Norwegian University
of Life Sciences

Master's Thesis 2022 30 ECTS
Faculty of Science and Technology (REALTEK)

Validation of Serviceability Limit State Calculation Models for Timber Shear Walls in Multi-Storey Buildings.

Johan Bjørkedal
Håkon Østraat Sævareid
Structural Engineering and Architecture

Acknowledgements

This thesis concludes two years of education at the Norwegian University of Life Science (NMBU), achieving a master's degree in Structural Engineering. Firstly, we would like to give a big thank you to our main supervisor, Roberto Tomasi, as well as our external supervisor Francesco Boggian for great help and engagement on this thesis. Furthermore, we would like to thank Dag Pasca and Angelo Aloisio for the guidance with modal updating and the interpretation of experimental data. Lastly, thanks to the InnoRenew CoE for providing the experimental test.

Ås, May 2022
Johan Bjørkedal
Håkon Østraat Sævareid

Abstract

This thesis presents a parametric analysis of LTF and CLT, comparing numerical and analytical calculation models and researching the significance of the different contributions presented in a draft for the next Eurocode 5. On top of this, the thesis explores the validity of a numerical modal analysis of the InnoRenew CoE new headquarters in comparison with an experimental campaign executed on the building. Dynamic identification and finite element model updating is performed.

The parametric analysis of single shear walls analysis utilizes Open Application Programming Interface in Python to manipulate the Finite Element Method software SAP2000 to perform a series of analyses. The modelling is executed in a manner that allows extraction of results for single contributions in the numerical model and compare the results with the analytical equations for the same contributions presented in the draft for the next Eurocode 5.

The Finite Element model updating is based on an experimental campaign and a SAP2000 model of InnoRenew CoE new headquarters. The Finite Element model is updated using a differential evolution where the masses in different parts of the building are modified to optimize the numerical calculations.

Overall, the parametric analysis of single shear walls generated results in the numerical and analytical calculations that were quite similar. The contributions that presented results where the numerical and analytical models deviated, the deviation can be explained.

The Finite Element Model updating provided a significant improvement to the Total Convergence Criterium. However, due to excessive differences between initial dynamic identification and the experimental values, improvements were not sufficient. Therefore, the Total Convergence Criterium cannot be said to validate the model sufficiently.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The use of timber shear walls in larger buildings is constantly increasing, and so the need for accurate and reliable design models. Analytical models for the Serviceability Limit State (SLS), both for Light Timber Frame (LTF) and Cross Laminated Timber (CLT) shear walls are currently in development for the next Eurocode 5, set to be published with the upcoming generation of Eurocodes in the near future. (CEN, 2021)

Light timber frames are today one of the most common construction systems for residential houses. Particularly common in countries with a Nordic climate (Grossi et al., 2015). Cross laminated timber is a relative new timber product, and not as widely used as LTF in timber houses and buildings. However, the product has become of global interest and a well-recognized timber product among engineers (Brandner et al., 2016). Both CLT and LTF shear walls are of great interest for use in multi-storey buildings.

With no European standard related to the design of these shear walls yet published, there are quite few buildings built with LTF and CLT shear walls as its main structural components. As a part of the development of new standardization, validation of performance of different calculation models, both statically and dynamic is an important step towards greater use of LTF and CLT shear walls in multi-storey buildings.

1.2 State of the Art

Performance of LTF and CLT shear walls under different load in the Serviceability Limit State (SLS) is an open problem. The Technical Committee CEN (2021) are currently developing a new Eurocode for timber structures which will have the inclusion of analytical equations for calculating the lateral deflection of these shear walls. (CEN, 2021, CEN, 2022). Both a draft for the complete Eurocode 5 (CEN, 2021) and a draft for a chapter 13.7, including analytical equations for lateral deflections of timber shear walls are produced (CEN, 2022).

Multiple studies on LTF shear walls have already been conducted, including the studies by Ricci (2020) and Boggian et al. (2021). In these studies, the lateral deflection of LTF shear walls were researched, and design methods examined. The study by Ricci (2020) addresses multiple experimental campaigns done on LTF shear walls, as well as an older proposal for the next upcoming Eurocode 5, including analytical calculation models for ULS and SLS for LTF shear walls. These analytical equations are equal or similar with minor differences to the equations presented in CEN (2021) and CEN (2022). Finite element models in SAP2000 were designed to do comparisons between both experimental campaigns and analytical models for lateral deflections. It was found that the numerical models in SAP2000 sufficiently represented the real-life testing from the different experimental campaigns, and that the results from the comparison between analytical and numerical model were quite similar, especially for low vertical distributed loads. Therefore, a conclusion that the analytical and numerical models represent the behaviour of shear walls tested in a laboratory sufficiently. However, it was suggested to do more investigation on comparison between analytical and numerical models, as only one wall configuration was verified. Testing multiple wall configurations might confirm the conclusions made. (Ricci, 2020)

The conference paper of Boggian et al. (2021) mostly is based around the studies conducted by Ricci (2020). However, the lateral deflection due to deformation of sheathing to framing fasteners are discussed in more detail. Two approaches of calculating the deflection were presented, either by considering the framing members in the LTF shear wall fully rigid or fully flexible. It was found that the real occurrence would lie in an intermediate area between fully rigid and fully flexible approach. (Boggian et al., 2021)

In the studies by Aloisio et al. (2020) and Gurholt and Mikalsen (2021), an eight-story CLT building located in Ås was subject to ambient vibration measurements, dynamic identification and model updating. Aloisio et al. (2020) performed the vibration measurements, while Gurholt and Mikalsen (2021) modelled a finite element model in SAP2000 and performed the dynamic identification and model updating. The results from the initial dynamic identification of the FE-model were inaccurate in comparison to the experimental values, providing Gurholt and Mikalsen (2021) the basis for concluding with that "the general reliability of FE-models portraying CLT buildings is somewhat weakened". However, it was mentioned that accuracy of the finite element model was dependent on modelling decisions. The process of model updating manage to create more representative finite element model in relation to the experimental data by correcting the errors of the initial dynamic identification. Furthermore, recommendations to apply the same method to other CLT building were made to further study the reliability of dynamic identification and model updating finite element analyses. (Gurholt and Mikalsen, 2021)

1.3 Research Questions

With background in the reviewed literature, this thesis will attempt to evaluate the research questions presented below:

1. How does the new analytical calculation models for lateral deflection of timber shear walls in the upcoming Eurocode 5 relate to Finite Element Analyses in SAP2000?
2. How sensitive are each contribution to the variation of different parameters and wall configurations?
3. How accurate is the modal analysis and model updating in SAP2000 compared to experimental values in a complex building of CLT shear walls?
4. How reliable are the different calculation models in the identification of timber shear walls subjected to different loading in the Serviceability Limit State?

1.4 Research Objectives

In this thesis, the performance of LTF and CLT shear walls technologies under different loading in the Serviceability Limit State are studied. The thesis essentially consists of two parts.

Part one of the thesis addresses the analytical equations for elastic lateral deflections at the top of timber shear walls presented in chapter 13.7 in the upcoming Eurocode 5 (CEN, 2022). A parametric analysis will be conducted to study the reliability of the equations under different circumstances. FEM models are going to be developed in SAP2000 in order to simulate the shear wall behaviour based on the Eurocode proposal. To run multiple parametric analyses, the Open Application Programming Interface function in conjunction with Python, will be used to study the influence of different parameters such as length of the wall, dimensions of frame elements, and dimension and set up of CLT panels. The objective is to do a comparison between the results from SAP2000 and the analytical proposal to discuss strengths and weaknesses.

The second part of the thesis addresses the dynamic behaviour of a multi storey building constructed with CLT shear walls. A modal analysis will be performed to analyse the reliability of a finite element analysis of the Serviceability Limit State. The objective is to do a comparison between modal analysis in SAP2000 and results from an experimental campaign on the InnoRenew CoE research institute’s new headquarters. The building is located in Isola, Slovenia. Furthermore, a model updating procedure will be executed on a finite element model using the Open Application Programming Interface function in conjunction with Python.

2 Theory

This chapter presents the theoretical foundation needed for the application of the parametric analysis of timber shear walls and the modal analysis of the multi storey building. The theory behind the properties of timber and the different timber shear wall technologies will be introduced in chapters 2.1 and 2.2. Chapter 2.3 summarize the background and the calculation models for lateral deflection of timber shear walls set to be included in the upcoming Eurocode 5. Chapter 2.4 describe the Finite Element Method, witch focus on SAP2000 Finite Element Method software and the Open Application Programming Interface, as well as usage of Finite Element Method in structural dynamics. Lastly, the mathematical theory behind modal analysis, identification of experimental data and Finite Element Model Updating are presented in chapters 2.5, 2.6 and 2.7.

2.1 Properties of Timber

Timber is an orthotropic material, with different properties along longitudinal, radial, and tangential axes, shown in Figure 2-1. From the figure we see how the different axes lay relative to the fibre (grain) direction of the wood. Important properties are tensile, compression, and shear properties. These properties along the three axes depend on the orientation relative to the grain, where radial and tangential axes are perpendicular to the grain and longitudinal is parallel to the grain. The radial and tangential properties can be hard to assess but are generally weaker than the properties in the longitudinal direction. (Sanborn et al., 2019)

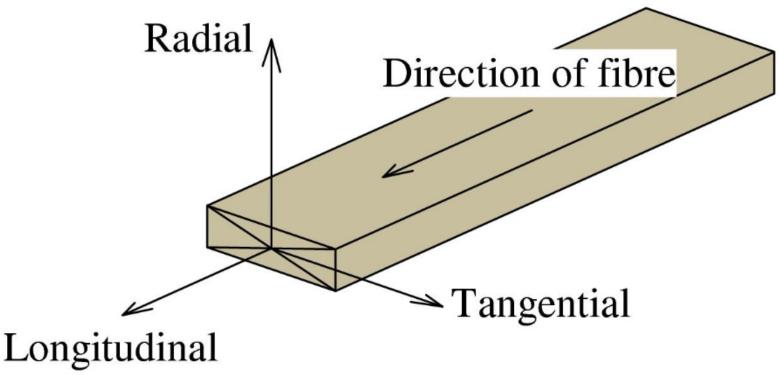


Figure 2-1: Axes in timber.

2.2 Shear Walls

Shear walls are vertical elements designed to have strength and stiffness to resist horizontal loads. Wind and earthquake loads are examples of lateral loads that might induce swaying and vibrations in high rise structures. These movements can develop high stresses in the structure; thus, sufficient resistance is important. Shear walls should be designed to resist in-plane lateral forces directed along the length of the wall. (Eusuf et al., 2012)

2.2.1 Light Timber Frame

LTF shear walls consists of four essential components: framing, sheathing, anchorage and fastening between the framing and the sheathing. The timber framing consists of vertical studs, and horizontal top and bottom rails, and are regarded as an unstable pin-jointed structure. Sheathing can be made of Oriented Strand Board (OSB), fibreboard, drywall, and many other materials. Usually, the fasteners between the framing and sheathing consists of a large number of nails, staples or screws working together in transferring forces. Anchorages are eighter brackets, rods, or robust screws, were as an anchorage system with angle brackets and hold-down brackets are quite common. Angle brackets resists sliding from the shear forces, and the Hold-down brackets resist overturing of the wall. (Marzaleh et al., 2018). An illustration of a LTF shear wall is presented in Figure 2-2.

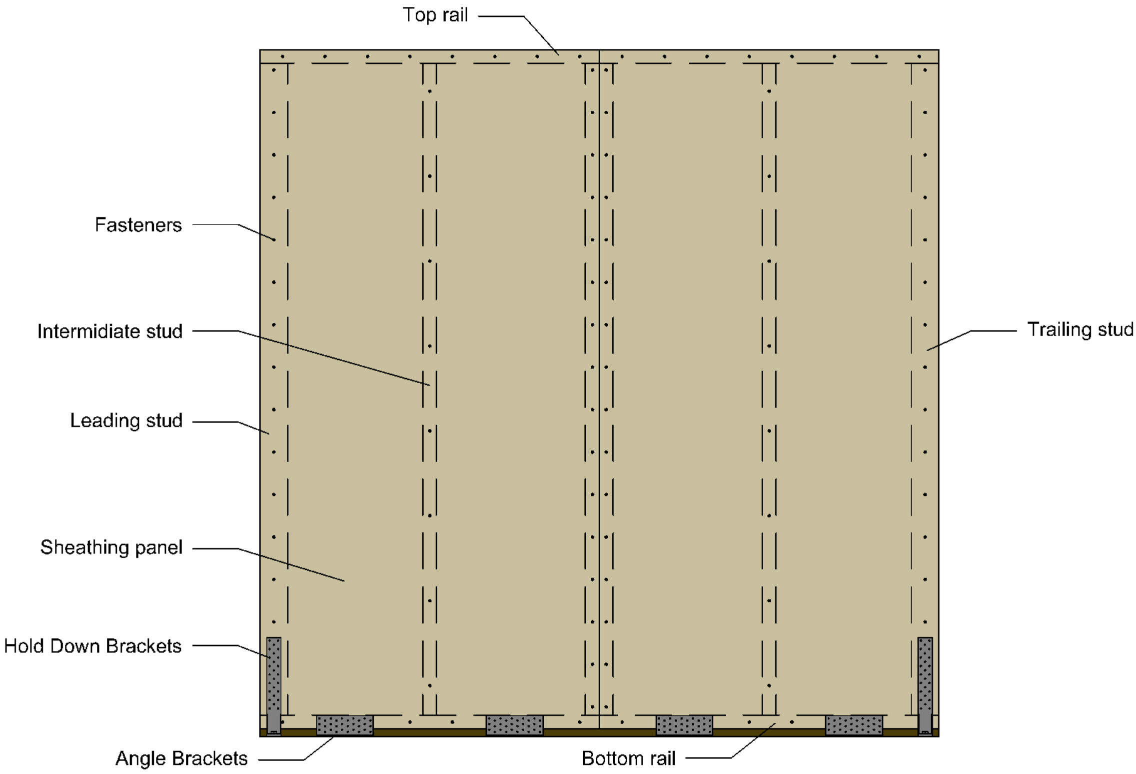


Figure 2-2: LTF shear wall.

Light timber frames are usually considered load-bearing walls and transfer both vertical and horizontal forces to the foundation. For multi storey buildings the shear capacity is of great importance. Lateral loads such as wind and seismic load will be transferred to the foundation through the light timber frames which act as a structural diaphragm. (Grossi et al., 2015)

Vertical forces are transmitted through the perimeter and internal timber studs, while the lateral forces are resisted mostly by the sheathing panels. The lateral forces are transferred to the framing elements by the fasteners and friction between the sheathing panels and the studs. The local mechanical connection system has a lot of influence in the rocking capacity of the shear wall. This can be observed as a high level of redundancy and ductile behaviour for the walls, leading to a global hysteretic dynamic response. There are a lot of contributions to this phenomenon, where fastener slip is one of the most relevant. (Grossi et al., 2015)

2.2.2 Cross Laminated Timber

Cross Laminated Timber (CLT) is a relatively new timber product that only have been in production since 1995. With the upcoming Eurocode 5 including standardisation on CLT not yet published, use of CLT to this day have relied on national or European technical approvals. (Wallner-Novak et al., 2014) However, the product has become of global interest and a well-recognized timber product among engineers. (Brandner et al., 2016)

CLT is a solid timber product that consists of multiple lamellas of wood stacked together in alternate directions. There are at least three layers glued together, and each layer is oriented perpendicular to the previous, making the product two-dimensional. By joining the layers of wood in this way, load-bearing applications are achieved, giving the CLT panel structural rigidity in both in-plane directions (Wallner-Novak et al., 2014). Figure 2-3 illustrates the structure of a CLT panel.

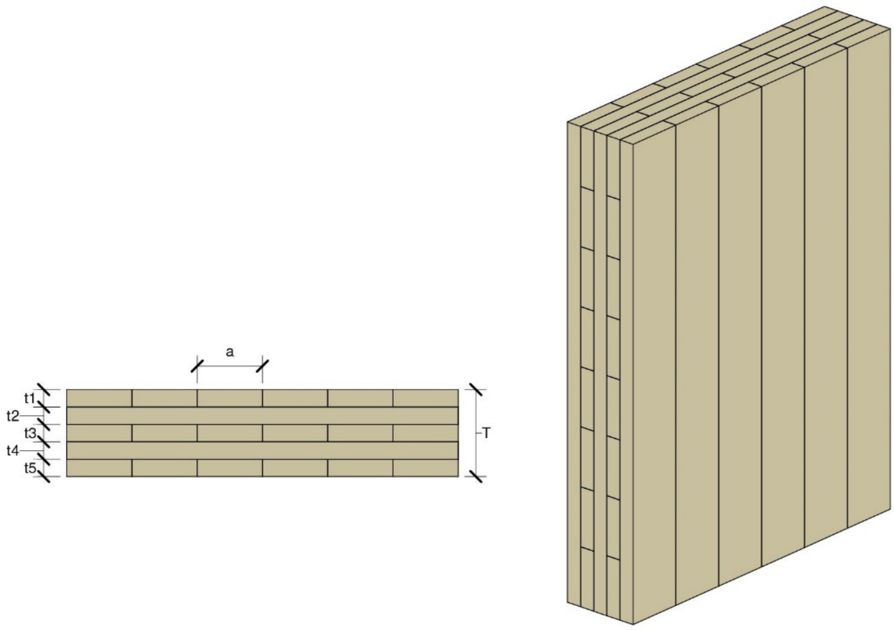


Figure 2-3: Structure of a CLT panel.

CLT Shear walls can consist of one monolithic panel or multiple segmented panels. There are several methods of connecting meeting panels in a segmented wall. Some of the techniques are screwing wooden boards to the face of the panels, screwing together rabbet edges, or screwing butted panels in an angle. The choice of joining technique is done considering the type of load the wall is subject to (Wallner-Novak et al., 2014). A typical setup of a CLT shear wall is illustrated in Figure 2-4.

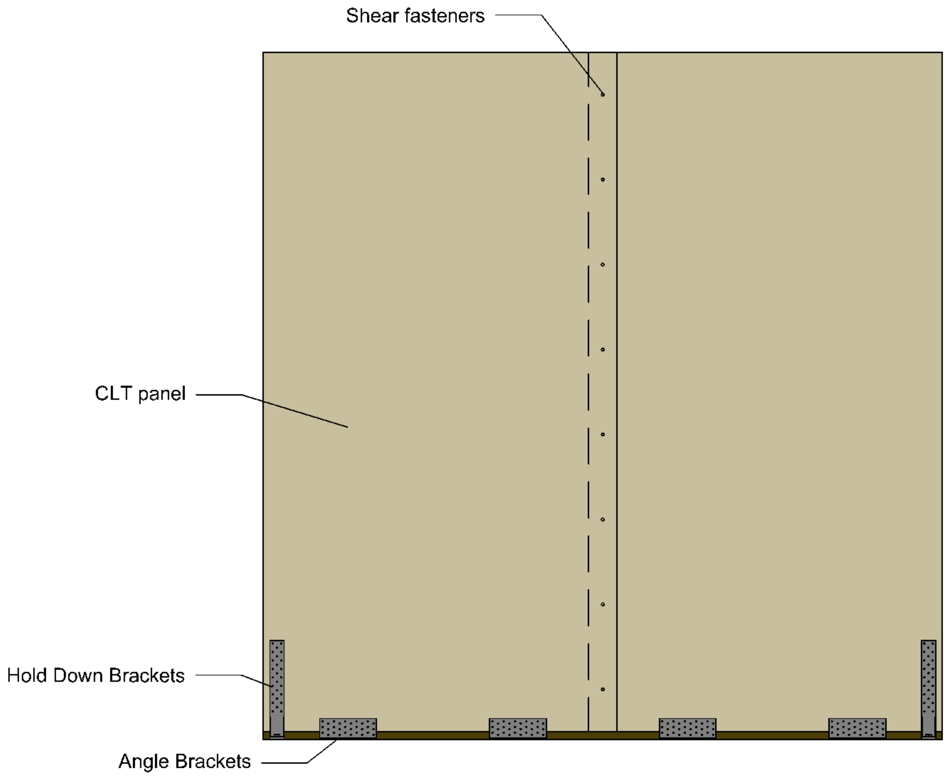


Figure 2-4: CLT shear wall.

Typically, four contributions are considered in the calculations of in-plane lateral deflection for CLT panels: sliding, kinematic rocking, panel shear and panel bending. Out of these, the sliding and kinematic rocking are considered to be the most influential for deflection and are governed by the capacity of base fasteners (Lukacs et al., 2019). The in-plane bending and in-plane shear are depending on the E-modulus, shear-modulus and the geometry of the wall. (Wallner-Novak et al., 2014)

E-modulus of transverse layers are neglected when calculating deflections in CLT panels because the E-modulus parallel to the grain in wood usually is about 30 times larger than the E-modulus perpendicular to the grain (Wallner-Novak et al., 2014). To simplify a CLT panel in an isometric model, the E-modulus can be calculated by adding the E-modulus of each layer according to equation (2.1).

$$E_{mean,0} = \frac{\sum_n E_0 \cdot t_n}{T} \tag{2.1}$$

Where:

E_0	is the modulus of elasticity parallel to the grain.
t_n	is the thickness of the n_{th} lamella in the longitudinal direction.
T	is the total thickness.

The shear stiffness can be calculated according to equation (2.2), or by the simplified equation (2.3).

$$G_{S,mean} = \frac{1}{1 + 6 \cdot \alpha_{FE} \cdot \left(\frac{d_{mean}}{a}\right)^2} \cdot G_{0,mean} \tag{2.2}$$

$$G_{S,mean} = 0{,}75 \cdot G_{0,mean} \tag{2.3}$$

$$\alpha_{FE} = 0{,}32 \cdot \left(\frac{d_{mean}}{a}\right)^{-0{,}77} \tag{2.4}$$

Where:

$G_{0,mean}$	is the mean shear-modulus parallel to the grain.
d_{mean}	is the average thickness of the cross-section.
a	is the board width, where 150mm is recommended.

2.3 Background of Chapter 13.7 Proposed for the Next Eurocode 5

Presented in this chapter are the background of the proposed chapter 13.7 set to be introduced to the next Eurocode 5. As briefly mentioned in chapter 1.2, the Technical Committee CEN 2021 are currently developing a new Eurocode for timber structures which will have the inclusion of analytical equations for calculating the lateral deflection for both LTF and CLT shear wall technologies. (CEN, 2021, CEN, 2022). These analytical equations are proposed to be included in chapter 13.7 (CEN, 2022).

In the proposal there can be found an analytical model for calculating elastic deflection at the top of a timber shear wall, for both LTF shear walls, monolithic CLT shear walls and segmented CLT shear walls. In the model, the total displacement is calculated through 7 different contributions where the sum is the total displacement. The contributions are listed under and illustrated in Figure 2-5.

The different contributions are:

a) Lateral deflection due to in-plane Shear.
b) Lateral deflection due to in-plane bending.
c) Lateral displacement due to rigid body sliding of the shear wall.
d) Lateral displacement due to kinematic rocking of the shear wall.
e) Lateral displacement due to deformation of sheathing-to-framing connections.
f) Lateral displacement due to deformation of the bottom rail perpendicular to the grain under the trailing stud.
g) Lateral displacement due to rotation at the top of the shear wall underneath namely, the shear wall at the i^{th} story

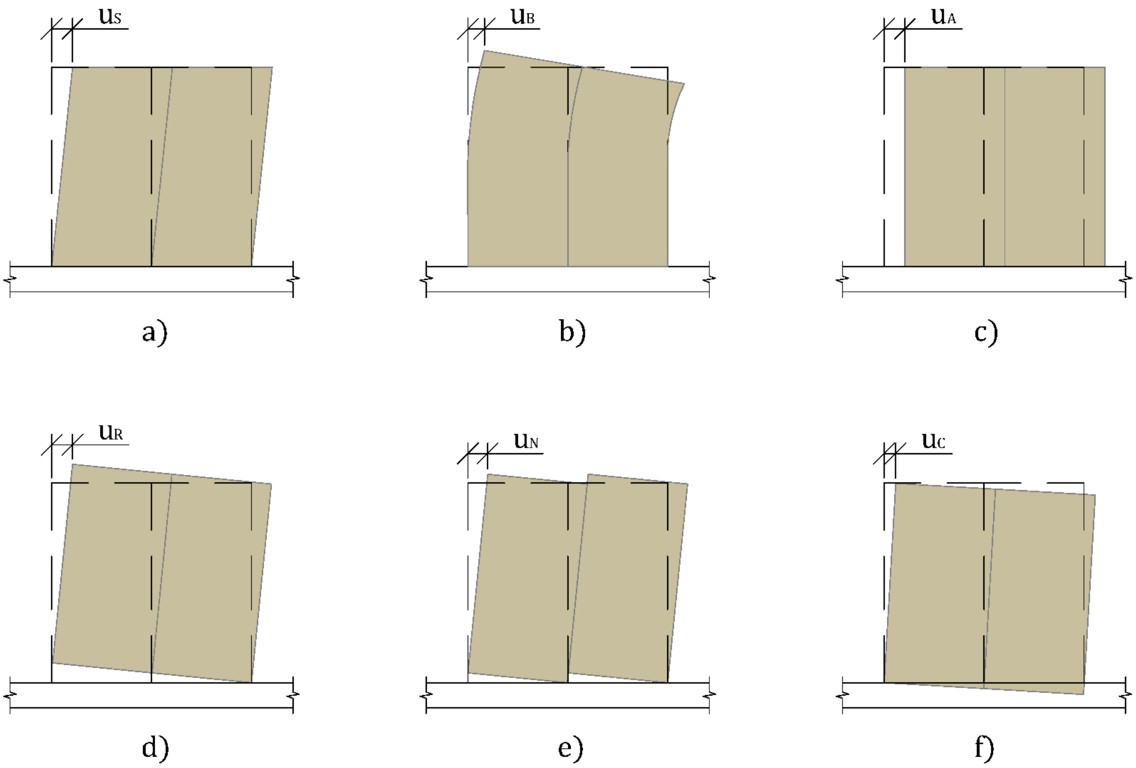


Figure 2-5: Contributions to elastic deflection at the top of a timber shear wall.

Definitions

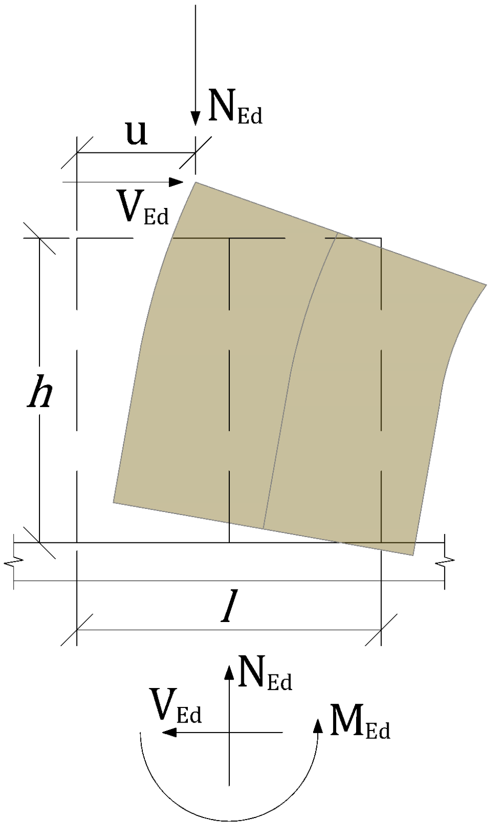


Figure 2-6: Mechanism of the shear walls.

h	is the height of the shear wall.
l	is the length of the shear wall.
V_{Ed}	is the design shear load.
N_{Ed}	is the design vertical force on the shear wall.
u	is the lateral deflection.
M_{Ed}	is the total design moment acting at the bottom of the shear wall, calculated as given by equation (2.5)

$$M_{Ed} = V_{Ed} \cdot h \tag{2.5}$$

Lateral Deflection due to In-plane Shear.

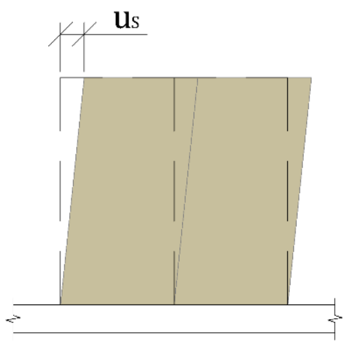


Figure 2-7: In-plane shear.

Deflection due to in-plane shear, illustrated in Figure 2-7, can be calculated according to equation (2.6) for LTF shear walls, and equation (2.7) for CLT shear walls.

$$u_S = \frac{V_{Ed} \cdot h}{l \cdot (G_{p,1} t_{p,1} + G_{p,2} t_{p,2})} \tag{2.6}$$

$$u_S = \frac{V_{Ed} \cdot h}{G_{xy,mean} \cdot t \cdot l} \tag{2.7}$$

Where:

$G_{p,1}, G_{p,2}$	are the shear moduli of the sheathing panels fixed to each side of the LTF framing.
$t_{p,1}, t_{p,2}$	are the thicknesses of the sheathing panels fixed to each side of the LTF framing.
$G_{xy,mean}$	is the mean effective in-plane shear modulus of the CLT shear wall.
t	is the total thickness of the CLT shear wall.

The elastic shear deflection has background in Hooke's law with linear elastic behaviour. Shear modulus for a plate is the ratio between shear stress and shear strain and can be calculated according to equation (2.8). (Hibbler, 2013)

$$G = \frac{\tau}{\gamma} = \frac{F/_A}{\Delta x/_h} = \frac{F \cdot h}{\Delta x \cdot A} \tag{2.8}$$

Where:

τ	is the stress.
γ	is the strain.
F	is the shear force.
h	is the hight of the plate.
A	is the cross sectional area of the plate.
Δx	is the shear deflection.

Rewriting equation (2.8) provides the general equation (2.9) on which equation (2.6) and (2.7) are created.

$$u_S = \Delta x = \frac{F \cdot h}{G \cdot A} \tag{2.9}$$

Lateral Deflection due to In-plane Bending.

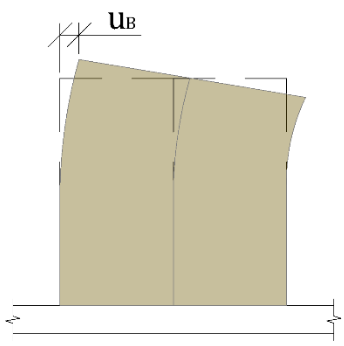


Figure 2-8: In-plane bending.

Deflection due to in-plane bending, illustrated in Figure 2-8, can be calculated according to equation (2.10).

$$u_B = \frac{V_{Ed} \cdot h^3}{3 \cdot EI_{eff}} \tag{2.10}$$

Where:

LTF:

$$EI_{eff} = \frac{E_{m,0,mean} \cdot A_{stud} \cdot l^2}{2} \tag{2.11}$$

CLT:

$$EI_{eff} = m \, \frac{E_{0,mean} \cdot t_z \cdot l^3}{12} \tag{2.12}$$

Where:

$E_{m,0,mean}$	is the mean modulus of elasticity parallel to the grain of the external studs.
A_{stud}	is the average cross-section area of the leading and trailing studs.
$E_{0,mean}$	is the mean modulus of elasticity parallel to the grain of the vertical laminations for CLT.
t_z	is the total thickness of the vertical layers for CLT shear walls.
m	is the number of segments in case of segmented CLT shear walls. $m = 1$ for monolithic CLT shear walls.

Equation (2.10) have background in the Euler–Bernoulli beam theory, with the deflection equation equal to the one for deflection of a cantilever beam.

Lateral Displacement due to Rigid Body Sliding of the Shear Wall.

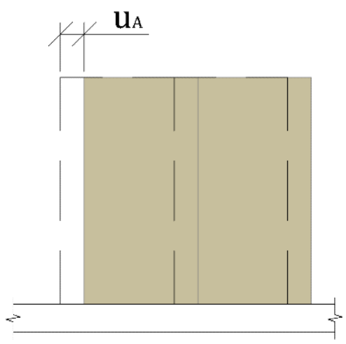


Figure 2-9: Rigid body sliding.

Rigid body sliding, illustrated in Figure 2-9,can be calculated according to equation (2.13).

$$u_A = \frac{V_{Ed}}{\sum_j K_{a,x,j}} \tag{2.13}$$

Where:

$\sum_j K_{a,x,j}$	is the total sliding stiffness of the shear wall.
$K_{a,x,j}$	is the horizontal-shear stiffness for serviceability limit state of the j^{th} angle bracket, illustrated in Figure 2-10.

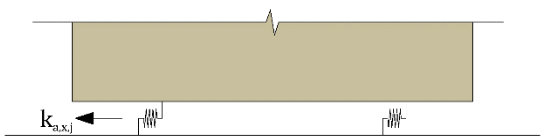


Figure 2-10: Angle brackets resisting sliding.

The theory of linear elasticity with the force-deflection relationship are the background of the elastic lateral displacement due to rigid body sliding.

Lateral Displacement due to Kinematic Rocking of the Shear Wall.

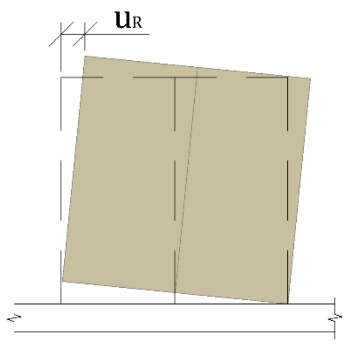


Figure 2-11: Kinematic rocking.

Equations (2.14) and (2.15) may be used to calculate the kinematic rocking, illustrated in Figure 2-11, for LTF and monolithic CLT shear walls.

$$u_R = max\left\{\left(\frac{M_{Ed}}{K_R} - \frac{N_{Ed}(l - l_c)}{2 \cdot K_R}\right) \cdot h\ ;\ 0\right\} \tag{2.14}$$

$$K_R = \sum_j \left[K_{a,z,j}\left(s_{a,j} - l_c\right)^2\right] \tag{2.15}$$

Where:

l_c	is the length of the compression zone. May be calculated as $0{,}1 \cdot l$
K_R	is the rocking stiffness of shear wall.
$K_{a,z,j}$	is the vertical-tensile stiffness for serviceability limit state of the j^{th} mechanical connections subjected to uplift. Illustrated in Figure 2-12.
$s_{a,j}$	is the distance of the j^{th} mechanical anchor from the edge of the shear wall. illustrated in Figure 2-12.

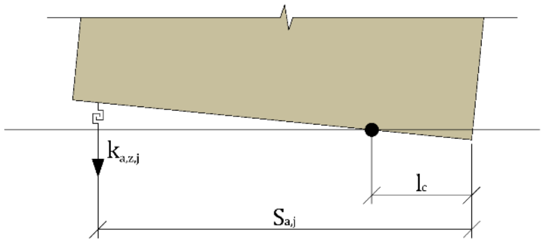


Figure 2-12: Hold-down brackets resisting kinematic rocking.

As for segmented CLT shear walls, kinematic rocking may occur as one of three different kinematic rocking modes depending on the relative stiffness of the hold-down (CEN, 2022). In Figure 2-13, Figure 2-14 and Figure 2-15, the three different modes with the respective centres off rotation are illustrated.

The three kinematic rocking modes are:

- Coupled panel (CP) kinematic mode
- Intermediate (IN) kinematic mode
- Single-wall (SW) kinematic mode

In coupled panel kinematic mode as illustrated in Figure 2-13, each panel is in contact with the floor underneath the shear wall (CEN, 2022). To achieve this mode, the prerequisites of equation (2.16) must be met.

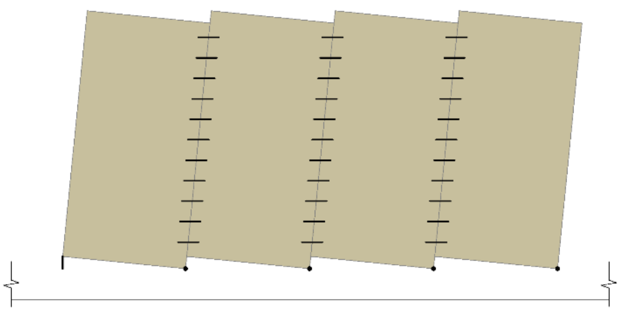


Figure 2-13: Coupled panel kinematic rocking mode.

$$\frac{K_{ser,anc}}{K_{ser,con}} \geq \frac{1 - \tilde{N}_l\,(3m-2)/m^2}{1 - \tilde{N}_l\,(m-2)/m^2} \tag{2.16}$$

Intermediate kinematic mode, as illustrated in Figure 2-14, occurs when some of the panels are in contact with the floor underneath the shear wall (CEN, 2022). This mode is achieved when the prerequisites of neither equation (2.16) nor (2.17) is met.

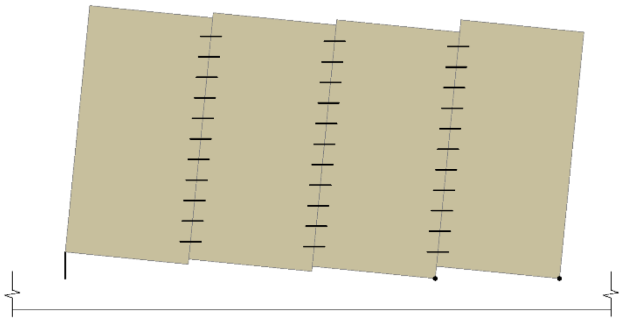


Figure 2-14: Intermediate kinematic mode

In single-wall kinematic mode as illustrated in Figure 2-15, only the last panel is in contact with the floor underneath the shear wall (CEN, 2022). To achieve this mode, the prerequisites of equation (2.17) must be met.

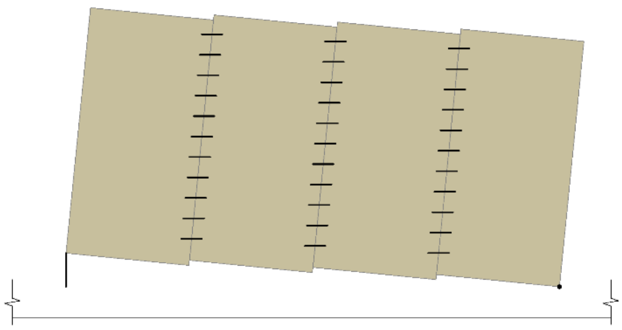


Figure 2-15: Single-wall kinematic rocking mode.

$$\frac{K_{ser,anc}}{K_{ser,con}} \leq \frac{1 - \tilde{N}_l}{1 + \tilde{N}_l\,(m-2)/m^2} \qquad (2.17)$$

Where:

$K_{ser,anc}$ is the tensile stiffness of the hold-down.

$K_{ser,con}$ is the stiffness of shear fastener between panels.

$\tilde{N}_l$ is the dimensionless vertical load on the shear wall. Calculated as given by equation (2.18)

$$\tilde{N}_l = \frac{N_{Ed} \cdot l}{2 \cdot M_{Ed}} \tag{2.18}$$

CP mode:

$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,CP}} - \frac{N_{Ed} \cdot l}{2 \cdot K_{R,CP}}\right) \cdot h\,;\ 0\right\} \tag{2.19}$$

SW mode:

$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,SW}} - \frac{N_{Ed}}{2 \cdot K_{ser,anc} \cdot l}\right) \cdot h\,;\ 0\right\} \tag{2.20}$$

Where:

$K_{R,CP}$ is the rocking stiffness of the shear wall in CP kinematic mode. Calculated as given by equation (2.21)

$K_{R,SW}$ is the rocking stiffness of the shear wall in SW kinematic mode. Calculated as given by equation (2.22)

$$K_{R,CP} = \frac{[K_{ser,anc} + (m-1)K_{ser,con}]}{m^2}\, l^2 \tag{2.21}$$

$$K_{R,SW} = \left[\frac{1}{K_{ser,anc}} + \frac{(m-1)}{K_{ser,con}}\right]^{-1} l^2 \tag{2.22}$$

Displacement u_R for the IN kinematic mode can be calculated by linear interpolation between the CP an SW kinematic modes.

Lateral Displacement due to Deformation of Sheathing-to-framing Connections.

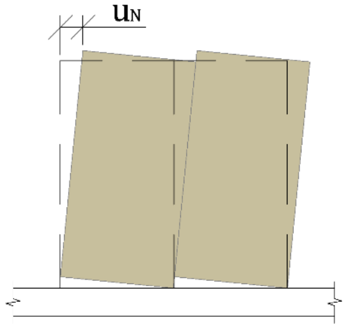


Figure 2-16: Sheathing-to-framing deflection.

Displacement due to deformation of the sheathing-to-framing connections, illustrated in Figure 2-16, can be calculated according to equation (2.23).

$$u_N = \frac{V_{Ed}/l^2}{\frac{K_{ser,1}}{a_{1,1}l_{per,1}} + \frac{K_{ser,2}}{a_{1,2}l_{per,2}}} \tag{2.23}$$

Where:

$K_{ser,1}, K_{ser,2}$	are the stiffnesses of the sheathing-to-framing fasteners on each side of the LTF framing
$a_{1,1}, a_{1,2}$	are the fastener spacings along the perimeter on each side of the LTF framing.
$l_{per,1}, l_{per,2}$	are the perimeter length of the sheathing panels fixed to each side of the LTF framing. Calculated as given by equation (2.24)

$$l_{per} = 2\sum_j b_{p,j} + 2ph \tag{2.24}$$

Where:

$b_{p,j}$	is the width of the *j*[th] sheathing panel.
p	Is the number of consecutive sheathing panels.

The lateral displacement due to the deformation of sheathing to framing fasteners can be calculated in two ways, either by considering the framing members fully rigid or fully flexible. The equation provided in the upcoming Eurocode 5 assume the framing members to be fully flexible, as this approach is the conservative one. As mentioned in chapter 1.2, the study by Boggian et al. (2021) found that the real occurrence would lie in an intermediate area between fully rigid and fully flexible approach. Furthermore, the increase of cross section of the framing members, thus increasing the stiffness, would reduce the deflection due to sheathing-to-framing deformation and approaching the fully rigid formulation case. (Boggian et al., 2021) This is not included in the equation (2.23) for the lateral displacement due to deformation of the sheathing-to-framing fasteners.

Lateral Displacement due to Deformation of the Bottom Rail Perpendicular to the Grain Under the Trailing Stud.

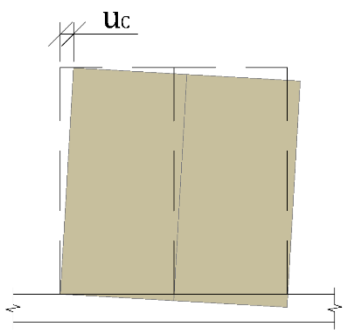


Figure 2-17: Compression perpendicular to the grain under the trailing stud.

Displacement due to compression perpendicular to the grain under the trailing stud, illustrated in Figure 2-17, can be calculated according to equation (2.25).

$$u_C = w_{ser,z} \frac{h}{l} \tag{2.25}$$

Where:

$w_{ser,z}$	is the compressive deformation of the bottom rail perpendicular to the grain. Calculated as given by equation (2.26), obtained from chapter 9.4 in the working draft of the next Eurocode 5 (CEN, 2021).

$$w_{ser,z} = \frac{h_{ef} F_{c,90,ser}}{2 b_c E_{90,mean}} \left(\frac{1}{l_c} + \frac{1}{l_{ef}} \right) \tag{2.26}$$

Where:

h_{ef}	is the effective beam height.
$F_{c,90,ser}$	is the applied compressive force perpendicular to the grain. Calculated according to equation (2.27).
b_c	is the width of the contact area.
$E_{90,mean}$	is the mean modulus of elasticity perpendicular to the grain.
l_c	is the length if the contact area of the applied force.
l_{ef}	is the effective spreading length of the compressive stresses.

$$F_{c,90,ser} = \frac{h}{l} V_{Ed} + F_z \tag{2.27}$$

Where:

F_z	is the compressive reaction force at the trailing edge of the wall.

2.4 Finite Element Method

Finite Element Method (FEM), or Finite Element Analysis (FEA) is a method used to numerically solve field problems. Field problems requires solutions of one or more dependent variables spatially distributed, as for example distribution of displacements and stresses in a component. In finite element method, a structure is divided into small pieces or finite elements. The finite elements are connected in a finite element structure where each finite element is connected at points called nodes. This display of elements is referred to as a mesh. (Cook et al., 2001)

Usually, a FEA produces an approximation to the solution which can be a good representation to the real solution, but not exact. The reasoning for this is that in an FEA the solution comes from numerically solve the algebraic equations for the unknowns at each node, element by element. The accuracy of the solution will increase with more elements to represent the structure. Thus, a finer mesh can be preferable to a coarse mesh. (Cook et al., 2001)

2.4.1 SAP2000

SAP2000 is a finite element method calculation program for general structures and is based on the general calculation engine referred to as SAPFire. Abilities in SAP2000 is linear modal calculations in three dimensions, linear and nonlinear static calculations in two and three dimensions and the ability to calculate shell and frame elements, among other. (Computers and Structures, 2017)

Area objects in SAP2000 can be defined as membrane, shell, or plate. Membrane elements supports only in-plane forces, plate elements the supports bending and transverse forces while full shell elements can handle all forces in-plane and out-of-plane. For the plate and shell definitions, there are thin and thick formulations, where the difference is that thin formulation doesn’t take transverse shear forces into account. All area elements can have a thickness assigned to it as the definition of the cross section of the element in question. (Computers and Structures, 2017)

Frame elements are defined as line elements and has six degrees of freedom at each end, which can be defined as a hinge or continuous. A section can be assigned to the frame element, and SAP2000 will calculate the behaviour of the element from the section properties. Sections are eighter user defined of predefined in the program. (Computers and Structures, 2017)

Link elements can be both point elements and line elements and can have several qualities assigned. The different types of links are linear, coupled linear, damper, gap, hook, multi-linear elastic, multi-linear plastic, plastic, hysteretic, friction pendulum and tension/compression friction pendulum. The links used in this study are linear, hook, and gap links. Linear links can be defied as fixed or as a spring in six degrees of freedom individually, hook links can only be defined to handle tension forces and gap links can only be defined to handle compressive forces. To activate hook and gap links, a nonlinear analyses must be run. (Computers and Structures, 2017).

The shear behaviour in SAP2000 is computed internally from the direct stress-strain curve, and assuming shear behaviour from tensile and compression acting at an angle of 45 degrees to material axes using Mohr's circle in the plane. This provides a symmetrical relationship for shear in Isotropic, Orthotropic, or Anisotropic materials. (Computers and Structures, 2017)

Nonlinearity in SAP2000 can be engaged by nonlinear links or supports, frames with tension or compression limitations, plastic behaviour in frame hinges, P-delta effect, or large displacements effects, by staged construction like changes or creep and shrinking. There are many initial conditions that can be applied to a model, like internal forces and stresses, external loads, energy values, displacements and velocity and internal state of variables for nonlinear elements. A nonlinear analysis will be more demanding than a linear analysis, and therefore more time consuming. (Computers and Structures, 2017)

SAP2000 can utilize both eigenvector analysis and Ritz-vector analysis in modal calculations. Eigenvector analysis is used for undampened and unforced calculations, and the Ritz-vector calculations are used for finding modes for situations where the construction is excited by a load or acceleration. Eigenvector analysis is based on the solution of the generalized eigenvalue problem. One eigenvector and eigenvalue will correspond to one mode and gives one set of eigenfrequency and natural period. The user may set a maximum number of modes the program will calculate, and SAP2000 will find the specified number of modes unless the number of modes specified surpasses the number of mass degrees of freedom in the model. It can be assigned a shift in the initial eigenfrequency, which means that you manually specify the range that SAP2000 will start looking for solutions to the eigenvalue problem. If done correctly, this can save some time in the calculations and even make the results more accurate. (Computers and Structures, 2017)

2.4.2 Open Application Programming Interface

Open Application Programming Interface (OAPI) can be utilized using a third-party program such as Spyder to control a calculation software such as SAP2000 with different programming languages, for example Python. The OAPI also links the two programs, making it possible to extract information from the SAP2000 software to the programming software and vice versa. This ability allows the user to automize creating, analysing, and extracting results. (Lagaros et al., 2019) (Computers and Structures, 2020)

Using the OAPI in conjunction with for example Python opens the possibility to control SAP2000 with external scripts, making it possible to automate the modelling and analysing process. Running multiple different analyses with pre-programmed input, each with different parameters and modifications can be performed over a short period of time, making it possible to extract a large number of results.

2.4.3 FEM in Structural Dynamics

In structural dynamics loading is time-dependent, and so is the structural response, making dynamic analyses more complicated than static analyses. Dynamic analyses often demand more computer resources as more operations may be needed to reach each intent of the analysis. Furthermore, planning of the numerical work, and do a preliminary analysis may be challenging, as it is difficult to anticipate structural response. (Cook et al., 2001)

The procedure of analysis of a structure by the finite element method may be summed up by the equations of motion following 5 steps given by Chopra (2020):

1. Model the structure as a finite element model with finite elements only interconnected at nodes and define degrees of freedom (DOF) at these nodes. (Chopra, 2020)

2. The stiffness matrix $\mathbf{k}_e$, the mass matrix $\mathbf{m}_e$, the geometric stiffness matrix $\mathbf{k}_{Ge}$ and the force vector $\mathbf{p}_e(t)$ should be defined for each finite element associated to the DOFs for the element. The force-displacement relation, the inertia force-acceleration relation, and force-displacement relation associated with gravity loads are obtained when assuming a displacement field over the element, expressed by nodal displacements. (Chopra, 2020) Equations for the different relations for each element are given by:

$$(f_S)_e = k_e u_e \quad (f_I)_e = m_e \ddot{u}_e \quad (f_G)_e = k_{Ge} u_e \tag{2.28}$$

3. For the relation between the displacement $\mathbf{u}_e$ and the forces $\mathbf{p}_e$ for the element to the displacement $\mathbf{u}$ and the forces $\mathbf{p}$ for the finite element, the transformation matrix $\mathbf{a}_e$ is required for the finite element assemblage (Chopra, 2020):

$$u_e = a_e u \quad p(t) = a_e^T p_e(t) \tag{2.29}$$

4. Establish matrices for stiffness, mass, and geometric stiffness, as well as the applied force vector $\mathbf{A}_e$, by the assemblage of element matrices. Then assemble the finite elements:

$$k = A_{e=1}^{Ne} k_e \quad m = A_{e=1}^{Ne} m_e \quad k_G = A_{e=1}^{Ne} k_{Ge} \quad p(t) = A_{e=1}^{Ne} p_e(t) \tag{2.30}$$

5. Equation of motion for the finite element assemblage:

$$m\ddot{u} + c\dot{u} + ku + k_G u = p(t) \tag{2.31}$$

Where:

c is the damping matrix.

2.5 Modal Analysis for Multi Storey Building

Natural frequencies and mode shapes are calculated using mass and stiffness matrices of the building in question. Depending on the distribution of stiffness and mass, there are several possible responses to excitation. A regular building, meaning a building where the centre of mass and the centre of stiffness is located in the same position and the structural elements are symmetrical, the three first modes will typically be one translational mode in x-direction, one translational mode in y-direction, and one torsional mode as shown in Figure 2-18. Other cases can give different modes, depending on the geometry of the building. (Chopra, 2020, Gokdemir et al., 2013)

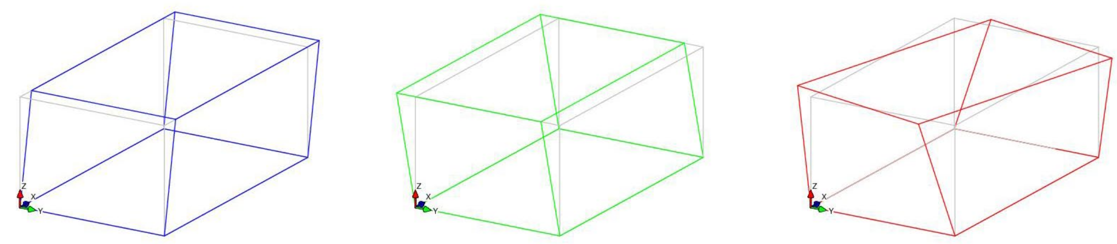


Figure 2-18: Mode shapes of a regular building.

Unsymmetric multistorey buildings with floor diaphragms the motion can be described by u_{jx}, u_{jx} and $u_{j\theta}$ degrees of freedom for each floor j in an N-story building, with the excitation $\ddot{u}_x$, $\ddot{u}_y$ and $\ddot{u}_\theta$. This is calculated by equation (2.32). Tall buildings create a large number of DOF, and a complicated calculation. The equation of undampened motion can be expressed as shown in equation (2.33). (Chopra, 2020)

$$u = \begin{Bmatrix} u_x \\ u_y \\ u_\theta \end{Bmatrix} \tag{2.32}$$

Where:

$$u_x = \langle u_{1x} \quad u_{2x} \quad \cdots \quad u_{Nx} \rangle^T \quad u_y = \langle u_{1y} \quad u_{2y} \quad \cdots \quad u_{Ny} \rangle^T \quad u_\theta = \langle u_{1\theta} \quad u_{2\theta} \quad \cdots \quad u_{N\theta} \rangle^T$$

$$\begin{bmatrix} m & & \\ & m & \\ & & I_O \end{bmatrix} \begin{Bmatrix} \ddot{u}_x \\ \ddot{u}_y \\ \ddot{u}_\theta \end{Bmatrix} + k \begin{Bmatrix} u_x \\ u_y \\ u_\theta \end{Bmatrix} = - \begin{bmatrix} m & & \\ & m & \\ & & I_O \end{bmatrix} \left(\begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix} \ddot{u}_{gx} + \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \ddot{u}_{gy} + \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \ddot{u}_{g\theta} \right) \quad (2.33)$$

Where:

m	is the lumped mass matrix of the N-story building.
k	is the stiffness matrix
I_0	is the diagonal matrix containing the moment of inertia for each storey.

2.6 Identification of Experimental Data

Stochastic subspace identification (SSI) is a method to extract state-space matrices from raw data from vibration tests. The method uses linear algebra and is desired because it avoids nonlinear calculations. For comparison with numerical values, the experimental mode shapes are modified in a way that makes them normalized. This is achieved by dividing all the displacements by the largest displacement in the mode, positive or negative. (Rainieri and Fabbrocino, 2014)

2.7 Finite Element Model Updating

The technique referred to as Finite Element Model Updating is an updating method which offer more accurate results from a finite element analysis in relation to experimental data (Panwar et al., 2018). Finite element model updating is utilized to analyse the dynamic behaviour of structural systems. As finite element analyses usually produce results that differ to experimental results from vibration tests, there is a need for a more accurate model, thus, finite element model updating is utilized. The goal of this type of analysis is to update the dynamic properties of the finite element model by implementing corrections to parameters in the model. The aim of updating dynamic properties is to close the gap to the experimental results (Velez et al., 2009). The procedure of finite model updating is illustrated in Figure 2-19.

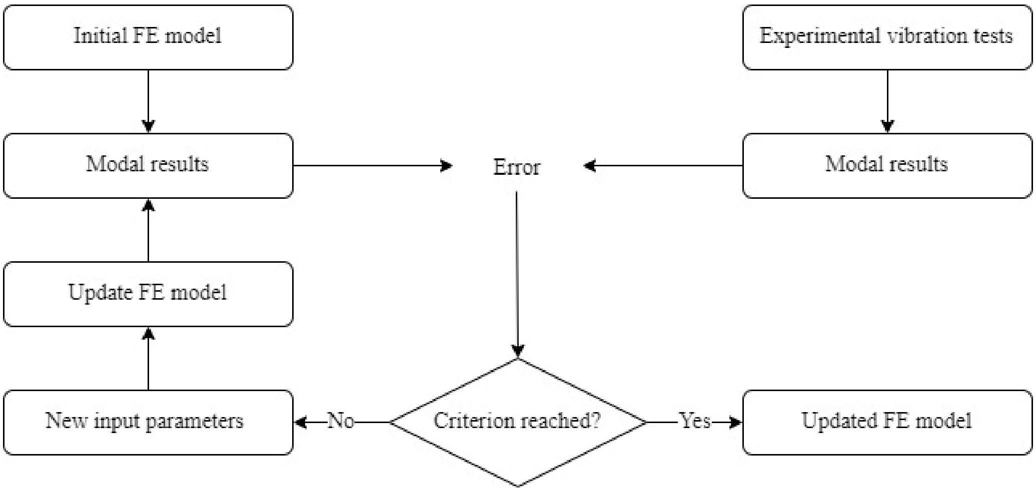


Figure 2-19: Flow chart of Finite Element Model Updating.

2.7.1 Error Calculation

A Total Convergence Criterium (CC_{tot}) can be used to evaluate the difference between two datasets presenting the behaviour of the same structure. The two contributing factors in the CC_{tot} are modal assurance criterion (MAC) and the deviation between the numerical and experimental eigen frequencies. Calculation of CC_{tot} can be done according to equation (2.34). The value of CC_{tot} represents the accuracy of the numerical model compared to the experimental measurements, and the closer the CC_{tot} is to zero, the more the two results correspond. (Mordini et al., 2007)

$$CC_{tot} = \sum_{i=1}^{n} \left(\frac{\left| f_{exp,i} - f_{num,i} \right|}{f_{exp,i}} + (1 - MAC_i) \right) \quad (2.34)$$

$$MAC(\varphi_j, \varphi_k) = \frac{\left| \varphi_j^T \varphi_k \right|^2}{(\varphi_k^T \varphi_k)(\varphi_j^T \varphi_j)} \quad (2.35)$$

2.7.2 Updating Procedure

Brownjohn et al. (2001) provides three general aspects to the updating procedure. These principals lay the foundation for the finite element model updating, and are as follows:

1. Based on the reference data, normally the measured data, a selection of responses is made.
2. A selection of parameters and a range in which the parameters are modified is selected for the updating procedure in order to change the selected responses.
3. An iterative process is implemented to modify the selected parameters based on the selected reference data, and this will result in a tuned model.

Differential evolution is a procedure that can be used to update a calculation model using gradient-based techniques in several iterations. The technique takes a given population size to create a set of parameter values to run at each iteration, and a function to optimize. To control the accuracy of the technique a tolerance of which the calculation will stop at if reached, a maximum number of iterations the calculation is limited to if the tolerance is not reached, a mutation constant, also known as the differential weight, that determines at what rate the calculations can mutate between each iteration, and a recombination constant that determines the number of mutations can be transferred to the next iteration. (community, 2022)

The range of values to test and the population size of each iteration is crucial when doing the updating and tuning of the model. Selecting a fitting population size can improve the chances of finding the global minimum deviation between the experimental and analytical models, without the updating getting stuck on a certain parameter value. (community, 2022)

3 Method

The aim of this chapter is to elucidate the method behind the parametric analysis and the modal analysis. Fastening is a major part of timber shear walls, and the fastening used in parametric analyses are presented in chapter 3.1. Chapter 3.2 and 3.3 presents the analytical and numerical models used for the parametric analysis presented in chapter 3.4. The setup of the finite element models of the shear walls is presented in chapter 3.5. The modal analysis, with the experimental campaign behind are presented in chapters 3.6 and 3.7. Lastly, the method of Finite Element updating are presented in chapter 3.8.

3.1 Fastening

Fastening elements used for the parametric analysis of different shear walls will be presented in this chapter. The fastening are mostly products from rothoblaas (2022), but also some fabricated elements to match the need for the analysis.

Nails and Screws

Calculation of stiffness for nails and screws as shear type fasteners between the sheathing and studs can be calculated as in the upcoming EC5, chapter 11 (Table 11.11a)(CEN, 2021) presented in equations (3.1) and (3.2).

Nails: $$K_{SLS,mean} = \frac{{\rho_{mean}}^{1.5} d^{0.8}}{30} \quad (3.1)$$

Screws: $$K_{SLS,mean} = \frac{{\rho_{mean}}^{1.5} d}{23} \quad (3.2)$$

Ring Nail:

- 2,8 mm: $k = \frac{\sqrt{550*420}^{1,5} \cdot 2,8^{0,8}}{30} = 800,41\ N/mm$
- 3,4mm: $k = \frac{\sqrt{550*420}^{1,5} \cdot 3,4^{0,8}}{30} = 934,91\ N/mm$

Small head screw:

- 5mm: $k = \frac{\sqrt{550 * 420}^{1,5} \cdot 1,1 \cdot 3,65}{23} = 1839,36$

Hold-down Brackets

WHT340 LBA:

- 14 nails 4,0 x 60mm: $k = 14 \cdot \frac{420^{1,5} \cdot 4,0^{0,8}}{30} = 12177\ N/mm$
- 20 nails 4,0 x 60mm: $k = 20 \cdot \frac{420^{1,5} \cdot 4,0^{0,8}}{30} = 17395\ N/mm$

WHT620 LBA:

- 35 nails 4,0 x 60mm: $k = 35 \cdot \frac{420^{1,5} \cdot 4,0^{0,8}}{30} = 30442\ N/mm$

Fabricated stiffness to achieve IN kinematic mode:

- $k = 19080\ N/mm$

Angle Brackets

TCF200/TTF200:

- 30 nails 4,0 x 60mm: $k = 30 \cdot \frac{420^{1,5} \cdot 4,0^{0,8}}{30} = 26093\ N/mm$
- 15 nails 4,0 x 60mm: $k = 15 \cdot \frac{420^{1,5} \cdot 4,0^{0,8}}{30} = 13046\ N/mm$

CLT Shear Fastener between Panels

HBS:

- 8mm: $k = \frac{420^{1,5} * 1,1 * 5,8}{23} = 2388\ N/mm$

3.2 Analytical Model for Lateral Deflection of Timber Shear Walls

The analytical calculations for lateral deflection of timber shear walls will be done using the models presented in chapter 2.3, set to be introduced to the new Eurocode 5 (CEN, 2021, CEN, 2022).

As mentioned in chapter 2.3, the different contributions are:

a) Lateral deflection due to in-plane Shear.
b) Lateral deflection due to in-plane bending.
c) Lateral displacement due to rigid body sliding of the shear wall.
d) Lateral displacement due to kinematic rocking of the shear wall.
e) Lateral displacement due to deformation of sheathing-to-framing connections.
f) Lateral displacement due to deformation of the bottom rail perpendicular to the grain under the trailing stud.
g) Lateral displacement due to rotation at the top of the shear wall underneath namely, the shear wall at the i^{th} story

In this analysis we have no contribution from lateral displacement due to the rotation at the top of the shear wall underneath namely, the shear wall at the i^{th} story. This is because only a single shear wall is considered.

Displacement contributions for LTF, monolithic CLT and segmented CLT shear walls without openings differ for the different types of timber shear walls. Under, in Table 3-1, there is presented which equations are representative for each contribution and for each type of timber shear wall.

Table 3-1: Representative equations for analytical calculations.

Contribution	LTF shear walls	Monolithic CLT shear walls	Segmented CLT shear walls
a) u_S	(2.6)	(2.7)	(2.7)
b) u_B	(2.10) and (2.11)	(2.10) and (2.12)	(2.10) and (2.12)
c) u_A	(2.13)	(2.13)	(2.13)
d) u_R	(2.14)	(2.14)	(2.19) or (2.20)
e) u_N	(2.23)	Not relevant	Not relevant
f) u_C	(2.25)	Not relevant	Not relevant

Adjustments.

When calculating kinematic rocking, two tests will be executed, one with the given equations from the suggested chapter 13.7 from 31.01.22 in the next Eurocode 5 (CEN, 2022), and one with some modified equations. The modified equations are based on the equations from the suggested chapter 13.7 from 31.01.22 in the next Eurocode 5 with some minor modifications.

Equations form the suggested chapter 13.7:

CP mode:

$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,CP}} - \frac{N_{Ed} \cdot l}{2 \cdot K_{R,CP}}\right) \cdot h\ ;\ 0\right\} \tag{3.3}$$

SW mode:

$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,SW}} - \frac{N_{Ed}}{2 \cdot K_{ser,anc} \cdot l}\right) \cdot h\ ;\ 0\right\} \tag{3.4}$$

It is possible that equation (3.3) might be inaccurate in including *l* (total wall length), and that it should be l_j (Segment length), when calculating CP mode.

Furthermore, it is also possible that equation (3.4) might be inaccurate when only dividing the design vertical force by hold-down stiffness multiplied by length, and that it should be the complete rocking stiffness for SW mode: $K_{R,SW}$. When doing so, the design vertical force also needs to be multiplied by segment length.

Modified equations:

CP mode:
$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,CP}} - \frac{N_{Ed} \cdot l_j}{2 \cdot K_{R,CP}}\right) \cdot h\, ;\ 0\right\} \tag{3.5}$$

SW mode:
$$u_R = max\left\{\left(\frac{M_{Ed}}{K_{R,SW}} - \frac{N_{Ed} \cdot l_j}{2 \cdot K_{R,SW}}\right) * h\, ;\ 0\right\} \tag{3.6}$$

3.3 Numerical Calculations for Lateral Deflection of Timber Shear Walls

The numerical calculations for lateral deflection of LTF and CLT shear walls will be done using the finite element program SAP2000. The program will be used along with the Open Application Programming Interface (OAPI) in conjunction with Python in order to calculate multiple results.

In order to numerically analyse comparable results to the analytical calculations, both for the total deflection and each contribution presented in the analytical model, several modifications need to be done to the SAP2000 models. Modifiers are applied to E-modulus and shear modulus (bending and shear stiffness), cross sections and stiffness of hold-downs, angle brackets, compression links and sheathing to framing links. All can be modified to keep the theoretical values or be modified to resist all motion in the specific way the mechanism moves, depending on the contribution.

3.4 Parametric Analysis of Shear Walls

In the parametric analysis of the shear walls, the focus will be on comparisons between the analytical model presented in the next Eurocode 5 and numerical analyses performed in SAP2000. Python will be used to generate comparisons between the numerical and analytical models. As mentioned in chapter 2.4.2, with the OAPI in conjunction with Python, we can open and run multiple numerical analyses in SAP2000. This provides the possibility to vary and modify different parameters and study the influence. The analysis will be done for LTF shear walls and monolithic- and segmented CLT shear walls.

The scrips will be created with the ability to run multiple consecutive analyses, all with different parameters. This is done to get a broad understanding of the behaviour of the different shear walls and get a good amount of data to do comparisons. With the amount of data provided from the parametric analysis, the different results can be studied and compared to create educated conclusions on the reliability of the analytical models. Scrips are available in Appendix B-D.

For the different shear walls, these are the parameters that will be varied:

LTF:

- Loading
- Number of consecutive sheathing panels.
- Cross sections of studs and rails.
- Stiffness and spacing of the sheathing-to-framing connectors.
- Stiffness of the hold-down brackets and angle brackets.

Monolithic CLT:

- Loading.
- Length of the wall.
- Cross section of the CLT wall.
- Stiffness of the hold-down brackets and angle brackets.

Segmented CLT:

- Loading.
- Number of consecutive CLT panels.
- Cross section of the CLT wall.
- Stiffness of the hold-down brackets, angle brackets and connectors.

The analytical models provide different contributions to the deflections of the shear walls. In order to study the reliability of each contribution, the parametric analyses are performed with different parametric input for each contribution separately. Not all contributions are impacted by all parameters, hence different parametric input for each contribution.

Total Deflection

The analysis of total deflection of LTF shear walls focus on the total deflection, and each contribution. The analysis gives an overview of the behaviour of the wall, and how each contribution contributes to the total deflection. For this analysis, not many parameters change. The parameters presented in Table 3-2, Table 3-3 and Table 3-4 will be the standard for all the analyses, and only the variating parameters will be presented in the further parametric tests.

Table 3-2: Parameters for LTF shear wall when analysing total deflection.

Parameter for LTF shear wall
Perimeter studs: 148x96mm
Internal studs and internal-perimeter studs: 148x48mm
Top rail and bottom rail: 148x96mm.
Timber quality: C24.
Panel thickness: 15mm.
One panel on each side of the wall.
3 studs per OSB panel, c/c 600mm.
Nail stiffness = 800.41 N/mm, c/c 100mm.
Angle backet stiffness = 13046 N/mm. 2 brackets at each panel.
Hold-down stiffness = 12177 N/mm. One at each end of the wall.
Loading: - 10 kN horizontal - 10 kN horizontal and 5 kN/m vertical.
Range 1 to 6 panels. One panel wxh = 1200x2400mm.

Table 3-3: Parameters for monolithic CLT shear wall when analysing total deflection.

Parameters for monolithic CLT shear wall
Panels: - CLT100 - CLT180
Timber quality: C24.
Angle backet stiffness = 13046 N/mm. 2 brackets at each panel.
Hold-down stiffness = 12177 N/mm. One at each end of the wall.
Loading: - 10 kN horizontal - 10 kN horizontal, 5 kN/m vertical.
Length 1200mm to 7200mm panel. 1200mm intervals.

Table 3-4: Parameters for segmented CLT shear wall when analysing total deflection.

Parameters for Segmented CLT shear wall
Panels: - CLT100 - CLT180
Timber quality: C24.
Angle backet stiffness = 13046 N/mm. 2 brackets at each panel.
Hold-down stiffness = 12177 N/mm. One at each end of the wall.
Stiffness shear fasteners = 2388 N/mm
Loading: - 10 kN horizontal - 10 kN horizontal, 5 kN/m vertical.
Range 1 to 6 panels. One panel wxh = 1200x2400mm.

Lateral Deflection due to In-plane Shear.

The analytical model presents the sheathing panels as the resisting element against the in-plane shear for LTF shear walls, and the CLT panels as the resisting element for CLT shear walls. Therefore, the main structural parameter changing for this analysis will be the sheathing panels for LTF, and the cross section of the CLT walls. Table 3-5, Table 3-6 and Table 3-7 presents the variable parameters for the analysis of elastic deflections due to in-plane shear.

Table 3-5: Parameters for LTF shear wall when analysing in-plane shear.

Parameter for LTF shear wall
Panel thickness: - 9 mm - 15mm - 22 mm
Loading: - 10 kN horizontal - 20 kN horizontal - 50 kN horizontal
Range 1 to 6 panels. One panel wxh = 1200x2400mm.

Table 3-6: Parameters for monolithic CLT shear wall when analysing in-plane shear.

Parameters for monolithic CLT shear wall
Panels: - CLT100s3 = [30,40,30] - CLT120s5 = [30,20,20,20,30] - CLT160s5 = [40,20,40,20,40] - CLT180s5 = [40,30,40,30,40]
Loading: - 10 kN horizontal - 20 kN horizontal
Length 1200mm to 7200mm panel. 1200mm intervals.

Table 3-7: Parameters for segmented CLT shear wall when analysing in-plane shear.

Parameters for Segmented CLT shear wall
Panels: - CLT100s3 = [30,40,30] - CLT120s5 = [30,20,20,20,30] - CLT160s5 = [40,20,40,20,40] - CLT180s5 = [40,30,40,30,40]
loading: - 10 kN horizontal - 20 kN horizontal
Range 1 to 6 panels. One panel wxh = 1200x2400mm.

Lateral Deflection due to In-plane Bending.

The analytical model presents the perimeter and trailing studs as the resisting element against the in-plane bending for LTF shear walls, and the CLT panels as the resisting element for CLT shear walls. Therefore, the main structural parameter changing for this analysis will be the perimeter and trailing studs for LTF, and the cross section of the CLT walls. Table 3-8, Table 3-9 and Table 3-10 presents the variable parameters for the analysis of elastic deflections due to in-plane bending.

Table 3-8: Parameters for LTF shear wall when analysing in-plane bending.

Parameter for LTF shear wall
Cross section perimeter and trailing studs: - 148x96 mm - 148x48 mm
Loading: - 10 kN horizontal - 20 kN horizontal
Range 1 to 6 panels. One panel wxh = 1200x2400mm.

Table 3-9: Parameters for monolithic CLT shear wall when analysing in-plane bending.

Parameters for monolithic CLT shear wall
Panels: - CLT100s3 = [30,40,30] - CLT120s5 = [30,20,20,20,30] - CLT160s5 = [40,20,40,20,40] - CLT180s5 = [40,30,40,30,40]
Loading: - 10 kN horizontal - 20 kN horizontal
Length 1200mm to 7200mm panel. 1200mm intervals.

Table 3-10: Parameters for segmented CLT shear wall when analysing in-plane bending.

Parameters for Segmented CLT shear wall
Panels: - CLT100s3 = [30,40,30] - CLT120s5 = [30,20,20,20,30] - CLT160s5 = [40,20,40,20,40] - CLT180s5 = [40,30,40,30,40]
loading: - 10 kN horizontal - 20 kN horizontal
Range 1 to 6 panels. One panel bxh = 1200x2400mm.

Lateral Displacement due to Rigid Body Sliding of the Shear Wall.

The analytical model presents the angle brackets as the resisting element against the rigid body sliding for both LTF shear walls and CLT shear walls. Therefore, the main structural parameter changing for this analysis will be the angle brackets. Table 3-11 presents the variable parameters for the analysis of elastic deflections due to rigid body sliding.

Table 3-11: Parameters for all shear walls when analysing rigid body sliding.

Parameters for all shear walls
Stiffness Angle brackets: - k = 13046 N/mm - k = 26093 N/mm
Loading: - 10 kN horizontal - 20 kN horizontal - 50 kN horizontal
Range 1 to 6 panels or 1200mm to 7200mm.

Lateral Displacement due to Kinematic Rocking of the Shear Wall.

The analytical model presents the hold-down brackets as the resisting element against the kinematic rocking for both LTF shear walls and CLT shear walls. Therefore, the main structural parameter changing for this analysis will be the hold-down brackets. Furthermore, the shear fasteners in segmented CLT shear walls will also contribute to the kinematic rocking. However, this parameter will stay unchanged to make the analysis less comprehensive. Table 3-12 presents the variable parameters for the analysis of elastic deflections due to kinematic rocking.

Table 3-12: Parameters for all shear walls when analysing kinematic rocking.

Parameters for all shear walls
Stiffness Hold-down brackets: - k = 12177 N/mm - k = 17395 N/mm - k = 19080 N/mm - k = 30442 N/mm
Loading: - 10 kN horizontal - 20 kN horizontal - 10 kN horizontal + 5 kN/m vertical - 10 kN horizontal + 10 kN/m vertical - 20 kN horizontal + 5 kN/m vertical - 20 kN horizontal + 10 kN/m vertical
Range 1 to 6 panels or 1200mm to 7200mm.

Lateral Displacement due to Deformation of Sheathing-to-framing Connections.

The analytical model presents the fastening as the resisting element against lateral displacement due to deformation of the sheathing-to-framing connections. Therefore, the main structural parameter changing for this analysis will be the fastening. Table 3-13 presents the variable parameters for the analysis of elastic deflections due the deformation of the sheathing-to-framing connections.

Table 3-13: Parameters for LTF shear walls when analysing sheathing-to-framing connections.

Parameters for LTF shear walls
Fastening: - Ring nail: d = 2,8 mm - Ring nail: d = 3,4 mm - Screw: d = 5 mm
Spacing between nails: - 75 mm - 100 mm - 120 mm - 150 mm
Loading: - 10 kN horizontal - 20 kN horizontal - 50 kN horizontal
Range 1 to 6 panel. One panel wxh = 1200x2400mm.

Lateral Displacement due to Deformation of the Bottom Rail Perpendicular to the Grain under the Trailing Stud.

The analytical model presents the bottom rail as the resisting element against lateral displacement due to deformation of the bottom rail perpendicular to the grain under the trailing stud. Therefore, the main structural parameter changing for this analysis will be the bottom rail. Table 3-14 presents the variable parameters for the analysis of elastic deflections due to the deformation of the bottom rail perpendicular to the grain under the trailing stud.

Table 3-14: Parameters for LTF shear walls when analysing deformation of the bottom rail.

Parameters for LTF shear walls
Cross section of the bottom rail: - 148x96 mm - 148x48 mm
Loading: - 10 kN horizontal - 20 kN horizontal - 10 kN horizontal + 5 kN/m vertical - 10 kN horizontal + 10 kN/m vertical - 20 kN horizontal + 5 kN/m vertical - 20 kN horizontal + 10 kN/m vertical
Range 1 to 6 panel. One panel wxh = 1200x2400mm.

3.5 Setup FE models of Shear walls

As mentioned in chapter 2.4.1, SAP2000 is a finite element modelling software, and is going to be the software used to model the numerical models and run numerical analyses. The setup of the FE model needs to be done with considerations to the different parameters presented in chapter 3.4, as well as the input needed in the analytical models presented chapter 2.3.

All the modelling of the parametric part is done using OAPI in conjunction with Python and can be found in Appendix B and C. The modelling is done in a way that gives opportunities to create many different wall configurations using the same procedure and changing the input.

3.5.1 LTF Shear Walls

The LTF model in SAP2000 consists of shell, frame and link elements defined in a way that simulate the behaviour of a LTF shear wall. The framing is hinged so it cannot distribute any rotational forces, and connections between the frame and the sheathing are defined as link elements resisting in-plane deformation. Sheathing elements are defined as membrane elements. To simulate hold-downs, angle brackets and compressive support, links are placed between bottom of the framing and restraints. Angle brackets are defined to only resist base shear forces as a linear link, and hold-downs are defined as hook link, so they only resist tensile forces. Also, gap links are used to withstand compressive forces under every stud without constraining tensile motion. Horizontal loading of the frame is applied in a point in the upper left corner while the vertical loads are applied on the top of the frame. A schematic visualization of the SAP2000 model is illustrated in Figure 3-1.

To avoid transfer of stress between the sheathing elements, a gap is set between the neighbouring sheets. At the meeting edges of the sheathing elements, a problem occurs for the in-plane resisting links. The links need to stay perpendicular to the sheathing elements to behave in the correct manner and the problem is solved by creating a body constraint between the ends of the pairing links and a node on the cohering stud. This is illustrated in Figure 3-2.

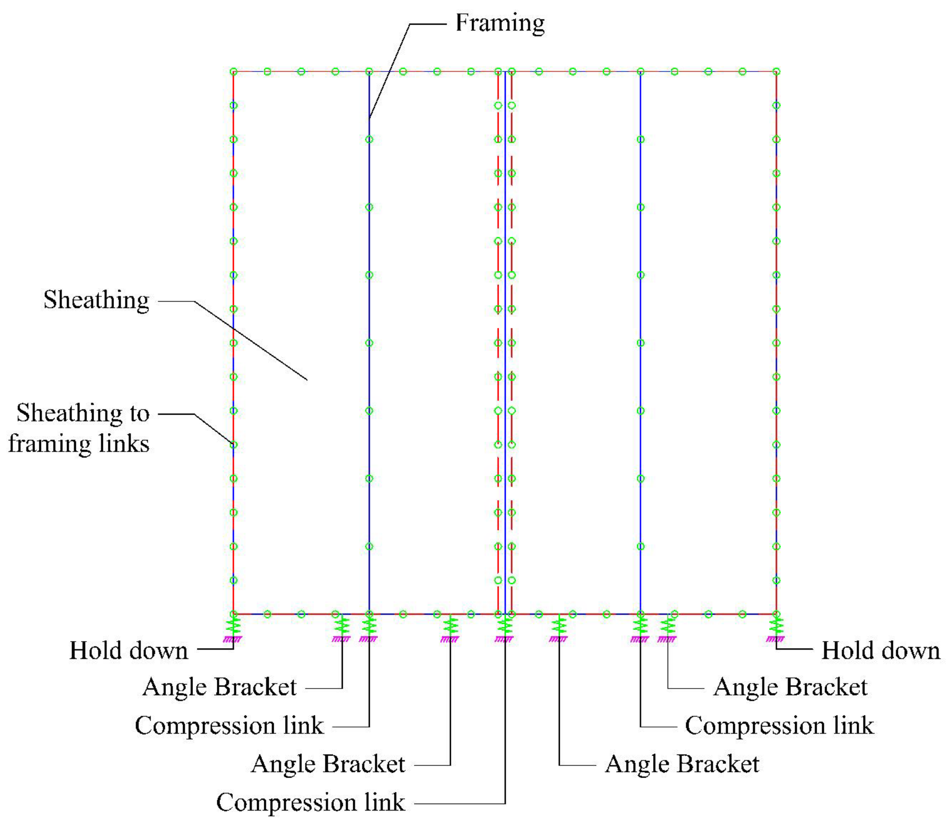


Figure 3-1: Schematic visualization of LTF in SAP2000.

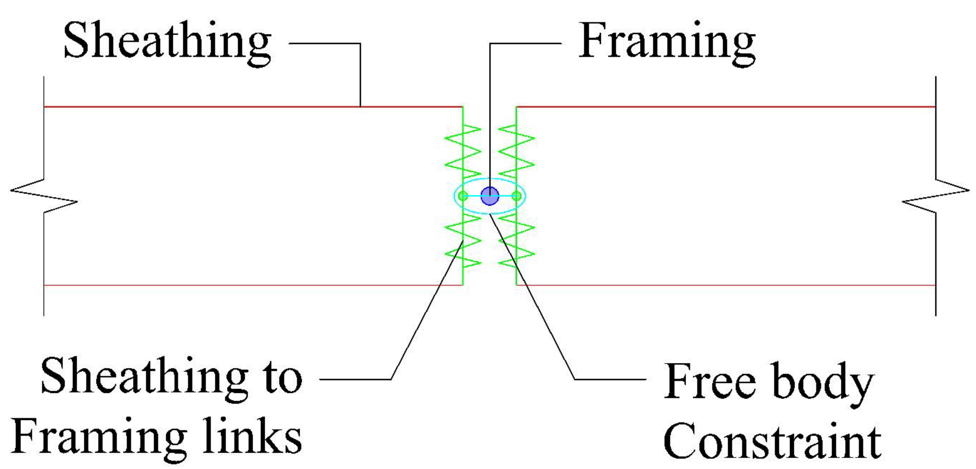


Figure 3-2: Schematic visualization of body constrain at meeting edges in SAP2000

To simulate the different contributions of the analytical calculations described in chapter 2.3, the parametric control of the numerical model needs to be able to manipulate the behaviour of individual components in the model.

For simulating lateral deflection due to in-plane shear, all links are set to act as rigid connections, the framing is set to be infinitely stiff, and the sheathing panels is modified to not be able to bend, so the only contributing factor in the simulations are the in-plane shear of the sheathing.

For simulating lateral deflection due to in-plane bending, all links are set to act as rigid connections, the framing is set to be infinitely stiff against shear, and the sheathing panels is modified to not be able to deform due to shear, so the only contributing factor in the simulations are the in-plane bending of the sheathing and framing.

For simulating lateral displacement due to rigid body sliding of the shear wall, sheathing panels are set to not deform due to both bending and shear forces, the framing elements are modified to not contribute to deflection and the sheathing-to-framing, hold-downs and compression links are modified to not contribute to the deflection.

Lateral displacement due to kinematic rocking of the shear wall is simulated by modifying sheathing panels to not deform due to both bending and shear forces, the framing elements are modified to not contribute to deflection and the sheathing-to-framing, angle brackets and compression links are modified to not contribute to the deflection.

Lateral displacement due to deformation of the sheathing-to-framing connections is modeled by modifying sheathing panels to not deform due to both bending and shear forces and the hold-downs, angle brackets and compression links are modified to not contribute to the deflection.

For lateral displacement due to the deformation of the bottom rail perpendicular to the grain under the trailing stud, sheathing panels are modified to not deform due to both bending and shear forces, the framing elements are modified to not contribute to deflection and the hold-downs, angle brackets and sheathing to framing links are modified to not contribute to the deflection.

3.5.2 CLT Shear Walls

The CLT model in SAP2000 consists of thin shell elements and link elements. The monolithic model consists of only one shell, and the segmented is put together by several shells connected by linear links and body constraints, to simulate the shear connectors between the panels and the resistance obtained by contact between the panels. The constraint stops deformation in the forced direction, and the link resists deformation due to sliding between the segments. Restraining of the model is done by placing links between the restraints and the bottom of the CLT panels. Hook links that only resist tensile forces are used to simulate the function of hold-downs, linear links are used to simulate angle brackets, and gap links that only resist motion due to compressive forces are used to restrain movement downwards.

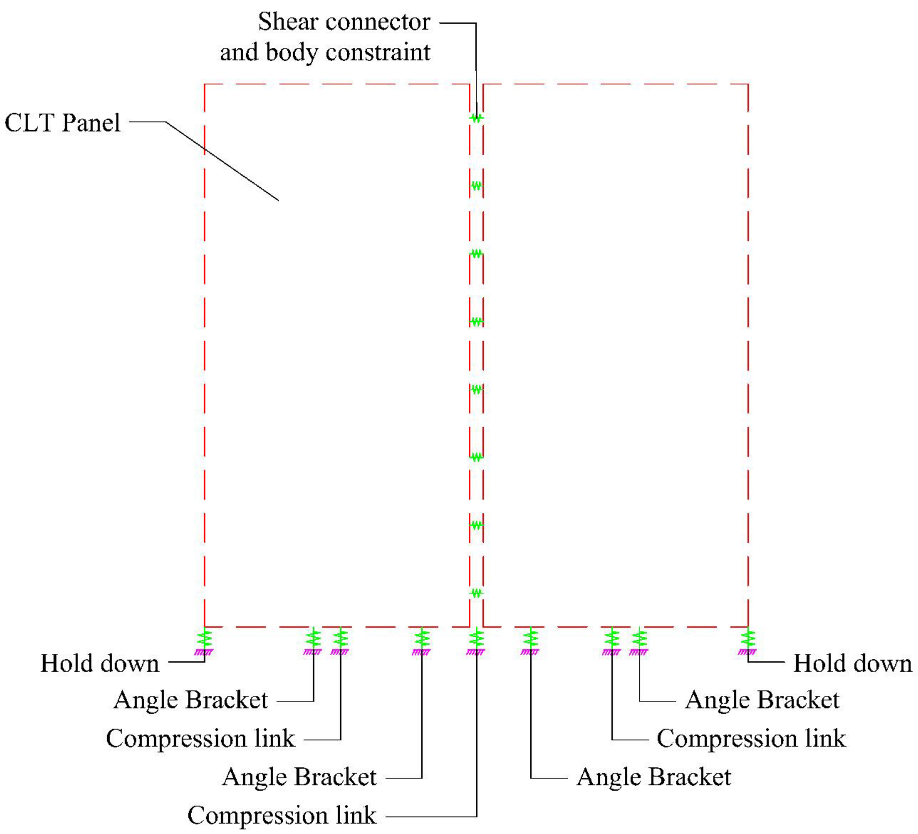


Figure 3-3: Schematic visualization of CLT in SAP2000.

To simulate the different contributions of the analytical calculations described in chapter 2.3, the parametric control of the numerical model needs to be able to manipulate the behaviour of individual components in the model.

For simulating lateral deflection due to in-plane shear, angle brackets, hold-downs and shear connectors between the panels are set to not deform, and the CLT panels is modified to not be able to bend, so the only contributing factor in the simulations are the in-plane shear of the panels.

For simulating lateral deflection due to in-plane bending, all links are set to act as rigid connections, and the CLT panels is modified to not be able to deform due to bending, so the only contributing factor in the simulations are the in-plane shear of the panels.

For simulating lateral displacement due to the rigid body sliding of the shear wall, CLT panels are set to not deform due to both bending and shear forces and the hold-downs, compression links and shear connectors between panels are modified to not contribute to the deflection.

Lateral displacement due to kinematic rocking of the shear wall is simulated by modifying CLT panels to not deform due to both bending and shear forces and the angle brackets and compression links are modified to not contribute to the deflection.

3.6 Modal Analysis of Building

After the static analysis of the different shear walls, the next step in analysing the reliability of the calculation models in Serviceability Limit State will be to do a modal analysis of a multi storey building. The modal analysis will be performed using SAP2000 in conjunction with Python by the help of the OAPI. The InnoRenew CoE research institute’s new headquarters, located in Isola, Slovenia, will be analysed.

3.6.1 Model

The InnoRenew building is a hybrid structure of CLT, steel and concrete elements. The main structural elements at the second, third and fourth floor of the building are CLT shear walls, while the walls on the first floor consists of concrete. A concrete slab is placed between the first and second floor, while the rest of the floors consists of CLT panels. In the core of the building there is a steel frame to support the floors in the third and fourth floor as well as the roof, this is visualized in Figure 3-5 a). A concrete shaft is placed in one of the corners of the building from the ground all the way to the roof, visualized in Figure 3-5 a). Furthermore, there are some timber frame elements around the perimeter of the building to support the CLT floor elements between the column-like structure of the outer walls, as shown in Figure 3-5 b). A visualization of the structure in its whole can be found in Figure 3-4.

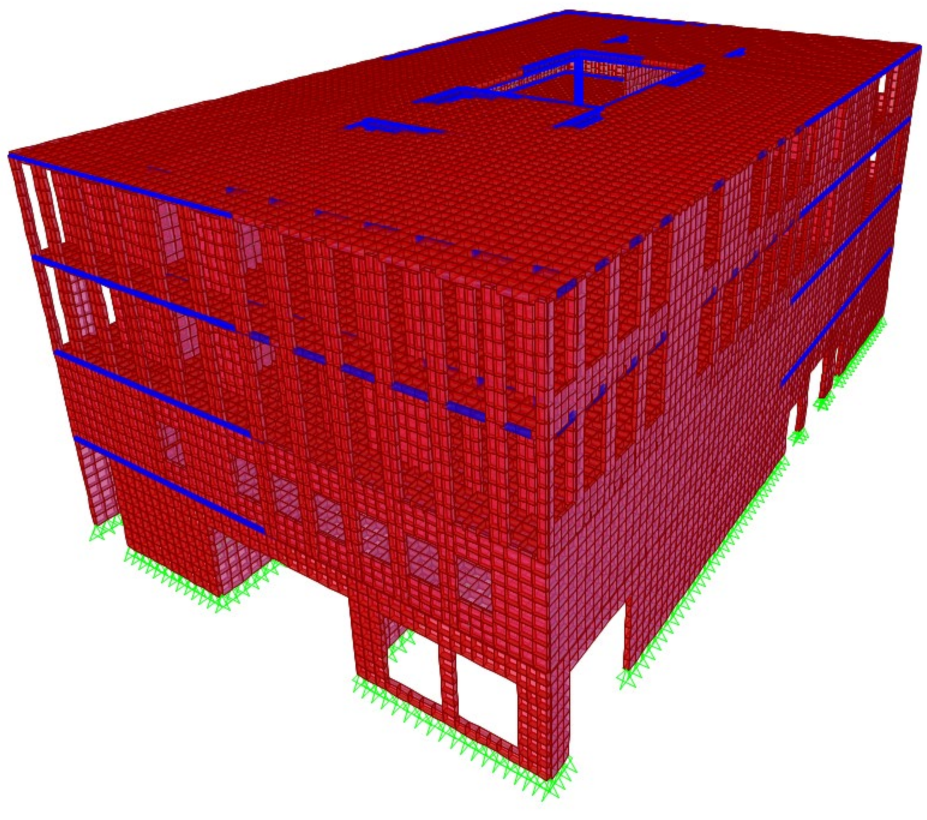
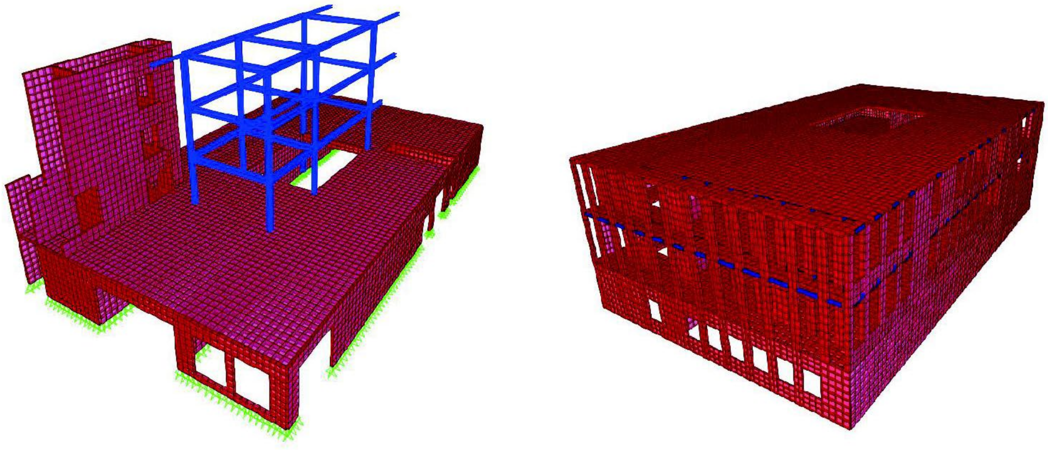


Figure 3-4: SAP2000 model of building.



a) b)

Figure 3-5: Structural elements in the building.

The geometry of the FE model is based on a 3D-model of the actual structure provided by the InnoRenew CoE. CLT panels are defined as thin shell sections with a specific orthotropic material, which gives the area section one E-modulus and one shear modulus for each direction as given by the material. Concrete walls are defined as thin shell sections with the same isotropic material for every section. The wood and steel framing are modelled as frame elements with cross section and material properties for the given elements.

SAP2000 has a limitation where meeting area elements does not connect to each other unless the two areas share at least two nodes. This problem has been taken care of by assigning edge constraints to all area elements, making them connect to each other without sharing corner nodes.

To restrain the bottom of the concrete walls in the basement, restraints are placed at every node created by the meshing at the base of the wall elements on the first floor. This makes the model restrained at the ground level with a consistent distance that is the same as the meshing size. All restraints stop movement in the three translational degrees of freedom and are free to rotate. It is also important to note that the building is surrounded by lower raised buildings; however, these are not included in the model stiffness.

3.6.2 Meshing

To create an accurate model, the mesh needs to be fine enough to give accurate results without making the model too complicated for the computer to run. To test this, several runs of the same model with different meshing sizes are run and compared. The results will be considered in choosing the mesh size for further analysis.

3.7 Experimental Campaign

The experimental tests have been executed by InnoRenew CoE between March 2021 to April 2022. During this time, the building has been through several stages of completion, from only structural elements to finished building. Also, some of the tests are executed with forced input, using a shaker in the locations showed in Figure 3-8 and Figure 3-9. The experimental measurements are extracted using five sensors in three different setups. The setup and locations off all sensors in the building are presented in Figure 3-6, Figure 3-7, Figure 3-8 and Figure 3-9. The first setup is sensors 1, 2, 3, 4 and 5, second setup is sensors 1', 2', 3', 4' and 5, and the third setup is sensors 1'', 2'', 3'', 4'' and 5. The three setups share sensor 5, which is a reference sensor.

The dataset used in this thesis is from a forced vibration test executed on 30.03.2022, from the sensors 1, 2, 3, 4 and 5. Response was recorded while a shaker sweeps across frequencies from 2 to 22 Hz, at a rate of 5 Hz/h. The data from the tests has been interpreted and normalized using SSI analysis before provided to the authors to be compared to the FE model for model updating.

Figure 3-6: Overview of sensors.

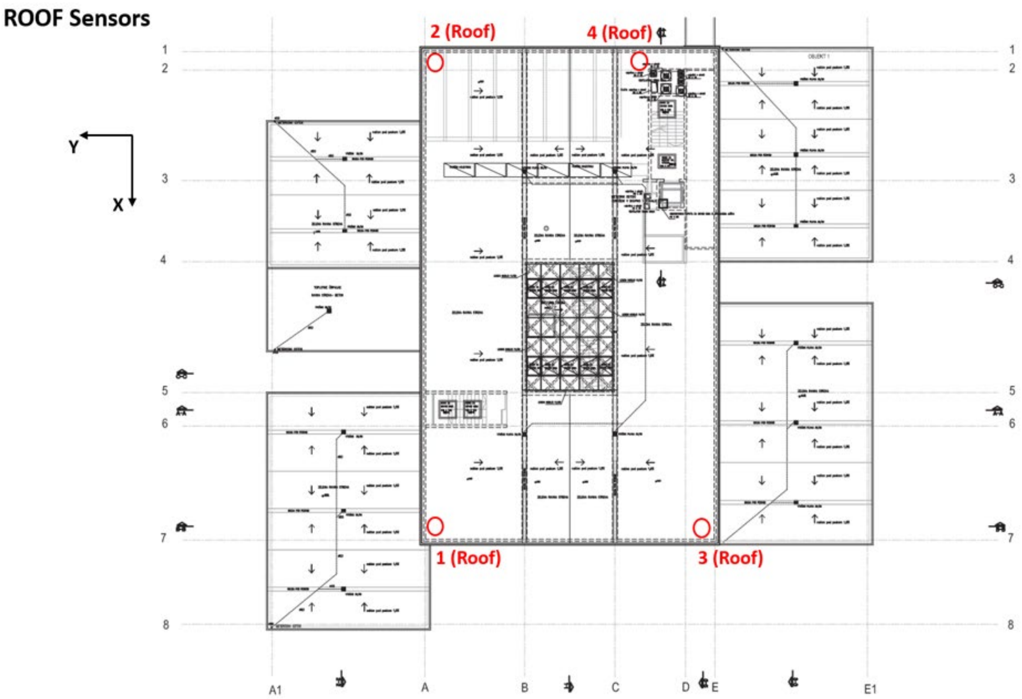


Figure 3-7: Location of roof sensors.

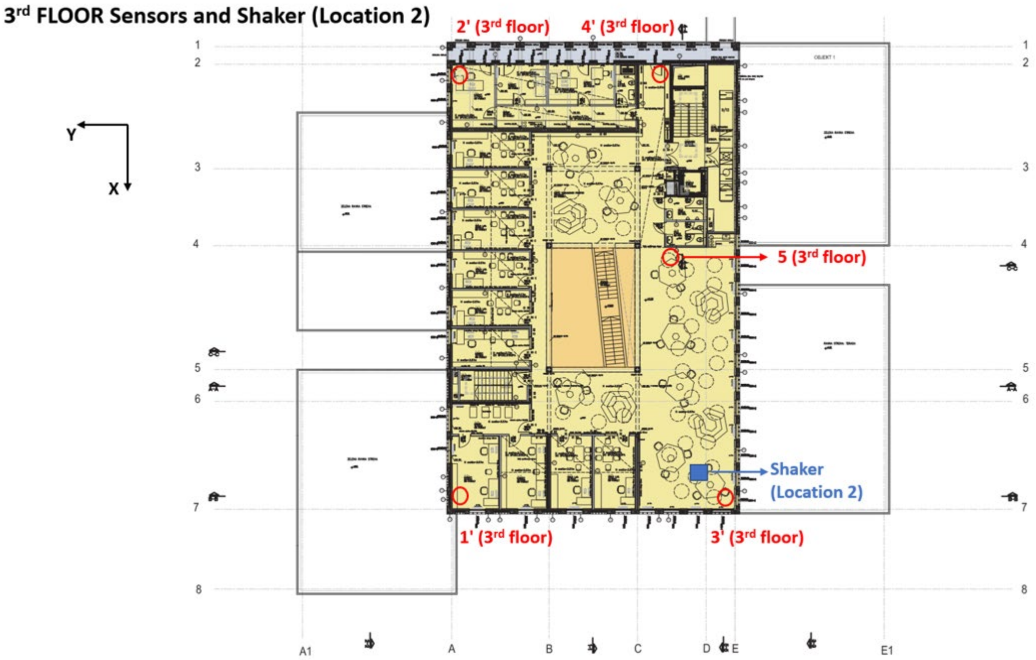


Figure 3-8: Location of 3rd floor sensors and shaker.

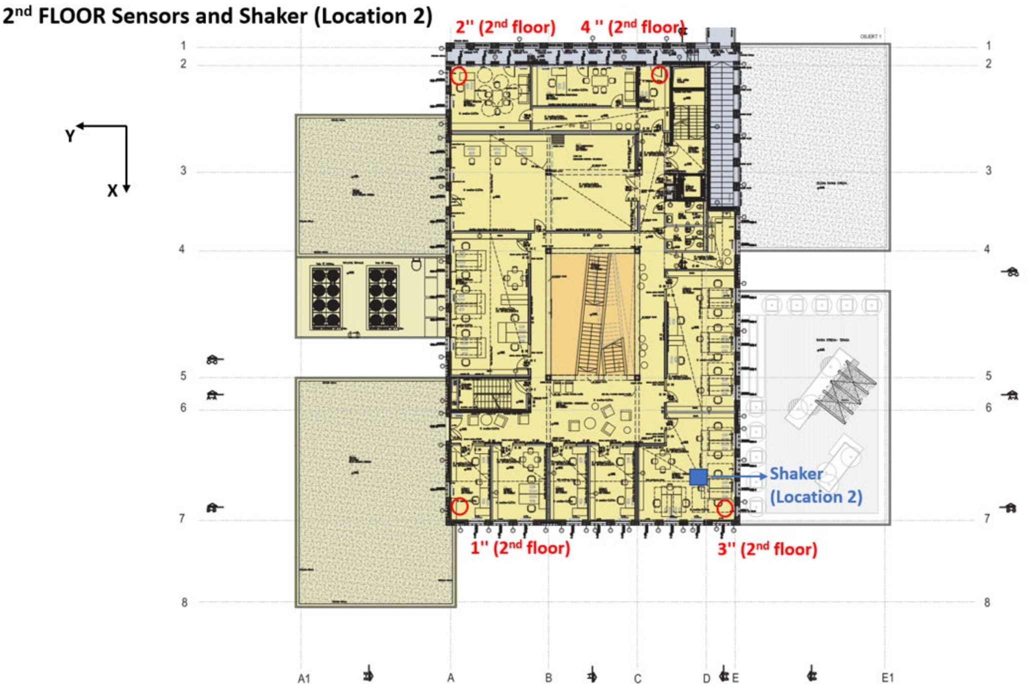


Figure 3-9: Location of 2nd floor sensors and shaker.

3.8 Finite Element Model Updating

Updating of the SAP2000 model is done using advanced programming tools in Python. The technique used is a differential evolution function, explained in chapter 2.7.2. To compare the experimental and the numerical results, equation (2.34) for CC_{tot}, as presented in chapter 2.7.1, is used.

For the updating process, the parameters used in the differential evolution is:

- Population size: 60
- Mutation constant: 0.9
- Recombination constant: 0.8

For this single set of experimental data, it is only possible to update the building for eighter mass or stiffness, and the mass is chosen for the updating. The parameters used in the updating are uniformly distributed load on the roof, uniformly distributed load on each internal floor and line load along the perimeter of the building to simulate facades. The loading is included into the mass source of the modal analysis and will behave like an additional mass. The range of loading on the roof is 0 kN/m^2 to 2 kN/m^2, the range of loading on the internal floors are 0 kN/m^2 to 2 kN/m^2 and the loading from the facades ranges from 0 kN/m to 4 kN/m. The initial additional mass parameter utilized in the identification of the modal analysis are 0 kN/m^2 additional mass from the roof, 0 kN/m^2 additional mass from the internal floors and 0 kN/m additional mass from the facades, making self-weight the only mass contribution.

4 Results

Presented in this chapter are the results both from the parametric analysis and the modal analysis with the model updating. Chapter 4.1 includes the results from the parametric analysis conducted on LTF shear walls. Chapter 4.2 and 4.3 includes the results from the parametric analysis conducted on CLT shear walls, respectively monolithic and segmented CLT shear walls. A summary of the extensive parametric analysis is presented in chapter 4.4. Lastly the results from modal analysis and model updating are presented in chapter 4.5.

4.1 Parametric Analysis of LTF Shear Walls

A selection of the result from the parametric analyses on the LTF shear walls are presented in this chapter. The most relevant results are presented, while all results can be found in Appendix A.

4.1.1 Total Deflection

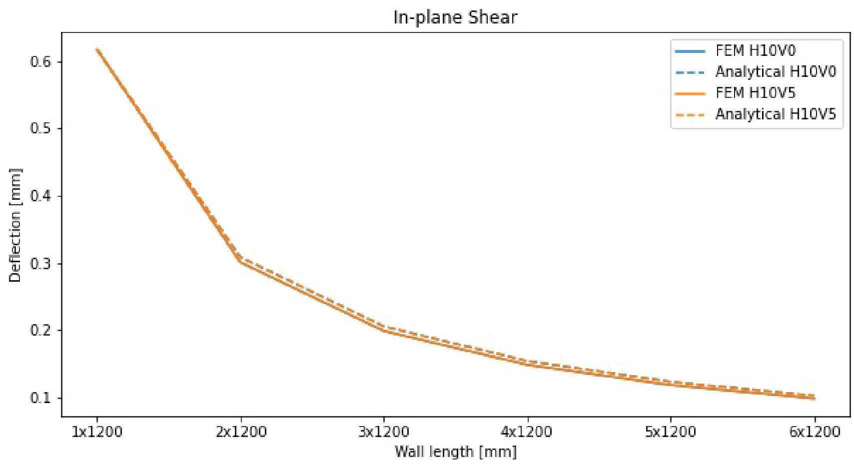


Figure 4-1: In-plane shear when calculating total deflection of LTF.

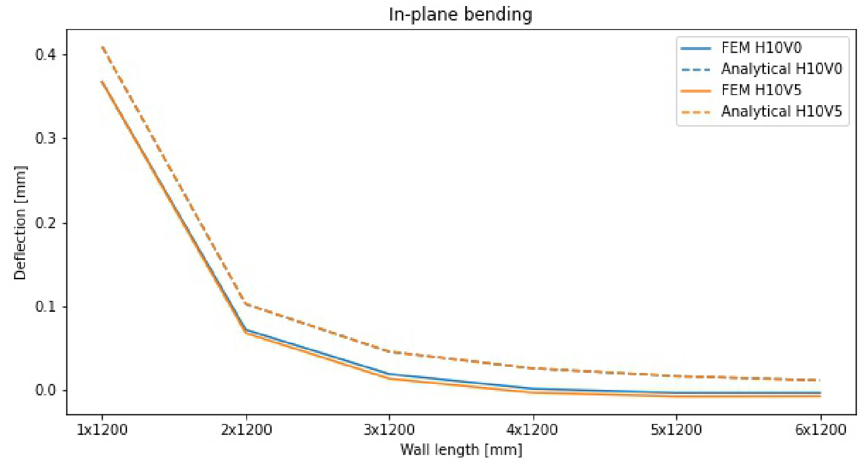


Figure 4-2: In-plane bending when calculating total deflection of LTF.

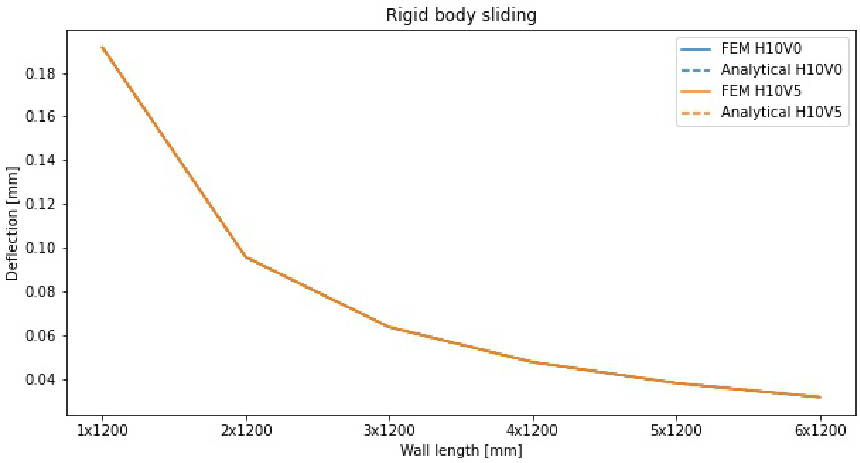


Figure 4-3: Rigid body sliding when calculating total deflection of LTF.

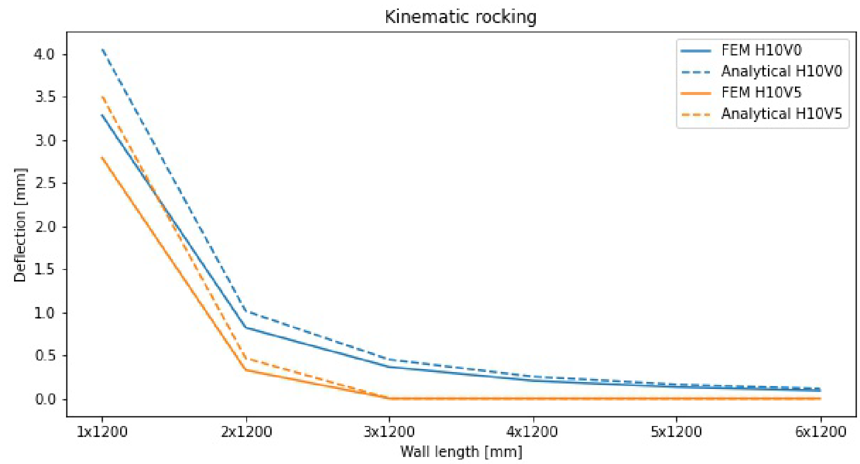


Figure 4-4: Kinematic rocking when calculating total deflection of LTF.

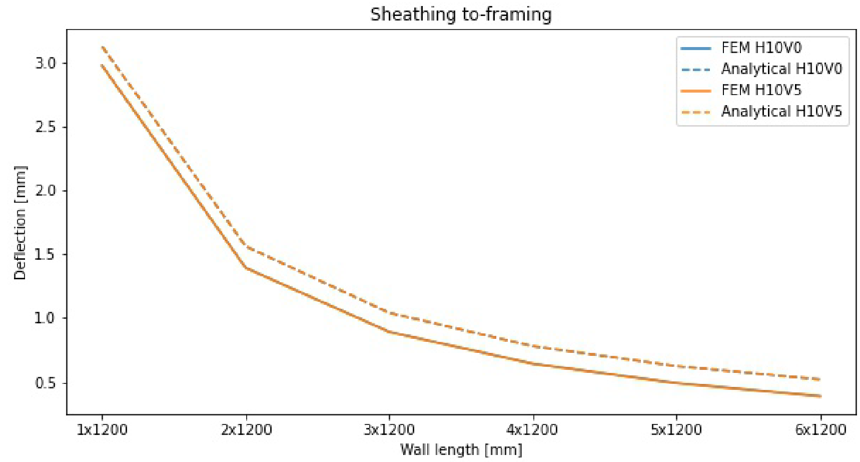


Figure 4-5: Sheathing-to-framing deflection when calculating total deflection of LTF.

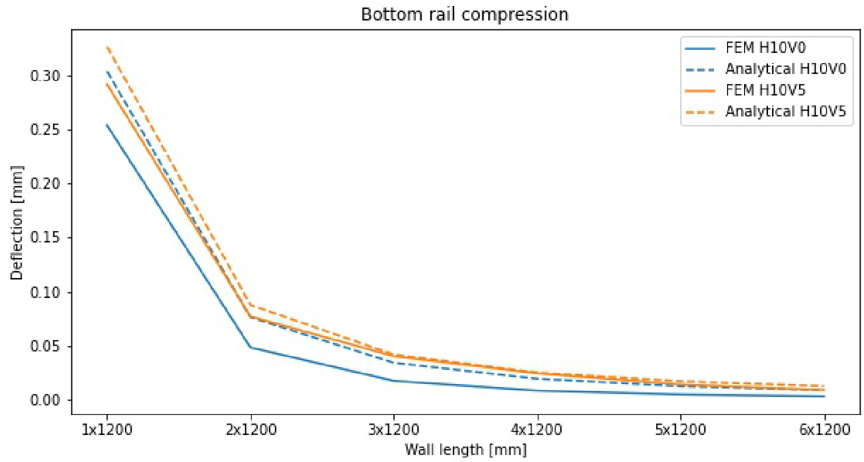


Figure 4-6: Bottom rail compression when calculating total deflection of LTF.

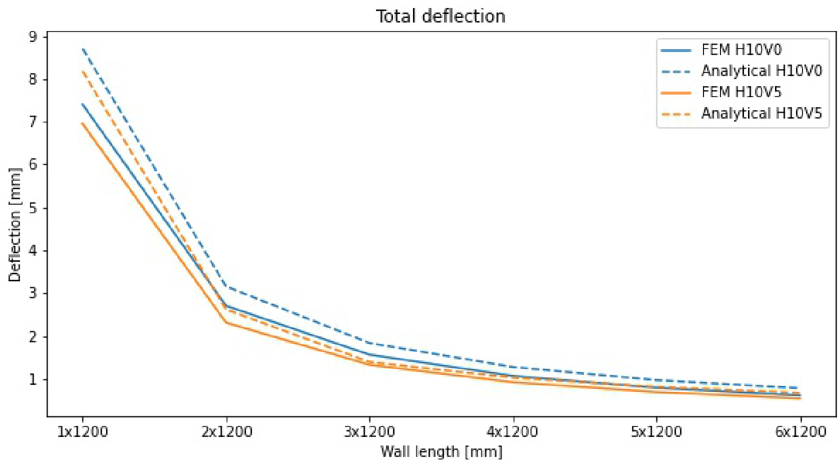


Figure 4-7: Total deflection of LTF.

The figures above illustrate results of lateral deflection of LTF shear walls. Lateral deflections in millimeter [mm] are shown in the vertical axes. Wall length is shown in mm at the horizontal axes as number of consecutive sheathing panels multiplied by length of one panel. Both results from analytical and numerical calculations are illustrated in order to present a comparison between the two models. The numerical results are plotted as continuous curves, whereas the analytical are plotted as dashed curves. Different parameters are plotted with colors, in this case different loading. H10V0 represents a horizontal load of 10 kN and zero vertical load, whereas H10V5 represents a horizontal load of 10 kN and a vertical load of 5 kN/m.

- From Figure 4-1, in-plane shear matches up almost perfectly, and there is no difference with or without vertical loading.
- From Figure 4-2, analytical calculations of in-plane bending seems to be somewhat conservative in comparison to the numerical calculations, and there is almost no difference with or without vertical loading.
- From Figure 4-3, rigid body sliding match up perfectly, and there is no difference with or without vertical loading.
- From Figure 4-4, the kinematic rocking does not match up, and a difference is observed on the lateral deflection with the addition of vertical loading.
- For sheathing-to-framing, form Figure 4-5, there is a difference between the two models where analytical seem to be conservative in comparison to the numerical. There can also be seen no difference with or without vertical loading.
- Bottom rail compression is one of the smallest contributions. From Figure 4-6, the analytical model also looks to be conservative in comparison to the numerical model.
- From Figure 4-7, it can be observed that the total deflection of analytical and numerical calculations are generates similar results, where analytical is somewhat conservative in comparison to the numerical.

4.1.2 Lateral Deflection due to In-plane Shear

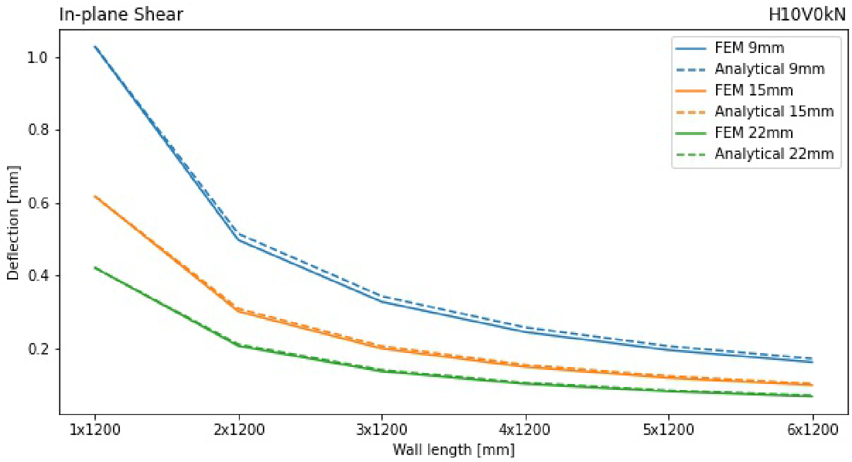


Figure 4-8: Lateral deflection of LTF due to in-plane shear.

Figure 4-8 illustrates the results for the lateral deflection due to in-plane shear of a LTF shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the thickness of the sheathing panels. Increasing loading do not affect the difference between the two models. Results from increasing loading can be found in Appendix A.

It can be observed that the analytical and numerical models match up almost perfect for all the different parameters.

4.1.3 Lateral Deflection due to In-plane Bending

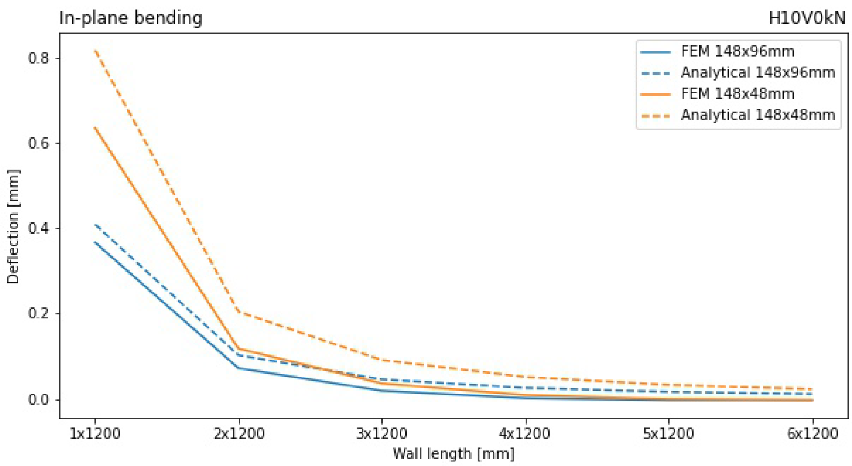


Figure 4-9: Lateral deflection of LTF due to in-plane bending.

Figure 4-9 illustrates the results for the lateral deflection due to in-plane bending of a LTF shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the cross sections of the external studs. Increasing loading does not affect the difference between the two models. Results from increasing loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up and the analytical model seem to be conservative in comparison to the numerical model. Also, a smaller cross section produces a bigger deviation between analytical and numerical calculations.

4.1.4 Lateral Displacement due to Rigid Body Sliding of the Shear Wall

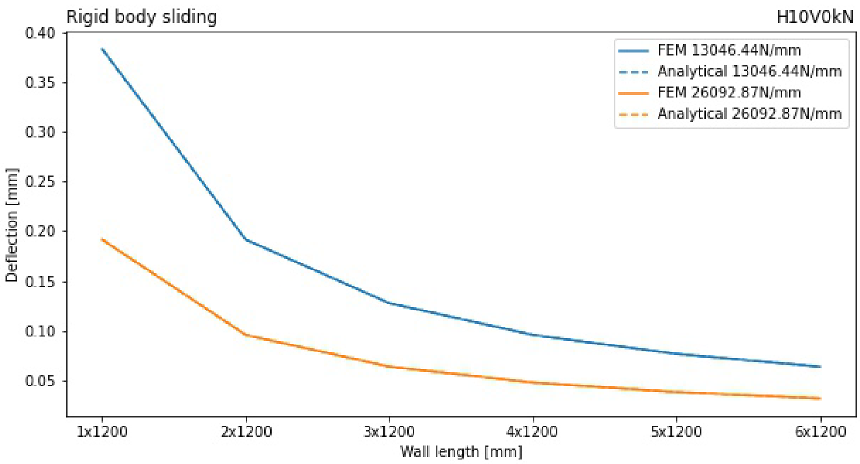


Figure 4-10: Lateral deflection of LTF due to rigid body sliding.

Figure 4-10 illustrates the results for the lateral deflection due to rigid body sliding of a LTF shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different stiffnesses used for the angle brackets. Increasing loading do not affect the difference between the two models. Results from increasing loading can be found in Appendix A.

It can be observed that the analytical and numerical models match up perfectly for all the different parameters. When results are spot on, the dashed curve overlaps with the continuous curve.

4.1.5 Lateral Displacement due to Kinematic Rocking of the Shear Wall

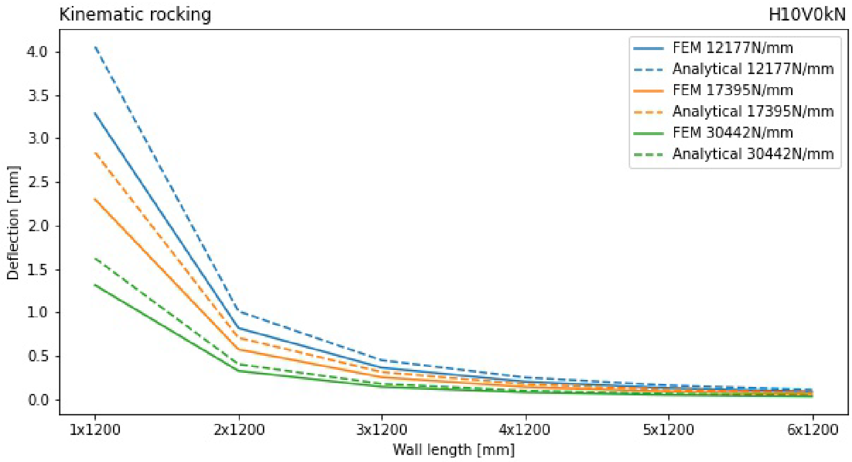


Figure 4-11: Lateral deflection of LTF loaded with H10V0 due to kinematic rocking.

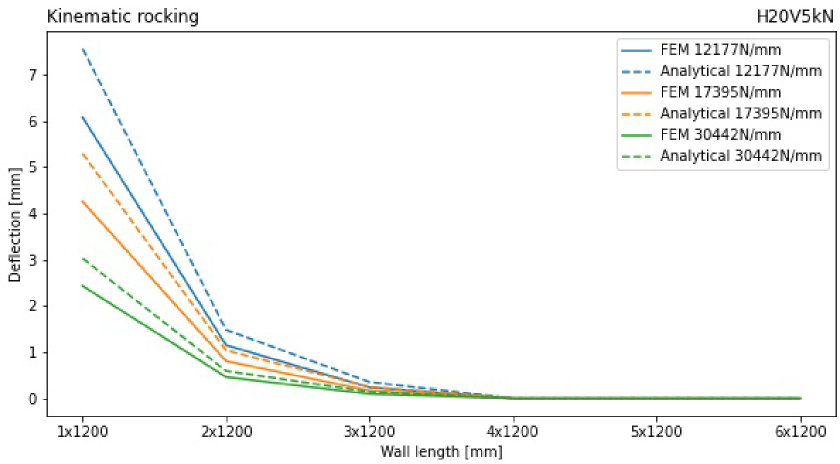


Figure 4-12: Lateral deflection of LTF loaded with H20V5 due to kinematic rocking.

Table 4-1: Difference for kinematic rocking of LTF with k = 12177 N/mm.

Difference between analytical and numerical models with hold-down stiffness, k = 12177 N/mm

Wall length [mm]	H = 10 kN, V = 0 kN/m	H = 20 kN, V = 5 kN/m
1x1200	19,00 %	19,65 %
2x1200	18,98 %	22,31 %
3x1200	18,97 %	32,84 %
4x1200	18,95 %	0,00 %
5x1200	18,93 %	0,00 %
6x1200	18,91 %	0,00 %

Table 4-2: Difference for kinematic rocking of LTF with k = 17395 N/mm.

Difference between analytical and numerical models with hold-down stiffness, k = 17395 N/mm

Wall length [mm]	H = 10 kN, V = 0 kN/m	H = 20 kN, V = 5 kN/m
1x1200	19,00 %	19,65 %
2x1200	18,98 %	22,31 %
3x1200	18,97 %	32,84 %
4x1200	18,95 %	0,00 %
5x1200	18,93 %	0,00 %
6x1200	18,91 %	0,00 %

Table 4-3: Difference for kinematic rocking of LTF with k = 30442 N/mm.

Difference between analytical and numerical models with hold-down stiffness, k = 30442 N/mm

Wall length [mm]	H = 10 kN, V = 0 kN/m	H = 20 kN, V = 5 kN/m
1x1200	19,00 %	19,65 %
2x1200	18,97 %	22,29 %
3x1200	18,94 %	32,78 %
4x1200	18,91 %	0,00 %
5x1200	18,88 %	0,00 %
6x1200	18,85 %	0,00 %

Figure 4-11 and Figure 4-12 illustrates the results for the lateral deflection due to kinematic rocking a LTF shear wall. In Figure 4-11 the wall is loaded with a lateral load of 10 kN, whereas in Figure 4-12 the wall is loaded with a lateral load of 20 kN and a vertical load of 5 kN/m. The changed parameter represented by the colours are stiffnesses of the hold-down brackets. Increasing loading do not affect the difference between the two models. Results from increased load can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up and the analytical model seem to be conservative. The analytical model is consistently around 18-19 % larger than the numerical model when the shear wall is only affected by lateral forces. When adding vertical load there are still matching differences, however, the difference increases with the length of the wall. The stiffness of the hold-down brackets makes a neglectable difference. This is illustrated in Table 4-1, Table 4-2 and Table 4-3.

4.1.6 Lateral Displacement due to Deformation of Sheathing-to-framing Connections

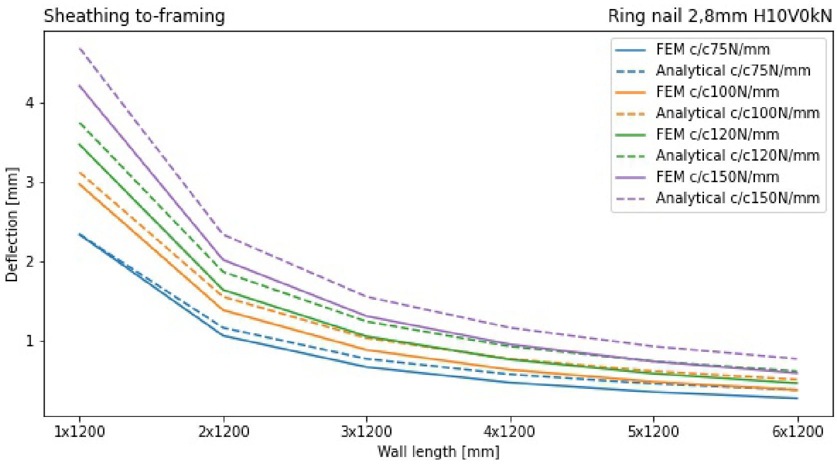


Figure 4-13: Sheathing-to-framing deflection with 2,8mm ring nail diameter.

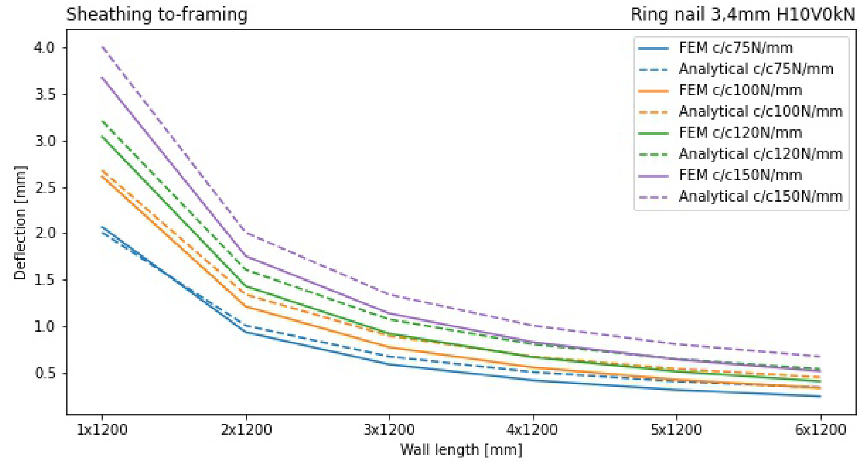


Figure 4-14: Sheathing-to-framing deflection with 3,4mm ring nail diameter.

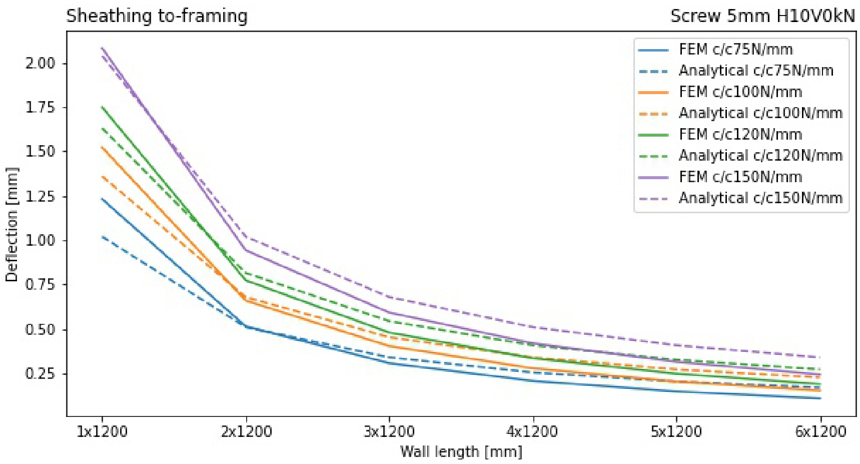


Figure 4-15: Sheathing-to-framing deflection with 5mm screw diameter.

Figure 4-13, Figure 4-14 and Figure 4-15 illustrates the results for the lateral deflection due to the deformation of sheathing-to-framing connections for a LTF shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different nail spacings. Increasing load do not affect the difference between the two models. Results from increased loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up. Furthermore, all the different parameters tested made an impact on the deviation between the analytical and numerical calculations. The analytical model led to grater deflections compared to the numerical model both when the stiffness of the fasteners reduces, and when the spacing between the fasteners increases. As well as when the length of the wall increases.

4.1.7 Lateral Displacement due to Deformation of the Bottom Rail Perpendicular to the Grain under the Trailing Stud

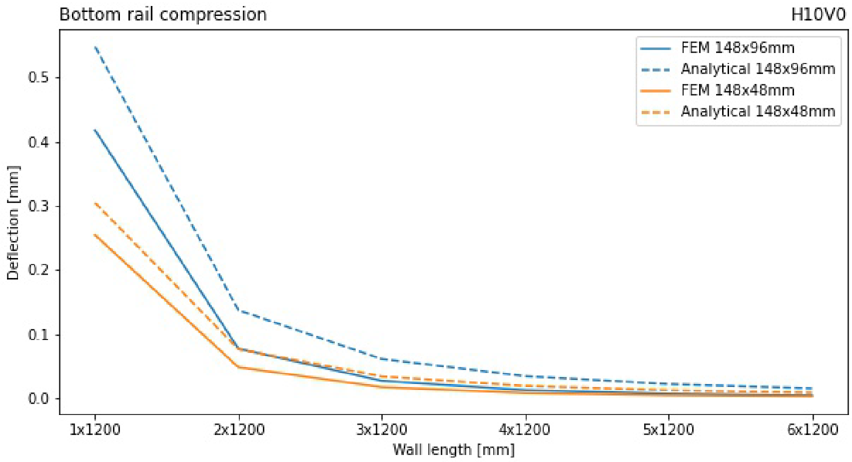


Figure 4-16: Bottom rail deflection loaded with H10V0.

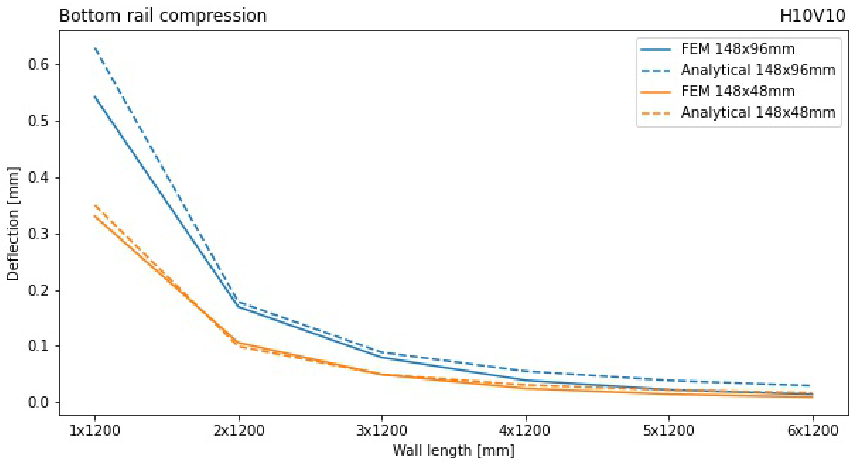


Figure 4-17: Bottom rail deflection loaded with H10V10.

Figure 4-16 and Figure 4-17 illustrates the results of lateral deflection due to deformation of the bottom rail perpendicular to the grain under the trailing stud. In Figure 4-16 the wall is loaded with a lateral load of 10 kN, whereas in Figure 4-17 the wall is loaded with a lateral load of 10 kN and a vertical load of 10 kN/m. Parameter represented by the colours are the cross sections of the bottom rail. Increasing load do not affect the difference between the two models. Results from increased loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up and the analytical model seem to be more conservative with decreased vertical loading. When only vertical load increases, the difference between the two model gets smaller. Also, a larger cross section produces a larger difference between analytical and numerical calculations.

4.2 Parametric Analysis of Monolithic CLT Shear Walls

A selection of the result from the parametric analyses performed on the monolithic CLT shear walls are presented in this chapter. Only the results of most relevance are presented, while all results can be found in Appendix A.

4.2.1 Total Deflection

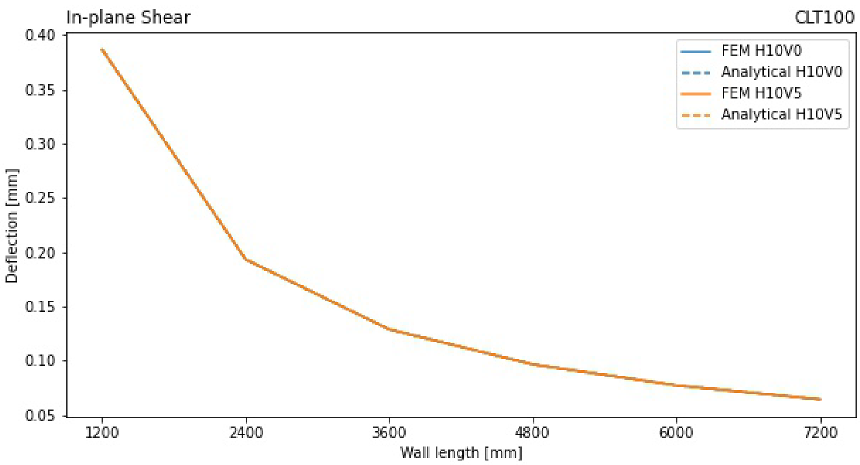


Figure 4-18: In-plane shear when calculating total deflection of monolithic CLT100.

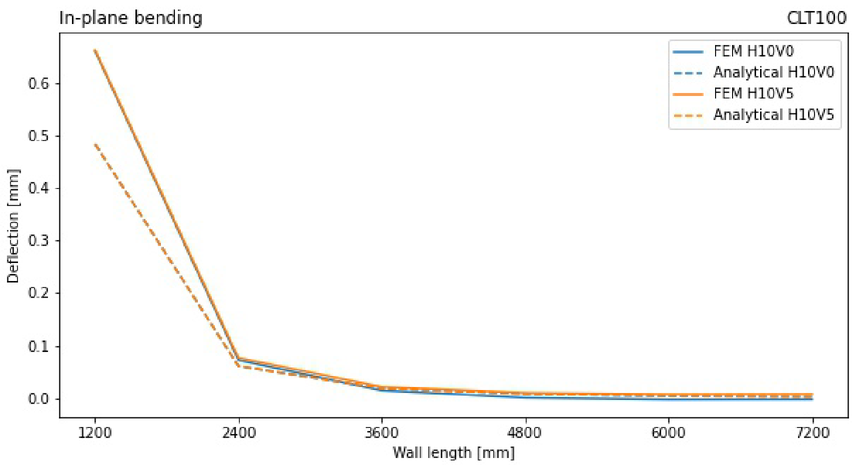


Figure 4-19: In-plane bending when calculating total deflection of monolithic CLT100.

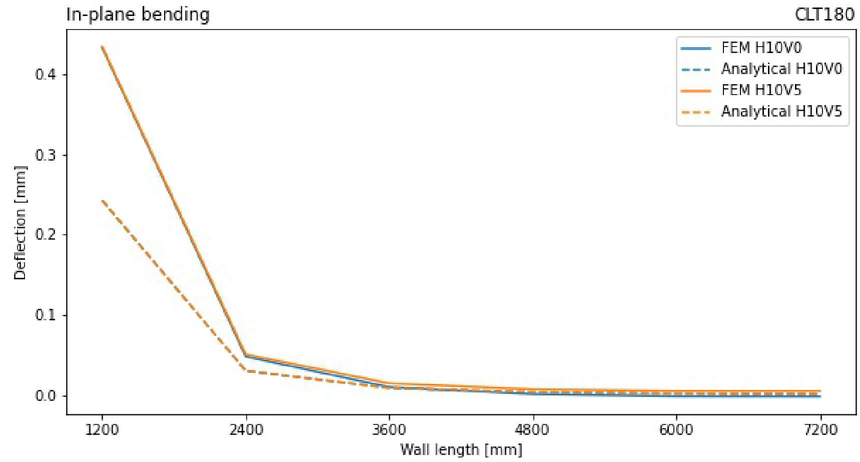


Figure 4-20 In-plane bending when calculating total deflection of monolithic CLT180.

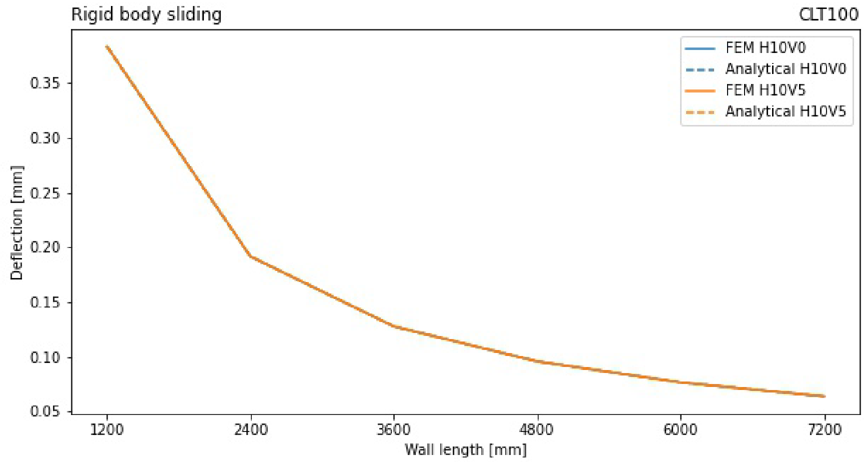


Figure 4-21: Rigid body sliding when calculating total deflection of monolithic CLT00.

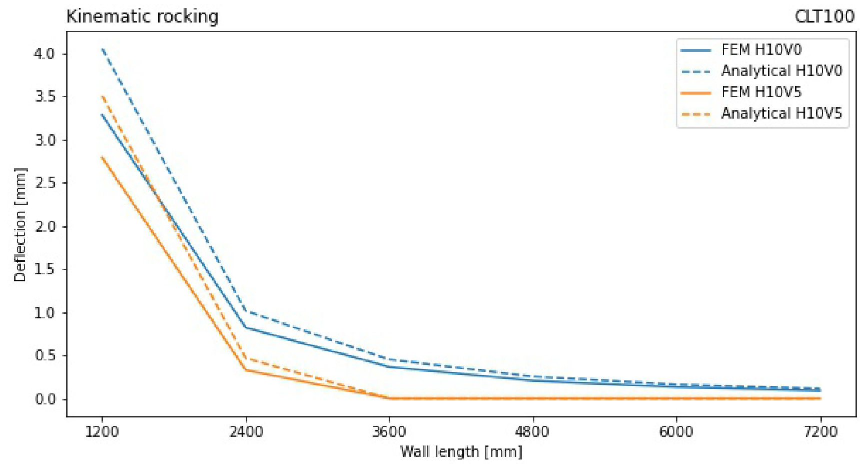


Figure 4-22 Kinematic rocking when calculating total deflection of monolithic CLT100.

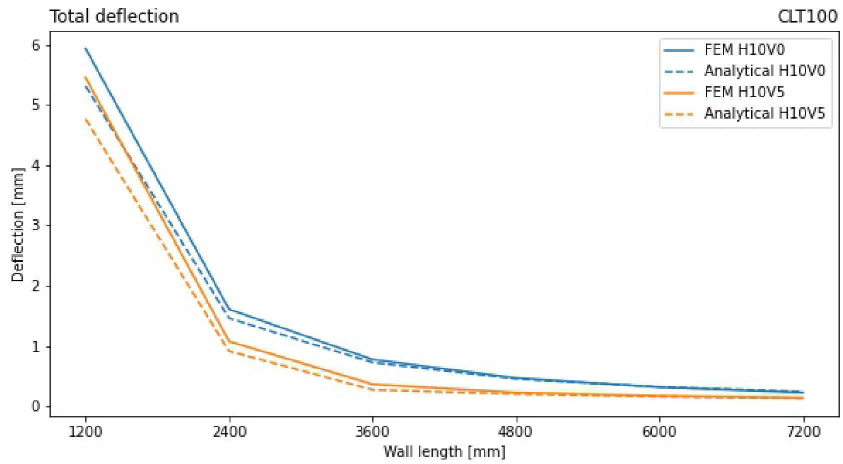


Figure 4-23: Total deflection of monolithic CLT100.

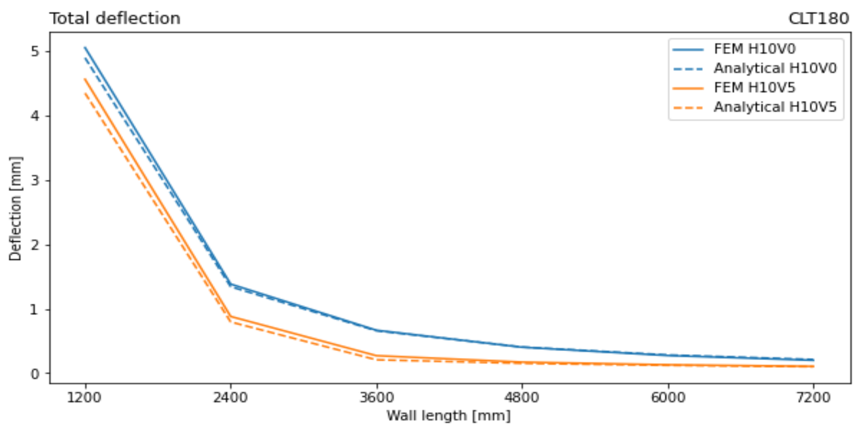


Figure 4-24: Total deflection of monolithic CLT180.

- In-plane shear match up perfectly, as shown in Figure 4-18. There is no difference with or without vertical loading.
- From Figure 4-19 and Figure 4-20, In-plane bending calculated by the numerical model generated larger deflections then analytical calculations. Also, there is a very small difference with or without vertical loading which the analytical model does not consider.
- Rigid body sliding match up perfectly, as shown in Figure 4-21. Also, there is no difference with or without vertical loading.
- From Figure 4-22, the kinematic rocking does not match up. Addition of vertical loading make a difference.
- From Figure 4-23 and Figure 4-24, it can be observed that the total deflection of analytical and numerical calculations are generates similar results.

4.2.2 Lateral Deflection due to In-plane Shear

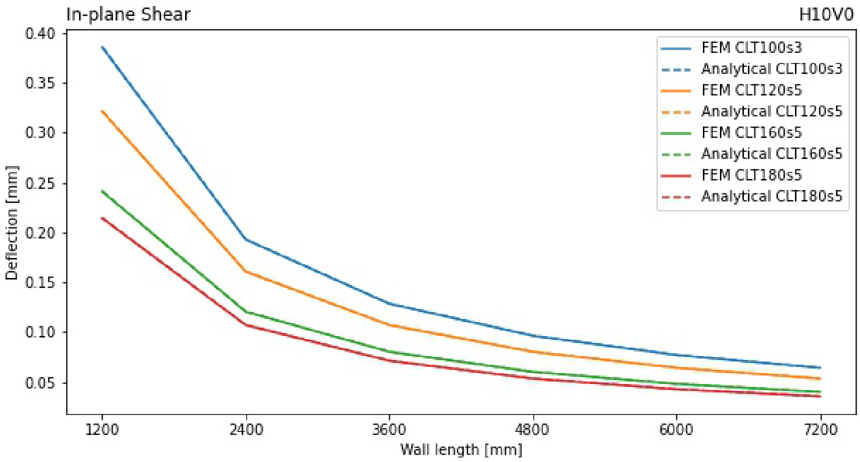


Figure 4-25: Lateral deflection of monolithic CLT due to in-plane shear.

Figure 4-25 illustrates the results for the lateral deflection due to in-plane shear of a CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different dimensions of CLT panels. Different loading does not affect the difference between the two models. Results from different loading can be found in Appendix A.

It can be observed that the analytical and numerical models perfectly match for all different parameters. When results are spot on, the dashed curve overlaps with the continuous curve.

4.2.3 Lateral Deflection due to In-plane Bending.

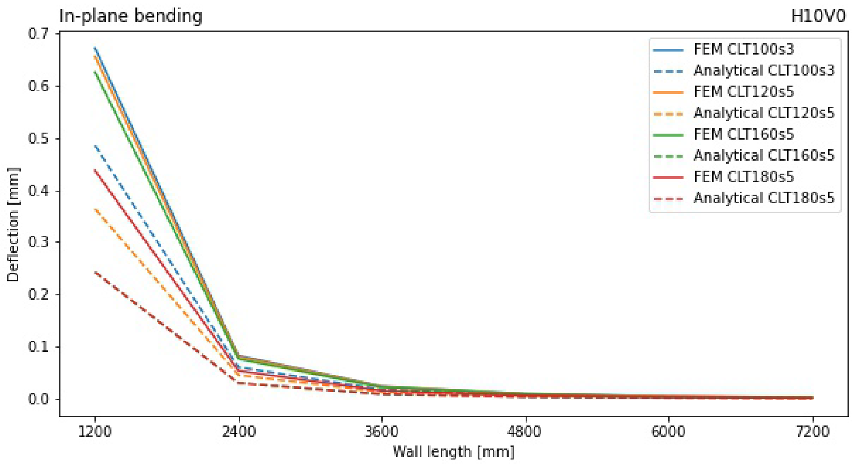


Figure 4-26: Lateral deflection of monolithic CLT due to in-plane bending.

Figure 4-26 illustrates the results for the lateral deflection due to in-plane bending of a CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different dimensions of CLT panels. Different loading does not affect the difference between the two models. Results from different loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up and analytical model seem to be radical. Larger panel cross sections produce an increase in difference between analytical and numerical calculations. The numerical model also produces very similar results for the different panels, except for CLT180s5, whereas the analytical has a linear difference.

4.2.4 Lateral Displacement due to Rigid Body Sliding of the Shear Wall

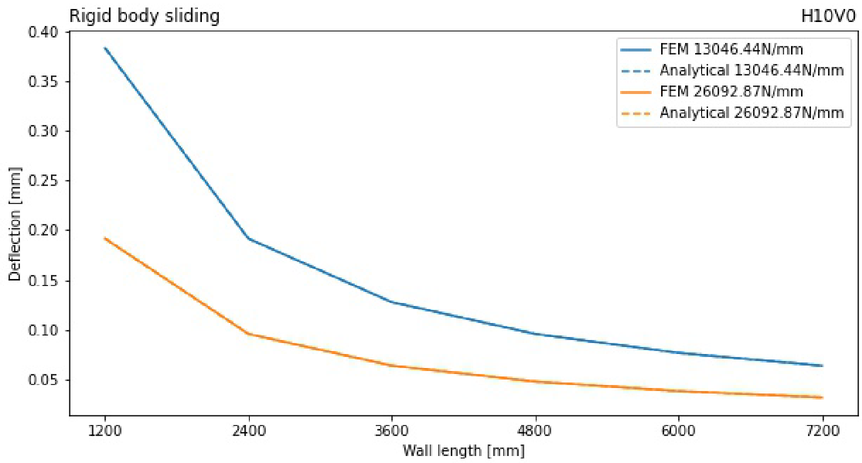


Figure 4-27. Lateral deflection of monolithic CLT due to rigid body sliding.

Figure 4-27 illustrates the results for the lateral deflection due to rigid body sliding of a CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different stiffness used for the angle brackets. Increasing load does not affect the difference between the two models. Results from increased loading can be found in Appendix A.

It can be observed that the analytical and numerical models match up perfectly for all the different parameters. When results are spot on, the dotted line overlaps with the consecutive line.

4.2.5 Lateral Displacement due to Kinematic Rocking of the Shear Wall

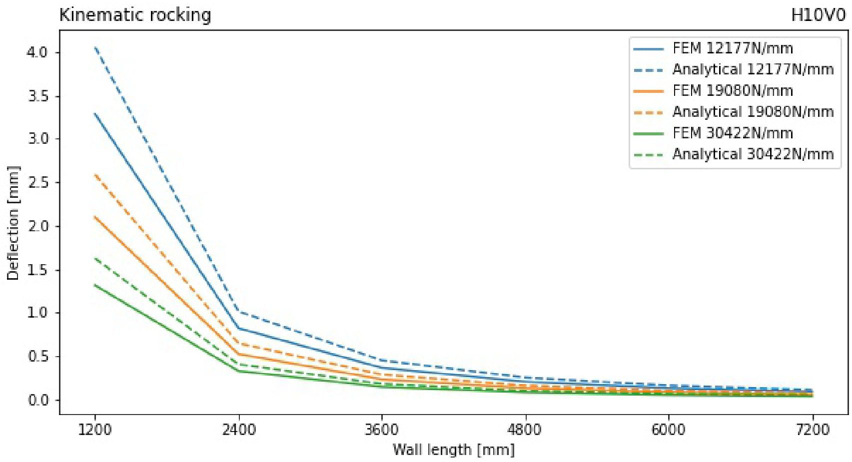


Figure 4-28: Lateral deflection of monolithic CLT with H10V0 due to kinematic rocking.

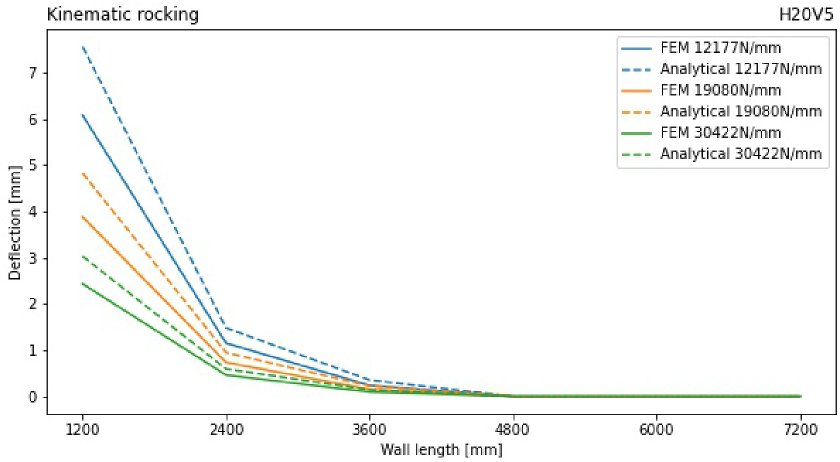


Figure 4-29: Lateral deflection of monolithic CLT with H20V5 due to kinematic rocking.

Table 4-4: Difference for kinematic rocking of monolithic CLT with k = 12177 N/mm.

Difference between analytical and numerical models with hold-down stiffness, k = 12177 N/mm

Wall length [mm]	H = 10 kN, V = 0 kN/m	H = 20 kN, V = 5 kN/m
1x1200	18,99 %	19,64 %
2x1200	18,99 %	19,64 %
3x1200	18,99 %	19,65 %
4x1200	19,00 %	22,32 %
5x1200	19,00 %	0 %
6x1200	19,00 %	0 %

Figure 4-28 and Figure 4-29 illustrates the results for lateral deflection due to kinematic rocking of a monolithic CLT shear wall. In Figure 4-28 the wall is loaded with a lateral load of 10 kN, whereas in Figure 4-29 the wall is loaded with a lateral load of 20 kN and a vertical load of 5 kN/m. Parameter represented by the colours are the stiffness of the hold-down brackets. Increasing load does not affect the difference between the two models. Results from increased loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up. The analytical model consistently generates around 18-19 % larger results than the numerical model when the shear wall is only affected by lateral forces. This is shown in Table 4-4. When adding vertical load there are still matching differences, however, the difference increases with the length of the wall. The variation of the hold-down stiffnesses made a neglectable difference in deviation between the calculation models.

4.3 Parametric Analysis of Segmented CLT Shear Walls

A selection of the results from the parametric analysis performed on the segmented CLT shear walls are presented in this chapter. Only the results of most relevance are presented, while all results can be found in Appendix A.

4.3.1 Total Deflection

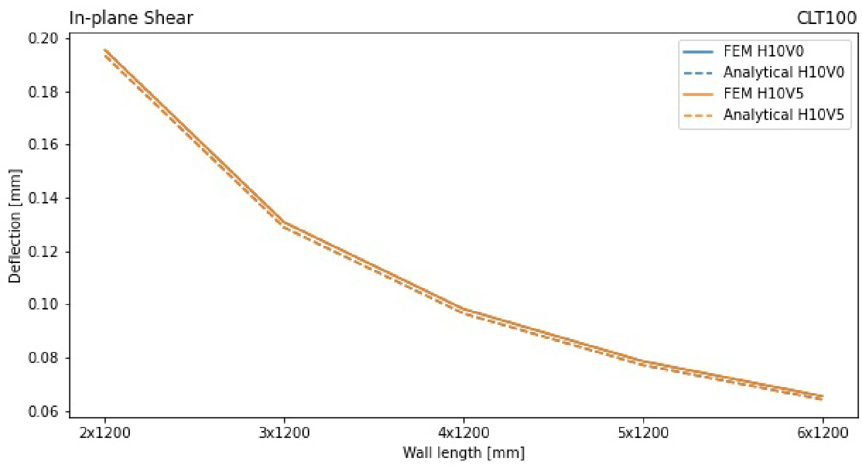


Figure 4-30: In-plane shear when calculating total deflection of segmented CLT100.

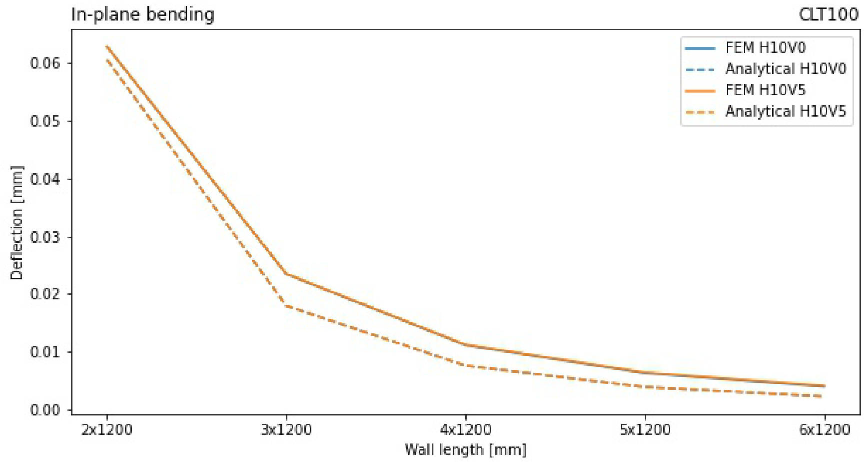


Figure 4-31: In-plane bending when calculating total deflection of segmented CLT100.

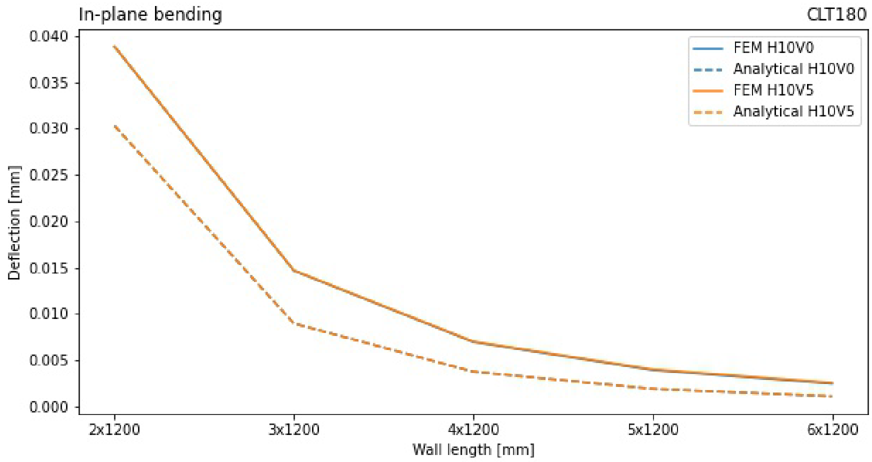


Figure 4-32: In-plane bending when calculating total deflection of segmented CLT180.

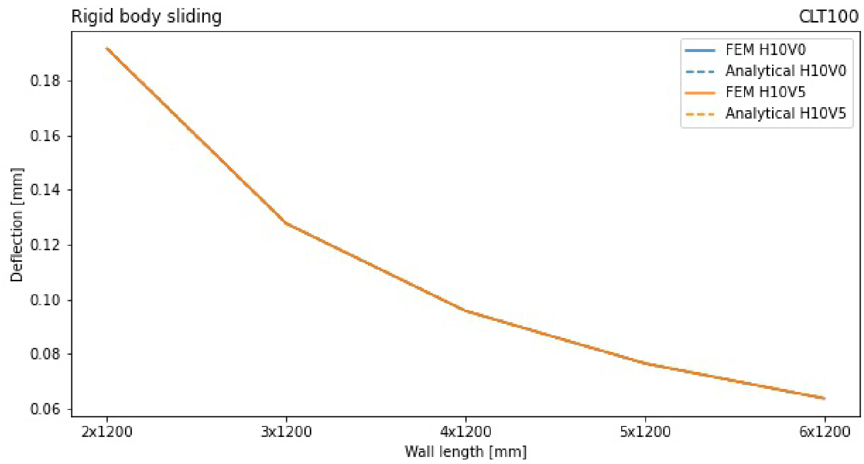


Figure 4-33: Rigid body sliding when calculating total deflection of segmented CLT100.

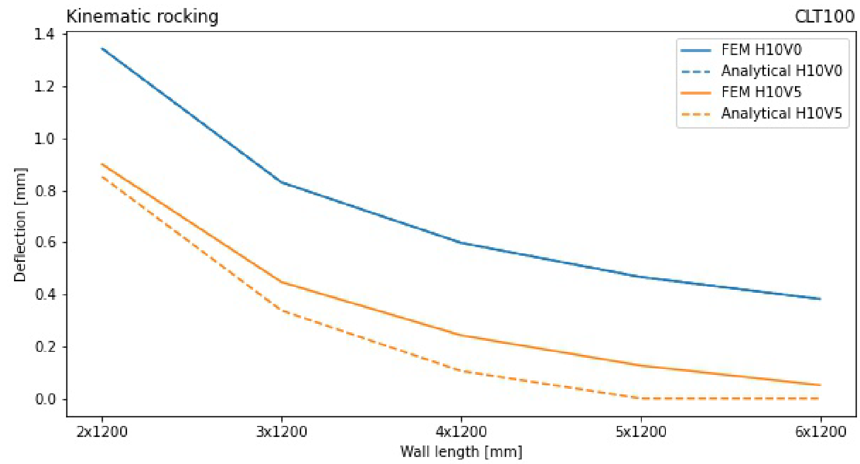


Figure 4-34: Kinematic rocking when calculating total deflection of segmented CLT100.

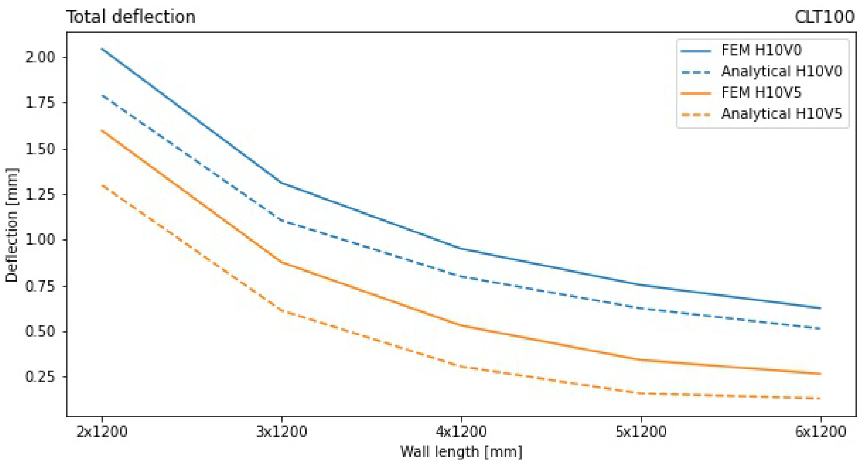


Figure 4-35: Total deflection of segmented CLT100.

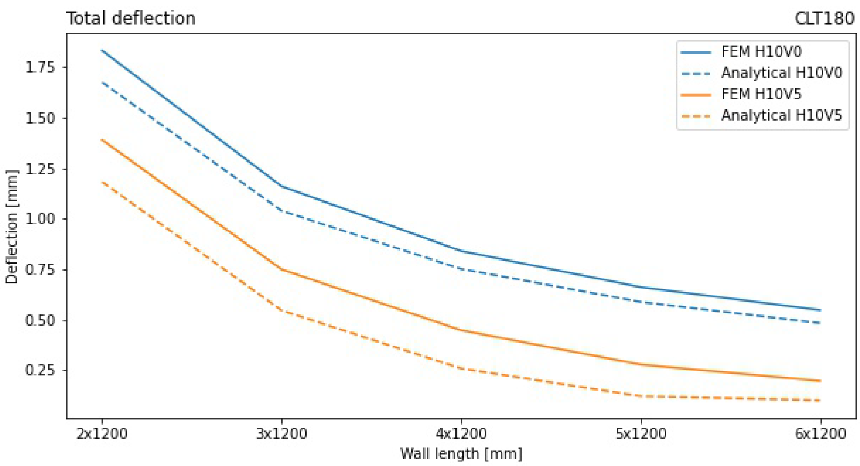


Figure 4-36: Total deflection of segmented CLT180.

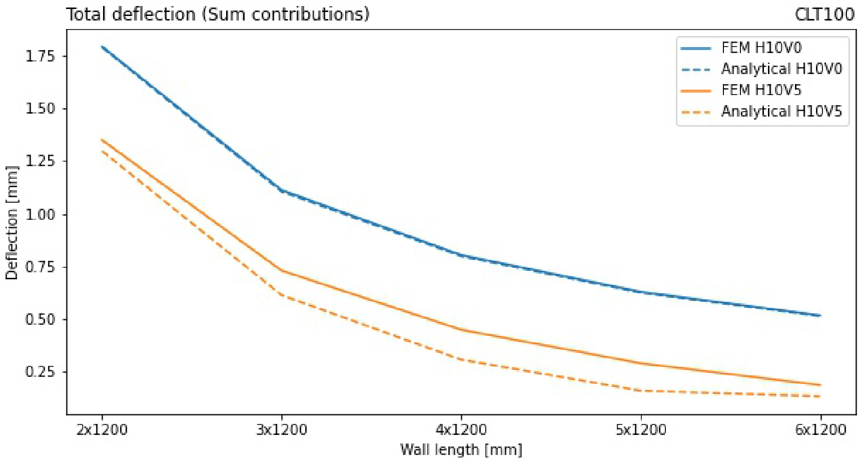


Figure 4-37: Total deflection summed by contributions of segmented CLT100.

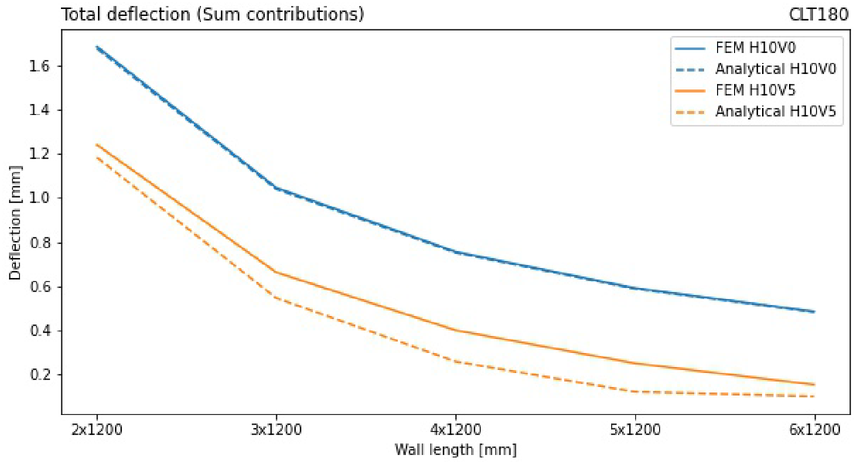


Figure 4-38: Total deflection summed by contributions of segmented CLT180.

- In-plane shear match up almost perfectly, as illustrated in Figure 4-30. There can be seen no difference with or without vertical loading.
- From Figure 4-31 and Figure 4-32, it can be observed that the analytical model generates smaller deflections than the numerical model for in-plane bending. There is no difference with or without vertical loading.
- Rigid body sliding match up perfectly, as shown in Figure 4-33. There can be seen no difference with or without vertical loading.
- Showing in Figure 4-34, kinematic rocking match up perfectly when only exposed to horizontal load, but when exposed to both horizontal and vertical load the models differ.

- The total deflections generated in the two models are close to each other. However, there is a significant difference when looking at the total deflection given by SAP2000 in Figure 4-35 and Figure 4-36, and the total deflection found by summing up the contributions given by SAP2000 in Figure 4-37 and Figure 4-38.

4.3.2 Lateral Deflection due to In-plane Shear

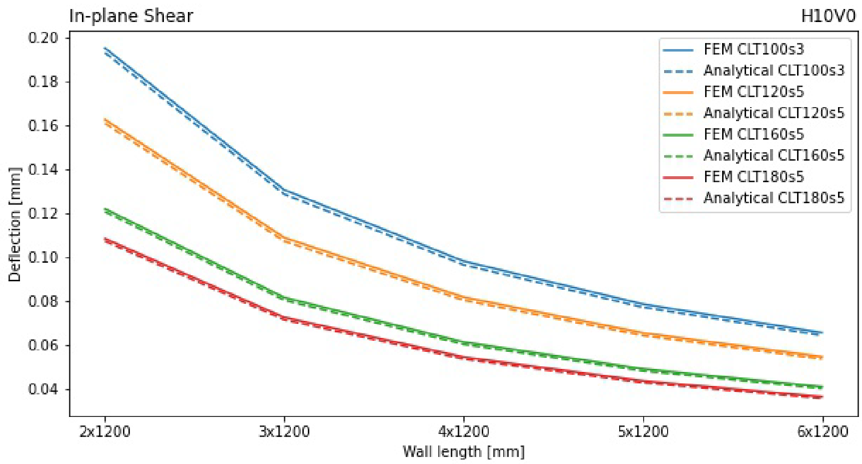


Figure 4-39: Lateral deflection of segmented CLT due to in-plane shear.

Figure 4-39 illustrates the results for the lateral deflection due to in plane-shear of a segmented CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different dimensions of CLT panels. Different loading does not affect the difference between the two models. Results from different loading can be found in Appendix A.

From the analysis it can be seen that the analytical and numerical models match up very close to perfect for all the different parameters.

4.3.3 Lateral Deflection due to In-plane Bending

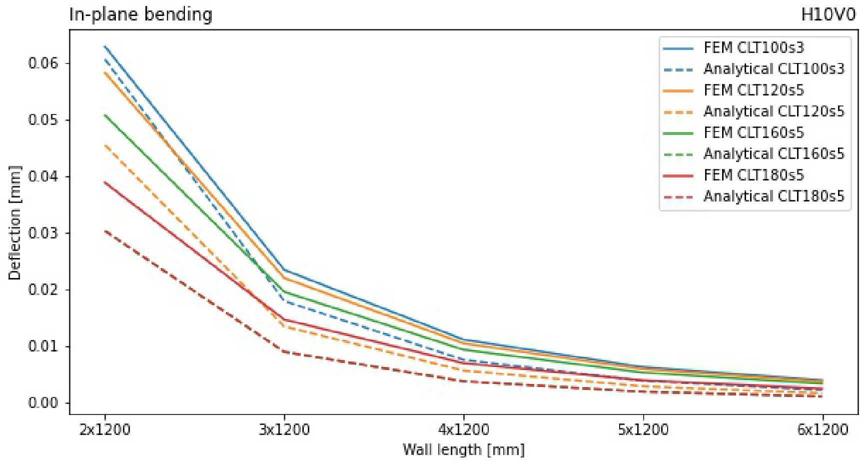


Figure 4-40: Lateral deflection of segmented CLT due to in-plane bending.

Figure 4-40 illustrates the results for the lateral deflection due to in-plane bending of a segmented CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different dimensions of CLT panels. Altering the loading does not affect the difference between the two models. Results from different loading can be found in Appendix A.

It can be observed that the analytical and numerical models do not match up and the analytical model seems to generate smaller deflections than the numerical model. The total thickness of the vertical layers for CLT180s5 and CLT160s5 are the same, and therefore the analytical calculations for the two different panels match up, and the dashed red and green curves are overlapping.

4.3.4 Lateral Displacement due to Rigid Body Sliding of the Shear Wall

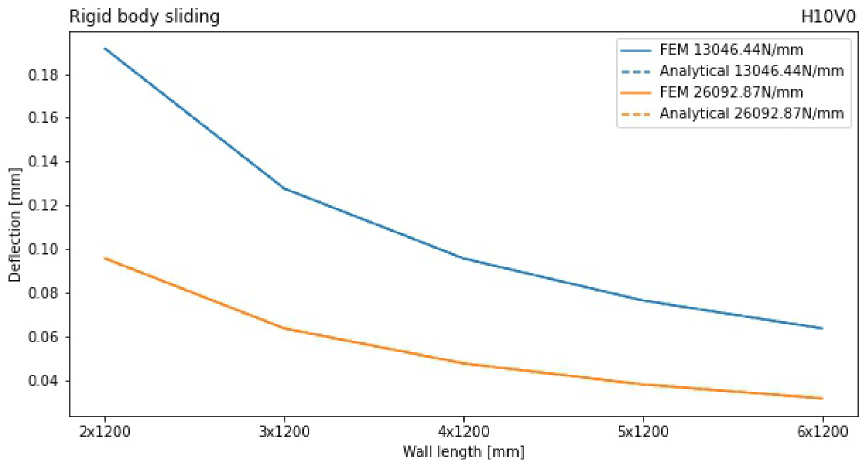


Figure 4-41: Lateral deflection of segmented CLT due to rigid body sliding.

Figure 4-41 illustrates the results for the lateral deflection due to ridged body sliding of a segmented CLT shear wall loaded with a lateral load of 10 kN. Parameter represented by the colours are the different stiffness used for the angle brackets. Increasing load do not affect the difference between the two models. Results from increased loading can be found in Appendix A.

It can be observed that the analytical and numerical models match up perfectly for all the different parameters. When results are spot on, the dashed curve overlaps with the continuous curve

4.3.5 Lateral Displacement due to Kinematic Rocking of the Shear Wall

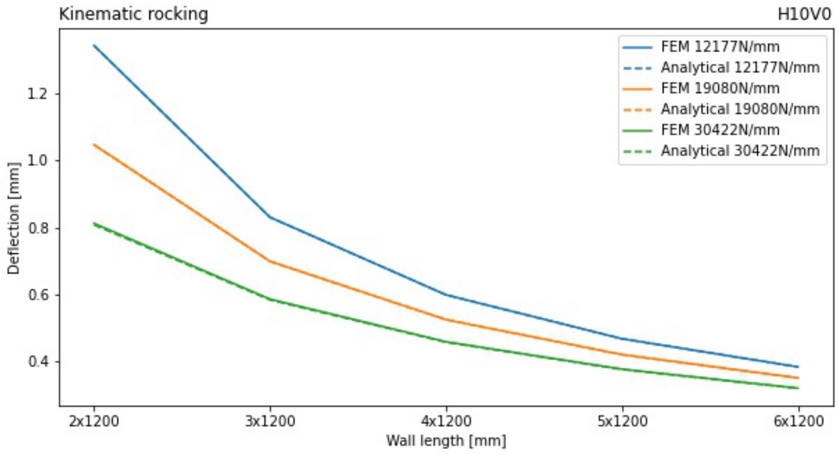


Figure 4-42: Lateral deflection of segmented CLT with H10V0 due to kinematic rocking.

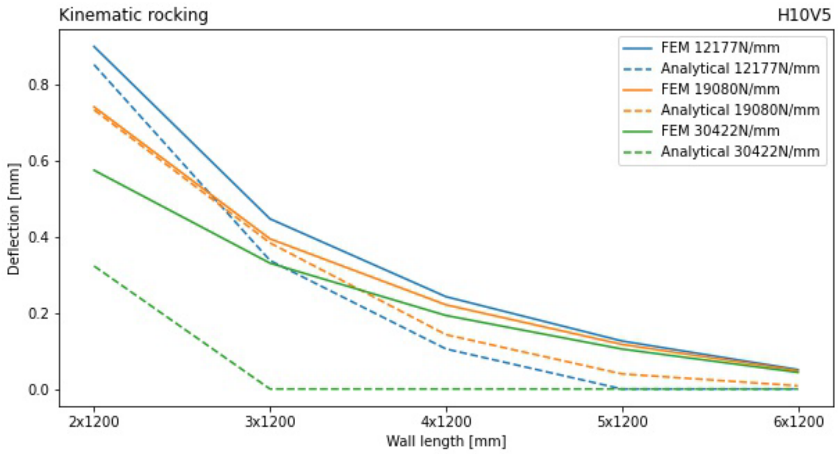


Figure 4-43: Lateral deflection of segmented CLT with H10V5 due to kinematic rocking.

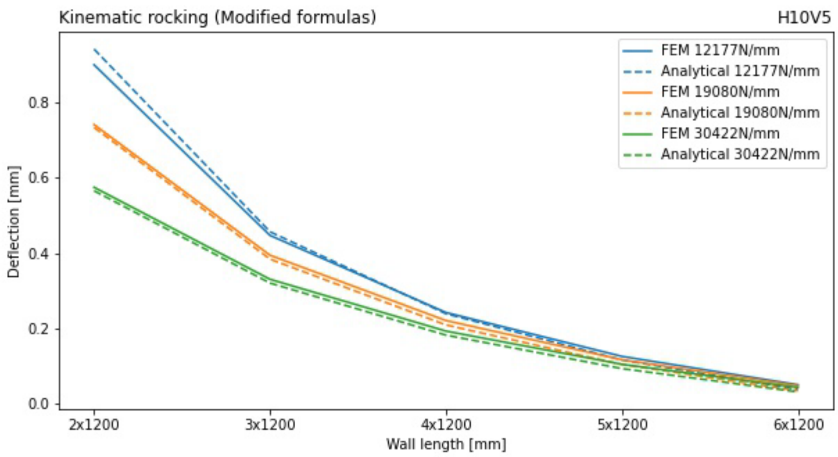


Figure 4-44: Lateral deflection with modified equations of segmented CLT with H10V5 due to kinematic rocking.

Figure 4-42, Figure 4-43 and Figure 4-44 illustrates the results for the lateral deflection due to kinematic rocking a segmented CLT shear wall. In Figure 4-42 the wall is loaded with a lateral load of 10 kN, whereas in Figure 4-43 and Figure 4-44 the wall is loaded with a lateral load of 10 kN and a vertical load of 5 kN/m. Figure 4-44 represents the results from the modified equations presented in chapter 3.2. Parameter represented by the colours are the stiffness of the hold-down brackets. Increasing load does not affect the difference between the two models. Results from increased loading can be found in Appendix A.

The analytical equations presented in the upcoming Eurocode 5 is perfectly matching the numerical model when subjected only to lateral forces. However, when including vertical load, the equations are way off. The modified equations on the other hand, match up almost perfectly.

4.4 Summary of the Results from the Parametric Analysis

Many of the analytical equations match up close to the numerical calculations from SAP2000. For all the shear wall technologies comparison of the in-plane shear between numerical and analytical models matches eighter perfect or very close to perfect, and rigid body sliding comparison gives results that is indistinguishable.

The analytical equation for kinematic rocking is somewhat conservative for both LTF and monolithic CLT shear wall with a difference of about 18-19 % between analytical and numerical models. As for segmented CLT shear wall, the analytical equations presented in the upcoming Eurocode 5 is perfectly matching the numerical model when only subjected to lateral forces. However, when including vertical load, the equations are way off. On the other hand, the modified equations presented in chapter 3.2 almost match up perfectly.

Results from both LTF and the two CLT shear walls for lateral deflection due to in-plane bending indicate that the numerical and analytical calculations do not match up. There can be seen no consistency in the results, where the analytical equation looks to be conservative when analysing the LTF, and radical when analysing the CLT shear walls.

Both lateral deflection due to deformation of sheathing-to-framing connections and lateral deflection due to deformation of the bottom rail perpendicular to the grain under the trailing stud only apply to LTF. The analytical equation for the sheathing-to-framing deflection do not match up to the numerical. Furthermore, all the different parameters tested made an impact on the deviation between the analytical and numerical calculations. The analytical model led to greater deflections compared to the numerical model both when the stiffness of the fasteners reduces, when the spacing between the fasteners increases, as well as when the length of the wall increases. The analytical equation for deflection due to compression perpendicular to the grain seem to be conservative when inflicted with less vertical load. When only vertical load increases, the deviation between the two model gets smaller. Furthermore, cross section seems to have an impact on difference between analytical and numerical models.

4.5 Modal Analysis and Finite Element Model Updating

4.5.1 Meshing Sensitivity

The range analysed is from 2000mm to 100mm with increments of 100mm, and it is limited by the capacity of the software and the computers running the analysis. The densest mesh gives an execution time of just over 3000 seconds on the hardware that is used, while the all the mesh sizes larger than 800mm gives around 14 seconds of calculation. The goal of an analysis like this is to find a threshold where the result from the changing mesh converges. However, results from meshing density analysis do not converge in the range that is possible to analyse with the tools available. Furthermore, it can be observed in Figure 4-45 that the calculated eigenfrequency declines with denser mesh. This makes the choice of mesh hard, however, to keep execution time in a sensible range a maximum 500mm mesh in all directions are chosen for the modal analysis.

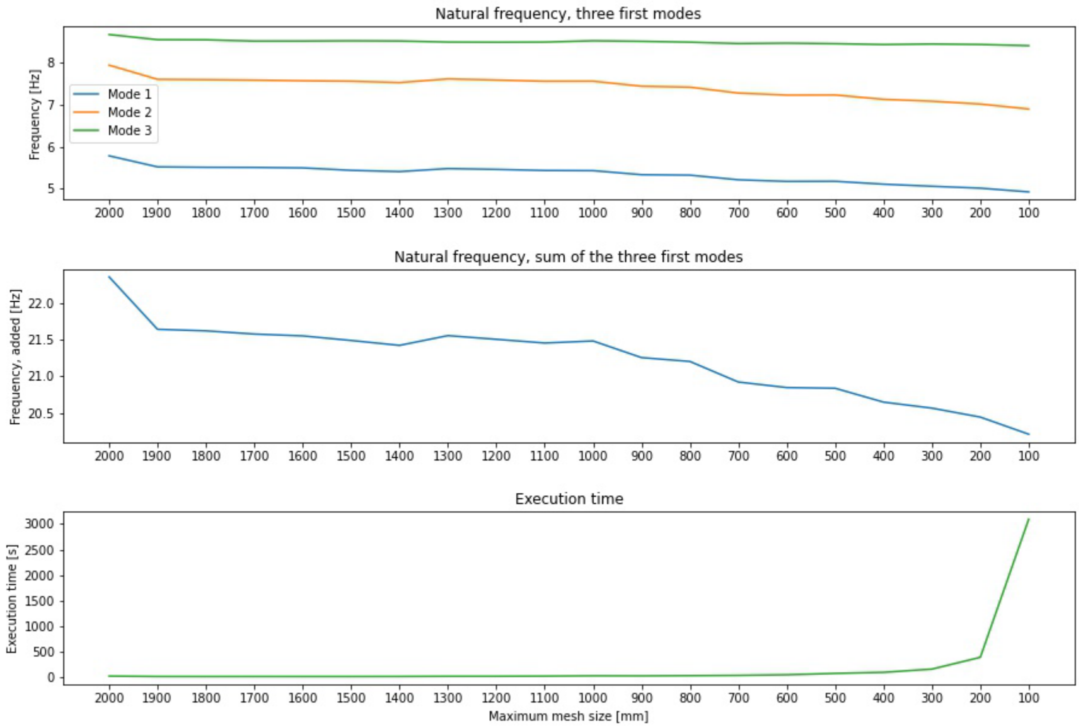


Figure 4-45: Result of mesh density analysis.

4.5.2 Finite Element Model Updating

The updating procedure gave additional masses presented in Table 4-5 for the roof, internal floors, and facades. All the updated results are extracted using the values in Table 4-5.

Table 4-5: Additional mass after FE updating.

Additional Mass		
Floors [kN/m^2]	Roof [kN/m^2]	Facades [kN/m]
0.054	1.880	0.000

The finite element model generated initial frequencies and frequencies after updating shown in Table 4-6 along with the experimental frequencies. The analysis provided the mode shapes for the three modes explored in this thesis. Using the mode shapes, a value of MAC is found, and presented alongside the error contribution of the frequency and lastly the CC_{tot} of both the initial and updated models in Table 4-7 and Table 4-8.

Table 4-6: Experimental, Initial and Updated frequencies

Frequencies from modal identification

Mode.	Experimental frequency. [Hz]	Initial frequency. [Hz]	Updated frequency. [Hz]
1	4,630	5,199	4,024
2	5,566	7,014	5,566
3	6,363	7,286	7,121

Table 4-7: Initial Error Calculation

Initial Error Calculation

Mode	$\frac{\lvert f_{exp} - f_{num} \rvert}{f_{exp}}$	MAC	CC_{tot}
1	0,123	0,954	
2	0,260	0,849	1,277
3	0,145	0,448	

Table 4-8: Updated Error Calculation

Updated Error Calculation			
Mode	$\frac{\lvert f_{exp} - f_{num} \rvert}{f_{exp}}$	MAC	CC_{tot}
1	0,131	0,967	
2	0,000	0,850	0,705
3	0,119	0,728	

The mode shapes generated in the initial and updated modal analysis of the InnoRenew CoE headquarters are visualized as columns in Figure 4-46, Figure 4-47 and Figure 4-48, along with the mode shapes from the experimental campaign for the three modes evaluated. The figures present normalized displacement for each of the five locations of sensors (S1 to S5), in X, Y and Z direction. It is worth noting the similarities between the three result sets for the first and the second mode, while there are little similarities in mode three.

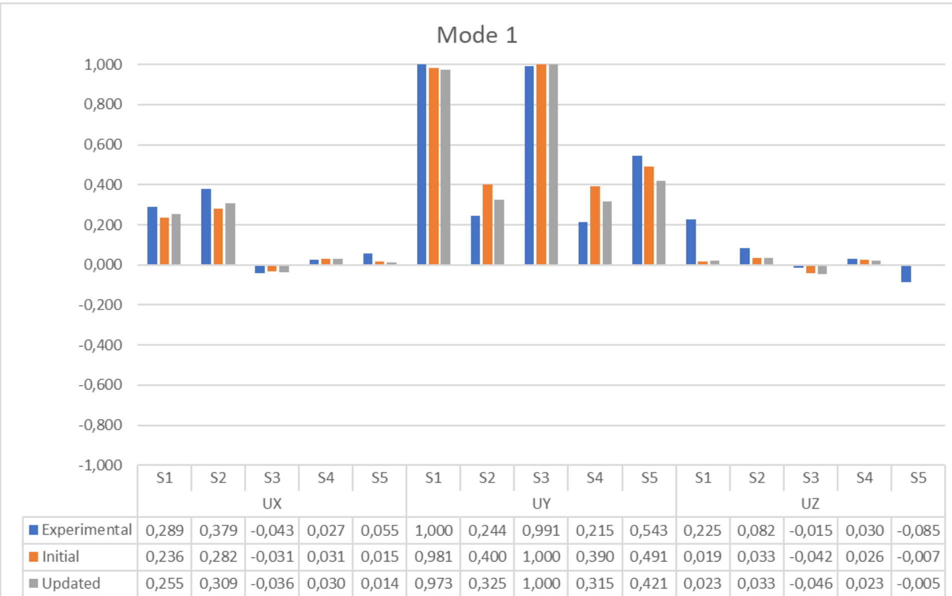
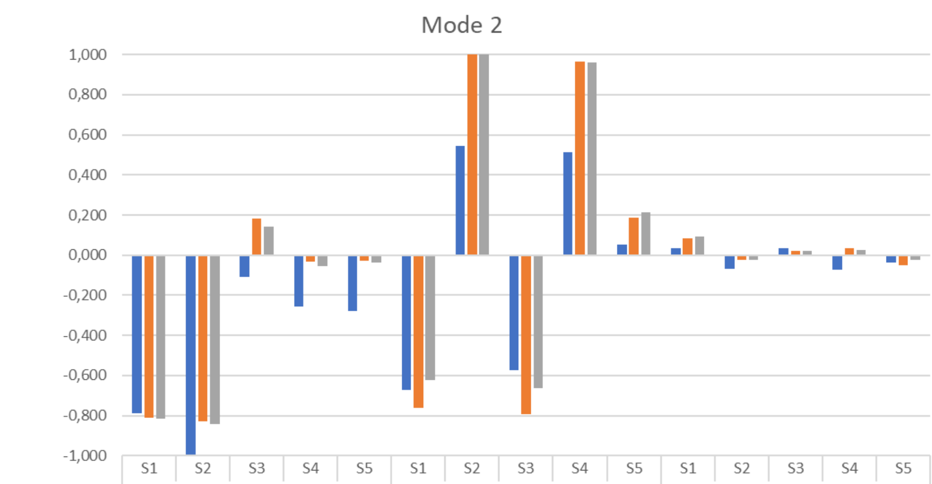
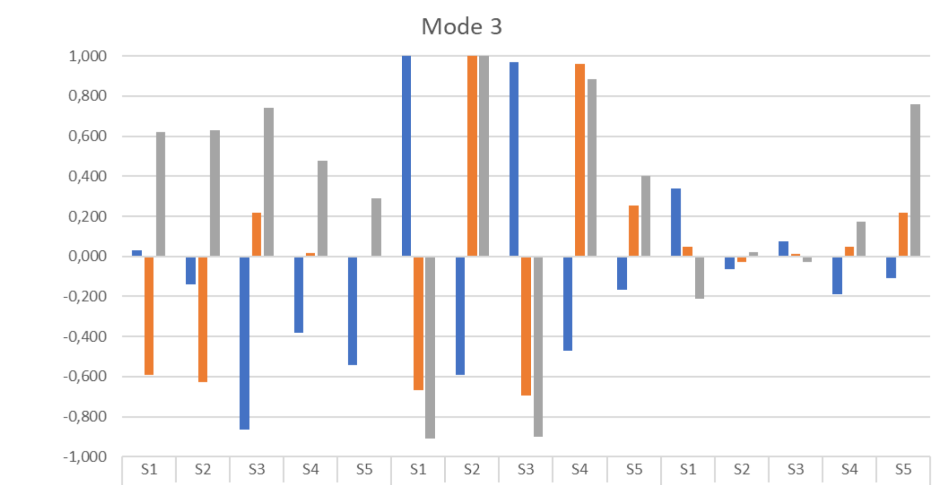


Figure 4-46: Comparison of mode shapes, mode 1.



	UX					UY					UZ				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
Experimental	-0,787	-1,000	-0,107	-0,257	-0,277	-0,671	0,543	-0,572	0,511	0,052	0,036	-0,071	0,034	-0,072	-0,039
Initial	-0,809	-0,829	0,180	-0,033	-0,027	-0,762	1,000	-0,792	0,966	0,185	0,084	-0,024	0,019	0,035	-0,049
Updated	-0,817	-0,844	0,142	-0,055	-0,039	-0,625	1,000	-0,665	0,961	0,212	0,093	-0,026	0,021	0,028	-0,024

Figure 4-47: Comparison of mode shapes, mode 2.



	UX					UY					UZ				
	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
Experimental	0,031	-0,141	-0,867	-0,384	-0,540	1,000	-0,592	0,971	-0,470	-0,167	0,339	-0,063	0,074	-0,191	-0,109
Initial	-0,590	-0,626	0,216	0,019	0,005	-0,670	1,000	-0,696	0,959	0,252	0,047	-0,030	0,011	0,049	0,218
Updated	0,621	0,629	0,740	0,476	0,290	-0,907	1,000	-0,898	0,883	0,400	-0,213	0,023	-0,028	0,174	0,760

Figure 4-48: Comparison of mode shapes, mode 3.

5 Discussions

In this chapter, a discussion and evaluation of the results for both the parametric and the modal analyses are presented. The parametric analysis is discussed and evaluated in chapter 5.1, and the modal analysis in chapter 5.2. Further works are presented in chapter 5.3.

5.1 Parametric Analysis

The parametric analysis of the elastic lateral deflection of the different timber shear walls provided a broad spectrum of results giving a possibility to discuss the strengths and weaknesses of the analytical equations suggested for the upcoming Eurocode 5. The layout of this chapter is divided into subchapters, one for each deflection contribution, as well as a subchapter with an overall discussion of the analysis.

Lateral Deflection due to In-plane Shear.

The lateral deflection due to in-plane shear provided almost identical results from the analytical and numerical calculations both for the LTF and CLT shear wall technologies. As such the reliability of the analytical equations seem to be acceptable.

The similarities begin with the shear modulus given to the shell elements used for CLT panels and the sheathing for the LTF in the numerical calculation are the same as the one used in the analytical calculations. As the deflections of the shear walls are elastic, the analytical calculation of the deflection, as mentioned in chapter 2.3, are based on Hooke's law with linear elastic behaviour, and the stress-strain ratio defining the shearing. This relates to the numerical model which also, as mentioned in chapter 2.4.1, computes shearing from the stress-strain relationship.

Because the shear resistance of the sheathing exceeds the rest of the structure in the LTF technology by significant amount, it looks to be adequate only to calculate the shearing deflection based on the sheathing and neglect the framing as a contributing factor. As the shear resistance of the CLT panels is the only resistance against shear in the CLT technology, the analytical equations are adequate to the model tested. Furthermore, as the CLT panels are modelled as a thin shell in this thesis, the correlation between shear modulus and panel thickness in the analytical and numerical calculations will be identical. In reality, this approach may not be exact, and a layered model which account for lamellas in the CLT panel might be more accurate as it includes longitudinal and perpendicular layers. However, the approach of thin shell is adequate in this thesis as the deflections does not surpass the elastic range and will not reach failure modes such as failure from torsional and un-directional shear stresses in the transfer of shear forces between adjacent layers.

Lateral Deflection due to In-plane Bending.

The analysis of this contribution provided little consistency in the results. The analytical equation seemed to be conservative when analysing the LTF, and radical when analysing the CLT shear walls in comparison to the numerical model. Furthermore, the results gave a lot of variation when cross sections, both for LTF and CLT walls were changed up.

As mentioned in chapter 2.3, the analytical equations are equal to the Euler-Bernoulli deflection for a cantilever beam. One can ponder on the transferability of calculation models from Euler-Bernoulli beam theory to the deflection of shear walls. If the knowledge of the effective bending stiffness, based on modulus of elasticity and the moment of inertia of the shear wall are correct, the Euler-Bernoulli theory might be a correct approach for the deflection due to in-plane bending. However, in the analytical equations, the calculation of the effective moment of inertia is considerably simplified.

In the analytical calculations of bending stiffness for the LTF shear walls, only the cross sections of the perimeter studs and the wall length are accounted for. For CLT shear walls, only the thickness of the vertical layers and the wall length are accounted for. This arranges for a dissimilarity to the numerical approach, where the complete framing resist the bending deflection for LTF, and the complete shell for CLT shear wall technologies.

It is worth noticing that the contribution of lateral deflection due to in-plane bending is relatively small in relation to the total deflection of the shear wall. Thus, the deviation between numerical and analytical calculation might be insignificant relative to the end result and can arguably be neglected.

Lateral Displacement due to Rigid Body Sliding of the Shear Wall.

Throughout the parametric analysis, this contribution has unsurprisingly been the most consistent. With the angle brackets being alike for all shear wall technologies, and the shear wall considered a rigid body, the result match up for all the shear wall technologies.

In the numerical calculations of this contribution in SAP2000, the angle brackets are modelled as linear links, giving deflections in the linear elastic range. This corresponds to the analytical calculation which, as mentioned in chapter 2.3, also are based on linear elastic theory. Both the stiffness and the lateral force are the same in both the numerical and analytical calculations, hence equal deflections. As such the reliability of the analytical equations seem to be acceptable.

Lateral Displacement due to Kinematic Rocking of the Shear Wall.

The analytical model for the lateral displacement due to kinematic rocking is, as mentioned in chapter 2.3, divided in to two different approaches. The first approach corresponds to LTF and monolithic CLT shear walls, while the second corresponds to segmented CLT shear walls.

There is a lot of consistency in the calculations of lateral deflection due to kinematic rocking for the LTF and monolithic CLT shear walls. As the shear walls are modified to be a rigid body and the angle brackets fully stiff in the lateral direction, the lateral deflection caused by kinematic rocking for LTF and monolithic CLT are equivalent. From the analytical calculation as presented in chapter 4.1.5 and 4.2.5, the deflection consistently turned out to be around 18-19% larger than the numerical calculations when subjected only to lateral force. The inclusion of vertical load slightly increased the difference between numerical and analytical calculations. This turned out to be the case for every parameter in the parametric analysis, suggesting the possibility for a reduction factor in the analytical calculation model. Neither change in stiffness of the hold-down brackets or magnitude in loading made any significant impact on the deviation between the numerical and analytical calculations.

The analytical models presented for the lateral deflection due to kinematic rocking for segmented CLT shear walls given in chapter 2.3, were conjected to be inaccurate. Therefore, in chapter 3.2, a set of modified equations based on the initial equations were presented. From the parametric analysis, the conjecture of the inaccuracy of the equations were indicated to be verified. When the shear wall only was subjected to lateral forces the equations both for coupled panel kinematic mode and single-wall kinematic mode presented in chapter 2.3 were close to perfectly matching the numerical models. It was the inclusion of vertical loading in the parametric analysis which seemingly formed these inaccuracies, confirming the conjecture that the second parts of the equations relating to the rocking resistance against vertical loading may be inaccurate.

Both the equations for coupled panel kinematic mode and single-wall kinematic mode were suspected to be inaccurate. However, as presented in chapter 3.2, the suspected inaccuracy in the calculation model for coupled panel kinematic mode was of minor extent. A possible typing error in the draft of the presented chapter 13.7 for the upcoming Eurocode 5 may be the reasoning behind the suspected inaccuracy of this equation. As for the suspected inaccuracy in the calculation model for single-wall kinematic mode, the extent may be greater. In the analytical equations presented, only the stiffness of the hold-down bracket is considered when including the effects of the vertical loading, whereas in the modified equations, the complete rocking stiffness is considered. This includes both the hold-down brackets and the shear connectors.

As presented in chapter 4.3.5, The modified equations matched up close to perfectly in the parametric analysis, while the presented equations matched up when only subjected to lateral force. All kinematic rocking modes, both coupled panel kinematic mode, Intermediate kinematic mode and single-wall kinematic mode were included in the testing, showing the analytical calculation models for all modes were matching numerical calculations. Neither change in stiffness of the hold-down brackets or magnitude in loading made any significant impact on the deviation between the numerical and analytical calculations.

Lateral Displacement due to Deformation of Sheathing-to-framing Connections.

The parametric analysis of this contribution induced very inconsistent results. All the parameters tested, except magnitude of loading, made an impact on the deviation between the analytical and numerical models. As mentioned in chapter 2.3 the analytical model is based on the principle that the framing elements in the shear wall is fully flexible. This is thought to be a conservative approach, and therefor one would expect the analytical model to provide higher deflections than the numerical model. As shown in chapter 4.1.6, this is mostly the case, however, when short walls, high stiffness of the fastener and short spacing between fasteners are implemented the numerical model generates greater deflections than the analytical model. This may be due to the relation between stiffness of the framing and fastening. As the numerical model allows deflections of the framing, which, as mentioned in chapter 2.3, the analytical model does not, the larger deflection of the numerical model may be due to the framing. When stiffness is high and spacing narrow, the deflections from the sheathing-to-framing connections are minor. This allows the inclusion of framing deflection in the numerical model to have a larger impact, making the numerical model generate larger deflections than the analytical model.

Although all parameter in this test changed the deviation between numerical and analytical models, the analytical calculations in most cases generated larger deflections than numerical calculations. This facilitates a foundation for discussing the general reliability of this contribution. As in most cases the analytical calculations are conservative in comparison to the numerical calculations, the analytical equations may be sufficient with some reservations. A note might be included to indicate when the equations are reliable.

Lateral Displacement due to Deformation of the Bottom Rail Perpendicular to the Grain under the Trailing Stud.

Different loading had a big impact on this contribution. The accuracy of the analytical calculations in relation to the numerical calculations seemed to increase when the ratio of vertical loading relative to the lateral loading increased. This indicating equation (2.27), presented in chapter 2.3, might not accurately transfer the lateral loading to the bottom trailing edge of the shear wall.

Total Deflection

The total deflection of the different shear walls calculated by the analytical models matched closely to the numerical model both for LTF and monolithic CLT shear walls. As for Segmented CLT shear walls, the total deflection calculated by the analytical models matched close to perfect to the numerical model when summing up the deflection from each contribution. However, when the deflection is calculated directly, with an un-modified SAP2000 model, there can be seen a deviation between the analytical and numerical calculation.

Although all the contributions for segmented CLT almost match up perfect between numerical and analytical calculations, the total deflection does not. This indicates that there might be something not accounted for in the analytical models, or the numerical calculation model in SAP2000 provides inaccurate results when not modifying parameters. Furthermore, there is a possibility that the model provides these larger deflections because of intertwining between all contributions. When analytically calculating the total deflection, the result comes from linearly summarizing every contribution, whereas in the numerical calculations, all contributions are calculated at once.

Summary of the Parametric Analysis

A total of 1830 comparisons between analytical and numerical calculation models have been conducted throughout the parametric analysis, where 594 on LTF, 564 on monolithic CLT and 672 on segmented CLT shear walls. In this comprehensive analysis, the analytical calculations for the most part provided similar results to the numerical calculations. Furthermore, when the results from the two calculation models deviated, the difference between the two models were not substantial and sufficient foundation for discussing reasoning in the former subchapters were available.

5.2 Modal Analysis

The dynamic identification in this thesis is done on a complex and unregular building. Therefore, the probability of an accurately result in relation to the experimental values in the initial modal analysis of the finite element model were always minor. A complex model can take a lot of time to calculate in an advanced solver like the modal calculations.

Assumptions were made to make the model ready for modal analysis:

- The material properties of CLT are modelled as a shell with monolithic and orthotropic properties of stiffness, instead of layered shell elements to make the calculations less comprehensive.
- The main stairs in the middle of the building are modelled as a frame element with the same dimensions as the rails of the stair.
- Some minor geometrical corrections have been made to make the elements join in a good way.

The stiffness that the adjacent buildings may employ on the lower levels of the structure is not included. This simplification is done because there is insufficient data to make any assumptions about the stiffness contribution that these buildings may impose on the main structure. One solution could be to create structural elements to simulate the buildings, but the amount of extra data in SAP2000 would make the analysis almost impossible to perform. The assumptions about the structural models may be a contributor to the poor correlation between the experimental data and the numerical data.

As presented in chapter 4.5.2, the error between the numerical and experimental results started out with a convergence criterium of 1.277. This is mainly due to the low MAC values in the initial test, at 0.954 for the first mode, 0.849 for the second mode and 0.448 for the third mode, which shows that the mode shapes do not correspond very well, especially the third mode. However, the difference is not as large for the calculated natural frequencies, where the experimental values are [4.630, 5.566, 6.363] [Hz] and the numerical values are [5.199, 7.014, 7.286] [Hz].

After the updating procedure, there was a significant improvement in the convergence criterium, reducing to 0.705. However, this is still not as low as desired. A convergence criterium close to 0 indicates a numerical model with perfect correlation to the experimental values. Therefore, 0.705 indicated that the model is still a dissimilar to the experiment. This can also be seen in the updated values, where there were only minor improvements to the MAC values for mode 1 and 2 with an increasement to 0.967 and 0.850. Mode 3 provided greater improvement with the increasement to 0.728. Natural frequencies also improved as an effect of the updating procedure providing these frequencies, [4.024, 5.566, 7.121] [Hz].

The results of the updating process present some interesting phenomena. The additional mass on the roof is the largest updated value, with 1.880 kN/m^2, while the facades is calculated to 0.000 kN/m and the internal floors is updated to only 0.054 kN/m^2. In a general case, it would be natural for most the additional mass to be distributed on the internal floors, that contains a heavier construction and can be subject to inventory mass, while roofs generally are lighter constructions, and not as often subject to additional mass. There are several possible reasons for the masses to distribute in this manner. One of the possibilities are that all the sensors that is evaluated is located on the roof except sensor five which is on the third floor, and that the mode shapes are mostly influenced by the additional mass on the roof. This can also be a result of the range of the boundaries of parameter values chosen, and that a wider boundary of values may return a more sensible result.

From Figure 4-48, we can draw that the mode shape of mode 3 looks quite inconsistent and chaotic at first glance, however, it presents some interesting points, first of which is the fact that the numerical calculation and the experimental campaign generates mostly inverse mode shapes. The majority of the difference in the mode shapes for the third mode is the magnitudes of deflection within the mode shape, not in the direction in which the deflection is occurring. The second point is that the third numerical mode seems to differ too much to the experimental values that it restricts the capability of the updating process for the two first modes.

In summery the updating process yielded large improvement on the validity of the finite element model. However, due to a complex structure and simplifications in the modelling, the Total Convergence Criterium cannot be said to validate the model sufficiently.

5.3 Further Work

With background in the results and discussions, different issues for further studies are presented.

From the parametric analysis:

- Review the analytical equations for in-plane bending.
- Study the possibility for reduction factors for the equations for lateral deflection due to kinematic rocking for LTF and monolithic CLT shear walls.
- Review the analytical equations for lateral deflection due to kinematic rocking for segmented CLT shear walls, and possibly implement modified equations.
- The significance of the different parameters in the calculations of lateral deflection due to deformation of sheathing-to-framing connection could be further studied.
- Review equations for lateral deflection due to compression of the bottom rail under the trailing stud, focusing particularly on the applied compressive force perpendicular to the grain.

From the modal analysis and model updating:

- More CLT-buildings should be subjected to dynamic identification and model updating, to provide sufficient foundation for reliability of modal analyses of these buildings.
- More advanced FE software should be utilized to execute finite element model updating more accurately.

6 Conclusions

The conclusions to the research questions are answered below:

1. How does the new analytical calculation models for lateral deflection of timber shear walls in the upcoming Eurocode 5 relate to Finite Element Analyses in SAP2000?

Overall, the new analytical equations provided deflections comparable to deflections calculated numerically in SAP2000. In cases where analytical and numerical models deviated, sufficient foundation for discussing reasoning were available. As such, it can be concluded that the FEM analysis in SAP2000 relate good to the analytical calculation models for lateral deflection of timber shear walls in the upcoming Eurocode 5.

2. How sensitive are each contribution to the variation of different parameters and wall configurations?

In the case of lateral deflection due to in-plane shear and rigid body sliding, none of the parametric input provided deviation in the analysis. As such, it can be concluded that these contributions are not sensitive to variation of parametric input.

All parameters except magnitude of loading proved to be sensitive in the calculations of lateral deflection due to in-plane bending and due to deformation of sheathing-to-framing connections.

For the lateral deflection due to kinematic rocking, the only parameter that made somewhat of a deviation in results for LTF and monolithic CLT was different combinations of lateral and vertical loading. For the modified equations for segmented CLT shear walls none of the parametric input proved to be sensitive to variation.

Lastly, the contribution of lateral deflection due to deformation of the bottom rail under the trailing stud, was somewhat sensitive to all the different parametric input.

3. How accurate is the modal analysis and model updating in SAP2000 compared to experimental values in a complex building of CLT shear walls?

The initial dynamic identification in SAP2000 provided excessive differences to the experimental values. As such, it can be concluded that the reliability of the modal analysis is unsatisfactory. Furthermore, the Finite Element Model updating provided a significant improvement to the Total Convergence Criterium. However, improvements were not satisfactory to confidently validate the updated model.

4. How reliable are the different calculation models in the identification of timber shear walls subjected to different loading in the Serviceability Limit State?

Results generated in the thesis provides a good basis for evaluating the calculation models presented.

The parametric analysis indicated good correlation between numerical calculations analytical calculations. As such, the reliability of the analytical equations seems to generally be acceptable in identifying the lateral deflection of timber shear walls.

As neither result from initial nor updated modal analyses in SAP2000 provided sufficient similarities to the experimental campaign, foundation to validate this procedure are not sufficient from the analyses in this thesis.

Bibliography

ALOISIO, A., PASCA, D., TOMASI, R. & FRAGIACOMO, M. 2020. Dynamic identification and model updating of an eight-storey CLT building. *Engineering Structures,* 213**,** 8. DOI:http://doi.org/10.1016/j.engstruct.2020.110593

BOGGIAN, F., RICCI, E., SALENIKOVICH, A., SPRINGHETTI, R. & TOMASI, R. 2021. Deflections of light timber frame shear wall segments: validation of a standard proposal. *World conference on timber engineering.* Santiago, Chile:

BRANDNER, R., FLATSCHER, G., RINGHOFER, A., SCHICKHOFER, G. & THIEL, A. 2016. Cross laminated timber (CLT): overview and development. *European Journal of Wood and Wood Products,* 74**,** 331-351. DOI:http://doi.org/10.1007/s00107-015-0999-5

BROWNJOHN, J. M. W., XIA, P. Q., HAO, H. & XIA, Y. 2001. Civil structure condition assessment by FE model updating: methodology and case studies. *Finite Elements in Analysis and Design,* 37**,** 761-775. DOI:http://doi.org/10.1016/s0168-874x(00)00071-8

CEN, T. C. 2021. Eurocode 5: Design of timber structures — Common rules and rules for buildings — Part 1-1: General - prEN 1995-1-1.: Technical Committee CEN

CEN, T. C. 2022. CEN/TC 250/SC 5 N 1546 CEN/TC - 13.7 Lateral displacement of multi storey timber shear walls - prEN 1995-1-1.: Technical Committee CEN

CHOPRA, A. K. 2020. *Dynamics of Structures: Theory and Applications to Earthquake Engineering.* 5th Edition in SI Units, Pearson Education, Inc.

COMMUNITY, S. 2022. SciPy Reference Guide. SciPy documentation

COMPUTERS AND STRUCTURES, I. 2017. CSI Analysis Reference Manual.

COMPUTERS AND STRUCTURES, I. 2020. CSi OAPI Documentation.

COOK, R. D., MALKUS, D. S., PLESHA, M. E. & WITT, R. J. 2001. *Concepts and Applications of Finite Element Analysis* 4th Edition, John Wiley and Sons

EUSUF, M., ABDUL RASHID, K., NOOR, W. & HASAN, A. 2012. Shear Wall Construction in Buildings: A Conceptual Framework on the Aspect of Analysis and Design. *Applied Mechanics and Materials,* 268-270**,** 706-711. DOI:http://doi.org/10.4028/www.scientific.net/AMM.268-270.706

GOKDEMIR, H., OZBASARAN, H., DOGAN, M., UNLUOGLU, E. & ALBAYRAK, U. 2013. Effects of torsional irregularity to structures during earthquakes. *Engineering Failure Analysis,* 35**,** 713-717. DOI:http://doi.org/10.1016/j.engfailanal.2013.06.028

GROSSI, P., SARTORI, T. & TOMASI, R. 2015. Tests on timber frame walls under in-plane forces: part 1. *Proceedings of the Institution of Civil Engineers-Structures and Buildings,* 168**,** 826-839. DOI:http://doi.org/10.1680/stbu.13.00107

GURHOLT, C.-U. D. & MIKALSEN, J. N. 2021. *Finite Element Model Updating of a Multi-Storey CLT-Building and Analysis of Modal Performance Indicators.* Structural Engineering and Architecture Masters, Norwegian University of Life Sciences

HIBBLER, R. C. 2013. *Mechanics of Materials, SI Edition.* 9th Edition, Pearson Education Centre

LAGAROS, N. D., VASILEIOU, N. & KAZAKIS, G. 2019. A C# code for solving 3D topology optimization problems using SAP2000. *Optimization and Engineering,* 20**,** 1-35. DOI:http://doi.org/10.1007/s11081-018-9384-7

LUKACS, I., BJORNFOT, A. & TOMASI, R. 2019. Strength and stiffness of cross-laminated timber (CLT) shear walls: State-of-the-art of analytical approaches. *Engineering Structures,* 178**,** 136-147. DOI:http://doi.org/10.1016/j.engstruct.2018.05.126

MARZALEH, A. S., NERBANO, S., CROCE, A. S. & STEIGER, R. 2018. OSB sheathed light-frame timber shear walls with strong anchorage subjected to vertical load, bending moment, and monotonic lateral load. *Engineering Structures,* 173**,** 787-799. DOI:http://doi.org/10.1016/j.engstruct.2018.05.044

MORDINI, A., SAVOV, K. & WENZEL, H. 2007. The Finite Element Model Updating: A Powerful Tool for Structural Health Monitoring. *Structural Engineering International,* 17**,** 352-358. DOI:http://doi.org/10.2749/101686607782359010

PANWAR, V., GUPTA, P., BAGHA, A. K. & CHAUHAN, N. 2018. A Review on studies of Finite Element Model Updating and Updating of Composite Materials. *Materials Today: Proceedings,* 5**,** 27912-27918. DOI:http://doi.org/https://doi.org/10.1016/j.matpr.2018.10.030

RAINIERI, C. & FABBROCINO, G. 2014. *Operational Modal Analysis of Civil Engineering Structures.* New York, Springer

RICCI, E. 2020. *Design methods for Light Timber Frame shear walls: validation with experimental data.* Corso di Laurea Magistrale in Ingegneria Civile Masters, Università degli Studi di Trento

ROTHOBLAAS. 2022. *Fastening* [Online]. rothoblaas. Available: https://www.rothoblaas.com/products/fastening [Accessed 23.03 2022].

SANBORN, K., GENTRY, T. R., KOCH, Z., VALKENBURG, A., CONLEY, C. & STEWART, L. K. 2019. Ballistic performance of Cross-laminated Timber (CLT). *International Journal of Impact Engineering,* 128**,** 11-23. DOI:http://doi.org/10.1016/j.ijimpeng.2018.11.007

VELEZ, W., GOMEZ, D. & THOMSON, P. 2009. FINITE ELEMENT MODEL UPDATING. *Dyna-Colombia,* 76**,** 177-189

WALLNER-NOVAK, M., KOPPELHUBER, J. & POCK, K. 2014. *Cross-Laminated Timber Structural Design | Basic design and engineering principles according to Eurocode.* proHolz Austria

Appendix

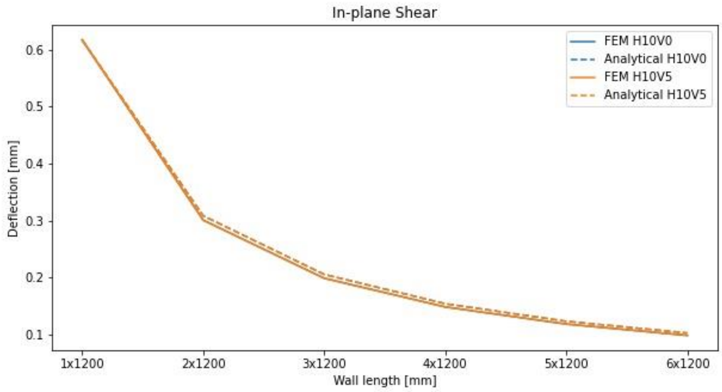
In this chapter the all the results from the parametric analysis are presented, as well as the different Python scripts used to perform the different analyses. In Table A-1, a short description of the Appendix is presented. All programming scripts are written in Python version 3.8.

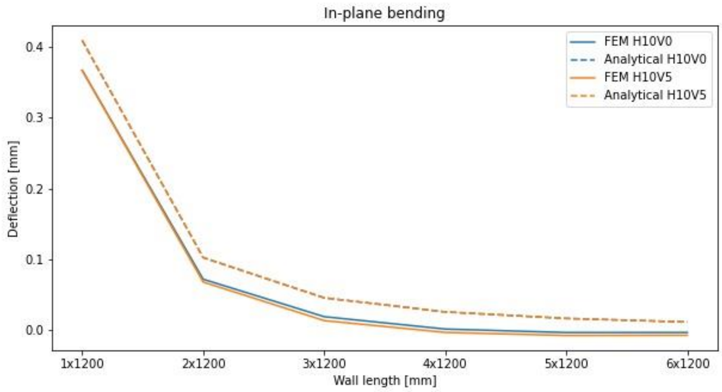
Table A-1: Description of the Appendix.

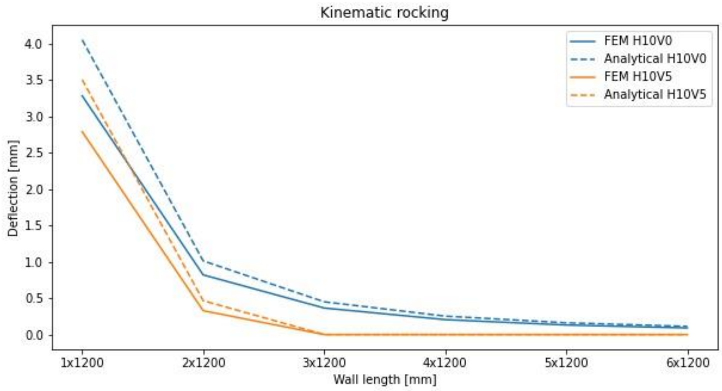
Location	**Description**
Appendix A	All results from the parametric analysis.
Appendix B	One of the scripts for performing the parametric analysis on the LTF shear walls. The script open SAP2000 with the help of the OAPI and runs multiple analyses, variating the different predefined parameters.
Appendix C	One of the scripts for performing the parametric analysis on the monolithic and segmented CLT shear walls. The script open SAP2000 with the help of the OAPI and runs multiple analyses, variating the different predefined parameters.
Appendix D	A script with the algorithm to perform model updating. A pre-created model in SAP2000 including defined groups and materials corresponding to the definitions in the script is extracted and updated with the help of the OAPI.

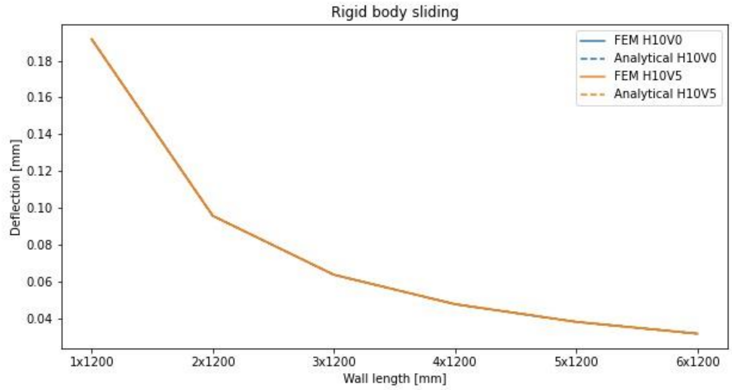
Appendix A – Results from parametric analysis.

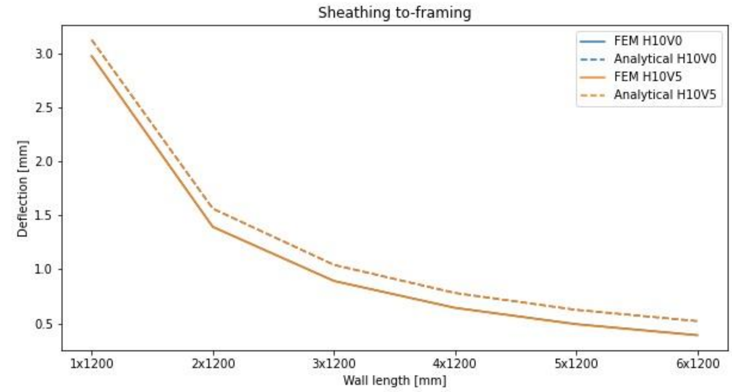
Total deflection of LTF:

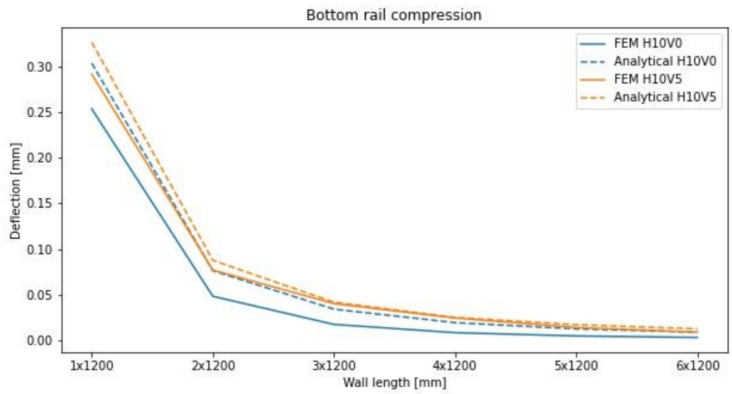


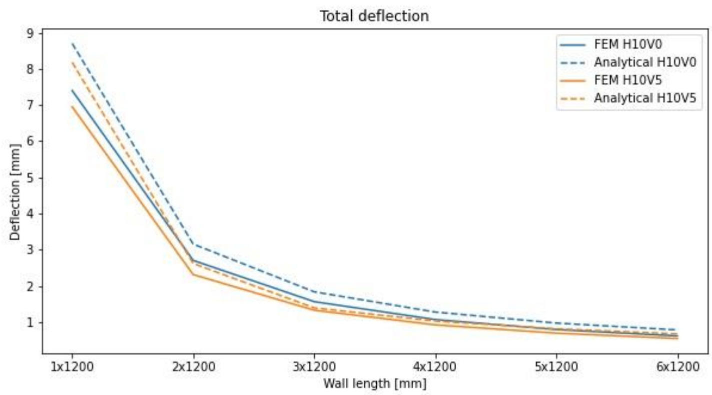




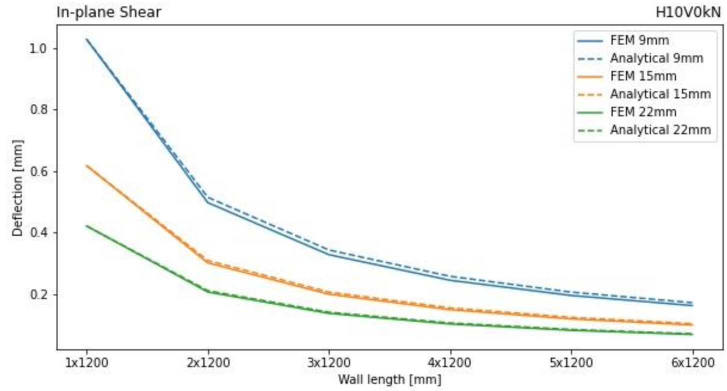

Rigid body sliding
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

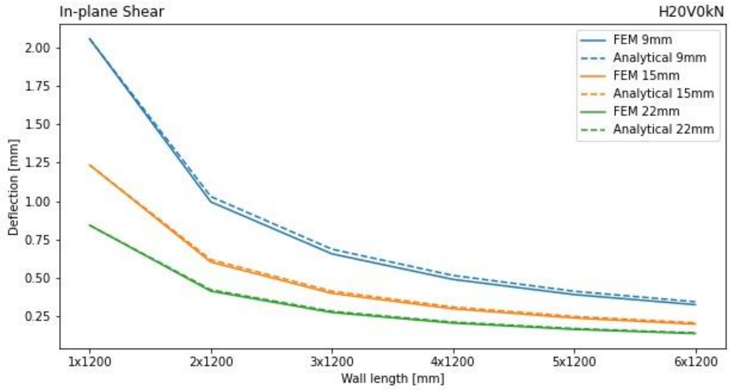

Sheathing to-framing
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
3.0
2.5
2.0
1.5
1.0
0.5
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

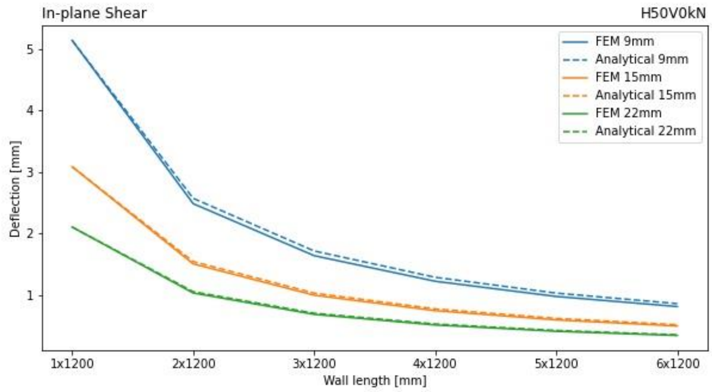

Bottom rail compression
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.30
0.25
0.20
0.15
0.10
0.05
0.00
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]


Total deflection
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
9
8
7
6
5
4
3
2
1
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

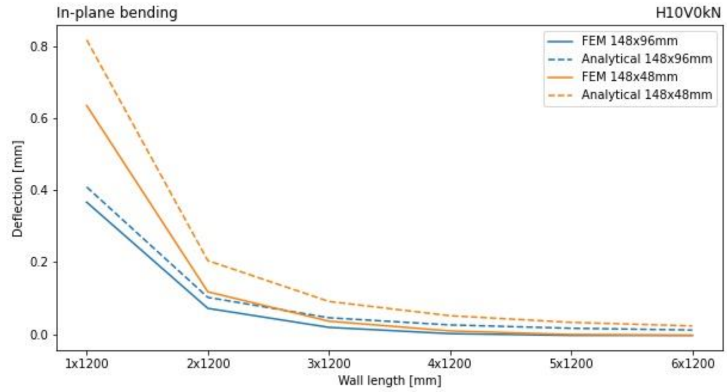
Lateral deflection due to in-plane Shear of LTF:

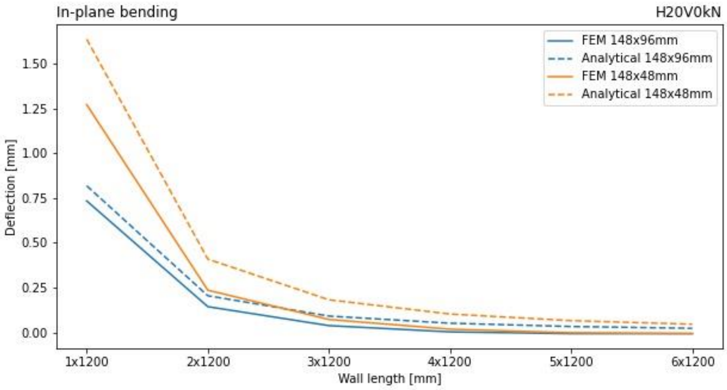




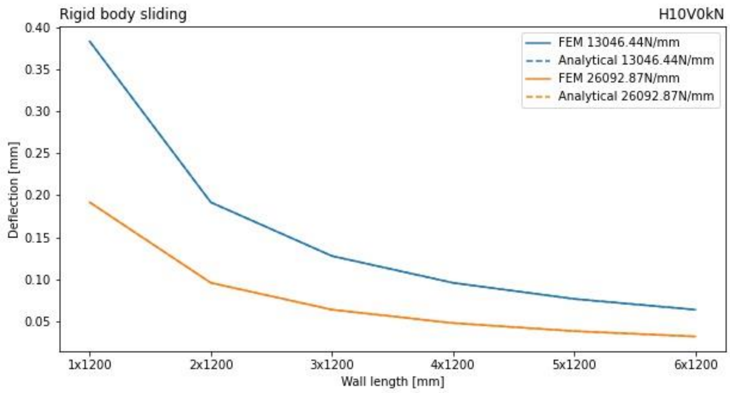


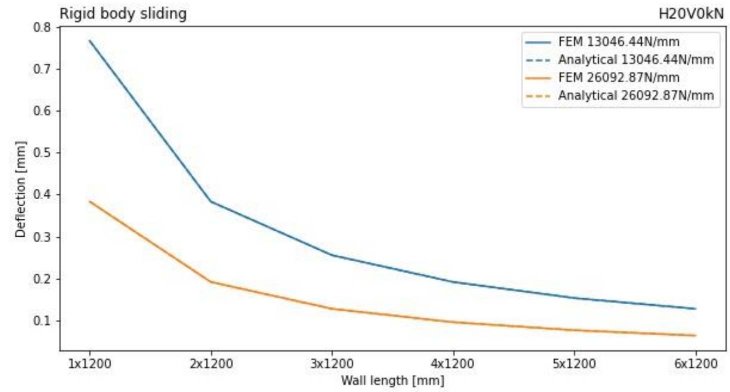
Lateral deflection due to in-plane Bending of LTF:

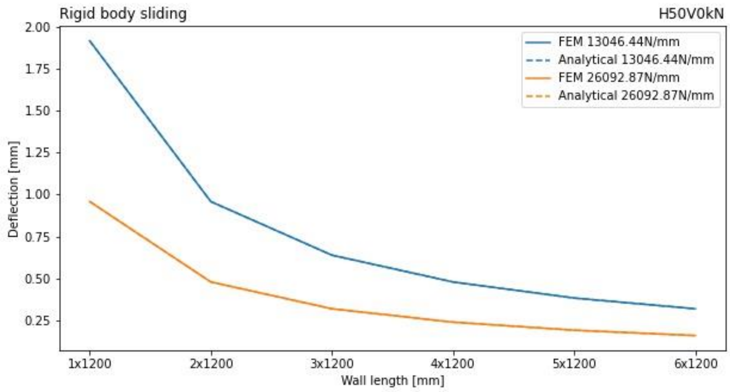




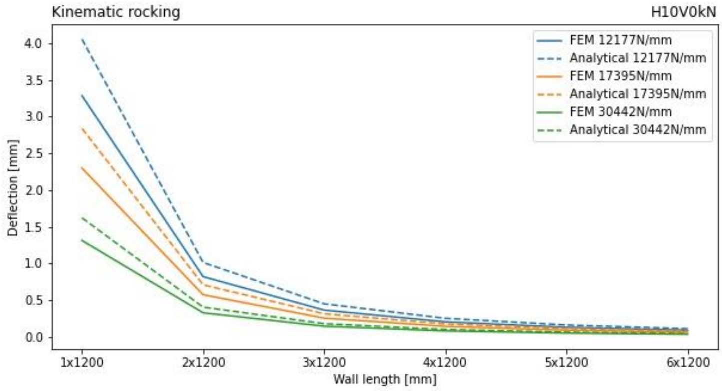
Lateral deflection due to Rigid Body Sliding of LTF:

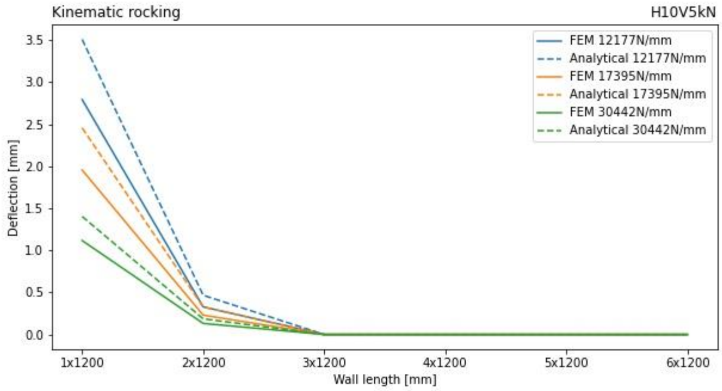


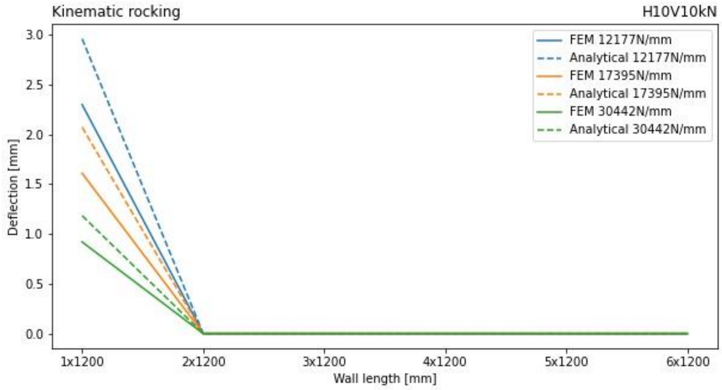


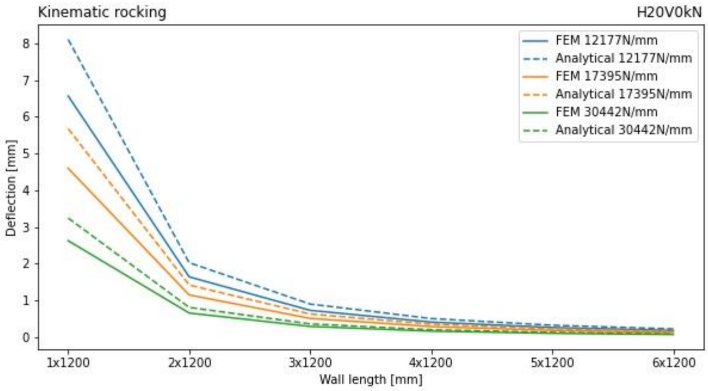


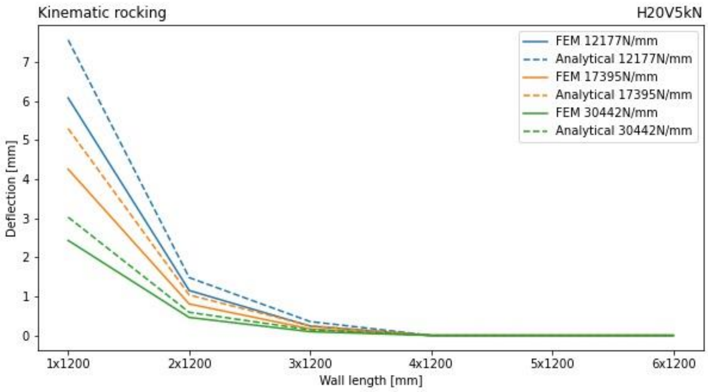
Lateral displacement due to Kinematic Rocking of LTF:

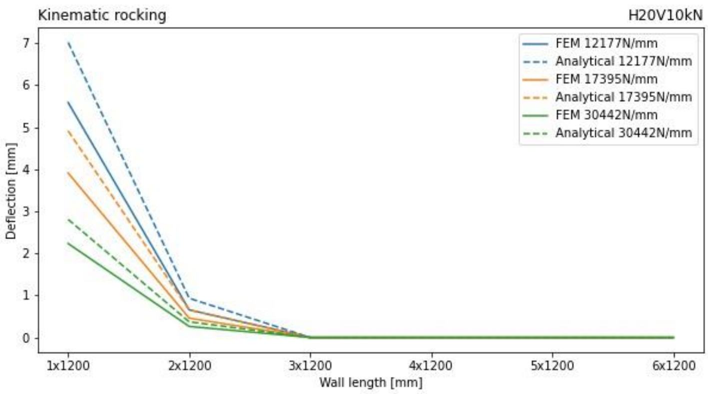




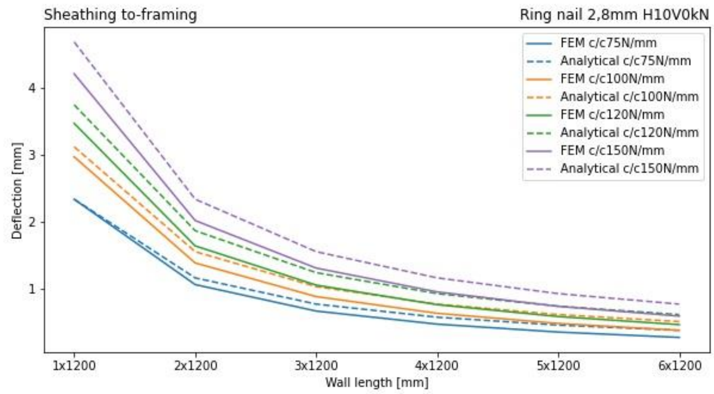


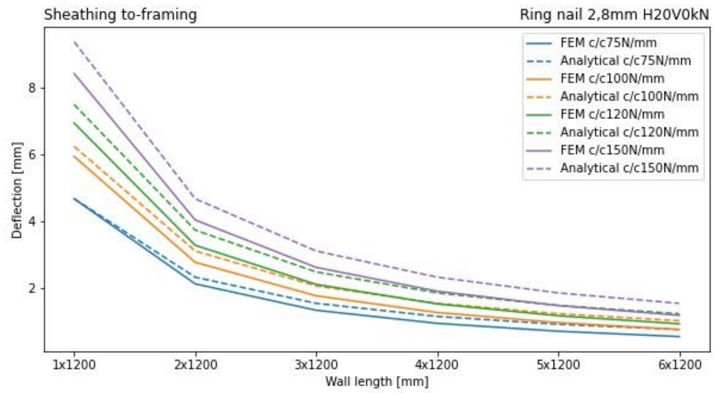

Kinematic rocking
H20V0kN
FEM 12177N/mm
Analytical 12177N/mm
FEM 17395N/mm
Analytical 17395N/mm
FEM 30442N/mm
Analytical 30442N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

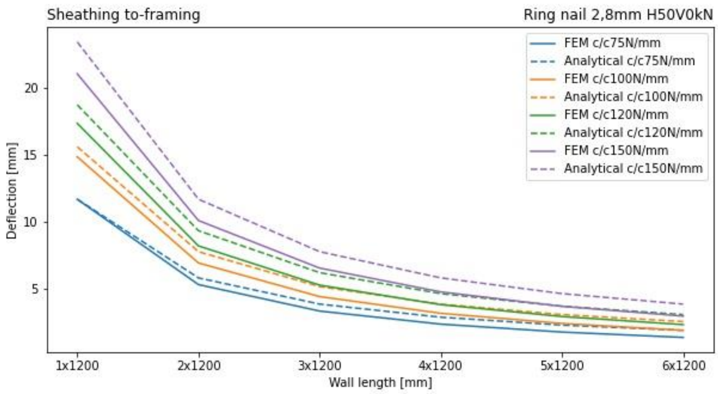

Kinematic rocking
H20V5kN
FEM 12177N/mm
Analytical 12177N/mm
FEM 17395N/mm
Analytical 17395N/mm
FEM 30442N/mm
Analytical 30442N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

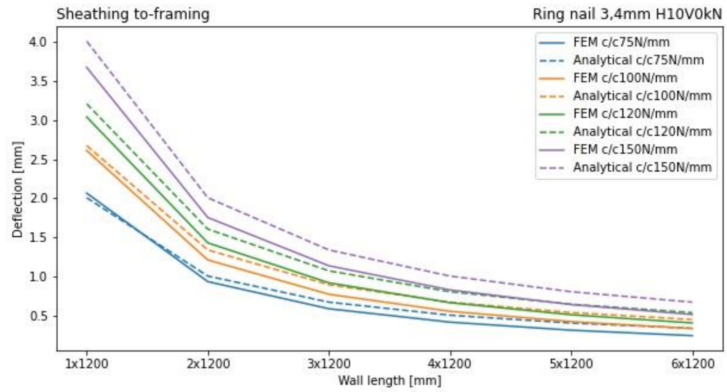

Kinematic rocking
H20V10kN
FEM 12177N/mm
Analytical 12177N/mm
FEM 17395N/mm
Analytical 17395N/mm
FEM 30442N/mm
Analytical 30442N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

Lateral displacement due to deformation of the sheathing-to-framing connections:

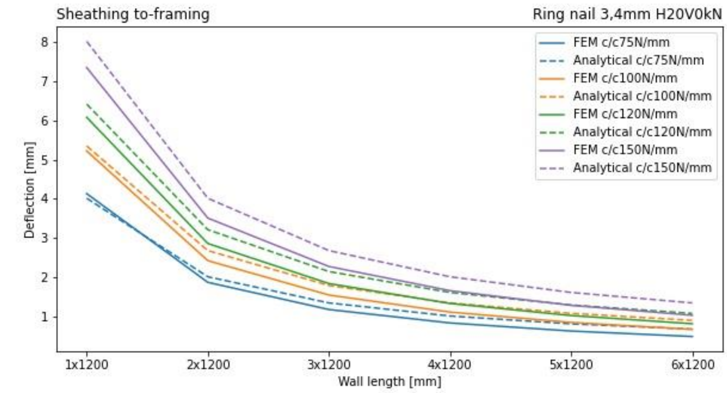




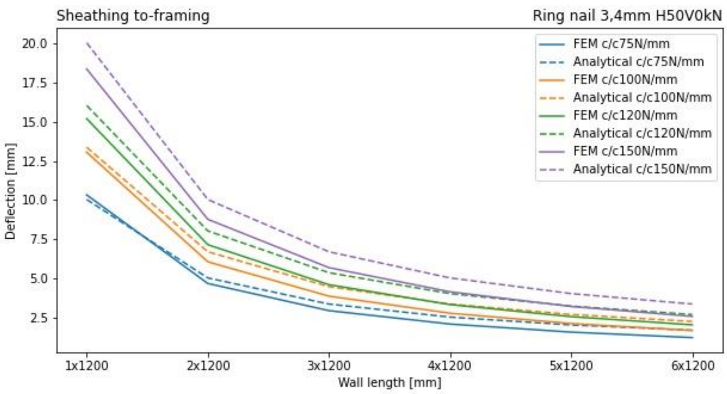




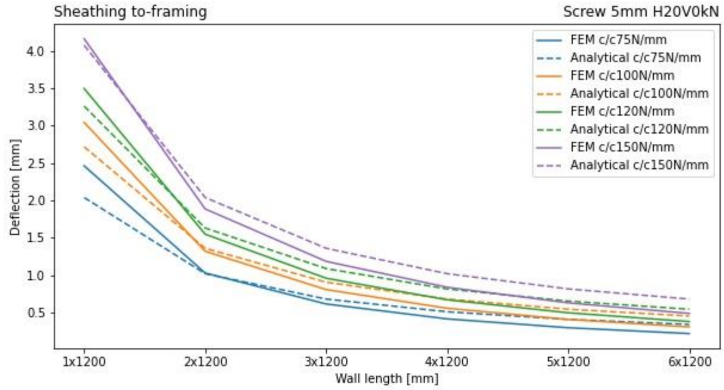
Sheathing to-framing
Ring nail 3,4mm H10V0kN
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm

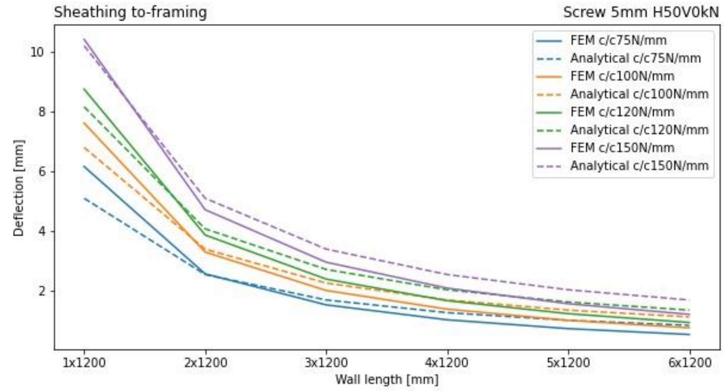


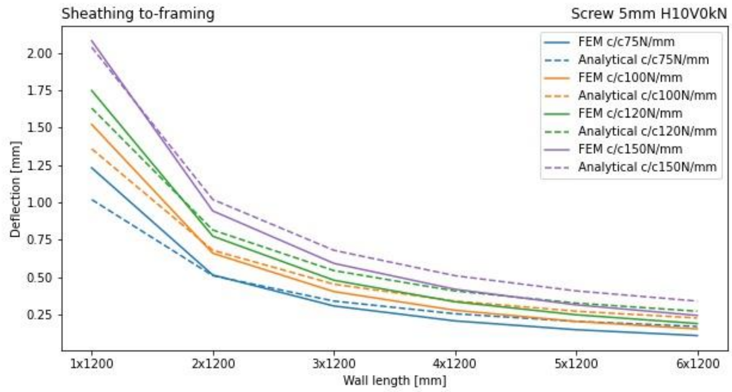
Sheathing to-framing
Ring nail 3,4mm H20V0kN
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm



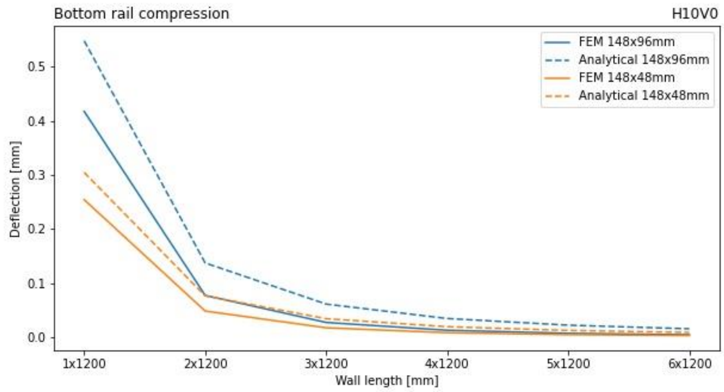
Sheathing to-framing
Ring nail 3,4mm H50V0kN
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm

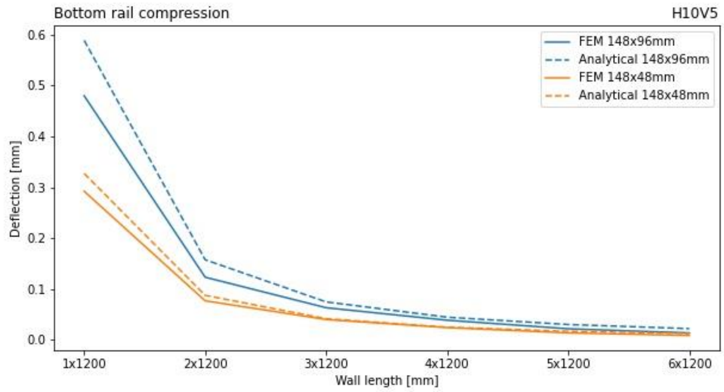

Sheathing to-framing
Screw 5mm H20V0kN
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

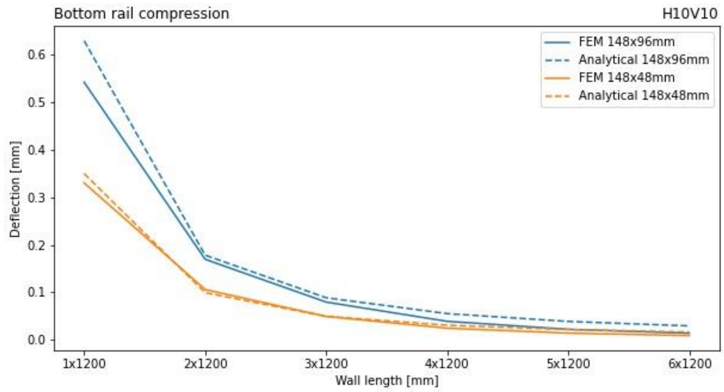

Sheathing to-framing
Screw 5mm H50V0kN
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

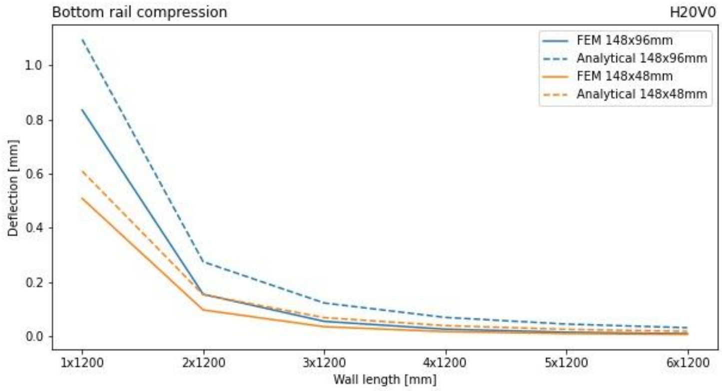

Sheathing to-framing
Screw 5mm H10V0kN
FEM c/c75N/mm
Analytical c/c75N/mm
FEM c/c100N/mm
Analytical c/c100N/mm
FEM c/c120N/mm
Analytical c/c120N/mm
FEM c/c150N/mm
Analytical c/c150N/mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

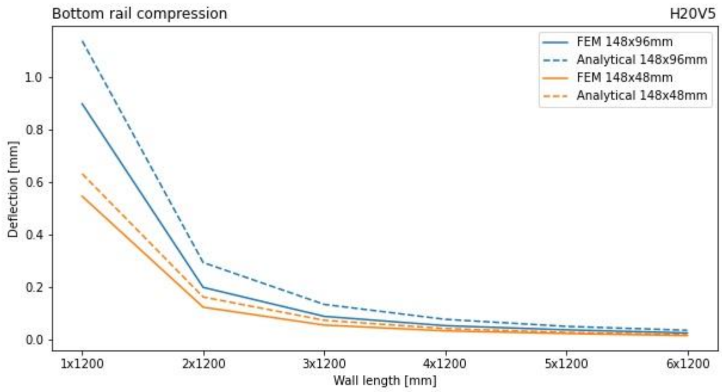
Lateral displacement due to deformation of the bottom rail perpendicular to the grain under the trailing stud:

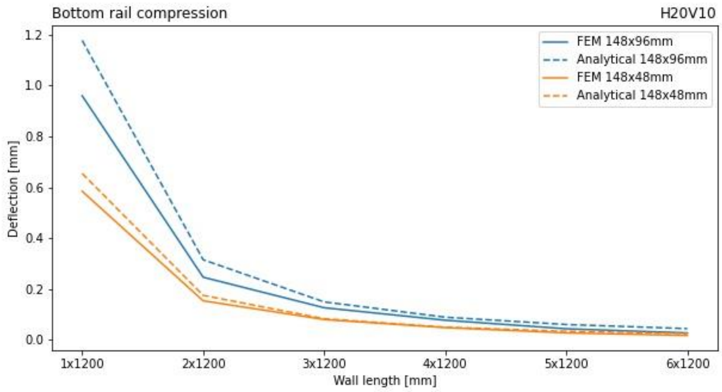




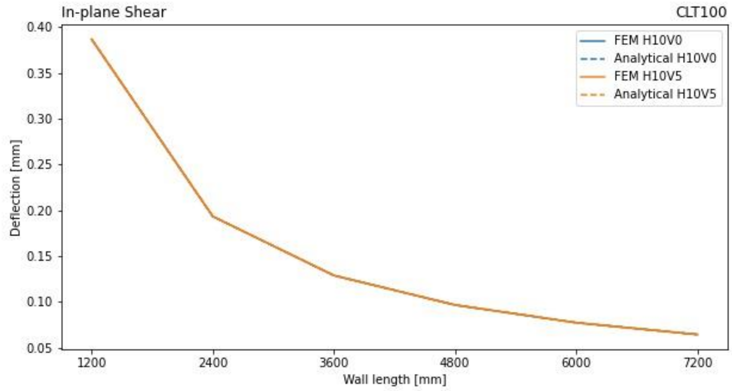


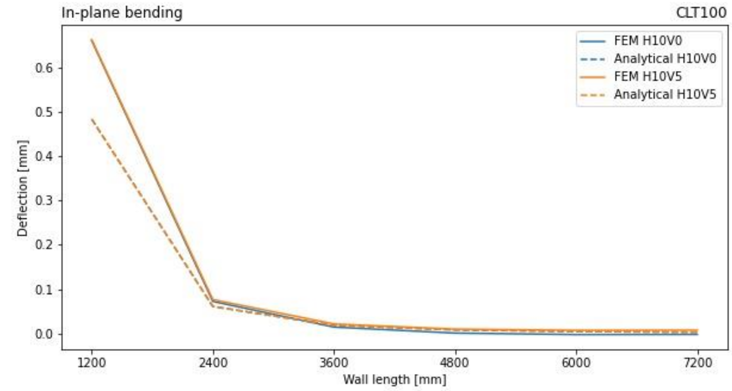

Bottom rail compression
H20V0
FEM 148x96mm
Analytical 148x96mm
FEM 148x48mm
Analytical 148x48mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

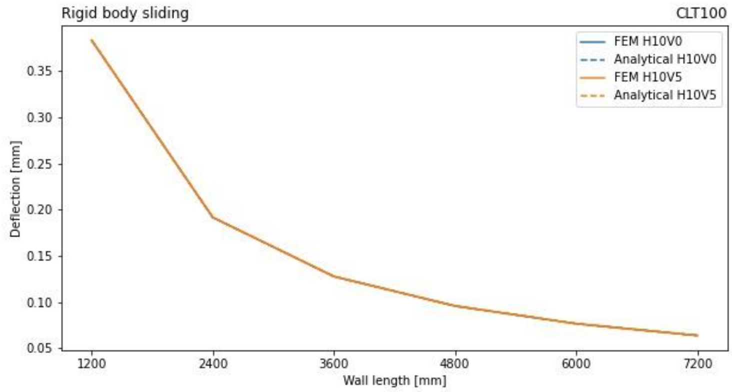

Bottom rail compression
H20V5
FEM 148x96mm
Analytical 148x96mm
FEM 148x48mm
Analytical 148x48mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

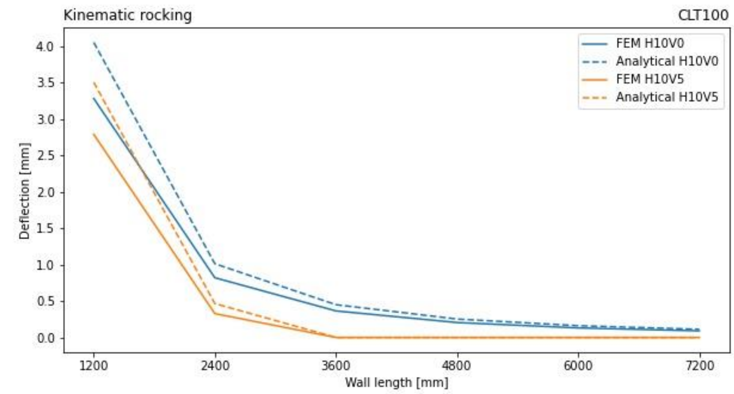

Bottom rail compression
H20V10
FEM 148x96mm
Analytical 148x96mm
FEM 148x48mm
Analytical 148x48mm
Deflection [mm]
Wall length [mm]
1x1200
2x1200
3x1200
4x1200
5x1200
6x1200

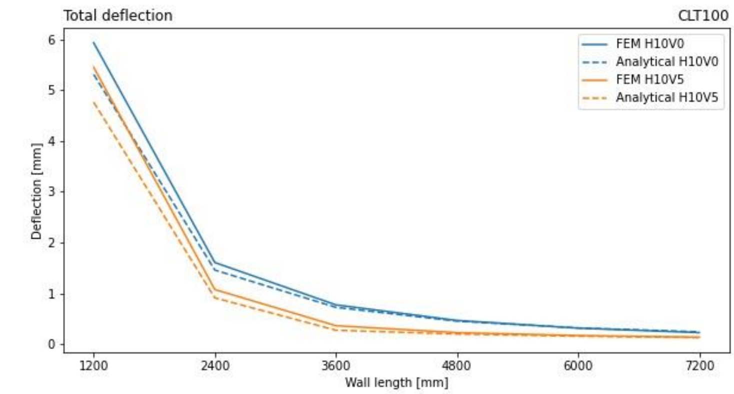
Total deflection of monolithic CLT:

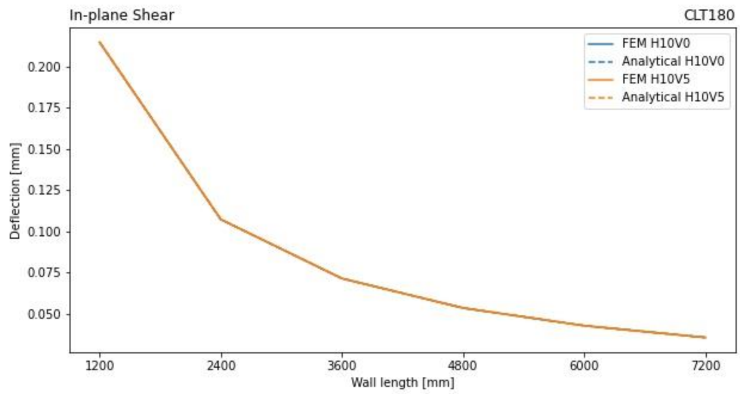


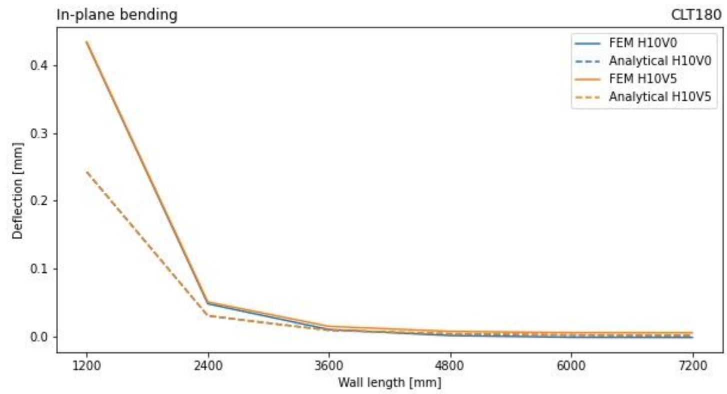


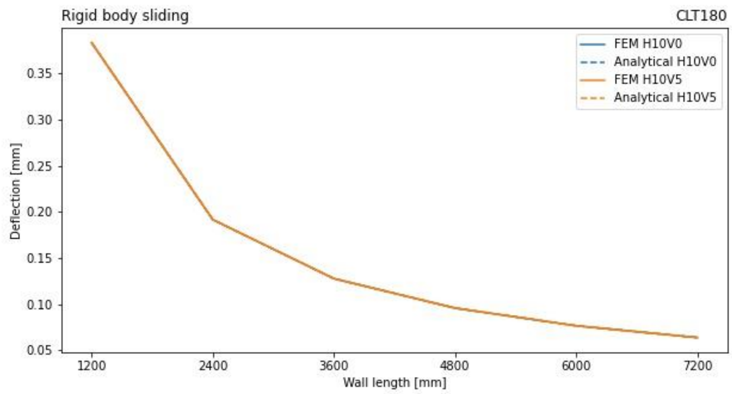


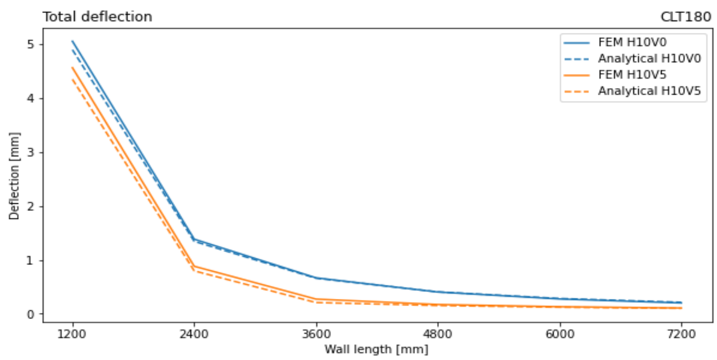

Kinematic rocking
CLT100
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]


Total deflection
CLT100
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]

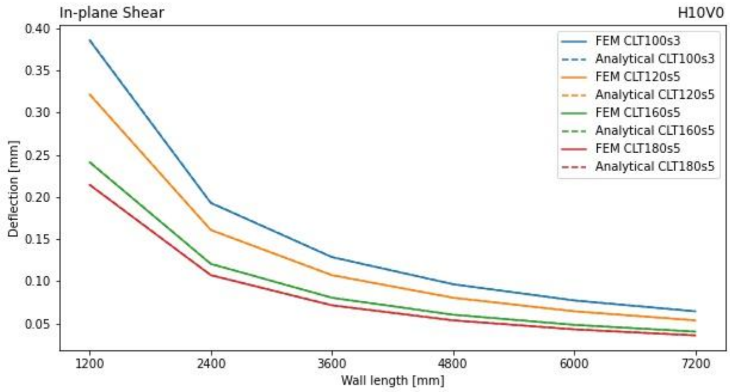

In-plane Shear
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]

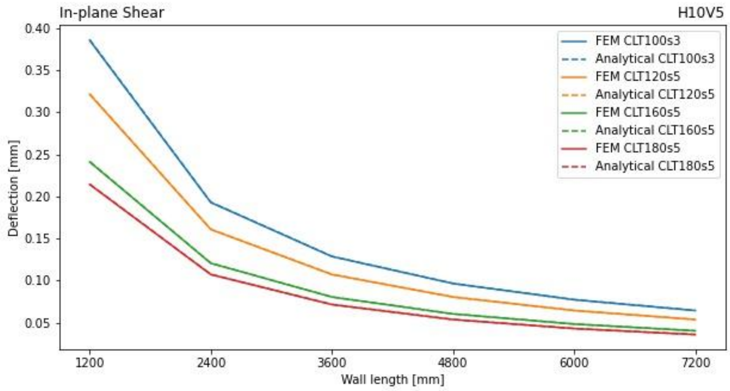

In-plane bending
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
0.0
0.1
0.2
0.3
0.4
1200
2400
3600
4800
6000
7200

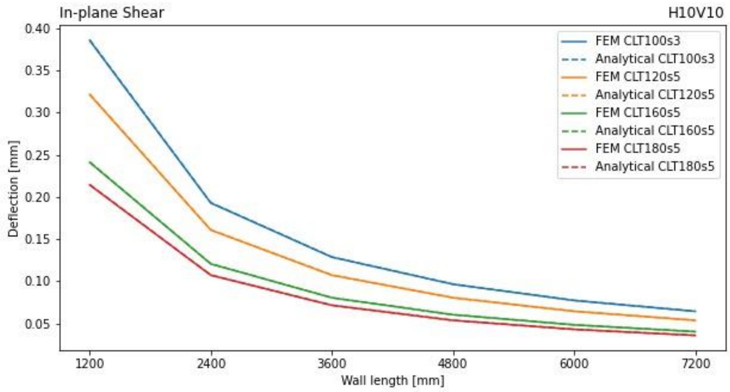

Rigid body sliding
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
0.05
0.10
0.15
0.20
0.25
0.30
0.35
1200
2400
3600
4800
6000
7200

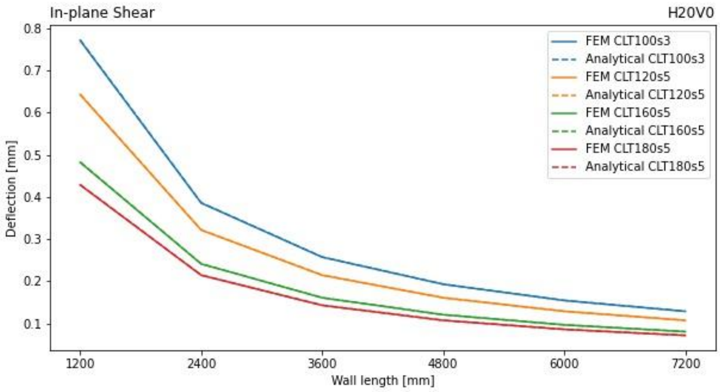

Total deflection
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
0
1
2
3
4
5
1200
2400
3600
4800
6000
7200

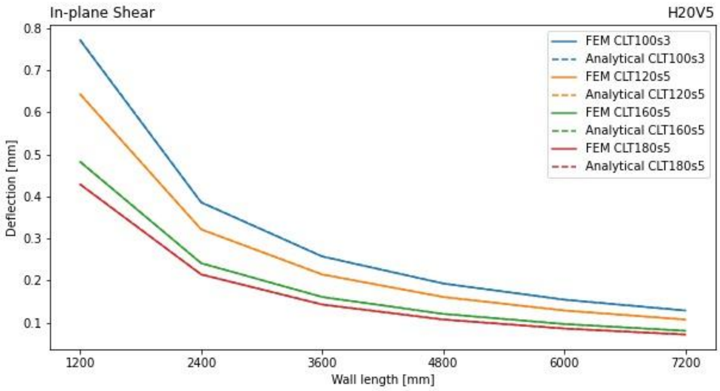
Lateral deflection due to in-plane Shear of monolithic CLT:

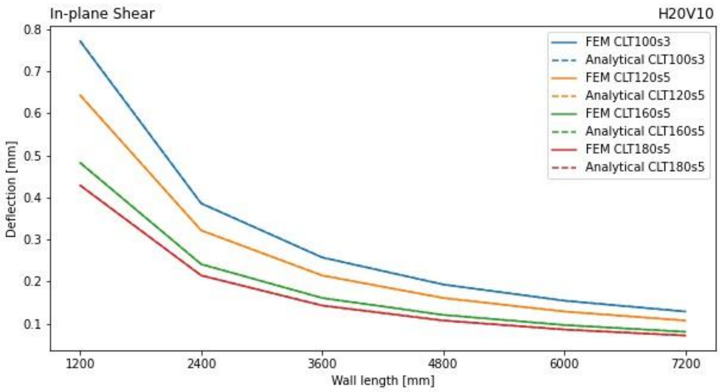




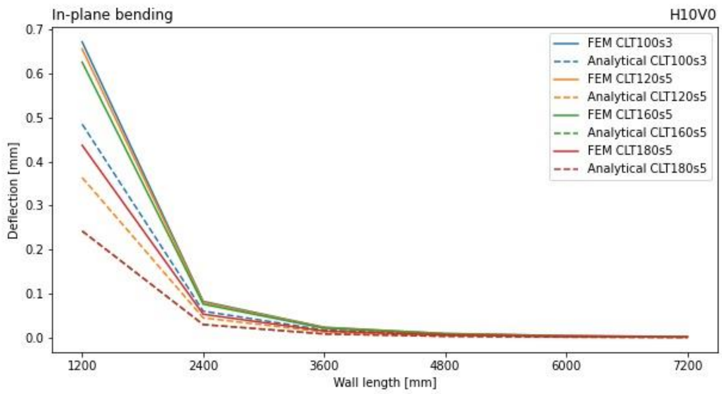


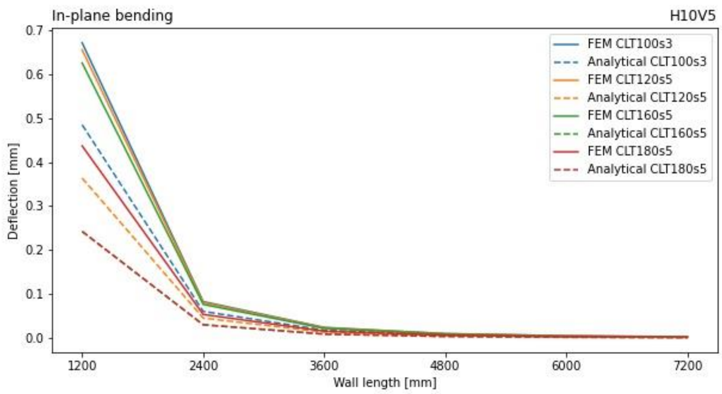

In-plane Shear
H20V0
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]

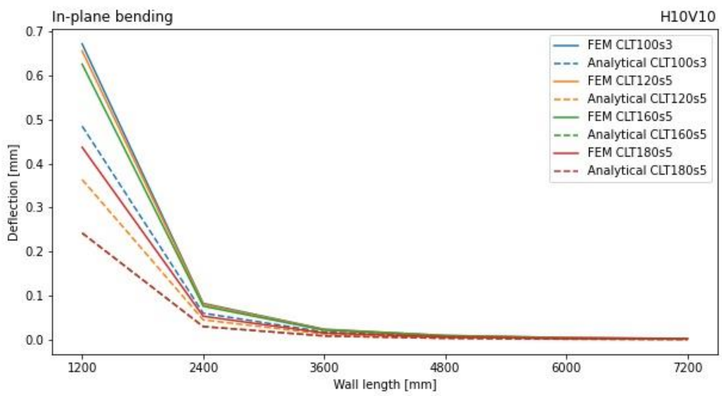

In-plane Shear
H20V5
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]

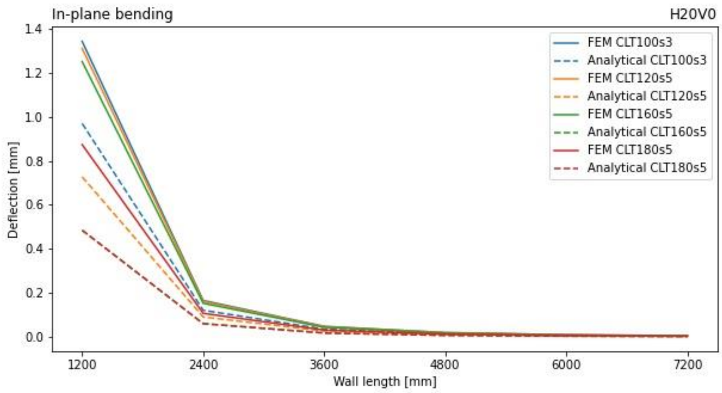

In-plane Shear
H20V10
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]

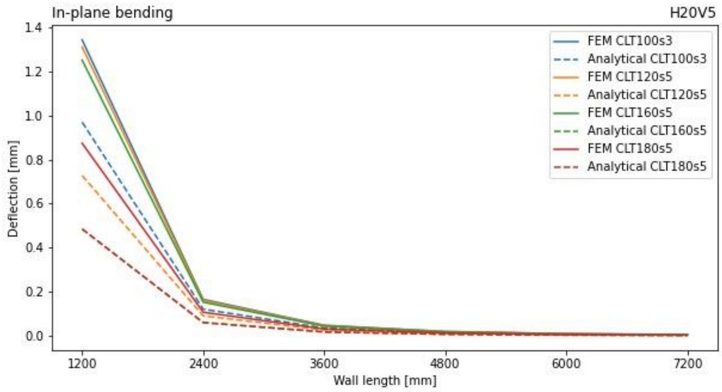
Lateral deflection due to in-plane Bending of monolithic CLT:

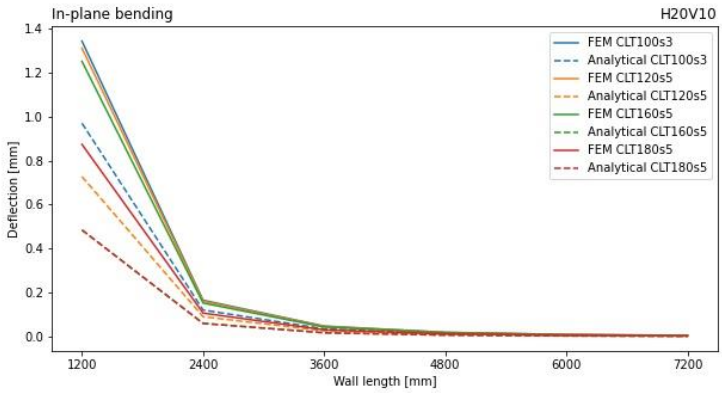




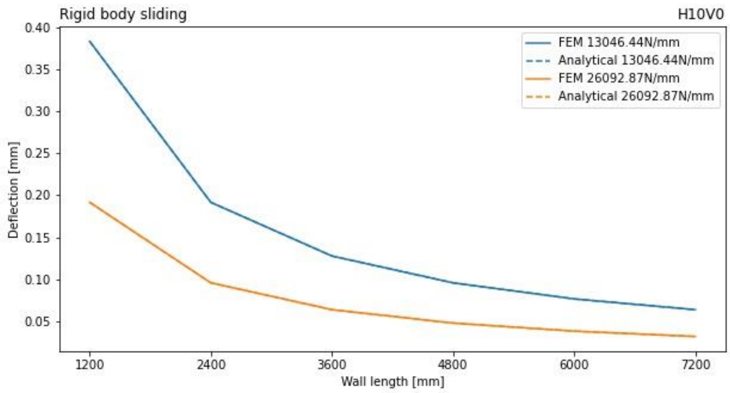


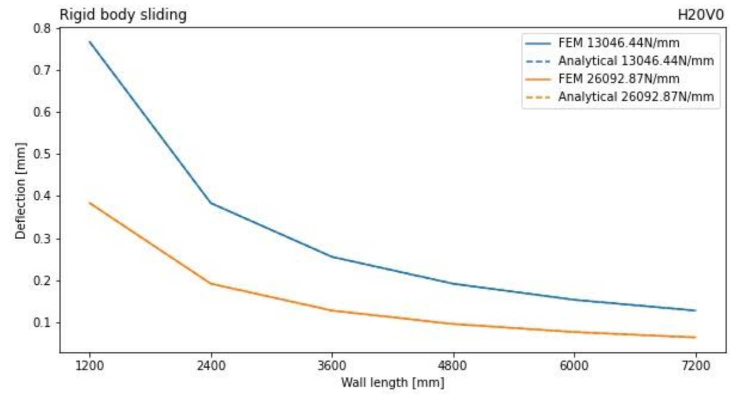

In-plane bending
H20V0
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
1200
2400
3600
4800
6000
7200
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4


In-plane bending
H20V5
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
1200
2400
3600
4800
6000
7200
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4

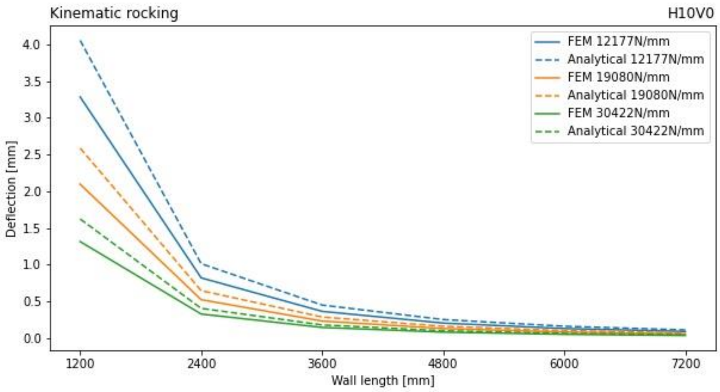

In-plane bending
H20V10
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
1200
2400
3600
4800
6000
7200
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4

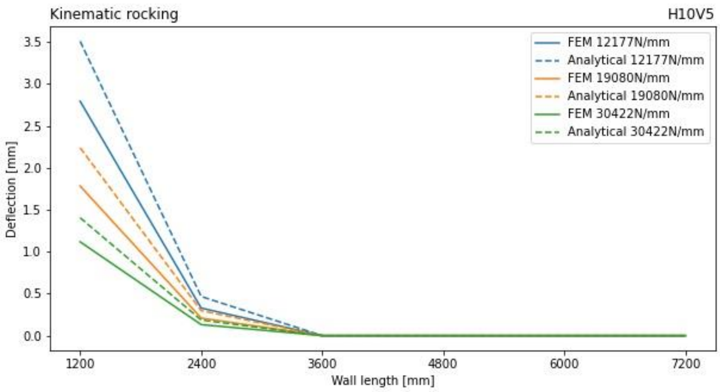
Lateral deflection due to Rigid Body Sliding of monolithic CLT:

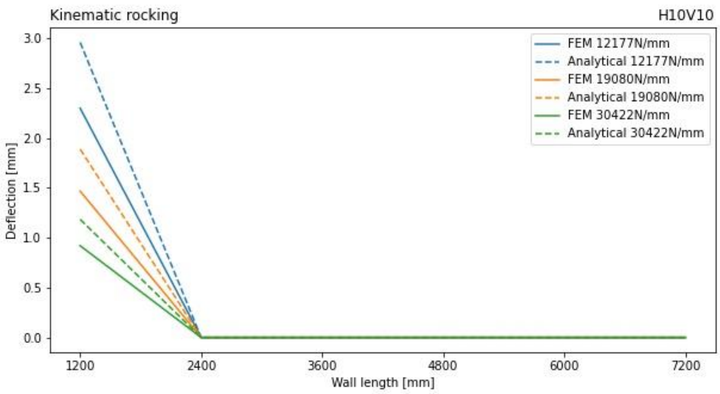


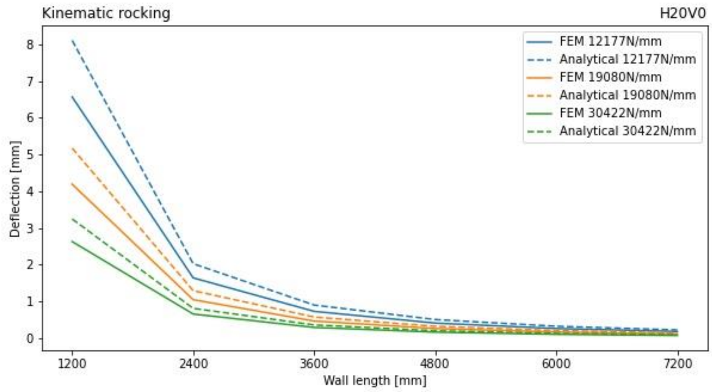


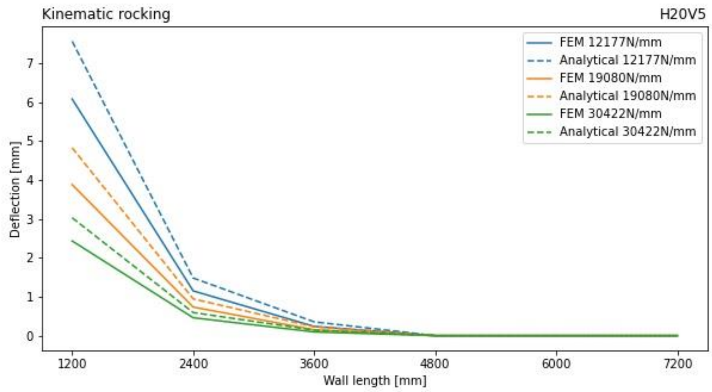
Lateral deflection due to Kinematic Rocking of monolithic CLT:

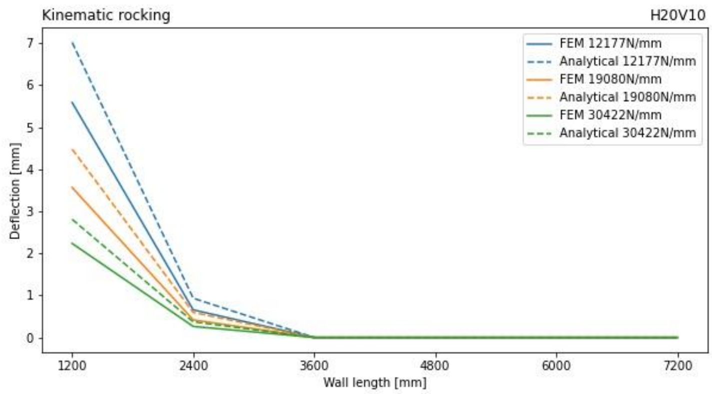




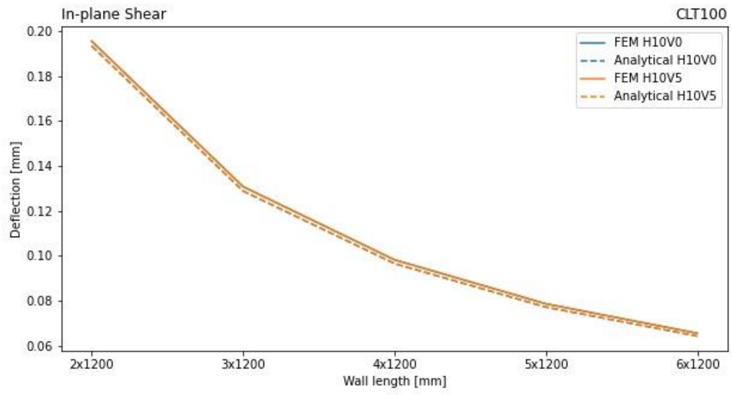


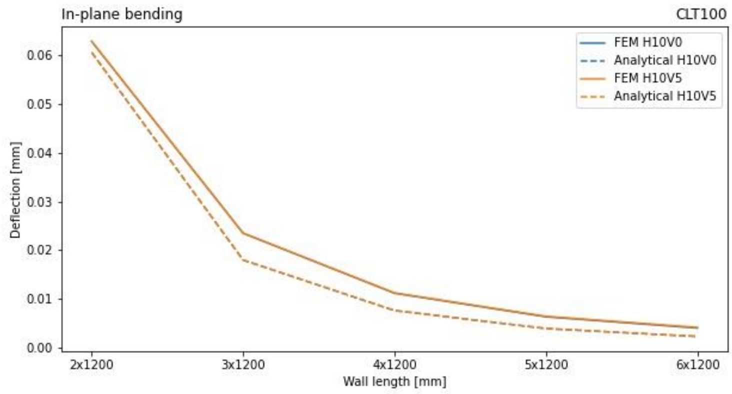

Kinematic rocking
H20V0
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
0
1
2
3
4
5
6
7
8
1200
2400
3600
4800
6000
7200

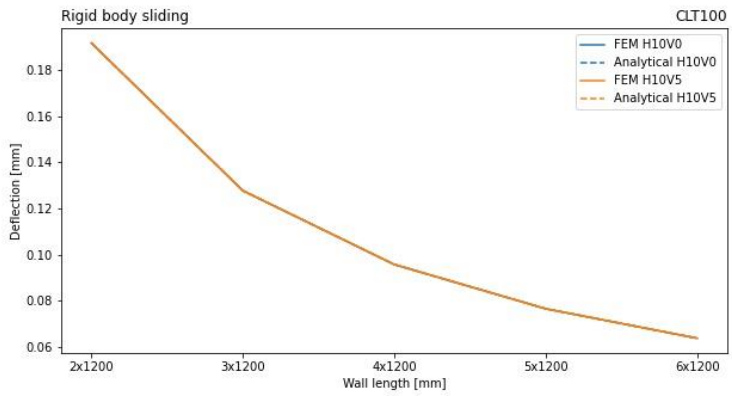

Kinematic rocking
H20V5
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
0
1
2
3
4
5
6
7
1200
2400
3600
4800
6000
7200

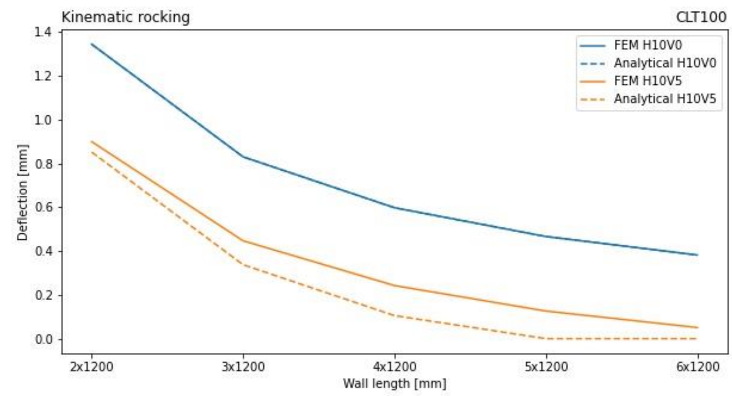

Kinematic rocking
H20V10
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
0
1
2
3
4
5
6
7
1200
2400
3600
4800
6000
7200

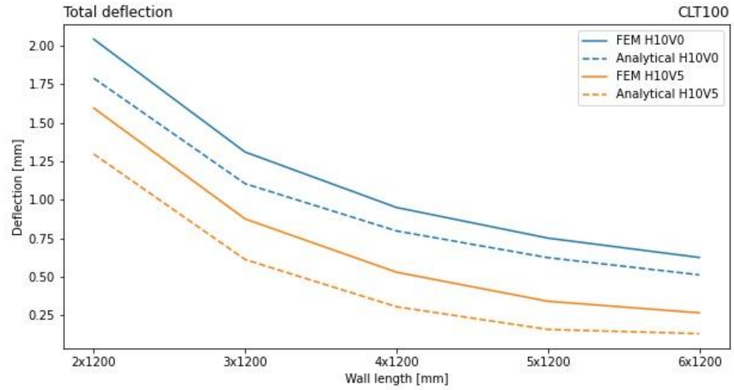
Total deflection of segmented CLT:

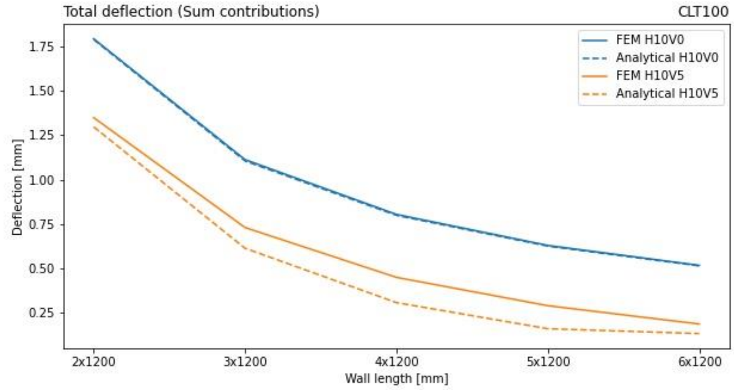


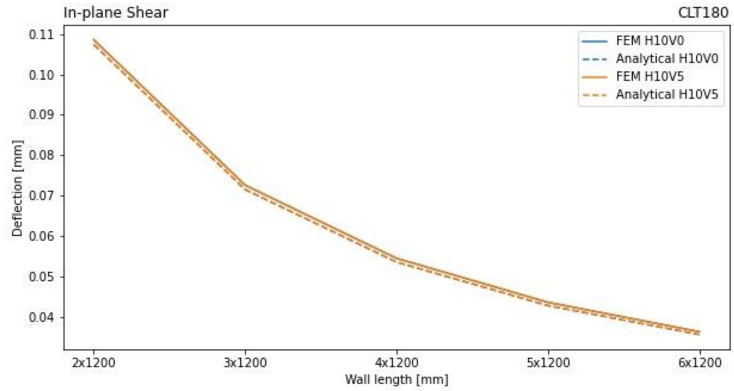


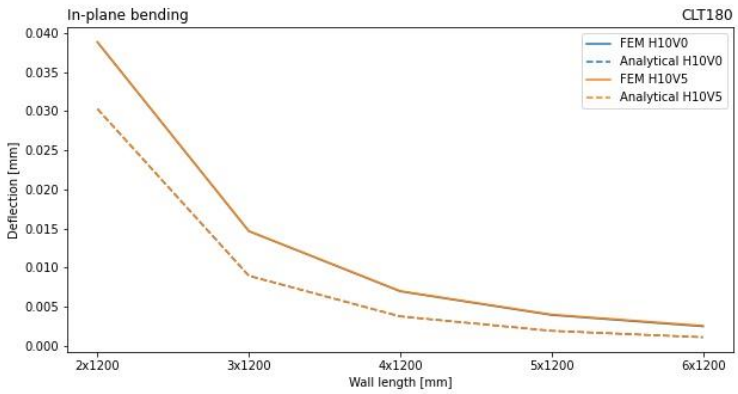


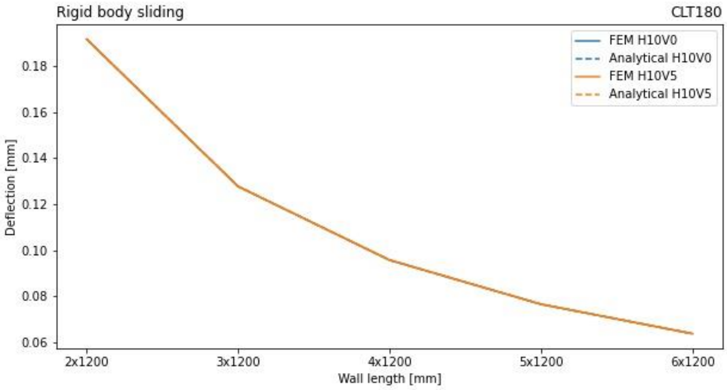


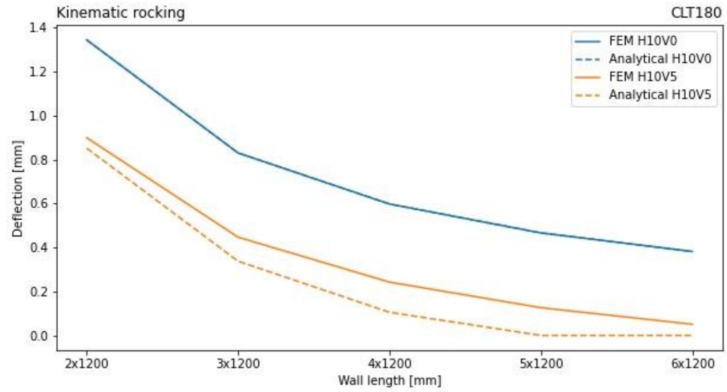

Total deflection
CLT100
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

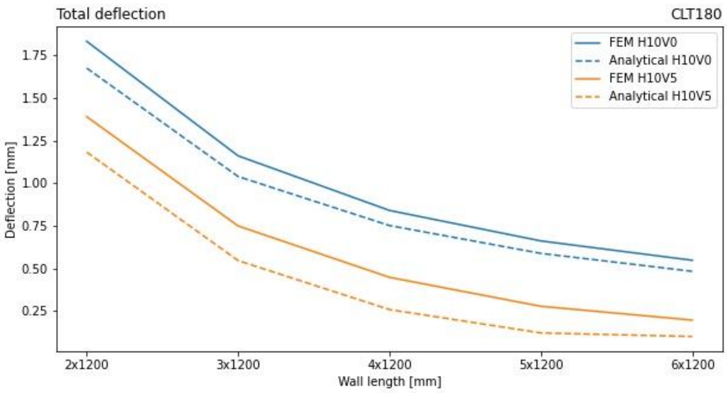

Total deflection (Sum contributions)
CLT100
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

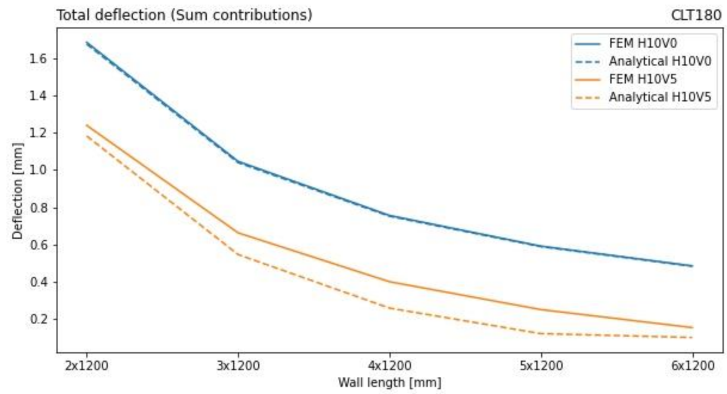

In-plane Shear
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


In-plane bending
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

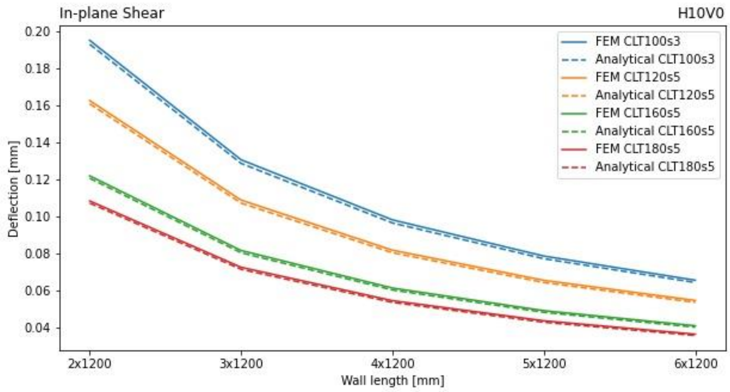

Rigid body sliding
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.06
0.08
0.10
0.12
0.14
0.16
0.18
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

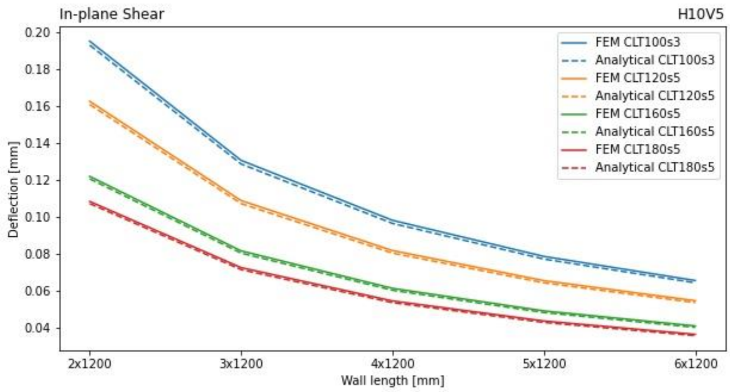

Kinematic rocking
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

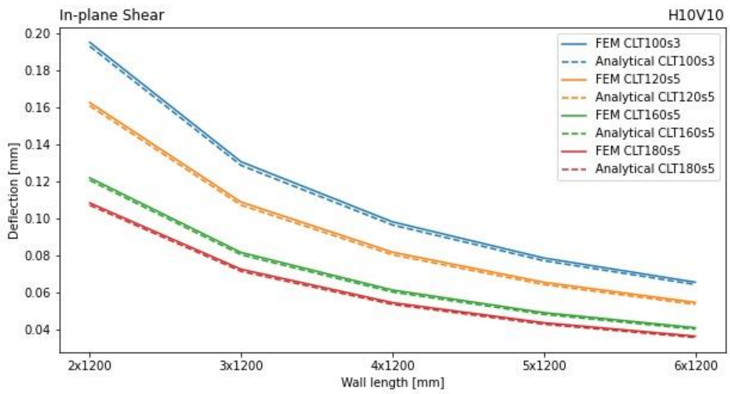

Total deflection
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

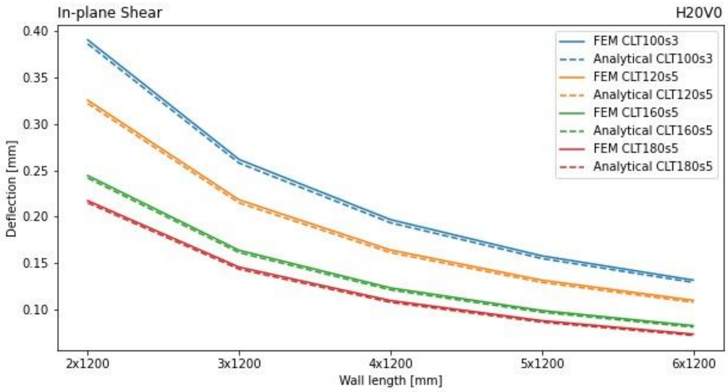

Total deflection (Sum contributions)
CLT180
FEM H10V0
Analytical H10V0
FEM H10V5
Analytical H10V5
Deflection [mm]
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

Lateral deflection due to in-plane Shear of segmented CLT:

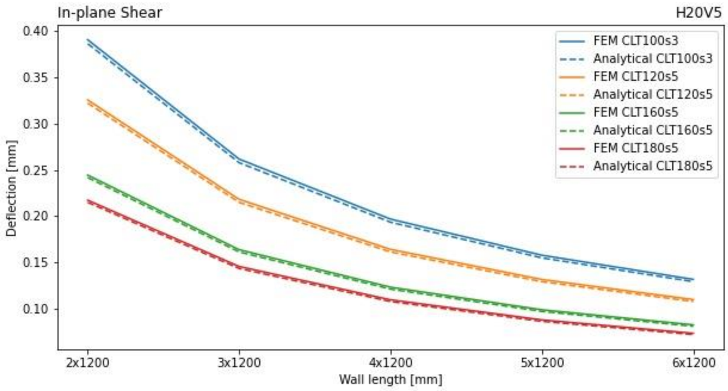




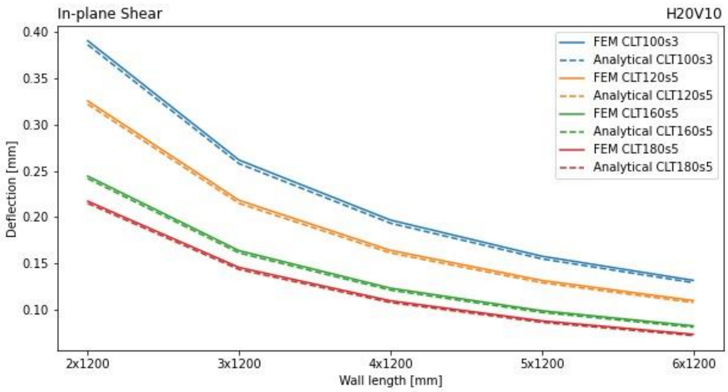




In-plane Shear
H20V0
Deflection [mm]
Wall length [mm]
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
2x1200
3x1200
4x1200
5x1200
6x1200

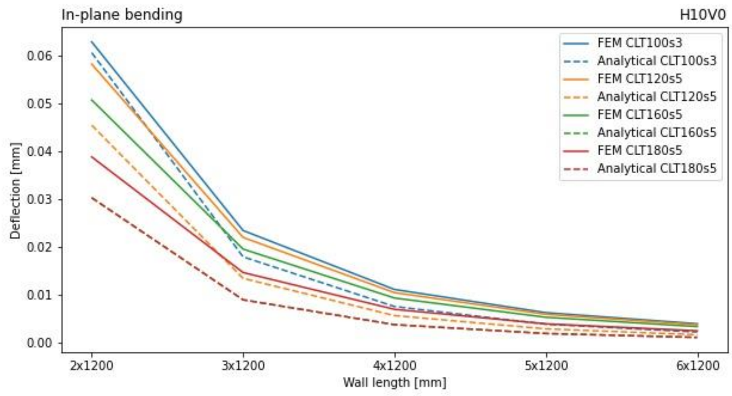


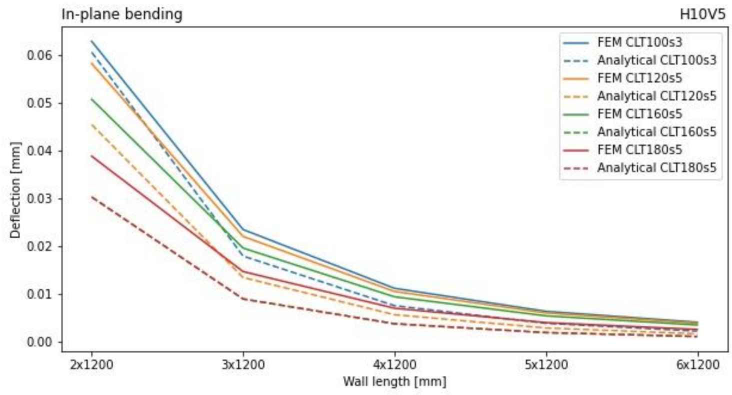
In-plane Shear
H20V5
Deflection [mm]
Wall length [mm]
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
2x1200
3x1200
4x1200
5x1200
6x1200

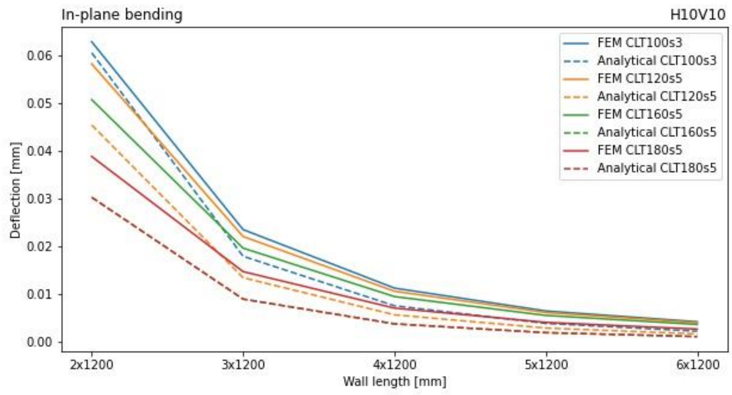


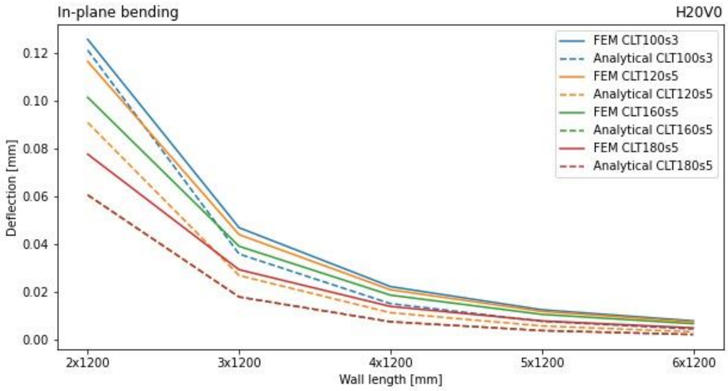
In-plane Shear
H20V10
Deflection [mm]
Wall length [mm]
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
2x1200
3x1200
4x1200
5x1200
6x1200

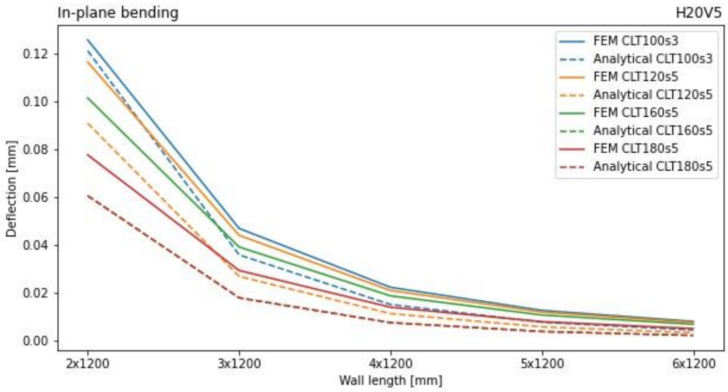
Lateral deflection due to in-plane Bending of segmented CLT:

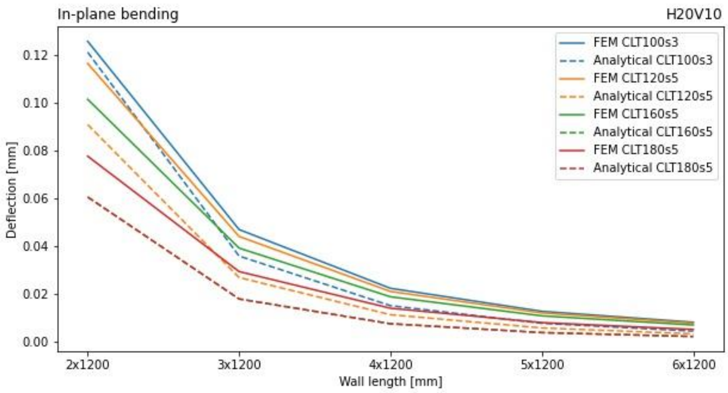




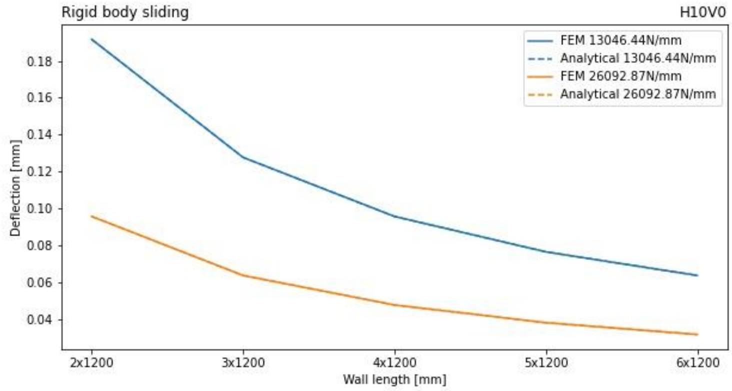


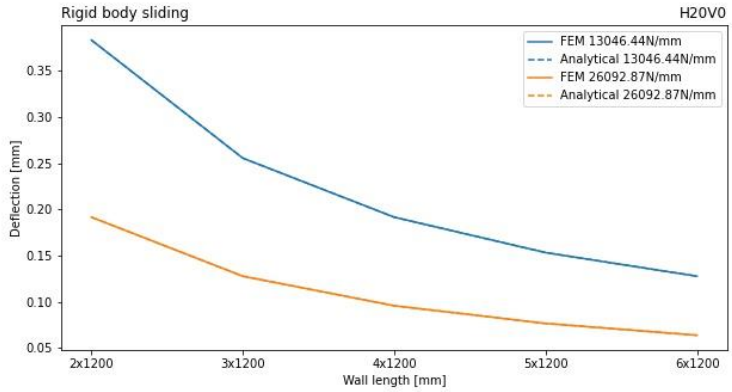

In-plane bending
H20V0
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


In-plane bending
H20V5
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

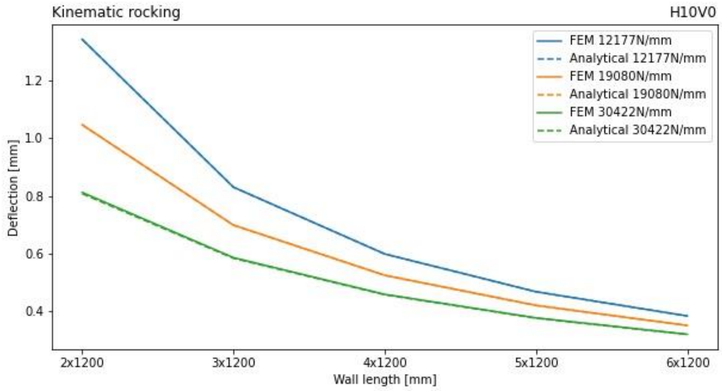

In-plane bending
H20V10
FEM CLT100s3
Analytical CLT100s3
FEM CLT120s5
Analytical CLT120s5
FEM CLT160s5
Analytical CLT160s5
FEM CLT180s5
Analytical CLT180s5
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

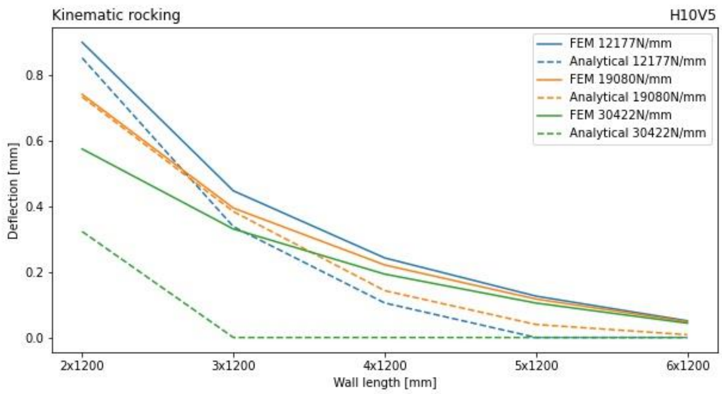
Lateral deflection due to Rigid Body Sliding of segmented CLT:

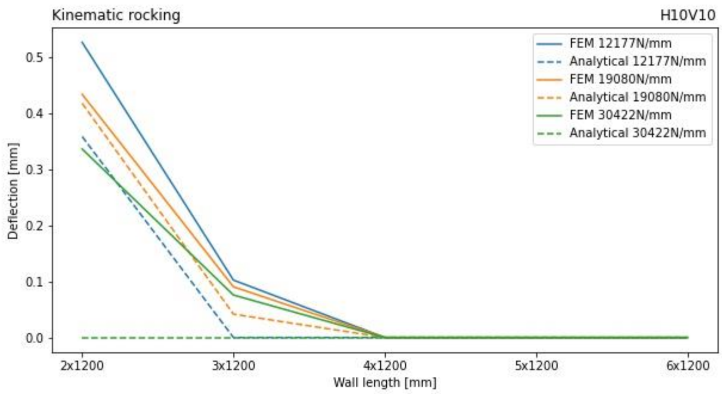


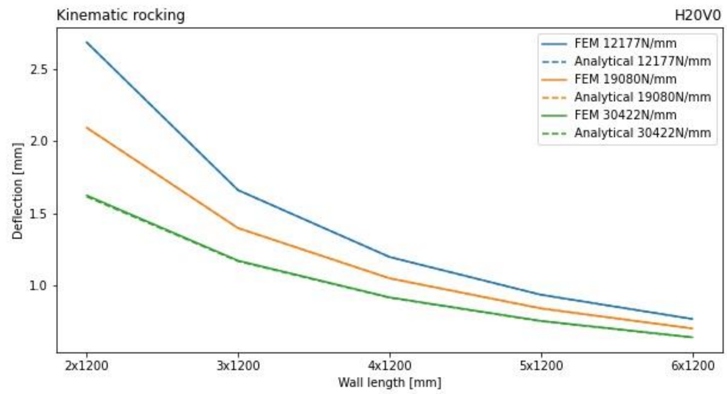


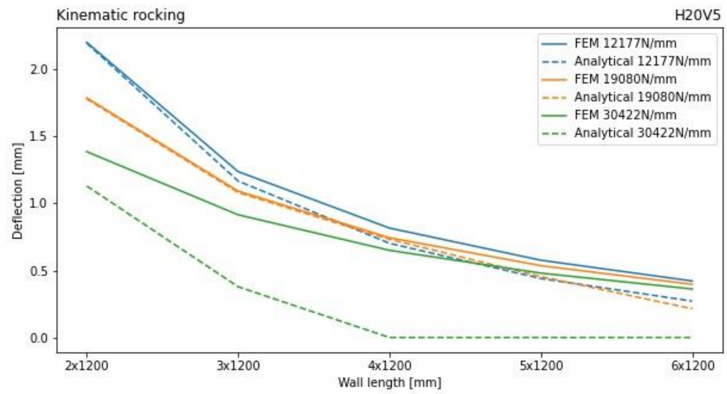
Lateral deflection due to Kinematic Rocking of segmented CLT:

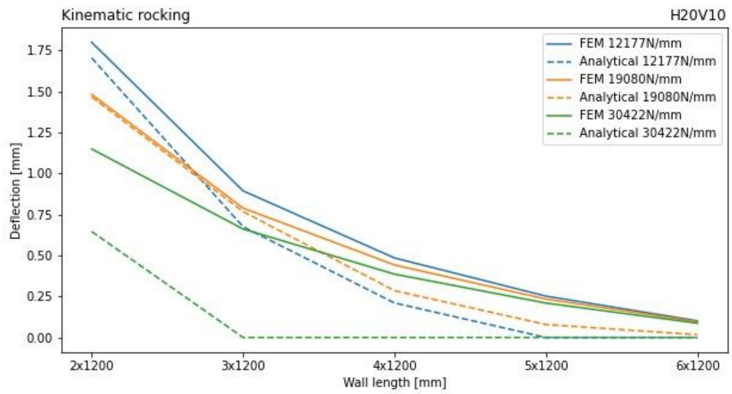


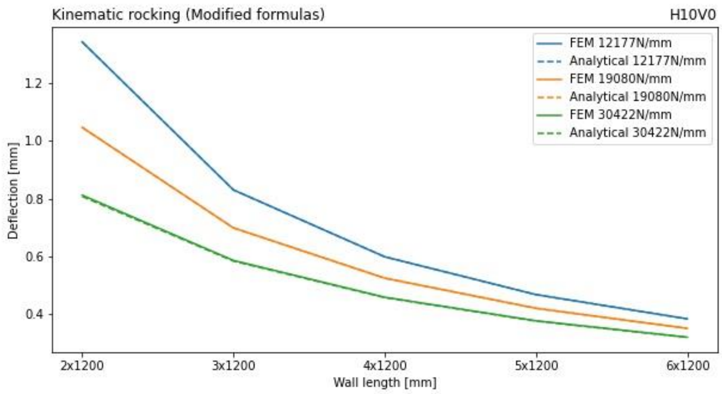


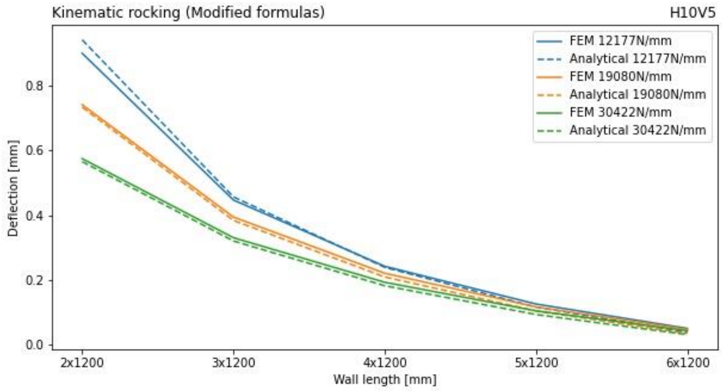


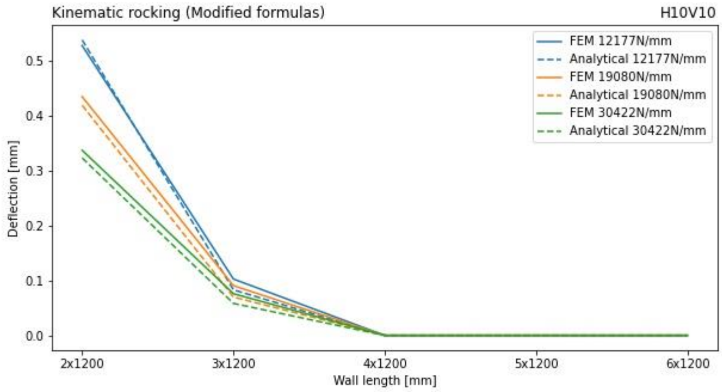

Kinematic rocking
H20V0
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
2.5
2.0
1.5
1.0
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

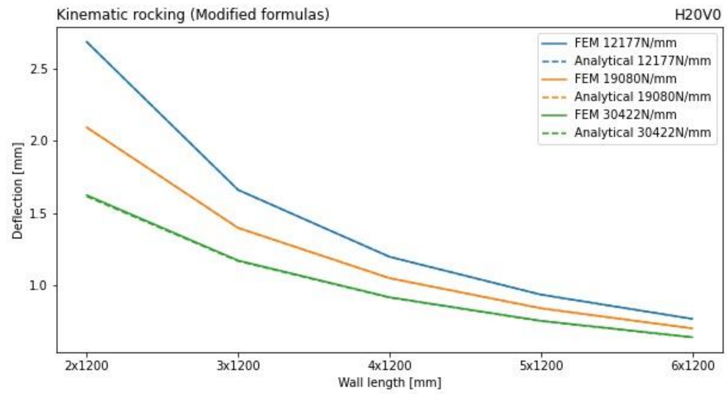

Kinematic rocking
H20V5
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
2.0
1.5
1.0
0.5
0.0
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

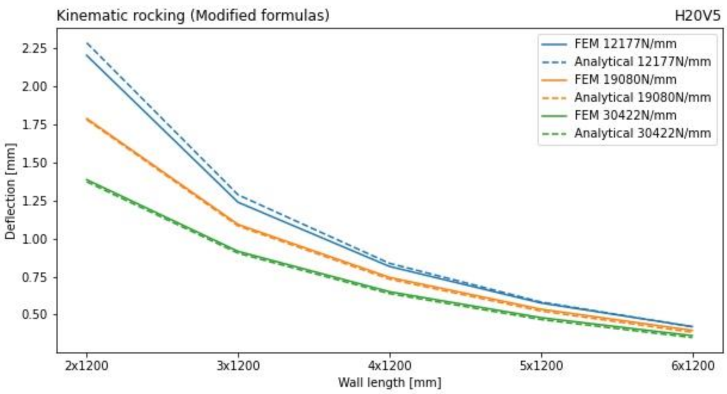

Kinematic rocking
H20V10
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2x1200
3x1200
4x1200
5x1200
6x1200
Wall length [mm]

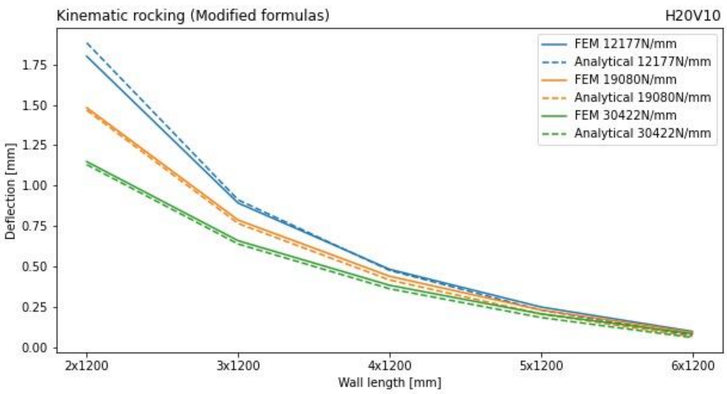

Kinematic rocking (Modified formulas)
H10V0
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


Kinematic rocking (Modified formulas)
H10V5
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


Kinematic rocking (Modified formulas)
H10V10
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


Kinematic rocking (Modified formulas)
H20V0
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


Kinematic rocking (Modified formulas)
H20V5
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200


Kinematic rocking (Modified formulas)
H20V10
FEM 12177N/mm
Analytical 12177N/mm
FEM 19080N/mm
Analytical 19080N/mm
FEM 30422N/mm
Analytical 30422N/mm
Deflection [mm]
Wall length [mm]
2x1200
3x1200
4x1200
5x1200
6x1200

Appendix B - LTF shear wall

```
# -*- coding: utf-8 -*-
"""
Master Thesis
May 2022
Håkon Østraat Sævareid & Johan Bjørkedal

"""
import os
import sys
import comtypes.client
import math as m
import matplotlib.pylab as plt
import pandas as pd

#%% Function

'''
W = Width of one sheet
H = Hight of wall
S = Number of sheets
F = Number of studs per sheet
D = Nail spacing
O = Offset stud to sheet
s = Number of angle brackets
HD = Altering parameter
'''

def LTF(W, H, S, F, D, L, O, s, HD):

#%% Standard Initsiation process for SAP2000

    Filename = 'test'
    APIPath = '*Path to main folder*'

    AttachToInstance = False

    SpecifyPath = False

    if not os.path.exists(APIPath):
            try:
                os.makedirs(APIPath)

            except OSError:
                pass

    ModelPath = APIPath + os.sep + Filename
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)

    if AttachToInstance:
        try:
                mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")

        except (OSError, comtypes.COMError):
            print("No running instance of the program found or failed.")
            sys.exit(-1)
```

```python
    else:
        if SpecifyPath:
            try:
                mySapObject = helper.CreateObject(ProgramPath)

            except (OSError, comtypes.COMError):
                print("Cannot start a new instance of the program from "
                      + ProgramPath)
                sys.exit(-1)

        else:
            try:
                mySapObject = helper.CreateObjectProgID(
                    "CSI.SAP2000.API.SapObject")

            except (OSError, comtypes.COMError):
                print("Cannot start a new instance of the program.")
                sys.exit(-1)

        mySapObject.ApplicationStart()

    SapModel = mySapObject.SapModel
    SapModel.InitializeNewModel()
    ret = SapModel.File.NewBlank()
    ret = SapModel.SetPresentUnits(9)

#%% Initial input

    # Horizontal force
    Fh = [10000]  #N

    # Vertical force
    Fv = [0,5]    #N

    # Sheet thickness
    d_s = 15

    # Perimeterter-studs (cross section)
    h_ps = 148
    b_ps = 96

    # InnerPerimete-studs (cross section)
    h_ips = 148
    b_ips = 48

    # Inner-studs (cross section)
    h_is = 148
    b_is = 48

    # Top-Cord (cross section)
    h_ct = 148
    b_ct = 96

    # Bottom-Cord (cross section)
    h_cb = 148
    b_cb = 48

    # Nail Stiffness
    k_n = 800.41
```

```
    # Angle backet stiffnes
    k_AB = 26092.87

    # Hold-down stiffness
    k_HD = HD #*2

#%% Analytical Calculations

    ANUs = []
    ANUB = []
    ANUA = []
    ANUR = []
    ANUN = []
    ANUC = []
    ANU = []
    for i in range(len(Fh)):
        for n in range(len(Fv)):

            # Inter-storey lateral deflection due to in-plane Shear
            Us = (Fh[i]*H)/((W*S)*(2*d_s*1080))
            ANUs.append(Us)

            # Inter-storey lateral deflection due to in-plane bending
            UB = (Fh[i]*H**3)/(3*(11000*h_ps*b_ps*(W*S)**2)/2)
            ANUB.append(UB)

            # Lateral displacement due to rigid body sliding of the shear wall
            UA = Fh[i]/(s*k_AB)
            ANUA.append(UA)

            # Lateral displacement due to rocking kinematic mode of the wall
            UR = max((((Fh[i]*H)/(k_HD*(W*S-0.1*W*S)**2))-(((Fv[n]*S*W)*
                                (W*S-0.1*W*S))/(2*(k_HD*(W*S-0.1*W*S)**2))))*H,0)
            ANUR.append(UR)

            # Lateral displacement due to the deformation of
            # sheathing-to-framing connections
            UN = (Fh[i]/(W*S)**2)/(2*(k_n/(D*((2*(S*W)+2*S*H)))))
            ANUN.append(UN)

            # Lateral displacement due to the deformation of the bottom rail
            # perpendicular to the grain under the trailing stud
            UC = (((((b_cb)*(Fh[i]*(H/(W*S)) + (Fv[n]*S*W)/(S*(F-1)*2)))/
                    ((2*h_cb*370)))*((1/b_ps)+(1/(b_ps + b_cb))))*(H/(W*S)))
            ANUC.append(UC)

            # Total Deflection
            U = Us + UB + UA + UR + UN + UC
            ANU.append(U)

    ANA = [ANUs,ANUB,ANUA,ANUR,ANUN,ANUC,ANU]

#%% Initial material props

    # Material
    ret = SapModel.PropMaterial.SetMaterial('OSB', 3)
    ret = SapModel.PropMaterial.SetWeightAndMass("OSB",2, (550/10**9)*(1/9.81))
    ret = SapModel.PropMaterial.SetMPOrthotropic("OSB", [4390, 4390, 1980],
                                                 [0.3,0.3,0.3], [0 ,0 ,0],
                                                 [1080, 1080, 0])
```

```python
ret = SapModel.PropMaterial.SetMaterial('C24', 3)
ret = SapModel.PropMaterial.SetWeightAndMass("C24",2, (420/10**9)*(1/9.81))
ret = SapModel.PropMaterial.SetMPIsotropic("C24", 11000 , 0.1, 1.170E-05)

# Links
ret = SapModel.PropLink.SetLinear('Nail',
                            [True, True, True, False, False, False],
                            [True, False, False, False, False, False],
                            [0, k_n, k_n, 0, 0, 0], [0, 0, 0, 0, 0, 0],
                            0, 0, False, False)
ret = SapModel.PropLink.SetGap('Sheet',
                            [True,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [100000000,0,0,0,0,0], [0,0,0,0,0,0],
                            [100000000,0,0,0,0,0],[0,0,0,0,0,0],15,0)
ret = SapModel.PropLink.SetGap('Stud-Rail',
                            [True,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [100000000000,0,0,0,0,0], [0,0,0,0,0,0],
                            [100000000,0,0,0,0,0],[0,0,0,0,0,0],0,0)
ret = SapModel.PropLink.SetGap('Compression-In',
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [((370*h_is*(b_is+ b_cb/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],
                            [((370*h_is*(b_is+ b_cb/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],24,24)
ret = SapModel.PropLink.SetGap('Compression-InPe',
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [((370*h_ips*(b_ips+ b_cb/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],
                            [((370*h_ips*(b_is/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],24,24)
ret = SapModel.PropLink.SetGap('Compression-Pe',
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [((370*h_ps*(b_ps+ b_cb/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],
                            [((370*h_ps*(b_ps/2))/b_cb),0,0,0,0,0],
                            [0,0,0,0,0,0],24,24)
ret = SapModel.PropLink.SetHook('HoldDown',
                            [True,False,False,False,False,False],
                            [False,False,False,False,False,False],
                            [True,False,False,False,False,False],
                            [k_HD,0,0,0,0,0], [0,0,0,0,0,0],
                            [k_HD,0,0,0,0,0],[0,0,0,0,0,0],0,0)
ret = SapModel.PropLink.SetLinear('AngleBracket',
                            [False,True,False,False,False,False],
                            [False,False,False,False,False,False],
                            [0,k_AB,0,0,0,0],[0,0,0,0,0,0], 60, 60,
                            False, False)
# Applying sections and material
ret = SapModel.PropFrame.SetRectangle('Perimeter', 'C24', b_ps, h_ps)
ret = SapModel.PropFrame.SetRectangle('Internal_perimeter', 'C24',
                                                      b_ips, h_ips)
```

```python
    ret = SapModel.PropFrame.SetRectangle('InnerStud', 'C24', b_is, h_is)
    ret = SapModel.PropFrame.SetRectangle('TopCord', 'C24', b_ct, h_ct)
    ret = SapModel.PropFrame.SetRectangle('BottomCord', 'C24', b_cb, h_cb)
    ret = SapModel.PropArea.SetShell_1('A', 5, True, 'OSB', 0, d_s, d_s, 1)

#%% Timber frame

    l = W/(F-1)
    M = [False, False, False, False, True, True]

    for n in range(S):
        for i in range(F-1):
            # First sheat
            if n == 0:
                # First stud
                if i == 0:
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                            'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                            'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
                    if L == 0:
                        ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                            'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
                        ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                            'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                        ret = SapModel.FrameObj.AddByPoint(
                            'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'TL',
                            'F'+str(n)+str(i), 'Perimeter', 'F'+str(n)+str(i))
                        ret = SapModel.LinkObj.AddByPoint(
                            'F'+str(n)+'p'+str(i)+'BR', 'F'+str(n)+'p'+str(i)+'BL',
                            'BL'+str(n)+str(i),False, 'Stud-Rail')
                        ret = SapModel.LinkObj.AddByPoint(
                            'F'+str(n)+'p'+str(i)+'TR', 'F'+str(n)+'p'+str(i)+'TL',
                            'TL'+str(n)+str(i),False, 'Stud-Rail')
                    else:
                        ret = SapModel.FrameObj.AddByPoint(
                            'F'+str(n)+'p'+str(i)+'BR', 'F'+str(n)+'p'+str(i)+'TR',
                            'F'+str(n)+str(i), 'Perimeter', 'F'+str(n)+str(i))
                        ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                            [0,0,0,0,0,0], [0,0,0,0,0,0])
                # All the other studs
                else:
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                        'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                        'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
                    ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'BR',
                        'F'+str(n)+'p'+str(i)+'BR','BR'+str(n)+str(i), 'BottomCord',
                        'BR'+str(n)+str(i))
                    ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'TR',
                        'F'+str(n)+'p'+str(i)+'TR','TR'+str(n)+str(i), 'TopCord',
                        'TR'+str(n)+str(i))
                    if L == 0:
                        ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                            'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
                        ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                            'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                        ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BL',
                            'F'+str(n)+'p'+str(i)+'TL','F'+str(n)+str(i), 'InnerStud',
                            'F'+str(n)+str(i))
```

```
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'BL', 'BL'+str(n)+str(i),False,
                    'Stud-Rail')
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'TR',
                    'F'+str(n)+'p'+str(i)+'TL', 'TL'+str(n)+str(i),False,
                    'Stud-Rail')
            else:
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+str(i), 'InnerStud',
                    'F'+str(n)+str(i))
                ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                [0,0,0,0,0,0], [0,0,0,0,0,0])
    # Last Sheat
    elif n == S-1:
        # First stud
        if i == 0:
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                    'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
            ret = SapModel.FrameObj.AddByPoint('F'+str(n-1)+'p'+str(F-2)+'BR',
                    'F'+str(n)+'p'+str(i)+'BR','BR'+str(n-1)+str(F-1),
                    'BottomCord', 'BR'+str(n-1)+str(F-1))
            ret = SapModel.FrameObj.AddByPoint('F'+str(n-1)+'p'+str(F-2)+'TR',
                    'F'+str(n)+'p'+str(i)+'TR','TR'+str(n-1)+str(F-1),
                    'TopCord', 'TR'+str(n-1)+str(F-1))
            if L == 0:
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                    'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                    'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BL',
                    'F'+str(n)+'p'+str(i)+'TL','F'+str(n)+str(i),
                    'Internal_perimeter', 'F'+str(n)+str(i))
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'BL', 'BL'+str(n)+str(i),False,
                    'Stud-Rail')
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'TR',
                    'F'+str(n)+'p'+str(i)+'TL', 'TL'+str(n)+str(i),False,
                    'Stud-Rail')
            else:
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+str(i),
                    'Internal_perimeter', 'F'+str(n)+str(i))
                ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                [0,0,0,0,0,0], [0,0,0,0,0,0])
        # All other studs
        else:
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                    'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
            ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'BR',
                    'F'+str(n)+'p'+str(i)+'BR','BR'+str(n)+str(i),
                    'BottomCord', 'BR'+str(n)+str(i))
            ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'TR',
                    'F'+str(n)+'p'+str(i)+'TR','TR'+str(n)+str(i),
                    'TopCord', 'TR'+str(n)+str(i))
            if L == 0:
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                    'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
```

```
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                    'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BL',
                    'F'+str(n)+'p'+str(i)+'TL','F'+str(n)+str(i),
                    'InnerStud', 'F'+str(n)+str(i))

                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'BL', 'BL'+str(n)+str(i),False,
                    'Stud-Rail')
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'TR',
                    'F'+str(n)+'p'+str(i)+'TL', 'TL'+str(n)+str(i),False,
                    'Stud-Rail')
            else:
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+str(i),
                    'InnerStud', 'F'+str(n)+str(i))
                ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                    [0,0,0,0,0,0], [0,0,0,0,0,0])
    # All the other sheats
    else:
        # First stud
        if i == 0:
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                    'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
            ret = SapModel.FrameObj.AddByPoint('F'+str(n-1)+'p'+str(F-2)+'BR',
                    'F'+str(n)+'p'+str(i)+'BR','BR'+str(n-1)+str(F-1),
                    'BottomCord', 'BR'+str(n-1)+str(F-1))
            ret = SapModel.FrameObj.AddByPoint('F'+str(n-1)+'p'+str(F-2)+'TR',
                    'F'+str(n)+'p'+str(i)+'TR','TR'+str(n-1)+str(F-1),
                    'TopCord', 'TR'+str(n-1)+str(F-1))
            if L == 0:
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                    'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
                ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                    'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BL',
                    'F'+str(n)+'p'+str(i)+'TL','F'+str(n)+str(i),
                    'Internal_perimeter', 'F'+str(n)+str(i))
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'BL', 'BL'+str(n)+str(i),False,
                    'Stud-Rail')
                ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'TR',
                    'F'+str(n)+'p'+str(i)+'TL', 'TL'+str(n)+str(i),False,
                    'Stud-Rail')
            else:
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+str(i),
                    'Internal_perimeter', 'F'+str(n)+str(i))
                ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                    [0,0,0,0,0,0], [0,0,0,0,0,0])
        # All other studs
        else:
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,0,
                    'F'+str(n)+'p'+str(i)+'BR','F'+str(n)+'p'+str(i)+'BR')
            ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H,
                    'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+'p'+str(i)+'TR')
            ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'BR',
                    'F'+str(n)+'p'+str(i)+'BR','BR'+str(n)+str(i),
                    'BottomCord', 'BR'+str(n)+str(i))
```

```
                ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i-1)+'TR',
                        'F'+str(n)+'p'+str(i)+'TR','TR'+str(n)+str(i),
                        'TopCord', 'TR'+str(n)+str(i))
                if L == 0:
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,D,
                        'F'+str(n)+'p'+str(i)+'BL', 'F'+str(n)+'p'+str(i)+'BL')
                    ret = SapModel.PointObj.AddCartesian(i*l+n*W,0,H-D,
                        'F'+str(n)+'p'+str(i)+'TL', 'F'+str(n)+'p'+str(i)+'TL')
                    ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BL',
                        'F'+str(n)+'p'+str(i)+'TL','F'+str(n)+str(i),
                        'InnerStud', 'F'+str(n)+str(i))
                    ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                        'F'+str(n)+'p'+str(i)+'BL', 'BL'+str(n)+str(i),False,
                        'Stud-Rail')
                    ret = SapModel.LinkObj.AddByPoint('F'+str(n)+'p'+str(i)+'TR',
                        'F'+str(n)+'p'+str(i)+'TL', 'TL'+str(n)+str(i),False,
                        'Stud-Rail')
                else:
                    ret = SapModel.FrameObj.AddByPoint('F'+str(n)+'p'+str(i)+'BR',
                        'F'+str(n)+'p'+str(i)+'TR','F'+str(n)+str(i),
                        'InnerStud', 'F'+str(n)+str(i))
                    ret = SapModel.FrameObj.SetReleases('F'+str(n)+str(i), M, M,
                                                    [0,0,0,0,0,0], [0,0,0,0,0,0])

ret = SapModel.PointObj.AddCartesian(S*W,0,0,'F'+str(S-1)+'p'+str(F-1)+'BR',
                                     'F'+str(S-1)+'p'+str(F-1)+'BR')
ret = SapModel.PointObj.AddCartesian(S*W,0,H,'F'+str(S-1)+'p'+str(F-1)+'TR',
                                     'F'+str(S-1)+'p'+str(F-1)+'TR')
if L == 0:
    ret = SapModel.PointObj.AddCartesian(S*W,0,0+D,'F'+str(S-1)+'p'+str(F-1)+'BL',
                                         'F'+str(S-1)+'p'+str(F-1)+'BL')
    ret = SapModel.PointObj.AddCartesian(S*W,0,H-D,'F'+str(S-1)+'p'+str(F-1)+'TL',
                                         'F'+str(S-1)+'p'+str(F-1)+'TL')
    ret = SapModel.FrameObj.AddByPoint('F'+str(S-1)+'p'+str(F-1)+'BL',
                        'F'+str(S-1)+'p'+str(F-1)+'TL', 'F'+str(S-1)+str(F-1),
                        'Perimeter', 'F'+str(S-1)+str(F-1))
    ret = SapModel.LinkObj.AddByPoint('F'+str(S-1)+'p'+str(F-1)+'BR',
                        'F'+str(S-1)+'p'+str(F-1)+'BL', 'BL'+str(S-1)+str(F-1),
                        False, 'Stud-Rail')
    ret = SapModel.LinkObj.AddByPoint('F'+str(S-1)+'p'+str(F-1)+'TR',
                        'F'+str(S-1)+'p'+str(F-1)+'TL', 'TL'+str(S-1)+str(F-1),
                        False, 'Stud-Rail')
else:
    ret = SapModel.FrameObj.AddByPoint('F'+str(S-1)+'p'+str(F-1)+'BR',
                        'F'+str(S-1)+'p'+str(F-1)+'TR', 'F'+str(S-1)+str(F-1),
                        'Perimeter', 'F'+str(S-1)+str(F-1))
    ret = SapModel.FrameObj.SetReleases('F'+str(S-1)+str(F-1),
                        [False, False, False, False, True, True],
                        [False, False, False, False, True, True],
                        [0,0,0,0,0,0], [0,0,0,0,0,0])
ret = SapModel.FrameObj.AddByPoint('F'+str(S-1)+'p'+str(F-2)+'BR',
                        'F'+str(S-1)+'p'+str(F-1)+'BR','BR'+str(S-1)+str(F-1),
                        'BottomCord', 'BR'+str(S-1)+str(F-1))
ret = SapModel.FrameObj.AddByPoint('F'+str(S-1)+'p'+str(F-2)+'TR',
                        'F'+str(S-1)+'p'+str(F-1)+'TR','TR'+str(S-1)+str(F-1),
                        'TopCord', 'TR'+str(S-1)+str(F-1))
```

```python
#%% Sheeting
    l = W/(F-1)
    for i in range(S):
        for n in range(F-1):
            if i != S-1 and n == F-2:
                ret = SapModel.PointObj.AddCartesian(i*W+l*n,O,0,
                            'A'+str(i)+'F'+str(n)+'p0f','A'+str(i)+'F'+str(n)+'p0f')
                ret = SapModel.PointObj.AddCartesian((i+1)*W-15,O,0,
                            'A'+str(i)+'F'+str(n)+'p1f','A'+str(i)+'F'+str(n)+'p1f')
                ret = SapModel.PointObj.AddCartesian((i+1)*W-15,O,H,
                            'A'+str(i)+'F'+str(n)+'p2f','A'+str(i)+'F'+str(n)+'p2f')
                ret = SapModel.PointObj.AddCartesian(i*W+l*n,O,H,
                            'A'+str(i)+'F'+str(n)+'p3f','A'+str(i)+'F'+str(n)+'p3f')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0f',
                            'A'+str(i)+'F'+str(n)+'p1f','A'+str(i)+'F'+str(n)+'p2f',
                            'A'+str(i)+'F'+str(n)+'p3f'], 'A'+str(i)+'F'+str(n)+'f',
                                                  'OSB', 'A'+str(i)+'F'+str(n)+'f')
                ret = SapModel.PointObj.AddCartesian(i*W+l*n,-O,0,
                            'A'+str(i)+'F'+str(n)+'p0r','A'+str(i)+'F'+str(n)+'p0r')
                ret = SapModel.PointObj.AddCartesian((i+1)*W-15,-O,0,
                            'A'+str(i)+'F'+str(n)+'p1r','A'+str(i)+'F'+str(n)+'p1r')
                ret = SapModel.PointObj.AddCartesian((i+1)*W-15,-O,H,
                            'A'+str(i)+'F'+str(n)+'p2r','A'+str(i)+'F'+str(n)+'p2r')
                ret = SapModel.PointObj.AddCartesian(i*W+l*n,-O,H,
                            'A'+str(i)+'F'+str(n)+'p3r','A'+str(i)+'F'+str(n)+'p3r')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0r',
                                                      'A'+str(i)+'F'+str(n)+'p1r',
                                                      'A'+str(i)+'F'+str(n)+'p2r',
                                                      'A'+str(i)+'F'+str(n)+'p3r'],
                                                  'A'+str(i)+'F'+str(n)+'r', 'OSB',
                                                  'A'+str(i)+'F'+str(n)+'r')

            elif i != 0 and n == 0:
                ret = SapModel.PointObj.AddCartesian(i*W+15,O,0,
                            'A'+str(i)+'F'+str(n)+'p0f','A'+str(i)+'F'+str(n)+'p0f')
                ret = SapModel.PointObj.AddCartesian(i*W+l,O,0,
                            'A'+str(i)+'F'+str(n)+'p1f','A'+str(i)+'F'+str(n)+'p1f')
                ret = SapModel.PointObj.AddCartesian(i*W+l,O,H,
                            'A'+str(i)+'F'+str(n)+'p2f','A'+str(i)+'F'+str(n)+'p2f')
                ret = SapModel.PointObj.AddCartesian(i*W+15,O,H,
                            'A'+str(i)+'F'+str(n)+'p3f','A'+str(i)+'F'+str(n)+'p3f')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0f',
                            'A'+str(i)+'F'+str(n)+'p1f','A'+str(i)+'F'+str(n)+'p2f',
                            'A'+str(i)+'F'+str(n)+'p3f'], 'A'+str(i)+'F'+str(n)+'f',
                                                  'OSB', 'A'+str(i)+'F'+str(n)+'f')

                ret = SapModel.PointObj.AddCartesian(i*W+15,-O,0,
                            'A'+str(i)+'F'+str(n)+'p0r','A'+str(i)+'F'+str(n)+'p0r')
                ret = SapModel.PointObj.AddCartesian(i*W+l,-O,0,
                            'A'+str(i)+'F'+str(n)+'p1r','A'+str(i)+'F'+str(n)+'p1r')
                ret = SapModel.PointObj.AddCartesian(i*W+l,-O,H,
                            'A'+str(i)+'F'+str(n)+'p2r','A'+str(i)+'F'+str(n)+'p2r')
                ret = SapModel.PointObj.AddCartesian(i*W+15,-O,H,
                            'A'+str(i)+'F'+str(n)+'p3r','A'+str(i)+'F'+str(n)+'p3r')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0r',
                                                      'A'+str(i)+'F'+str(n)+'p1r',
                                                      'A'+str(i)+'F'+str(n)+'p2r',
                                                      'A'+str(i)+'F'+str(n)+'p3r'],
                                                  'A'+str(i)+'F'+str(n)+'r', 'OSB',
                                                  'A'+str(i)+'F'+str(n)+'r')
```

```
            else:
                ret = SapModel.PointObj.AddCartesian(i*W+n*l,O,0,
                            'A'+str(i)+'F'+str(n)+'p0f','A'+str(i)+'F'+str(n)+'p0f')
                ret = SapModel.PointObj.AddCartesian(i*W+(n+1)*l,O,0,
                            'A'+str(i)+'F'+str(n)+'p1f','A'+str(i)+'F'+str(n)+'p1f')
                ret = SapModel.PointObj.AddCartesian(i*W+(n+1)*l,O,H,
                            'A'+str(i)+'F'+str(n)+'p2f','A'+str(i)+'F'+str(n)+'p2f')
                ret = SapModel.PointObj.AddCartesian(i*W+n*l,O,H,
                            'A'+str(i)+'F'+str(n)+'p3f','A'+str(i)+'F'+str(n)+'p3f')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0f',
                                                      'A'+str(i)+'F'+str(n)+'p1f',
                                                      'A'+str(i)+'F'+str(n)+'p2f',
                                                      'A'+str(i)+'F'+str(n)+'p3f'],
                                                  'A'+str(i)+'F'+str(n)+'f', 'OSB',
                                                  'A'+str(i)+'F'+str(n)+'f')

                ret = SapModel.PointObj.AddCartesian(i*W+n*l,-O,0,
                            'A'+str(i)+'F'+str(n)+'p0r','A'+str(i)+'F'+str(n)+'p0r')
                ret = SapModel.PointObj.AddCartesian(i*W+(n+1)*l,-O,0,
                            'A'+str(i)+'F'+str(n)+'p1r','A'+str(i)+'F'+str(n)+'p1r')
                ret = SapModel.PointObj.AddCartesian(i*W+(n+1)*l,-O,H,
                            'A'+str(i)+'F'+str(n)+'p2r','A'+str(i)+'F'+str(n)+'p2r')
                ret = SapModel.PointObj.AddCartesian(i*W+n*l,-O,H,
                            'A'+str(i)+'F'+str(n)+'p3r','A'+str(i)+'F'+str(n)+'p3r')
                ret = SapModel.AreaObj.AddByPoint(4, ['A'+str(i)+'F'+str(n)+'p0r',
                                                      'A'+str(i)+'F'+str(n)+'p1r',
                                                      'A'+str(i)+'F'+str(n)+'p2r',
                                                      'A'+str(i)+'F'+str(n)+'p3r'],
                                                  'A'+str(i)+'F'+str(n)+'r', 'OSB',
                                                  'A'+str(i)+'F'+str(n)+'r')

#%% Nails

# Following the studs (Vertical)
    n1 = m.ceil(H/D)
    w = (H-(n1*D))/2

    for i in range(S):
        for n in range(F-1):
            for k in range(n1+1):

                if i == 0:
                    if n == 0:
                        if k != 0 and k != n1:
                            ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                                i*W+n*l, O, w+k*D,
                                'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f', False,
                                'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f')
                            ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                                i*W+n*l,-O, w+k*D,
                                'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r', False,
                                'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r')
                    else:
                        if k % 2 == 0 and k != n1 and k != 0:
                            ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                                i*W+n*l, O, w+k*D,
                                'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f', False,
                                'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f')
```

```
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                        i*W+n*l,-O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r')

        else:
            if n == 0:
                if k != 0 and k != n1:
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l-15, 0,
                        w+k*D, i*W+n*l-15, O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'fl', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'fl')
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l-15, 0,
                        w+k*D, i*W+n*l-15,-O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'rl', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'rl')
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l+15, 0,
                        w+k*D, i*W+n*l+15, O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'fr', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'fr')
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l+15, 0,
                        w+k*D, i*W+n*l+15,-O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'rr', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'rr')
                    ret = SapModel.PointObj.AddCartesian(i*W+n*l, 0,
                        w+k*D, 'PL_S'+str(i)+'_N'+str(k),
                        'PL_S'+str(i)+'_N'+str(k))
                    ret = SapModel.PointObj.AddCartesian(i*W+n*l-15, 0,
                        w+k*D, 'PL_S'+str(i)+'_N'+str(k)+'l',
                        'PL_S'+str(i)+'_N'+str(k)+'l')
                    ret = SapModel.PointObj.AddCartesian(i*W+n*l+15, 0,
                        w+k*D, 'PL_S'+str(i)+'_N'+str(k)+'r',
                        'PL_S'+str(i)+'_N'+str(k)+'r')
                    ret = SapModel.ConstraintDef.SetBody(
                        'CONS_S'+str(i)+'_N'+str(k),
                        [True, True, True, True, True, True])
                    ret = SapModel.PointObj.SetConstraint(
                        'PL_S'+str(i)+'_N'+str(k),
                        'CONS_S'+str(i)+'_N'+str(k))
                    ret = SapModel.PointObj.SetConstraint(
                        'PL_S'+str(i)+'_N'+str(k)+'r',
                        'CONS_S'+str(i)+'_N'+str(k))
                    ret = SapModel.PointObj.SetConstraint(
                        'PL_S'+str(i)+'_N'+str(k)+'l',
                        'CONS_S'+str(i)+'_N'+str(k))

            else:
                if k % 2 == 0 and k != n1 and k != 0:
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                        i*W+n*l, O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'f')
                    ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, w+k*D,
                        i*W+n*l,-O, w+k*D,
                        'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r', False,
                        'Nail', 'L_S'+str(i)+'_F'+str(n)+'_N'+str(k)+'r')
        if i == S-1 and n == F-2 and k != n1 and k != 0:
            ret = SapModel.LinkObj.AddByCoord((S-1)*W+(F-1)*l, 0, w+k*D,
                        (S-1)*W+(F-1)*l, O, w+k*D,
                        'L_S'+str(S-1)+'_F'+str(F-1)+'_N'+str(k)+'f', False,
                        'Nail', 'L_S'+str(S-1)+'_F'+str(F-1)+'_N'+str(k)+'f')
```

```
                    ret = SapModel.LinkObj.AddByCoord((S-1)*W+(F-1)*l, 0, w+k*D,
                          (S-1)*W+(F-1)*l,-O, w+k*D,
                          'L_S'+str(S-1)+'_F'+str(F-1)+'_N'+str(k)+'r', False,
                          'Nail', 'L_S'+str(S-1)+'_F'+str(F-1)+'_N'+str(k)+'r')

# Following the cords (Horizontal)
    for n in range(m.ceil(S*W/D)):
        if D*n % W != 0 or n == 0:
            ret = SapModel.LinkObj.AddByCoord(D*n, 0, 0, D*n, O, 0,
                  'L_N'+str(n)+'fb', False, 'Nail', 'L_N'+str(n)+'fb')
            ret = SapModel.LinkObj.AddByCoord(D*n, 0, H, D*n, O, H,
                  'L_N'+str(n)+'ft', False, 'Nail', 'L_N'+str(n)+'ft')
            ret = SapModel.LinkObj.AddByCoord(D*n, 0, 0, D*n,-O, 0,
                  'L_N'+str(n)+'rb', False, 'Nail', 'L_N'+str(n)+'rb')
            ret = SapModel.LinkObj.AddByCoord(D*n, 0, H, D*n,-O, H,
                  'L_N'+str(n)+'rt', False, 'Nail', 'L_N'+str(n)+'rt')
        else:
            ret = SapModel.LinkObj.AddByCoord(D*n+15, 0, 0, D*n+15, O, 0,
                  'L_N'+str(n)+'fb', False, 'Nail', 'L_N'+str(n)+'fb')
            ret = SapModel.LinkObj.AddByCoord(D*n+15, 0, H, D*n+15, O, H,
                  'L_N'+str(n)+'ft', False, 'Nail', 'L_N'+str(n)+'ft')
            ret = SapModel.LinkObj.AddByCoord(D*n+15, 0, 0, D*n+15,-O, 0,
                  'L_N'+str(n)+'rb', False, 'Nail', 'L_N'+str(n)+'rb')
            ret = SapModel.LinkObj.AddByCoord(D*n+15, 0, H, D*n+15,-O, H,
                  'L_N'+str(n)+'rt', False, 'Nail', 'L_N'+str(n)+'rt')
            ret = SapModel.LinkObj.AddByCoord(D*n-15, 0, 0, D*n-15, O, 0,
                  'L_N'+str(n)+'fb', False, 'Nail', 'L_N'+str(n)+'fb')
            ret = SapModel.LinkObj.AddByCoord(D*n-15, 0, H, D*n-15, O, H,
                  'L_N'+str(n)+'ft', False, 'Nail', 'L_N'+str(n)+'ft')
            ret = SapModel.LinkObj.AddByCoord(D*n-15, 0, 0, D*n-15,-O, 0,
                  'L_N'+str(n)+'rb', False, 'Nail', 'L_N'+str(n)+'rb')

            ret = SapModel.LinkObj.AddByCoord(D*n-15, 0, H, D*n-15,-O, H,
                  'L_N'+str(n)+'rt', False, 'Nail', 'L_N'+str(n)+'rt')
    ret = SapModel.LinkObj.AddByCoord(S*W, 0, 0, S*W, O, 0,
                  'L_N'+str(m.ceil(S*W/D))+'fb', False, 'Nail',
                  'L_N'+str(m.ceil(S*W/D))+'fb')
    ret = SapModel.LinkObj.AddByCoord(S*W, 0, H, S*W, O, H,
                  'L_N'+str(m.ceil(S*W/D))+'ft', False, 'Nail',
                  'L_N'+str(m.ceil(S*W/D))+'ft')
    ret = SapModel.LinkObj.AddByCoord(S*W, 0, 0, S*W,-O, 0,
                  'L_N'+str(m.ceil(S*W/D))+'rb', False, 'Nail',
                  'L_N'+str(m.ceil(S*W/D))+'rb')
    ret = SapModel.LinkObj.AddByCoord(S*W, 0, H, S*W,-O, H,
                  'L_N'+str(m.ceil(S*W/D))+'rt', False, 'Nail',
                  'L_N'+str(m.ceil(S*W/D))+'rt')

#%% Links between meeting sheets
    n1 = m.ceil(H/D)
    w = (H-(n1*D))/2

    if L == 0:
        for i in range(S):
            for k in range(n1+1):
                if i != 0 and k == 0:
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, O, 0, i*W+15, O, 0,
                        'L_SS'+str(i)+'_N'+str(k)+'f', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'f')
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, -O, 0, i*W+15,-O, 0,
                        'L_SS'+str(i)+'_N'+str(k)+'r', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'r')
```

```
                elif i != 0 and k == n1:
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, O, H, i*W+15, O, H,
                        'L_SS'+str(i)+'_N'+str(k)+'f', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'f')
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, -O, H, i*W+15,-O, H,
                        'L_SS'+str(i)+'_N'+str(k)+'r', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'r')
                elif i != 0:
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, O, w+k*D, i*W+15,
                        O, w+k*D, 'L_SS'+str(i)+'_N'+str(k)+'f', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'f')
                    ret = SapModel.LinkObj.AddByCoord(i*W-15, -O, w+k*D, i*W+15,
                        -O, w+k*D, 'L_SS'+str(i)+'_N'+str(k)+'r', False, 'Sheet',
                        'L_SS'+str(i)+'_N'+str(k)+'r')

#%% Supports

    a = (W*S)/(s+1)

    for i in range(S+1):
        if i == 0 or i == S:
            ret = SapModel.LinkObj.AddByCoord(i*W, 0, 0, i*W, 0, -60,
                        'LH_S'+str(i), False, 'HoldDown', 'LH_S'+str(i))
        for n in range(F-1):
            if i < S and n != 0:
                ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, 0, i*W+n*l, 0, -60,
                        'LC_S'+str(i)+'_F'+str(n), False, 'Compression-In',
                        'LC_S'+str(i)+'_F'+str(n))
                ret = SapModel.PointObj.AddCartesian(i*W+n*l, 0, -60,
                        'PRC_S'+str(i)+'_F'+str(n),'PRC_S'+str(i)+'_F'+str(n))

                ret = SapModel.PointObj.SetRestraint('PRC_S'+str(i)+'_F'+str(n),
                                                [True,True,True,True,True,True])
            elif i < S and n == 0 and i != 0:
                ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, 0, i*W+n*l, 0, -60,
                        'LC_S'+str(i)+'_F'+str(n), False, 'Compression-InPe',
                        'LC_S'+str(i)+'_F'+str(n))
                ret = SapModel.PointObj.AddCartesian(i*W+n*l, 0, -60,
                        'PRC_S'+str(i)+'_F'+str(n),'PRC_S'+str(i)+'_F'+str(n))
                ret = SapModel.PointObj.SetRestraint('PRC_S'+str(i)+'_F'+str(n),
                                                [True,True,True,True,True,True])
            elif i == 0 and n == 0:
                ret = SapModel.LinkObj.AddByCoord(i*W+n*l, 0, 0, i*W+n*l, 0, -60,
                        'LC_S'+str(i)+'_F'+str(n), False, 'Compression-Pe',
                        'LC_S'+str(i)+'_F'+str(n))
                ret = SapModel.PointObj.AddCartesian(i*W+n*l, 0, -60,
                        'PRC_S'+str(i)+'_F'+str(n),'PRC_S'+str(i)+'_F'+str(n))
                ret = SapModel.PointObj.SetRestraint('PRC_S'+str(i)+'_F'+str(n),
                                                [True,True,True,True,True,True])
            else:
                if n == 0:
                    ret = SapModel.LinkObj.AddByCoord(i*W, 0, 0, i*W, 0, -60,
                        'LC_S'+str(i)+'_F'+str(n), False, 'Compression-Pe',
                        'LC_S'+str(i)+'_F'+str(n))
                    ret = SapModel.PointObj.AddCartesian(i*W, 0, -60,
                        'PRC_S'+str(i)+'_F'+str(n),'PRC_S'+str(i)+'_F'+str(n))
                    ret = SapModel.PointObj.SetRestraint(
                        'PRC_S'+str(i)+'_F'+str(n), [True,True,True,True,True,True])
```

```python
    for i in range(s):
        if D % a < 5:
            ret = SapModel.LinkObj.AddByCoord(a*(i+1)+10, 0, 0, a*(i+1)+30,
                                    0, -10, 'LAB_S'+str(i), False, 'AngleBracket',
                                    'LAB_S'+str(i))
            ret = SapModel.PointObj.AddCartesian(a*(i+1)+10, 0, -60,
                                    'PAB_S'+str(i),'PAB_S'+str(i))
            ret = SapModel.PointObj.SetRestraint('PAB_S'+str(i),
                                    [True,True,True,True,True,True])

        else:
            ret = SapModel.LinkObj.AddByCoord(a*(i+1), 0, 0, a*(i+1), 0, -60,
                                    'LAB_S'+str(i), False, 'AngleBracket',
                                    'LAB_S'+str(i))
            ret = SapModel.PointObj.AddCartesian(a*(i+1), 0, -60,
                                    'PAB_S'+str(i),'PAB_S'+str(i))
            ret = SapModel.PointObj.SetRestraint('PAB_S'+str(i),
                                    [True,True,True,True,True,True])

    ret = SapModel.View.RefreshView(0, False)

#%% Mesh

    # Defining group and adding area elements to the group
    ret = SapModel.GroupDef.SetGroup("GroupOSB-first")
    ret = SapModel.SelectObj.PropertyMaterial("OSB")
    ret = SapModel.AreaObj.SetGroupAssign("", "GroupOSB-first", False, 2)
    SapModel.SelectObj.All()
    SapModel.SelectObj.InvertSelection()



    # Fetching a list of the names of the areas
    numobj=0
    obtype=[]
    shellnamesfirst=[]
    [numobj, obtype, shellnamesfirst, ret] = SapModel.GroupDef.GetAssignments(
        "GroupOSB-first", numobj, obtype, shellnamesfirst)

    # Dividing areas at links to make them connect using the group
    for s in shellnamesfirst:
        areanames=[]
        numberareas=0
        SapModel.SelectObj.All()
        ret=SapModel.EditArea.Divide(str(s), 3, numberareas, areanames,
                                     PointOnEdgeFromPoint=True)

    # Creating the automatic mesh
    ret = SapModel.AreaObj.SetAutoMesh("GroupOSB-first", 2, MaxSize1 = 50,
                                       MaxSize2 =50, ItemType = 1 )


#%% Applying load

# Creating load pattern for horizontal load and applying
    for i in range(len(Fh)):
        ret = SapModel.LoadPatterns.Add("Fh"+str(i+1), 3, 0, False)
        ret = SapModel.PointObj.SetLoadForce("F0p0TR",'Fh'+str(i+1),
                                             [Fh[i],0,0,0,0,0])
```

```
# Creating load pattern for vetrical load
    for i in range(len(Fv)):
        ret = SapModel.LoadPatterns.Add("Fv"+str(i+1), 3, 0, False)
# Applying loads to for vetrical load patterns
    for n in range(S):
        for i in range(F):
            for q in range(len(Fv)):
                if n == S-1 and i == F-1:
                    ret = SapModel.PointObj.SetLoadForce('F'+str(S-1)+'p'+str(F-1)+'TR',
                            'Fv'+str(q+1),[0,0,-Fv[q]*S*W/(S*(F-1)*2),0,0,0])
                elif n == 0 and i == 0:
                    ret = SapModel.PointObj.SetLoadForce("F0p0TR",
                            'Fv'+str(q+1),[0,0,-Fv[q]*S*W/(S*(F-1)*2),0,0,0])
                else:
                    ret = SapModel.PointObj.SetLoadForce('F'+str(n)+'p'+str(i)+'TR',
                            'Fv'+str(q+1),[0,0,-Fv[q]*S*W/(S*(F-1)),0,0,0])
# Creating load combos
    for i in range(len(Fh)):
        for n in range(len(Fv)):
            ret = SapModel.LoadCases.StaticNonlinear.SetCase(
                'H'+str(int(Fh[i]/1000))+'V'+str(Fv[n]))
            ret = SapModel.LoadCases.StaticNonlinear.SetLoads(
                'H'+str(int(Fh[i]/1000))+('V'+str(Fv[n])),2,["Load","Load"],
                ["Fv"+str(n+1),"Fh"+str(i+1)],[1,1])
# Setting only the loadcases we want to run, to 'run'
    ret = SapModel.File.Save(ModelPath)
    ret = SapModel.Analyze.SetActiveDOF([True, False, True, False, True, False])
    ret = SapModel.Analyze.SetRunCaseFlag('', False, True)
    Combinations = []
    for i in range(len(Fh)):
        for n in range(len(Fv)):
            ret = SapModel.Analyze.SetRunCaseFlag(
                'H'+str(int(Fh[i]/1000))+'V'+str(int(Fv[n])), True, False)
            Combinations.append('H'+str(int(Fh[i]/1000))+'V'+str(Fv[n]))

#%% Applying loads and properties parametrically

    Mod = 10**7

    ModShear =  [1, Mod, Mod, Mod, Mod, Mod,1]
    ModShear2 =  [Mod, 1, 1, 1, 1, 1, 1]
    ModBend = [Mod, 1, Mod, Mod, 1, Mod, 1]
    ModBend2 = [Mod, 1, 1, Mod, 1, Mod, 1]
    ModBend3 = [Mod, 1, Mod, Mod, Mod, Mod, 1]
    ModAB = [Mod, Mod, 1, Mod, Mod, Mod, 1]
    ModHD = [Mod, Mod, Mod, 1, Mod, Mod, 1]
    ModSTF = [Mod, Mod, Mod, Mod, 1, Mod, 1]
    ModC = [Mod, Mod, Mod, Mod, Mod, 1, 1]
    Cont = ['In-plane Shear','In-plane bending','Rigid body sliding',
            'Kinematic rocking','Sheathing to-framing','Bottom rail compression',
            'Total deflection']

    Displa = []
    Contributions = []
    Analytical = []
```

```python
for i in range(len(ModShear)):
    # Links
    ret = SapModel.PropLink.SetLinear('Nail',
                        [False, True, True, False, False, False],
                        [False, False, False, False, False, False],
                        [0, k_n*ModSTF[i], k_n*ModSTF[i], 0, 0, 0],
                        [0, 0, 0, 0, 0, 0], 10, 10, False, False)
    ret = SapModel.PropLink.SetGap('Compression-In',
                        [True,False,False,False,False,False],
                        [False,False,False,False,False,False],
                        [True,False,False,False,False,False],
                        [((370*h_is*(b_is+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],
                        [((370*h_is*(b_is+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],0,0)
    ret = SapModel.PropLink.SetGap('Compression-InPe',
                        [True,False,False,False,False,False],
                        [False,False,False,False,False,False],
                        [True,False,False,False,False,False],
                        [((370*h_ips*(b_ips+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],
                        [((370*h_ips*(b_ips+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],0,0)
    ret = SapModel.PropLink.SetGap('Compression-Pe',
                        [True,False,False,False,False,False],
                        [False,False,False,False,False,False],
                        [True,False,False,False,False,False],
                        [((370*h_ps*(b_ps+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],
                        [((370*h_ps*(b_ps+ b_cb/2))/b_cb)*ModC[i],0,0,0,0,0],
                        [0,0,0,0,0,0],0,0)
    ret = SapModel.PropLink.SetHook('HoldDown',
                        [True,False,False,False,False,False],
                        [False,False,False,False,False,False],
                        [True,False,False,False,False,False],
                        [k_HD*ModHD[i],0,0,0,0,0], [0,0,0,0,0,0],
                        [k_HD*ModHD[i],0,0,0,0,0],[0,0,0,0,0,0],0,0)
    ret = SapModel.PropLink.SetLinear('AngleBracket',
                        [False,True,False,False,False,False],
                        [False,False,False,False,False,False],
                        [0,k_AB*ModAB[i],0,0,0,0],[0,0,0,0,0,0],
                        60, 60, False, False)
    #Studs and sheet
    ret = SapModel.PropFrame.SetModifiers('Perimeter',
                            [ModBend[i],1,1,1,ModBend[i],ModBend[i],1,1])
    ret = SapModel.PropFrame.SetModifiers('Internal_perimeter',
                            [ModBend[i],1,1,1,ModBend[i],ModBend[i],1,1])
    ret = SapModel.PropFrame.SetModifiers('InnerStud',
                            [ModBend[i],1,1,1,ModBend[i],ModBend[i],1,1])
    ret = SapModel.PropFrame.SetModifiers('TopCord',
                            [ModBend[i],1,1,1,ModBend[i],ModBend[i],1,1])
    ret = SapModel.PropFrame.SetModifiers('BottomCord',
                            [ModBend[i],1,1,1,ModBend[i],ModBend[i],1,1])
    ret = SapModel.PropArea.SetModifiers('A', [ModBend3[i], ModBend3[i],
                                         ModShear[i],ModBend2[i],
                                         ModBend2[i], ModShear2[i],
                                         ModShear2[i], 1, 1, 1])

    ret = SapModel.Analyze.RunAnalysis()
    Dis = []
    Analytical.append(ANA[i])
```

```
        for n in Combinations:

            ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
            ret = SapModel.Results.Setup.SetCaseSelectedForOutput(n)

            NumberResults = 0
            Obj = []
            Elm = []
            Name = 'F'+str(S-1)+'p'+str(F-2)+'TR'
            ACase = []
            StepType = []
            StepNum = []
            U1 = []
            U2 = []
            U3 = []
            R1 = []
            R2 = []
            R3 = []
            ObjectElm = 0

            [NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1,
             R2, R3, ret] = SapModel.Results.JointDispl(Name, ObjectElm,
            NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1, R2, R3)

            Dis.append(U1[0]) #save results of the single cycle
        SapModel.SetModelIsLocked(False)
        Displa.append(Dis)
        Contributions.append(Cont[i])

    ret = mySapObject.ApplicationExit(False)
    SapModel = None
    mySapObject = None

    return  Displa, Analytical, Combinations, Contributions

#%% Call

Sheats = [1,2,3,4,5,6]
HoldDown = [12177]
Name = []
Call = []
for i in range(len(Sheats)):
    Name.append(str(Sheats[i])+'x1200')
    for n in range(len(HoldDown)):
        Call.append(eval(str(Sheats[i])+'_'+str(HoldDown[n])))
        Call[-1]= LTF(1200, 2400, Sheats[i], 3, 100, 1, 70, Sheats[i]*2,
                      HoldDown[n])
Combinations = Call[0][2]
FigName = Call[0][3]
colours = ['tab:blue','tab:orange','tab:green','tab:pink','tab:purple',
           'tab:cyan','tab:olive','tab:gray','tab:red','tab:brown']
```

```python
for n in range(len(FigName)):
    GR1 = []
    GR2 = []
    GR3 =[]
    plt.figure(figsize=[10,5])
    for t in range(len(Combinations)):
        GR1.append([])
        GR2.append([])
        for i in range(len(Sheats)):
            GR1[t].append(Call[i][0][n][t])
            GR2[t].append(Call[i][1][n][t])
        GR3.append('FEM '+Combinations[t])
        GR3.append('Analytical '+Combinations[t])
        plt.plot(Name,GR1[t],color = colours[t],ls = '-')
        plt.plot(Name,GR2[t],color = colours[t], ls = '--')
    plt.title(FigName[n])
    plt.ylabel('Deflection [mm]', size = 10)
    plt.xlabel('Wall length [mm]', size = 10)
    plt.legend(GR3)
    plt.savefig(FigName[n]+'.jpg')
    plt.show()

initial_data = {'Contributions': FigName}
xlWriter = pd.ExcelWriter('Total.xlsx')
df = pd.DataFrame(initial_data, columns = ['Contributions'])
for t in range(len(Combinations)):
    r = []
    s = []
    for i in range (len(Sheats)):
        FEM = []
        Analytical = []
        for n in range(len(FigName)):
            FEM.append(Call[i][0][n][t])
            Analytical.append(Call[i][1][n][t])
        r.append(FEM)
        s.append(Analytical)
        df['FEM_'+str(Sheats[i])+'x1200'] = r[i]
        df['Analytical_'+str(Sheats[i])+'x1200'] = s[i]
        df.to_excel(xlWriter, sheet_name = Combinations[t] , index=False)

xlWriter.close()

file = open('Call.txt', 'w')
file.write('Call=')
file.write('{}'.format(Call))

file.close()

file = open('Call100.txt', 'w')
file.write('Call=')
file.write('{}'.format(Call))

file.close()
```

Appendix C - CLT Shear wall

```
# -*- coding: utf-8 -*-
"""
Master Thesis
May 2022
Håkon Østraat Sævareid & Johan Bjørkedal

"""
import os
import sys
import comtypes.client
import numpy as np
import math as m
import matplotlib.pylab as plt
import pandas as pd

#%% Model

def E(layers):
# Calculation of moduli for a CLT panels
    E0 = 11*10**3
    E90 = 0.370*10**3
    G0 = 0.690*10**3
    E1 = 0
    E2 = 0
    E3 = 0

    for i in range(len(layers)):

        if i % 2 == 0:
            E1 = E1 + layers[i]*E0
            E2 = E2 + layers[i]*E90
        elif i % 2 != 0:
            E1 = E1 + layers[i]*E90
            E2 = E2 + layers[i]*E0
    E1 = E1 / sum(layers)
    E2 = E2 / sum(layers)
    E3 = E90
    G12 = G0*0.75 #E1 /20
    G23 = G12 /10
    G13 = G12 /10

    return E1 ,E2 ,E3 , G12 , G23 , G13, E0,

'''
W = Width Of Wall
H = Hight
S = Segments
L = Layers(as Array)
N = Number of shear connectors (segmented CLT)
s = Number Of AngleBrackets
'''
def CLT(W, H, S, L, N, s, X ):

#%% Standard Initsiation process for SAP2000

    Filename = 'test'
    APIPath = '*Path to main folder*'
```

```
    AttachToInstance = False

    SpecifyPath = False

    if not os.path.exists(APIPath):
            try:
                os.makedirs(APIPath)

            except OSError:
                pass

    ModelPath = APIPath + os.sep + Filename
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)

    if AttachToInstance:
        try:
                mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")

        except (OSError, comtypes.COMError):
            print("No running instance of the program found or failed.")
            sys.exit(-1)

    else:
        if SpecifyPath:
            try:
                mySapObject = helper.CreateObject(ProgramPath)

            except (OSError, comtypes.COMError):
                print("Cannot start a new instance of the program from "
                      + ProgramPath)
                sys.exit(-1)

        else:
            try:
                mySapObject = helper.CreateObjectProgID(
                    "CSI.SAP2000.API.SapObject")

            except (OSError, comtypes.COMError):
                print("Cannot start a new instance of the program.")
                sys.exit(-1)

        mySapObject.ApplicationStart()

    SapModel = mySapObject.SapModel
    SapModel.InitializeNewModel()
    ret = SapModel.File.NewBlank()
    ret = SapModel.SetPresentUnits(9)

#%% Input

    # Horizontal force
    Fh = [10]  #kN

    # Vertical force
    Fv = [0,5]      #N/mm

    # Angle backet stiffnes
    k_AB = 13046.44
```

```python
    # Hold-down stiffness
    k_HD = X

    # Shear Connector
    k_SH = 2387.63

#%% Analytical Calculations

    ANUs = []
    ANUB = []
    ANUA = []
    ANUR = []
    ANU = []
    for i in range(len(Fh)):
        for n in range(len(Fv)):

            # Lateral deflection due to in-plane shear
            Us = (Fh[i]*1000*H)/((W*S)*sum(L)*E(L)[3])
            #t=total thickness,G = Gyx,mean
            ANUs.append(Us)

            # Lateral deflection due to in-plane bending
            l1 = 0
            for t in range(len(L)):
                if t % 2 == 0:
                    l1 = l1+ L[t]
            UB = (Fh[i]*1000*H**3)/(3*(((E(L)[6]*l1*((S*W)**3))/12)))
            # E0 = Emean, tv 0 sum thickness vertical members
            ANUB.append(UB)

            # Inter-storey lateral displacement due to the rigid body sliding
            UA = Fh[i]*1000/(s*k_AB)
            ANUA.append(UA)

            # Lateral displacement due to kinematic rocking

            if S == 1:
                UR = max((((Fh[i]*1000*H)/(k_HD*(W*S-0.1*W*S)**2)) -
                    (((Fv[n]*S*W)*(W*S-0.1*W*S))/(2*(k_HD*(W*S-0.1*W*S)**2))))*H,0)
                ANUR.append(UR)

            else:
                Ni = (Fv[n]*(S*W))/(2*Fh[i]*1000*H)
                ak = (k_HD/(k_SH*N))
                bk = (1-(Ni)*((3*S)-2)/(S**2))/(1-(Ni)*(S-2)/(S**2))
                ck = (1-(Ni))/(1+(Ni)*(S-2))
                Kcp = ((k_HD+(S-1)*(k_SH*N))/(S**2))*((S*W)**2)
                Ksw = (((1/k_HD)+((S-1)/(k_SH*N)))**(-1))*((S*W)**2)
                # modified formulas:
                #URb = max((((Fh[i]*1000*H)/Kcp)-((Fv[n]*S*W*W)/(2*Kcp)))*H,0)
                #URc = max((((Fh[i]*1000*H)/Ksw)-((Fv[n]*S*W*W)/(2*Ksw)))*H,0)
                # Old formulas
                URb = max((((Fh[i]*1000*H)/Kcp)-((Fv[n]*S*W*W*S)/(2*Kcp)))*H,0)
                URc = max((((Fh[i]*1000*H)/Ksw)-((Fv[n]*S*W)/(2*k_HD*S*W)))*H,0)

                if ak >= bk:
                    UR = URb
                    ANUR.append(UR)
                    print(1)
```

```
            elif ak <= ck:
                UR = URc
                ANUR.append(UR)
                print(2)
            else:
                UR = (((ak-bk)*(URc-URb))/(ck-bk))+URb
                ANUR.append(UR)
                print(3)

        # Total Deflection
        U = Us + UB + UA + UR
        ANU.append(U)

    ANA = [ANUs,ANUB,ANUA,ANUR,ANU]

#%% Initial material props

    # Material
    ret = SapModel.PropMaterial.SetMaterial('CLT', 3)
    ret = SapModel.PropMaterial.SetWeightAndMass('CLT', 2, (420/10**9)*(1/9.81))
    ret = SapModel.PropMaterial.SetMPOrthotropic('CLT', [E(L)[0], E(L)[1], E(L)[2]]
                                , [0.3,0.3,0.3], [0 ,0 ,0],[E(L)[3], E(L)[4], E(L)[5]])
    # Links
    ret = SapModel.PropLink.SetLinear('Shear', [False,True,False,False,False,False],
         [False,False,False,False,False,False],[0,k_SH,0,0,0,0],[0,0,0,0,0,0],
          30, 30, False, False)
    ret = SapModel.PropLink.SetGap('Compression', [True,False,False,False,False,False]
         ,[False,False,False,False,False,False],[True,False,False,False,False,False],
          [100000000000,0,0,0,0,0], [0,0,0,0,0,0],[100000000000,0,0,0,0,0],
          [0,0,0,0,0,0],10,10)
    ret = SapModel.PropLink.SetHook('HoldDown', [True,False,False,False,False,False]
        ,[False,False,False,False,False,False],[True,False,False,False,False,False],
        [k_HD,0,0,0,0,0], [0,0,0,0,0,0],[k_HD,0,0,0,0,0],[0,0,0,0,0,0],10,10)
    ret = SapModel.PropLink.SetLinear('AngleBracket', [False,True,False,False,False,False]
        ,[False,False,False,False,False,False],[0,k_AB,0,0,0,0],[0,0,0,0,0,0],
         10, 10, False, False)
    ret = SapModel.PropLink.SetGap('B', [True,False,False,False,False,False],
         [False,False,False,False,False,False],[True,False,False,False,False,False],
         [100000000000,0,0,0,0,0],[0,0,0,0,0,0],[100000000000,0,0,0,0,0],
         [0,0,0,0,0,0],10,10)

    # Applying sections and material
    ret = SapModel.PropArea.SetShell_1('A', 5, True, 'CLT', 0, sum(L), sum(L), -1)
    # Load applying frame
    ret = SapModel.PropMaterial.SetMaterial('L', 3)
    ret = SapModel.PropMaterial.SetWeightAndMass("L", 2, (1/10**10))
    ret = SapModel.PropMaterial.SetMPOrthotropic("L", [10**7, 10**7, 10**7],
         [0.3,0.3,0.3], [0 ,0 ,0],[10**7, 10**7, 10**7])

    ret = SapModel.PropFrame.SetRectangle('Load', 'L', 50, sum(L))

#%% Walls

    if S == 1:
        ret = SapModel.AreaObj.AddByCoord(4, [0,W,W,0], [0,0,0,0], [0,0,H,H],
                                          'S', 'A', 'S', "Global")
        ret = SapModel.FrameObj.AddByCoord(0, 0, H, W, 0, H, 'FF0', 'Load', 'FF0')
        ret = SapModel.FrameObj.AddByCoord(0, 0, 0, W, 0, 0, 'FB0', 'Load', 'FB0')
```

```
else:
    for i in range(S):
        if i == 0:
            ret = SapModel.AreaObj.AddByCoord(4, [i*W,i*W+W-15,i*W+W-15,i*W],
                  [0,0,0,0],[0,0,H,H],'S_'+str(i),'A', 'S_'+str(i), "Global")
            ret = SapModel.FrameObj.AddByCoord(i*W, 0, H, i*W+W-15, 0, H,
                  'FF'+str(i),'Load','FF'+str(i))
            ret = SapModel.FrameObj.AddByCoord(i*W, 0, 0, i*W+W-15, 0, 0,
                  'FB'+str(i),'Load','FB'+str(i))
        elif i == S-1:
            ret = SapModel.AreaObj.AddByCoord(4, [i*W+15,i*W+W,i*W+W,i*W+15],
                  [0,0,0,0],[0,0,H,H],'S_'+str(i),'A', 'S_'+str(i), "Global")
            ret = SapModel.FrameObj.AddByCoord(i*W+15, 0, H, i*W+W, 0, H,
                  'FF'+str(i),'Load','FF'+str(i))
            ret = SapModel.FrameObj.AddByCoord(i*W+15, 0, 0, i*W+W, 0, 0,
                  'FB'+str(i),'Load','FB'+str(i))
        else:
            ret = SapModel.AreaObj.AddByCoord(4,[i*W+15,i*W+W-15,i*W+W-15,i*W+15],
                  [0,0,0,0],[0,0,H,H],'S_'+str(i),'A','S_'+str(i),"Global")
            ret = SapModel.FrameObj.AddByCoord(i*W+15, 0, H, i*W+W-15, 0, H,
                  'FF'+str(i),'Load','FF'+str(i))
            ret = SapModel.FrameObj.AddByCoord(i*W+15, 0, 0, i*W+W-15,0,0,
                  'FB'+str(i),'Load','FB'+str(i))

    for i in range(S):
        for n in range(N+2):
            if i != 0 and n !=0 and n != (N+1):
                ret = SapModel.LinkObj.AddByCoord(i*W-15, 0, n * H/(N+1),
                      i*W+15, 0, n * H/(N+1), 'SH_'+str(i)+'_'+str(n),
                      False, 'Shear','SH_'+str(i) +'_'+str(n))

                #Body constraint
                ret = SapModel.PointObj.AddCartesian(i*W-15, 0,n*H/((N+2)-1),
                      'PL_S'+str(i)+'_N'+str(n)+'l', 'PL_S'+str(i)+'_N'
                      +str(n)+'l')
                ret = SapModel.PointObj.AddCartesian(i*W+15, 0,n*H/((N+2)-1),
                      'PL_S'+str(i)+'_N'+str(n)+'r', 'PL_S'+str(i)
                      +'_N'+str(n)+'r')
                ret = SapModel.ConstraintDef.SetBody('CONS_S'+str(i)+'_N'
                      +str(n),[True,False,False,False,False,False])
                ret = SapModel.PointObj.SetConstraint('PL_S'+str(i)+'_N'
                      +str(n)+'r', 'CONS_S'+str(i)+'_N'+str(n))
                ret = SapModel.PointObj.SetConstraint('PL_S'+str(i)+'_N'
                      +str(n)+'l', 'CONS_S'+str(i)+'_N'+str(n))
            elif i !=0:
                ret = SapModel.PointObj.AddCartesian(i*W-15, 0, n * H/((N+2)-1),
                      'PL_S'+str(i)+'_N'+str(n)+'l','PL_S'+str(i)+'_N'+str(n)+'l')
                ret = SapModel.PointObj.AddCartesian(i*W+15, 0, n * H/((N+2)-1),
                      'PL_S'+str(i)+'_N'+str(n)+'r','PL_S'+str(i)+'_N'+str(n)+'r')
                ret = SapModel.ConstraintDef.SetBody('CONS_S'+str(i)+'_N'+str(n),
                      [True, False, False, False, False, False])
                ret = SapModel.PointObj.SetConstraint('PL_S'+str(i)+'_N'
                      +str(n)+'r', 'CONS_S'+str(i)+'_N'+str(n))
                ret = SapModel.PointObj.SetConstraint('PL_S'+str(i)+'_N'
                      +str(n)+'l', 'CONS_S'+str(i)+'_N'+str(n))
```

```python
#%% Supports

    a = (W*S)/(s+1)
    if S == 1:
        ret = SapModel.LinkObj.AddByCoord(0,0,0,0,0,-10,'LH_SL_',False,
              'HoldDown', 'LH_SL_')
        ret = SapModel.LinkObj.AddByCoord(0,0,0,0,0,-10,'LC_SL_',False,
              'Compression', 'LC_SL_')
        ret = SapModel.LinkObj.AddByCoord(0,0,0,0,0,-10,'LBL',False,'B','LBL_')
        ret = SapModel.PointObj.AddCartesian(0, 0, -10,'PC_SL_','PC_SL_')
        ret = SapModel.PointObj.SetRestraint('PC_SL_',[True,True,True,True,True,True])
        ret = SapModel.LinkObj.AddByCoord(W, 0, 0, W, 0, -10, 'LC_ST_', False,
                                        'Compression', 'LC_ST_')
        ret = SapModel.LinkObj.AddByCoord(W, 0, 0, W, 0, -10, 'LC_HT_', False,
                                        'HoldDown', 'LH_ST_')
        ret = SapModel.LinkObj.AddByCoord(W, 0, 0, W, 0, -10, 'LBT_', False,
                                        'B', 'LBT_')
        ret = SapModel.PointObj.AddCartesian(W, 0, -10,'PC_ST_','PC_ST_')
        ret = SapModel.PointObj.SetRestraint('PC_ST_',
                                           [True,True,True,True,True,True])


    else:
        for i in range(S):
            if i == 0:
                ret = SapModel.LinkObj.AddByCoord(0, 0, 0, 0, 0, -10,
                    'LH_SL_'+str(i), False, 'HoldDown', 'LH_SL_'+str(i))
                ret = SapModel.LinkObj.AddByCoord(0, 0, 0, 0, 0, -10,
                    'LC_SL_'+str(i), False, 'Compression', 'LC_SL_'+str(i))
                ret = SapModel.LinkObj.AddByCoord(0, 0, 0, 0, 0, -10,
                    'LBL_'+str(i), False, 'B', 'LBL_'+str(i))
                ret = SapModel.PointObj.AddCartesian(0, 0, -10,
                    'PC_SL_'+str(i),'PC_SL_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL_'+str(i),
                    [True,True,True,True,True,True])
                ret = SapModel.LinkObj.AddByCoord(W-15, 0, 0, W-15, 0, -10,
                    'LC_ST_'+str(i), False, 'Compression', 'LC_ST_'+str(i))
                ret = SapModel.LinkObj.AddByCoord(W-15, 0, 0, W-15, 0, -10,
                    'LBT_'+str(i), False, 'B', 'LBT_'+str(i))
                ret = SapModel.PointObj.AddCartesian(W-15, 0, -10,
                    'PC_ST_'+str(i),'PC_ST_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_'+str(i),
                    [True,True,True,True,True,True])
            elif i == S-1:
                ret = SapModel.LinkObj.AddByCoord(i*W+15, 0, 0, i*W+15, 0, -10,
                    'LC_SL_'+str(i), False, 'Compression', 'LC_SL_'+str(i))
                ret = SapModel.LinkObj.AddByCoord(i*W+15, 0, 0, i*W+15, 0, -10,
                    'LBL_'+str(i), False, 'B', 'LBL_'+str(i))
                ret = SapModel.PointObj.AddCartesian(i*W+15, 0, -10,
                    'PC_SL_'+str(i),'PC_SL_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL_'+str(i),
                    [True,True,True,True,True,True])
                ret = SapModel.LinkObj.AddByCoord((i+1)*W, 0, 0, (i+1)*W, 0, -10,
                    'LC_ST_'+str(i), False, 'Compression', 'LC_ST_'+str(i))
                ret = SapModel.LinkObj.AddByCoord((i+1)*W, 0, 0, (i+1)*W, 0, -10,
                    'LBT_'+str(i), False, 'B', 'LBT_'+str(i))
                ret = SapModel.LinkObj.AddByCoord((i+1)*W, 0, 0, (i+1)*W, 0, -10,
                    'LH_ST_'+str(i), False, 'HoldDown', 'LH_ST_'+str(i))
                ret = SapModel.PointObj.AddCartesian((i+1)*W, 0, -10,
                    'PC_ST_'+str(i),'PC_ST_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_'+str(i),
                    [True,True,True,True,True,True])
```

```python
            else:
                ret = SapModel.LinkObj.AddByCoord(i*W+15, 0, 0, i*W+15, 0, -10,
                    'LC_SL_'+str(i), False, 'Compression', 'LC_SL_'+str(i))
                ret = SapModel.LinkObj.AddByCoord(i*W+15, 0, 0, i*W+15, 0, -10,
                    'LBL_'+str(i), False, 'B', 'LBL_'+str(i))
                ret = SapModel.PointObj.AddCartesian(i*W+15, 0, -10,
                    'PC_SL_'+str(i),'PC_SL_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL_'+str(i),
                    [True,True,True,True,True,True])
                ret = SapModel.LinkObj.AddByCoord((i+1)*W-15, 0, 0, (i+1)*W-15,
                    0, -10, 'LC_ST_'+str(i), False, 'Compression', 'LC_ST_'+str(i))
                ret = SapModel.LinkObj.AddByCoord((i+1)*W-15, 0, 0, (i+1)*W-15,
                    0, -10, 'LBT_'+str(i), False, 'B', 'LBT_'+str(i))
                ret = SapModel.PointObj.AddCartesian((i+1)*W-15, 0, -10,
                    'PC_ST_'+str(i),'PC_ST_'+str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_'+str(i),
                    [True,True,True,True,True,True])

    for i in range(s):
        ret = SapModel.LinkObj.AddByCoord(a*(i+1), 0, 0, a*(i+1), 0, -10,
                    'LAB_S'+str(i), False, 'AngleBracket', 'LAB_S'+str(i))
        ret = SapModel.PointObj.AddCartesian(a*(i+1), 0, -10,
                    'PAB_S'+str(i),'PAB_S'+str(i))
        ret = SapModel.PointObj.SetRestraint('PAB_S'+str(i),
                    [True,True,True,True,True,True])

    for i in range (int((S*W)/200)):
        if i*200 % W != 0:
            ret = SapModel.LinkObj.AddByCoord(i*200, 0, 0, i*200, 0, -10,
                    'LC_S'+str(i), False, 'B', 'LC_S'+str(i))
            ret = SapModel.PointObj.AddCartesian(i*200, 0, -10,
                    'LC_S'+str(i),'LC_S'+str(i))
            ret = SapModel.PointObj.SetRestraint('LC_S'+str(i),
                    [True,True,True,True,True,True])
    ret = SapModel.View.RefreshView(0, False)

#%% Mesh
    ret = SapModel.GroupDef.SetGroup("GroupCLT")
    ret = SapModel.SelectObj.ClearSelection()
    ret = SapModel.SelectObj.PropertyMaterial('CLT')
    ret = SapModel.AreaObj.SetGroupAssign("", "GroupCLT", False, 2)
    ret = SapModel.SelectObj.ClearSelection()

    numobj=0
    obtype=[]
    shellnamesfirst=[]
    [numobj, obtype, shellnamesfirst, ret] = SapModel.GroupDef.GetAssignments(
        "GroupCLT", numobj, obtype, shellnamesfirst)

    for s in shellnamesfirst:
        areanames=[]
        numberareas=0
        SapModel.SelectObj.All()
        ret=SapModel.EditArea.Divide(str(s), 3, numberareas, areanames,
                                     PointOnEdgeFromPoint=True)

    ret = SapModel.AreaObj.SetAutoMesh("GroupCLT", 2, MaxSize1 = 50,
                                       MaxSize2 =50, ItemType = 1 )
```

```
#%% Applying load


# Creating load pattern for horizontal load and applying
    ret = SapModel.PointObj.AddCartesian(0, 0, H,'PF','PF')
    for i in range(len(Fh)):
        ret = SapModel.LoadPatterns.Add("Fh"+str(i+1), 3, 0, False)
        ret = SapModel.PointObj.SetLoadForce('PF','Fh'+str(i+1),
                                              [Fh[i]*1000,0,0,0,0,0])
# Creating load pattern for vetrical load
    for i in range(len(Fv)):
        ret = SapModel.LoadPatterns.Add("Fv"+str(i+1), 3, 0, False)
        for n in range(S):
            ret = SapModel.FrameObj.SetLoadDistributed('FF'+str(n), "Fv"+str(i+1),
                                                        1, 10, 0, 1, Fv[i], Fv[i],
                                                        "Global", True, False)

# Creating load combos
    for i in range(len(Fh)):
        for n in range(len(Fv)):
            ret = SapModel.LoadCases.StaticNonlinear.SetCase(
                'H'+str(Fh[i])+'V'+str(Fv[n]))
            ret = SapModel.LoadCases.StaticNonlinear.SetLoads(
                'H'+str(Fh[i])+'V'+str(Fv[n]),2,["Load","Load"],
                ['Fh'+str(i+1),'Fv'+str(n+1)],[1,1])
# Setting only the loadcases we want to run, to 'run'
    ret = SapModel.File.Save(ModelPath)
    ret = SapModel.Analyze.SetActiveDOF([True, False, True, False, True, False])
    ret = SapModel.Analyze.SetRunCaseFlag('', False, True)
    Combinations = []
    for i in range(len(Fh)):
        for n in range(len(Fv)):
            ret = SapModel.Analyze.SetRunCaseFlag('H'+str(Fh[i])+'V'+str(Fv[n]),
                                                  True, False)
            Combinations.append('H'+str(Fh[i])+'V'+str(Fv[n]))

#%% Applying loads and properties parametrically

    Mod = 10**7

    ModShear =  [1, Mod, Mod, Mod, 1]
    Modshear3 =[ True, True, True, False, False]
    ModBend3 =[False, True, False, False, False]
    ModBend = [Mod, 1, Mod, Mod, 1]
    ModAB = [Mod, Mod, 1, Mod, 1]
    ModHD = [Mod, Mod, Mod, 1, 1]
    Cont = ['In-plane Shear','In-plane bending','Rigid body sliding',
            'Kinematic rocking','Total deflection']

    Displa = []
    Contributions = []
    Analytical = []
    ret = SapModel.PointObj.AddCartesian(S*W, 0, H,'PA','PA')
    for i in range(len(ModShear)):
    #for i in range(4,5):
        # Links
        ret = SapModel.PropLink.SetLinear('Shear',
                                          [False,True,False,False,False,False],
                                          [False,False,False,False,False,False],
                                          [0,ModHD[i]*k_SH,0,0,0,0],[0,0,0,0,0,0],
                                           20, 20, False, False)
```

```
    ret = SapModel.PropLink.SetGap('B',
                                   [True,ModBend3[i],False,False,False,False],
                                   [ModBend3[i],ModBend3[i],False,False,False,False],
                                   [True,False,False,False,False,False],
                                   [100000000000,0,0,0,0,0], [0,0,0,0,0,0],
                                   [100000000000,0,0,0,0,0],[0,0,0,0,0,0],10,10)

    ret = SapModel.PropLink.SetHook('HoldDown',
                                   [True,False,False,False,False,False],
                                   [False,False,False,False,False,False],
                                   [True,False,False,False,False,False],
                                   [ModHD[i]*k_HD,0,0,0,0,0], [0,0,0,0,0,0],
                                   [ModHD[i]*k_HD,0,0,0,0,0],[0,0,0,0,0,0],0,0)
    ret = SapModel.PropLink.SetLinear('AngleBracket',
                                   [False,True,False,False,False,False],
                                   [False,False,False,False,False,False],
                                   [0,ModAB[i]*k_AB,0,0,0,0],[0,0,0,0,0,0],
                                    10, 10, False, False)
    ret = SapModel.PropArea.SetModifiers('A', [ModBend[i], ModBend[i],
                                    ModShear[i],1, 1, 1, 1, 1, 1])

    for l in range(S):
        for n in range(N):
            if l != 0:
                ret = SapModel.ConstraintDef.SetBody('CONS_S'+str(l)+'_N'+str(n),
                      [True, False, Modshear3[i], False, False, False])

    ret = SapModel.Analyze.RunAnalysis()
    Analytical.append(ANA[i])
    Dis = []

    for n in Combinations:
        ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
        ret = SapModel.Results.Setup.SetCaseSelectedForOutput(n)

        NumberResults = 0
        Obj = []
        Elm = []
        Name = 'PA'
        ACase = []
        StepType = []
        StepNum = []
        U1 = []
        U2 = []
        U3 = []
        R1 = []
        R2 = []
        R3 = []
        ObjectElm = 0

        [NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3,
         R1, R2, R3, ret] = SapModel.Results.JointDispl(Name, ObjectElm,
         NumberResults, Obj, Elm, ACase, StepType, StepNum, U1, U2, U3, R1, R2, R3)

        Dis.append(U1[0]) #save results of the single cycle
    SapModel.SetModelIsLocked(False)
    Displa.append(Dis)
    Contributions.append(Cont[i])

ret = mySapObject.ApplicationExit(False)
SapModel = None
```

```python
    mySapObject = None

    return  Displa, Analytical, Combinations, Contributions

#%% Call
CLT100s3 = [30,40,30]
CLT120s3 = [40,40,40]
CLT100s5 = [20,20,20,20,20]
CLT120s5 = [30,20,20,20,30]
CLT140s5 = [40,20,20,20,40]
CLT160s5 = [40,20,40,20,40]
CLT180s5 = [40,30,40,30,40]
CLT200s5 = [40,40,40,40,40]

#Wall = CLT(1200,2400,2,CLT180s5, 10, 4)
Panels = [2,3,4, 5, 6]
X = [CLT100s3]
x = [12177] #,17395,30442]

Name = []
Call = []
for i in range(len(Panels)):
    Name.append(str(Panels[i])+'x1200')
    for n in range(len(X)):
        Call.append(eval(str(i+1)+'_'+str(n+1)))
        Call[-1] = CLT(1200, 2400, Panels[i], X[n], 8 , Panels[i]*2, x[0])
Combinations = Call[0][2]
FigName = Call[0][3]

colours = ['tab:blue','tab:orange','tab:green','tab:red','tab:purple',
           'tab:cyan','tab:olive','tab:gray','tab:pink','tab:brown']

for n in range(len(FigName)):
    GR1 = []
    GR2 = []
    GR3 = []
    plt.figure(figsize=[10,5])
    for t in range(len(Combinations)):
        GR1.append([])
        GR2.append([])
        for i in range(len(Panels)):
            GR1[t].append(Call[i][0][n][t])
            GR2[t].append(Call[i][1][n][t])
        GR3.append('FEM '+Combinations[t])
        GR3.append('Analytical '+Combinations[t])
        plt.plot(Name,GR1[t],color = colours[t],ls = '-')
        plt.plot(Name,GR2[t],color = colours[t], ls = '--')
    plt.title(FigName[n], loc = 'Left')
    plt.title('CLT100', loc = 'Right')
    plt.ylabel('Deflection [mm]', size = 10)
    plt.xlabel('Wall length [mm]', size = 10)
    plt.legend(GR3)
    plt.savefig(FigName[n]+'CLT100.jpg')
    plt.show()
```

```
plt.figure(figsize=[10,5])
for t in range(len(Combinations)):
    GR1 = []
    GR2 = []
    for i in range(len(Panels)):
        GR = 0
        GR2.append(Call[i][1][4][t])
        for n in range(len(FigName)-1):
            GR = GR + (Call[i][0][n][t])
        GR1.append(GR)
    plt.plot(Name,GR1,color = colours[t],ls = '-')
    plt.plot(Name,GR2,color = colours[t], ls = '--')
    GR3.append('FEM '+Combinations[t])
    GR3.append('Analytical '+Combinations[t])
plt.title('Total deflection (Sum contributions)', loc = 'Left')
plt.title('CLT100', loc = 'Right')
plt.ylabel('Deflection [mm]', size = 10)
plt.xlabel('Wall length [mm]', size = 10)
plt.legend(GR3)
plt.savefig('Total deflection (Sum contributions)'+'CLT100.jpg')
plt.show()

initial_data = {'Contributions': FigName}
xlWriter = pd.ExcelWriter('TotalCLT100.xlsx')
df = pd.DataFrame(initial_data, columns = ['Contributions'])
for t in range(len(Combinations)):
    r = []
    s = []
    for i in range (len(Panels)):
        FEM = []
        Analytical = []
        for n in range(len(FigName)):
            FEM.append(Call[i][0][n][t])
            Analytical.append(Call[i][1][n][t])
        r.append(FEM)
        s.append(Analytical)
        df['FEM_'+str(Panels[i])+'x1200'] = r[i]
        df['Analytical_'+str(Panels[i])+'x1200'] = s[i]
        df.to_excel(xlWriter, sheet_name = Combinations[t] , index=False)

xlWriter.close()


file = open('Call100.txt', 'w')
file.write('Call=')
file.write('{}'.format(Call))

file.close()
```

Appendix D - Modal Updating

```python
# -*- coding: utf-8 -*-
"""
Master Thesis
May 2022
Håkon Østraat Sævareid & Johan Bjørkedal
"""

import sys
import os
import time
import comtypes.client

import numpy as np
import pandas as pd
import scipy.optimize as opt

# Directory information
_Main_Folder = r"*Insert Main Folder Path*"
_Save_Path = _Main_Folder + os.sep + 'temp' + os.sep + 'SIMULATION'
_ModelPath = _Main_Folder + os.sep + r'Model.sdb'

os.chdir(_Main_Folder)

#%% Functions

def MaC(Fi1,Fi2):

    # Formula to calculate MAC
    MAC = np.abs(Fi1.T @ Fi2)**2 / \
        ((Fi1.T @ Fi1)*(Fi2.T @ Fi2))

    return MAC

def Mod(par, SAPObj, SapModel, path):

    # Unlocking the SAP model
    SapModel.SetModelIsLocked(False)

    # Setting units to kN and meters
    _unit = 6
    ret = SapModel.SetPresentUnits(_unit)

    # Reading excel result file ==============================================

    # This method is spesific for the dataset

    xl_filepath = r'*Insert Path To desired Excel Source File*'

    CH = ['CH1','CH1','CH1','CH2','CH2','CH2',
          'CH3','CH3','CH3','CH4','CH4','CH4','CH5','CH5','CH5']
    DOF = ['U1','U2','U3','U1','U2','U3','U1','U2','U3',
            'U1','U2','U3','U1','U2','U3']

    arrays = [CH,DOF]
    tuples = list(zip(*arrays))
    index = pd.MultiIndex.from_tuples(tuples, names=["Channels", "Dof"])
```

```python
ModeNames = ['Mode 1','Mode 2','Mode 3']

df = pd.read_excel(xl_filepath,skiprows=1,names=ModeNames,
                   usecols=[2,3,4],sheet_name='Val')

M1 = list(df['Mode 1'])
M2 = list(df['Mode 2'])
M3 = list(df['Mode 3'])

Def_Exp = pd.DataFrame({1:M1,2:M2,3:M3},index=index)
Def_Exp = Def_Exp.sort_index()

df = pd.read_excel(xl_filepath,names=ModeNames,
                   usecols=[2,3,4],nrows=1,sheet_name='Val')
Freq_Exp = [float(df['Mode 1']),float(df['Mode 2']),float(df['Mode 3'])]

# Numerical analysis and results =========================================

# The parameters from updating is applied
SapModel.AreaObj.SetLoadUniform("Concrete floor","SuperDeadArea",
                                par[0] ,10 , True , "GLOBAL" ,1)

SapModel.AreaObj.SetLoadUniform('Internal Floors',"SuperDeadArea",
                                par[0] ,10 , True , "GLOBAL" ,1)

SapModel.AreaObj.SetLoadUniform('Roof',"SuperDeadArea",
                                par[2] ,10 , True , "GLOBAL" ,1)

SapModel.FrameObj.SetLoadDistributed('Line_fasade_roof', "SuperDeadLine",
                                     1, 10, 0, 1, par[1]/2, par[1]/2,
                                     "GLOBAL", True, True, 1)

SapModel.FrameObj.SetLoadDistributed('Line_fasade_internal',
                                     "SuperDeadLine", 1, 10, 0, 1, par[1],
                                     par[1],"GLOBAL", True, True, 1)

# Setting solver options
_Ss = 0 # Standard solver
_As = 1 # Advanced solver
_Mts = 2 # Multi-threaded solver
#------------------------
_Auto = 0 # Auto (program determined)
_GUI = 1 # GUI process
_SP = 2 # Separate process
#------------------------
_F32 = False # Force 32bit solver (True/False)

SapModel.Analyze.SetSolverOption_1(_Mts, _Auto, _F32, "MODAL")

# Number of modes
N_Modes = 15
SapModel.LoadCases.ModalEigen.SetNumberModes("MODAL", N_Modes, 2)

# Modal analysis settings
_ESF = 0        # Eigen shift
_ECO = 0        # Eigen cut off
_ETol = 1E-09   # EigenTol
_AAFS = 1       # AllowAutoFreqShift (0=no, 1=yes)
```

```python
SapModel.LoadCases.ModalEigen.SetParameters("MODAL", _ESF, _ECO,
                                            _ETol, _AAFS)
ret = SapModel.Analyze.SetRunCaseFlag("MODAL", True)

# Run analysis
SapModel.Analyze.RunAnalysis()

# Selecting the modal results
SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")

# Exstract frequencies

NumberResults = 0
Obj = []
Elm = []
ACase = []
StepType = []
StepNum = []
Period = []
Freq = []
CircFreq = []
EigenValue = []

Ana1 = SapModel.Results.ModalPeriod(NumberResults, ACase, StepType,
                                    StepNum, Period, Freq, CircFreq,
                                    EigenValue)
StepNum = Ana1[3]
Freq = Ana1[5]

F = pd.DataFrame({'Freq':list(Freq)},index=[int(i) for i in list(StepNum)])
F = F.sort_index()

# Displacements for all chanels

NR = 0
Obj = []
Elm = []
Name = 'Channels'
ACase = []
StepType = []
StepNum = []
U1 = []
U2 = []
U3 = []
R1 = []
R2 = []
R3 = []
ObjectElm = 2


Ana2 = SapModel.Results.JointDispl(Name, ObjectElm, NumberResults,
                                   Obj, Elm, ACase, StepType, StepNum,
                                   U1, U2, U3, R1, R2, R3)
Elm = Ana2[2]
StepNum = Ana2[5]
U1 = Ana2[6]
U2 = Ana2[7]
U3 = Ana2[8]
```

```python
# Fetching and organizing the numerical data ===============================
arrays = [list(Elm),[int(i) for i in list(StepNum)]]
tuples = list(zip(*arrays))
I = pd.MultiIndex.from_tuples(tuples, names=["Channels", "Modes"])
Def_Num = pd.DataFrame({'U1':list(U1),'U2':list(U2),'U3':list(U3)},index=I)
Def_Num = Def_Num.sort_index()

# Copy the data
Def_Num_Norm = Def_Num.stack().unstack(level=1)

# Normalizing the data
for i in StepNum:
    if Def_Num_Norm[i].abs().max() > Def_Num_Norm[i].max():
        Def_Num_Norm[i] = Def_Num_Norm[i] /Def_Num_Norm[i].min()
    else:
        Def_Num_Norm[i] = Def_Num_Norm[i] /Def_Num_Norm[i].max()

C_Def_Num = []
C_Freq_Num = []
C_Mode = []
for i in range(len(Freq_Exp)):  # Loop trough the experimental modes
    Err = []                    # Initialize/reset error list
    _freq_exp = Freq_Exp[i]     # Experimental frequency
    _def_exp = Def_Exp[i+1]     # Experimental mode shape

    for j in range(len(list(Def_Num_Norm.loc[('CH1','U1')]))): # Numerical
        _freq_num = F.loc[j+1] # Numerical frequenzy for the mode
        _def_num = Def_Num_Norm[int(j+1)] # Numerical modeshape
        _Macche = MaC(_def_num, _def_exp) # MAC
        E = ((_freq_exp - _freq_num)/_freq_exp)**2 + np.real(1-_Macche)
        Err.append(E)           # Append the error
    correct = np.min(Err)       # The "correct" mode shape
    c_index = np.argmin(Err)    # Index of the "correct" mode shape
    # Collecting results from the "correct" mode:
    C_Freq_Num.append(float(F.loc[c_index+1]))
    C_Def_Num.append(Def_Num_Norm[int(c_index+1)])
    C_Mode.append(c_index)

diagMacORD = []
for i in range(len(Freq_Exp)): # Loop trough the experimental modes
    _freq_exp = Freq_Exp[i] # Experimental frequency
    _def_exp = Def_Exp[i+1] # Experimental mode shape

    _freq_num = C_Freq_Num  # Numerical frequency
    _def_num = C_Def_Num[i] # Numerical modeshape

    diagMacORD.append(np.real(MaC(_def_exp,_def_num))) # MAC

# Creating numpy arrays (Frequency and MAC)
Freq_NUM = np.array(C_Freq_Num)
Freq_EXP = np.array(Freq_Exp)
diagMacORD = np.array(diagMacORD)

# Calculating the error
E1 = np.sum(1*((abs(Freq_EXP - Freq_NUM))/Freq_EXP))
E2 = 1*(np.sum((1-diagMacORD)))
err = E1 + E2

return err
```

```python
#%% Standard Initsiation process for SAP2000

Filename = r'Model'
APIPath = r"*Insert API Folder Path*"

AttachToInstance = True

SpecifyPath = False

if not os.path.exists(APIPath):
        try:
            os.makedirs(APIPath)

        except OSError:
            pass

ModelPath = APIPath + os.sep + Filename
ProgramPath = r"C:\Program Files\Computers and Structures\SAP2000 23\SAP2000"

helper = comtypes.client.CreateObject('SAP2000v1.Helper')
helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)

if AttachToInstance:
    try:
            mySapObject = helper.GetObject("CSI.SAP2000.API.SapObject")

    except (OSError, comtypes.COMError):
        print("No running instance of the program found or failed to attach.")
        sys.exit(-1)
else:
    if SpecifyPath:
        try:
            #'create an instance of the SAPObject from the specified path
            mySapObject = helper.CreateObject(ProgramPath)

        except (OSError, comtypes.COMError):
            print("Cannot start a new instance of the program from "
                  +ProgramPath)

            sys.exit(-1)

    else:
        try:
            mySapObject= helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")

        except (OSError, comtypes.COMError):
            print("Cannot start a new instance of the program.")
            sys.exit(-1)

    mySapObject.ApplicationStart()

SapModel = mySapObject.SapModel

# Setting units to kN and Meters
_unit = 6
ret = SapModel.SetPresentUnits(_unit)
```

```python
#%% Main analysis

# Bounds
# Load applied on internal floors                        [kN/m**2]
# Load applied on the perimeter to simulate facades  [kN/m]
# Load applied on roof                                   [kN/m**2]

bounds = [(0., 2.),
          (0., 4.),
          (0., 2.)]

# Differential Evolution algorithm

_strategies=['best1bin','best1exp','rand1exp','randtobest1exp',
             'currenttobest1exp','best2exp','rand2exp','randtobest1bin',
             'currenttobest1bin','best2bin','rand2bin','rand1bin']

_start = time.time()

Updating = opt.differential_evolution(func=Mod, bounds=bounds,
                            args=(mySapObject, SapModel, _ModelPath),
                            strategy='best1exp',
                            maxiter=30,   # maxfev=(maxiter+1)*popsize*len(x)
                            popsize=60,   # The population has popsize*len(x)
                            mutation=0.8, # The mutation constant
                            recombination=0.9, # The recombination constant
                            disp=True, # Prints the evaluated func at every it.
                            polish=True, # Can improve minimization slightly
                            init='latinhypercube')

_end = time.time()
print(_end - _start)
```

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway