the span length is larger than three times the height. However, this structural element is part of a more comprehensive system and subjected to 3D flow of forces. Therefore, as a simplification, the crossbeam is seen as a deep beam with discontinuity regions where the ST method is applied.

For all calculations performed in this thesis, the distributed load  $q$  in Figure 3.2 is assumed to act at the bottom of the beam. In the real case, the major part of the load is transferred to the crossbeam from the main girder, connected to the side of the crossbeam. The load application is therefore distributed over its height. As a simplification, the assumption of load application to the bottom is made, to take the need for lifting reinforcement into account. In the ST models, the load is further simplified as point loads, acting at the bottom of the crossbeam.

Both the concrete and steel class, together with the design loads, are obtained from the provided documents describing the calculations of the crossbeam. C35/45 is used for the concrete and K500C-T is used for the steel. The design load are set to 6670 kN and are based on the total reaction for the bearings. This is the total force

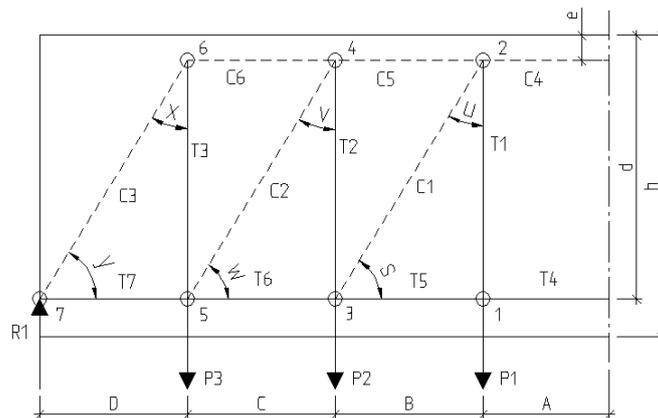
that should be lead through the beam down to the bearings. The dimensions of the bearings are  $400 \times 600 \text{ mm}^2$ .

## 3.2 Analytical development of Model A and B

The development of these two models was done in an analytical way, i.e. structural optimization was not implemented on these models. The two models were developed and then hand calculations were performed to calculate the forces in the struts and ties. CALFEM was then used to verify the hand calculations. The two following subsections describes the development of Model A and B and presents the ST models on which further calculations has been performed.

### 3.2.1 Development of Model A

Model A is obtained by using the conventional ST method, following the guidelines and recommendations given by *Eurocode 2* (2008) and Engström (2011). Another project was found in IHT's database, were a similar crossbeam was designed using conventional ST method. The calculations from that project were used as a base for comparison with the calculation performed for Model A. Model A is visualized in Figure 3.3 and the associated angles, point loads, reaction force and distances are presented in Table 3.1. The labeling of struts, ties and nodes are also presented in the figure. With these inputs, forces in the struts and ties were calculated and the capacities of the nodes were checked according to the procedure described in Section 2.2. These calculations and checks can be found in Appendix A.1.



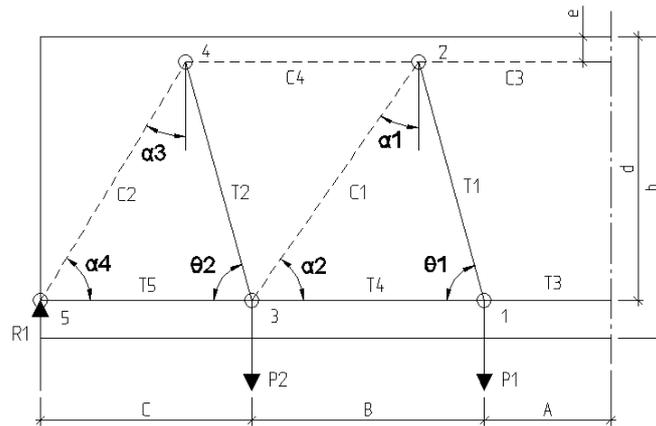
**Figure 3.3:** Picture of the developed Model A, on which further calculations were performed

**Table 3.1:** Distances, angles and forces associated with Model A

Distance	[mm]	Angle	[°]	Force	[kN]
$A$	498	$y$	58	$R_1$	3 335
$B$	584	$x$	32	$P_1$	1 171
$C$	584	$w$	58	$P_2$	886
$D$	584	$v$	32	$P_3$	1 298
$d$	1 050	$s$	58		
$e$	100	$u$	32		
$h$	1 200				

### 3.2.2 Development of Model B

From literature studies, knowledge about inclined ties and their potential benefits was obtained. Mathern & Chantelot (2010) describes an example where three vertical ties were replaced with two inclined ties and the same type of procedure were implemented on Model A in this thesis. Thus, Model B was developed with Model A as a base. The external force  $P_1$  was placed with the same distance from the symmetry line as in Model A, and then angles were chosen so that the number of vertical ties was reduced to two inclined ties and so that the guidelines, given by *Eurocode 2* (2008), were followed. Model B is visualized in Figure 3.4 and the associated angles, point loads, reaction force and distances are presented in Table 3.2. The labeling of struts, ties and nodes are also presented in the figure. With these inputs, forces in the struts and ties were calculated and the capacities of the nodes were checked according to the procedure described in Section 2.2. These calculations and checks can be found in Appendix A.2.

**Figure 3.4:** Picture of the developed Model B, on which further calculations were performed

**Table 3.2:** Distances, angles and forces associated with Model B

Distance	[mm]	Angle	[°]	Force	[kN]
$A$	498	$\theta_1$	75	$R_1$	3 335
$B$	919	$\theta_2$	75	$P_1$	1 916
$C$	833	$\alpha_1$	35	$P_2$	1 419
$d$	1 050	$\alpha_2$	55		
$e$	100	$\alpha_3$	31		
$h$	1 200	$\alpha_4$	59		

### 3.2.3 Verification of Model A using CALFEM

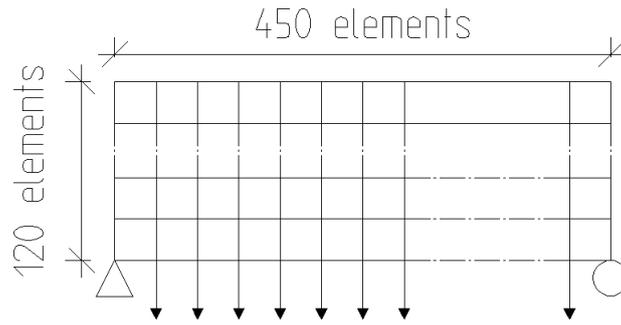
The hand calculations performed on the analytically developed Model A was verified by modelling the same problems in MATLAB. The toolbox CAFLEM was used which made it possible to model the problem as a truss system in a simple way. The FE analysis was implemented by using the CALFEM scripts *bar2e.m*, *assem.m* and *solvec.m*.

## 3.3 Development of Model C - 1 using structural optimization in MATLAB

The development of sub-model Model C - 1 was carried out by using structural optimization in MATLAB. The resulting optimized FE model was then translated into a ST model. Here, the model setup for the FE analysis is described, followed by the implementation of BESO in MATLAB.

### 3.3.1 Model C - 1 setup

The structural optimization routine in MATLAB provided a less complex code and procedure than the one for BRIGADE/Plus. The MATLAB script (m-script) uses square plane stress elements and does only perform 2D-simulation. The mesh was created so that it reflected the dimensions of the studied deep beam, i.e. 450x120 elements. Thus, each element was given an area of 1x1 cm<sup>2</sup>. To simulate distributed load on the bottom boundary, point loads was applied on all nodes on the bottom boundary, except the left- and rightmost node, see Figure 3.5. The properties of the mesh are presented in Table 3.3 and the inputs for the structural optimization is presented in 3.4. These inputs was chosen, based on a number of performed trial simulations.



**Figure 3.5:** Model setup for the MATLAB script, describing mesh, applied load and boundary conditions

**Table 3.3:** Inputs for the mesh for Model C - 1

Input	Value
Number of elements in x-direction [-]	450
Number of elements in y-direction [-]	120
Total number of elements [-]	54 000

**Table 3.4:** Inputs for the structural optimization for Model C - 1

Input	Value
Volume fraction, $V_f$ [-]	0.45
Filter radius, $r_{min}$ [cm]	6
Evolutionary ratio, $ER$ [-]	0.02

### 3.3.2 BESO implementation using MATLAB

The implementation of BESO in MATLAB was made to more easily investigate how the different inputs (volume fraction, evolutionary rate and filter radius) effects the resulting model and to create sub-model Model C - 1, one of the sub-models on which the final Model C is based. From literature studies, a well written and explanatory m-script, written by Xia et al. (2016), was found. This m-script was used in this thesis, with some modifications that will be described here. Even though hard-kill BESO are implemented, the elements are not truly deleted. Instead, the elements are assigned an very low Young's modulus, similar to the case of soft-kill used in the python code.

In this work, the script was modified so that the external load was applied as a distributed load on the bottom boundary of the beam and so that it produced the graphs of the volume fraction and the objective function. The graphs were desired to easily study the convergence of the analysis. Several lines were added to the script, which can be seen in Figure 3.6 and Figure 3.7.

### 3. Case study and applied methods

---

```
20 %% Nodes on the bottom boundary*
21 - B=[(nely+1)*2:(nely+1)*2:(nely+1)*2*(nelx+1)];
22
23 %% Distributed load*
24 - B(nelx+1)=[]; % Remove support boundary
25 - B(1)=[]; % Remove support boundary
26 - i=B;
27 - j=ones(1,length(B));
28 - v=j*-1;
29
30 %% DEFINE LOADS AND SUPPORTS
31 - switch(ctp)
32 - case 1 % HALF-MBB BEAM
33 - F = sparse(2,1,-1,2*(nely+1)*(nelx+1),1);
34 - fixeddofs = union(1:2*2*(nely+1), 2*(nely+1)*(nelx+1));
35 - case 2 % CANTILEVER
36 - F = sparse(2*(nely+1)*(nelx+1)-nely,1,-1,2*(nely+1)*(nelx+1),1);
37 - fixeddofs = (1:2*(nely+1));
38 - case 3 % HALF-WHEEL
39 - F = sparse(i,j,v,4*(nely+1)*(nelx+1),1); % NEW*
40 - fixeddofs = union(2*nely+1:2*(nely+1), 2*(nely+1)*(nelx+1));
41 - end
```

**Figure 3.6:** Print screen of the used m-script. Line 20-28 was added to the script and line 39 was modified, to apply the load as a distributed load on the bottom boundary of the beam.

```
100 %% Plot volfrac and objective function development*
101 - figure(2)
102 - plot(It_all,vol_all)
103 - xlabel('Iteration')
104 - ylabel('Volume fraction')
105 - figure(3)
106 - plot(It_all,c)
107 - xlabel('Iteration')
108 - ylabel('Objective function')
```

**Figure 3.7:** Print screen of the used m-script. Line 100-108 was added to plot the graphs for the volume fraction and the objective function.

## 3.4 Development of Model C - 2 and Model C - 3 using structural optimization in BRIGADE/Plus

The development of sub-models Model C - 2 and Model C - 3 was carried out by using structural optimization in BRIGADE/Plus. The resulting optimized FE models was then translated into ST models. Here, the model setups for the FE analyses is described, followed by the implementation of BESO in BRIGADE/Plus. In order to perform BESO in BRIGADE/Plus a Python script created by Zuo & Xie (2015) was used.

### 3.4.1 Model C - 2 setup

This subsection describes the different settings in the different modules in BRIGADE/Plus and the inputs for the BESO routine for the setup for Model C - 2.

#### Part

The part was created defining the structural model to be a 2D plane stress model.

**Load**

- *Load*: The load was applied as a distributed load on the bottom edge of the model, by using pressure load in the load module.
- *Boundary conditions*: The boundary conditions was applied to the nodes in the bottom corners of the model. The beam was modelled as simply supported.

**Mesh**

The elements chosen for the 2D analysis was 8-node bi-quadratic plane stress quadrilateral with reduced integration (CPS8R). The mesh size was implemented according to Table 3.5.

**Table 3.5:** Inputs for the mesh for Model C - 2

Input	Value
Number of elements in x-direction [-]	450
Number of elements y-direction [-]	120
Total number of elements [-]	54000

**Inputs for the BESO algorithm**

The inputs  $V_f$ ,  $r_{min}$  and  $ER$ , described in Section 2.4, needed to be chosen for the optimization simulation. From trial simulations, it was found that the values presented in Table 3.6 gave satisfying results.

**Table 3.6:** Inputs for the structural optimization for Model C - 2

Input	Value
Volume fraction, $V_f$ [-]	0.35
Filter radius, $r_{min}$ [cm]	3
Evolutionary ratio, $ER$ [-]	0.02

**3.4.2 Model C - 3 setup**

This subsection describes the different settings in the different modules in BRIGADE/-Plus and the inputs for the BESO routine for the setup for Model C - 3.**Part**

For the 3D case three parts was created.

- *Part-1*: Solid extrusion defining the body to be optimized.
- *Part-2*: Rigid plate defining the supports.
- *Part-3*: Planar shell to distribute the load.

**Interaction**

Two types of constraints was defined for the interaction between *Part-1* and *Part-3* and between *Part-2* and *Part-3* respectively. For the interaction between *Part-1* and *Part-3* a tie constraint was defined, with *Part-1* defined as the master surface and *Part-2* defined as the slave surface. For *Part-2* and *Part-3*, a rigid body constraint

was defined. *Part-2* defines the rigid body and was selected as the body element. A reference point was placed in the center of gravity of the plate. *Part-3* was partitioned so that the surface to be tied in the rigid body constraint corresponded to the size of *Part-2*.

When defining the tie constraint the following guidelines was followed (Vishark n.d.):

- The larger of the surface should act as the master surface.
- The stiffer body should act as the master surface.
- The part with the coarser mesh should act as master surface.

When defining the rigid body constraint the following was used (*Defining rigid body constraints* 2017):

- Body elements will be defined by the elements of the rigid body.
- A tie constraint will tie the deformable part to the rigid body.
- A reference point needs to be defined, this point should preferably be placed in the center of gravity of the rigid part. This point will govern the rotation of the part.

#### Load

- *Load*: The load was defined as a pressure load action and applied on the part of *Part-3*, between the supports, as visualized in Figure 3.2.
- *Boundary condition*: The boundary conditions was defined in the reference point of the rigid plate. In order to arrange the beam to work as a simply supported beam the boundary condition is extended so that the beam only can rotate around the x-axis.

#### Mesh

For *Part-1*, 8 nodal brick elements was assigned as element type and under element properties, reduced integration and enhanced hourglass control was applied (C3D8R). The reduced integration reduces the computational time required to run the analysis and the enhanced hourglass control reduces the occurrence of hourglassing i.e. element distortion.

Choosing the mesh size was a somewhat heuristic process, but two guidelines was established:

- The mesh should not be too coarse. A too coarse mesh leads to the risk of single element rows, which easily rotates/distorts and affects the final solution. The mesh should also not be too fine since it leads to long computational time, as described in section 3.4.3.
- When the model contains different parts it is important that the slave surface has at least the same mesh size as the master surface. Thus, governs the size of *Part-1* the size of *Part-3's* mesh. Presented in Table 3.7 are the mesh size used in the optimization.

**Table 3.7:** Inputs for the mesh for Model C - 3

Input	Value
Element size in x-direction [-]	200
Element size in y-direction [-]	40
Element size in z-direction [-]	16
Total number of elements [-]	128000

**Inputs for the BESO algorithm**

The inputs  $V_f$ ,  $r_{min}$  and  $ER$ , described in Section 2.4, needed to be chosen for the optimization simulation. From trial simulations, it was found that the values presented in Table 3.8 gave satisfying results.

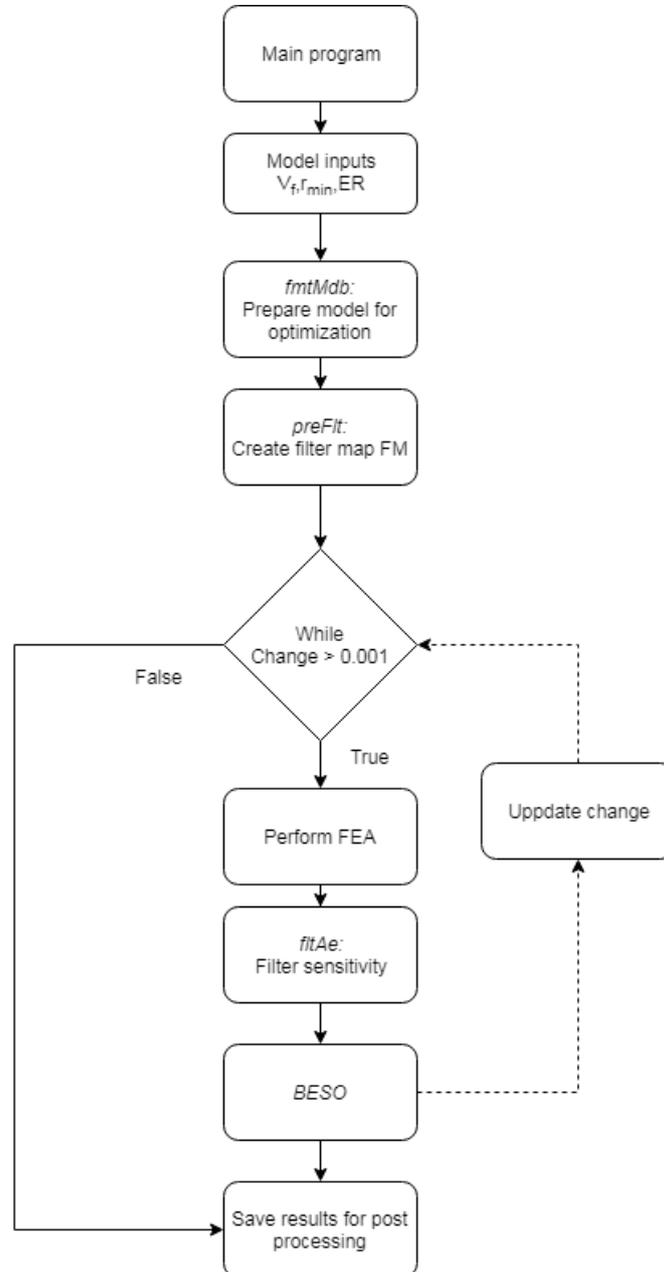
**Table 3.8:** Inputs for the structural optimization for Model C - 3

Input	Value
Volume fraction, $V_f$ [-]	0.25
Filter radius, $r_{min}$ [cm]	30
Evolutionary ratio, $ER$ [-]	0.02

**3.4.3 BESO implementation using Python**

In this thesis, a general python code for BESO-optimization created and presented in Zuo & Xie (2015), was used and modified. The code is implementing the BESO-algorithm presented in section 2.4.3 and subsection 2.4.3.1. The algorithm is divided into separate functions which then is called on by a main execution program. In this subsection these parts will be explained to give the reader a insight in how the BESO-algorithm can be implemented to perform topology optimization within a FEA-solver, in this case BRIGADE/Plus.

The workflow of the code is visualized in Figure 3.8. The outline of the main program, executed as a script in BRIGADE/Plus, is that it calls on separate functions to perform parts of the optimization. These different parts of the optimization is described below.



**Figure 3.8:** Flowchart visualizing the BESO algorithm

The function *fmtMdb* prepares the model for the optimization routine. A solid material is defined by creating an elastic material with the young's modulus  $E = 1$  and Poisson's ratio  $\nu = 0.3$ . The python script implements the soft-kill BESO method and for the parts with a "void" material the properties are assigned to  $E = 1e^{-9}$  and Poisson's ratio  $\nu = 0.3$ . Two types of outputs is requested, one field output (**ELEDEN**) corresponding the energy density equivalent to the element

strains used when calculating element sensitivity's. The second output is the history output (**ALLWK**) stored as the external work. The external work can be visualized as the compliance function  $C(x)$  which is the function to be minimized in Equation 2.17. In the code, this corresponds to the objective function of the optimization and governs the development of the optimization.

The function *preFlt* creates the filter map,  $FM$ , which assures the mesh independence of the optimization.  $FM$  contains the influence of the surrounding elements within the distance  $r_{min}$  of each element's center. This is computed by looping over all elements twice in order to obtain the influence of all elements.

Utilizing *fmtMdb* and *preFlt*, BESO is performed by creating a while loop governed by the function *change* corresponding to the convergence error of the objective function of the optimization. As visualized in Figure 3.8, three steps are then executed:

- *Perform FEA* - uses the model defined in *fmtMdb* to perform linear-elastic analysis to extract the elemental strain energy (**ESEDEN**), which then is used to calculate the element sensitivity  $\alpha_e$ .
- *fltAe: Filter sensitivity* - applies the filtering scheme, created in *preFlt*, by multiplying the raw sensitivities with the total weight factor  $\eta_j$  to obtain mesh-independent sensitivities as presented in Equation 2.20.
- *BESO* - By using the filtered sensitivities and target volume as inputs, the function *BESO* performs the main BESO routine. This routine is performed as described in section 2.4.3.1, by assigning the two variables *lo* and *hi*, corresponding to the largest and smallest filtered sensitivities. The target volume, *tv*, is calculated by multiplying the volume fraction with the initial volume of the solid elements described in 2.23 given as an input from the main program. Then, *while*  $\frac{(hi-lo)}{hi} > 1.0e - 5$ , a threshold *th* is calculated by averaging out the sensitivities, which is used to update the convergence error *change* that governs the optimization.

When the change criterion is false the while loop is terminated and the results is saved as `Final_design.cae`, which was used for post processing the results.

### 3.4.3.1 Saving the filter map

The filter mapping scheme, described in section 3.4.3 is created by looping over all the elements twice. This becomes a time consuming task to perform, as the number of loops increases quadratically with the number of elements. To reduce the computation time, the python library NumPy was deployed to decrease the need for arithmetic calculations.

The NumPy library is optimized for scientific computational calculations and treats all entries within a list to decreases the time preparing the filter map  $FM$  (Zuo & Xie 2015). Thus, in order to avoid performing the filtering scheme every time the code is executed, the code was modified to save the filter map as a **pickle**-file. Pickle is a python module that uses binary protocols for serializing and de-serializing

a Python object structure (Python Software Foundation 2019). A total of five lines were added to the code in order to save and implement the filter map which are visualized in Figure 3.9.

```
47 with open("test.py", "wb") as fp:
48     pickle.dump(Fm, fp)
```

(a) Two lines added on line 48-49 used to save the filter map

```
87 if rmin>0: #preFlt(rmin, elmts, nds, fm)
88     with open("test.py", "rb") as fp:
89         fm = pickle.load(fp)
```

(b) Three lines added on line 87-89 used to reload the filter map

**Figure 3.9:** Lines of code added to the Python script created by Zuo & Xie (2015) used to save the filter map, FM

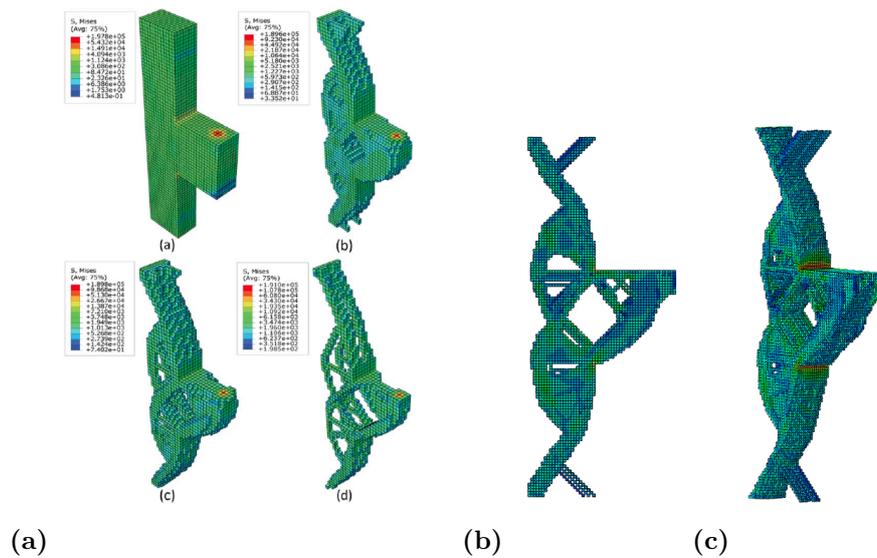
This made it possible to produce filter maps with different filter radius,  $r_{min}$ , once and reuse them with e.g. different elemental properties,  $V_f$  and  $ER$ .

#### 3.4.3.2 Producing viewpoints

A common engineering practice is to automatize the time consuming parts of extracting relevant results from a number of output databases simultaneously. By using the option of running scripts written as a Python code, pre-selected outputs can be extracted. Such outputs can for example be stresses, strains, deformations etc. The easiest way to create these type of scripts is to use the .rpy-file, which is generated automatically when working in BRIGADE/Plus and stores all changes made on the model as a python code. Thus, by performing the executions aimed to automatize, the part of the code describing this execution can be extracted from the .rpy-file and placed in a loop over all .odb-files (Obbink-Huizer 2018). This method was used to automate the extraction of pictures used in this thesis.

#### 3.4.3.3 Verification of the python code

Presented in Shobeiri & Ahmadi-Nedushan (2017), is several structures which BESO is performed on. In order to asses if the developed python script was valid, a corbel-column structure was modeled with the same dimensions and boundary conditions as the one presented by Shobeiri & Ahmadi-Nedushan (2017). The results from the optimizations was compared to validate the script. This is visualized in Figure 3.10, were it can be seen that the optimized results resembles each other. The model made by (Shobeiri & Ahmadi-Nedushan 2017) are more refined than the developed model but since clear similarities were shown, the python-script was assumed to be verified.



**Figure 3.10:** Optimization of a corbel structure made to verify the python code used. (a) Optimization made in Shobeiri & Ahmadi-Nedushan (2017). Reprinted with permission. (b) Optimization made to verify the python code, 2D visualization. (c) Optimization made to verify the python code, 3D visualization.

## 3.5 Development of Model C based on the sub-models

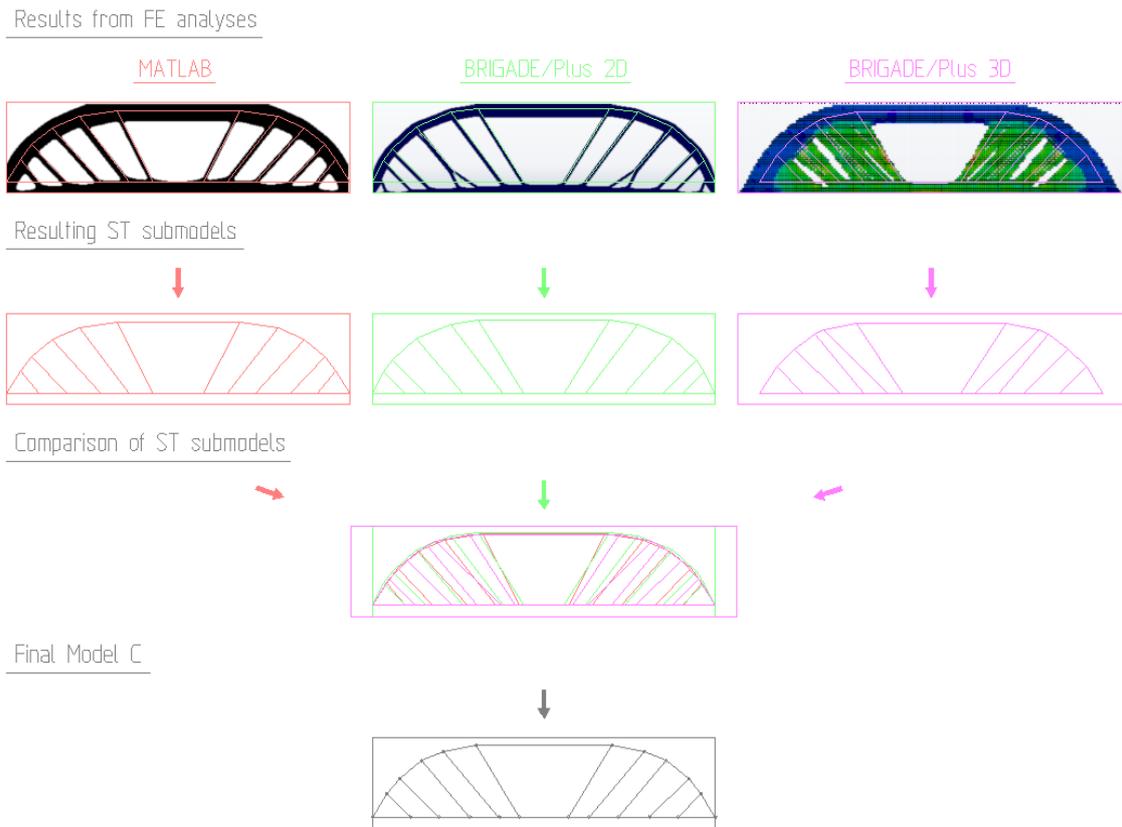
To be able to perform hand calculations on the obtained optimized sub-models, the sub-models had to be translated into ST models along with relevant angles and distances. This section describes how the three sub-models were combined into one final Model C.

### 3.5.1 Translation of the results from the sub-models

The results from the FE analyses were translated and interpreted using AutoCAD. Pictures of the optimized models was extracted from BRIGADE/Plus and MATLAB and then imported in AutoCAD. Then the lines were drawn after the existing struts and ties. The next step was to measure the relevant angles and distances needed for the hand calculations. Those quantities was easily measured in AutoCAD and eventually this procedure lead to an computable ST model.

As mentioned in the introduction to this chapter, Model C was developed based on the results of the three different sub-models from the FE analyses. The procedure is visualized in Figure 3.11. The ST models from the different FE analyses was combined and placed over each other, and then the final Model C was drawn after this combination.

### 3. Case study and applied methods



**Figure 3.11:** Picture describing how the sub-models were combined into the final Model C

#### 3.5.2 The resulting Model C

Model C is visualized in Figure 3.12 and the associated angles, forces and distances are presented in Table 3.9. The calculations performed for Model A can be found in Appendix A.3. Note that all the angles between the struts and ties are presented directly in Figure 3.12.

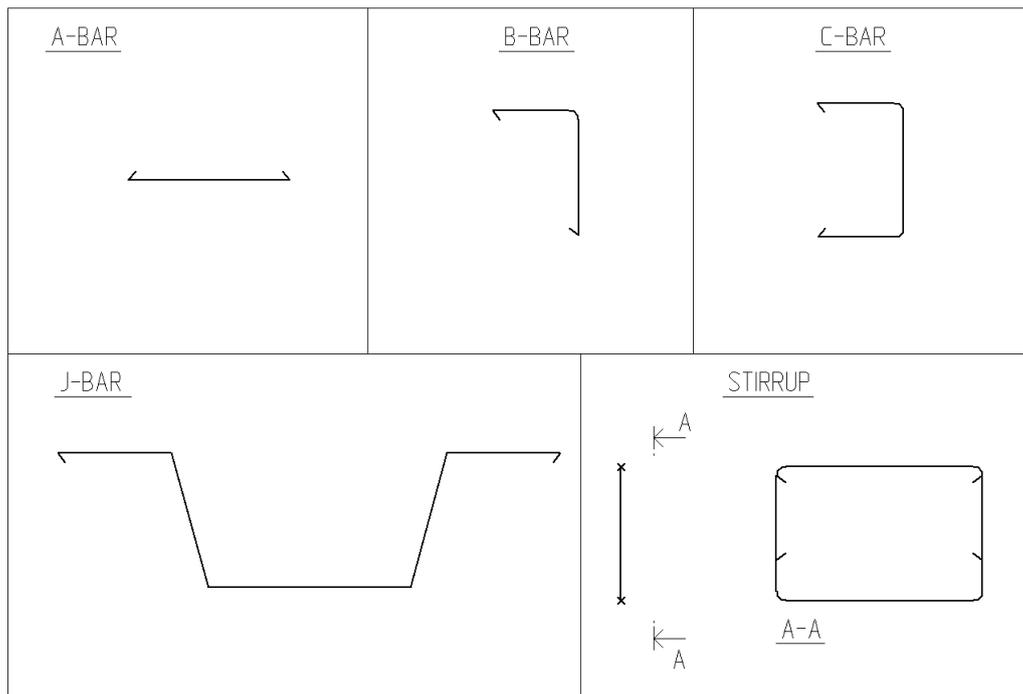
**Table 3.9:** Distances, angles and forces associated with Model C

Distance	[mm]	Angle	[°]	Force	[kN]
<i>A</i>	321	$\theta_1$	59	$R_1$	3 335
<i>B</i>	268	$\theta_2$	50	$P_1$	674
<i>C</i>	386	$\theta_3$	50	$P_2$	485
<i>D</i>	405	$\theta_4$	45	$P_3$	586
<i>E</i>	376	$\theta_5$	45	$P_4$	579
<i>F</i>	494			$P_5$	1 011
<i>d</i>	1 050				
<i>e</i>	104				
<i>h</i>	1 200				



### 3. Case study and applied methods

---



**Figure 3.13:** Shapes of the reinforcement bars used in this thesis

# 4

## Results

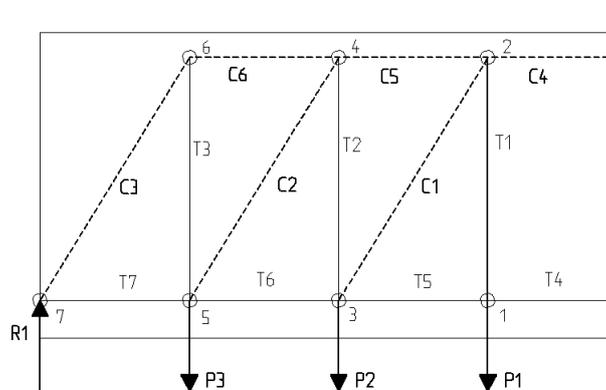
This chapter presents the results from the different analyses and ST models. Firstly, the results for Model A and B are presented, which both do not include any structural optimization. Secondly the results from the sub-models to Model C are presented and finally the results for Model C.

### 4.1 Results from Model A

In this section, the results from Model A are presented, in terms of forces in struts and ties, checks of critical nodes and the final reinforcement design.

#### 4.1.1 Forces in struts and ties from Model A

The locations and labeling of struts, ties, nodes and external and reaction forces are presented in Figure 4.1 and the associated magnitudes are presented in Table 4.1.



**Figure 4.1:** Locations of struts, ties, nodes and external and reaction forces for Model A

**Table 4.1:** Normal forces in struts and ties together with external and reaction forces for Model A

Strut	[kN]	Tie	[kN]	Point loads and reaction force	[kN]
$C_1$	1 375	$T_1$	1 171	$R_1$	3 335
$C_2$	2 391	$T_2$	2 037	$P_1$	1 171
$C_3$	3 915	$T_3$	3 335	$P_2$	866
$C_4$	4 022	$T_4$	4 022	$P_3$	1 298
$C_5$	3 302	$T_5$	4 022		
$C_6$	2 050	$T_6$	3 302		
		$T_7$	2 050		

#### 4.1.2 Verification of hand calculations for Model A using CALFEM

The results from the FE analysis in CALFEM are presented in Table 4.2. They correspond very well with the values in Table 4.1 and thus confirm the correctness of the hand calculations.

**Table 4.2:** Normal forces in struts and ties from the results of FE analysis in CALFEM for Model A

Strut	[kN]	Tie	[kN]
$C_1$	1 375	$T_1$	1 171
$C_2$	2 391	$T_2$	2 037
$C_3$	3 914	$T_3$	3 335
$C_4$	4 023	$T_4$	4 022
$C_5$	3 302	$T_5$	4 022
$C_6$	2 050	$T_6$	3 302
		$T_7$	2 050

#### 4.1.3 Check of critical nodes for Model A

Node 2 and 7 in Figure 4.1 are the critical nodes in Model A and are checked against the requirements given by *Eurocode 2* (2008). The results are presented in Table 4.3. The first index, in the node columns, stands for node number and the last one for which force that is used for the check.

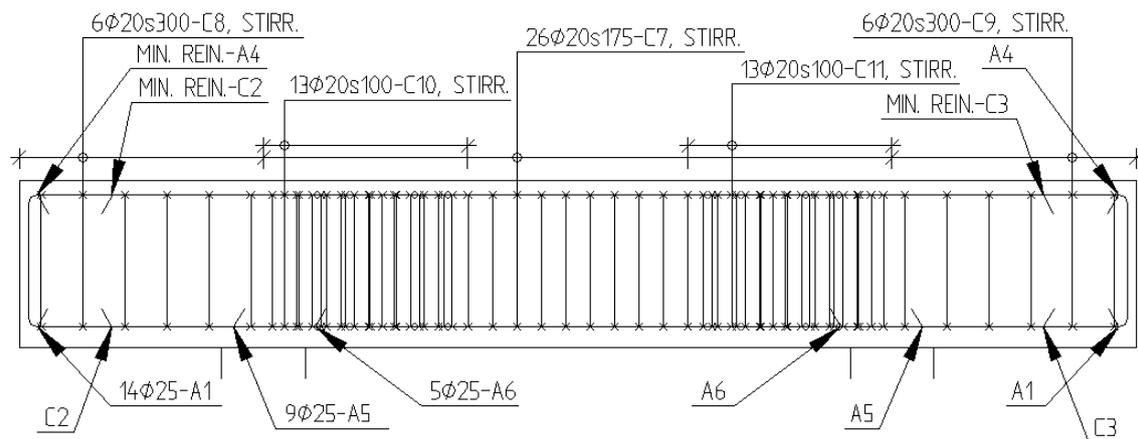
**Table 4.3:** Checks according to requirements from *Eurocode 2* (2008) for Model A

Maximal stress		Node 2		Node 7	
$\sigma_{Rd,max}$	17.1 MPa	$\sigma_{2,C_4}$	13.0 MPa	$\sigma_{7,R_1}$	13.9 MPa
				$\sigma_{7,C_3}$	13.1 MPa

#### 4.1.4 Reinforcement design for Model A

The reinforcement design for Model A is presented in Figure 4.2. The volume of the reinforcement, that corresponds to this design, and its ratio to the amount of concrete is presented in Table 4.4.

Common for all final reinforcement designs, Figure 4.2, 4.4 and 4.16, is that they all have the same amount of minimum reinforcement. The bars marked with "MIN. REIN." in the above stated figures, are thus not included in the calculations for the volumes of reinforcement.



**Figure 4.2:** Reinforcement design for Model A

**Table 4.4:** The volume of the reinforcement used for the reinforcement design of Model A

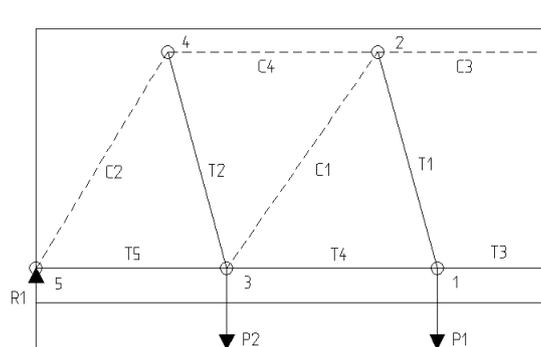
Stirrups [m <sup>3</sup> ]	0.087
Horizontal bars [m <sup>3</sup> ]	0.085
Total reinforcement [m <sup>3</sup> ]	0.172
Concrete [m <sup>3</sup> ]	14.880
Volume ratio [-]	0.012
Mass to volume ratio [kg/m <sup>3</sup> ]	90.161

## 4.2 Results from Model B

In this section, the results from Model B are presented, in terms of forces in struts and ties, checks of critical nodes and the final reinforcement design.

### 4.2.1 Forces in struts and ties for Model B

The locations and labeling of struts, ties, nodes and external and reaction forces are presented in Figure 4.3 and the associated magnitudes are presented in Table 4.5.



**Figure 4.3:** Locations of struts, ties, nodes and external and reaction forces for Model B

**Table 4.5:** Normal forces in struts and ties together with external and reaction forces for Model B

Strut	[kN]	Tie	[kN]	Point loads and reaction force	[kN]
$C_1$	2 339	$T_1$	1 983	$R_1$	3 335
$C_2$	3 891	$T_2$	3 453	$P_1$	1 916
$C_3$	4 752	$T_3$	4 752	$P_2$	1 419
$C_4$	2 897	$T_4$	4 239		
		$T_5$	2 004		

### 4.2.2 Check of critical nodes for Model B

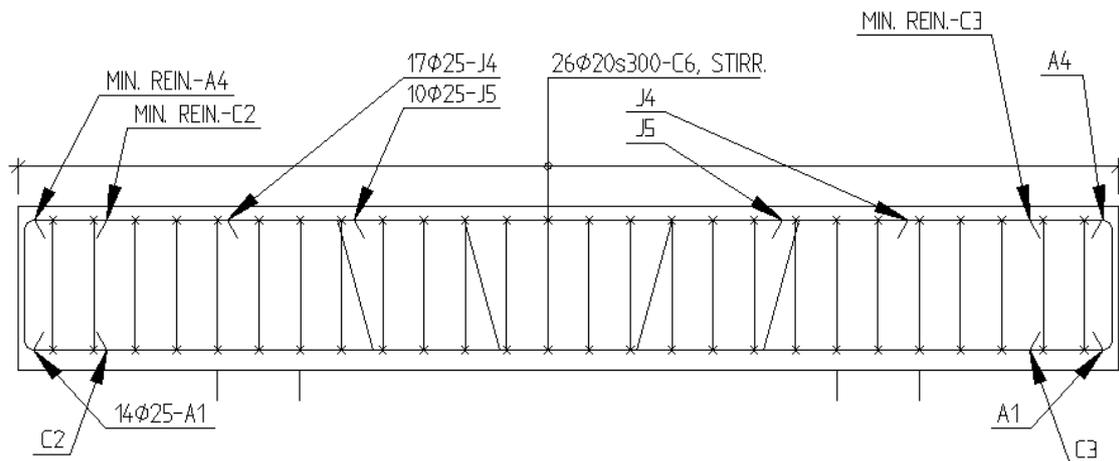
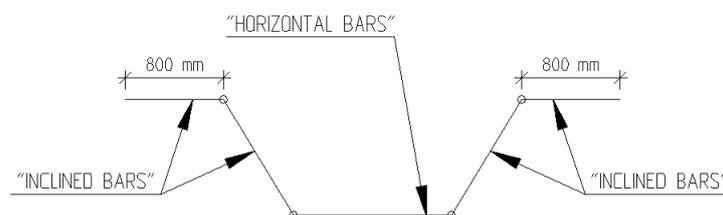
Node 2 and 5 in Figure 4.3 are the critical nodes in Model B and are checked against the requirements given by *Eurocode 2* (2008). The results are presented in Table 4.6. The first index, in the node columns, stands for node number and the last one for which force that is used for the check.

**Table 4.6:** Checks according to requirements from *Eurocode 2* (2008) for Model B

Maximal stress		Node 2		Node 5	
$\sigma_{Rd,max}$	17.1 MPa	$\sigma_{2,C_3}$	15.3 MPa	$\sigma_{7,R_1}$	13.9 MPa
				$\sigma_{7,C_2}$	13.0 MPa

### 4.2.3 Reinforcement design for Model B

The reinforcement design for Model B is presented in Figure 4.4. The volume of the reinforcement, that corresponds to this design, and its ratio to the amount of concrete is presented in Table 4.7. The volume from "Inclined bars", in Table 4.7, represents the volume for a part of the bars named J, in Figure 4.4. They are calculated from the point where the bars leave the bottom layer till the point where they are anchored, 800 mm behind the nodes. Thus, the bottom part of the J-bars are included in "Horizontal bars" in Table 4.7 (see Figure 4.5).

**Figure 4.4:** Reinforcement design for Model B**Figure 4.5:** Description of how the J-bars are divided into "Horizontal bars" and "Inclined bars" in Table 4.7 and Table 4.10

**Table 4.7:** The volume of the reinforcement used for the reinforcement design of Model B

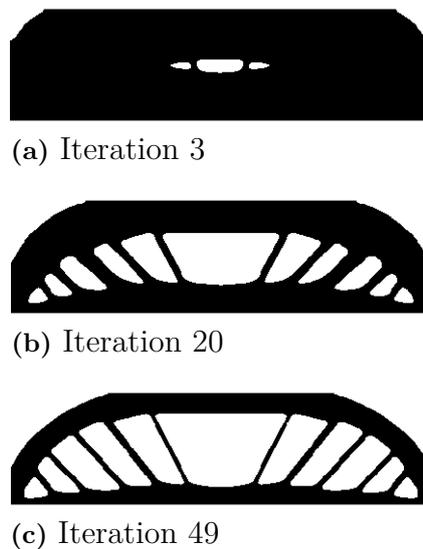
Inclined bars [m <sup>3</sup> ]	0.047
Stirrups [m <sup>3</sup> ]	0.039
Horizontal bars [m <sup>3</sup> ]	0.083
Total reinforcement [m <sup>3</sup> ]	0.169
Concrete [m <sup>3</sup> ]	14.880
Volume ratio [-]	0.011
Mass to volume ratio [kg/m <sup>3</sup> ]	88.589

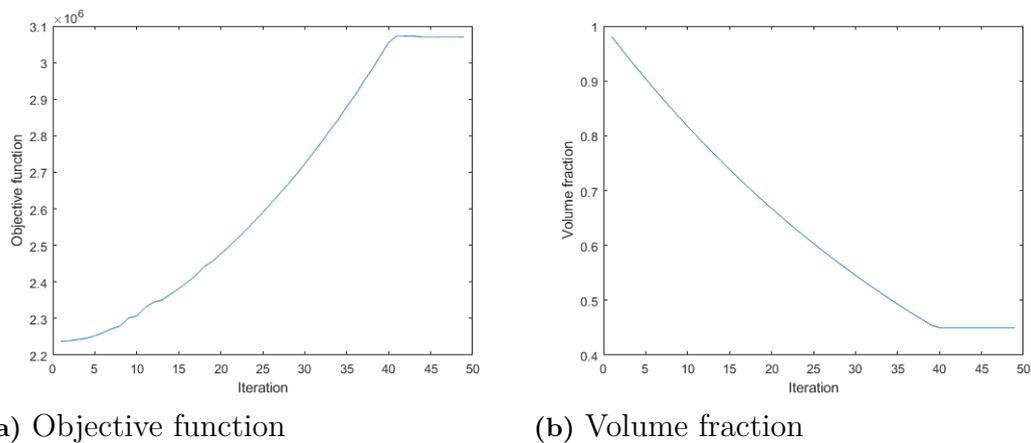
### 4.3 Results from the sub models on which Model C are based

In this section the results from the optimization of *Model C - 1*, *Model C - 2* and *Model C - 3* will be presented. The calculations of the developed final Model C will be presented in section 4.4.

#### 4.3.1 Model C - 1

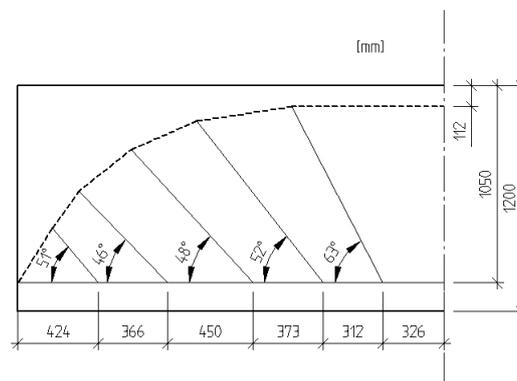
The development of the structural optimization for Model C - 1, which was performed in MATLAB, can be seen in Figure 4.6. Here, three of the 49 iterations are visualized. Figure 4.7 visualises the corresponding graphs for the objective function and volume fraction. Iteration 49 is the last iteration from this analysis and the resulting model from this iteration was further analyzed.

**Figure 4.6:** The development of the structural optimization for Model C - 1



**Figure 4.7:** Corresponding graphs for the objective function and the volume fraction for the final design of sub-model Model C - 1. Graphs created in MATLAB.

The angles and relevant lengths for the ST model, that comes out of the 49:th iteration of the FE analysis, are visualized in Figure 4.8. This ST model contributes to the final design of Model C.



**Figure 4.8:** The angles and relevant lengths for the ST model that comes out of Model C - 1

### 4.3.2 Model C - 2

The development of the structural optimization for Model C - 2, which was performed in BRIGADE/Plus, can be seen in Figure 4.9. Here, three of the 60 iterations are visualized. Figure 4.10 visualizes the corresponding graphs for the objective function and volume fraction. Iteration 60 is the last iteration for this analysis and the resulting model from this iteration was further analyzed.



(a) Iteration 10

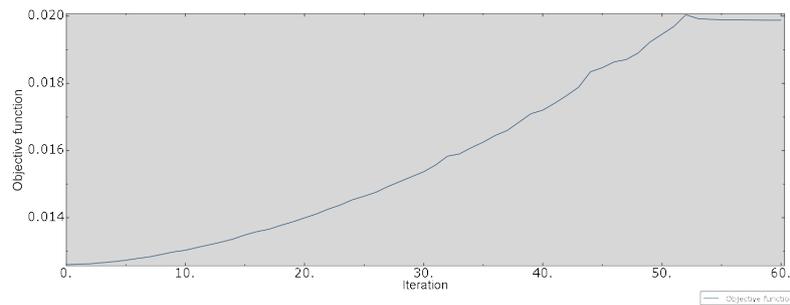


(b) Iteration 40

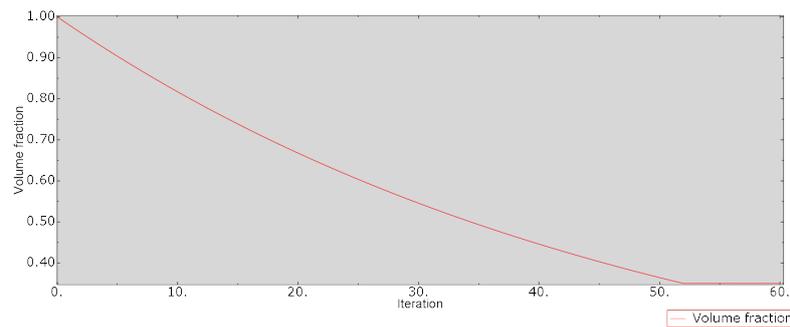


(c) Iteration 60

**Figure 4.9:** The development of the structural optimization for Model C - 2



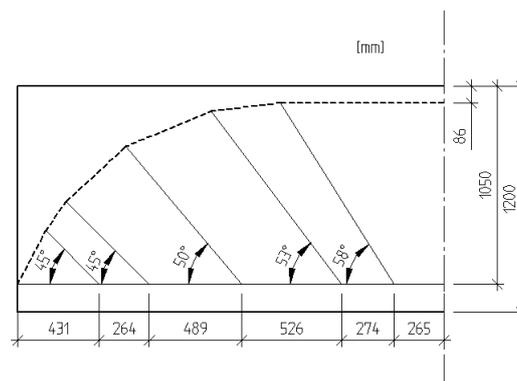
(a) Objective function



(b) Volume fraction

**Figure 4.10:** Corresponding graphs for the objective function and the volume fraction for the final design of sub-model Model C - 2. Graphs created in BRIGADE/Plus.

The angles and relevant lengths for the ST model, that comes out of the 60:th iteration of the FE analysis, are visualized in Figure 4.11. This ST model contributes to the final design of Model C.



**Figure 4.11:** The angles and relevant lengths for the ST model that comes out of Model C - 2

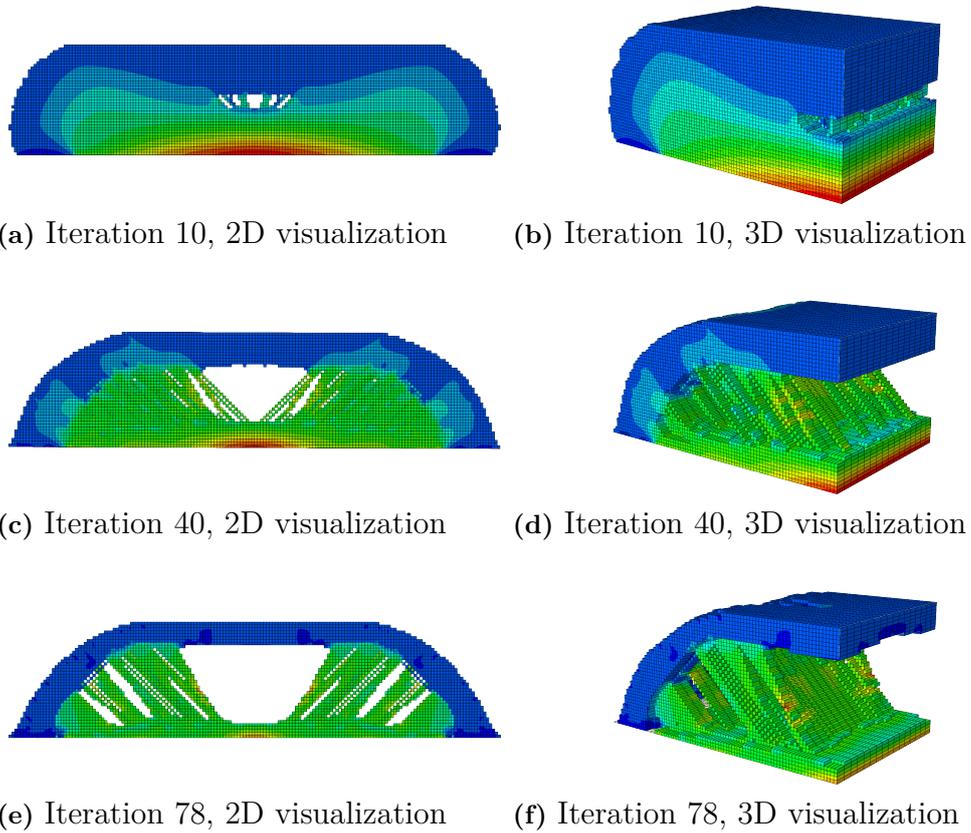
### 4.3.3 Model C - 3

The development of the structural optimization for Model C - 3, which was performed in BRIGADE/Plus, can be seen in Figure 4.12. Here, three of the 78 itera-

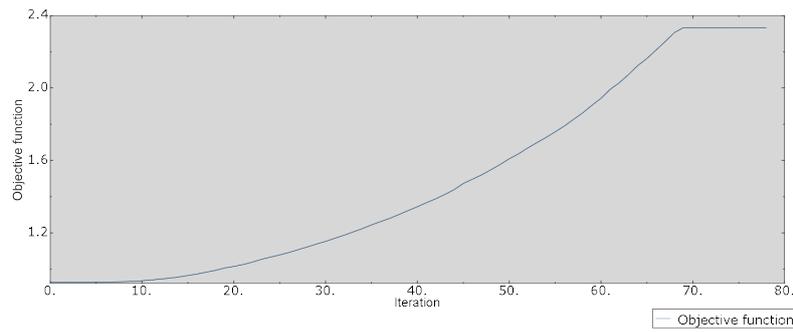
## 4. Results

---

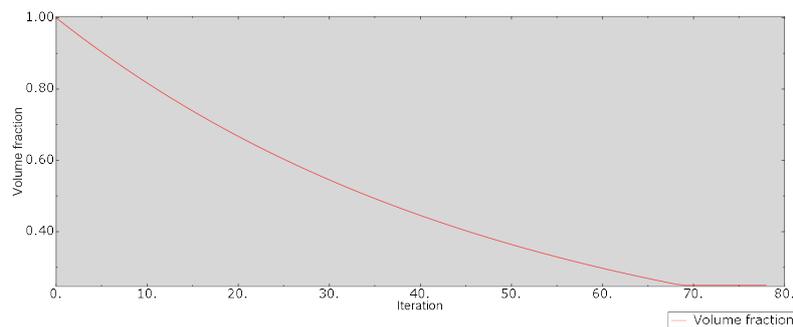
tions are visualized. Note that both a 2D and 3D view are presented. Figure 4.13 visualizes the corresponding graphs for the objective function and volume fraction. Iteration 78 was the last iteration before convergence and the resulting model from this iteration was further analyzed.



**Figure 4.12:** The development of the structural optimization for Model C - 3



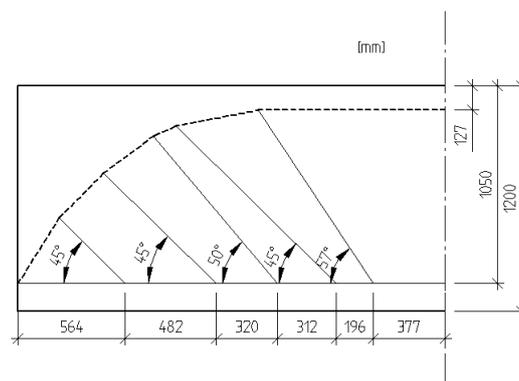
(a) Objective function



(b) Volume fraction

**Figure 4.13:** Corresponding graphs for the objective function and the volume fraction for the final design of sub-model Model C - 3. Graphs created in BRIGADE/Plus.

The angles and relevant lengths for the ST model, that comes out of the 78:th iteration of the FE analysis, are visualized in Figure 4.14. This ST model contributes to the final design of Model C.



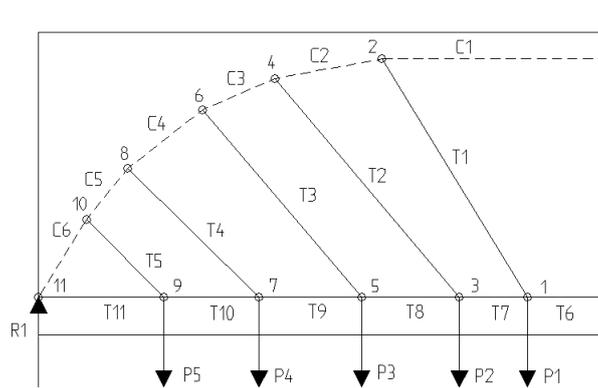
**Figure 4.14:** The angles and relevant lengths for the ST model that comes out of Model C - 3

## 4.4 Results from Model C

In this section, the results from Model C are presented, in terms of forces in struts and ties, checks of critical nodes and the final reinforcement design.

### 4.4.1 Forces in struts and ties from Model C

The locations and labeling of struts, ties, nodes and external and reaction forces are presented in Figure 4.15 and the associated magnitudes are presented in Table 4.8.



**Figure 4.15:** Locations of struts, ties, nodes and external and reaction forces for Model C

**Table 4.8:** Normal forces in struts and ties together with external and reaction forces for Model C

Strut	[kN]	Tie	[kN]	Point loads and reaction force	[kN]
$C_1$	4 897	$T_1$	787	$R_1$	3 335
$C_2$	4 561	$T_2$	633	$P_1$	674
$C_3$	4 472	$T_3$	765	$P_2$	485
$C_4$	4 560	$T_4$	819	$P_3$	586
$C_5$	4 790	$T_5$	1 430	$P_4$	579
$C_6$	3 891	$T_6$	4 897	$P_5$	1011
		$T_7$	4 492		
		$T_8$	4 085		
		$T_9$	3 594		
		$T_{10}$	3 015		
		$T_{11}$	2 004		

### 4.4.2 Check of critical nodes for Model C

Node 2 and 11 in Figure 4.15 are the critical nodes in Model C and are checked against the requirements given by *Eurocode 2* (2008). The results are presented in

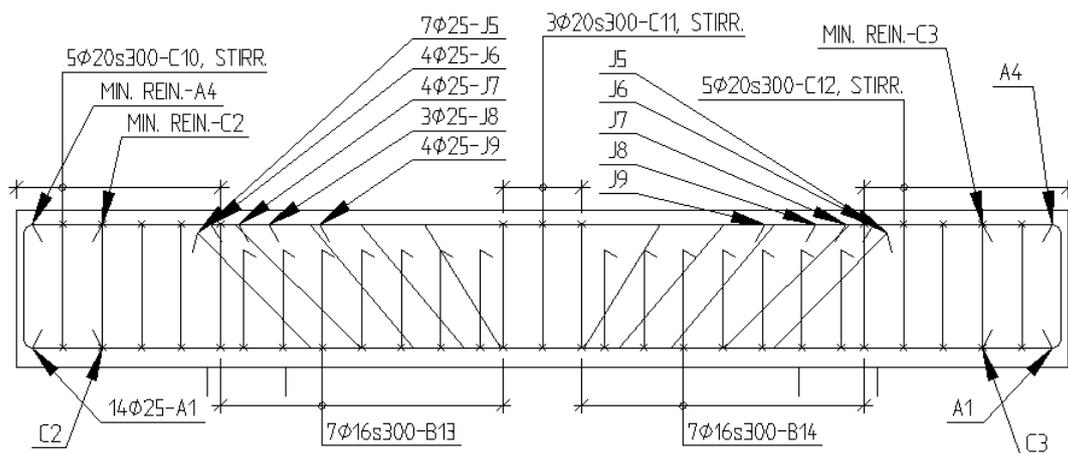
Table 4.9. The first index, in the node columns, stands for node number and the last one for which force that is used for the check.

**Table 4.9:** Checks according to requirements from *Eurocode 2* (2008) for Model C

Maximal stress		Node 2	Node 11	
$\sigma_{Rd,max}$	17.1 MPa	$\sigma_{2.C_1}$	15.2 MPa	
			$\sigma_{11.R_1}$	13.9 MPa
			$\sigma_{11.C_6}$	14.5 MPa

### 4.4.3 Reinforcement design for Model C

The reinforcement design for Model C is presented in Figure 4.16. The volume of the reinforcement, that corresponds to this design, and its ratio to the amount of concrete is presented in Table 4.10. The same distribution for the J-bars are applied, as described in section 4.2.3 and Figure 4.5. The volumes for the bars named B13 and B14 are included in "Stirrups" in Table 4.10.



**Figure 4.16:** Reinforcement design for Model C

**Table 4.10:** Volume of the reinforcement used for the reinforcement design of Model C

Inclined bars [m <sup>3</sup> ]	0.035
Stirrups [m <sup>3</sup> ]	0.025
Horizontal bars [m <sup>3</sup> ]	0.079
Total reinforcement [m <sup>3</sup> ]	0.139
Concrete [m <sup>3</sup> ]	14.880
Volume ratio [-]	0.009
Mass to volume ratio [kg/m <sup>3</sup> ]	72.863

# 5

## Discussion

The studies performed in Chapter 2 and 3 together with the results presented in Chapter 4 are based on the aim of finding an effective ST model by using structural topology optimization. By adopting the BESO method and applying it on the crossbeam described in Section 3.1, three types of optimized sub-models could be developed. These sub models were combined into Model C, which was analyzed and then compared with the models developed by hand, Model A and B. The different models will be discussed in the following sections, including advantages and disadvantages and problems that arose during this thesis. Finally, the results from the different models will be compared and discussed.

### 5.1 Model A

Model A was based on the conventional ST method and how IHT implements it when designing crossbeams for supports in bridges. The procedure was straight forward and results could be extracted within a day. The method could be used for crossbeams with different geometries and loading conditions, with few modifications, which made it possible to experiment with the dimensions of the crossbeam. The ST method is a rational approach to how the forces should be modeled and how the reinforcement should be arranged, which makes it suitable for quick but reliable estimations.

For Model A, the results corresponded well to the documents provided by IHT. However, to get more comparable results to Model B and C, Model A was slightly optimized by shortening of the horizontal reinforcement to only fulfill the need for a particular cross-section. In IHT's reinforcement design, the same amount of horizontal reinforcement was placed over the whole crossbeam. This was done since the savings from reduction of reinforcement amount was judged to be smaller than the cost of adjusting the lengths of the reinforcement bars. Such considerations are common in the construction industry, but in this thesis the cost of that something is complicated or time consuming to build is not taken into account. Instead, focus was to investigate the reinforcement amount needed and finding an optimal ST model for this.

Compared to Model B and C, the type of ST modelling applied for Model A, vastly increases the need for vertical stirrups, which can be seen in the comparison in Table 5.2. The main reason for this is that the vertical ties carry the load straight vertically while the inclined ties are oriented so that they utilizes the concrete in a more effective way.

## 5.2 Model B

The procedure of creating Model B was also straightforward and the model could be developed quickly following the procedure presented in Section 3.2.2. Since only four inclined ties were used, the distance between them became quite large. Therefore, it was chosen to let the stirrups be placed over the whole beam to satisfy the minimum reinforcement amount.

One question that arose regarding the inclined bars was how complicated and time consuming it would be to construct such a reinforcement design and whether or not it is inconvenient for the construction workers. But with the same argument that was mentioned in previous section, the focus was placed on arranging the reinforcement bars in line with the inclined ties, which resulted in the final designs of this thesis.

## 5.3 Model C

Model C was obtained by using structural topology optimization, namely BESO. By applying the BESO algorithm to both 2D and 3D models, optimized models were found. These models displayed optimums for transferring the distributed load in terms of load paths to the supports. Three sub-models were created and similarities in load paths were searched for. This allowed the stiffness and load to be treated arbitrarily, as long as convergence was met. By combining the three sub-models, discussed in the following sections, the final ST model for Model C could be established. Model C is more complicated than Model A and Model B, and to develop and analyze this model has been comparably time consuming.

When comparing the simulations in MATLAB versus BRIGADE/Plus the difference in time, required to run the simulations, should be noted. The required times for the simulations of Models C - 1, C - 2 and C - 3 are summarized in Table 5.1. As can be seen, the difference in time is huge and therefore MATLAB was an excellent tool for investigating the effects of various inputs, which then increased the knowledge for the simulations in BRIGADE/Plus. The reason to why there is a substantial difference in the computational time, between MATLAB and BRIGADE/Plus, might be that MATLAB is a computational program which is very efficient when it comes to creating and compute matrices. Another reason could be how the creation of the filter map is performed for the different programs, but to draw conclusions regarding this, more knowledge about the different FEA solvers needs to be obtained.

**Table 5.1:** Times for simulations of the sub-models

Model	Elements	Create filter map	Performing iterations	Total time
C - 1	54 000	3 s	40 s	43 s
C - 2	54 000	8 h 20 min	1 h 18 min	9 h 38 min
C - 3	128 000	46 h 40 min	2 h 23 min	49 h 3 min

### 5.3.1 Model C - 1

This sub-model was created in MATLAB so in this sub-model, many combinations of inputs could be tested in an time efficient way. This contributed to the understanding of the effects of how the filter radius ( $r_{min}$ ) affected the occurrence of checkerboard patterns as visualized in Figure 2.11. It also gave insight to how the two other parameters, evolutionary ratio ( $ER$ ) and volume fraction ( $V_f$ ), affected the results. By using an evolutionary ratio somewhere between 2-4%, the optimized results converged while using larger values gave inconsistency in the final models.  $V_f$  affected the results of the model in the way that when it got too low, changes in the bottom edge of the crossbeam occurred. For Model C - 1 the consistency of the model was kept and an optimized model could be produced even with low  $V_f$ . However, for Model C - 2 and Model C - 3, the final result was affected, which will be discussed in the following subsections.

### 5.3.2 Model C - 2

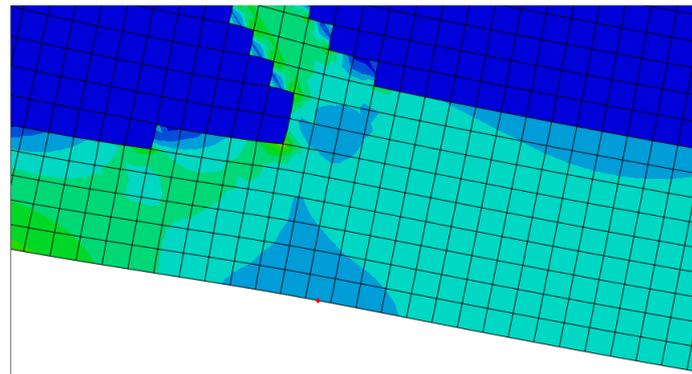
By varying  $V_f$  for the model it could be seen that for  $V_f < 35\%$  convergence was lost and distortion problem arose. This was due to the fact that elements on the bottom were assigned void properties and therefore the smallest stresses occurred in the nodes of these elements. This is shown in 5.1 where the blue part indicates low stresses in the elements.

For the optimizations conducted in BRIGADE/Plus it was discovered that the optimization simulations became time consuming as the number of elements increased. This problem was reduced by modifications to the python script which allowed to save the filter map ( $FM$ ) for models and reuse them for models with the same mesh. Thus, the optimization time could be reduced and optimization simulations could be performed during nights and weekends.

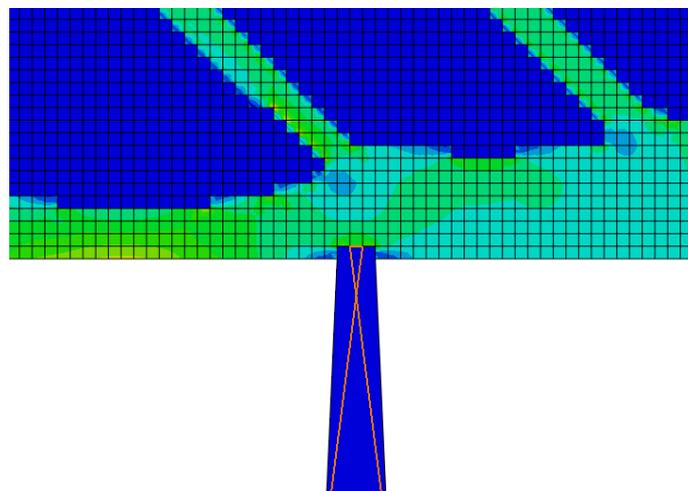
### 5.3.3 Model C - 3

Model C - 3 is a 3D model which was translated into a 2D ST model, in order to combine it with Model C - 1 and Model C - 2. The 3D optimization was performed on a beam modeled with rigid plates as supports. It could be seen from the final model of the beam that two arches were created to transfer the load to the corners of the plates. Instead of translating this 3D model into a 2D ST model, the model could have been translated into a 3D ST model. With a 3D ST model, the reinforcement could have a 3D arrangement which might had improved the reinforcement usage even further.

To reduce the time to create filter maps, it was decided to assign the elements different lengths in different directions. Since the final objective with the 3D model, in this case, was to find a model that could be translated into a 2D model, the elements were assigned smaller lengths in x- and y-direction than in z-direction. This was made since the beam was simply supported meaning that the load was carried mainly in the xy-plane and not in the z-direction.



(a) Iteration 56 undistorted



(b) Iteration 57 distorted

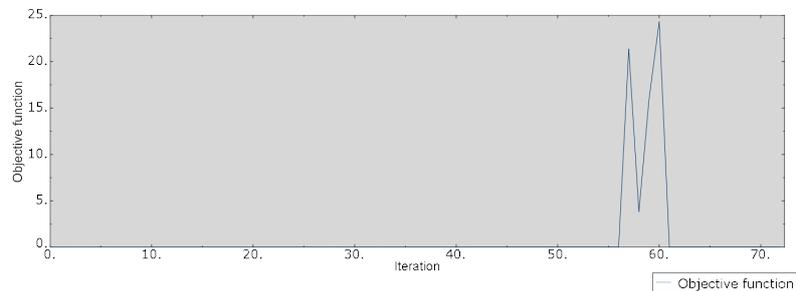
**Figure 5.1:** Iteration step 56 and 57 displaying how the code assigns void properties to the elements on the bottom edge which leads to distortion in iteration step 57

### 5.3.4 Convergence of the sub-models

A big part of the model development was understanding how the optimization routine worked and what inputs and refinements to the models were needed to get converging results.

One major issue to overcome in this thesis was to deal with the distortion of elements used in the simulations. A trial and error process was conducted to overcome the obstacle of elements distortion which affected the final results of the BESO iterations. By plotting the objective function and the volume fraction, as visualized in e.g. Figure 4.13, a tool for convergence investigation was obtained. The two curves should correlate to each other in order to obtain good results. It could be seen that if the objective function had "spikes" that disrupted the graph as visualized in Figure 5.2, distortion of elements had occurred. The graph in Figure 5.2 describes the objective function for the analysis shown in Figure 5.1. Distortion can indicate excessive increase in external work, but since the load was constant on the structure

the spikes here implicated that the elemental strains increased towards infinity as a result of element distortions.



**Figure 5.2:** Objective function with distorting elements

One reason why this happened could be that the filter scheme, applied in the python code, was too rough and would need further improvement. Another reason could be that the mesh was too coarse or that an element type that is too sensitive to distortion was used. It was also discovered that when the BESO algorithm switches two elements that shared the same node into void elements, and the same elements were subjected to external load in the next iteration, distortion was unavoidable.

Distortion was not a problem for Model C - 1 so improvements were only done for Model C - 2 and Model C - 3. For Model C - 2 it was discovered that a  $V_f = 35\%$  together with second order elements with reduced integration (CPS8R) gave results resembling the results obtained by Model C - 1. For Model C - 3 the problem was solved by applying a load dividing thin shell on the bottom of the beam, used to distribute the load, and by applying 8 node brick elements and reduced integration with enhanced hourglass control (C3D8R).

Even though the spikes on the objective function curve indicate problem with the models, there was a possibility that the results could be useful anyway. If the results, i.e. the model to be translated into a ST-model, was extracted from iterations before or after the element distortion, reasonable models could be obtained in many cases.

Another thing that affected the final results was to choose the right  $r_{min}$ . From literature it was found that a sufficient  $r_{min}$  should be around three times the element size. To confirm this, optimization was performed on the same model but with alternating  $r_{min}$ . For Model C - 2 and Model C - 3, a  $r_{min}$  of approximately three times the largest element length was found suitable, while for Model C - 1 an  $r_{min}$  of six times the element length gave similar results as the other models.

## 5.4 Comparison between the models

The comparison of the reinforcement volume for the final design is presented in Table 5.2. Following notes can be made regarding the results shown in this table.

The difference in horizontal reinforcement bars is relatively small. This means that all models have similar need for horizontal reinforcement in the bottom part of the beam, which is reasonable since geometry and loading conditions are the same. For Model B and C, there is more reinforcement bars placed in mid span than what the middle tie requires. The reason for this is that the J-bars deviates at certain sections while the requirement for the ties closest to support still needs to be satisfied. This means that it might be possible to further optimize the models by e.g. using other angles for the ties.

The need for stirrups is the largest factor to the difference in the final results. Model A uses vertical stirrups to take care of the lifting of the forces, while this is taken care of by the inclined ties in Model B and C. Even though the ULS requirement is met by all ST models, the requirement of maximum center to center distance between reinforcement bars has to be fulfilled. Therefore, Model B and C has been assigned the largest allowable center to center distance of 300 mm for the stirrups, which corresponds to the maximum distance chosen for this thesis. This means that the vertical stirrups in mid span in Model B and C do not fulfill any particular purpose when it comes to lifting of the forces, which in turn means that the span is oversized for those two models.

**Table 5.2:** Comparison of the reinforcement volume for the final designs

Model	Inclined bars [ $m^3$ ]	Stirrups [ $m^3$ ]	Horizontal bars [ $m^3$ ]	Sum [ $m^3$ ]
A	-	0.087	0.085	0.172
B	0.047	0.039	0.083	0.169
C	0.035	0.025	0.079	0.139

In Table 5.3, the forces in the ties are summarized. All the tie forces that are not horizontal are summarized in "Sum of the lifting ties" while the largest single tie force in the mid span is presented in "Horizontal tie in mid span". Thus, there are some horizontal tie forces that are excluded in this table, since they are not relevant for this discussion.

It is clear that the forces in the lifting ties become larger as the the angle between the horizontal ties and the lifting ties increases. This is reasonable using simple trigonometry knowledge, but it also means that the forces in the compression struts become larger. Therefore, a ST model with inclined ties utilizes the compression strength of the concrete in a more efficient way. Another note to make, regarding this phenomena, is that the force in the horizontal tie in mid span becomes larger as the forces in the lifting ties decrease. This is also logical since the inclined ties contribute with horizontal tension in the bottom layer.

For this type of structural element, it is common to place an amount of horizontal reinforcement bars that satisfy the requirement given by the mid span tie along the whole crossbeam. As mentioned in Section 5.1, this simplification is not done in this thesis since the optimal design is sought for. It is clear, if comparing Table

5.2 to Table 5.3, that even though the tie force is the smallest for Model A, the reinforcement design for Model A uses the largest horizontal reinforcement amount. The reason for this is, as mentioned in Section 5.1, that shortening of the horizontal reinforcement bars is taken into account. The need in mid span is the largest for Model C, but this need is also required along a shorter length.

**Table 5.3:** Comparison of forces in the ties

Model	Sum of the lifting ties [ $kN$ ]	The horizontal tie in mid span [ $kN$ ]
A	6 543	4 022
B	5 436	4 752
C	4 434	4 897

# 6

## Conclusion

The aim of this master thesis was to investigate the potential in using structural optimization to find effective ST models. This has been achieved by performing a case study on a crossbeam designed by the civil engineering company Inhouse Tech AB. By applying structural optimization, in this thesis BESO, to both 2D and 3D models it was shown that it is possible to use structural optimization to generate effective ST models, which also proves the statement about conventional ST method being over-conservative.

Regarding the convergence of the models, it could be seen that the inputs and how the model was constructed had a large influence of how the final results came out. The filter radius, volume fraction and type of element needed to be chosen correctly. It was found that to get coherent models, the filter radius for Model C - 1 should be around six times the element size, while for Model C - 2 and C - 3, it was sufficient with three times the largest element size. The volume fraction was chosen to 45%, 35% and 25% for the three sub-models.

It can also be concluded that the development of the objective function is an important indicator for how the final model is going to look. The development of the objective function needs to reflect the development of the volume fraction curve. Otherwise it was proven that, in most cases, convergence was lost and the results could not be translated to a ST model.

By comparing Model A, Model B and Model C it can be seen that the influence of inclined reinforcement contributes to a more efficient utilization of the compressive strength of the concrete, which leads to a reduced need of reinforcement. For the case study in this thesis, it was seen that when comparing Model B and C to Model A, the reduction of reinforcement is 1.7% and 19.2%, respectively.

By studying Shobeiri & Ahmadi-Nedushan (2017) it could be seen that BESO was applicable to a wide range of different types of structural problems. This means that this method of finding an effective ST model can be applied to structural elements where the load paths is hard to predict in an analytical way.

BRIGADE/Plus, together with the Python script used, were not optimal for structural optimization. It was time consuming to build appropriate models and the final results were much dependent on the different inputs. In addition, the simulations required much time. The m-script was much more time efficient to use. In this script the model was already built within the script and this saved a lot of time. However, the m-script was limited to only perform 2D analysis and to perform structural op-

## 6. Conclusion

---

timization on a beam with three different load cases and boundary conditions. To build different models in MATLAB would of course be possible to do but this, on the other hand, is much more convenient to do in BRIGADE/Plus.

# 7

## Further work

The optimization routine could easily be extended to investigate other commonly used ST-models in order to refine how they are structured in accordance to Eurocode.

As mentioned in section 3.1 the load is assumed to act at the bottom of the crossbeam, but in the real case it is distributed over the height of the beam. By modeling the whole system, crossbeams and main girder, with the correct loads applied, BESO could be used to find and investigate the load paths from the main girder via the crossbeam to the supports. In this way, conclusions regarding how the load could be applied to the ST models developed in this thesis, could be drawn. This could contribute to refinement of the ST models used when designing crossbeams.

Since this thesis has dealt with the development of ST models, which are valid for ULS only, SLS calculations also needs to be performed. This, in combination with the question of whether the handling of inclined ties are too complex or not, would be a suiting continuation for further work.

As mentioned in conclusion, the software used in this thesis were not optimal for structural optimization. Another further work could be to use commercial optimization software, such as TOSCA, and develop a routine for designing 3D ST models. Since commercial software are more refined than the Python script used in this thesis, reduced computational times and less complications with the models could be expected. Thus, by developing a routine that could be applied to a wide range of different structures, it would be possible to help the designer in her/his everyday work.

# References

- Austrell, P.-E., Dahlbom, O., Lindeman, J., Olsson, A., Olsson, K.-G., Persson, K., Petersson, H., Ristinmaa, M., Sandberg, G. & Wernberg, P.-A. (2004), *CALFEM, A FINITE ELEMENT TOOLBOX*, version 3.4 edn, Lund.
- BRIGADE/Plus 6.2-5* (2019).  
**URL:** <https://scanscot.com/products/bridge-design/brigade-plus/>
- Christiansen, P. W. & Klarbring, A. (2009), *An Introduction to Structural Optimization*, Springer, Linköping.
- Dahl, T. (2018), Optimering av böjarmering genom FEM-analys och strut-and-tie modellering, Technical report, Division of Structural Mechanics, LTH, Lund.
- Defining rigid body constraints* (2017).  
**URL:** <https://abaqus-docs.mit.edu/2017/English/SIMACAECAERefMap/simacae-t-itnhelptopicrigid.htm>
- Engström, B. (2011), *Design and analysis of deep beams, plates and other discontinuity regions.*, Chalmers University of Technology, Göteborg.
- Eurocode 2* (2008), ss-en 1992-1-1:2005 edn, SIS, Stockholm.
- Hendy C. R. & Smith D. A. (2007), *Designers' Guide to En 1992-2. Eurocode 2: Design of Concrete Structures. Part 2: Concrete Bridges*, ICE Publishing, London.
- Huang, X. X. & Xie, Y. M. (2010), *Evolutionary topology optimization of continuum structures : methods and applications*, Wiley.
- Mathern, A. & Chantelot, G. (2010), Strut-and-tie modelling of reinforced concrete pile caps, Technical report, Division of Structural Engineering, Gothenburg.
- Mathern, A., Chantelot, G., Svahn, P.-O., Kettil, P., Rasmus, R. & Engström, B. (2017), Enhanced strut-and-tie model for reinforced concrete pile caps, Technical report, IABSE Symposium, Vancouver.
- Obbink-Huizer, C. (2018), 'Automate boring postprocessing in Abaqus using python scripting'.  
**URL:** <https://info.simuleon.com/blog/automate-boring-postprocessing-in-abaqus-using-python-scripting>
- Oswaldo M, Q., Mariano, V., Cristina, A., Rubén, A. & Pascual, M. (2017), *Topology Design Methods for Structural Optimizaation*, Elsevier Ltd, London.

- 
- Plevris, V. & Tsiatas, G. C. (2018), ‘Computational Structural Engineering: Past Achievements and Future Challenges’, *Frontiers in Built Environment* 4.
- Python Software Foundation (2019), ‘11.1. pickle — Python object serialization — Python 2.7.16 documentation’.  
**URL:** <https://docs.python.org/2/library/pickle.html>
- Qing Quan Liang, Xie, Y. M. & Grant Prentice Steven (2000), ‘Topology Optimization of Strut-and-Tie Models in Reinforced Concrete Structures Using an Evolutionary Procedure’, *ACI Structural Journal* (2).
- Saabye Ottosen, N. & Petersson, H. (1992), *Introduction to the finite element method*, Prentice Hall, New York.
- Saraswat, S. (n.d.), ‘Python Language Introduction - GeeksforGeeks’.
- Shobeiri, V. & Ahmadi-Nedushan, B. (2017), ‘Bi-directional evolutionary structural optimization for strut-and-tie modelling of three-dimensional structural concrete’, *Engineering Optimization* .
- Vishark, R. (n.d.), ‘What are the three main criteria to determine the slave and master surfaces in ABAQUS? - Quora’.
- Xia, L., Xia, Q., Huang, X. & Xie, Y. M. (2016), ‘Bi-directional Evolutionary Structural Optimization on Advanced Structures and Materials: A Comprehensive Review’, *Archives of Computational Methods in Engineering* .
- Zuo, Z. H. & Xie, Y. M. (2015), ‘A simple and compact Python code for complex 3D topology optimization’, *Advances in Engineering Software* .

# A

## Hand calculations

### A.1 Hand calculations on Model A

#### Geometry

$H := 1200 \text{ mm}$

Height of the beam

$b := 1550 \text{ mm}$

Width of the beam

$L := 4500 \text{ mm}$

Span length

#### Strut and tie model

$A := 498 \text{ mm}$

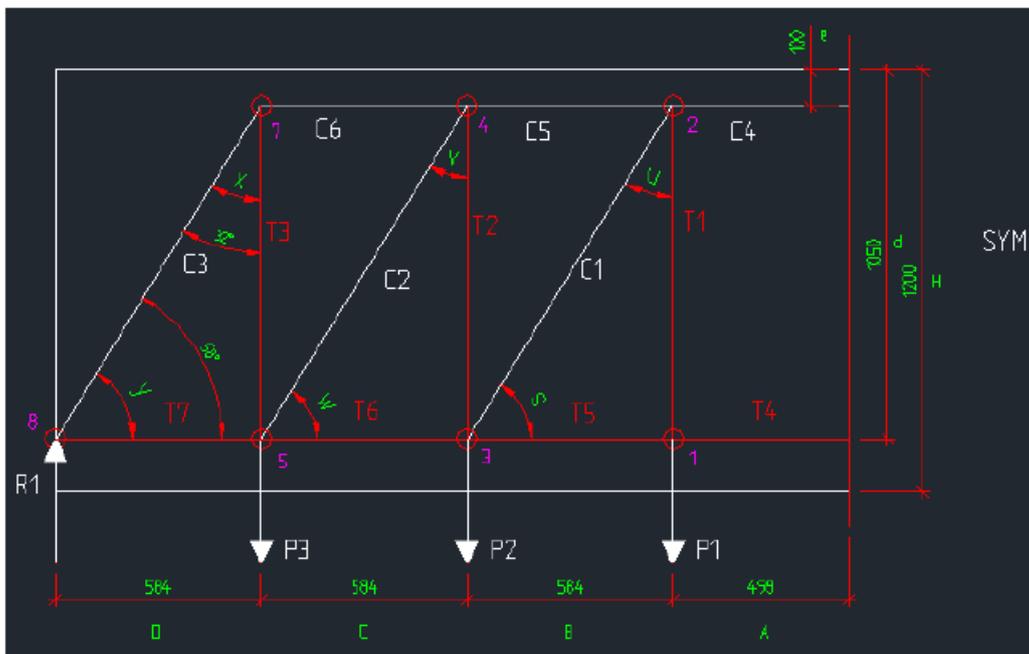
$B := 584 \text{ mm}$

$C := 584 \text{ mm}$

$D := 584 \text{ mm}$

$d := 1050 \text{ mm}$

$e := 100 \text{ mm}$



### Angles

$$y := \operatorname{atan}\left(\frac{d-e}{D}\right) = 58.419 \text{ deg}$$

$$x := 90 \text{ deg} - y = 31.581 \text{ deg}$$

$$w := \operatorname{atan}\left(\frac{d-e}{C}\right) = 58.419 \text{ deg}$$

$$v := 90 \text{ deg} - w = 31.581 \text{ deg}$$

$$s := \operatorname{atan}\left(\frac{d-e}{B}\right) = 58.419 \text{ deg}$$

$$u := 90 \text{ deg} - s = 31.581 \text{ deg}$$

### Loads

$$V_d := 1779 \text{ kN}$$

$$P_{Ed} := 6670 \text{ kN}$$

$$R_1 := \frac{P_{Ed}}{2} = 3335 \text{ kN}$$

$$q_{Ed} := \frac{P_{Ed}}{L} = 1.482 \frac{\text{kN}}{\text{mm}}$$

$$P_1 := q_{Ed} \cdot \left(A + \frac{B}{2}\right) = 1170.956 \text{ kN}$$

$$P_2 := q_{Ed} \cdot \left(\frac{B}{2} + \frac{C}{2}\right) = 865.618 \text{ kN}$$

$$P_3 := q_{Ed} \cdot \left(\frac{C}{2} + D\right) = 1298.427 \text{ kN}$$

$$P := P_1 + P_2 + P_3 = 3335 \text{ kN}$$

$$R_1 = 3335 \text{ kN}$$

OK

## Forces in the truss

$$T_1 := P_1 = 1170.956 \text{ kN}$$

$$C_1 := \frac{T_1}{\cos(u)} = 1374.515 \text{ kN}$$

$$T_2 := P_2 + C_1 \cdot \cos(u) = 2036.573 \text{ kN}$$

$$C_2 := \frac{T_2}{\cos(v)} = 2390.612 \text{ kN}$$

$$T_3 := P_3 + C_2 \cdot \cos(v) = 3335 \text{ kN}$$

$$C_3 := \frac{T_3}{\cos(x)} = 3914.758 \text{ kN}$$

$$R_1 := C_3 \cdot \sin(y) = 3335 \text{ kN} \quad \mathbf{OK}$$

$$C_6 := C_3 \cdot \sin(x) = 2050.147 \text{ kN}$$

$$C_5 := C_6 + C_2 \cdot \sin(v) = 3302.104 \text{ kN}$$

$$C_4 := C_5 + C_1 \cdot \sin(u) = 4021.934 \text{ kN}$$

$$T_7 := C_3 \cdot \cos(y) = 2050.147 \text{ kN}$$

$$T_6 := T_7 + C_2 \cdot \cos(w) = 3302.104 \text{ kN}$$

$$T_5 := T_6 + C_1 \cdot \cos(s) = 4021.934 \text{ kN}$$

$$T_4 := T_5 = 4021.934 \text{ kN}$$

$$C_4 = 4021.934 \text{ kN} \quad \mathbf{OK}$$

## Check of nodes

Compression - tension node with anchored ties in one direction and where the the bars are anchored within the node.

$$f_{ck} := 35 \text{ MPa}$$

$$f_{cd} := \frac{f_{ck}}{1.5} = 23.333 \text{ MPa}$$

$$k_2 := 0.85$$

$$\nu := 1 - \frac{f_{ck}}{250 \text{ MPa}} = 0.86$$

$$\sigma_{rd,max} := k_2 \cdot \nu \cdot f_{cd} = 17.057 \text{ MPa}$$

## Control of node 7

$$u := 2 \cdot (H - d) = 300 \text{ mm}$$

$$a_1 := 400 \text{ mm}$$

$$b_1 := 600 \text{ mm}$$

$$a_2 := a_1 \cdot \sin(y) + u \cdot \cos(y) = 497.871 \text{ mm}$$

$$b_2 := b_1 = 600 \text{ mm}$$

$$\sigma_{d1.7} := \frac{R_1}{a_1 \cdot b_1} = 13.896 \text{ MPa}$$

**OK**

$$\sigma_{d2.7} := \frac{C_3}{a_2 \cdot b_2} = 13.105 \text{ MPa}$$

**OK**

## Control of node 2

$$a_1 := 2 \cdot e = 200 \text{ mm}$$

$$b_1 := b = 1550 \text{ mm}$$

$$\sigma_{d1.2} := \frac{C_4}{a_1 \cdot b_1} = 12.974 \text{ MPa} \quad \text{OK}$$

## Design of ties

$$f_{st} := 300 \text{ MPa} \quad f_{yk} := 500 \text{ MPa}$$

$$f_{st.lift} := 435 \text{ MPa}$$

## Vertical reinforcement:

Reinforcement equal to T1 is used on the length  $A + \frac{B}{2} = 790 \text{ mm}$

$$A_{sT1} := \frac{T_1}{f_{st.lift}} = 2691.852 \text{ mm}^2 \quad A_{siT1} := 314 \text{ mm}^2 \quad d=20\text{mm}$$

$$n_{T1} := \frac{A_{sT1}}{A_{siT1}} = 8.573 \quad n_{T1} := 9 \quad \text{Number of stirrups}$$

Reinforcement equal to T2 is used on the length  $\frac{B}{2} + \frac{C}{2} = 584 \text{ mm}$

$$A_{sT2} := \frac{T_2}{f_{st.lift}} = 4681.778 \text{ mm}^2 \quad A_{siT2} := 314 \text{ mm}^2$$

$$n_{T2} := \frac{A_{sT2}}{A_{siT2}} = 14.91 \quad n_{T2} := 15$$

Reinforcement equal to T3 is used on the length  $\frac{C}{2} + D = 876 \text{ mm}$

$$A_{sT3} := \frac{T_3}{f_{st.lift}} = 7666.667 \text{ mm}^2 \quad A_{siT3} := 314 \text{ mm}^2$$

$$n_{T3} := \frac{A_{sT3}}{A_{siT3}} = 24.416 \quad n_{T3} := 25$$

Total amount of lifting reinforcement:

$$A_{lift} := 2 \cdot A_{sT1} + 2 \cdot A_{sT2} + 2 \cdot A_{sT3} = 30080.593 \text{ mm}^2$$

### Horizontal reinforcement

$$A_{sT4} := \frac{T_4}{f_{st}} = 13406.445 \text{ mm}^2 \quad A_{siT4} := 491 \text{ mm}^2 \quad d=25\text{mm}$$

$$n_{T4} := \frac{A_{sT4}}{A_{siT4}} = 27.304 \quad n_{T3} := 28$$

Total amount of reinforcement:

$$A_{lift} = 30080.593 \text{ mm}^2$$

$$A_{horizontal} := A_{sT4} = 13406.445 \text{ mm}^2$$

$$A_{tot} := A_{lift} + A_{horizontal} = 43487.038 \text{ mm}^2$$

### Reinforcement design

#### Bars

$$\phi_{horizontal} := 25 \text{ mm}$$

$$A_{s.i.h} := \frac{\pi \cdot \phi_{horizontal}^2}{4} = 490.874 \text{ mm}^2$$

$$\phi_{vertical} := 20 \text{ mm}$$

$$A_{s.i.v} := \frac{\pi \cdot \phi_{vertical}^2}{4} = 314.159 \text{ mm}^2$$

## Stirrups

**26 stirrups over whole span length, s175**

$$A_{s.wholespan} := 2 \cdot 26 \cdot A_{s.i.v} = 16336.282 \text{ mm}^2$$

**12 stirrups outside span, s300**

$$A_{s.outside} := 2 \cdot 12 \cdot A_{s.i.v} = 7539.822 \text{ mm}^2$$

**26 stirrups in span close to supports, s100**

$$A_{s.between} := 2 \cdot 26 \cdot A_{s.i.v} = 16336.282 \text{ mm}^2$$

**Distribution on the beam's deep will have the same areas as above**

## Lengths

$$V := d - e = 950 \text{ mm}$$

$$H_{long} := b - 100 \text{ mm} = 1450 \text{ mm}$$

$$H_{short} := H_{long} \cdot 0.6 = 870 \text{ mm}$$

## Volume stirrups

$$V_{v.wholespan} := A_{s.wholespan} \cdot V = 15519467.709 \text{ mm}^3$$

$$V_{v.outside} := A_{s.outside} \cdot V = 7162831.25 \text{ mm}^3$$

$$V_{v.between} := A_{s.between} \cdot V = 15519467.709 \text{ mm}^3$$

$$V_{h.wholespan} := A_{s.wholespan} \cdot H_{long} = 23687608.608 \text{ mm}^3$$

$$V_{h.outside} := A_{s.outside} \cdot H_{long} = 10932742.434 \text{ mm}^3$$

$$V_{h.between} := A_{s.between} \cdot H_{short} = 14212565.165 \text{ mm}^3$$

$$V_{stirr.tot} := V_{v.wholespan} + V_{v.outside} + V_{v.between} + V_{h.wholespan} + V_{h.outside} + V_{h.between} = 0.087 \text{ m}^3$$

## Horizontal reinforcement

**Distance AB**  $AB_h := 3764 \text{ mm}$

**Distance C**  $C_h := 4932 \text{ mm}$

**Distance D**  $D_h := 7900 \text{ mm}$

### Need on lenth A:

$$A_{s.T4} := \frac{T_4}{f_{st}} = 13406.445 \text{ mm}^2$$

$$n_{T4} := \frac{A_{s.T4}}{A_{s.i.h}} = 27.311 \quad n_{T4} := \text{ceil}(n_{T4}) = 28$$

$$A_{s.T4.real} := n_{T4} \cdot A_{s.i.h} = 13744.468 \text{ mm}^2$$

### Need on lenth B:

$$A_{s.T5} := \frac{T_5}{f_{st}} = 13406.445 \text{ mm}^2$$

$$n_{T5} := \frac{A_{s.T5}}{A_{s.i.h}} = 27.311 \quad n_{T5} := \text{ceil}(n_{T5}) = 28$$

$$A_{s.T5.real} := n_{T5} \cdot A_{s.i.h} = 13744.468 \text{ mm}^2$$

### Need on lenth C:

$$A_{s.T6} := \frac{T_6}{f_{st}} = 11007.013 \text{ mm}^2$$

$$n_{T6} := \frac{A_{s.T6}}{A_{s.i.h}} = 22.423 \quad n_{T6} := \text{ceil}(n_{T6}) = 23$$

$$A_{s.T6.real} := n_{T6} \cdot A_{s.i.h} = 11290.099 \text{ mm}^2$$

### Need on lenth D:

$$A_{s.T7} := \frac{T_7}{f_{st}} = 6833.825 \text{ mm}^2$$

$$n_{T7} := \frac{A_{s.T7}}{A_{s.i.h}} = 13.922 \quad n_{T7} := \text{ceil}(n_{T7}) = 14$$

$$A_{s.T7.real} := n_{T7} \cdot A_{s.i.h} = 6872.234 \text{ mm}^2$$

### Volume of horisontal reinforcement

$$V_D := A_{s.i.h} \cdot n_{T7} \cdot D_h = 54290648.045 \text{ mm}^3$$

$$V_C := A_{s.i.h} \cdot (n_{T6} - n_{T7}) \cdot C_h = 21788908.548 \text{ mm}^3$$

$$V_{AB} := A_{s.i.h} \cdot (n_{T5} - n_{T6}) \cdot AB_h = 9238245.897 \text{ mm}^3$$

$$V_{h.tot} := V_D + V_C + V_{AB} = 0.085 \text{ m}^3$$

### Total volume of reinforcement

$$V_{tot.rein} := V_{h.tot} + V_{stirr.tot} = 0.172 \text{ m}^3$$

## A.2 Hand calculations on Model B

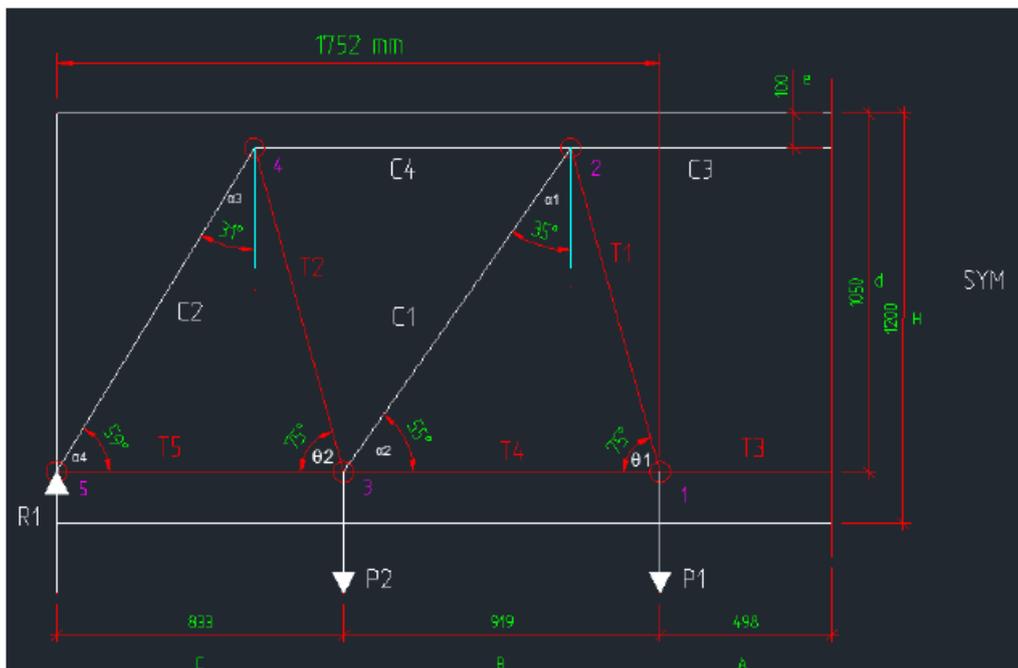
### Geometry

$H := 1200 \text{ mm}$  Height of the beam  
 $b := 1550 \text{ mm}$  Width of the beam  
 $L := 4500 \text{ mm}$  Span length

### Strut and tie model

$A := 833 \text{ mm}$   
 $B := 919 \text{ mm}$   
 $C := 498 \text{ mm}$

$d := 1050 \text{ mm}$   
 $e := 100 \text{ mm}$



## Angles

$$\theta_1 := 75 \text{ deg}$$

$$\theta_2 := \theta_1 = 75 \text{ deg}$$

$$\alpha_1 := 35 \text{ deg}$$

$$\alpha_2 := 55 \text{ deg}$$

$$\alpha_3 := 31 \text{ deg}$$

$$\alpha_4 := 59 \text{ deg}$$

## Loads

$$P_{Ed} := 6670 \text{ kN}$$

$$R_1 := \frac{P_{Ed}}{2} = 3335 \text{ kN}$$

$$q_{Ed} := \frac{P_{Ed}}{L} = 1.482 \frac{\text{kN}}{\text{mm}}$$

$$P_1 := q_{Ed} \cdot \left( A + \frac{B}{2} \right) = 1915.772 \text{ kN}$$

$$P_2 := q_{Ed} \cdot \left( \frac{B}{2} + C \right) = 1419.228 \text{ kN}$$

$$P := P_1 + P_2 = 3335 \text{ kN}$$

OK

## Forces in the truss

### Node 5

$$C_2 := \frac{R_1}{\sin(\alpha_4)} = 3890.722 \text{ kN}$$

$$T_5 := C_2 \cdot \cos(\alpha_4) = 2003.87 \text{ kN}$$

**Node 4**

$$T_2 := \frac{C_2 \cdot \cos(\alpha_3)}{\sin(\theta_2)} = 3452.646 \text{ kN}$$

$$C_4 := T_2 \cdot \cos(\theta_2) + C_2 \cdot \sin(\alpha_3) = 2897.481 \text{ kN}$$

**Node 3**

$$C_1 := \frac{T_2 \cdot \sin(\theta_2) - P_2}{\sin(\alpha_2)} = 2338.726 \text{ kN}$$

$$T_4 := C_1 \cdot \cos(\alpha_2) + T_2 \cdot \cos(\theta_2) + T_5 = 4238.919 \text{ kN}$$

**Node 2**

$$T_1 := \frac{C_1 \cdot \cos(\alpha_1)}{\sin(\theta_1)} = 1983.353 \text{ kN}$$

$$C_3 := C_4 + C_1 \cdot \sin(\alpha_1) + T_1 \cdot \cos(\theta_1) = 4752.248 \text{ kN}$$

**Node 1**

$$T_3 := T_1 \cdot \cos(\theta_1) + T_4 = 4752.248 \text{ kN}$$

$$C_3 = 4752.248 \text{ kN}$$

**OK**

## Check of nodes

Compression - tension node with anchored ties in one direction and where the the bars are anchored within the node.

$$f_{ck} := 35 \text{ MPa}$$

$$f_{cd} := \frac{f_{ck}}{1.5} = 23.333 \text{ MPa}$$

$$k_2 := 0.85$$

$$\nu := 1 - \frac{f_{ck}}{250 \text{ MPa}} = 0.86$$

$$\sigma_{rd,max} := k_2 \cdot \nu \cdot f_{cd} = 17.057 \text{ MPa}$$

## Control of node 5

$$u := 2 \cdot (H - d) = 300 \text{ mm}$$

$$a_1 := 400 \text{ mm}$$

$$b_1 := 600 \text{ mm}$$

$$a_2 := a_1 \cdot \sin(\alpha_4) + u \cdot \cos(\alpha_4) = 497.378 \text{ mm}$$

$$b_2 := b_1 = 600 \text{ mm}$$

$$\sigma_{d1.5} := \frac{R_1}{a_1 \cdot b_1} = 13.896 \text{ MPa} \quad \text{OK}$$

$$\sigma_{d2.5} := \frac{C_2}{a_2 \cdot b_2} = 13.037 \text{ MPa} \quad \text{OK}$$

## Control of node 2

$$a_1 := 2 \cdot e = 200 \text{ mm}$$

$$b_1 := b = 1550 \text{ mm}$$

$$\sigma_{d1.2} := \frac{C_3}{a_1 \cdot b_1} = 15.33 \text{ MPa} \quad \text{OK}$$

## Design of ties

$$f_{st} := 300 \text{ MPa}$$

$$f_{st.lift} := 435 \text{ MPa}$$

Reinforcement equal to T1 is used on the length

$$A + \frac{B}{2} = 1292.5 \text{ mm}$$

## With an angle of 75 deg

$$A_{sT1} := \frac{T_1}{f_{st.lift}} = 4559.433 \text{ mm}^2 \quad A_{siT1} := 201 \text{ mm}^2 \quad \mathbf{d=20mm}$$

$$n_{T1} := \frac{A_{sT1}}{A_{siT1}} = 22.684 \quad n_{T1} := 23$$

Reinforcement equal to T2 is used on the length

$$\frac{B}{2} + C = 957.5 \text{ mm}$$

## With an angle of 75 deg

$$A_{sT2} := \frac{T_2}{f_{st.lift}} = 7937.117 \text{ mm}^2 \quad A_{siT2} := 201 \text{ mm}^2 \quad \mathbf{d=20mm}$$

$$n_{T2} := \frac{A_{sT2}}{A_{siT2}} = 39.488 \quad n_{T2} := 40$$

Total amount of lifting reinforcement:

$$A_{lift} := 2 \cdot A_{sT1} + 2 \cdot A_{sT2} = 24993.101 \text{ mm}^2$$

### Horizontal reinforcement

$$A_{sT3} := \frac{T_3}{f_{st}} = 15840.828 \text{ mm}^2 \quad A_{siT3} := 314 \text{ mm}^2 \quad \mathbf{d=25mm}$$

$$n_{T3} := \frac{A_{sT3}}{A_{siT3}} = 50.448 \quad n_{T3} := 49$$

**Total amount of reinforcement:**

$$A_{lift} = 24993.101 \text{ mm}^2$$

$$A_{horizontal} := A_{sT3} = 15840.828 \text{ mm}^2$$

$$A_{tot} := A_{lift} + A_{horizontal} = 40833.929 \text{ mm}^2$$

## Reinforcement design

### Bars

$$\phi := 25 \text{ mm}$$

$$A_{s.i.h} := \frac{\pi \cdot \phi^2}{4} = 490.874 \text{ mm}^2$$

$$\phi_{vertical} := 20 \text{ mm}$$

$$A_{s.i.v} := \frac{\pi \cdot \phi_{vertical}^2}{4} = 314.159 \text{ mm}^2$$

### Distances

$$A_{rein} := 996 \text{ mm}$$

$$B_{rein} := 2834 \text{ mm}$$

$$C_{rein} := 7900 \text{ mm}$$

### Need on distance tie T3

$$A_{s.T3} := \frac{T_3}{f_{st}} = 15840.828 \text{ mm}^2$$

$$n_{T3} := \frac{A_{s.T3}}{A_{s.i.h}} = 32.271$$

$$n_{T3} := \text{ceil}(n_{T3}) = 33 \quad \text{Need}$$

$$n_{T3} := 41 \quad \text{To satisfy need in T5}$$

### Need for the inclined tie T1

$$A_{s.T1} := \frac{T_1}{f_{st.lift}} = 4559.433 \text{ mm}^2$$

$$n_{T1} := \frac{A_{s.T1}}{A_{s.i.h}} = 9.288$$

$$n_{T1} := \text{ceil}(n_{T1}) = 10$$

Need on distance tie T4

$$A_{s.T4} := \frac{T_4}{f_{st}} = 14129.73 \text{ mm}^2$$

$$n_{T4} := \frac{A_{s.T4}}{A_{s.i.h}} = 28.785$$

$$n_{T4} := \text{ceil}(n_{T4}) = 29$$

Need

$$n_{T4} := n_{T3} - n_{T1} = 31$$

To satisfy need in T5

Need for the inclined tie T2

$$A_{s.T2} := \frac{T_2}{f_{st.lift}} = 7937.117 \text{ mm}^2$$

$$n_{T2} := \frac{A_{s.T2}}{A_{s.i.h}} = 16.169$$

$$n_{T2} := \text{ceil}(n_{T2}) = 17$$

Need on distance tie T5

$$A_{s.T5} := \frac{T_5}{f_{st}} = 6679.567 \text{ mm}^2$$

$$n_{T5} := \frac{A_{s.T5}}{A_{s.i.h}} = 13.608$$

$$n_{T5} := \text{ceil}(n_{T5}) = 14$$

Need

$$n_{T5} := n_{T4} - n_{T2} = 14$$

To satisfy need in T5

Volume

Bottom layer

Distance C

$$V_{h.C} := n_{T5} \cdot A_{s.i.h} \cdot C_{rein} = 54290648.045 \text{ mm}^3$$

Distance B

$$V_{h.B} := (n_{T2}) \cdot A_{s.i.h} \cdot B_{rein} = 23649320.448 \text{ mm}^3$$

Distance A

$$V_{h.A} := n_{T1} \cdot A_{s.i.h} \cdot A_{rein} = 4889103.567 \text{ mm}^3$$

Total horizaontal reinforcement

$$V_{h.tot} := V_{h.A} + V_{h.B} + V_{h.C} = 0.083 \text{ m}^3$$

Inclined reinforcement

Distances inclined ties plus förankringslängd

$$d_T := 985 \text{ mm} + 800 \text{ mm} = 1.785 \text{ m}$$

Volumes

$$V_{i.T1} := A_{s.i.h} \cdot n_{T1} \cdot d_T = 8762098.26 \text{ mm}^3$$

$$V_{i.T2} := A_{s.i.h} \cdot n_{T2} \cdot d_T = 14895567.043 \text{ mm}^3$$

$$V_{i.tot} := 2 \cdot (V_{i.T1} + V_{i.T2}) = 0.047 \text{ m}^3$$

Both sides (\*2)

Stirrups for minimum reinforcement

$$n_{stirr} := 26$$

$$h_{stirr} := d - e = 950 \text{ mm}$$

$$b_{stirr} := b - 100 \text{ mm} = 1450 \text{ mm}$$

$$A_{s.stirr} := n_{stirr} \cdot A_{s.i.v} = 8168.141 \text{ mm}^2$$

$$V_{stirr} := A_{s.stirr} \cdot (2 \cdot h_{stirr} + 2 \cdot b_{stirr}) = 0.039 \text{ m}^3$$

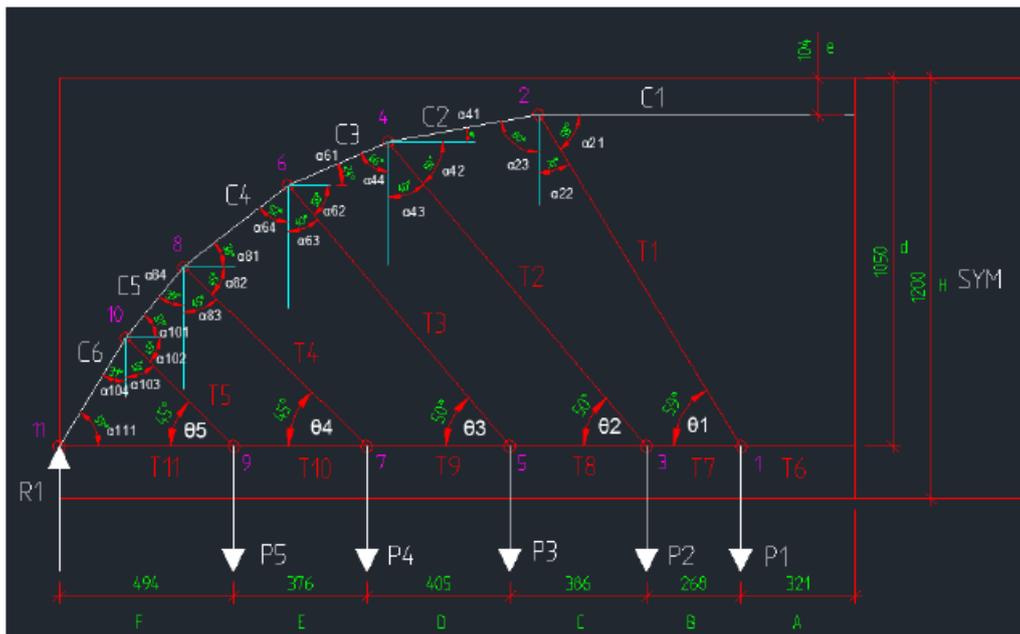
Total reinforcement

$$V_{rein.tot} := V_{stirr} + V_{i.tot} + V_{h.tot} = 0.169 \text{ m}^3$$

## A.3 Hand calculations on Model C

### Geometry

$H := 1200 \text{ mm}$                       Height of the beam  
 $b := 1550 \text{ mm}$                       Width of the beam  
 $L := 4500 \text{ mm}$                       Span length



### Strut and tie model

$A := 321 \text{ mm}$   
 $B := 268 \text{ mm}$   
 $C := 386 \text{ mm}$   
 $D := 405 \text{ mm}$   
 $E := 376 \text{ mm}$   
 $F := 494 \text{ mm}$

$d := 1050 \text{ mm}$   
 $e := 104 \text{ mm}$

## Angles

$$\theta_1 := 59 \text{ deg}$$

$$\theta_2 := 50 \text{ deg}$$

$$\theta_3 := 50 \text{ deg}$$

$$\theta_4 := 45 \text{ deg}$$

$$\theta_5 := 45 \text{ deg}$$

$$\alpha_{21} := 59 \text{ deg}$$

$$\alpha_{22} := 31 \text{ deg}$$

$$\alpha_{23} := 80 \text{ deg}$$

$$\alpha_{41} := 10 \text{ deg}$$

$$\alpha_{42} := 50 \text{ deg}$$

$$\alpha_{43} := 40 \text{ deg}$$

$$\alpha_{44} := 66 \text{ deg}$$

$$\alpha_{61} := 24 \text{ deg}$$

$$\alpha_{62} := 50 \text{ deg}$$

$$\alpha_{63} := 40 \text{ deg}$$

$$\alpha_{64} := 52 \text{ deg}$$

$$\alpha_{81} := 38 \text{ deg}$$

$$\alpha_{82} := 45 \text{ deg}$$

$$\alpha_{83} := 45 \text{ deg}$$

$$\alpha_{84} := 39 \text{ deg}$$

$$\alpha_{101} := 51 \text{ deg}$$

$$\alpha_{102} := 45 \text{ deg}$$

$$\alpha_{103} := 45 \text{ deg}$$

$$\alpha_{104} := 31 \text{ deg}$$

$$\alpha_{111} := 59 \text{ deg}$$

## Loads

$$P_{Ed} := 6670 \text{ kN}$$

$$q_{Ed} := \frac{P_{Ed}}{L} = 1482.222 \frac{\text{N}}{\text{mm}}$$

$$R_1 := \frac{P_{Ed}}{2} = 3335 \text{ kN}$$

$$P_1 := q_{Ed} \cdot \left( A + \frac{B}{2} \right) = 674.411 \text{ kN}$$

$$P_2 := q_{Ed} \cdot \left( \frac{B}{2} + \frac{C}{2} \right) = 484.687 \text{ kN}$$

$$P_3 := q_{Ed} \cdot \left( \frac{C}{2} + \frac{D}{2} \right) = 586.219 \text{ kN}$$

$$P_4 := q_{Ed} \cdot \left( \frac{D}{2} + \frac{E}{2} \right) = 578.808 \text{ kN}$$

$$P_5 := q_{Ed} \cdot \left( \frac{E}{2} + F \right) = 1010.876 \text{ kN}$$

$$P := P_1 + P_2 + P_3 + P_4 + P_5 = 3335 \text{ kN}$$

OK

### Forces in the truss

$$T_1 := \frac{P_1}{\sin(\theta_1)} = 786.791 \text{ kN}$$

$$T_2 := \frac{P_2}{\sin(\theta_2)} = 632.714 \text{ kN}$$

$$T_3 := \frac{P_3}{\sin(\theta_3)} = 765.254 \text{ kN}$$

$$T_4 := \frac{P_4}{\sin(\theta_4)} = 818.558 \text{ kN}$$

$$T_5 := \frac{P_5}{\sin(\theta_5)} = 1429.594 \text{ kN}$$

$$T_{tot} := T_1 + T_2 + T_3 + T_4 + T_5 = 4432.91 \text{ kN}$$

Starting from node 11, using x-components in every node except node 11

Node 11:

$$C_6 := \frac{R_1}{\sin(\alpha_{111})} = 3890.722 \text{ kN}$$

$$T_{11} := C_6 \cdot \cos(\alpha_{111}) = 2003.87 \text{ kN}$$

Node 10:

$$C_5 := \frac{C_6 \cdot \cos(\alpha_{111}) + T_5 \cdot \cos(\alpha_{102})}{\cos(\alpha_{101})} = 4790.478 \text{ kN}$$

Node 8:

$$C_4 := \frac{C_5 \cdot \sin(\alpha_{84}) + T_4 \cdot \cos(\alpha_{82})}{\cos(\alpha_{81})} = 4560.285 \text{ kN}$$

Node 6:

$$C_3 := \frac{C_4 \cdot \sin(\alpha_{64}) + T_3 \cdot \cos(\alpha_{62})}{\cos(\alpha_{61})} = 4472.081 \text{ kN}$$

Node 4:

$$C_2 := \frac{C_3 \cdot \sin(\alpha_{44}) + T_2 \cdot \cos(\alpha_{42})}{\cos(\alpha_{41})} = 4561.449 \text{ kN}$$

Node 2:

$$C_1 := C_2 \cdot \sin(\alpha_{23}) + T_1 \cdot \cos(\alpha_{21}) = 4897.377 \text{ kN}$$

Node 9:

$$T_{10} := T_{11} + T_5 \cdot \cos(\theta_5) = 3014.746 \text{ kN}$$

Node 7:

$$T_9 := T_{10} + T_4 \cdot \cos(\theta_4) = 3593.553 \text{ kN}$$

Node 5:

$$T_8 := T_9 + T_3 \cdot \cos(\theta_3) = 4085.45 \text{ kN}$$

Node 3:

$$T_7 := T_8 + T_2 \cdot \cos(\theta_2) = 4492.15 \text{ kN}$$

Node 1:

$$T_6 := T_7 + T_1 \cdot \cos(\theta_1) = 4897.377 \text{ kN}$$

Compare:

$$C_1 = 4897.377 \text{ kN}$$

OK

Lifting reinforcement:

$$T_{lift} := T_1 + T_2 + T_3 + T_4 + T_5 = 4432.91 \text{ kN}$$

## Check of nodes

$$f_{ck} := 35 \text{ MPa}$$

$$f_{cd} := \frac{f_{ck}}{1.5} = 23.333 \text{ MPa}$$

$$k_2 := 0.85$$

$$\nu := 1 - \frac{f_{ck}}{250 \text{ MPa}} = 0.86$$

$$\sigma_{rd,max} := k_2 \cdot \nu \cdot f_{cd} = 17.057 \text{ MPa}$$

## Control of node 11

$$u := 2 \cdot (H - d) = 300 \text{ mm}$$

$$a_1 := 600 \text{ mm}$$

$$b_1 := 400 \text{ mm}$$

$$a_2 := a_1 \cdot \sin(\alpha_{111}) + u \cdot \cos(\alpha_{111}) = 0.669 \text{ m}$$

$$b_2 := b_1 = 0.4 \text{ m}$$

$$\sigma_{d1.9} := \frac{R_1}{a_1 \cdot b_1} = 13.896 \text{ MPa}$$

$$\sigma_{d2.9} := \frac{C_6}{a_2 \cdot b_2} = 14.543 \text{ MPa}$$

Both controls OK

## Control of node 2

$$a_1 := 2 \cdot e = 208 \text{ mm}$$

$$b_1 := b = 1550 \text{ mm}$$

$$\sigma_{d1.2} := \frac{C_1}{a_1 \cdot b_1} = 15.19 \text{ MPa}$$

OK

## Design of ties

$$f_{st} := 300 \text{ MPa}$$

$$f_{st.lift} := 435 \text{ MPa}$$

## Inclined reinforcement:

Reinforcement equal to T1 is used on the length

$$A + \frac{B}{2} = 455 \text{ mm}$$

With an angle of 59 deg

$$A_{sT1} := \frac{T_1}{f_{st.lift}} = 1808.714 \text{ mm}^2$$

$$A_{siT1} := 314 \text{ mm}^2 \quad \mathbf{d=20mm}$$

$$n_{T1} := \frac{A_{sT1}}{A_{siT1}} = 5.76 \quad n_{T1} := 6$$

**Number of stirrups**

Reinforcement equal to T2 is used on the length

$$\frac{B}{2} + \frac{C}{2} = 327 \text{ mm}$$

With an angle of 50 deg

$$A_{sT2} := \frac{T_2}{f_{st.lift}} = 1454.514 \text{ mm}^2$$

$$A_{siT2} := 314 \text{ mm}^2$$

$$n_{T2} := \frac{A_{sT2}}{A_{siT2}} = 4.632 \quad n_{T2} := 5$$

Reinforcement equal to T3 is used on the length

$$\frac{C}{2} + \frac{D}{2} = 395.5 \text{ mm}$$

With an angle of 50 deg

$$A_{sT3} := \frac{T_3}{f_{st.lift}} = 1759.206 \text{ mm}^2 \quad A_{siT3} := 314 \text{ mm}^2$$

$$n_{T3} := \frac{A_{sT3}}{A_{siT3}} = 5.603 \quad n_{T3} := 6$$

Reinforcement equal to T4 is used on the length

$$\frac{D}{2} + \frac{E}{2} = 390.5 \text{ mm}$$

With an angle of 45 deg

$$A_{sT4} := \frac{T_4}{f_{st.lift}} = 1881.742 \text{ mm}^2 \quad A_{siT4} := 314 \text{ mm}^2$$

$$n_{T4} := \frac{A_{sT4}}{A_{siT4}} = 5.993 \quad n_{T4} := 6$$

Reinforcement equal to T5 is used on the length

$$\frac{E}{2} + F = 682 \text{ mm}$$

With an angle of 45 deg

$$A_{sT5} := \frac{T_5}{f_{st.lift}} = 3286.423 \text{ mm}^2 \quad A_{siT5} := 314 \text{ mm}^2$$

$$n_{T5} := \frac{A_{sT5}}{A_{siT5}} = 10.466 \quad n_{T5} := 11$$

Total amount of lifting reinforcement:

$$A_{lift} := 2 \cdot A_{sT1} + 2 \cdot A_{sT2} + 2 \cdot A_{sT3} + 2 \cdot A_{sT4} + 2 \cdot A_{sT5} = 20381.196 \text{ mm}^2$$

## Horizontal reinforcement

$$A_{sT6} := \frac{T_6}{f_{st}} = 16324.59 \text{ mm}^2$$

$$A_{siT6} := 491 \text{ mm}^2$$

$$d=25\text{mm}$$

$$n_{T5} := \frac{A_{sT6}}{A_{siT6}} = 33.248 \quad n_{T6} := 34$$

## Total amount of reinforcement

$$A_{lift} = 20381.196 \text{ mm}^2$$

$$A_{horizontal} := A_{sT6} = 16324.59 \text{ mm}^2$$

$$A_{tot} := A_{lift} + A_{horizontal} = 36705.786 \text{ mm}^2$$

## Reinforcement design

### Bars

$$\phi := 25 \text{ mm}$$

$$A_{s.i.h} := \frac{\pi \cdot \phi^2}{4} = 490.874 \text{ mm}^2$$

$$\phi_{vertical} := 20 \text{ mm}$$

$$A_{s.i.v} := \frac{\pi \cdot \phi_{vertical}^2}{4} = 314.159 \text{ mm}^2$$

$$\phi_{bmin} := 16 \text{ mm}$$

$$A_{s.i.bmin} := \frac{\pi \cdot \phi_{bmin}^2}{4} = 201.062 \text{ mm}^2$$

Distances

$$A_{rein} := 641 \text{ mm}$$

$$D_{rein} := 2759 \text{ mm}$$

$$B_{rein} := 1177 \text{ mm}$$

$$E_{rein} := 3512 \text{ mm}$$

$$C_{rein} := 1950 \text{ mm}$$

$$F_{rein} := 7900 \text{ mm}$$

Need on distance tie T6

$$A_{s.T6} := \frac{T_6}{f_{st}} = 16324.59 \text{ mm}^2$$

$$n_{T6} := \frac{A_{s.T6}}{A_{s.i.h}} = 33.256$$

$$n_{T6} := \text{ceil}(n_{T6}) = 34 \quad \text{Need}$$

$$n_{T6} := 36 \quad \text{To satisfy need in T11}$$

Need for the inclined tie T1

$$A_{s.T1} := \frac{T_1}{f_{st.lift}} = 1808.714 \text{ mm}^2$$

$$n_{T1} := \frac{A_{s.T1}}{A_{s.i.h}} = 3.685$$

$$n_{T1} := \text{ceil}(n_{T1}) = 4$$

Need on distance tie T7

$$A_{s.T7} := \frac{T_7}{f_{st}} = 14973.833 \text{ mm}^2$$

$$n_{T7} := \frac{A_{s.T7}}{A_{s.i.h}} = 30.504$$

$$n_{T7} := \text{ceil}(n_{T7}) = 31 \quad \text{Need}$$

$$n_{T7} := n_{T6} - n_{T1} = 32 \quad \text{To satisfy need in T11}$$

Need for the inclined tie T2

$$A_{s,T2} := \frac{T_2}{f_{st.lift}} = 1454.514 \text{ mm}^2$$

$$n_{T2} := \frac{A_{s,T2}}{A_{s.i.h}} = 2.963$$

$$n_{T2} := \text{ceil}(n_{T2}) = 3$$

Need on distance tie T8

$$A_{s,T8} := \frac{T_8}{f_{st}} = 13618.165 \text{ mm}^2$$

$$n_{T8} := \frac{A_{s,T8}}{A_{s.i.h}} = 27.743$$

$$n_{T8} := \text{ceil}(n_{T8}) = 28$$

Need

$$n_{T8} := n_{T7} - n_{T2} = 29$$

To satisfy need in T11

Need for the inclined tie T3

$$A_{s,T3} := \frac{T_3}{f_{st.lift}} = 1759.206 \text{ mm}^2$$

$$n_{T3} := \frac{A_{s,T3}}{A_{s.i.h}} = 3.584$$

$$n_{T3} := \text{ceil}(n_{T3}) = 4$$

Need on distance tie T9

$$A_{s,T9} := \frac{T_9}{f_{st}} = 11978.512 \text{ mm}^2$$

$$n_{T9} := \frac{A_{s,T9}}{A_{s.i.h}} = 24.402$$

$$n_{T9} := \text{ceil}(n_{T9}) = 25$$

Need

$$n_{T9} := n_{T8} - n_{T3} = 25$$

To satisfy need in T11

Need for the inclined tie T4

$$A_{s.T4} := \frac{T_4}{f_{st.lift}} = 1881.742 \text{ mm}^2$$

$$n_{T4} := \frac{A_{s.T4}}{A_{s.i.h}} = 3.833$$

$$n_{T4} := \text{ceil}(n_{T4}) = 4$$

Need on distance tie T10

$$A_{s.T10} := \frac{T_{10}}{f_{st}} = 10049.152 \text{ mm}^2$$

$$n_{T10} := \frac{A_{s.T10}}{A_{s.i.h}} = 20.472$$

$$n_{T10} := \text{ceil}(n_{T10}) = 21$$

Need

$$n_{T10} := n_{T9} - n_{T4} = 21$$

To satisfy need in T11

Need for the inclined tie T5

$$A_{s.T5} := \frac{T_5}{f_{st.lift}} = 3286.423 \text{ mm}^2$$

$$n_{T5} := \frac{A_{s.T5}}{A_{s.i.h}} = 6.695$$

$$n_{T5} := \text{ceil}(n_{T5}) = 7$$

Need on distance tie T11

$$A_{s.T11} := \frac{T_{11}}{f_{st}} = 6679.567 \text{ mm}^2$$

$$n_{T11} := \frac{A_{s.T11}}{A_{s.i.h}} = 13.608$$

$$n_{T11} := \text{ceil}(n_{T11}) = 14$$

Need

$$n_{T11} := n_{T10} - n_{T5} = 14$$

To satisfy need in T11

Volume

Bottom layer

Distance F

$$V_{h.F} := n_{T11} \cdot A_{s.i.h} \cdot F_{rein} = 54290648.045 \text{ mm}^3$$

Distance E

$$V_{h.E} := (n_{T5}) \cdot A_{s.i.h} \cdot E_{rein} = 12067642.781 \text{ mm}^3$$

Distance D

$$V_{h.D} := n_{T4} \cdot A_{s.i.h} \cdot D_{rein} = 5417283.832 \text{ mm}^3$$

Distance C

$$V_{h.C} := n_{T3} \cdot A_{s.i.h} \cdot C_{rein} = 3828816.047 \text{ mm}^3$$

Distance B

$$V_{h.B} := n_{T2} \cdot A_{s.i.h} \cdot B_{rein} = 1733275.572 \text{ mm}^3$$

Distance A

$$V_{h.A} := n_{T1} \cdot A_{s.i.h} \cdot A_{rein} = 1258600.557 \text{ mm}^3$$

Total horizaontal reinforcement

$$V_{h.tot} := V_{h.A} + V_{h.B} + V_{h.C} + V_{h.D} + V_{h.E} + V_{h.F} = 0.079 \text{ m}^3$$

Inclined reinforcement

Distances inclined ties plus förankringslängd

$$d_{T1} := 1106 \text{ mm} + 800 \text{ mm} = 1.906 \text{ m}$$

$$d_{T2} := 1134 \text{ mm} + 800 \text{ mm} = 1.934 \text{ m}$$

$$d_{T3} := 973 \text{ mm} + 800 \text{ mm} = 1.773 \text{ m}$$

$$d_{T4} := 728 \text{ mm} + 800 \text{ mm} = 1.528 \text{ m}$$

$$d_{T5} := 435 \text{ mm} + 800 \text{ mm} = 1.235 \text{ m}$$

Volumes

$$V_{i.T1} := A_{s.i.h} \cdot n_{T1} \cdot d_{T1} = 3742422.249 \text{ mm}^3$$

$$V_{i.T2} := A_{s.i.h} \cdot n_{T2} \cdot d_{T2} = 2848050.09 \text{ mm}^3$$

$$V_{i.T3} := A_{s.i.h} \cdot n_{T3} \cdot d_{T3} = 3481277.359 \text{ mm}^3$$

$$V_{i.T4} := A_{s.i.h} \cdot n_{T4} \cdot d_{T4} = 3000220.984 \text{ mm}^3$$

$$V_{i.T5} := A_{s.i.h} \cdot n_{T5} \cdot d_{T5} = 4243604.452 \text{ mm}^3$$

$$V_{i.tot} := 2 \cdot (V_{i.T1} + V_{i.T2} + V_{i.T3} + V_{i.T4} + V_{i.T5}) = 0.035 \text{ m}^3 \quad \text{Both sides (*2)}$$

Stirrups for minimum reinforcement

$$n_{stirr} := 13$$

$$h_{stirr} := d - e = 946 \text{ mm}$$

$$b_{stirr} := b - 100 \text{ mm} = 1450 \text{ mm}$$

$$A_{s.stirr} := n_{stirr} \cdot A_{s.i.v} = 4084.07 \text{ mm}^2$$

$$V_{stirr} := A_{s.stirr} \cdot (2 \cdot h_{stirr} + 2 \cdot b_{stirr}) = 0.02 \text{ m}^3$$