Norwegian University
of Life Sciences

Master's Thesis 2021 30 ECTS
Faculty of Science and Technology

Finite Element Model Updating of a Multi-Storey CLT-Building and Analysis of Modal Performance Indicators

Carl-Ulrik Dahle Gurholt
Jonas Næss Mikalsen
Structural Engineering and Architecture
Faculty of Science and Technology (REALTEK)

Acknowledgements

This thesis concludes five years of education on the Master's degree in Structural Engineering and Architecture at the Norwegian University of Life Science (NMBU). We would like to show our deepest gratitude towards our main supervisor, Ebenezer Ussher, for the continued support and guidance throughout the semester. Furthermore we would like to thank our additional supervisor, Angelo Aloisio, for the brilliant and inspiring ideas. We would also like to thank Roberto Tomasi for the opportunity to widen our knowledge in structural engineering of timber structures. Lastly, thanks to Dag Pasquale Pasca for help with the experimental data.

Ås, May 2021

Carl-Ulrik Dahle Gurholt
Jonas Næss Mikalsen

Summary

The dynamic behaviour of multi-storey CLT-buildings due to operational excitation is an open issue; few in-situ test have been carried out to estimate their modal parameters. Moreover, there is a gap in knowledge about the use of ambient vibration measurements for model-updating purposes of timber buildings, and the performance indicators of timber buildings due to operational excitation is not yet thoroughly studied. In this thesis, the results of Operational Modal Analysis is interpreted and analysed in light of multiple Finite Element models in order to better understand the dynamic behaviour of CLT-buildings.

The analytical procedure is divided into three parts. A Finite Element model is constructed based on the geometry and estimates of parameter values of the Palisaden building. A large set of parameters is chosen for an optimization algorithm to minimize the difference between analytical and experimental modal data. The true parameter values are then extracted from a Finite Element Model Updating scheme based on sensitivity analysis. In the second part of analysis, the Finite Element model is adjusted to involve the numerical behaviour of connector elements to the optimization algorithms. The role of connector elements to the low-amplitude dynamics of the building is analysed and discussed based on sensitivity analysis. The third part of analysis consists of a parametric study to investigate the effects of eccentricity between center of mass and center of rigidity. A set of new Finite Element models are constructed to highlight the modal responses of changes in plan geometry of the building.

The initial Finite Element model do not correctly represent the modal behaviour of the building, with modal analysis providing fundamental frequencies about 30 % higher than the experimental values. Sensitivity analysis and model updating highlights the impact of all parameter values to the dynamic behaviour of the building, and an updated model is shown to have a high accuracy. Similar analysis on the connector elements prove that the connectors do in fact have very little impact on the low-amplitude dynamics of the building. In the third case, several interesting modal phenomena arise due to differences in eccentricity and plan shape geometry.

Table of Contents

List of Figures

List of Tables

Definitions

Abbreviations

CLT	Cross Laminated Timber
CoM	Center of Mass
CoR	Center of Rigidity
DOF	Degree of Freedom
FE	Finite Element
FEM	Finite Element Method
FEMU	Finite Element Model Updating
FSDD	Frequency Spatial Domain Decomposition
MAC	Modal Assurance Criterion
MDOF	Multi Degree of Freedom System
OAPI	Open Application Programming Interface
OMA	Operational Modal Analysis
SDOF	Single Degree of Freedom System
SHM	Structural Health Monitoring
SSI	Stochastic Subspace Identification method
ULS	Ultimate Limit State

Symbols

Ω	Diagonal matrix of eigenvalues
∂P	Change in parameter value
∂R	Change in response value
Φ	Matrix of eigenvectors
ϕ_n	Mode-shape vector
q_n	Modal coordinate
$[S]$	Sensitivity matrix

$[S]^+$	The pseudo-inverse sensitivity matrix
$[S_{norm}]$	Normalized sensitivity matrix
$\ddot{\mathbf{u}}$	Acceleration vector
$\dot{\mathbf{u}}$	Velocity vector
ν	Poisson's ratio
ϕ_a	Analytical Mode-shape vector
ϕ_e	Experimental Mode-shape vector
ρ	Self weight
$\mathbf{c}$	Damping-matrix
$\mathbf{k}$	Stiffness-matrix
$\mathbf{m}$	Mass-matrix
$\mathbf{u}$	Displacement vector
$\{P\}$	Parameter vector
$\{R\}$	Response vector
f_a	Analytical Eigenfrequency
f_e	Experimental Eigenfrequency
f_{dev}	Eigenfrequency deviation
ω_n	Natural circular frequency of vibration
f_n	Natural cyclic frequency of vibration
T_n	Natural period of vibration
C	Cost function
E	Modulus of Elasticity
E_0	Modulus of Elasticity parallel to the grain for a wood board
E_{90}	Modulus of Elasticity perpendicular to the grain for a wood board
E1	Modulus of Elasticity for CLT-panel for the strong in plane axis
E2	Modulus of Elasticity for CLT-panel for the weak in plane axis
E3	Modulus of Elasticity for CLT-panel for the out of plane
G	Shear modulus
G_0	Shear modulus parallel to the grain for a wood board
G_{90}	Shear modulus perpendicular to the grain for a wood board
G1	Shear modulus for CLT-panel about the 1-2 plane
G2	Shear modulus for CLT-panel about the 2-3 plane
G3	Shear modulus for CLT-panel about the 1-3 plane

RZ	Mode shape: torsion about the z-direction
UD	Mode shape: translation in diagonal direction
UX	Mode shape: translation in x-direction
UY	Mode shape: translation in y-direction

1. Introduction

1.1 Background

The use of timber as a construction material has rapidly grown over the past few decades. This is largely due to the commercial launch of the innovative laminar timber product, Cross Laminated Timber (CLT). This product has the capacity to bear loads in- and out-of-plane, making it suitable for full-size wall- and floor elements (Brandner et al., 2016).

There are numerous advantages of timber as a construction material. The raw materials are renewable and sustainable. As stated by Hill and Zimmer, 2018; encouragement of the cross laminated timber industry in Norway is essential to maintain the carbon absorbing properties of the forests. By taking into account the sequestration of forests, engineered wood products outperform more traditional building materials in terms of global warming potential. As the building sector is responsible for a considerable amount of the primary energy demand and energy related CO_2-emissions of industrialized countries, increased use of CLT as construction material can be viewed as a panacea to the global warming menace (Hill and Zimmer, 2018).

Historically, the use of timber buildings has been challenged in cities for its combustibility, although the massive wood structure provides fairly good behaviour in case of fire, and also good thermal insulation (Ceccotti, 2008). As the wooden surface of the CLT-panels are left visible in final use, aesthetics are also a benefit of CLT. The low gravitational weights of timber make the product applicable to construction in seismic prone areas, and may also reduce costs related to foundation and overall building assembly. This development may lead to the emergence of innovation regarding new engineered wood products.

However, as a relatively new building material, there are little data concerning the performance of CLT-structures, in comparison to other building materials. There is also yet no European standard related to the design of CLT-elements.

1.2 State of the Art

Vibrations and their effects on structures and construction works may present hazards or operating limitations, i.e. discomfort, malfunctioning, breakdown or failure in the construction. As a result, practising engineers rely on accurate mathematical models to describe the vibration characteristics of buildings or structures, and apply these models for design purposes to negate the consequences of the vibrations (Arora, 2011). Finite Element Model Updating, particularly based on Operational Modal Analysis identification techniques, has been considered an accurate and appropriate method to evaluate the performance of mathematical models, and to calibrate parameter values applied in Finite Element-models. This method may be performed both on structures under high (e.g. seismic) or low (e.g. operational) excitation. As the nature of the method is non-destructive to construction works, it has been applied to a multitude of historic or culturally valuable buildings, such as the four bell-towers analysed in the study by Standoli et al., 2021, or the historic minaret tower analysed by Alpaslan et al., 2020. The method has also been applied to a multitude of other types of construction works, such as bridges (e.g. Tuhta et al., 2020), or dam-reservoir-foundation systems (e.g. Bayraktar et al., 2011).

Ambient vibrations result in low levels of building motion, and may be caused by wind, traffic or human activities. Studies have been conducted to extract the dynamic characteristics of a wide array of civil engineering structures based on ambient vibration measurements (Mugabo et al., 2019). However, only a few studies have been performed on timber buildings (Reynolds et al., 2016, Worth et al., 2012, Aloisio et al., 2020). As stated by Aloisio et al., 2020; the understanding of the dynamic behaviour of timber buildings under operational conditions is still an open issue, and the gap in knowledge is magnified in the possibly linear response observed from CLT-structures excited by operational conditions to the nonlinear behaviour under seismic action. Some of the investigations previously mentioned are briefly reviewed below.

In the study by Worth et al., 2012, a three-storey building with an implemented post-tensioned Laminated Veneer Lumber shear wall system, ambient vibration measurements were continuously monitored throughout three stages of construction. Events during the construction process were evaluated, such as addition of non-structural elements, or addition of concrete floor topping. Addition of non-structural elements significantly contributed to the global stiffness of the system. The addition of concrete floor topping as a structural diaphragm significantly increased the stiffness of mode 1, but not for mode 2.

Reynolds et al., 2016 compare the dynamic properties of two structural systems used for multi-storey timber buildings; one is sheated stud-and-rail timber construction, while the other is a Cross Laminated Timber system. Both buildings have a reinforced concrete core, and ambient vibration measurements were taken. The authors discuss that the similarities in the dynamic responses are striking between the two buildings. A drawback in hybrid construction is also highlighted; as the lower stiffness parts of the structure may not fully contribute the the global stiffness of the building.

In the study by Aloisio et al., 2020, the eight-story Palisaden building located in Ås was subject to ambient vibration measurements, with the objective to perform dynamic identification and model updating. The results of dynamic identification were interpreted in light of a simplified shear type numerical model, and the minimization of a modal-based objective function gave an estimate of the unknown parameters, the storey masses. The authors found that, in light of the model updating procedure, the connections do not significantly contribute to the low amplitude dynamics of the structure, and the behaviour of the structure may be described as continuum-like.

When a building is asymmetric in plan or elevation, the responses from large excitation may be complex. These irregularities and asymmetries may be due to architectural or functional constraints. Irregularities in plan frequently arise when structural elements contributing to horizontal stiffness, like elevator cores and shear walls, are concentrated on one side of the building (Alecci and De Stefano, 2019), or in a more general sense: if the global center of mass and the global center of rigidity do not coincide. Multiple studies have been conducted to investigate the dynamic effects of irregularities in plan, and some of these investigations are briefly reviewed below.

Raheem et al., 2018 evaluates the effects of plan irregularity on seismic response demands of a variety of constructed L-shaped multi-storey reinforced concrete buildings. Multiple building models are generated through gradual reduction in plan of a reference square-shaped model. A free-vibration and response-spectrum analysis is conducted, and calculations on fundamental frequencies, story displacements, inter-story drift ratios e.g. are evaluated. The authors conclude that the building models with high irregularities are more vulnerable, both due to lateral torsional coupling behaviour and high stress concentrations.

Gokdemir et al., 2013 constructed reinforced concrete building models of a variety of different plans, such as L-shape, rectangular shape or square shape. Within each building model, additional sub-models were constructed, which have differences in distance between center of mass and center of rigidity. Analyses of the structures were made with particular interest in torsion and shear responses of structural members due to seismic forces. Calculations show a nearly linear relationship between torsional responses and eccentricity between center of mass and center of rigidity.

1.3 Problem Statement and Research Questions

The following problem statement is constructed in light of the literature review described above:

- There is a gap in knowledge concerning the use of ambient vibration measurements for model updating purposes of timber buildings.
- The performance indicators of timber buildings due to operational excitation are not thoroughly investigated.

This thesis will attempt the evaluation of the following research questions:

1. How reliable are Finite Element models to replicate the real modal behaviour of timber buildings?
2. What are the modal performance indicators of timber buildings?
3. What is the influence of connectors on the modal parameters?
4. How do construction features/variables affect timber buildings' modal performance?

1.4 Research Objectives and Scientific Contributions

In this thesis, the ambient vibration measurements of the experimental campaign of Aloisio et al., 2020, are used to perform a Finite Element Model Updating of a selection of parameters applied in a numerical Finite Element model constructed in the commercially available software SAP2000. This analysis is considered the initial part of this thesis. The objectives of this research are to attempt the evaluation of modal performance in timber buildings, in particular to identify the performance indicators in the building and to evaluate the importance of each performance indicator. Analysis will also be made in regard to the reliability of an Finite Element model in light of model updating procedures. Furthermore, in the second part of this thesis, the modelling of connectors, such as angle brackets or hold-down brackets, is evaluated; the findings of Aloisio et al., 2020 are reflected upon with basis in a more advanced FE-model, and the use of optimization algorithms to indicate stiffness values of said connectors is addressed. Lastly, in the third part of the thesis, by application of updated parameter values as indicated by the initial model updating, the low amplitude characteristics of multiple other structures are discussed, and the effect of eccentricities between global center of mass and center of rigidity is evaluated and discussed based on a set of plan geometries constructed using the Open Application Programming Interface with SAP2000.

As the model updating procedure is well documented in scientific literature, the parameter data acquired from the analysis is to be seen as the main scientific contribution of the initial part of this thesis. Further, in the second part of the thesis, the results moreover reflect as an extension of the previous research conducted by Aloisio et al., 2020. The major scientific contribution of this thesis is considered the analysis made in the third part, by numerically examining the effects of irregularities in plan geometries, and evaluating the effects in a parametric study.

The Finite Element software used in this thesis is selected based on its user-friendly layout and ease of usage for practising engineers. Although, this choice limits some possibilities in analysis, in particular in terms of the non-linear behaviour of structures, which is therefore not performed in this thesis. As the empirical data used in this thesis is limited to fundamental frequencies and mode shape vectors, no analysis is made on behalf of moisture content of the timber elements. Financial aspects of analyses are not considered as the thesis' focus is structural engineering and structural dynamics.

2. Theory

In this chapter, the theoretical framework considering timber as a construction material is submitted with a specific focus on the behaviour of CLT. Furthermore, the background of mathematical modal analysis is presented. Next, the Finite Element Method is described, with a specific focus on SAP2000 Finite Element Method software. Lastly, the mathematical theory behind Finite Element Model Updating and sensitivity analysis is presented.

2.1 Timber as a Construction Material

Timber as a construction material differs from steel and reinforced concrete for a number of reasons. It is a natural, biological, and hygroscopic material with varying mechanical properties. Timber is considered as an orthotropic material (Serano et al., 2015), with its unique and independent mechanical properties in three mutually perpendicular axes; longitudinal (L), radial (R), and tangential (T). The longitudinal axis is parallel to the grain, the radial axis is normal to the growth rings (perpendicular to the grain) and the tangential axis is perpendicular to the grain and tangential to the growth rings (Ross, 2010). The difference between the mechanical properties parallel to the grain vs. the mechanical properties perpendicular to the grain are substantial, with the longitudinal axis much stronger in e.g. compression than the perpendicular axis (Serano et al., 2015).

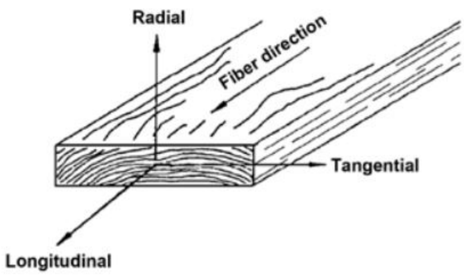


Figure 2.1: Axis definition for a wood board (Ross, 2010).

Timber has an elastic region and a plastic region in its strain-stress diagram. The elastic region is considered linear; therefore Hooke's law applies. Stresses lower than the elastic limit produce recoverable deformations after the loadings are removed. To accurately describe the elastic properties of timber, nine independent constants are needed (Ross, 2010); Modulus of Elasticity (E) in the longitudinal-, radial- and tangential- direction; Modulus of Rigidity (G) in the LR, LT and RT plane; Poisson's ratio (ν) in the LR, LT and RT plane. However, during design, the properties of the radial- and tangential axis are often simplified to be equivalent, and only two axes are used; parallel with the grain (0), and perpendicular to the grain (90). In this case, the stiffness properties of the material are described by four independent constants; Two moduli of elasticity, one parallel to the grain (E_0) and one perpendicular to the grain (E_{90}). Two shear moduli, one parallel to the grain (G_0) and one perpendicular to the grain (G_{90}). Loading higher than the elastic limit causes plastic deformation or failure (Ross, 2010). From the APA – The Engineered Wood Association, 2012, some simple relationships between the modulus of elasticity and the shear modulus for lumber are given by equations 2.1 - 2.3:

$$E_{90} = E_0/30 \tag{2.1}$$

$$G_0 = E_0/16 \tag{2.2}$$

$$G_{90} = G_0/10 \tag{2.3}$$

European Committee For Standardization, 2004b gives some extra conditions for wood-based construction material versus steel and concrete. The moisture content, load duration and the "size effect" affect the strength and stiffness significantly. European Committee For Standardization, 2004b introduces amending factors for these effects. Greater moisture content entails larger creep deformations, introduced via the climate classes. Longer load deformations reduce the capacity, via the load duration classes. And the bigger cross section also reduces the capacity.

2.1.1 Cross Laminated Timber

Cross-Laminated Timber (CLT) is a wood panel product consisting of layers of solid-sawn timber glued together. The panels most commonly consist of an odd number of alternating board layers, with an angle 90^o between the grain direction of the board layers. This configuration provides sufficient strength in both of the in-plane directions, and achieves higher structural rigidity in all directions compared to non-engineered wood products. The board layers are glued together in the entire surface area, and sometimes the individual boards are glued together inside the layer (Wallner-Novak et al., 2014). In Central Europe there is a generally accepted standard for CLT-layer thickness of 20, 30 and 40 mm (Brandner et al., 2016), but sizes from 6 to 45 mm are also produced

(Wallner-Novak et al., 2014). The board width varies from 40 to 300mm (Wallner-Novak et al., 2014), with a proposed criteria of board width bigger or equal than 4 times the board thickness to prevent a reduction in rolling shear resistance (Brandner et al., 2016). Most commonly, finger joint profiles are used in the boards.

CLT mainly has a strong (0^o) and a weak (90^o) in-plane direction. The stiffest and strongest axis often correspond with the axis of the top layer (parallel to grain direction of this layer). When designing CLT elements, the E_{90} may be neglected because of the high ratio $E_0/E_{90} \approx 30$ (Brandner et al., 2016). The minor normal stress is neglected and this means that the transverse layer only is subjected to shear. Shear failure normally occurs tangential to the annual ring, and is called rolling-shear failure. The rolling-shear strength is between a third to a half of the shear strength running parallel to the fiber (Wallner-Novak et al., 2014).

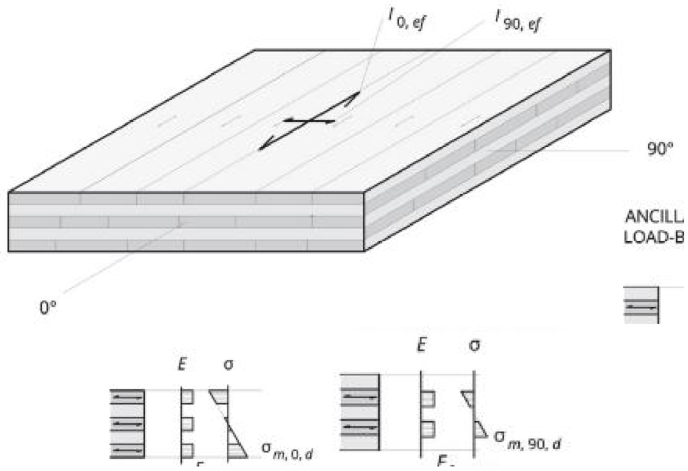


Figure 2.2: Load-bearing capacity for a 5-layer CLT element (Wallner-Novak et al., 2014).

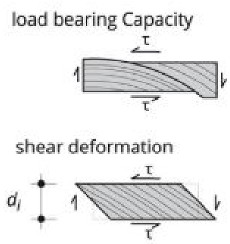


Figure 2.3: Shear behavior of the transverse layer (Wallner-Novak et al., 2014).

Wallner-Novak et al., 2014 introduces a method for calculation of the stiffness values for CLT, based on the effective, or net cross sectional values. This approach reduces the moment of inertia and the net area of the sections based on insignificant stiffness provided from boards in the weak direction. This approach is somewhat inconvenient to implement in the Finite-Element Method (FEM) software employed in this thesis, SAP2000 (CSI america, år). Aranha, 2016 presents an alternative in his studies by adjusting the elastic modulus and the shear modulus for the cross section in the different directions, which is more convenient to implement in SAP2000. Equations 2.4-2.9 provide details of Aranha's formulations for calculation of stiffness values for CLT.

$$\mathrm{E}_{x,3} = \frac{\mathrm{E}_0 t_1 + \mathrm{E}_{90} t_2 + \mathrm{E}_0 t_3}{t} \tag{2.4}$$

$$\mathrm{E}_{y,3} = \frac{\mathrm{E}_{90} t_1 + \mathrm{E}_0 t_2 + \mathrm{E}_{90} t_3}{t} \tag{2.5}$$

$$\mathrm{G}_{xy} = \frac{\mathrm{G}_0}{1 + 6a_T(t_{mean}/a)^2} \tag{2.6}$$

$$\mathrm{a}_T = 0.32(t_{mean}/a)^{-0.77} \tag{2.7}$$

For five layer CLT-panels, eq. 2.4 and eq. 2.5 are extended to:

$$\mathrm{E}_{x,5} = \frac{\mathrm{E}_0 t_1 + \mathrm{E}_{90} t_2 + \mathrm{E}_0 t_3 + \mathrm{E}_{90} t_4 + \mathrm{E}_0 t_5}{t} \tag{2.8}$$

$$\mathrm{E}_{y,5} = \frac{\mathrm{E}_{90} t_1 + \mathrm{E}_0 t_2 + \mathrm{E}_{90} t_3 + \mathrm{E}_0 t_4 + \mathrm{E}_{90} t_5}{t} \tag{2.9}$$

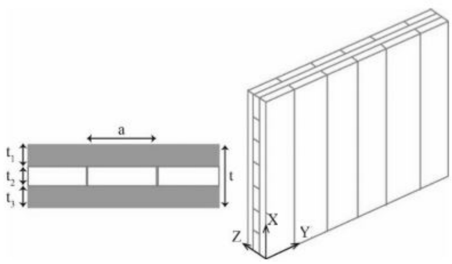


Figure 2.4: Orientation of the CLT wall panel (Aranha, 2016).

Based on its relatively high strength and stiffness in both in-plane directions, CLT-panels may be employed as shear-walls in a building or structure. A shear-wall is a wall element designed to transfer horizontal loads, e.g. loads from wind and earthquake. The horizontal load-bearing capacity is enforced in the connections and fasteners employed in the panel-to-panel or panel-to-slab connections. Typically, these connections are hold-down- and angle brackets. The hold-down brackets are usually designed to transfer axial loads and thereby prevent a rocking deformation of the panels, while the angle-brackets are employed to transfer shear forces and prevent a sliding movement of the panels.

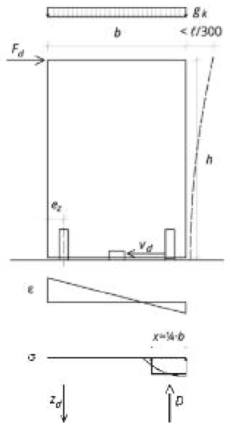


Figure 2.5: Typical fasteners for a CLT-shear-wall (Wallner-Novak et al., 2014).

Flatscher et al., 2015 describes four different types of deflections for a horizontally loaded CLT-panel: bending, shear, slipping and rocking, as illustrated by figure 2.6.

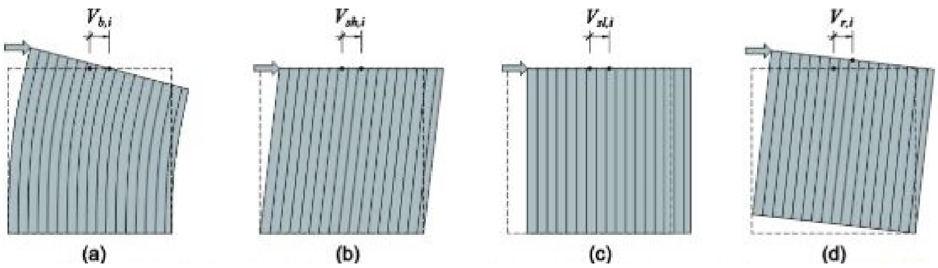


Figure 2.6: CLT wall deflection components (a) bending, (b) shear, (c) slip and (d) rocking (Flatscher et al., 2015).

As wood is considered a brittle material, the ductility in a CLT wall panel will mainly arise from the deformation of the connections, and the panel remains elastic without obtaining any damage (Aranha, 2016). A study by Gavric et al., 2015 confirms this statement. They reported that the forces and deformations mainly occurred in the connections, as the in-plane deformations of the CLT panels were negligible. A study conducted by Wallner-Novak et al., 2014 reported the deformation of connectors is normally dominating in comparison to CLT panels. As reported by Aranha, 2016, the main deformation for shorter panels is rocking, while for longer panels, the main deformation is shear.

2.2 Modal Analysis

All structural systems vibrate when subjected to a dynamic load or operational conditions such as traffic or wind, or when disturbed from its equilibrium position. Mathematical models have been formulated to describe the behaviour of a body under a disturbance. Equation 2.10 formulates the behaviour of a Single Degree of Freedom (SDOF) system.

$$\mathrm{m\ddot{u} + c\dot{u} + ku = p}(t) \tag{2.10}$$

A SDOF-system is a system where the mass and elastic properties are assumed to be concentrated in a single physical element, and the movement of the element can be described by a single coordinate. An example of a typical SDOF-system is a water-tower. Figure 2.7 illustrates a lumped mass SDOF-system, where m is the mass, c is the damping-coefficient, k is the stiffness-coefficient, p(t) is excitation force, and u is the displacement, and therefore u̇ is the velocity and ü is the acceleration (Chopra, 2007).

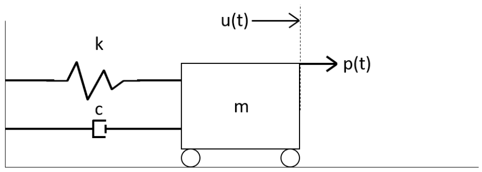


Figure 2.7: Illustration of a SDOF-system with lumped mass, stiffness and damping.

Altering the SDOF equation to a a Multi Degree of Freedom (MDOF) system produces equation 2.11,

$$\mathbf{m\ddot{u} + c\dot{u} + ku = p}(t) \tag{2.11}$$

where **m** is the mass-matrix, **c** is the damping-matrix, **k** is the stiffness-matrix, and **p**(t) is excitation force vector, and **u** is the displacement vector, and therefore $\dot{\mathbf{u}}$ is the velocity vector and $\ddot{\mathbf{u}}$ is the acceleration vector (Chopra, 2007).

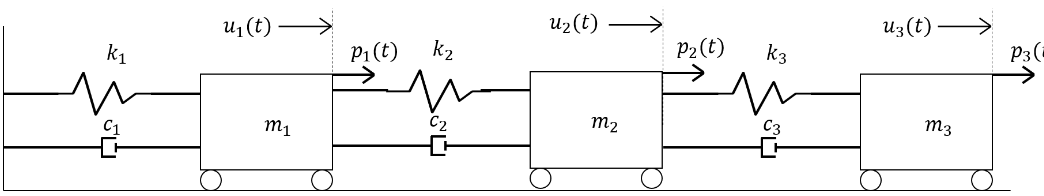


Figure 2.8: Illustration of a MDOF-system with lumped mass, stiffness and damping.

For an undamped system undergoing free vibration, the equation of motion alters, the damping terms and the force are taken out.

$$\mathbf{m\ddot{u} + ku} = 0 \tag{2.12}$$

For an undamped system, some characteristic deflection shapes exist, the natural mode of vibration or sort mode shapes. Subjecting this type of structure to an initial deflection equal to the deflection shapes, the structure would undergo simple harmonic motion (Chopra, 2007). The relationship between the natural period of vibration, T_n, the natural circular frequency of vibration, ω_n, and the natural cyclic frequency of vibration is shown by equation 2.13:

$$T_n = \frac{2\pi}{\omega_n} \qquad f_n = \frac{1}{T_n} \tag{2.13}$$

For an undamped free vibration structure, the displacement vector can be expanded using the natural modes (Chopra, 2007). This expansion is called the modal expansion of displacements, as shown in equation 2.14,

$$\mathbf{u}(t) = \phi_n q_n(t) \tag{2.14}$$

Where q_n are the modal coordinates, also called normal coordinates that is a scalar multiplier, and ϕ_r is a natural mode shape.

$$q_n(t) = A_n * \cos\omega_n t + B_n \sin\omega_n t \tag{2.15}$$

Where A_n and B_n are constants. Combining eq. 2.14 and eq. 2.15 gives

$$\mathbf{u}(t) = \phi_n(A_n * \cos\omega_n t + B_n \sin\omega_n t) \tag{2.16}$$

Substituting eq. 2.16 in eq. 2.12 leads to

$$[-\omega_n^2 \mathbf{m}\phi_n + \mathbf{k}\phi_n]q_n(t) = 0 \tag{2.17}$$

This equation can either be solved through the non-trivial solution $q(t) = 0 \implies \mathbf{u}(t) = 0$ or that the modal frequencies and mode shapes must satisfy the following equation, called the matrix eigenvalue problem.

$$\mathbf{k}\phi_n = \omega_n^2 m\phi_n \implies [\mathbf{k} - \omega_n^2\mathbf{m}]\phi_n = 0 \tag{2.18}$$

On matrix form

$$[\mathbf{k} - \Omega^2\mathbf{m}]\Phi = 0 \tag{2.19}$$

where Ω is the diagonal matrix of eigenvalues, and Φ is the matrix of corresponding eigenvectors. The new equation can be solved to either with a new trivial solution $\phi_r = 0$, which also implies no motion (Chopra, 2007), or:

$$\det[\mathbf{k} - \omega_n^2\mathbf{m}] = 0 \tag{2.20}$$

The mode shape is an important phenomenon. It describes the deformation the system is undergoing when vibrating at the natural frequency. In a system similar to 2.8 with only one allowable direction of freedom, the mode shape is quite simple. For a system in 2 or 3 allowable directions of freedom, the mode shape becomes more complex, especially considering that it opens up for rotations called torsion. In this thesis four different mode shape types are defined; the translation shape in the x direction (UX), the translation shape in the y direction (UY), the translation shape in a diagonal (UX + UY) direction (UD) and the torsion shape rotation around the z direction (RZ) from fig 2.9.

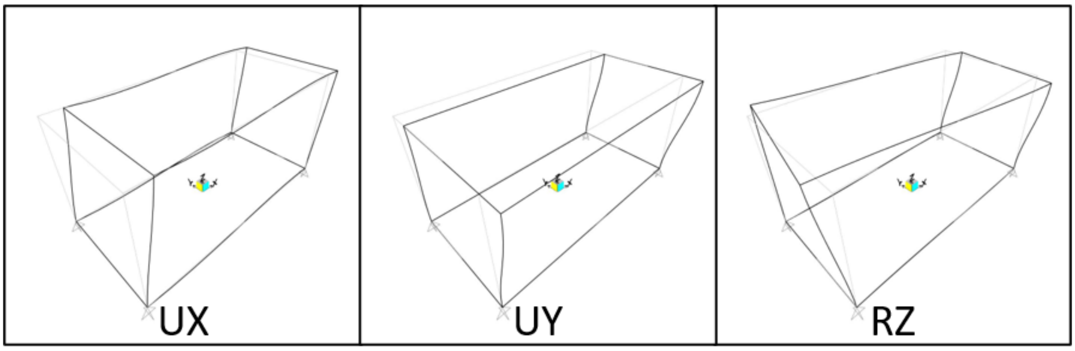


Figure 2.9: The different mode shape types.

A torsional mode shape may cause structural problems. Torsional problems occur when the location of center of mass end center of rigidity differs. By increasing the difference, the structure is subjected to greater torsional moments (Gokdemir et al., 2013). Heavy damage, called "knife cut", will occur to walls and columns undergoing excessive torsion. An earthquake load acts at the center of mass, but the resisting force acts in the center of rigidity, which can lead to torsion (Gokdemir et al., 2013). A torsional mode is therefore undesirable as the fundamental mode of the system, and measures should be made to shift the torsional mode to a higher frequency.

For earthquake analysis, either for performance based design (PBD) or force-based design (FBD), most design codes and guidelines require participation of modes contributing to about 90 % of total mass participation. Consequently, modal analysis on the structure is performed. It has been observed that the greater the number of modes participating, the higher the likely amplification of the response. European Committee For Standardization, 2004c recommends avoiding the fundamental mode being torsional, in particular for seismic analysis. European Committee For Standardization, 2004c chapter: 4.3.3.3.1 defines the significant modes for this global response from the effective modal mass as:

- the sum of the effective modal masses for the mass taken into account amounts to at least 90% of the total mass of the structure

- All modes with effective modal masses greater than 5% of the total mass are taken into account.

Participating mass ratios correspond to the effective modal mass. Effective modal mass is an important measurement of the impact of the mode. This factor is important in order to calculate the total base shear response V_b for the structure. Modal mass M_n^* and modal height h_n^* for the n-th mode is given in eq. 2.21 and 2.22 from Chopra, 2007

$$M_n^* = \frac{(L_n^h)^2}{M_n} \tag{2.21}$$

$$h_n^* = \frac{L_n^\theta}{L_n^h} \tag{2.22}$$

M_n^* and h_N^* are independent of how the mode is normalized, unlike the M_n and L_n^h. For a simpler introduction, a MDOF system modeled with lumped masses is chosen to describe M_n, L_n^h and L_n^θ in eq. 2.23, eq. 2.24 and eq. 2.25 respectively.

$$M_n = \sum_{j=1}^{N} m_j \phi_{jn}^2 \tag{2.23}$$

$$L_n^h = \sum_{j=1}^{N} m_j \phi_{jn} \tag{2.24}$$

$$L_n^\theta = \sum_{j=1}^{N} h_j m_j \phi_{jn} \tag{2.25}$$

where ϕ_{jn} is the mode shape value at nth mode, m_j is the lumped mass at the jth floor and h_j is the height of the jth floor above the base.

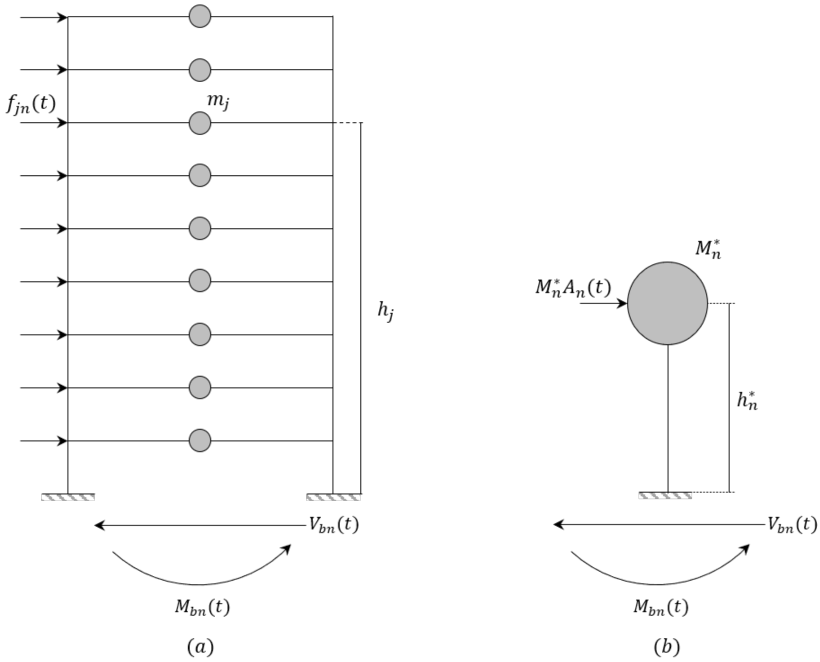


Figure 2.10: Illustration of a MDOF system (a), with SDOF system representing the effective modal mass and effective modal height (b) (inspiration from Chopra, 2007).

The procedure becomes more advanced when introducing an unsymmetrical floor plan. Chopra, 2007 describes a method for a rectangular floor plan. A brief description of the procedure in SAP2000 is given in section 2.3.1.

In a three dimensional space, each mode has mass-participation in each of the different directions. If a mode is purely translational in a certain direction, it will display 100 % of its mass participation in that direction. To identify the characteristics of the shape of the mode, the modal direction factor is used. The modal direction factor is a measure of the percentage of modal participation in a certain direction in space, i.e. U_x, U_y or R_z, and may be calculated by formula 2.26:

$$U_x = \frac{U_x}{U_x + U_y + R_z + R_x + R_y} * 100\% \tag{2.26}$$

2.3 Finite Element Method

The finite element method (FEM) is a versatile method that can be applied to a wide range of problems. The method is one of the most important developments in applied mechanics (Chopra, 2007). The concept is based on dividing real structures with infinite degrees of freedom, to a model with finite degrees of freedom. A finite element model consist of elements connected through nodal points. The models DOFs are located in the nodes, and some interpolation is chosen for the elements (Chopra, 2007). The FEM has some benefits (Chopra, 2007): (1) Simple interpolation functions can be chosen for each finite element, (2) Improved accuracy by increasing the number of finite elements, (4) narrowly banded mass and stiffness matrices that reduce the computational effort, (5) The nodal displacements are directly given from the generalized displacements.

Chopra, 2007 gives an analysis procedure in 5 steps:

1. Idealize the structure as a finite element model, with nodes and with elements connecting them. Define the DOF at the nodes.

2. Define the stiffness matrix $\mathbf{k}_e$, the mass matrix $\mathbf{m}_e$, the geometric stiffness matrix $\mathbf{k}_{Ge}$ and the (applied) force vector $\mathbf{p}_e(t)$ for each element. The force-displacement relation, the inertia force-acceleration relation, and force-displacement relation associated with gravity loads for each element are given by:

$$(\mathbf{f}_S)_e = \mathbf{k}_e\mathbf{u}_e \quad (\mathbf{f}_I)_e = \mathbf{m}_e\ddot{\mathbf{u}}_e \quad (\mathbf{f}_G)_e = \mathbf{k}_{Ge}\mathbf{u}_e \tag{2.27}$$

3. The forming of the transformation matrix $\mathbf{a}_e$ is necessary to relate the displacements $\mathbf{u}_e$ and force $\mathbf{p}_e$ for the elements, to the displacements $\mathbf{u}$ and force $\mathbf{p}$ for the assemblage:

$$\mathbf{u}_e = \mathbf{a}_e\mathbf{u} \quad \mathbf{p}(t) = \mathbf{a}_e^T\mathbf{p}_e(t) \tag{2.28}$$

4. Determine the stiffness, mass and geometric stiffness matrices and the force vector via the assembly of the element matrices, for the assemblage of the finite elements:

$$\mathbf{k} = A_{e=1}^{Ne}\mathbf{k}_e \quad \mathbf{m} = A_{e=1}^{Ne}\mathbf{m}_e \quad \mathbf{k}_G = A_{e=1}^{Ne}\mathbf{k}_{Ge} \quad \mathbf{p}(t) = A_{e=1}^{Ne}\mathbf{p}_e(t) \tag{2.29}$$

   A denotes the direct assembly procedure according to the matrix $\mathbf{a}_e$, the element stiffness matrix, elements mass matrix and the element force vector. N_e is the number of elements into the assemblage.

5. The equations of motion for the assemblage:

$$\mathbf{m}\ddot{\mathbf{u}} + \mathbf{c}\dot{\mathbf{u}} + \mathbf{k}\mathbf{u} + \mathbf{k}_G\mathbf{u} = \mathbf{p}(t) \tag{2.30}$$

   where $\mathbf{c}$ is the damping matrix.

2.3.1 SAP2000

SAP2000 is a simpler structural analysis software with a user friendly layout (Aranha, 2016). The software is based on the finite element model for both linear and non-linear model analysis. An analysis consists of two phases that interact with each other: 1) node-elements discretization model and 2) finite element model discretization (Rivera, 2015). SAP2000 has 5 different types of objects for different use: Point, Line, Area and Solid. For this study, the area-objects, the point objects and line objects are used.

Joints is one of the two point-objects, also known as nodes. They have six degrees of freedom, three translational and three rotational (Computers & Structures, Inc., 2017). If the displacement of a joint is known then this is represented as a restraint. To enforce certain types of rigid-body behavior, to connect together different parts of the model, and to impose certain types of symmetry constraints can be added to a set of two or mote joints (Computers & Structures, Inc., 2017).

The link element can be a point object or a line object, either the one joint to ground support or the two joints connector. Link element can exhibit up to three different types of behavior: linear, non-linear, and frequency-dependent. The frequency-dependent property is optional, and a linear/non-linear property must be assigned. The non-linear behavior can be modeled in a variety of ways (Computers & Structures, Inc., 2017):

- Viscoelastic damping
- Gap (compression only) and hook (tension only)
- Multi-linear uniaxial elasticity
- Uniaxial plasticity (Wen model)
- Multi-linear uniaxial plasticity (kinematic, Takeda, and pivot)
- Biaxial-plasticity base isolator
- Friction-pendulum base isolator

The link element also has six degrees of freedom (axial, shear, torsion, and pure bending), represented as six separate "springs". Any number or all of the six degrees for freedom can be fixed, i.e., that their deformation is zero. Similar to a restraint for the one joint elements, and constraint for a two joints element (Computers & Structures, Inc., 2017).

Area object, known as shell-element, is a three- or four-node element that combines membrane and plate-bending behavior. Plate-bending included two-way, out-of-plane, plate rotational stiffness components and a translational stiffness component (Computers & Structures, Inc., 2017). In SAP2000, there are two types of shell-elements to select from: thin-plate (Kirchhoff) or thick-plate (Mindlin/Reissner). The difference is that the thin-plate neglects transverse shearing deformation, and the thick-plate includes this effect. Floor- and wall-systems can be modeled with shell-elements(Computers &

Structures, Inc., 2017). Area objects are assigned user-defined material properties.

SAP2000 provides the option for automatic meshing of objects. This creates additional joints corresponding to the assigned elements (Computers & Structures, Inc., 2017). The automatic mesh-tool is limited compared to other FEM softwares with sophisticated algorithms to mesh critical regions (Rivera, 2015). The meshing in SAP2000 can therefore be described as "coarse" compared to other FEM softwares. It is required to choose an appropriate mesh size, because it directly affects the accuracy of the analysis result for the FE-model (Rivera, 2015).

Modal analysis is used to evaluate the dynamic behaviour of a structure in terms of vibration modes, the frequencies of the structure and the participating mass ratio. Modal analysis can be done with either eigenvector analysis or Ritz-vector analysis and are always linear. Eigenvector analysis determines the undamped free-vibration mode shapes and frequencies of the structure, involving eq. 2.19. Ritz-vector analysis finds modes that are excited by a particular loading (Computers & Structures, Inc., 2017).

Participating mass ratios in SAP2000 are calculated in the 6 DOF, for each mode. Participating mass ratios in translation are calculated by eq. 2.31-2.33 and participating mass ratios in rotation are calculated by eq. 2.34-2.36 (Computers & Structures, Inc., 2017).

$$r_{xn} = \frac{(\phi_n^T m_x)^2}{M_x} \tag{2.31}$$

$$r_{yn} = \frac{(\phi_n^T m_y)^2}{M_y} \tag{2.32}$$

$$r_{zn} = \frac{(\phi_n^T m_z)^2}{M_z} \tag{2.33}$$

where ϕ_n is the mode shape, M_x, M_y, M_z are the total unrestrained masses acting in the global x, y and z, and m_x, m_y, m_z are the unit rotational acceleration loads. The program generates three unit acceleration loads acting on the structure following d'Alembert's principle (Computers & Structures, Inc., 2017).

$$r_{rxn} = \frac{(\phi_n^T m_{rx})^2}{M_{rx}} \tag{2.34}$$

$$r_{ryn} = \frac{(\phi_n^T m_{ry})^2}{M_{ry}} \tag{2.35}$$

$$r_{rzn} = \frac{(\phi_n^T m_{rz})^2}{M_{rz}} \tag{2.36}$$

where M_{rx}, M_{ry} and M_{rz} are the total rotational inertias of the unrestrained masses action about the global axes x, y and z, and m_{rx}, m_{ry} and m_{rz} are the unit rotational acceleration acting on the structure following d'Alembert's principle (Computers & Structures, Inc., 2017)

2.4 Finite Element Model Updating

Finite Element Model Updating (FEMU) is a method for updating or calibrating a numerical Finite Element Model using data acquired from Operational Modal Analysis (OMA). OMA is a method which aims to identify the modal properties of a structure using vibration response measurements, e.g. by placing accelerometers in points of interest in the structure and analysing the responses. Interest for the method emerged in the 1990s, and has since been of importance to civil engineering structures among other things (Mottershead and Friswell, 1995). The basic approach is to update some selected structural parameters (such as Young's modulus, gravitational density, spring stiffness or boundary conditions), to obtain similarity between the numerical modal analysis and the OMA (Mordini et al., 2007). Improved performance and better understanding of structures will reduce the energy usage and the material usage, which is an important topic in the modern world (Mottershead and Friswell, 1995).

In this study, a building is monitored using a number of accelerometers placed in different positions of different floors. The FEMU is based on an optimization of a cost function, which expresses the difference between experimental and numerical results, through multiple iterations of modal analysis (Mordini et al., 2007).

The choice of parameters to be included in FEMU is of great importance to the reliability of the updated model. In most cases, the choice of parameters are based on the initial uncertainty corresponding to each parameter. For example, the geometry of the building (e.g. floor span, dimensions of beams/columns) is often known with a high certainty, while the loading applied to the building during measurements could be considered more uncertain. The maximum and minimum values of the parameters may be updated to, are also of significance, and requires engineering judgement; the parameters should not be updated to values considered unreasonable or unrealistic.

2.4.1 Sensitivity Analysis

The initial step of FEMU is to perform a sensitivity analysis. In a sensitivity analysis, the aim is to determine the sensitivity of each parameter, i.e. the change between measured and numerical data (∂R) related to a change in parameter value (∂P) (Mordini et al., 2007). Equation 2.37 gives the formula for the sensitivity matrix [S]:

$$[S] = \frac{\partial R}{\partial P} \tag{2.37}$$

The sensitivity matrix is calculated for M parameters, regarding N different responses; therefore the S matrix is N by M in dimensions.

$$[S] = \begin{bmatrix} \frac{\partial R_1}{\partial P_1} & \frac{\partial R_1}{\partial P_2} & \cdots & \frac{\partial R_1}{\partial P_M} \\ \frac{\partial R_2}{\partial P_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial R_N}{\partial P_1} & \cdots & \cdots & \frac{\partial R_N}{\partial P_M} \end{bmatrix} \tag{2.38}$$

To easily compare the effects of the different types of parameters, the sensitivity matrix may be normalized, both for the responses and the parameter values (Brownjohn et al., 2001), as shown in equation 2.39

$$[S_{norm}]_{ij} = [R_i]^{-1} \left[\frac{\partial R_i}{\partial P_j}\right] [P_j] \tag{2.39}$$

The sensitivity matrix may be computed analytically or numerically. In the analytical method, direct derivation is used, and the systems stiffness and mass matrices are required for the solution. The numerical perturbation technique only requires the results from multiple FE analyses (Mordini et al., 2007). Mordini et al., 2007 provides a formula for the sensitivity matrix for the perturbation method in his studies, as shown in equation 2.40 and 2.41.

$$S_{ij} = \frac{\partial R_i}{\partial P_j} \approx \frac{\Delta R_i}{\Delta P_j} = \frac{R_i(P_j + \Delta P_j) - R_i(P_j)}{\Delta P_j} \tag{2.40}$$

$$\Delta P_j = \Delta D * (\overline{P_j} - \underline{P_j}) \tag{2.41}$$

$R_i(P_j)$ is the ith response for the starting value, and $R_i(P_j + \Delta P_j)$ is the ith response of the perturbation of the jth parameter. ΔP_j is the perturbation for the jth parameter.$(\overline{P_j} - \underline{P_j})$ is the difference between the upper and lower limit for the jth parameter and ΔD is the set step size. Computation of the S matrix requires one FE-model run per perturbed parameter, plus one FE-run with all the start values of the parameters (M + 1 FE-model runs).

2.4.2 The Updating Technique

The updating can be represented as minimizing of a penalty function, which involves the difference between the measured and the estimated mode shapes and the eigenvalues. The nature of this type of penalty function requires the problem to be linearised and optimised iteratively (Mottershead and Friswell, 1995). To keep the parameters inside proper values, an upper and lower bound could be applied (Mordini et al., 2007). These methods are versatile with a wide choice of parameters to be updated and the possibility to weight both the measured data and the analytical parameters (Mottershead and Friswell, 1995). The usage of weighting matrices needs engineering insight, but may be used as a powerful tool to obtain excellent correlation between experimental and numerical dynamic properties (Mordini et al., 2007 and Mottershead and Friswell, 1995).

A normal convergence criteria is the Modal Assurance Criterion (MAC), given in eq. 2.42 and another criteria is the eigenfrequency deviation. The second criteria comes in different forms, one is given in eq. 2.43 from Mordini et al., 2007.

$$MAC(\phi_e, \phi_a) = \frac{|\phi_e^T * \phi_a|^2}{(\phi_a^T, \phi_a) * (\phi_e^T, \phi_e)} \tag{2.42}$$

$$f_{dev} = \frac{1}{n} \sum_{x=1}^{n} \frac{|f_{a,x} - f_{c,x}|}{f_{a,x}} \tag{2.43}$$

Both MAC and f_{dev} are calculated for each iteration. Subscript a indicates analytical value from the FE model and subscript e indicates experimental value. n is the number of eigenfrequencies considered. MAC varies between a value of one and zero; a value of one entails that the mode shape vectors are scalar multiples of one another. f_{dev} approaches zero as the analytical eigenfrequencies gets closer to the numerical ones.

An effective and popular method for model updating is based on the sensitivity matrix. It is represented in terms of a first order Taylor series, as given in equation 2.44 (Brownjohn et al., 2001):

$$\{P_u\} = \{P_o\} + [S]^+(\{R_e\} - \{R_a\}) \tag{2.44}$$

Where $\{R_e\} - \{R_a\}$ is the difference between the experimental and the analytical responses considered.$\{P_u\}$ is the updated parameter-value and $\{P_o\}$ is the current parameter-value. The pseudo-inverse matrix of sensitivity $[S]^+$ depends on the number of updating parameters (M) and the number of responses considered (N). Brownjohn et al., 2001 give this equation for $[S]^+$:

$$[S]^+ = \begin{cases} [S]^{-1} & \text{for N} = \text{M} \\ ([S]^T[S])^{-1}[S]^T & \text{for N} > \text{M} \\ ([S]^T([S][S]^T)^{-1} & \text{for N} < \text{M} \end{cases} \tag{2.45}$$

The sensitivity matrix may be calculated for each iteration or constant. Mordini et al., 2007 recommend a separate computation of the sensitivity matrix for each iteration. This computation requires one FE-model run per perturbed parameter, and therefore it may be time consuming. Mordini et al., 2007 also open for using the initial matrix in some time consuming cases.

The Finite Element model to be used in model updating, and the parameters to be included in the updating need to be thoroughly prepared. As stated by Brownjohn et al., 2001, the following may lead to problems with ill-conditioning or divergence of the model updating:

- If the initial discrepancies between the analytical and experimental modal parameters are too large.
- If model updating is attempted on too many parameters, or ineffective parameters.
- If parameter intervals are set too narrowly, there may not be any combination of parameter values resulting in excellent correlation between experimental and numerical modal parameters.
- Comparisons of parameters or responses with varying "sizes".

For this reason, the initial FE-model should be analysed and macro-modelled to limit the initial discrepancies between analytical and experimental modal parameters. After the initial sensitivity matrix is calculated, insensitive parameters should be removed from the parameter matrix used in model updating. If there are still a large number of parameters, considerations to remove more parameters should be made, based on the relative sensitives of the remaining parameters. To counteract the problem with comparisons of responses with, in general, vastly different values, a weighing matrix should be adopted.

3. Methods

One of the two identical Palisaden buildings from the Pentagon II project is modelled based on the production drawings provided from Høyer Finseth AS. The building is modelled using the popular FE-software, SAP2000 with the use of Open Application Programming Interface (OAPI). OAPI facilitates the use of a variety of programming languages with SAP2000. The programming language Python is used to access the OAPI. This chapter will provide a detailed description of the building and construction works, the experimental campaign, and following the FE-model analysis of the structure in the following cases;

1. FE continuum model
2. Connector elements
3. Parametric study of changes in plan geometry

Figure 3.1: Photograph of the building.

3.1 Description of the Building

The Palisaden building is located in Ås, and is used for student apartments. It is eight-storeys tall in addition to a concrete basement. The total height of the building is approximately 23.6 meters, and the plan area is approximately 15 by 23 meters.

The construction works consist of massive cross-laminated timber panels for both wall- and slab elements. The slab elements span from outer wall to outer wall and have dimensions of up till 15 m by 2,5 m. The longitudinal walls consist of two elements, each spanning half the length of the building. In the centre of the building, a cross-laminated timber elevator shaft is located, which braces the building from lateral loading. The thickness of the wall elements gradually decrease along the height of the building, as indicated in figure 3.2. The slab elements have a constant thickness of 180 mm, with the exception of the roof slab, which has a thickness of 200 mm.

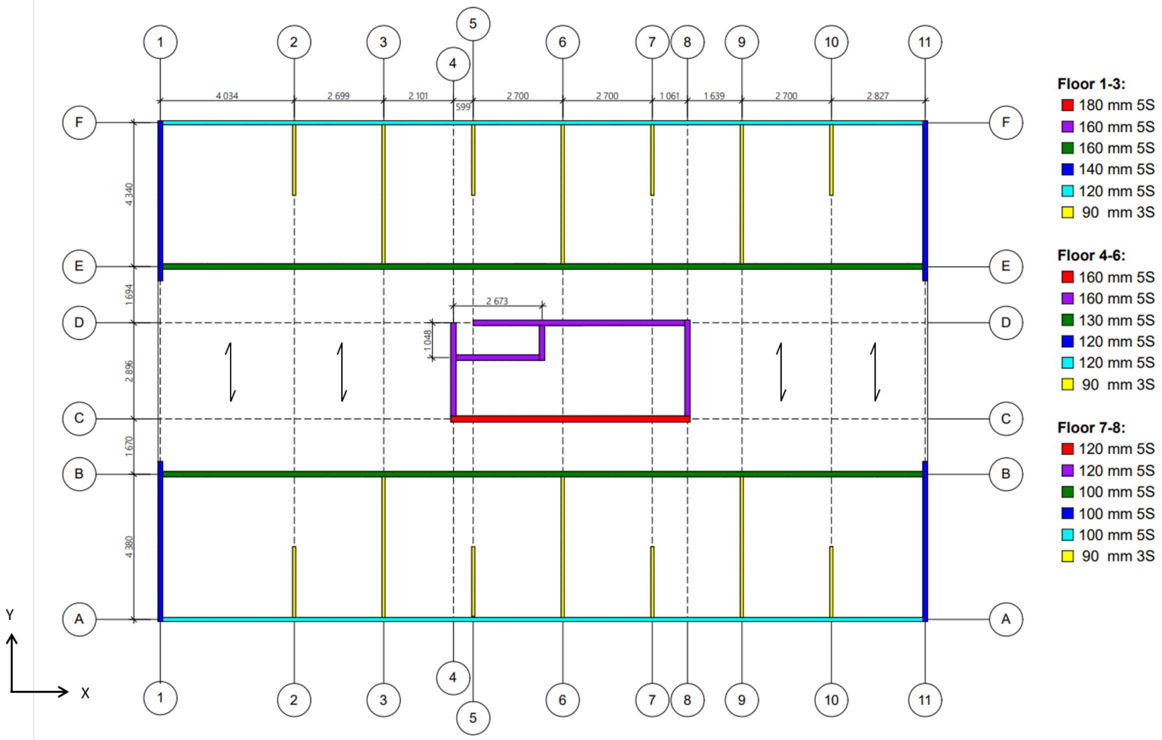


Figure 3.2: Floor plan of the Palisaden building.

The CLT-panels are connected through screws, angle brackets and steel plates. The shear walls are connected with steel plates that are continuously welded throughout the height of the building. The steel plates are also welded to the foundation of the building.

3.2 Experimental Campaign

On October 25th, 2019, ambient vibration measurements were conducted on the building. Accelerometers of type PCB 393812 were used. They have a sensitivity of approximately 10000 mV/g, a frequency range from 0.15 Hz to 1000 Hz, and a measurements range up to 10 m/s^2. The cut-off frequency of the anti-aliasing filter was set to 10 Hz. The number of samples was set to N = 360000, which resulted in a measurement time of 1 h (Aloisio et al., 2020). To estimate the modal parameters, the Stochastic Subspace identification method (SSI) was used.

Three stable modes were detected in the 0-10 Hz range; The first two are translational and the third is torsional. The first mode of vibration, at frequency of 1.913 Hz is a UY mode; the second at frequency of 2.414 is UX mode; the third at a frequency of 2.688 Hz is RZ mode (Aloisio et al., 2020).

The mode shape vector data provided to the authors were SSI-cov and FSDD processed measurements of floor 7 and floor 8. A total of five accelerometers were placed in the building; three reference accelerometers were placed on 8th floor, as indicated by position 1, 2 and 3 in figure 3.3, and two were placed on the 7th floor in position 1 and 2. The accelerometers measured vibrations in two orthogonal directions i.e. the x- and y direction.

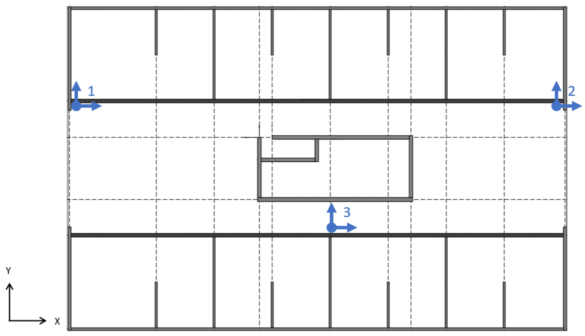


Figure 3.3: Measurement points.

The provided mode shape vectors were processed to later be used as response data with the finite element model. As the vectors were imaginary numbers, their magnitudes were calculated, and the average values of their amplitudes were used to generate three mode shape vectors, one for each of the stable modes discovered within the 0-10 Hz range. The UX- and UY- mode shape vectors contain two numbers, each of the average values of the amplitudes of vibration in the x- and y-direction, in the 7th and 8th floor. The values are then scaled on the 8th floor. To calculate the rotational mode shape vector, the mean amplitude of vibration in the y-direction, y_{mean}, is calculated for measurement points 1 and 2 in the 7th and 8th floor. The rotations are then calculated by formula 3.1, with L_x being the length of the building in the x-direction.

$$\Theta = \tan^{-}1 \frac{y_{mean}}{0.5 * L_x} \tag{3.1}$$

The calculated experimental mode shapes for floor 8 and 7 is:

$$UX_e = \begin{Bmatrix} 1 \\ 0.707 \end{Bmatrix}, UY_e = \begin{Bmatrix} 1 \\ 0.734 \end{Bmatrix}, RZ_e = \begin{Bmatrix} 1 \\ 0.783 \end{Bmatrix}$$

3.3 OAPI with Python

OAPI is a powerful tool, which opens up for controlling a large number of SAP2000's functions in external scripts. This has a lot of benefits, as it enables the possibility of running series of analyses with pre-programmed changes to the model for each new run. Analysing sensitivities of a large number of model parameters without any automation would be extensively time-consuming in comparison.

Multiple scripts are created to perform different analyses or modifications to the models, and to provide more information about the analysis performed, some of the more important scripts are included in Appendix A. Table 3.1 gives an overview of the different scripts provided in the appendix, with brief description of their functionality. All scripts are written in Python version 3.8.

Table 3.1: Overview of Python scripts.

File name	Page no.	Description
C1_SA_MU	69	A script for sensitivity analysis and model updating. The script is intended to work on a pre-created model with defined groups and materials corresponding to the names in the script. The algorithm consist of, first the sensitivity analysis of the defined parameters. The next step is exclusion of the insensitive parameters. Last the algorithm performs the model updating and plotting of the results.
C2_Split	82	A script to make the original model ready for the implementation of link elements. The algorithm deletes and redraws walls to create nodes for the link elements. It also split some floor/walls elements to crates new nodes for link elements.
C2_SA	84	A script for sensitivity analysis. The script is intended to work with a pre-created model using the updated materials and pre-created link elements. The algorithm consist of a sensitivity analysis of the different link elements.
C3_Geo	96	A script for creating Geo-1, Geo-2 and Geo-3, and then running through the variation of eccentricity. The script creates and saves a new model for each variation. The new models are created with the updated material and the results are collected at each variation.

3.4 FE Continuum Model

3.4.1 FE-Modelling

The geometry of the building was first modelled in Archicad (Graphisoft, 2019), and converted to an IFC-format to be imported to SAP2000. This method did however not support the inclusion of windows or doors to the base FE-model. The base building was modelled considering the CLT-elements as a continuum; implying that the connectivity of the structural elements is contained by shared nodes. This means no additional stiffness is provided through the connectors of the building. As the openings are not included in the geometry, their effects are included in reductions factors. It was observed that only G1 needed the reduction from early updating procedures. A factor of 0.75 for the door sections and 0.77 for the window sections, based on ratio between total wall and opening area. Door sections will be denoted with a d, and window sections with a v. Door sections are located in axis B-B and E-E and have a green color, while window sections are located in axis F-F and A-A and have a cyan color in figure 3.2.

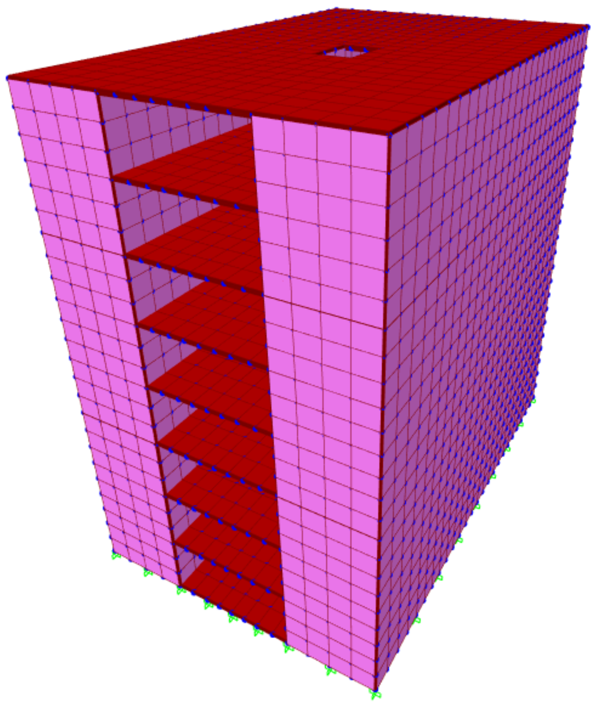


Figure 3.4: FE-model.

The floor and wall elements were modelled as four-node orthotropic, thin shell elements with reference values of stiffness as given in table 3.2. The shell elements are considered to be thin, meaning the Kirchhoff plate formulation is applied, and the transverse shear deformation is to be neglected. This is justified as ratio between plate thickness and span of the bending curvature does not exceed 1/10. E1, meaning young's modulus in the strong in-plane direction of the CLT-panels is calculated using eq. 2.4 or 2.8 depending on the number of layers. E2, meaning Young's modulus in the weaker in-plane direction, is calculated by eq. 2.5 or 2.9. E3, meaning Young's modulus in the out-of-plane direction is assumed to have an equal reference value in all panels (Aranha, 2016). G1 is the in-plane modulus of rigidity, and its reference value is approximated as $E1/20$. To simplify G2 (shear modulus about the 2-3 plane) and G3 (shear modulus about the 1-3 plane) , they are assumed to have the same relationship to G1 as G_{90} to G_0 from eq. 2.3. The self weight of CLT-panels is obtained from Wallner-Novak et al., 2014. The precise CLT layer configuration is unknown for the authors, but it is assumed a setup consisting of board widths between 20 mm and 40 mm.

Table 3.2: Reference values for CLT-panels.

CLT-Panel (Lamellaes (mm))	**E1** (GPa)	**E2** (GPa)	**E3** (GPa)	**G1** (GPa)	**G2** (GPa)	**G3** (GPa)
CLT 90 3S (30-30-30)	7.46	3.91	0.30	0.37	0.04	0.04
CLT 100 5S (20-20-20-20-20)	6.75	4.62	0.30	0.34	0.03	0.03
CLT 120 5S (30-20-20-20-30)	7.46	3.91	0.30	0.37	0.04	0.04
CLT 130 5S (30-20-30-20-30)	7.73	3.64	0.30	0.39	0.04	0.04
CLT 140 5S (40-20-20-20-40)	7.96	3.41	0.30	0.40	0.04	0.04
CLT 160 5S (40-20-40-20-40)	8.34	3.03	0.30	0.42	0.04	0.04
CLT 180 5S (40-30-40-30-40)	7.46	3.91	0.30	0.37	0.04	0.04
CLT 200 5S (40-40-40-40-40)	6.75	4.62	0.30	0.34	0.03	0.03

3.4.2 Sensitivity Analysis and Model Updating

The parameters to be included in this sensitivity analysis are the stiffness properties of the CLT-elements, the self weight of CLT, Poisson's ratio of the CLT-elements, and the load on the internal floors and the roof. The reference values used for loading and global material properties are given in table 3.3. In early test runs, the updated loading values from Aloisio et al., 2020 were used. It was later observed that this loading was relatively small, and as a result it was decided to increase the roof live loading with a factor of four and the internal live load with a factor of two.

Silseth and Roald, 2013, suggests a total self weight of the roof to be $1.4kN/m^2$ for a roof construction consisting of 250mm pressure-resistant mineral wool, asphalt roofing and 200mm CLT. Withdrawing the CLT self-weight leaves a total self-weight of $0.5kN/m^2$. The exact layout of the roof construction is uncertain to the authors, but the choice of roof loading is still conservative. No loading factors are used, meaning there is no distinction between live loading and self-weight other than the self-weight of the CLT, which is distinguished as an independent parameter.

For the internal floors, Silseth and Roald, 2013, suggests $1.3kN/m^2$ for construction of parquet on parquet base, floorboard, heavy mineral wool (20mm + 130mm) and CLT (180mm). Subtracting the CLT self weight leaves $0.49kN/m^2$. The mass contribution for the live load suggested by European Committee For Standardization, 2004c equals $0.48kN/m^2$, with loading factors corresponding to category A building. Movable partitions loading between $0.5 - 1.2kN/m^2$ from European Committee For Standardization, 2004a. Adding these contributions equals a maximum loading of $2.17kN/m^2$, suggesting that our chosen load is a little high, although this does not accommodate for the imposed load from cladding and the possible additional weigh from the bathrooms.

As for the Poisson's ratio, the scientific literature makes different suggestions; Aranha, 2016 suggests a value of 0.2, for the reason that the panels were not glued on their narrow face. The authors do not know if that is the case in the Palisaden building. Zhang et al., 2021 propose a value of 0.34, while Awad et al., 2017 proposes a value of 0.35. As the model updating is constructed to approximate parameter values corresponding to a certain set of modal parameters, the reference value does not need to be exact; although the real parameter value must be within the set intervals of each parameter.

Table 3.3: Reference values for loading and global material properties.

Parameter	Symbol	Reference value
Load on internal floors	F_i	2.55 kN/m^2
Load on the roof	F_r	0.168 kN/m^2
Density, CLT	ρ	450 kg/m^3
Poisson's ratio, CLT	ν	0.3

All parameters are given intervals based on the maximum and minimum value they may be updated to in the optimization algorithms. These intervals are based on the uncertainty related to their reference value, and the realistic values each parameter may have. With the exception of G1 in sections with doors and windows, all material properties are not to exceed ± 50% of their original value; Door and window sections' G1 properties are not to exceed ± 80 % of their original value.

The sensitivity analysis is performed using the perturbation technique as described in chapter 2.4.1. The set step size in formula 2.41 is chosen to be 10 %. The reference values and upper and lower bounds of the parameters to be perturbed are chosen as described earlier in this chapter.

To evaluate the performance of the updated model, a cost function is implemented, consisting of both the MAC and a eigenfrequency deviation criterion. The value of the cost-function (C) is calculated using eq. 3.2. 100% correlation between analytical and experimental results in a C-value of zero.

$$C = \frac{1}{n} \sum_{x=1}^{n} \left(\frac{|f_{e,x} - f_{a,x}|}{f_{e,x}} + (1 - MAC_x) \right) \tag{3.2}$$

where n is the number of response frequencies with linked mode shapes considered. $n = 3$ for our updating algorithm.

3.4.3 Model Validity

Two tests were carried out to ensure the validity of the base model and to investigate the meshing density to be used in analysis and optimization. The total reaction forces at the base of the building were compared with the total applied forces, and this relationship was investigated with an increased mesh density. The influence of mesh density to the results of modal analysis were also compared.

The FE-program allows meshing of elements to be computed based on both number of element divisions along edge lines of each macro element, and meshing based on maximum element width and height. The difference between these two types of meshing becomes apparent for small macro elements; when meshed based on number of divisions, the meshing density becomes finer. For this reason, meshing based on maximum element size is chosen. The macro elements are divided into new elements with maximum width and height of a length n. The n size having an interval between 2m and 0.2m and step size 0.2m. The results of this analysis is shown in figure 3.5.

From this analysis, the relationship between applied loading and reaction forces stays constant with increased meshing density, which strengthens the validity of the base FE-model. From the modal analysis, the fundamental frequencies decrease with increased mesh density, and this development does not seem to converge. As explained in section 2.3.1, meshing in SAP2000 is considered limited when compared to other more sophisticated FE-programs (Rivera, 2015). This may be the reason to why mesh convergence is not achieved. To keep further analyses comparable, a meshing density of maximum size of 2 by 2 m is chosen.

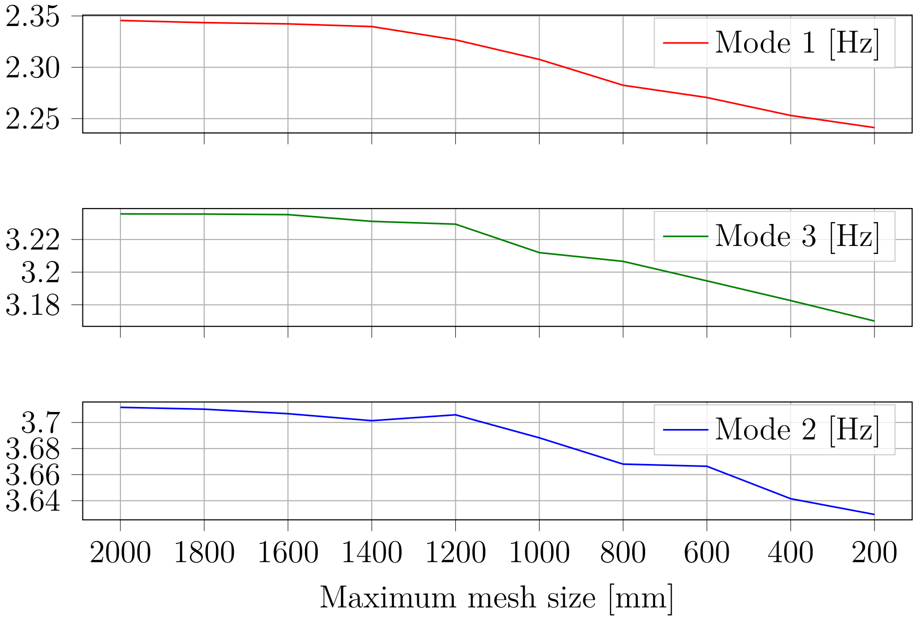


Figure 3.5: Mesh density analysis.

3.5 Connector Elements

3.5.1 FE-Modelling

In this model, the shear wall connectors are included to the model as zero-length linear link elements, acting as springs with a given stiffness between two nodes in all six degrees of freedom found in three-dimensional space. The procedure of applying the link elements is shown schematically in figure 3.6.

The macro element model is adapted as described in table 3.1; the nodes of wall and slab panels are adjusted so that they no longer coincide, with an added thickness of half of the slab thickness between them. Furthermore, the slab nodes and wall nodes are connected with the link elements, based on experimental connector stiffness values. All hold-down brackets are considered to have similar or equal stiffness properties and will mainly provide stiffness in the tensile direction. The configuration of angle brackets consists of various different types of sections, and mainly contribute to stiffness in the shear and tensile direction.

The links are placed in accordance with provided production drawings, with the hold-downs located on the outer edges of the shear wall panels, and the angle brackets located in the middle. All hold-down brackets of corner walls are simplified to be located at the same point, having the sum of reference stiffness from all nearby hold-downs. For this reason, HD4 and HD5 exist.

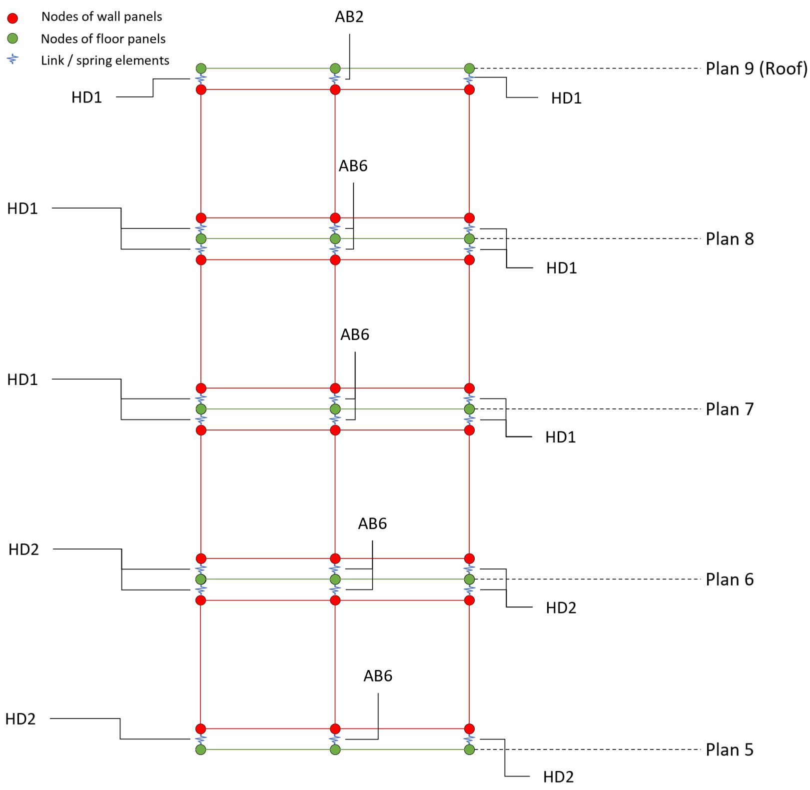


Figure 3.6: Schematic representation of link objects and corresponding properties adopted.

To calculate the reference value of stiffness to be applied to the different types of link elements, the results of experimental analysis from Gavric et al., 2015, are used. For the angle brackets, a linear relationship between the number of screws and stiffness value is assumed, and the values for tensile and shear stiffness are estimated by eq. 3.3:

$$k_a = n_a \frac{k_{exp}}{n_{exp}}, \text{where;} \tag{3.3}$$

where k_a is the analytical stiffness value to be computed, k_{exp} is the experimental stiffness value from Gavric et al., 2015, n_a is the number of nails used in the analytical connector and n_{exp} is the number of nails used in the experiment.

Table 3.4: Overview of types of hold down brackets employed in the building.

Connector type	No. of brackets	Tensile stiffness [kN/mm]
HD1	1	2.65
HD2	2	5.30
HD3	3	7.95
HD4	4	10.60
HD5	5	13.25

As the dimensions of the hold-down brackets are approximately equal throughout the building, the reference value for corresponding tensile stiffness is estimated as a multiplication of number of hold-down brackets acting in the point of interest, based on the experimental values calculated from Gavric et al., 2015. To ensure structural stability of structural objects, a large value of stiffness is attributed to all link objects in the out-of plane translational direction. Figure 3.7 and table 3.5 specify the locations of the different connector types, and the stiffness values applied to the link elements for each type is specified in table 3.6 and 3.4

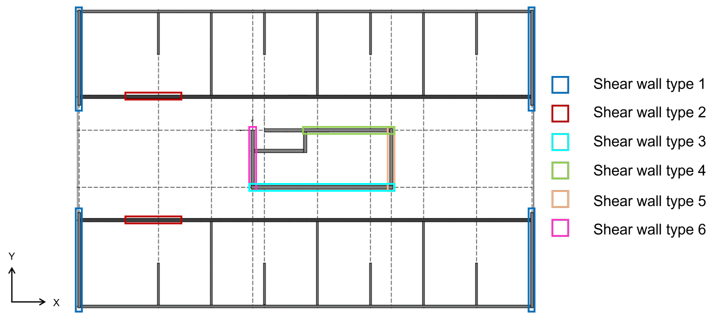
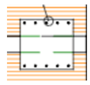
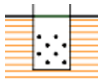
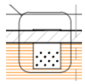
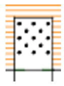
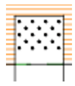
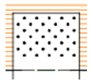
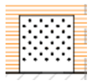
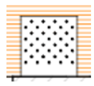


Figure 3.7: Types of shear walls employed in the building.

Table 3.5: Specifications of HD and AB types employed in the different shear wall types.

Shear Wall type	Floor	AB-type	HD-type
Type 1	1-4	AB7	HD3
	5-6	AB6	HD2
	7-8	AB6	HD1
	9	AB2	HD1
Type 2	1-8	-	HD1
	9	-	-
Type 3	1-5	AB9	HD3
	6	AB9	HD2
	7-8	AB8	HD1
	9	AB3	HD1
Type 4	1-3	AB5	HD1
	4-7	AB1	HD1
	8	AB1	-
	9	-	-
Type 5	1-2	AB5	HD2
	3-4	AB4	HD2
	5-6	AB4	HD1
	7-8	AB1	HD1
	9	-	-
Type 6	1-2	AB5	HD2
	3-4	AB4	HD2
	5-6	AB4	HD1
	7-8	AB1	HD1
	9	-	-

Table 3.6: Overview of the types of angle brackets employed in the building (Høyer Finseth AS).

Connector type	**No. of nails**	**Tensile stiffness** [kN/mm]	**Shear stiffness** [kN/mm]	**Illustration**
AB1	5	1.86	0.69	
AB2	8	2.98	1.10	
AB3	12	4.47	1.65	
AB4	13	4.85	1.79	
AB5	18	6.71	2.48	
AB6	25	9.31	3.44	
AB7	35	13.04	4.81	
AB8	36	13.41	4.95	
AB9	36 x 2	26.82	9.90	

3.5.2 Sensitivity Analysis

Another sensitivity analysis is computed with the object to analyse the effects the connectors on the fundamental frequencies of the building under operational conditions. The parameter matrix is expanded with addition of the reference stiffness values in tension and shear for all types of angle brackets, and the stiffness values in tension for the hold-down brackets.

3.6 Geometry Variation

The third and final part of analysis is a parametric study on the effects of irregularities in plan shape of timber buildings, in particular in terms of the order and nature of the fundamental mode shapes and corresponding frequencies.

European Committee For Standardization, 2004c requires modes contributing to a total of 90 % mass participation to be considered in earthquake analysis. For this reason, an analysis is conducted on the number of modes contributing to a total on 90 % mass participation to evaluate the likely amplification of modes corresponding to each plan geometry. The calculation of approximate fundamental frequency from a few different earthquake codes; European Committee For Standardization, 2004c, American Society of Civil Engineers, 2010 and Bureau of Indian Standards, 2002. Then the different approximations of the fundamental frequency is compared for the different geometries.

As briefly described in table 3.1, a Python script is constructed with the intention of creating a variety of Finite Element models corresponding to a variety of different plan geometries. Initially, the algorithms in the script construct a three-dimensional grid, based on length values L_1-L_5, and apply node elements to each point. Another algorithm takes input of a wall-setup matrix, and applies thin, orthotropic shell elements between specified nodes with elastic properties as suggested by the model updating of the original building.

The geometries are picked out due to the interesting likelihood of significant distance between the global center of mass and center of rigidity, and may briefly be described as follows; A right angle shape with equal lengths of each flange (Geo-1); A right angle shape with different lengths of each flange (Geo-2); and a T-shape (Geo-3).

To be able to effectively compare the different geometries, both with one another and the original updated model, the floor area, number of stories and wall height are kept constant for all variations between center of mass and center of rigidity. The specifications of the walls are also kept similar to the original building. The practical usage of these new constructed buildings are not considered of great importance in this study, although the same number and plan area of dormitories is applied. In each of the new geometries, one staircase- and elevator-shaft is implemented with inspiration from the original building. To ensure validity of the new models, total applied force vs. total reaction forces are controlled.

To enforce a change in eccentricity between center of mass and center of rigidity, the lengths of certain walls are elongated. This is obtained through changing the geometry of the original grid of nodes, and may be done in iterations to capture a variety of eccentricities. To ensure the area is kept constant, as a major contributor to total system mass through applied loading, certain other walls are shortened. A more detailed description of this procedure follows in chapters 3.6.1 to 3.6.3. As certain variations of plan geometries may increase the total mass of the system as more volume of CLT with certain thicknesses may be added, the total weight of the system is also monitored.

The first object of analysis is to compare the modal performance of the three implemented geometries to the geometry of the original building. To ensure comparable results, the total mass of the geometries is kept equal. As a matter of definition, this occurs when the width of all dormitories (L_3) is approximately equal to the average width of dormitories in the original building (2.75 m). Table 3.7 shows some of the reference values from the initial building model.

Table 3.7: Reference values of the initial building model.

Parameter	Value
Total Weight	11 900 kN
Total area	345 m^2
Number of storeys	8

The second object of analysis is to describe the modal performance of each new geometry with basis in a calculated eccentricity between center of mass and center of rigidity. This eccentricity is calculated by setting up three load cases, and comparing the results, namely; (Computers & Structures, Inc., 2021)

1. A unit force in the x-direction is applied to the center of mass, and the corresponding rotation caused by this loading, R_{zx}, is measured.

2. A unit force in the y-direction is applied to the center of mass, and the corresponding rotation caused by this loading, R_{zy}, is measured.

3. A unit moment is applied to the center of mass, and the corresponding rotation, R_{zz}, is measured.

4. The coordinates of the center of rigidity is computes as $X_{CoR} = X_{CoM} - \frac{R_{yz}}{R_{zz}}$, and $Y_{CoR} = Y_{CoM} + \frac{R_{xz}}{R_{zz}}$

The length parameter, L_3, is used to enforce the change in eccentricity. For all geometries, values of L_3 between 1.75m and 4.00 m with a step size of 0.25 m.

3.6.1 The Angle-Geometry (Geo-1)

The first geometry to be constructed is the angle-geometry, as shown in figure 3.8. Door sections are marked with a green color, and window sections are marked with a cyan color. L_1 corresponds to the longitudinal length of the protruding parts of the building, while L_2 corresponds to the transverse length of the same parts. L_3 is the width of the dormitories, while L_5 is the length of the dormitories. L_4 is the width of the hallways, and is kept at a constant value of 1.67 m, L_1 is defined as $4 * L_3$, and L_2 is defined as $2 * L_5 + L_4$.

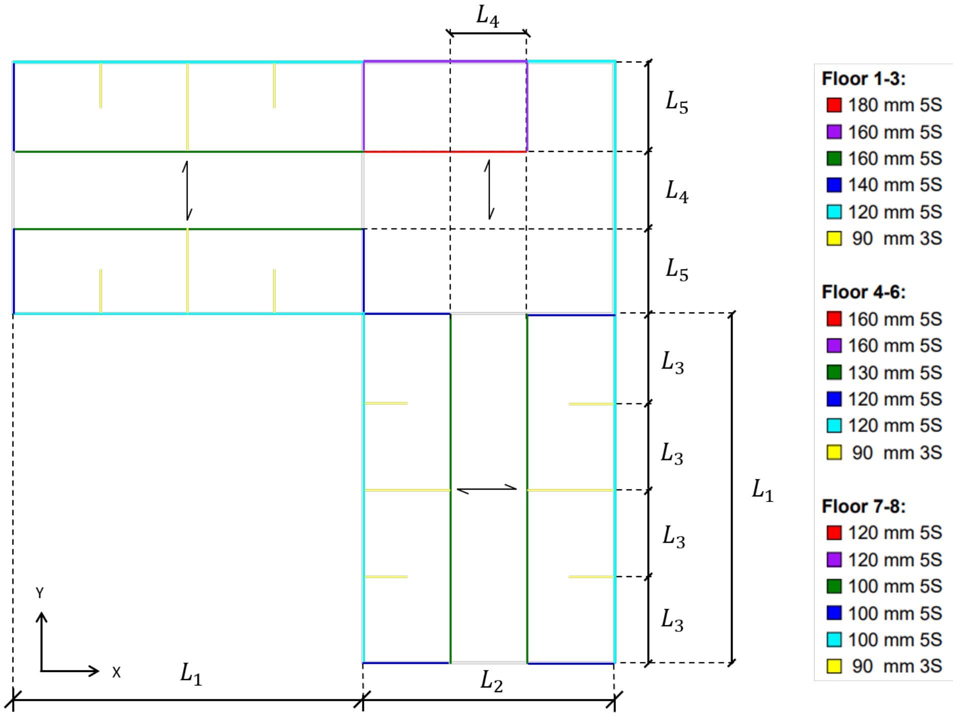


Figure 3.8: Plan drawing of Geo-1.

To keep the area equal throughout the geometric variation, some dependencies between the L parameters and the area ($\mathbf{A}$) are set from the geometry:

$$\mathbf{A} = 2 * (L_2 * L_1) + L_2 * L_2 \tag{3.4}$$

where:

$$L_3 = \frac{L_1}{4} \tag{3.5}$$

$$L_5 = \frac{L_2 - L_4}{2} \tag{3.6}$$

As L_4 is set equal to $1.67m$. Solving equation 3.4 for L_2 yields:

$$L_2 = \frac{-2L_1 + \sqrt{(2L_1)^2 + 4\mathbf{A}}}{2} \tag{3.7}$$

3.6.2 The L-Geometry (Geo-2)

The second geometry to be constructed is the L-geometry, as shown in figure 3.9. Door sections are marked with a green color, and window sections are marked with a cyan color. This geometry has one protruding part one third of the length of the longer part. This means the length parameter L_1 is split into two; $L_{1,l}$, which is defined as $6 * L_3$, and $L_{1,s}$ which is defined as $2 * L_3$.

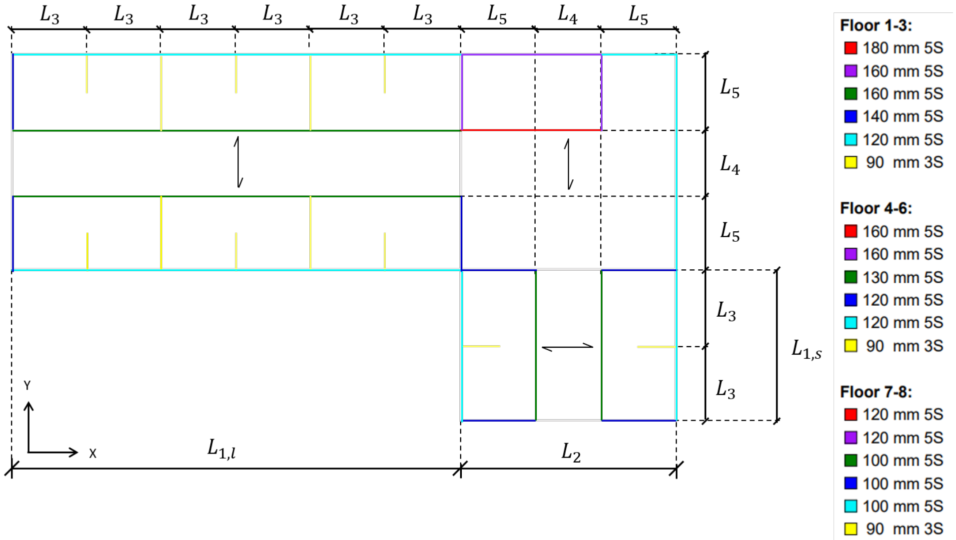


Figure 3.9: Plan drawing of Geo-2.

To keep the area equal throughout the geometric variation, some dependencies between the L parameters and the area ($\mathbf{A}$) are set by the geometry:

$$\mathbf{A} = L_{1,l} * L_2 + L_{1,s} * L_2 + L_2 * L_2 \tag{3.8}$$

where:

$$L_{1,l} = 6L_3 \tag{3.9}$$

$$L_{1,s} = 2L_3 \tag{3.10}$$

$$L_5 = \frac{L_2 - L_4}{2} \tag{3.11}$$

The L_4 is set equal to $1.67m$. Solving the equation 3.8 for L_2 yields:

$$L_2 = \frac{-\frac{4L_{1,l}}{3} + \sqrt{(\frac{4L_{1,l}}{3})^2 + 4\mathbf{A}}}{2} \tag{3.12}$$

3.6.3 The T-Geometry (Geo-3)

The third geometry to be constructed is the T-geometry, as shown in figure 3.10. Door sections are marked with a green color, and window sections are marked with a cyan color. In this geometry, the floor plan of the T geometry is specified in figure 3.10. All the different L constants used in the calculation of the geometry, is defined in the figure.

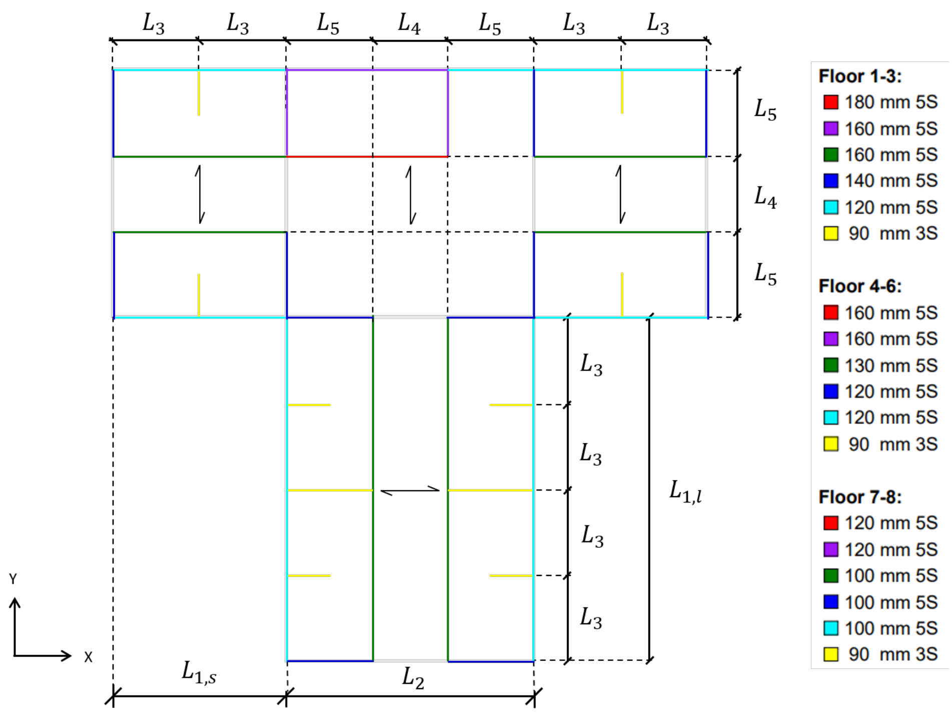


Figure 3.10: Plan drawing of Geo-3.

To keep the area equal throughout the variation, some dependencies between the L parameters and the area ($\mathbf{A}$) are set by the geometry:

$$\mathbf{A} = L_{1,l} * L_2 + 2 * L_{1,s} * L_2 + L_2 * L_2 \tag{3.13}$$

where:

$$L_{1,l} = 4L_3 \tag{3.14}$$

$$L_{1,s} = 2L_3 \tag{3.15}$$

$$L_5 = \frac{L_2 - L_4}{2} \tag{3.16}$$

The L_4 is set equal to $1.67m$. Solving the equation 3.13 for L_2:

$$L_2 = \frac{-2L_{1,l} + \sqrt{(2L_{1,l})^2 + 4\mathbf{A}}}{2} \tag{3.17}$$

4. Results

In this chapter, the results of all analysis earlier described are listed and briefly explained. The sections of this chapter follow the same structure as chapter 3. For discussions based on the results, the reader is directed to chapter 5.

4.1 FE Continuum Model

As described in chapter 3.4, the sensitivity analysis is performed, and the results of this analysis is shown here. The model updating is performed based on the results of the sensitivity analysis. To clearly distinguish between these two operations, the results for the initial part of the analysis is split into two sections. In the sensitivity analysis, a total number of 82 parameters are tested and compared to 6 different responses. The 40 most sensitive parameters are picked out to contribute to the model updating. After 22 FE-model runs, the model updating algorithm reaches the lowest value of the cost function. This run provides the updated parameter values to be used in the following analysis.

4.1.1 Sensitivity Analysis

The normalized sensitivity matrix is displayed in figure 4.1. To isolate the parameters with sensitivities not equal to zero, the sum of the absolute values for each parameter is calculated. Parameters with a sum lower than 10^{-7} are deemed insensitive, and removed from the sensitivity matrix. The sensitivities to the MAC responses are scaled, as their initial values were, on average, two orders of magnitude lower than the frequency responses. A blue color corresponds to a negative sensitivity, i.e. an increase in parameter value decreases the response value. A red color corresponds to a positive sensitivity, meaning an increase in parameter value corresponds to an increase in response value.

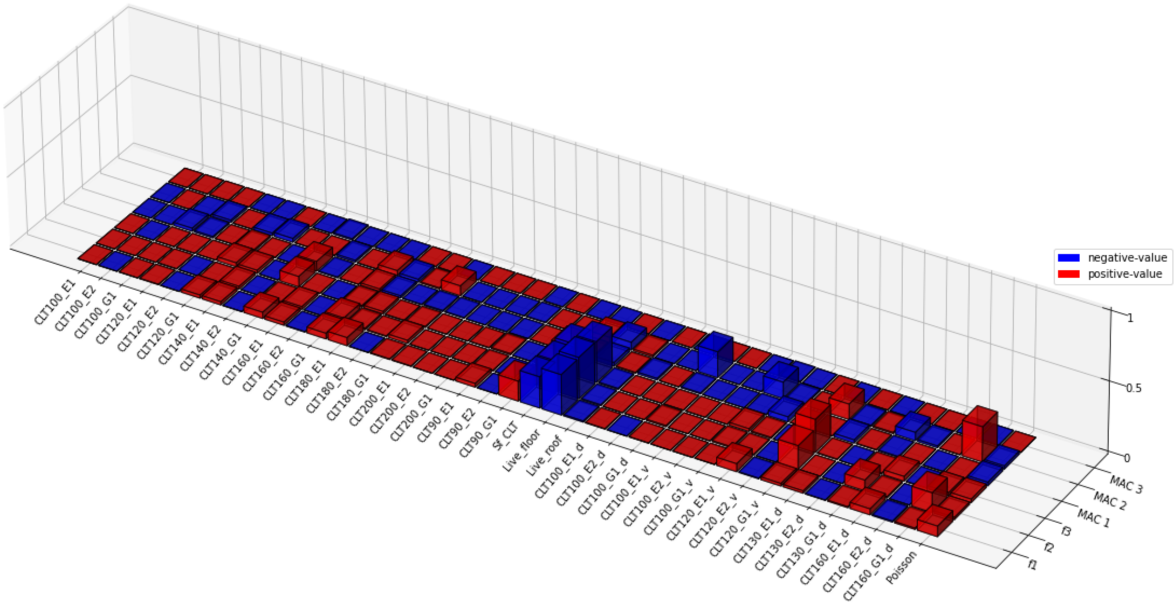


Figure 4.1: Normalised sensitivity values.

4.1.2 Model Updating

The model updating procedure is run based on the sensitivities aforementioned. Figure 4.2 portrays the convergence of the cost-function. A minimum value of the cost function occurs in the 22. run, and the updated parameter values and corresponding response values are picked from this iteration.

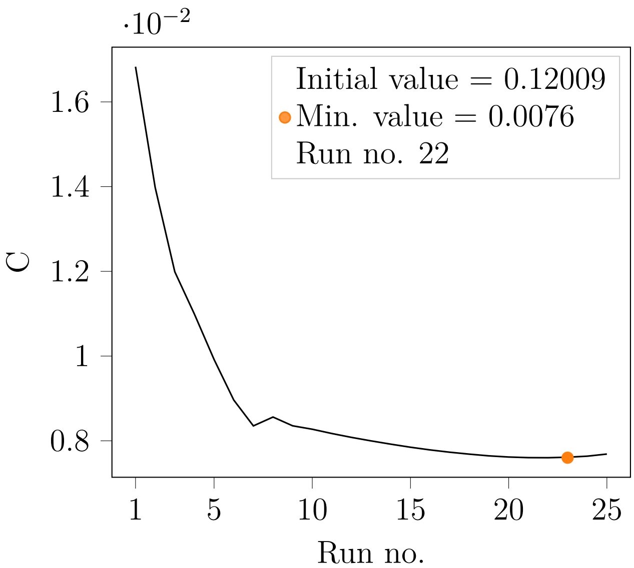


Figure 4.2: Convergence of the cost function during model updating.

A comparison between the experimental and analytical response values is displayed in table 4.1. In this table, the responses are sorted by mode shapes, which explains why the initial fundamental frequency of mode 2 is larger than for mode 3. A MAC value of 1.0 means the analytical mode shape vector is a scalar multiple of the measured mode shape vector.

Table 4.1: Initial-, updated- and experimental response values sorted on mode-shape.

Mode no.	**Initial** [Hz] $f_{a,i}$	**Updated** [Hz] $f_{a,u}$	**Experimental** [Hz] f_e	$\frac{f_{a,u} - fe}{f_e}$ [%]	**Initial MAC**	**Updated MAC**
1	1.988	1.917	1.913	0.209	1.000	1.000
2	3.139	2.455	2.414	1.698	0.999	0.998
3	2.744	2.697	2.693	0.149	1.000	1.000

The initial analytical mode shape vectors for floor 8 and 7 is:

$$UX_{a,i} = \begin{Bmatrix} 1 \\ 0.741 \end{Bmatrix}, UY_{a,i} = \begin{Bmatrix} 1 \\ 0.789 \end{Bmatrix}, RZ_{a,i} = \begin{Bmatrix} 1 \\ 0.783 \end{Bmatrix}$$

The updated analytical mode shape vectors for floor 8 and 7 is:

$$UX_{a,u} = \begin{Bmatrix} 1 \\ 0.739 \end{Bmatrix}, UY_{a,u} = \begin{Bmatrix} 1 \\ 0.800 \end{Bmatrix}, RZ_{a,u} = \begin{Bmatrix} 1 \\ 0.767 \end{Bmatrix}$$

In table 4.2, the 16 parameters undergoing an absolute change larger than 1% is displayed. They are sorted on absolute change, from largest to smallest. These parameter values, also including parameters undergoing a change lower than 1 %, will be applied to the FE-models in the following cases of analysis.

A normal distribution is applied to evaluate the change in stiffness parameter value for E1, E2 and G1. The mean value of change (μ) and the standard deviation of the change (σ) is calculated. The distribution is given in fig. 4.3.

Table 4.2: Initial- and updated parameter value with an absolute change greater than 1.0%.

Parameter		Initial value	Updated value	Change [%]
CLT100 G1 d	(GPa)	0.253	0.051	-80.0
CLT130 G1 d	(GPa)	0.29	0.058	-80.0
CLT160 G1 d	(GPa)	0.313	0.063	-80.0
CLT160 G1	(GPa)	0.417	0.209	-50.0
CLT100 G1	(GPa)	0.337	0.503	49.145
Poisson		0.3	0.429	42.895
CLT100 G1 v	(GPa)	0.26	0.181	-30.389
Live roof	(kN/m^2)	0.168	0.212	25.931
CLT140 G1	(GPa)	0.398	0.494	24.085
CLT180 G1	(GPa)	0.373	0.455	21.949
CLT90 G1	(GPa)	0.373	0.342	-8.241
CLT120 G1	(GPa)	0.373	0.4	7.325
CLT120 G1 v	(GPa)	0.287	0.308	7.143
Live floor	(kN/m^2)	2.55	2.72	6.653
CLT200 G1	(GPa)	0.337	0.354	4.898
Sf CLT	(kN/m^3)	4.5	4.563	1.405

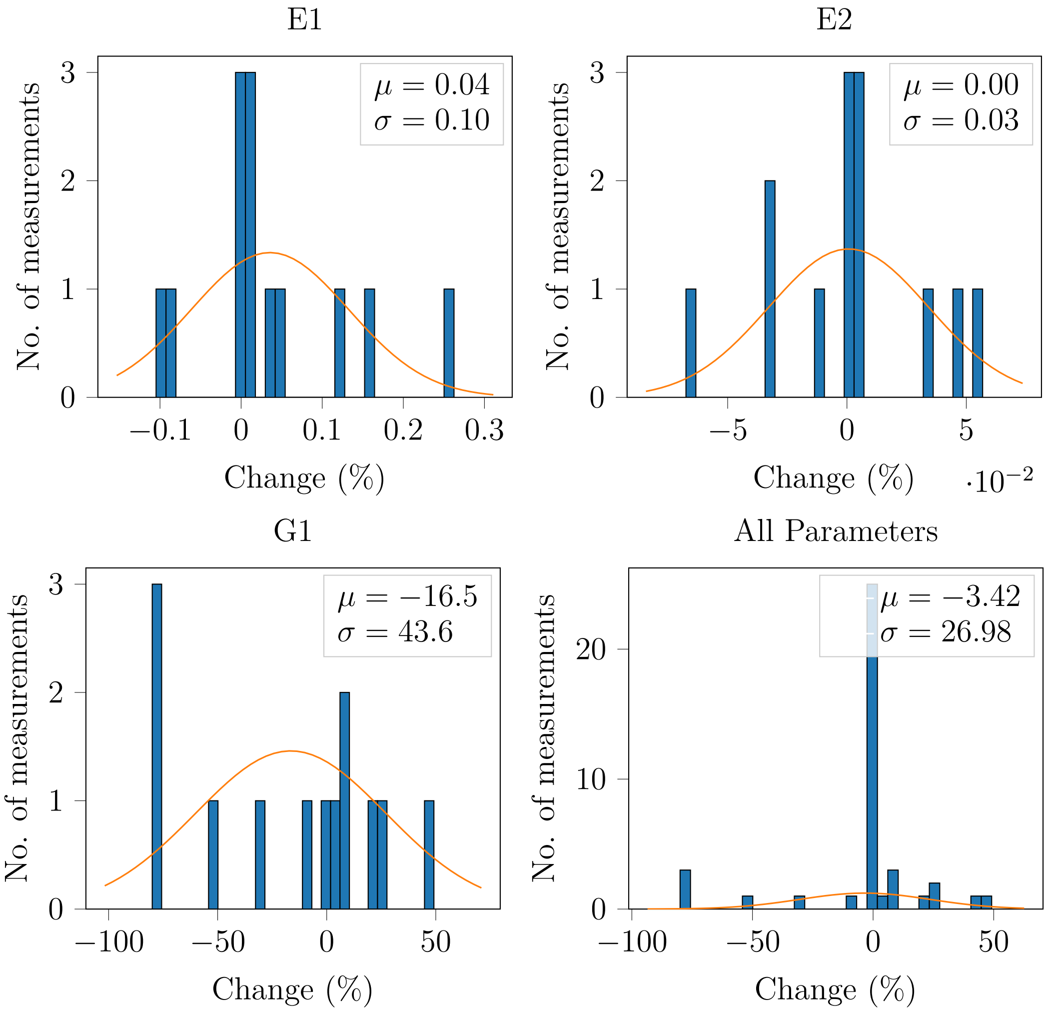


Figure 4.3: Normal distribution of the change in E1, E2, G3, and for all parameters.

When evaluating the results, it is important to notice the the different quantity distribution of wall types. For this reason, these are provided in table 4.3.

Table 4.3: Quantity distribution of wall types.

CLT-Panel	**Tot. length** (m)	**Volume** (m^3)	**Volume fraction** (%)
CLT 120 v	275.04	97.36	19.44
CLT 90	350.72	93.12	18.59
CLT 160 d	137.52	64.91	12.96
CLT 160	116.64	55.05	10.99
CLT 130 d	137.52	52.74	10.53
CLT 120	103.94	36.79	7.35
CLT 100 v	91.68	27.05	5.40
CLT 100 d	91.68	27.05	5.40
CLT 140	57.84	23.89	4.77
CLT 180	21.66	11.50	2.30
CLT 100	38.56	11.38	2.27

4.2 Connector Elements

The sensitivity analysis is run for the new connector parameters. The sensitivity for each parameter is calculated for each response, which provides 6 individual sensitivities for each parameter. The individual normalised sensitivities are provided in figure 4.4. The values of the MAC sensitivities are not scaled in the figure, as they were in the previous analysis. The blue color is corresponding to a negative sensitivity value, therefore a increase in parameter value decreases the response value. The red color is corresponding to the opposite; a positive sensitivity value, therefore an increase in parameter value increases the response value. The taller the box, the more a response is sensitive to a change in the parameter.

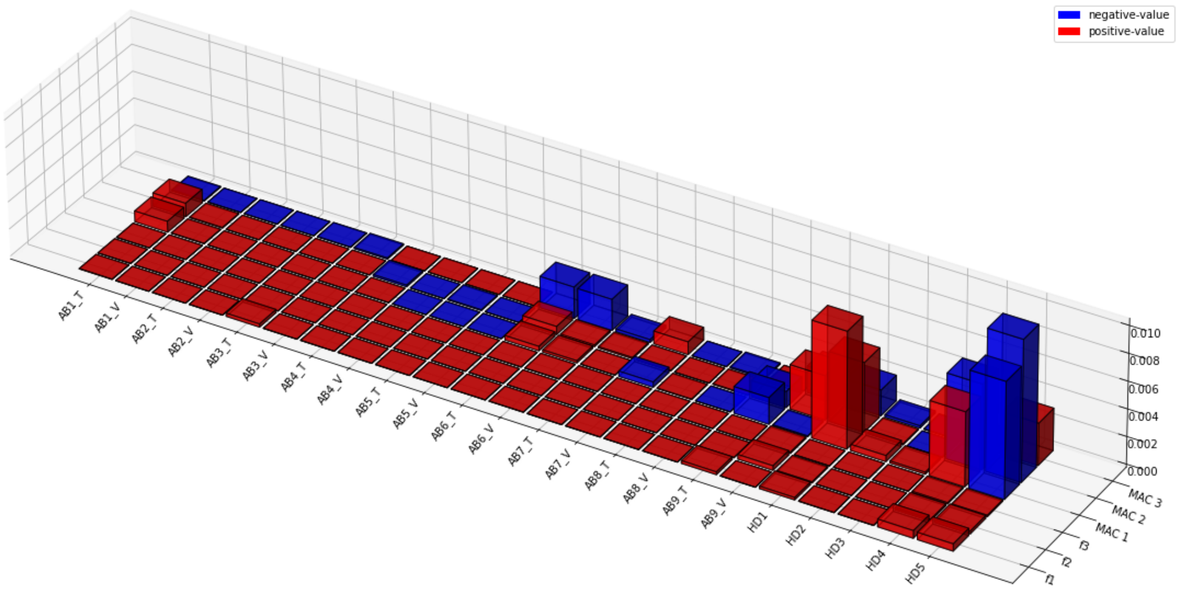


Figure 4.4: Normalised sensitivity values for connector parameters.

4.3 Geometry Variation

The third analysis consists of three new geometries (Geo-1, Geo-2 and Geo-3) and the updated original building (Geo-0). Two analyses are done in this case with the updated parameter values, as described in section 3.6. The results of the first analysis, i.e. the comparison between the four different geometries, are described in section 4.3.1. The results of the second analysis are presented in section 4.3.2.

4.3.1 Comparison of Geometries

In this analysis, the L_3 parameters are equal to $2.75m$ for the new geometries. The results address the mode shape and their frequency values. The difference in frequency values between modes are an important value when assessing the risk of modal amplification, and are therefore provided. The eccentricity in table 4.4 correlates to the eccentricity in the geometry variation analysis.

Table 4.4: Fundamental frequencies for the first five modes.

	Geo-0		Geo-1		Geo-2		Geo-3		Mode shape		
Mode no.	f (Hz)	Δf (Hz)	f (Hz)	Δf (Hz)	f (Hz)	Δf (Hz)	f (Hz)	Δf (Hz)	UX	UY	RZ
1	1.92	-	2.40	-	2.26	-	2.35	-	-	x	-
2	2.45	0.53	2.45	0.05	2.54	0.28	2.40	0.05	x	-	-
3	2.70	0.25	3.04	0.59	2.93	0.39	3.09	0.69	-	-	x
4	6.25	3.55	7.02	3.98	6.81	3.88	6.94	3.85	-	x	-
5	6.94	0.69	7.27	0.25	7.31	0.50	7.07	0.13	x	-	-
Eccentricity (m):	≈ 0		4.4		4.2		2.9				

Table 4.5 display the difference between the fundamental frequencies of the constructed geometries and the original geometry. This reveals some interesting similarities between the new geometries.

Table 4.5: Change in frequency between the constructed geometries and the original geometry.

Mode no.	Geo-1 Δf	Geo-2 Δf	Geo-3 Δf	mean Δf
1	25%	18%	22%	22%
2	0%	4%	-2 %	1%
3	13%	9%	14%	12%
4	12%	9%	11%	11%
5	5%	5%	2%	4%

Table 4.6 provides the modal direction factors for the three first modes.

Table 4.6: Vibration characteristics of the four geometries.

Mode no.		Geo-0	Geo-1	Geo-2	Geo-3
1	U_X	0.0	8.3	0.4	0.4
	U_Y	99.6	91.7	99.4	99.6
	R_Z	0.4	0.0	0.2	0.0
2	U_X	99.8	87.7	96.6	97.7
	U_Y	0.0	8.1	0.3	0.3
	R_Z	0.2	4.2	3.1	2.0
3	U_X	0.2	4.5	3.5	2.4
	U_Y	0.4	0.5	0.5	0.1
	R_Z	99.4	95.0	96.0	97.5

U_X**:** Modal direction factor in the X-direction
U_Y**:** Modal direction factor in the Y-direction
R_Z**:** Modal direction factor about the Z-direction

It is often necessary for engineers to approximate the fundamental frequency of a building for earthquake design purposes. This approximation is carried out in table 4.7 for the European, American and Indian seismic codes.

Table 4.7: Approximation of the fundamental frequency according to different seismic codes.

Standard/guideline	Section	Formula			Value
EN-1998-1 (2004)	4.3.3.2.2 (2)	$f = 1/0.050H^{0.75}$			1.868
ASCE 7-10 (2010)	12.8.2.1	$f = 1/0.0488H^{0.75}$			1.913
IS 1893-1 (2002)	7.6.2	$f = \sqrt{d}/0.09H$	Geo-0	$d = 15.0m$	1.823
			Geo-1	$d = 21.6m$	2.188
			Geo-2	$d = 22.3m$	2.224
			Geo-3	$d = 21.8m$	2.198

H: Total height of the building (= 23.6m)
d: Base dimension of the building along the considered direction

European Committee For Standardization, 2004c requires a number of modes corresponding to at least 90 % mass participation to be considered for earthquake analysis. The total number of modes and corresponding frequencies needed for the geometries to reach a mass participation ratio of over 90% in both UX- and UY-direction is calculated and displayed in table 4.8.

Table 4.8: Mode number and frequency needed to exceed 90 % participating mass ratio.

	Geo 0		Geo 1		Geo 2		Geo 3	
	UX	UY	UX	UY	UX	UY	UX	UY
No. mode	53	55	39	38	39	38	36	33
f (Hz)	15.07	15.28	15.90	15.36	15.59	15.38	15.36	15.06

4.3.2 Eccentricity Variation of the New Geometries

The L_3 interval is run through, and the three first resulting frequencies and their mode shapes are provided for Geo-1, Ge-2 and Geo-3 in fig. 4.5. In this figure, a dotted line denote that the mode changes shape. UX is a translational shape in the x-direction, UY is a translational shape in the y-direction, UD is a translational shape in a diagonal (UX + UY) direction and RZ is a torsional shape around the z-direction, as defined in figure 2.9.

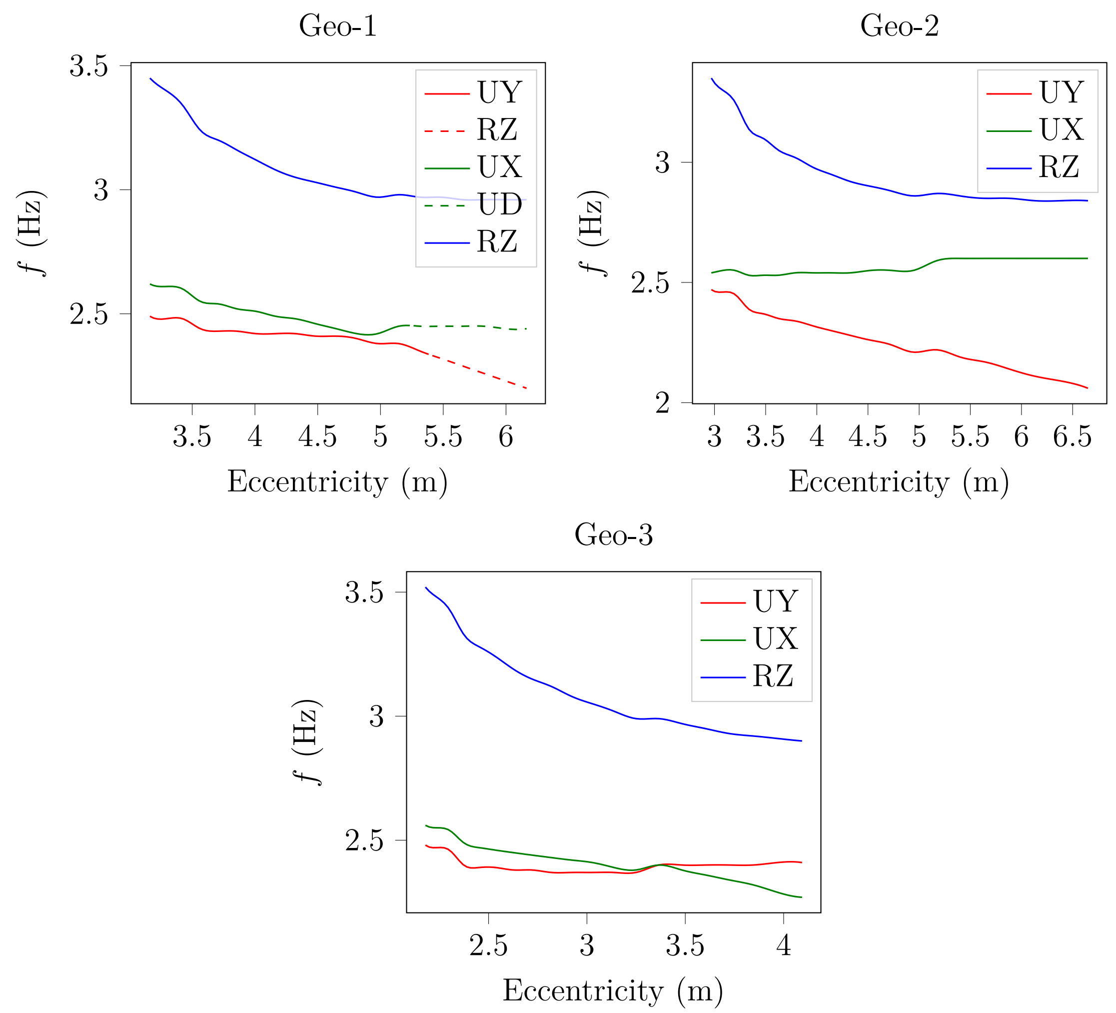


Figure 4.5: Frequency development for three first modes.

To clarify how the variations impact the total masses, the total mass development is plotted in figure 4.6. The figure is scaled on the updated mass from the original geometry, m_0. From min y-value of 94% to max y -value of 1.02% of the original weight, the weight varies from around 11200 kN to 12100 kN (Δ Weight = 900 kN).

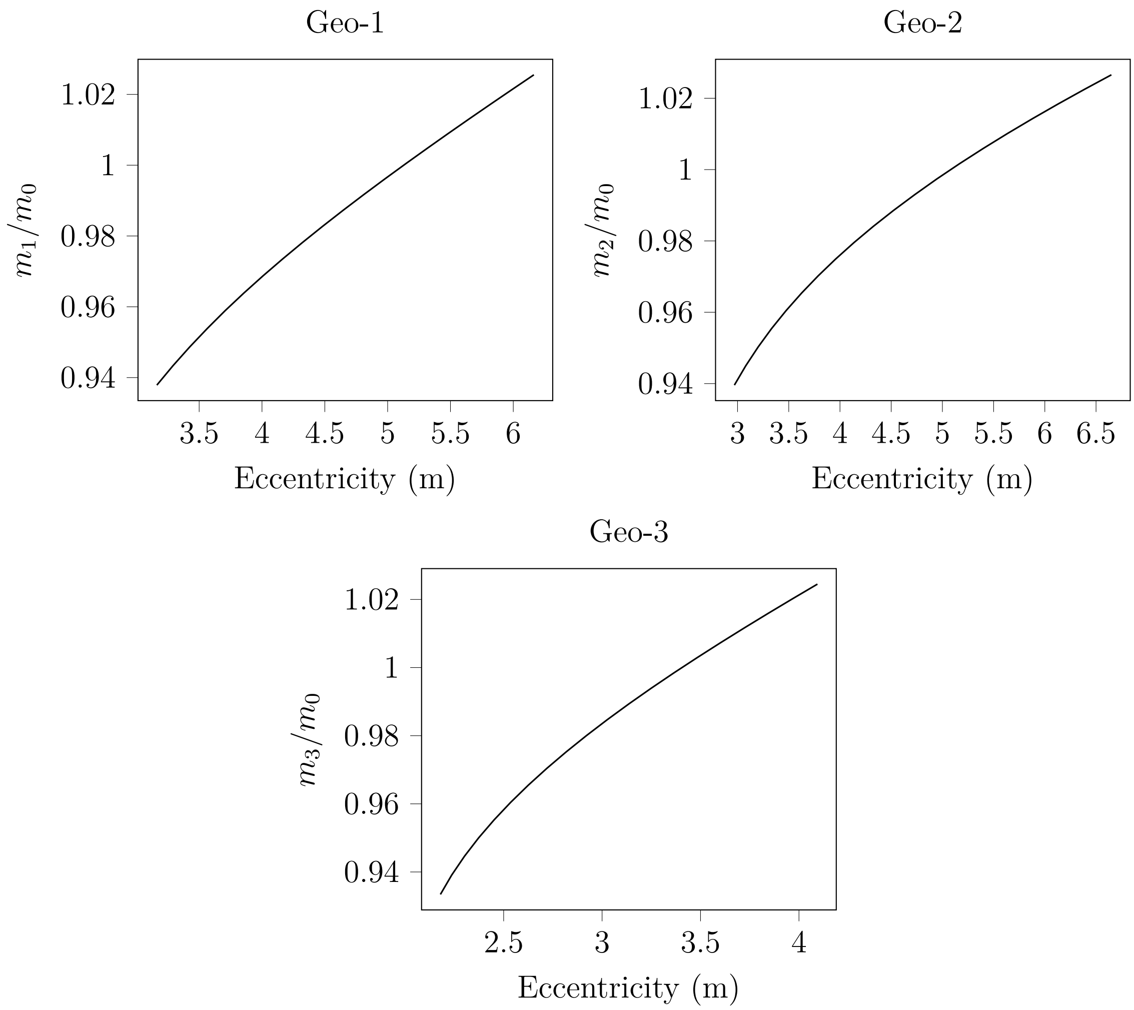


Figure 4.6: Development of total mass.

5. Discussion

The discussion chapter focuses on explaining and evaluating the results provided in chapter 4. The layout of this chapter is kept exactly the same as chapter 4. This is considered the best way to keep the results and discussion orderly. The main conclusion will be drawn in chapter 6.

5.1 FE Continuum Model

A large number of parameters are tested to ensure that the model updating algorithm can reach the experimental data. Especially the effect of shifting modes is important, since initially the second and third mode are in the reverse order. Early tests of the optimization algorithms, showed that using only six stiffness-related parameters and three mass-related parameters would not be sufficient to shift the order of the modes. The number of parameters to be included was subsequently increased, until a total of 82 parameters were included. Splitting up these stiffness-related parameters into a large number of wall-specific moduli has some disadvantages; The first is that the calculation of the sensitivity matrix becomes more computationally heavy and is quite time-consuming. The second disadvantage is that the extent and placement of each wall-type is largely variable, both throughout the floor-plan and throughout the height of the building. This will have some impact on the sensitivity of each parameter. This effect is further explained in the next section.

The differences in sensitivity value between the MAC-responses versus the frequency-responses are large, causing problems in the model updating algorithm. To solve this problem, a scaling factor was implemented, giving the MAC responses higher sensitivity and therefore higher importance to the model updating. Different scaling factors were tried, and the factor with the best results was chosen.

For the sake of clarity, the discussion for Case 1 is also divided into two sections. Section 5.1.1 provides an explanation why the different sensitivities occurs for the different parameters, with a focus on direction and size of the sensitivity. Section 5.1.2 presents the updated parameters, and discusses the difference between the experimental- and analytical data.

5.1.1 Sensitivity Analysis

It is clear that the sensitivity of the frequencies is much larger than the sensitivity of the MAC. The scaling factor applied to the MAC-responses reduce this difference. There is not a clear connection between a parameter's sensitivity regarding the frequencies and the sensitivity regarding the MAC. This is not surprising due to the way MAC is implemented. The frequencies have a high theoretical dependence on stiffness and mass, and the MAC in our case is more dependent of the local difference in movement between the measured floors. To make the most sense of the discussion, the frequencies are discussed first, then the mode shape vectors.

To focus first on the three frequencies, the results are to some degree as expected. The mass-related parameters all pull in a negative direction for an increase in parameter value. The mass parameters are also the most important parameters for changing the frequencies. Each mass parameter has approximately the same sensitivity for each frequency response. Therefore these parameters does not contribute to mode shifting. The most important mass parameter is the live-loading of the internal floors, followed by ρ_{CLT} and then the less-important live-loading on the roof. Live-floor and live-roof are a combination of the live loading and all imposed self weight, excluding the self weight of the CLT-panels. This is due to the extent of the parameter in the building; eight floor-masses versus only one roof-mass. ρ_{CLT} is a "global" parameter responding to all the CLT wall/floor types. It is interesting that ρ_{CLT} have a smaller sensitivity than the floor mass.

For the stiffness parameters the G1 and then E1 have the highest sensitivity. Each of the E1 and G1 parameters have positive values for sensitivity; an increase in the stiffness parameter value causes an increase in the frequencies, which is expected. E3, G2 and G3 for all CLT-panels are determined as insensitive. E2 has the smallest impact of the three stiffness parameters included in the results. The impact of E2 is negative for most of the parameters and frequencies, but not for all of them. The impact on the natural frequencies of all the E2 parameters is quite inconsequential.

The different percentage of total volume of each different wall type impacts the specific wall types' sensitivity. More prevalent wall types will have a larger sensitivity. For example CLT 120 v has relatively high sensitivity, the same goes for CLT 90 as well. Their placement relative to the axis system impacts the individual sensitivity relative to the frequencies. This effect is quite clear between CLT 90 and CLT 120 v. CLT 90 has its second axis parallel to the y-axis. CLT 120 v has its second axis parallel to the x-axis. CLT 90 G1 has high sensitivity to the first frequency (UY) and low to the second

(UX), and the effect is the opposite for CLT 120 v G1 which has a high sensitivity to the second (UX) and low to the first frequency (UY). For the E1 parameters for these two wall types the effect is reversed, i.e. that CLT 120 E1 impacts the first frequency and not the second, and the effect is the opposite for the CLT 90 E1 parameter. G1 prevents in plane rotation, and E1 prevents out of plane movements along the height of the wall.

For the third frequency, walls with a long distance to the stiffness center has a bigger impact in comparison with walls with a shorter distance to the center of stiffness. The stiffness center is approximately in the middle of the building, because of the symmetry in the floor-plan. Therefore the CLT 120 v G1 has a big impact on this frequency, and CLT 160 G1 has much less impact than the difference in prevalence would suggest.

Poisson ratio is an interesting parameter. The Poisson ratio is a measurement of the deformation perpendicular to the loading direction. When this ratio increases, the frequencies increase; especially the first. Shifting the focus to the MAC, the sensitivity to the two first modes are negative while the third is positive. The MAC-response sensitivity to the Poisson' ratio is relatively small for all three responses.

The MAC sensitivities is more complicated than the frequencies. A MAC value equal to one implies corresponding experimental- and measured mode shape vectors, hence an increase in MAC is a preferred response. It is also important to consider the uneven distribution of experimental measurement points throughout the height of the building. This is a disadvantage for the MAC sensitivity and updating in a number of ways. The MAC is only dependent on the difference in movement of the two measured floors. It would be preferable to have more measured floors and a more even distribution of sensors throughout the height of the building. The low number of measurement points also contributes to MAC being very close to one for several different mode shapes. MAC_2 is the farthest from 1, then MAC_1 and then MAC_3. For both MAC_1 and MAC_2 the experimental data has higher displacement than the analytical model. It is the opposite for MAC_3 with higher rotation in the analytical model than the experimental data.

Similar to the frequencies, the G1 stiffness parameter is the most sensitive stiffness parameter for the MAC-responses. The highest sensitivity values are located for MAC_2 for some G1 parameters; CLT 100 d, CLT 100 v, CLT 120 v and CLT 160 d. A notable difference between these are the sign of the sensitivity values. This difference may relate to the difference between the analytical and the experimental mode shapes. Gathering more of the total displacement and reducing the rotation in the top of the building will therefore be beneficial. CLT 100 v is located in the two top floors, therefore decreasing the G1 parameter will result in increasing the movement and rotation in the top floors. CLT 120 v is located in the six bottom floors, therefore increasing the G1 will decrease the movement and rotation of the 6 bottom floors. This leaves more movement and rotation at the top. Note that CLT 120 v has lower sensitivity even with almost four times as much total volume as CLT 100 v.

The rest of the stiffness parameters have relatively low sensitivities, with the E2 parameter as the lowest. Unlike for the frequencies, the mass parameters have smaller sensitivities compared to the other parameters. Increasing the mass on the roof result in increased MAC values for mode 1 and mode 2, and a decrease in mode 3. The opposite effect is observed for the parameter regarding loading on internal floors. The ρ_{CLT} works similarly to the roof mass. As a global parameter, its effect is observed in all structural elements in the building.

To summarize, the parameters most significant for the modal behaviour of the building are the mass- and loading-related parameters, and the G1 and E1 stiffness parameters for the wall elements. These may be seen as important performance indicators for the modal behaviour of the building; to alter the fundamental frequencies and mode shapes of the building, measures made to the mass, loading and specifically to the G1 and E1 stiffness of wall elements should be considered.

5.1.2 Model Updating

As for the model updating, the updated FE model corresponds well to the measured data. The highest difference between analytical and experimental modal data occur in the second translational mode (UX). As given in table 4.1, the difference is around 1.7 % in terms of frequency value, and 0.002 in MAC-value. The value of the cost function changes from an initial value of 0.12 to 0.0076. A value of zero corresponds to 100% correlation between the experimental data and the analytical model. One reason for the high correspondence between analytical and experimental modal data is the low number of responses adopted in the model updating.

Mode 2 is initially about 30 % larger than the experimental frequency value, while mode 1 and mode 3 are quite close (+4 % and +2 %, respectively). This means the initial stiffness in the x-direction is too high, more specifically the shear stiffness about the 1-2 axis for certain wall elements.

The uneven distribution of quantity of wall elements in each parameter group is also an uncertainty in regard to the updated parameter values. The changes in stiffness parameters of wall elements with either a small quantity or unimportant geometrical placement will be lower as their sensitivities to global modal behaviour is low. A consequent initial misjudgment on stiffness values will not be updated for these walls. For example, if all wall elements are given an unreasonably high shear stiffness, only the walls with high quantity or placement corresponding to a change in the second translational mode will be updated.

Both stiffness parameters E1 and E2 undergo relatively small changes in parameter value (under 1% maximum), with respectively a mean change of 0.04 and 0.00 and standard deviation at 0.1 and 0.03. Because of the small changes, the assigned values are not in violation of the experimental data. Due to the relatively low sensitivities of these parameters, it is difficult to conclude that they are assigned with the correct values.

The G1 stiffness parameters undergo much larger change. This is expected since they have larger sensitivity and they contribute to mode shifting. The mean value of change is -16.5% and a standard deviation of 43.5%. It is also important to notice that a large change occurs in both directions, e.g. CLT 160 changes with - 50% and CLT 100 changes with + 49%. If the change was in one direction the conclusion could be that the initial value is either too high or too low. Change in both directions are more difficult to explain. Some of the difference may arise from actual difference in strength relative to calculation method for the different wall types, thus the formula underestimates values for CLT 100 and overestimates for CLT 160. As the exact configuration layout of the CLT-panels are unknown to the authors, the estimation of parameter values are somewhat uncertain.

For all the three different wall types with doors, the change reaches the increased uncertainty limit of 80% change in a negative direction. The difference in opening area for the walls with doors versus the walls with windows is small. It is not observed the same change for the walls with windows. The difference may be explained in some different ways. First, a door opening may have a much larger effect on the G1 stiffness of a CLT panel in comparison to a window. Alternatively, it may be proposed an alternated load path, condemning the door walls as partition walls with zero horizontal stiffness, as only small portions of these walls are restrained with hold-downs. Because of the hold-downs, it is expected that they should have some kind of contribution to the total stiffness. To reduce the uncertainties around openings, the authors suggest using a more complete FE model with the openings implemented.

The change of the CLT self weight is small (+1.4%), which indicates that the initial value is correct. For the floor loading the change is a little larger (+6.7%). For the roof loading, the change is much larger (+26%). This is probably related to the initial value being too small, as stated in the methods chapter.

Poisson is a parameter with a large change in value (+43%). Scientific literature estimates a value of 0.35, which is higher than the initial value 0.3. The updated value is 0.43, which is even higher than the estimates suggest. The updated value may be a little high, and the given interval of the parameter should perhaps have been set smaller. Either way, this suggests that the initial value is too small.

There are two main reasons that the sensitivity analysis is not calculated at each iteration, as suggested by Mordini et al., 2007. The first reason is that with inclusion of a the large number of parameters, the updating procedure would be computationally heavy and be quite time-consuming. Second reason is that the updating reaches a good correlation. A result of not updating the sensitivity analysis is that the relationship between parameter and response is modeled as linear.

Weighting is used for the MAC response to prevent the breaking effect of the size difference. It was decided that some more weighting would have been complicated to do right and it is considered unnecessary when the updating reaches good correlation.

Although there are some uncertainties regarding the results, the FE model replicates the experimental data quite well. Therefore it is still worth using the results in the other cases to represent how these types of structural elements behave.

5.2 Connector Elements

For the 23 different types of connector elements; i.e. 5 hold-downs and 18 angle-brackets, the 6 responses f_1, f_2, f_3, MAC_1, MAC_2 and MAC_3 are calculated. It is important to note that the MAC responses are not scaled in this case. It is also important to note that the z-axis of figure 4.4 and 4.1 are scaled according to the maximum and minimum sensitivities. As discussed previously, there is no significant correlation between the MAC responses and the frequency responses, therefore frequency responses are commented first and then MAC responses are discussed.

The frequency responses are small positive values for all connector parameters. Thus, increasing the stiffness of the connector result in a small increase in the frequencies. This behavior is expected, as they contribute to the total stiffness of the building. Since the extent of each connection is so small, they have very little impact on the total stiffness of the building. Under low-amplitude excitation, a linear behavior of the connectors is expected. This shows that the implementation of the connection has little impact on the natural frequencies, as concluded by Aloisio et al., 2020.

The MAC responses are much larger than the frequency responses, especially for the hold down brackets. This contradicts with the expectations. As stated in section 5.1.1, there are some disadvantages regarding the experimental mode shapes, which may disturb the results. It would have been very interesting to see if the results were similar with higher resolution experimental mode shapes.

Although the changes in mode shape appear to be quite large in figure 4.4, they are actually quite small in comparison with other parameters. The connectors are introduced in a simplified way, and there are some uncertainties regarding their actual stiffness. Some types of connectors are more common than other types. This is one of the reasons behind the different sensitivity values. Another reason is the location inside the floor plan, with respect to the distance to center of stiffness as explained earlier. A third reason is their distribution throughout the height of the structure, impacting the sign of the sensitivity value.

5.3 Floor Plan Variation

The structure of this section follow the layout of the corresponding results section, with the discussion of the first analysis ($L_3 = 2.75m$) in section 5.3.1. Then, three alternations of the individual geometries is discussed in section 5.3.2.

5.3.1 Geometry Variation

For all the four geometries, the order of the mode shapes is the same, with the torsional mode as the third and the rest of the modes as translational. Nearly all frequencies increase for the new geometries compared to Geo-0. The first mode having the largest increase, and the second the smallest. This leads to the first and second modes becoming closer in natural frequency, and amplifications of modes may occur as a result. For Geo-1 and Geo-3 the difference between f_1 and f_2 is 0.05 Hz, while Geo 2 have 0.28 Hz versus 0.53 Hz for Geo-0. Bigger difference in the increase from the second to the third modes, making them farther apart in the new geometries. Mode 4 has a bigger increase than mode 5 making them closer for the new geometries.

In the first analysis in case 3, there is no clear correlation between the eccentricity and the frequencies. The main difference is related to the shift of stiffness caused by the change in wall layout. All three new geometries have more walls parallel to the y-axis, and therefore increased stiffness in this direction. This entails higher frequencies for the UY-mode.

The same logic may be applied to the UX and RZ modes as well. The RZ modes also have increases in frequency in the new models, entailing that their torsional stiffness is higher. Since the UX mode have a small change between the models, the stiffness is probably similar in all models, which is surprising since some stiffness is moved to the Y-direction. The masses are nearly the same in the geometries, meaning that the total wall volume is similar. Therefore the difference is probably coming from the location of the walls and the distance to the center of stiffness.

The percentage of modal direction factor is manually monitored to determine the characteristics of the mode shapes. The percentage of modal direction factor differs between the models; Geo-1 has 8.3% of this factor in the U_x-direction in mode 1 (UY), which implies a small, diagonal movement. The same is visible for Geo-1's second mode with $U_y = 8.1\%$. All the three new geometries have a small rotation contribution in mode 2 and a small U_x contribution in mode 3, with Geo-1 having the largest number in both instances.

The fundamental frequencies calculated from the different earthquake codes and guidelines, all have good correspondence with Geo-0. The European and American code only takes into account the height of the building, which is a weakness compared to the Indian seismic code. The European and American codes provide a poorer estimate for the new, constructed geometries. The Indian code, which takes into account base dimensions of the building, provides a better approximation for the fundamental frequency of the new geometries. It is not surprising that they do not predict the advanced geometries, as they are simple formulas intended for use on simple buildings.

As for the mass participation factor, the new geometries reach the 90 % criteria from European Committee For Standardization, 2004c, in fewer modes but with higher frequencies. This is an interesting phenomenon. Some of the difference relates to Geo-0's more advanced geometry, with several different room types and a more advanced elevator- and stair-shaft, resulting in more modes in the frequency range. Therefore the difference in number of modes probably would be smaller if the new geometries were also more advanced. It was also observed that the mesh size impacted the number of modes, but not the frequencies needed to reach the criteria. The difference in modes and frequencies needed to exceed the criteria for the UX- and UY-direction is quite small for all the geometries. The criteria is exceeded between $f = 15Hz$ and $f = 16Hz$ for all geometries. Since Geo-0 needs the most modes to reach the criteria, the risk of modal amplification increases compared to the others.

5.3.2 Eccentricity Variation of the New Geometries

This section first addresses some challenges with the variation procedure, then discusses the individual results for each of the three geometries.

The mass increase with the eccentricity for all geometries. This effect will lower the natural frequencies of the model. The mass increase is a result of increased wall volume, as the plan area is kept constant. The increased wall volume also causes the stiffness to be increased, and as a result also the frequencies increase. These two effects are important to keep in mind when comparing the models as it may to some degree obscure the results.

The total mass of the building is continuously monitored and this development is shown in figure 4.6. This development has the largest slope in the beginning of the eccentricity interval, after which the curve flattens out. This phenomenon explains the similar, nonlinear development of the frequencies, at the very least for the RZ-mode. The RZ-mode's development is similar for the three geometries, with a flattening of the curve at the end of the interval. This suggests that the effect of increased mass is somewhat nullified by increased stiffness in this area of the eccentricity interval. The other developments that violate the nonlinear development of the frequencies will be explained below.

Geo-1 may be the most interesting geometry of analysis. The effect earlier explained is visible; the natural frequencies decrease with an increase in both eccentricity and total mass. The total mass versus eccentricity relationship is the most linear of the three geometries. From the starting value of eccentricity, to a value of about 5 m, the UX- and UY-modes portray a somewhat linear reduction in frequency, while the RZ-mode displays an exponential reduction in frequency. The UX-mode having a larger slope than the UY-mode, meaning the difference between their frequencies decrease. This development is unfavorable, as the likely amplification of modes increases. From eccentricity beyond about 5 m, some interesting phenomena are displayed; the frequencies of the RZ-mode start to flatten, while the second mode changes shape from UX to UD and the first mode change shape from UY to RZ. This change may imply that the UX- and UY-modes have merged into a diagonal mode, while at about the same time a new mode has occurred. This phenomena may be seen in the development of modal direction factors for each of the modes. As stated earlier, a torsional mode as the first mode is not recommended.

The development of frequencies and mode-shapes of Geo-2 is less drastic; none of the fundamental modes change in character or shift order. The RZ-mode has an exponential reduction in the beginning of the eccentricity interval, and a flattening of the curve towards the end of the interval. The UY-mode display a somewhat linear reduction for the entire interval, while the UX-mode differs quite a bit from the other geometries. It has a small increase in frequency, indicating that the structure becomes stiffer in the UX-direction, in fact stiff enough to counteract the effect of increased mass. The UX- and UY-mode start off quite close in frequency value, and the difference increases throughout the interval. The UX- and RZ-mode approach each other's frequency value, but do not become particularly close.

The eccentricity variation of Geo-3 display mode shifting, but no change in the characteristics of the modes. The RZ-mode display an exponential reduction with a lesser flattening of the curve towards the end of the eccentricity interval. In contrast to the other geometries, the UY-mode has a nearly constant frequency throughout the interval. This could mean that the stiffness increase in the UY-direction counteracts the effect of the increased mass. The UX-mode display a small, linear decrease in frequency. Around an eccentricity of about 3.4 m the UY- and UX-modes shift, making this an unfavorable geometry of the building due to the amplification effect of modes. It is worth noting that the eccentricity changes less in this geometry than the other two, due to some functional constraints in the model creation algorithm.

5.4 Further Work

The authors have made some recommendations for further studies to be conducted in regard to the analysis made in this thesis.

- The methodology should be applied in other CLT-buildings and structures to compare the results of dynamic identification and model updating. In further studies, where advanced and high resolution FE-models are applied for model updating purposes, the experimental investigation should contain a higher resolution of the mode shape vectors.
- Results of the eccentricity analysis should be elaborated with an empirical, numerical simplified model to adequately reproduce the results.
- The results of seismic analysis should be performed on a representative geometry variation of CLT-buildings.

6. Conclusions

To conclude, the research questions are briefly answered below:

1. How reliable are Finite Element models to replicate the real modal behaviour of timber buildings?

   As the modal parameters of the initial FE-model is incorrect in comparison to the experimental values, the general reliability of FE-models portraying CLT-buildings is somewhat weakened. This is obviously dependent on a number of modelling decisions, as thoroughly described in earlier chapters. In our case, the initial stiffness in the x-direction is too high, resulting in a too high frequency in the corresponding translational mode. The model updating scheme is successful in correcting this error, creating a more representative FE-model to the experimental data.

2. What are the modal performance indicators of timber buildings?

   Sensitivity analysis and model updating of the initial FE-model indicate certain parameters more significant to the modal behaviour of the building. These parameters are the loading- and mass-related parameters, such as the self weight of the structural CLT-elements, and the imposed loads on the roof and floors, and the stiffness parameters, specifically the G1 and E1 stiffness of the CLT-panels. This is as expected; the modal behaviour of structures is a function of mass and stiffness contributions in the system, and the G1 and E1 parameters are the most important stiffness parameters of the structural elements.

3. What is the influence of connectors on the modal parameters?

   The results from sensitivity analysis of shear wall connectors, modelled as linear link elements in an updated FE-model, indicate that the connector elements are insignificant to the modal behaviour of the building. This is in accordance with the findings of Aloisio et al., 2020, and suggest that the behaviour of the building is in fact continuum-like under operational excitation. However, the connector elements are sensitive to modal accordance with experimental values. This may be explained by the low resolution of the experimental mode shape vectors used in this thesis.

4. How do construction features/variables affect timber buildings' modal performance?

   Examination on the modal behaviour of four different plan geometries suggests that the fundamental frequency increase for all geometries compared to the original model. Furthermore, the difference in frequency between the first and second mode is smaller for all the constructed geometries compared to the original model. This is expected as the stiffness is more equal in both translational directions of the constructed geometries. All constructed geometries have a higher modal participation in the R_z-direction in the second mode compared to Geo-0. Geo-1 indicates a higher amount of diagonal movement in the first and second mode of vibration.

   Approximation formulas from the European, American and Indian seismic codes of the fundamental mode is quite correct for the original plan geometry, although only the Indian code provide a nearly correct approximation of the fundamental frequencies of the altered plan geometries. The number of modes contributing to a total of 90 % mass participation is lower for the constructed plan geometries compared to the original model.

   Alteration of plan dimensions of the constructed geometries increase the difference in eccentricity between center of mass and center of rigidity. This effect proves to shift the order of the modes for Geo-3, and to fundamentally change the nature of the first and second mode in Geo-1. The third mode of all geometries undergo a exponential change in frequency, and this development flattens as the eccentricity increases. The results are to some degree obscured as the total mass increases with higher eccentricity as more volume of wall elements are constructed.

References

Alecci, V. and De Stefano, M. (Jan. 2019). Building irregularity issues and architectural design in seismic areas. *Frattura ed Integrita Strutturale* 13 (47): 161–168. DOI: 10.3221/IGF-ESIS.47.13.

Aloisio, A., Pasca, D., Tomasi, R., and Fragiacomo, M. (June 2020). Dynamic identification and model updating of an eight-storey CLT building. *Engineering Structures* 213. DOI: 10.1016/j.engstruct.2020.110593.

Alpaslan, E., Hacıefendioğlu, K., Demir, G., and Birinci, F. (June 2020). Response surface-based finite-element model updating of a historic masonry minaret for operational modal analysis. *Structural Design of Tall and Special Buildings* 29 (9): e1733. DOI: 10.1002/tal.1733.

American Society of Civil Engineers (2010). *ASCE 7-10 Minimum Design Loads for Buildings and Other Structures.*

APA – The Engineered Wood Association (Oct. 2012). *Standard for Performance-Rated Cross-Laminated Timber.*

Aranha, C. A. (2016). *Experimental and Numerical Assessment of the Seismic Behaviour of Log and Cross-Laminated Timber Systems.* Tech. rep. Universidade do Minho Escola de Engenharia: 205.

Arora, V. (Nov. 2011). Comparative study of finite element model updating methods. *JVC/Journal of Vibration and Control* 17 (13): 2023–2039. DOI: 10.1177/1077546310395967.

Awad, V., Giresini, L., Koshihara, M., Puppio, M. L., and Sassu, M. (Mar. 2017). Experimental Analyses and Numerical Models of CLT Shear Walls under Cyclic Loading. *Wood in Civil Engineering.* DOI: 10.5772/65024.

Bayraktar, A., Sevim, B., and Can Altunişik, A. (Feb. 2011). Finite element model updating effects on nonlinear seismic response of arch damreservoirfoundation systems. *Finite Elements in Analysis and Design* 47 (2): 85–97. DOI: 10.1016/j.finel.2010.09.005.

Brandner, R., Flatscher, G., Ringhofer, A., Schickhofer, G., and Thiel, A. (May 2016). Cross laminated timber (CLT): overview and development. *European Journal of Wood and Wood Products* 74 (3): 331–351. DOI: 10.1007/s00107-015-0999-5.

Brownjohn, J. M., Xia, P. Q., Hao, H., and Xia, Y. (2001). Civil structure condition assessment by FE model updating: Methodology and case studies. *Finite Elements in Analysis and Design* 37 (10): 761–775. DOI: 10.1016/S0168-874X(00)00071-8.

Bureau of Indian Standards (2002). *IS 1893-1 Criteria for Earthquake Resistant Design of Structures.*

Ceccotti, A. (May 2008). New technologies for construction of medium-rise buildings in seismic regions: The XLAM case. *Structural Engineering International: Journal of the International*

Association for Bridge and Structural Engineering (IABSE) 18 (2): 156–165. DOI: 10.2749/101686608784218680.

Chopra, A. K. (2007). *Dynamics of structures: theory and applications to earthquake engineering.* 5th ed. Pearson/Prentice Hall.: 191.

Computers & Structures, Inc. (Nov. 2017). *CSI. SAP2000 ultimate (v22.0.0) – Structural Analysis Program - CSI Analysis Reference Manual.*

Computers & Structures, Inc. (2021). *Center of Rigidity.* URL: https://wiki.csiamerica.com/display/etabs/Center+of+rigidity (visited on 05/31/2021).

European Committee For Standardization (2004a). *EN 1991-1-1 (English): Eurocode 1: Actions on structures - Part 1-1: General actions - Densities, self-weight, imposed loads for buildings.* The European Union.

European Committee For Standardization (2004b). *EN 1995-1-1 (English): Eurocode 5: Design of timber structures - Part 1-1: General - Common rules and rules for buildings.* The European Union.

European Committee For Standardization (2004c). *EN 1998-1 (English): Design of structures for earthquake resistance - Part 1 : General rules, seismic actions and rules for buildings.* The European Union.

Flatscher, G., Bratulic, K., and Schickhofer, G. (2015). Experimental tests on cross-laminated timber joints and walls. *Proceedings of the Institution of Civil Engineers - Structures and Buildings* 168 (11): 868–877. DOI: 10.1680/stbu.13.00085.

Gavric, I., Fragiacomo, M., and Ceccotti, A. (2015). Cyclic Behavior of CLT Wall Systems: Experimental Tests and Analytical Prediction Models. *Journal of Structural Engineering* 141 (11): 04015034. DOI: 10.1061/(ASCE)ST.1943-541X.0001246.

Gokdemir, H., Ozbasaran, H., Dogan, M., Unluoglu, E., and Albayrak, U. (2013). Effects of torsional irregularity to structures during earthquakes. *Engineering Failure Analysis* 35: 713–717. DOI: 10.1016/j.engfailanal.2013.06.028.

Hill, C. and Zimmer, K. (2018). *The environmental impacts of wood compared to other building materials.* Vol. 4.

Mordini, A., Savov, K., and Wenzel, H. (Nov. 2007). The finite element model updating: A powerful tool for structural health monitoring. *Structural Engineering International: Journal of the International Association for Bridge and Structural Engineering (IABSE)* 17 (4): 352–358. DOI: 10.2749/101686607782359010.

Mottershead, M. I. and Friswell, J. E. (1995). Finite Element Model Updating in Structural Dynamics. *Solid mechanics and its applications* 38: 282. DOI: 10.1007/978-94-015-8508-8.

Mugabo, I., Barbosa, A. R., Riggio, M., and Batti, J. (Apr. 2019). Ambient vibration measurement data of a four-story mass timber building. *Frontiers in Built Environment* 5: 67. DOI: 10.3389/fbuil.2019.00067.

Raheem, S. E., Ahmed, M. M., Ahmed, M. M., and Abdel-shafy, A. G. (Sept. 2018). Evaluation of plan configuration irregularity effects on seismic response demands of L-shaped MRF buildings. *Bulletin of Earthquake Engineering* 16 (9): 3845–3869. DOI: 10.1007/s10518-018-0319-7.

Reynolds, T., Casagrande, D., and Tomasi, R. (Jan. 2016). Comparison of multi-storey cross-laminated timber and timber frame buildings by in situ modal analysis. *Construction and Building Materials* 102: 1009–1017. DOI: 10.1016/j.conbuildmat.2015.09.056.

Rivera, H. A. (Jan. 2015). Translation of SAP2000 Models to Equivalent-Models for Finite Element Command-Based Softwares: 37–49.

Ross, R. J. (2010). *Wood Handbook, Wood as an Engineering Material: Forest Products Laboratory. Wood handbook - Wood as an engineering material.* Centennial. Vol. 39. 4: 508. DOI: 10.1161/01.RES.39.4.523.

Serano, E., Kliger, R., Eva Frühwald, H., Crocetti, R., Danielsson, H., Mårtensson, A., and Pizza, M. (2015). *Limtreboka*. Norske Limtreprodusenters Forening: 312. URL: https://www.moelven.com/globalassets/moelven-limtre/limtreboka%7B%5C_%7D2015%7B%5C_%7Del2.pdf.

Silseth, M. K. and Roald, B. (Aug. 2013). *471.031 Egenlaster for bygningsmaterialer, byggevarer og bygningsdeler*. SINTEF Byggforsk.

Standoli, G., Giordano, E., Milani, G., and Clementi, F. (2021). Model Updating of Historical Belfries Based on Oma Identification Techniques. *International Journal of Architectural Heritage* 15 (1): 132–156. DOI: 10.1080/15583058.2020.1723735.

Tuhta, S., Aydin, H., and Günday, F. (2020). *Updating For Structural Parameter Identification of the Model Steel Bridge Using OMA*. Tech. rep. URL: www.ijltemas.in.

Wallner-Novak, M., Koppelhuber, J., and Pock, K. (2014). *Cross-Laminated Timber Structural Design*. proHolz Austria: 191. URL: www.proholz.at.

Worth, M., Gaul, A., Jager, S., Omenzetter, P., and Morris, H. (2012). *Dynamic performance assessment of a multi-storey timber building via ambient and forced vibration testing, continous monitoring and finite element model updating*. Tech. rep. Department of Civil and Environmental Engineering, The University of Auckland.

Zhang, L., Tiemann, A., Zhang, T., Gauthier, T., Hsu, K., Mahamid, M., Moniruzzaman, P. K., and Ozevin, D. (Jan. 2021). Nondestructive assessment of cross-laminated timber using non-contact transverse vibration and ultrasonic testing. *European Journal of Wood and Wood Products*: 1–13. DOI: 10.1007/s00107-020-01644-4.

Appendix A.

The appendix consist of four Python scripts; A.1 consist the scripts performing the sensitivity analysis and model updating for the FE continuum model. A.2 is a script which adjusts the original FE-model to implement the link-elements for the connector elements part of analysis. A.3 is a script which performs the sensitivity analysis for the connector elements.A.4 consist of a script for creating various geometries, and running through a variation of wall-lengths to enforce a change in eccentricity between center of mass and center of rigidity.

A.1 C1_SA_MU

```
"""
--------------------------------------------------------------------------------
IMPORTS
--------------------------------------------------------------------------------
"""
import os
import sys
import comtypes.client
import numpy as np
import matplotlib.pyplot as plt
import tikzplotlib
from mpl_toolkits.mplot3d import Axes3D
from math import atan
"""
--------------------------------------------------------------------------------
COPY FROM SAP2000 OAPI
--------------------------------------------------------------------------------
"""
#set the following flag to True to attach to an existing instance of the program
#otherwise a new instance of the program will be started
AttachToInstance = True
#set the following flag to True to manually specify the path to SAP2000.exe
#this allows for a connection to a version of SAP2000 other than the latest
    installation
#otherwise the latest installed version of SAP2000 will be launched
SpecifyPath = False
#if the above flag is set to True, specify the path to SAP2000 below
#ProgramPath = 'C:\Program Files\Computers and Structures\SAP2000 22\SAP2000.exe'
#full path to the model
#set it to the desired path of your model
APIPath = 'C:\CSiAPIexample'
```

```python
if not os.path.exists(APIPath):
        try:
            os.makedirs(APIPath)
        except OSError:
            pass
ModelPath = APIPath + os.sep + 'API_1-001.sdb'
#create API helper object
helper = comtypes.client.CreateObject('SAP2000v1.Helper')
helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
if AttachToInstance:
    #attach to a running instance of SAP2000
    try:
        #get the active SapObject
            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:
            #create an instance of the SAPObject from the latest installed SAP2000
            mySapObject = helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")
        except (OSError, comtypes.COMError):
            print("Cannot start a new instance of the program.")
            sys.exit(-1)
    #start SAP2000 application
    mySapObject.ApplicationStart()
#create SapModel object
SapModel = mySapObject.SapModel
"""
-------------------------------------------------------------------------------
FUNCTIONS:
-------------------------------------------------------------------------------
"""
def E(layers):
    #Calculation of moduli for a CLT panels
    E0  = 11.000
    E90 = 0.370
    a  = 150
    G0 = 0.650

    E1 = (layers[0]*E0  +layers[1]*E90 +layers[2]*E0  +layers[3]*E90 +layers[4]*E0 )/sum(layers)
    E2 = (layers[0]*E90 +layers[1]*E0  +layers[2]*E90 +layers[3]*E0  +layers[4]*E90)/sum(layers)
    E3 = 0.300
    if layers == "CLT90":
        ps = 0.53
    else:
        ps = 0.43

    qs = 1.21
```

```python
    dmax = max(layers)
    G12 = G0/(1+6*ps*(dmax/a)**qs) #E1 /20
    G23 = G12/10
    G13 = G12/10

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

def MAC(Mc,Mm):
    #Calculating MAC
    A = np.dot(np.transpose(Mm),Mc)
    B = np.absolute(np.dot(A,A))
    C = np.dot(np.transpose(Mc),Mc)
    D = np.dot(np.transpose(Mm),Mm)
    E = np.dot(C,D)
    macut = np.divide(B,E)
    return macut

def NMD(Mc,Mm):
    #Calculating NMD
    macut = MAC(Mc,Mm)
    nmdut = ((1 - macut )/macut)**0.5
    return nmdut

def run_model(mat):
    SapModel.SetModelIsLocked(False)
    #Entering parameter values -----------------------------------------------
    SapModel.AreaObj.SetLoadUniform("ALL","DEAD",0,10, True, "GLOBAL",1)
    SapModel.SetPresentUnits(6)
    SapModel.AreaObj.SetLoadUniform("Aroof","DEAD",mat[50,0],10, True, "GLOBAL",1)
    SapModel.AreaObj.SetLoadUniform("Aint","DEAD",mat[49,0],10, True, "GLOBAL",1)
    SapModel.PropMaterial.SetWeightAndMass("CLT_100", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_120", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_130", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_140", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_160", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_180", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_200", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_100_Dor", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_100_Vindu", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_120_Vindu", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_130_Dor", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_160_Dor", 1, mat[48,0])
    SapModel.SetPresentUnits(5)
    SapModel.PropMaterial.SetMPOrthotropic("CLT_100", [mat[0,0], mat[1,0], mat[2,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[3,0], mat[4,0], mat[5,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_120", [mat[6,0], mat[7,0], mat[8,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[9,0], mat[10,0], mat[11,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_130", [mat[12,0], mat[13,0], mat[14,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[15,0], mat[16,0], mat[17,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_140", [mat[18,0], mat[19,0], mat[20,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[21,0], mat[22,0], mat[23,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_160", [mat[24,0], mat[25,0], mat[26,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[27,0], mat[28,0], mat[29,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_180", [mat[30,0], mat[31,0], mat[32,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[33,0], mat[34,0], mat[35,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_200", [mat[36,0], mat[37,0], mat[38,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[39,0], mat[40,0], mat[41,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_90" , [mat[42,0], mat[43,0], mat[44,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[45,0], mat[46,0], mat[47,0]] )
```

```
SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Dor" , [mat[51,0], mat[52,0], mat[53,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[54,0], mat[55,0], mat[56,0]])
SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Vindu" , [mat[57,0], mat[58,0], mat[59,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[60,0], mat[61,0], mat[62,0]])
SapModel.PropMaterial.SetMPOrthotropic("CLT_120_Vindu" , [mat[63,0], mat[64,0], mat[65,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[66,0], mat[67,0], mat[68,0]])
SapModel.PropMaterial.SetMPOrthotropic("CLT_130_Dor" , [mat[69,0], mat[70,0], mat[71,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[72,0], mat[73,0], mat[74,0]])
SapModel.PropMaterial.SetMPOrthotropic("CLT_160_Dor" , [mat[75,0], mat[76,0], mat[77,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[78,0], mat[79,0], mat[80,0]])

#Run model-------------------------------------------------------------
ret = SapModel.Analyze.RunAnalysis()

#Calculate eigenfrequencies------------------------------------------------
NumberResults = 0
Period = []
Frequency = []
CircFreq = []
EigenValue = []
StepNum = []
Modal = []
Mode = []
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
[NumberResults,Modal,Mode,StepNum,Period, Frequency, CircFreq, EigenValue,ret] = SapModel.Results.ModalPeriod(NumberResults, "Modal", "Mode", StepNum, Period, Frequency, CircFreq,EigenValue)
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
GroupElm = 0
NumberResults = 0
Obj = []
Elm = []
LoadCase = ["Modal"]
StepType = ["Mode"]
StepNum = []
U1 = []
U2 = []
U3 = []
R1 = []
R2 = []
R3 = []
ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
[NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,R3,ret] = SapModel.Results.JointDispl("633", GroupElm, NumberResults, Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3)
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
ind = [0,1,2]
U1 = np.absolute(U1)
U2 = np.absolute(U2)
indModeX = np.argmax(U1)
indModeY = np.argmax(U2)
ind.remove(indModeX)
ind.remove(indModeY)
```

```
    indModeZ = int(ind[0])

    #Analyticval mode shape--------------------------------------------------
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_427,U2_427,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("427", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_434,U2_434,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("434", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_554,U2_554,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("554", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_579,U2_579,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("579", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_586,U2_586,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("586", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    ModeA = np.ones((3,3))

    Mode17 = (abs(U2_554[indModeY]) + abs(U2_579[indModeY]) + abs(U2_586[indModeY]))/3
    Mode15 = (abs(U2_427[indModeY]) + abs(U2_434[indModeY]))/2

    ModeA[1,0] = Mode15/Mode17

    Mode27 = (abs(U1_554[indModeX]) + abs(U1_579[indModeX]) + abs(U1_586[indModeX]))/3
    Mode25 = (abs(U1_427[indModeX]) + abs(U1_434[indModeX]))/2

    ModeA[1,1] = Mode25/Mode27

    Mode37 = (abs(U2_579[indModeZ]) + abs(U2_586[indModeZ]))/2
    Mode35 = (abs(U2_427[indModeZ]) + abs(U2_434[indModeZ]))/2

    L = 23.06/2

    rot7 = atan(Mode37/L)
    rot5 = atan(Mode35/L)

    ModeA[1,2] = rot5/rot7
    ModeA[2,0] = 0;ModeA[2,1] = 0;ModeA[2,2] = 0;
    mac1 = 1 - MAC(ModeA[:,0],ModeE[:,0])
    mac2 = 1 - MAC(ModeA[:,1],ModeE[:,1])
    mac3 = 1 - MAC(ModeA[:,2],ModeE[:,2])

    nmd1 = NMD(ModeA[:,0],ModeE[:,0])
    nmd2 = NMD(ModeA[:,1],ModeE[:,1])
```

```
    nmd3 = NMD(ModeA[:,2],ModeE[:,2])

    #MAC or NMD as output-------------------------------------------------------
    mod1 = mac1;mod2 = mac2;mod3 = mac3;
    #mod1 = nmd1;mod2 = nmd2;mod3 = nmd3;

    #Calculate C-value -------------------------------------------------------
    C =  ((abs(f1e-Frequency[indModeY])/f1e) + (abs(f2e-Frequency[indModeX])/f2e) + (
    abs(f3e-Frequency[indModeZ])/f3e))/3 + (mod1 + mod2 + mod3)/3

    f1 = Frequency[indModeY]
    f2 = Frequency[indModeX]
    f3 = Frequency[indModeZ]

    #Ractionforce vs. applied force -------------------------------------------

    Ubrukelig1 = []
    Ubrukelig2 = []
    Ubrukelig3 = []
    Name1 = []
    Name2 = []
    ItemTypeElm = 3
    NumberResults = 0
    Obj = []
    Elm = []
    LoadCase = []
    StepType = []
    StepNum = []
    F1 = []
    F2 = []
    F3 = []
    M1 = []
    M2 = []
    M3 = []

    ret = SapModel.SelectObj.All(True)
    ret = SapModel.SelectObj.CoordinateRange(-10000000, 10000000, -100000000, 10000000,
     999, 1001,False,"Global" ,True ,True, False, True,False, False)
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("DEAD")
    [Ubrukelig1, Name1, Name2, LoadCase, Steptype,Ubrukelig2, F1, F2, F3, M1, M2, M3,
    Ubrukelig3] = SapModel.Results.JointReact("ALL", ItemTypeElm, NumberResults, Obj,
    Elm, "DEAD", StepType, StepNum, F1, F2, F3, M1, M2, M3)
    ret = SapModel.SelectObj.All(True)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    OrgAF = 327.1
    OrgAR = 340.5
    OrgL  = 1039.9573

    OrgM = 8*inp1[49,0]*OrgAF + inp1[50,0]*OrgAR + OrgL*inp1[48,0]
    SapM = sum(F3)

    OrgSap = OrgM/SapM

    return C,f1,f2,f3,mod1,mod2,mod3,ModeA,OrgSap

def S1(mat,F,C,MoverM):
    print('Calculating the sensitivity matrix...')
    f = np.zeros((n+1,6))     # Empty list; frequency
```

```
    moverm = np.zeros(n+1)     # Empty list; frequency
    cmat = np.copy(mat)        # Copy of input parameter matrix

    #Run model with input reference values:
    C[0],F[0,0],F[0,1],F[0,2],F[0,3],F[0,4],F[0,5],kast,MoverM[0] = run_model(mat)
    f[0,0] = F[0,0]; f[0,1] = F[0,1]; f[0,2] = F[0,2]; f[0,3] = F[0,3]; f[0,4] = F[0,4]; f[0,5] = F[0,5];moverm[0]=MoverM[0]
    print('Run with initial parameter values: ')
    print("C = ",round(C[0],5))
    print("f1 = ",round(F[0,0],5))
    print("f2 = ",round(F[0,1],5))
    print("f3 = ",round(F[0,2],5))
    print("MCRIT 1 = ",round(F[0,3],5))
    print("MCRIT 2 = ",round(F[0,4],5))
    print("MCRIT 3 = ",round(F[0,5],5))

    for i in range(n):
        delta_P = prosent * ((mat[i,2]-mat[i,1]))

        mat[i,0] +=  delta_P

        #Run FE-model with perturbed parameter values
        C_i,f[i+1,0],f[i+1,1],f[i+1,2],f[i+1,3],f[i+1,4],f[i+1,5],kast,moverm[i+1] = run_model(mat)
        print(' ')
        print('Run no.: ',i+1)
        print('Perturbation of parameter: ',x_label[i])
        for ii in range(fm):
            if ii>=3:
                delta_R = (f[i+1,ii] - f[0,ii])*ScaMode
            else:
                delta_R = (f[i+1,ii] - f[0,ii])

            S[ii,i] = (delta_R/delta_P)                         #Absolutt
            #S[ii,i] = (delta_R/delta_P)*(mat[i,0]/f[i+1,ii]) #Normalisert
            print('R',ii+1,':  dR =',round(delta_R,4),'  ; dP = ',round(delta_P,4),' ; dR/dP = ',round(S[ii,i],9))
        mat[i,0]   = cmat[i,0]
    return S

def remove_parameters(limit,S,x_label):
    print('Removing insensitive parameters...')
    print('Initial number of parameters: ',len(S[0,:]))
    x_label_all = list(np.copy(x_label))

    Stot = []
    sletta = []
    i = 0
    while i < len(S[0,:]):
        Stot.append(abs(S[0,i]) + abs(S[1,i]) + abs(S[2,i]) + abs(S[3,i]) + abs(S[4,i]) + abs(S[5,i]))
        if Stot[-1] < limit:
            S = np.delete(S,i,1)
            sletta.append(x_label[i])
            x_label = np.delete(x_label,i,0)
        else:
            i += 1
    index = list(range(len(x_label)))
    for i in range(len(index)):
```

```
        index[i] = x_label_all.index(x_label[i])
    print('Number of deleted parameters:  ',len(sletta))
    print('Number of remaining parameters:  ', len(S[0,:]))
    print(' ')
    return S,x_label,index

def update(mat,S,F,C,MoverM,index):
    print('Updating parameters...')
    ST = np.transpose(S)
    A  = np.dot(S,ST)
    B  = np.linalg.inv(A)
    Ce  = np.dot(ST,B)

    C[0],F[0,0],F[0,1],F[0,2],F[0,3],F[0,4],F[0,5],kast,MoverM[0] = run_model(mat)
    print('Run with initial parameter values:')
    print("C = ",round(C[0],5))
    print("f1 = ",round(F[0,0],5))
    print("f2 = ",round(F[0,1],5))
    print("f3 = ",round(F[0,2],5))
    print("MCRIT 1 = ",round(F[0,3],5))
    print("MCRIT 2 = ",round(F[0,4],5))
    print("MCRIT 3 = ",round(F[0,5],5))
    dR =[[(f1e - F[0,0])],
         [(f2e - F[0,1])],
         [(f3e - F[0,2])],
         [(0   - F[0,3])],
         [(0   - F[0,4])],
         [(0   - F[0,5])]]
    dP = np.dot(Ce,dR)

    for ii in range(r):
        print(str(ii+1)+'. Updating:')
        #Endrer parameter verdier
        for i in index:
            new = mat[i,0] + dP[index.index(i)]
            if new > mat[i,2]:
                new = mat[i,2]
            elif new < mat[i,1]:
                new = mat[i,1]
            mat[i,0] = new
        #Run model with updated material
        C[ii+1],F[ii+1,0],F[ii+1,1],F[ii+1,2],F[ii+1,3],F[ii+1,4],F[ii+1,5],kast,MoverM
    [ii+1] = run_model(mat)
        print("C = ",round(C[ii+1],5))
        print("f1 = ",round(F[ii+1,0],5))
        print("f2 = ",round(F[ii+1,1],5))
        print("f3 = ",round(F[ii+1,2],5))
        print("MCRIT 1 = ",round(F[ii+1,3],5))
        print("MCRIT 2 = ",round(F[ii+1,4],5))
        print("MCRIT 3 = ",round(F[ii+1,5],5))
        print(' ')
        dR =[[(f1e - F[ii+1,0])],
             [(f2e - F[ii+1,1])],
             [(f3e - F[ii+1,2])],
             [(0   - F[ii+1,3])],
             [(0   - F[ii+1,4])],
             [(0   - F[ii+1,5])]]
        dP = np.dot(Ce,dR)

```

```
    fors1 = (f1e-F[-1,0])/f1e
    fors2 = (f2e-F[-1,1])/f2e
    fors3 = (f3e-F[-1,2])/f3e
    print('Percentage difference in frequencies from the last update: ')
    print('f1: ',round(fors1,5)*100,'%')
    print('f2: ',round(fors2,5)*100,'%')
    print('f3: ',round(fors3,5)*100,'%')
    print("MCRIT 1 = ",round(1-F[ii+1,3],5))
    print("MCRIT 2 = ",round(1-F[ii+1,4],5))
    print("MCRIT 3 = ",round(1-F[ii+1,5],5))

    return mat,F,C,MoverM


"""
--------------------------------------------------------------------------------
GENERELLE VARIABLER:
--------------------------------------------------------------------------------
"""
#No. update iterations:
r = 25

#Limit value for determine insensitive parameters:
limit = 10^-7

#material properties upper and lower limit:
mup = 1.5
mdown = 0.5

#Special material properties upper and lower limit:
sopp = 1.8
sned = 0.2

#Lodaing factors:
Sf_CLT = 4.5
Live_roof = 0.168
Live_floor = 2.55

#Poissons ratio
pois = 0.3

#Reduction factors for openings
v = 0.77 # Windows
d = 0.75 # Doors

#Pertubasjon for the sensitivity matrix calculation:
prosent = 0.1

#Scaling of the modes hapes
ScaMode = 241.81466810617732/2

#Matrices for storing answers:
#fm = 3 # Updates on only frequencies
fm = 6  # Updates on frequencies and mode shapes
n  = 82 # Number of parmeters
F  = np.zeros((r+1,6))
C  = np.zeros(r+1)
MoverM  = np.zeros(r+1)
S  = np.zeros((fm,n))
```

```

"""
-------------------------------------------------------------------------------
EXPERIMENTAL DATA
-------------------------------------------------------------------------------
"""
#Experimental frequencies:
f1e = 1.913
f2e = 2.414
f3e = 2.693

#Experimental mode shapes:
ModeE = np.array([[ 1.         ,  1.         ,  1.         ],
                  [ 0.70737374,  0.73400209,  0.78296554],
                  [ 0         ,  0         ,  0         ]])
"""
-------------------------------------------------------------------------------
DEFAULT MODEL SETTINGS:
-------------------------------------------------------------------------------
"""
SapModel.SetModelIsLocked(False)
SapModel.SetPresentUnits(5)
#kN_mm_C = 5
#kN_m_C  = 6
#N_mm_C  = 9
#N_m_C   = 10
SapModel.AreaObj.SetMass("ALL",0, True, 1)
SapModel.AreaObj.SetAutoMesh("ALL", 0, 0, 0, 2000, 2000,False, False, False, False, False, 0 ,False, False, False ,"ALL" , 0,False, 1)
SapModel.SourceMass.SetMassSource("MyMassSource", False, False, True, True, 1, ["DEAD"], [1])

"""
-------------------------------------------------------------------------------
CALCULATION OF PARAMETER MATRIX:
-------------------------------------------------------------------------------
"""

CLT90  = [30,30,30,00,00]
CLT100 = [20,20,20,20,20]
CLT120 = [30,20,20,20,30]
CLT130 = [30,20,30,20,30]
CLT140 = [40,20,20,20,40]
CLT160 = [40,20,40,20,40]
CLT180 = [40,30,40,30,40]
CLT200 = [40,40,40,40,40]

#Different cross sections:
E1_90 , E2_90 , E3_90 , G1_90 , G2_90 , G3_90  = E(CLT90 )
E1_100, E2_100, E3_100, G1_100, G2_100, G3_100 = E(CLT100)
E1_120, E2_120, E3_120, G1_120, G2_120, G3_120 = E(CLT120)
E1_130, E2_130, E3_130, G1_130, G2_130, G3_130 = E(CLT130)
E1_140, E2_140, E3_140, G1_140, G2_140, G3_140 = E(CLT140)
E1_160, E2_160, E3_160, G1_160, G2_160, G3_160 = E(CLT160)
E1_180, E2_180, E3_180, G1_180, G2_180, G3_180 = E(CLT180)
E1_200, E2_200, E3_200, G1_200, G2_200, G3_200 = E(CLT200)

CLT_100 = [[E1_100,E1_100*mdown,E1_100*mup],
           [E2_100,E2_100*mdown,E2_100*mup],
```

```
            [E3_100,E3_100*mdown,E3_100*mup],
            [G1_100,G1_100*mdown,G1_100*mup],
            [G2_100,G2_100*mdown,G2_100*mup],
            [G3_100,G3_100*mdown,G3_100*mup]]

CLT_100d =[[E1_100*d,E1_100*d*mdown,E1_100*d*mup],
            [E2_100*d,E2_100*d*mdown,E2_100*d*mup],
            [E3_100*d,E3_100*d*mdown,E3_100*d*mup],
            [G1_100*d,G1_100*d*sned,G1_100*d*sopp],
            [G2_100*d,G2_100*d*mdown,G2_100*d*mup],
            [G3_100*d,G3_100*d*mdown,G3_100*d*mup]]

CLT_100v =[[E1_100*v,E1_100*v*mdown,E1_100*v*mup],
            [E2_100*v,E2_100*v*mdown,E2_100*v*mup],
            [E3_100*v,E3_100*v*mdown,E3_100*v*mup],
            [G1_100*v,G1_100*v*sned,G1_100*v*sopp],
            [G2_100*v,G2_100*v*mdown,G2_100*v*mup],
            [G3_100*v,G3_100*v*mdown,G3_100*v*mup]]

CLT_120 = [[E1_120,E1_120*mdown,E1_120*mup],
           [E2_120,E2_120*mdown,E2_120*mup],
           [E3_120,E3_120*mdown,E3_120*mup],
           [G1_120,G1_120*mdown,G1_120*mup],
           [G2_120,G2_120*mdown,G2_120*mup],
           [G3_120,G3_120*mdown,G3_120*mup]]

CLT_120v =[[E1_120*v,E1_120*v*mdown,E1_120*v*mup],
           [E2_120*v,E2_120*v*mdown,E2_120*v*mup],
           [E3_120*v,E3_120*v*mdown,E3_120*v*mup],
           [G1_120*v,G1_120*v*sned,G1_120*v*sopp],
           [G2_120*v,G2_120*v*mdown,G2_120*v*mup],
           [G3_120*v,G3_120*v*mdown,G3_120*v*mup]]

CLT_130 = [[E1_130,E1_130*mdown,E1_130*mup],
           [E2_130,E2_130*mdown,E2_130*mup],
           [E3_130,E3_130*mdown,E3_130*mup],
           [G1_130,G1_130*mdown,G1_130*mup],
           [G2_130,G2_130*mdown,G2_130*mup],
           [G3_130,G3_130*mdown,G3_130*mup]]

CLT_130d =[[E1_130*d,E1_130*d*mdown,E1_130*d*mup],
           [E2_130*d,E2_130*d*mdown,E2_130*d*mup],
           [E3_130*d,E3_130*d*mdown,E3_130*d*mup],
           [G1_130*d,G1_130*sned*d,G1_130*sopp*d],
           [G2_130*d,G2_130*d*mdown,G2_130*d*mup],
           [G3_130*d,G3_130*d*mdown,G3_130*d*mup]]

CLT_140 = [[E1_140,E1_140*mdown,E1_140*mup],
           [E2_140,E2_140*mdown,E2_140*mup],
           [E3_140,E3_140*mdown,E3_140*mup],
           [G1_140,G1_140*mdown,G1_140*mup],
           [G2_140,G2_140*mdown,G2_140*mup],
           [G3_140,G3_140*mdown,G3_140*mup]]

CLT_160 = [[E1_160,E1_160*mdown,E1_160*mup],
           [E2_160,E2_160*mdown,E2_160*mup],
           [E3_160,E3_160*mdown,E3_160*mup],
           [G1_160,G1_160*mdown,G1_160*mup],
           [G2_160,G2_160*mdown,G2_160*mup],
```

```
          [G3_160,G3_160*mdown,G3_160*mup]]

CLT_160d =[[E1_160*d,E1_160*d*mdown,E1_160*d*mup],
          [E2_160*d,E2_160*d*mdown,E2_160*d*mup],
          [E3_160*d,E3_160*d*mdown,E3_160*d*mup],
          [G1_160*d,G1_160*d*sned,G1_160*d*sopp],
          [G2_160*d,G2_160*d*mdown,G2_160*d*mup],
          [G3_160*d,G3_160*d*mdown,G3_160*d*mup]]

CLT_180 = [[E1_180,E1_180*mdown,E1_180*mup],
          [E2_180,E2_180*mdown,E2_180*mup],
          [E3_180,E3_180*mdown,E3_180*mup],
          [G1_180,G1_180*mdown,G1_180*mup],
          [G2_180,G2_180*mdown,G2_180*mup],
          [G3_180,G3_180*mdown,G3_180*mup]]

CLT_200 = [[E1_200,E1_200*mdown,E1_200*mup],
          [E2_200,E2_200*mdown,E2_200*mup],
          [E3_200,E3_200*mdown,E3_200*mup],
          [G1_200,G1_200*mdown,G1_200*mup],
          [G2_200,G2_200*mdown,G2_200*mup],
          [G3_200,G3_200*mdown,G3_200*mup]]

CLT_90  = [[E1_90,E1_90*mdown,E1_90*mup],
           [E2_90,E2_100*mdown,E2_90*mup],
           [E3_90,E3_100*mdown,E3_90*mup],
           [G1_90,G1_100*mdown,G1_90*mup],
           [G2_90,G2_100*mdown,G2_90*mup],
           [G3_90,G3_100*mdown,G3_90*mup]]

poisson = [pois,    pois*mdown, pois*mup]

M       = [[Sf_CLT,Sf_CLT*mdown,Sf_CLT*mup],
           [Live_floor,Live_floor*mdown,Live_floor*mup],
           [Live_roof,Live_roof*mdown,Live_roof*mup]]

inp1 = np.vstack((CLT_100,CLT_120,CLT_130,CLT_140,CLT_160,CLT_180,CLT_200,CLT_90,M,
    CLT_100d,CLT_100v,CLT_120v,CLT_130d,CLT_160d,poisson))
inp2 = np.vstack((CLT_100,CLT_120,CLT_130,CLT_140,CLT_160,CLT_180,CLT_200,CLT_90,M,
    CLT_100d,CLT_100v,CLT_120v,CLT_130d,CLT_160d,poisson))
x_label = ['CLT100_E1','CLT100_E2','CLT100_E3','CLT100_G1','CLT100_G2','CLT100_G3',
             'CLT120_E1','CLT120_E2','CLT120_E3','CLT120_G1','CLT120_G2','CLT120_G3',
             'CLT130_E1','CLT130_E2','CLT130_E3','CLT130_G1','CLT130_G2','CLT130_G3',
             'CLT140_E1','CLT140_E2','CLT140_E3','CLT140_G1','CLT140_G2','CLT140_G3',
             'CLT160_E1','CLT160_E2','CLT160_E3','CLT160_G1','CLT160_G2','CLT160_G3',
             'CLT180_E1','CLT180_E2','CLT180_E3','CLT180_G1','CLT180_G2','CLT180_G3',
             'CLT200_E1','CLT200_E2','CLT200_E3','CLT200_G1','CLT200_G2','CLT200_G3',
             'CLT90_E1' ,'CLT90_E2' ,'CLT90_E3' ,'CLT90_G1' ,'CLT90_G2' ,'CLT90_G3' ,
             'Sf_CLT'   ,'Live_floor','Live_roof',
             'CLT100_E1_d','CLT100_E2_d','CLT100_E3_d','CLT100_G1_d','CLT100_G2_d','
    CLT100_G3_d',
             'CLT100_E1_v','CLT100_E2_v','CLT100_E3_v','CLT100_G1_v','CLT100_G2_v','
    CLT100_G3_v',
             'CLT120_E1_v','CLT120_E2_v','CLT120_E3_v','CLT120_G1_v','CLT120_G2_v','
    CLT120_G3_v',
             'CLT130_E1_d','CLT130_E2_d','CLT130_E3_d','CLT130_G1_d','CLT130_G2_d','
    CLT130_G3_d',
             'CLT160_E1_d','CLT160_E2_d','CLT160_E3_d','CLT160_G1_d','CLT160_G2_d','
    CLT160_G3_d',
```

```
              'Poisson']

x_label_o = ['CLT100_E1','CLT100_E2','CLT100_E3','CLT100_G1','CLT100_G2','CLT100_G3',
              'CLT120_E1','CLT120_E2','CLT120_E3','CLT120_G1','CLT120_G2','CLT120_G3',
              'CLT130_E1','CLT130_E2','CLT130_E3','CLT130_G1','CLT130_G2','CLT130_G3',
              'CLT140_E1','CLT140_E2','CLT140_E3','CLT140_G1','CLT140_G2','CLT140_G3',
              'CLT160_E1','CLT160_E2','CLT160_E3','CLT160_G1','CLT160_G2','CLT160_G3',
              'CLT180_E1','CLT180_E2','CLT180_E3','CLT180_G1','CLT180_G2','CLT180_G3',
              'CLT200_E1','CLT200_E2','CLT200_E3','CLT200_G1','CLT200_G2','CLT200_G3',
              'CLT90_E1' ,'CLT90_E2' ,'CLT90_E3' ,'CLT90_G1' ,'CLT90_G2' ,'CLT90_G3' ,
              'Sf_CLT'   ,'Live_floor','Live_roof',
              'CLT100_E1_d','CLT100_E2_d','CLT100_E3_d','CLT100_G1_d','CLT100_G2_d','CLT100_G3_d',
              'CLT100_E1_v','CLT100_E2_v','CLT100_E3_v','CLT100_G1_v','CLT100_G2_v','CLT100_G3_v',
              'CLT120_E1_v','CLT120_E2_v','CLT120_E3_v','CLT120_G1_v','CLT120_G2_v','CLT120_G3_v',
              'CLT130_E1_d','CLT130_E2_d','CLT130_E3_d','CLT130_G1_d','CLT130_G2_d','CLT130_G3_d',
              'CLT160_E1_d','CLT160_E2_d','CLT160_E3_d','CLT160_G1_d','CLT160_G2_d','CLT160_G3_d',
              'Poisson']


"""
--------------------------------------------------------------------------------
RUNNING FUNCTIONS:
--------------------------------------------------------------------------------
"""
S = S1(inp1,F,C,MoverM)
S,x_label,index = remove_parameters(limit, S, x_label)
Oppdatert_mat, F, C, MoverM = update(inp1,S,F,C,MoverM,index)


"""
--------------------------------------------------------------------------------
PLOTS:
--------------------------------------------------------------------------------
"""
#3D bar plot
# setup the figure and axes
fig = plt.figure(figsize=(80, 30))
ax = fig.add_subplot(111, projection='3d')
# fake data
_x = np.arange(len(x_label))
_y = np.arange(fm)
_xx, _yy = np.meshgrid(_x, _y)
x, y = _xx.ravel(), _yy.ravel()
Srav = S.ravel()
top = np.abs(S).ravel()
bottom = np.zeros_like(top)
cs = np.zeros_like(top,dtype='<U11')
width = 0.9
depth = 0.9
for i in range(len(Srav)):
    if Srav[i] > 0:
        cs[i] = "red"
    else:
        cs[i] = "blue"
```

```
ax.bar3d(x, y, bottom, width, depth, top, color=cs, shade=True, alpha = 0.7,edgecolor='
    black')
ax.set_xticks(_x+0.5) # values
ax.set_xticklabels(x_label,rotation=50, horizontalalignment='right') # labels
ax.set_yticks(_y+0.5) # values
ax.set_yticklabels(['f1','f2','f3','MAC 1','MAC 2', 'MAC 3'],rotation=-20,
    horizontalalignment='left') # labels
ax.set_zticks([0,0.5,1]) # values
ax.set_zticklabels([0,0.5,1],horizontalalignment='left') # labels
ax.get_proj = lambda: np.dot(Axes3D.get_proj(ax), np.diag([1, 0.2, 0.2, 1]))
blue_proxy = plt.Rectangle((0, 0), 1, 1, fc="blue")
red_proxy = plt.Rectangle((0, 0), 1, 1, fc="red")
ax.legend([blue_proxy,red_proxy],['negative-value','positive-value'])
plt.show()

#2D plot av c value
x = [0,4,9,14,19,24]
_x = [1,5,10,15,20,25]
ymin = round(min(C[1:]),3);ymax =round(max(C[1:]),3)
y = [ymin,round((ymax-ymin)/2+ymin,3),ymax]
plt.plot(C[1:])
mintex = "Min C-value = "+str(round(min(C),5))
stex = "Initial C-value ="+str(round(C[0],5))
plt.plot(np.argmin(C),min(C),"o",label=stex,color="w")
plt.plot(np.argmin(C),min(C),"o",label=mintex)
plt.xticks(x,_x)
plt.yticks(y,y)
plt.xlabel("Run nr.")
plt.tight_layout()
plt.ylabel("C value")
plt.legend()
#tikzplotlib.save("mytikz.tex")
plt.show()
```

A.2 C2_Split

```
"""
--------------------------------------------------------------------------------
IMPORTS
--------------------------------------------------------------------------------
"""
import os
import sys
import comtypes.client
import numpy as np
"""
--------------------------------------------------------------------------------
COPY FROM SAP2000 OAPI
--------------------------------------------------------------------------------
"""
#set the following flag to True to attach to an existing instance of the program
#otherwise a new instance of the program will be started
AttachToInstance = True
#set the following flag to True to manually specify the path to SAP2000.exe
#this allows for a connection to a version of SAP2000 other than the latest
    installation
#otherwise the latest installed version of SAP2000 will be launched
SpecifyPath = False
```

```
#if the above flag is set to True, specify the path to SAP2000 below
#ProgramPath = 'C:\Program Files\Computers and Structures\SAP2000 22\SAP2000.exe'
#full path to the model
#set it to the desired path of your model
APIPath = 'C:\CSiAPIexample'
if not os.path.exists(APIPath):
        try:
            os.makedirs(APIPath)
        except OSError:
            pass
ModelPath = APIPath + os.sep + 'API_1-001.sdb'
#create API helper object
helper = comtypes.client.CreateObject('SAP2000v1.Helper')
helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
if AttachToInstance:
    #attach to a running instance of SAP2000
    try:
        #get the active SapObject
            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:
            #create an instance of the SAPObject from the latest installed SAP2000
            mySapObject = helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")
        except (OSError, comtypes.COMError):
            print("Cannot start a new instance of the program.")
            sys.exit(-1)
    #start SAP2000 application
    mySapObject.ApplicationStart()
#create SapModel object
SapModel = mySapObject.SapModel
"""
--------------------------------------------------------------------------------
FUNCTIONS:
--------------------------------------------------------------------------------
"""
def shear(indexListe):
    for j in indexListe:
        P = np.zeros((4,4));NumberPoints = 4;PropName="";
        NumberPoints,P[:,0],ret = SapModel.AreaObj.GetPoints(j, 4, [])
        PropName,ret = SapModel.AreaObj.GetProperty(j, PropName)
        ret = SapModel.AreaObj.Delete(j)
        for i in range(len(P)):
            P[i,1],P[i,2],P[i,3],ret = SapModel.PointObj.GetCoordCartesian(str(P[i,0])[0:-2], P[i,1], P[i,2], P[i,3])

        maxim = max(P[:,3])
        minim = min(P[:,3])

```

```
        for i in range(len(P)):
            if P[i,3] == maxim: #and P[i,3] != 24600:
                P[i,3] -= 90;
            elif P[i,3] == minim: #and P[i,3] != 1000:
                P[i,3] += 90;

        User = "S"+j
        ret = SapModel.AreaObj.AddByCoord(4, tuple(P[:,1]), tuple(P[:,2]), tuple(P
    [:,3]),"",PropName,User,"GLOBAL")

        for i in range(len(P)):
            P[i,1],P[i,2],P[i,3],ret = SapModel.PointObj.GetCoordCartesian(str(P[i,0])
    [0:-2], P[i,1], P[i,2], P[i,3])

        #LINK"
        Pny = np.zeros((4,4));NumberPoints = 4;PropName="";
        NumberPoints,Pny[:,0],ret = SapModel.AreaObj.GetPoints("S"+j, 4, [])
        for i in range(len(P)):
            ret = SapModel.PointObj.SetGroupAssign(str(Pny[i,0])[0:-2],"Links")
            ret = SapModel.PointObj.SetGroupAssign(str(P[i,0])[0:-2],"Links")

        Pny = np.zeros((4,4));NumberPoints = 4;PropName="";
        NumberPoints,Pny[:,0],ret = SapModel.AreaObj.GetPoints("S"+j, 4, [])
"""
-------------------------------------------------------------------------------
RUNNING FUNCTIONS:
-------------------------------------------------------------------------------
"""
#All areas
NumberNames = 0
MyName= []
nr,alle,ret = SapModel.AreaObj.GetNameList(NumberNames, MyName)

#Wals for spliting
ret = SapModel.SelectObj.All(True)
ret = SapModel.AreaObj.SetSelected("Shear_SXP", True, 1)

Wall_Split= []
for i in alle:
    Selected = True
    Selected,ret = SapModel.AreaObj.GetSelected(i, Selected)
    if Selected == True :
        Wall_Split.append(i)
"""
-------------------------------------------------------------------------------
RUNNING FUNCTIONS:
-------------------------------------------------------------------------------
"""

shear(Wall_Split)
```

A.3 C2_SA

```
"""
-------------------------------------------------------------------------------
IMPORTS
-------------------------------------------------------------------------------
"""
```

```
import os
import sys
import comtypes.client
import numpy as np
import matplotlib.pyplot as plt
import tikzplotlib
from mpl_toolkits.mplot3d import Axes3D
from math import atan
"""
--------------------------------------------------------------------------------
COPY FROM SAP2000 OAPI
--------------------------------------------------------------------------------
"""
#set the following flag to True to attach to an existing instance of the program
#otherwise a new instance of the program will be started
AttachToInstance = True
#set the following flag to True to manually specify the path to SAP2000.exe
#this allows for a connection to a version of SAP2000 other than the latest
    installation
#otherwise the latest installed version of SAP2000 will be launched
SpecifyPath = False
#if the above flag is set to True, specify the path to SAP2000 below
#ProgramPath = 'C:\Program Files\Computers and Structures\SAP2000 22\SAP2000.exe'
#full path to the model
#set it to the desired path of your model
APIPath = 'C:\CSiAPIexample'
if not os.path.exists(APIPath):
        try:
            os.makedirs(APIPath)
        except OSError:
            pass
ModelPath = APIPath + os.sep + 'API_1-001.sdb'
#create API helper object
helper = comtypes.client.CreateObject('SAP2000v1.Helper')
helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
if AttachToInstance:
    #attach to a running instance of SAP2000
    try:
        #get the active SapObject
            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:
            #create an instance of the SAPObject from the latest installed SAP2000
            mySapObject = helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")
        except (OSError, comtypes.COMError):
            print("Cannot start a new instance of the program.")
            sys.exit(-1)
    #start SAP2000 application
```

```
    mySapObject.ApplicationStart()
#create SapModel object
SapModel = mySapObject.SapModel
"""
-----------------------------------------------------------------------------
FUNCTIONS:
-----------------------------------------------------------------------------
"""
def ABProp_run(name,ABT,ABV):
    #Set links in sap anglebracets
    MyDOF       = (True,True,True,False,False,False)
    MyFixed     = (False,False,False,False,False,False)
    MyKe        = (ABT,ABV,trans3,rot1,rot2,rot3) # effective stiffness
    MyCe        = (0,0,0,0,0,0)  # effective damping
    SapModel.PropLink.SetLinear(name, MyDOF, MyFixed, MyKe, MyCe, lk, lk)
    SapModel.LinkObj.SetProperty(name,name,1)

def HDProp_run(name,HDT):
    #Set links in sap holddowns
    MyDOF       = (True,False,True,False,False,False)
    MyFixed     = (False,False,False,False,False,False)
    MyKe        = (HDT,trans2,trans3,rot1,rot2,rot3) # effective stiffness
    MyCe        = (0,0,0,0,0,0)  # effective damping
    SapModel.PropLink.SetLinear(name, MyDOF, MyFixed, MyKe, MyCe, lk, lk)
    SapModel.LinkObj.SetProperty(name,name,1)

def ABT(n):
    nexp = 8
    kel = ko*2.98 #[kN/mm]
    #kpl = 0.50 #[kN/mm]

    k = n * (kel/nexp)
    return k

def ABV(n):
    nexp = 8
    kel  = ko*1.10 #[kN/mm]
    #kpl = 0.18 #[kN/mm]
    k = n * (kel/nexp)
    return k

def MAC(Mc,Mm):
    #Calculating MAC
    A = np.dot(np.transpose(Mm),Mc)
    B = np.absolute(np.dot(A,A))
    C = np.dot(np.transpose(Mc),Mc)
    D = np.dot(np.transpose(Mm),Mm)
    E = np.dot(C,D)
    macut = np.divide(B,E)
    return macut

def NMD(Mc,Mm):
    #Calculating NMD
    macut = MAC(Mc,Mm)
    nmdut = ((1 - macut )/macut)**0.5
    return nmdut

def run_model(mat):
    SapModel.SetModelIsLocked(False)
```

```
    #Entering parameter values ----------------------------------------------
    SapModel.AreaObj.SetLoadUniform("ALL","DEAD",0,10, True, "GLOBAL",1)
    SapModel.SetPresentUnits(6)
    SapModel.AreaObj.SetLoadUniform("Aroof","DEAD",mat[50,0],10, True, "GLOBAL",1)
    SapModel.AreaObj.SetLoadUniform("Aint","DEAD",mat[49,0],10, True, "GLOBAL",1)
    SapModel.PropMaterial.SetWeightAndMass("CLT_100", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_120", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_130", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_140", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_160", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_180", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_200", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_100_Dor", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_100_Vindu", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_120_Vindu", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_130_Dor", 1, mat[48,0])
    SapModel.PropMaterial.SetWeightAndMass("CLT_160_Dor", 1, mat[48,0])
    SapModel.SetPresentUnits(5)
    SapModel.PropMaterial.SetMPOrthotropic("CLT_100", [mat[0,0], mat[1,0], mat[2,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[3,0], mat[4,0], mat[5,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_120", [mat[6,0], mat[7,0], mat[8,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[9,0], mat[10,0], mat[11,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_130", [mat[12,0], mat[13,0], mat[14,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[15,0], mat[16,0], mat[17,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_140", [mat[18,0], mat[19,0], mat[20,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[21,0], mat[22,0], mat[23,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_160", [mat[24,0], mat[25,0], mat[26,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[27,0], mat[28,0], mat[29,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_180", [mat[30,0], mat[31,0], mat[32,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[33,0], mat[34,0], mat[35,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_200", [mat[36,0], mat[37,0], mat[38,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[39,0], mat[40,0], mat[41,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_90" , [mat[42,0], mat[43,0], mat[44,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[45,0], mat[46,0], mat[47,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Dor" , [mat[51,0], mat[52,0], mat[53,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[54,0], mat[55,0], mat[56,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Vindu" , [mat[57,0], mat[58,0], mat[59,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[60,0], mat[61,0], mat[62,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_120_Vindu" , [mat[63,0], mat[64,0], mat[65,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[66,0], mat[67,0], mat[68,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_130_Dor" , [mat[69,0], mat[70,0], mat[71,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[72,0], mat[73,0], mat[74,0]] )
    SapModel.PropMaterial.SetMPOrthotropic("CLT_160_Dor" , [mat[75,0], mat[76,0], mat[77,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[78,0], mat[79,0], mat[80,0]] )
    ABProp_run("AB0",mat[82,0],mat[83,0])
    ABProp_run("AB1",mat[84,0],mat[85,0])
    ABProp_run("AB2",mat[86,0],mat[87,0])
    ABProp_run("AB3",mat[88,0],mat[89,0])
    ABProp_run("AB4",mat[90,0],mat[91,0])
    ABProp_run("AB5",mat[92,0],mat[93,0])
    ABProp_run("AB6",mat[94,0],mat[95,0])
    ABProp_run("AB7",mat[96,0],mat[97,0])
    ABProp_run("AB8",mat[98,0],mat[99,0])
    ABProp_run("AB9",mat[100,0],mat[101,0])
```

```
    HDProp_run("HDx0",mat[102,0])
    HDProp_run("HDx1",mat[103,0])
    HDProp_run("HDx2",mat[104,0])
    HDProp_run("HDx3",mat[105,0])
    HDProp_run("HDx4",mat[106,0])
    HDProp_run("HDx5",mat[107,0])

    #Run model--------------------------------------------------------------
    ret = SapModel.Analyze.RunAnalysis()

    #Calculate eigenfrequencies-------------------------------------------------
    NumberResults = 0 #Ukjent
    Period = []
    Frequency = []
    CircFreq = []
    EigenValue = []
    StepNum = []
    Modal = []
    Mode = []
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
    [NumberResults,Modal,Mode,StepNum,Period, Frequency, CircFreq, EigenValue,ret] = SapModel.Results.ModalPeriod(NumberResults, "Modal", "Mode", StepNum, Period, Frequency, CircFreq,EigenValue)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
    GroupElm = 0
    NumberResults = 0
    Obj = []
    Elm = []
    LoadCase = ["Modal"]
    StepType = ["Mode"]
    StepNum = []
    U1 = []
    U2 = []
    U3 = []
    R1 = []
    R2 = []
    R3 = []
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,R3,ret] = SapModel.Results.JointDispl("633", GroupElm, NumberResults, Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()
    ind = [0,1,2]
    U1 = np.absolute(U1)
    U2 = np.absolute(U2)
    indModeX = np.argmax(U1)
    indModeY = np.argmax(U2)
    ind.remove(indModeX)
    ind.remove(indModeY)
    indModeZ = int(ind[0])

    #Analytivcal mode shape----------------------------------------------------
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_427,U2_427,U3,R1,R2,R3,ret] = SapModel.Results.JointDispl("427", GroupElm, NumberResults, Obj, Elm, LoadCase, StepType, StepNum, U1, U2, U3, R1, R2, R3)
```

```
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_434,U2_434,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("434", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_554,U2_554,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("554", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_579,U2_579,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("579", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    GroupElm = 0;NumberResults = 0;Obj = [];Elm = [];LoadCase = Modal;StepType = Mode;
    StepNum = [];R1 = [];R2 = [];R3 = [];U1 = [];U2 = [];U3 = []
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1_586,U2_586,U3,R1,R2,R3,ret] =
    SapModel.Results.JointDispl("586", GroupElm, NumberResults, Obj, Elm, LoadCase,
    StepType, StepNum, U1, U2, U3, R1, R2, R3)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    ModeA = np.ones((3,3))

    Mode17 = (abs(U2_554[indModeY]) + abs(U2_579[indModeY]) + abs(U2_586[indModeY]))/3
    Mode15 = (abs(U2_427[indModeY]) + abs(U2_434[indModeY]))/2

    ModeA[1,0] = Mode15/Mode17

    Mode27 = (abs(U1_554[indModeX]) + abs(U1_579[indModeX]) + abs(U1_586[indModeX]))/3
    Mode25 = (abs(U1_427[indModeX]) + abs(U1_434[indModeX]))/2

    ModeA[1,1] = Mode25/Mode27

    Mode37 = (abs(U2_579[indModeZ]) + abs(U2_586[indModeZ]))/2
    Mode35 = (abs(U2_427[indModeZ]) + abs(U2_434[indModeZ]))/2

    L = 23.06/2

    rot7 = atan(Mode37/L)
    rot5 = atan(Mode35/L)

    ModeA[1,2] = rot5/rot7
    ModeA[2,0] = 0;ModeA[2,1] = 0;ModeA[2,2] = 0;
    mac1 = 1 - MAC(ModeA[:,0],ModeE[:,0])
    mac2 = 1 - MAC(ModeA[:,1],ModeE[:,1])
    mac3 = 1 - MAC(ModeA[:,2],ModeE[:,2])

    nmd1 = NMD(ModeA[:,0],ModeE[:,0])
    nmd2 = NMD(ModeA[:,1],ModeE[:,1])
    nmd3 = NMD(ModeA[:,2],ModeE[:,2])

    #MAC or NMD as output-----------------------------------------------------
    mod1 = mac1;mod2 = mac2;mod3 = mac3;
    #mod1 = nmd1;mod2 = nmd2;mod3 = nmd3;

    #Calculate C-value ------------------------------------------------------
    C =  ((abs(f1e-Frequency[indModeY])/f1e) + (abs(f2e-Frequency[indModeX])/f2e) + (
    abs(f3e-Frequency[indModeZ])/f3e))/3 + (mod1 + mod2 + mod3)/3
```

```

    f1 = Frequency[indModeY]
    f2 = Frequency[indModeX]
    f3 = Frequency[indModeZ]

    #Ractionforce vs. applied force ------------------------------------------

    Ubrukelig1 = []
    Ubrukelig2 = []
    Ubrukelig3 = []
    Name1 = []
    Name2 = []
    ItemTypeElm = 3
    NumberResults = 0
    Obj = []
    Elm = []
    LoadCase = []
    StepType = []
    StepNum = []
    F1 = []
    F2 = []
    F3 = []
    M1 = []
    M2 = []
    M3 = []

    ret = SapModel.SelectObj.All(True)
    ret = SapModel.SelectObj.CoordinateRange(-10000000, 10000000, -100000000, 10000000,
     999, 1001,False,"Global" ,True ,True, False, True,False, False)
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("DEAD")
    [Ubrukelig1, Name1, Name2, LoadCase, Steptype,Ubrukelig2, F1, F2, F3, M1, M2, M3,
    Ubrukelig3] = SapModel.Results.JointReact("ALL", ItemTypeElm, NumberResults, Obj,
    Elm, "DEAD", StepType, StepNum, F1, F2, F3, M1, M2, M3)
    ret = SapModel.SelectObj.All(True)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    OrgAF = 327.1
    OrgAR = 340.5
    OrgL  = 1039.9573

    OrgM = 8*inp1[49,0]*OrgAF + inp1[50,0]*OrgAR + OrgL*inp1[48,0]
    SapM = sum(F3)

    OrgSap = OrgM/SapM

    return C,f1,f2,f3,mod1,mod2,mod3,ModeA,OrgSap


def S1(mat,F,C,MoverM):
    print('Calculating the sensitivity matrix...')
    f = np.zeros((n+1,6))      # Empty list; frequency
    moverm = np.zeros(n+1)     # Empty list; frequency
    cmat = np.copy(mat)        # Copy of input parameter matrix

    #Run model with input reference values:
    C[0],F[0,0],F[0,1],F[0,2],F[0,3],F[0,4],F[0,5],kast,MoverM[0] = run_model(mat)
    f[0,0] = F[0,0]; f[0,1] = F[0,1]; f[0,2] = F[0,2]; f[0,3] = F[0,3]; f[0,4] = F
    [0,4]; f[0,5] = F[0,5];moverm[0]=MoverM[0]
    print('Run with initial parameter values: ')
```

```
    print("C = ",round(C[0],5))
    print("f1 = ",round(F[0,0],5))
    print("f2 = ",round(F[0,1],5))
    print("f3 = ",round(F[0,2],5))
    print("MCRIT 1 = ",round(F[0,3],5))
    print("MCRIT 2 = ",round(F[0,4],5))
    print("MCRIT 3 = ",round(F[0,5],5))

    for i in range(n):
        delta_P = prosent * ((mat[82+i,2]-mat[82+i,1]))

        mat[82+i,0] +=  delta_P

        #Run FE-model with perturbed parameter values
        C_i,f[i+1,0],f[i+1,1],f[i+1,2],f[i+1,3],f[i+1,4],f[i+1,5],kast,moverm[i+1] =
    run_model(mat)
        print(' ')
        print('Run no.: ',i+1)
        print('Perturbation of parameter: ',x_label[82+i])
        for ii in range(fm):
            if ii>=3:
                delta_R = (f[i+1,ii] - f[0,ii])*ScaMode
            else:
                delta_R = (f[i+1,ii] - f[0,ii])

            #S[ii,i] = (delta_R/delta_P)                       #Absolutt
            S[ii,i] = (delta_R/delta_P)*(mat[i,0]/f[i+1,ii]) #Normalisert
            print('R',ii+1,':  dR =',round(delta_R,4),'  ; dP = ',round(delta_P,4),' ;
    dR/dP = ',round(S[ii,i],9))
        mat[82+i,0]  = cmat[82+i,0]
    return S

"""
--------------------------------------------------------------------------------
GENERELLE VARIABLER:
--------------------------------------------------------------------------------
"""
#Pertubasjon for the sensitivity matrix calculation:
prosent = 0.1

#Scaling of the modes hapes
ScaMode = 241.81466810617732/2

#Matrices for storing answers:
#fm = 3 # Updates on only frequencies
fm = 6  # Updates on frequencies and mode shapes
n  = 82 # Number of parmeters
S  = np.zeros((fm,n))
C  = [0]
MoverM = [0]
F = [0]
"""
--------------------------------------------------------------------------------
EXPERIMENTAL DATA
--------------------------------------------------------------------------------
"""
#Experimental frequencies:
f1e = 1.913
f2e = 2.414
```

```
f3e = 2.693

#Experimental mode shapes:
ModeE = np.array([[ 1.        ,  1.        ,  1.        ],
                  [ 0.70737374,  0.73400209,  0.78296554],
                  [ 0         ,  0         ,  0         ]])
"""
-------------------------------------------------------------------------------
DEFAULT MODEL SETTINGS:
-------------------------------------------------------------------------------
"""
SapModel.SetModelIsLocked(False)
SapModel.SetPresentUnits(5)
#kN_mm_C = 5
#kN_m_C  = 6
#N_mm_C  = 9
#N_m_C   = 10
SapModel.AreaObj.SetMass("ALL",0, True, 1)
SapModel.AreaObj.SetAutoMesh("ALL", 2, 0, 0, 2000, 2000,False, False, False, False, False, 0 ,False, False, False ,"ALL" , 0,False, 1)
SapModel.SourceMass.SetMassSource("MyMassSource", False, False, True, True, 1, ["DEAD"], [1])

"""
-------------------------------------------------------------------------------
CALCULATION OF PARAMETER MATRIX:  Connections:
-------------------------------------------------------------------------------
"""
ko = 1
HDT = ko*2.65 #[kN/mm]

AB0_T = ABT(5 );AB0_V = ABV(5 )
AB1_T = ABT(5 );AB1_V = ABV(5 )
AB2_T = ABT(8 );AB2_V = ABV(8 )
AB3_T = ABT(12);AB3_V = ABV(12)
AB4_T = ABT(13);AB4_V = ABV(13)
AB5_T = ABT(18);AB5_V = ABV(18)
AB6_T = ABT(25);AB6_V = ABV(25)
AB7_T = ABT(35);AB7_V = ABV(35)
AB8_T = ABT(36);AB8_V = ABV(36)
AB9_T = ABT(72);AB9_V = ABV(72)

HD0_T = HDT*1
HD1_T = HDT*1
HD2_T = HDT*2
HD3_T = HDT*3
HD4_T = HDT*4
HD5_T = HDT*5

kdown = 0.5
kup = 1.5

AB0 = [[AB0_T,AB0_T*kdown,AB0_T*kup],
       [AB0_V,AB0_V*kdown,AB0_V*kup]]

AB1 = [[AB1_T,AB1_T*kdown,AB1_T*kup],
       [AB1_V,AB1_V*kdown,AB1_V*kup]]

AB2 = [[AB2_T,AB2_T*kdown,AB2_T*kup],
```

```
       [AB2_V,AB2_V*kdown,AB2_V*kup]]

AB3 = [[AB3_T,AB3_T*kdown,AB3_T*kup],
       [AB3_V,AB3_V*kdown,AB3_V*kup]]

AB4 = [[AB4_T,AB4_T*kdown,AB4_T*kup],
       [AB4_V,AB4_V*kdown,AB4_V*kup]]

AB5 = [[AB5_T,AB5_T*kdown,AB5_T*kup],
       [AB5_V,AB5_V*kdown,AB5_V*kup]]

AB6 = [[AB6_T,AB6_T*kdown,AB6_T*kup],
       [AB6_V,AB6_V*kdown,AB6_V*kup]]

AB7 = [[AB7_T,AB7_T*kdown,AB7_T*kup],
       [AB7_V,AB7_V*kdown,AB7_V*kup]]

AB8 = [[AB8_T,AB8_T*kdown,AB8_T*kup],
       [AB8_V,AB8_V*kdown,AB8_V*kup]]

AB9 = [[AB9_T,AB9_T*kdown,AB9_T*kup],
       [AB9_V,AB9_V*kdown,AB9_V*kup]]

HD0 = [HD0_T,HD0_T*kdown,HD0_T*kup]
HD1 = [HD1_T,HD1_T*kdown,HD1_T*kup]
HD2 = [HD2_T,HD2_T*kdown,HD2_T*kup]
HD3 = [HD3_T,HD3_T*kdown,HD3_T*kup]
HD4 = [HD4_T,HD4_T*kdown,HD4_T*kup]
HD5 = [HD5_T,HD5_T*kdown,HD5_T*kup]

rot1 = 0
rot2 = 0
rot3 = 0
trans2 = 0
trans3 = 10000
lk = 0

"""
-------------------------------------------------------------------------------
PARAMETER MATRIX:
-------------------------------------------------------------------------------
"""
inp1 = np.array([[ 6.74794564,  3.374     , 10.122     ],
           [ 4.62206714,  2.311     ,  6.933     ],
           [ 0.3       ,  0.15      ,  0.45      ],
           [ 0.46911976,  0.1687    ,  0.5061    ],
           [ 0.03374   ,  0.01687   ,  0.05061   ],
           [ 0.03374   ,  0.01687   ,  0.05061   ],
           [ 7.45694048,  3.72833333, 11.185     ],
           [ 3.91292597,  1.95666667,  5.87      ],
           [ 0.3       ,  0.15      ,  0.45      ],
           [ 0.39543461,  0.18641667,  0.55925   ],
           [ 0.03728333,  0.01864167,  0.055925  ],
           [ 0.03728333,  0.01864167,  0.055925  ],
           [ 7.72923077,  3.86461538, 11.59384615],
           [ 3.64076923,  1.82038462,  5.46115385],
           [ 0.3       ,  0.15      ,  0.45      ],
           [ 0.38646154,  0.19323077,  0.57969231],
           [ 0.03864615,  0.01932308,  0.05796923],
```

```
            [ 0.03864615,  0.01932308,  0.05796923],
            [ 7.95790706,  3.98142857, 11.94428571],
            [ 3.40830835,  1.70357143,  5.11071429],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.46413044,  0.19907143,  0.59721429],
            [ 0.03981429,  0.01990714,  0.05972143],
            [ 0.03981429,  0.01990714,  0.05972143],
            [ 8.33626101,  4.17125   , 12.51375   ],
            [ 3.02863437,  1.51375   ,  4.54125   ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.2085625 ,  0.2085625 ,  0.6256875 ],
            [ 0.0417125 ,  0.02085625,  0.06256875],
            [ 0.0417125 ,  0.02085625,  0.06256875],
            [ 7.46060274,  3.72833333, 11.185     ],
            [ 3.91443731,  1.95666667,  5.87      ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.44055026,  0.18641667,  0.55925   ],
            [ 0.03728333,  0.01864167,  0.055925  ],
            [ 0.03728333,  0.01864167,  0.055925  ],
            [ 6.7500431 ,  3.374     , 10.122     ],
            [ 4.62222073,  2.311     ,  6.933     ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.3503614 ,  0.1687    ,  0.5061    ],
            [ 0.03374   ,  0.01687   ,  0.05061   ],
            [ 0.03374   ,  0.01687   ,  0.05061   ],
            [ 7.4574407 ,  3.72833333, 11.185     ],
            [ 3.91346507,  2.311     ,  5.87      ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.38303263,  0.1687    ,  0.55925   ],
            [ 0.03728333,  0.01687   ,  0.055925  ],
            [ 0.03728333,  0.01687   ,  0.055925  ],
            [ 4.54235651,  2.25      ,  6.75      ],
            [ 2.66726807,  1.275     ,  3.825     ],
            [ 0.19336342,  0.084     ,  0.252     ],
            [ 5.06587119,  2.5305    ,  7.5915    ],
            [ 3.46555758,  1.73325   ,  5.19975   ],
            [ 0.225     ,  0.1125    ,  0.3375    ],
            [ 0.05061   ,  0.05061   ,  0.45549   ],
            [ 0.025305  ,  0.0126525 ,  0.0379575 ],
            [ 0.025305  ,  0.0126525 ,  0.0379575 ],
            [ 5.19628085,  2.59798   ,  7.79394   ],
            [ 3.55896399,  1.77947   ,  5.33841   ],
            [ 0.231     ,  0.1155    ,  0.3465    ],
            [ 0.20338306,  0.0519596 ,  0.4676364 ],
            [ 0.0259798 ,  0.0129899 ,  0.0389697 ],
            [ 0.0259798 ,  0.0129899 ,  0.0389697 ],
            [ 5.75462719,  2.87081667,  8.61245   ],
            [ 3.01233337,  1.50663333,  4.5199    ],
            [ 0.231     ,  0.1155    ,  0.3465    ],
            [ 0.30001279,  0.05741633,  0.516747  ],
            [ 0.02870817,  0.01435408,  0.04306225],
            [ 0.02870817,  0.01435408,  0.04306225],
            [ 5.8042818 ,  2.89846154,  8.69538462],
            [ 2.72898192,  1.36528846,  4.09586538],
            [ 0.225     ,  0.1125    ,  0.3375    ],
            [ 0.05796923,  0.05796923,  0.52172308],
            [ 0.02898462,  0.01449231,  0.04347692],
            [ 0.02898462,  0.01449231,  0.04347692],
            [ 6.25802976,  3.1284375 ,  9.3853125 ],
```

```
            [ 2.27042119,  1.1353125 ,  3.4059375 ],
            [ 0.225     ,  0.1125    ,  0.3375    ],
            [ 0.06256875,  0.06256875,  0.56311875],
            [ 0.03128438,  0.01564219,  0.04692656],
            [ 0.03128438,  0.01564219,  0.04692656],
            [ 0.43222457,  0.15      ,  0.45      ]])

inp1 = np.vstack((inp1,AB0,AB1,AB2,AB3,AB4,AB5,AB6,AB7,AB8,AB9,HD0,HD1,HD2,HD3,HD4,HD5)
    )
x_label = ['CLT100_E1','CLT100_E2','CLT100_E3','CLT100_G1','CLT100_G2','CLT100_G3',
             'CLT120_E1','CLT120_E2','CLT120_E3','CLT120_G1','CLT120_G2','CLT120_G3',
             'CLT130_E1','CLT130_E2','CLT130_E3','CLT130_G1','CLT130_G2','CLT130_G3',
             'CLT140_E1','CLT140_E2','CLT140_E3','CLT140_G1','CLT140_G2','CLT140_G3',
             'CLT160_E1','CLT160_E2','CLT160_E3','CLT160_G1','CLT160_G2','CLT160_G3',
             'CLT180_E1','CLT180_E2','CLT180_E3','CLT180_G1','CLT180_G2','CLT180_G3',
             'CLT200_E1','CLT200_E2','CLT200_E3','CLT200_G1','CLT200_G2','CLT200_G3',
             'CLT90_E1' ,'CLT90_E2' ,'CLT90_E3' ,'CLT90_G1' ,'CLT90_G2' ,'CLT90_G3' ,
             'Sf_CLT'   ,'Live_floor','Live_roof',
             'CLT100_E1_d','CLT100_E2_d','CLT100_E3_d','CLT100_G1_d','CLT100_G2_d','
    CLT100_G3_d',
             'CLT100_E1_v','CLT100_E2_v','CLT100_E3_v','CLT100_G1_v','CLT100_G2_v','
    CLT100_G3_v',
             'CLT120_E1_v','CLT120_E2_v','CLT120_E3_v','CLT120_G1_v','CLT120_G2_v','
    CLT120_G3_v',
             'CLT130_E1_d','CLT130_E2_d','CLT130_E3_d','CLT130_G1_d','CLT130_G2_d','
    CLT130_G3_d',
             'CLT160_E1_d','CLT160_E2_d','CLT160_E3_d','CLT160_G1_d','CLT160_G2_d','
    CLT160_G3_d',
             'Poisson',
             'AB0_T','AB0_V','AB1_T','AB1_V','AB2_T','AB2_V','AB3_T','AB3_V','AB4_T','
    AB4_V',
             'AB5_T','AB5_V','AB6_T','AB6_V','AB7_T','AB7_V','AB8_T','AB8_V','AB9_T','
    AB9_V',
             'HD0','HD1','HD2','HD3','HD4','HD5']

"""
-------------------------------------------------------------------------------
RUNNING FUNCTIONS:
-------------------------------------------------------------------------------
"""
S = S1(inp1,F,C,MoverM)

"""
-------------------------------------------------------------------------------
PLOTS:
-------------------------------------------------------------------------------
"""
#3d plot:
# setup the figure and axes
x_label = x_label[82::]
x_label = x_label[2:20] + x_label[21:26]
fjerna = [0,0,18]
for i in fjerna:
    S = np.delete(S,i,1)
fig = plt.figure(figsize=(80, 30))
ax = fig.add_subplot(111, projection='3d')
# fake data
_x = np.arange(len(x_label))
_y = np.arange(fm)
```

```
_xx, _yy = np.meshgrid(_x, _y)
x, y = _xx.ravel(), _yy.ravel()
Srav = S.ravel()
top = np.abs(S).ravel()
bottom = np.zeros_like(top)
cs = np.zeros_like(top,dtype='<U11')
width = 0.9
depth = 0.9
for i in range(len(Srav)):
    if Srav[i] > 0:
        cs[i] = "red"
    else:
        cs[i] = "blue"
csmin = ['r']*(len(x_label)) + ['g']*(len(x_label)) + ['b']*(len(x_label)) + ['y']*(len
    (x_label)) + ['c']*(len(x_label)) + ['w']*(len(x_label))
ax.bar3d(x, y, bottom, width, depth, top, color=cs, shade=True, alpha = 0.7,edgecolor='
    black')
ax.set_xticks(_x+0.5) # values
ax.set_xticklabels(x_label,rotation=50, horizontalalignment='right') # labels
ax.set_yticks(_y+0.5) # values
ax.set_yticklabels(['f1','f2','f3','MAC 1','MAC 2', 'MAC 3'],rotation=-20,
    horizontalalignment='left') # labels
# OUR ONE LINER ADDED HERE:
ax.get_proj = lambda: np.dot(Axes3D.get_proj(ax), np.diag([1, 0.2, 0.2, 1]))
blue_proxy = plt.Rectangle((0, 0), 1, 1, fc="blue")
red_proxy = plt.Rectangle((0, 0), 1, 1, fc="red")
ax.legend([blue_proxy,red_proxy],['negative-value','positive-value'])
plt.show()
```

A.4 C3_Geo

```
"""
-------------------------------------------------------------------------------
IMPORTS
-------------------------------------------------------------------------------
"""
import os
import sys
import comtypes.client
import numpy as np
import matplotlib.pyplot as plt
import tikzplotlib
from mpl_toolkits.mplot3d import Axes3D
from math import atan
"""
-------------------------------------------------------------------------------
FUNCTIONS:
-------------------------------------------------------------------------------
"""
def GridPoints(x,y,z,Unused):
    #Makes points in all spesified grid loations.
    for i in range(len(x[:,0])):
        for ii in range(len(y[:,0])):
            for iii in range(len(z[:,0])):

                u1 = float(x[i,0]);
                u2 = float(y[ii,0]);
                u3 = float(z[iii,0]);
```

```
                Name="";
                MyName= x[i,1]+y[ii,1]+z[iii,1]
                if MyName[:-1] not in Unused:
                    Name,ret = SapModel.PointObj.AddCartesian(u1, u2, u3, Name, MyName)
                    if iii == 0:
                        Value = (True,True,True,True,True,True)
                        ret = SapModel.PointObj.setRestraint(Name,Value,0)
    ret = SapModel.View.RefreshView(0, False)

def WallX(m,x,y,z,wv):
    #Makes walls in indicated locations.
    for i in range(len(z[:,0])-1):
        for ii in range(len(y[:,0])):
            for iii in range(len(x[:,0])-1):
                if m[i,ii,iii] != "":
                    Point = list(np.zeros(4).astype('str'))
                    Point[0] = x[iii,1]  +y[ii,1]+z[i,1]
                    Point[1] = x[iii+1,1]+y[ii,1]+z[i,1]
                    Point[2] = x[iii+1,1]+y[ii,1]+z[i+1,1]
                    Point[3] = x[iii,1]  +y[ii,1]+z[i+1,1]
                    pname = m[i,ii,iii]
                    Name = x[iii,1]+"-"+x[iii+1,1]+"_"+y[ii,1]+"-"+y[ii,1]+"_"+z[i,1]+"-"+z[i+1,1]
                    ret = SapModel.AreaObj.AddByPoint(4, Point,Name,pname,Name)
                    if ret[2] == 0:
                        ret = SapModel.AreaObj.SetLocalAxes(Name, 90)
                        l = abs(float(x[iii,0])-float(x[iii+1,0]))
                        h = abs(float(z[i,0])  -float(z[i+1,0]))
                        if m[i,ii,iii][4:6] == '90':
                            t = 0.09;
                        else:
                            t = float('0.'+m[i,ii,iii][4:7])
                        v = l*h*t
                        ind = list(wv[:,0]).index(m[i,ii,iii]);
                        wv[ind,1] = str(v + float(wv[ind,1]))
    ret = SapModel.View.RefreshView(0, False)

def WallY(m,x,y,z,wv):
    #Makes walls in indicated locations.
    for i in range(len(z[:,0])-1):
        for ii in range(len(x[:,0])):
            for iii in range(len(y[:,0])-1):
                if m[i,ii,iii] != "":
                    Point = list(np.zeros(4).astype('str'))
                    Point[0] = x[ii,1]+y[iii,1]  +z[i,1]
                    Point[1] = x[ii,1]+y[iii+1,1]+z[i,1]
                    Point[2] = x[ii,1]+y[iii+1,1]+z[i+1,1]
                    Point[3] = x[ii,1]+y[iii,1]  +z[i+1,1]
                    pname = m[i,ii,iii]
                    Name = x[ii,1]+"-"+x[ii,1]+"_"+y[iii,1]+"-"+y[iii+1,1]+"_"+z[i,1]+"-"+z[i+1,1]
                    ret = SapModel.AreaObj.AddByPoint(4, Point,Name,pname,Name)
                    if ret[2] == 0:
                        ret = SapModel.AreaObj.SetLocalAxes(Name, 90)
                        l = abs(float(y[iii,0])-float(y[iii+1,0]))
                        h = abs(float(z[i,0])  -float(z[i+1,0]))
                        if m[i,ii,iii][4:6] == '90':
                            t = 0.09;
                        else:
```

```
                            t = float('0.'+m[i,ii,iii][4:7])
                        v = l*h*t
                        ind = list(wv[:,0]).index(m[i,ii,iii]);
                        wv[ind,1] = str(v + float(wv[ind,1]))
    ret = SapModel.View.RefreshView(0, False)

def Dekke(x,y,z,Undekke,wv):
    a = np.zeros(len(z))
    for i in range(len(z[:,0])):
        for ii in range(len(x[:,0])-1):
            for iii in range(len(y[:,0])-1):
                Point = list(np.zeros(4).astype('str'))
                Point[0] = x[ii,1]+y[iii,1]+z[i,1]
                Point[1] = x[ii+1,1]+y[iii,1]+z[i,1]
                Point[2] = x[ii+1,1]+y[iii+1,1]+z[i,1]
                Point[3] = x[ii,1]+y[iii+1,1]+z[i,1]
                Name = x[ii,1]+"-"+x[ii+1,1]+"_"+y[iii,1]+"-"+y[iii+1,1]+"_"+z[i,1]+"-"+z[i,1]

                if Point[0][:2] not in Undekke:
                    ret = SapModel.AreaObj.SetLocalAxes(Name, 90)
                    if i == len(z[:,0])-1:
                        Group = "Aroof"
                        Type = "CLT_200"
                    else:
                        Group = "Aint"
                        Type = "CLT_180"
                    ret = SapModel.AreaObj.AddByPoint(4, Point,Name,Type,Name)
                    ret = SapModel.AreaObj.SetGroupAssign(Name,Group,False,0)
                    ret = SapModel.AreaObj.SetLocalAxes(Name, 90)
                    if ret == 0:
                        l = abs(float(x[ii,0])-float(x[ii+1,0]))
                        h = abs(float(y[iii,0])  -float(y[iii+1,0]))
                        t = float('0.'+Type[4:7])
                        v = l*h*t
                        ind = list(wv[:,0]).index(Type);
                        wv[ind,1] = str(v + float(wv[ind,1]))
                        a[i] += l*h

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

def stiff_center():
    SapModel.SetModelIsLocked(False)
    ret = SapModel.LoadPatterns.Add('X', 1, 0, True)
    ret = SapModel.LoadPatterns.Add('Y', 1, 0, True)
    ret = SapModel.LoadPatterns.Add('Z', 1, 0, True)

    name = "1A9"

    ret = SapModel.PointObj.SetLoadForce(name, 'X',[1,0,0,0,0,0])
    ret = SapModel.PointObj.SetLoadForce(name, 'Y',[0,1,0,0,0,0])
    ret = SapModel.PointObj.SetLoadForce(name, 'Z',[0,0,0,0,0,1])

def run_model(mat):
    SapModel.SetModelIsLocked(False)
    #Legge inn parameterverdier ----------------------------------------------

    SapModel.SetPresentUnits(5)
```

```
SapModel.AreaObj.SetLoadUniform("ALL","DEAD",0,10, True, "GLOBAL",1)
SapModel.AreaObj.SetAutoMesh("ALL", 2, 0, 0, 2000, 2000,False, False, False, False,
 False, 0 ,False, False, False ,"ALL" , 0,False, 1)

SapModel.SetPresentUnits(6)

SapModel.AreaObj.SetLoadUniform("Aroof","DEAD",mat[50,0],10, True, "GLOBAL",1)
                         #Masse
SapModel.AreaObj.SetLoadUniform("Aint","DEAD",mat[49,0],10, True, "GLOBAL",1)
SapModel.PropMaterial.SetWeightAndMass("CLT_100", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_120", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_130", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_140", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_160", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_180", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_200", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_90", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_100_Dor", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_100_Vindu", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_120_Vindu", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_130_Dor", 1, mat[48,0])
SapModel.PropMaterial.SetWeightAndMass("CLT_160_Dor", 1, mat[48,0])

SapModel.SetPresentUnits(5)

SapModel.PropMaterial.SetMPOrthotropic("CLT_100", [mat[0,0], mat[1,0], mat[2,0]] ,[
mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[3,0], mat[4,0], mat[5,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_120", [mat[6,0], mat[7,0], mat[8,0]] ,[
mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[9,0], mat[10,0], mat[11,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_130", [mat[12,0], mat[13,0], mat[14,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[15,0], mat[16,0], mat[17,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_140", [mat[18,0], mat[19,0], mat[20,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[21,0], mat[22,0], mat[23,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_160", [mat[24,0], mat[25,0], mat[26,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[27,0], mat[28,0], mat[29,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_180", [mat[30,0], mat[31,0], mat[32,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[33,0], mat[34,0], mat[35,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_200", [mat[36,0], mat[37,0], mat[38,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[39,0], mat[40,0], mat[41,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_90" , [mat[42,0], mat[43,0], mat[44,0]]
 ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[45,0], mat[46,0], mat[47,0]] )
SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Dor" , [mat[51,0], mat[52,0], mat
[53,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[54,0], mat[55,0], mat[56,0]]
 )
SapModel.PropMaterial.SetMPOrthotropic("CLT_100_Vindu" , [mat[57,0], mat[58,0], mat
[59,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[60,0], mat[61,0], mat[62,0]]
 )
SapModel.PropMaterial.SetMPOrthotropic("CLT_120_Vindu" , [mat[63,0], mat[64,0], mat
[65,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[66,0], mat[67,0], mat[68,0]]
 )
SapModel.PropMaterial.SetMPOrthotropic("CLT_130_Dor" , [mat[69,0], mat[70,0], mat
[71,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[72,0], mat[73,0], mat[74,0]]
 )
SapModel.PropMaterial.SetMPOrthotropic("CLT_160_Dor" , [mat[75,0], mat[76,0], mat
[77,0]] ,[mat[81,0],mat[81,0],mat[81,0]], [0,0,0],[mat[78,0], mat[79,0], mat[80,0]]
 )
Number_Mode = 30
ret = SapModel.LoadCases.ModalEigen.SetNumberModes("MODAL", Number_Mode,
```

```
    Number_Mode)

    #Beregne eigenmatrise-------------------------------------------------------
    ret = SapModel.Analyze.RunAnalysis()

    #Beregne egenfrekvens-------------------------------------------------------

    NumberResults = 0 #Ukjent
    Period = []
    Frequency = []
    CircFreq = []
    EigenValue = []
    StepNum = []
    Modal = []
    Mode = []
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    [NumberResults,Modal,Mode,StepNum,Period, Frequency, CircFreq, EigenValue,ret] =
    SapModel.Results.ModalPeriod(NumberResults, "Modal", "Mode", StepNum, Period,
    Frequency, CircFreq,EigenValue)

    #Beregner Modal Participationg Mass Ratio

    NumberResults = 0 #ukjent
    StepType = ""
    StepNum = []
    Period = []
    Ux = []
    Uy = []
    Uz = []
    SumUx = []
    SumUy = []
    SumUz = []
    Rx = []
    Ry = []
    Rz = []
    SumRx = []
    SumRy = []
    SumRz = []

    NumberResults,LoadCase, StepType, StepNum, Period, Ux, Uy, Uz, SumUx, SumUy, SumUz,
     Rx, Ry, Rz, SumRx, SumRy, SumRz,ret = SapModel.Results.
    ModalParticipatingMassRatios(NumberResults, "MODAL", StepType, StepNum, Period, Ux,
     Uy, Uz, SumUx, SumUy, SumUz, Rx, Ry, Rz, SumRx, SumRy, SumRz)

    MT = np.zeros(len(Ux))
    for ii in range(len(Ux)):
        if Ux[ii] + Uy[ii] + Rz[ii] < 0.01:
            MT[ii] = 4.
        elif abs(Ux[ii] - Uy[ii]) < 0.1:
            if Rz[ii] > Ux[ii]/2 and Rz[ii] > Uy[ii]/2 :
                MT[ii] = 2.
            else:
                MT[ii] = 3.
        elif Ux[ii] > Uy[ii] and Ux[ii] > Rz[ii]:
            MT[ii] = 0.
        elif Uy[ii] > Ux[ii] and Uy[ii] > Rz[ii]:
            MT[ii] = 1.
        elif Rz[ii] > Ux[ii] and Rz[ii] > Uy[ii]:
```

```
            MT[ii] = 2.
        else:
            MT[ii] = 5.
    # Finner mode type
    andelMode = np.zeros((3,len(Ux)))
    for ii in range(len(Ux)):
        tot_ii = Ux[ii] + Uy[ii] + Rz[ii]
        andelMode[0,ii] = ((Ux[ii] + Uy[ii])/tot_ii)*100
        andelMode[1,ii] =          ((Rz[ii])/tot_ii)*100
        andelMode[2,ii] = (math.atan(Uy[ii]/Ux[ii]))*(180/math.pi)

    GroupElm = 2
    NumberResults = 0
    Obj = []
    Elm = []
    LoadCase = "Modal"
    StepType = "Mode"
    StepNum = []
    U1 = []
    U2 = []
    U3 = []
    R1 = []
    R2 = []
    R3 = []
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("MODAL")

    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,R3,ret] = SapModel.
    Results.JointDispl("Aroof", GroupElm, NumberResults, Obj, Elm, LoadCase, StepType,
    StepNum, U1, U2, U3, R1, R2, R3)

    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    #Finn reaaksjoner i gruppe ii z retning

    Ubrukelig1 = []
    Ubrukelig2 = []
    Ubrukelig3 = []
    Name1 = []
    Name2 = []
    ItemTypeElm = 3
    NumberResults = 0
    Obj = []
    Elm = []
    LoadCase = []
    StepType = []
    StepNum = []
    F1 = []
    F2 = []
    F3 = []
    M1 = []
    M2 = []
    M3 = []

    ret = SapModel.SelectObj.All(True)
    ret = SapModel.SelectObj.CoordinateRange(-10000000, 10000000, -100000000, 10000000,
     -10, 10,False,"Global" ,True ,True, False, True,False, False)
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("DEAD")

    [Ubrukelig1, Name1, Name2, LoadCase, Steptype,Ubrukelig2, F1, F2, F3, M1, M2, M3,
```

```
    Ubrukelig3] = SapModel.Results.JointReact("ALL", ItemTypeElm, NumberResults, Obj,
    Elm, "DEAD", StepType, StepNum, F1, F2, F3, M1, M2, M3)
    ret = SapModel.SelectObj.All(True)

    return Frequency,[Ux,Uy,SumUx,SumUy],sum(F3),MT,andelMode


def ModeType(Ux,Uy,Rz):
    MT = np.zeros(len(Ux))
    #MT = list(range(len(Ux)))
    for ii in range(len(Ux)):
        if Ux[ii] + Uy[ii] + Rz[ii] < 0.01:
            MT[ii] = 4.
        elif abs(Ux[ii] - Uy[ii]) < 0.1:
            if Rz[ii] > Ux[ii]/2 and Rz[ii] > Uy[ii]/2 :
                MT[ii] = 2.
            else:
                MT[ii] = 3.
        elif Ux[ii] > Uy[ii] and Ux[ii] > Rz[ii]:
            MT[ii] = 0.
        elif Uy[ii] > Ux[ii] and Uy[ii] > Rz[ii]:
            MT[ii] = 1.
        elif Rz[ii] > Ux[ii] and Rz[ii] > Uy[ii]:
            MT[ii] = 2.
        else:
            MT[ii] = 5.
    return MT

def traVStor(liste):
    if liste[-1] < 1:
        ret = "Tra"
    elif liste[-1] < 2:
        ret = "Tor/Tra"
    else:
        ret = "Tor"
    return ret

def masse(inp1,aDekke,vCLT):
    OrgAF = 327.1
    OrgAR = 340.5
    OrgL  = 1039.9573

    OrgM = 8*inp1[49,0]*OrgAF + inp1[50,0]*OrgAR + OrgL*inp1[48,0]

    m = sum(aDekke[0:-1] * inp1[49,0]) + aDekke[-1]*inp1[50,0]

    for ii in vCLT[:,1]:
        m += float(ii)*inp1[48,0]

    return m,OrgM

"""
--------------------------------------------------------------------------------
Matrices for storing answers:
--------------------------------------------------------------------------------
"""

L3it = np.linspace(1.75,4.00,20)
e_verdi = list(range(len(L3it)))
```

```
f_verdi = np.zeros((5,2*(len(L3it))))
tra_andel = np.zeros((5,(len(L3it))))
tor_andel = np.zeros((5,(len(L3it))))
vinkel_a  = np.zeros((5,(len(L3it))))
Masse_tot = np.zeros(len(L3it))

for hei in range(len(L3it)):
    """
    -------------------------------------------------------------------------------
    COPY FROM SAP2000 OAPI
    -------------------------------------------------------------------------------
    """
    #set the following flag to True to attach to an existing instance of the program
    #otherwise a new instance of the program will be started
    AttachToInstance = False
    #set the following flag to True to manually specify the path to SAP2000.exe
    #this allows for a connection to a version of SAP2000 other than the latest
    installation
    #otherwise the latest installed version of SAP2000 will be launched
    SpecifyPath = False
    #if the above flag is set to True, specify the path to SAP2000 below
    #ProgramPath = 'C:\Program Files\Computers and Structures\SAP2000 22\SAP2000.exe'
    #full path to the model
    #set it to the desired path of your model
    APIPath = 'C:\CSiAPIexample'
    if not os.path.exists(APIPath):
            try:
                os.makedirs(APIPath)
            except OSError:
                pass
    ModelPath = APIPath + os.sep + 'API_1-001.sdb'
    #create API helper object
    helper = comtypes.client.CreateObject('SAP2000v1.Helper')
    helper = helper.QueryInterface(comtypes.gen.SAP2000v1.cHelper)
    if AttachToInstance:
        #attach to a running instance of SAP2000
        try:
            #get the active SapObject
                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:
                #create an instance of the SAPObject from the latest installed SAP2000
                mySapObject = helper.CreateObjectProgID("CSI.SAP2000.API.SapObject")
            except (OSError, comtypes.COMError):
                print("Cannot start a new instance of the program.")
                sys.exit(-1)
        #start SAP2000 application
        mySapObject.ApplicationStart()
```

```
    #create SapModel object
    SapModel = mySapObject.SapModel
    #initialize model
    SapModel.InitializeNewModel()
    #create new blank model
    ret = SapModel.File.NewBlank()

    """
    --------------------------------------------------------------------------------
    DEFAULT MODEL SETTINGS:
    --------------------------------------------------------------------------------
    """
    kN_mm_C = 5
    kN_m_C = 6
    ret = SapModel.SetPresentUnits(kN_mm_C)
    #Save the new model
    path = "/Sap/L"
    path = path+str(int(round(time.time())))+".sdb"
    ret = SapModel.File.Save(path)

    #Defines the material
    ret = SapModel.PropMaterial.SetMaterial('CLT_100',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_120',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_130',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_140',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_160',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_180',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_200',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_90',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_100_Dor',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_100_Vindu',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_120_Vindu',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_130_Dor',3)
    ret = SapModel.PropMaterial.SetMaterial('CLT_160_Dor',3)
    ret = SapModel.PropArea.SetShell("CLT_100", 1, "CLT_100", 0, 100, 100)
    ret = SapModel.PropArea.SetShell("CLT_120", 1, "CLT_120", 0, 120, 120)
    ret = SapModel.PropArea.SetShell("CLT_130", 1, "CLT_130", 0, 130, 130)
    ret = SapModel.PropArea.SetShell("CLT_140", 1, "CLT_140", 0, 140, 140)
    ret = SapModel.PropArea.SetShell("CLT_160", 1, "CLT_160", 0, 160, 160)
    ret = SapModel.PropArea.SetShell("CLT_180", 1, "CLT_180", 0, 180, 180)
    ret = SapModel.PropArea.SetShell("CLT_200", 1, "CLT_200", 0, 200, 200)
    ret = SapModel.PropArea.SetShell("CLT_90", 1, "CLT_90", 0, 90, 90)
    ret = SapModel.PropArea.SetShell("CLT_100_Dor", 1, "CLT_100_Dor", 0, 100, 100)
    ret = SapModel.PropArea.SetShell("CLT_100_Vindu", 1, "CLT_100_Vindu", 0, 100, 100)
    ret = SapModel.PropArea.SetShell("CLT_120_Vindu", 1, "CLT_120_Vindu", 0, 120, 120)
    ret = SapModel.PropArea.SetShell("CLT_130_Dor", 1, "CLT_130_Dor", 0, 130, 130)
    ret = SapModel.PropArea.SetShell("CLT_160_Dor", 1, "CLT_160_Dor", 0, 160, 160)
    inp1 = np.array([[ 6.74794564,  3.374     , 10.122     ],
           [ 4.62206714,  2.311     ,  6.933     ],
           [ 0.3       ,  0.15      ,  0.45      ],
           [ 0.46911976,  0.1687    ,  0.5061    ],
           [ 0.03374   ,  0.01687   ,  0.05061   ],
           [ 0.03374   ,  0.01687   ,  0.05061   ],
           [ 7.45694048,  3.72833333, 11.185     ],
           [ 3.91292597,  1.95666667,  5.87      ],
           [ 0.3       ,  0.15      ,  0.45      ],
           [ 0.39543461,  0.18641667,  0.55925   ],
           [ 0.03728333,  0.01864167,  0.055925  ],
           [ 0.03728333,  0.01864167,  0.055925  ],
```

```
            [ 7.72923077,  3.86461538, 11.59384615],
            [ 3.64076923,  1.82038462,  5.46115385],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.38646154,  0.19323077,  0.57969231],
            [ 0.03864615,  0.01932308,  0.05796923],
            [ 0.03864615,  0.01932308,  0.05796923],
            [ 7.95790706,  3.98142857, 11.94428571],
            [ 3.40830835,  1.70357143,  5.11071429],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.46413044,  0.19907143,  0.59721429],
            [ 0.03981429,  0.01990714,  0.05972143],
            [ 0.03981429,  0.01990714,  0.05972143],
            [ 8.33626101,  4.17125   , 12.51375   ],
            [ 3.02863437,  1.51375   ,  4.54125   ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.2085625 ,  0.2085625 ,  0.6256875 ],
            [ 0.0417125 ,  0.02085625,  0.06256875],
            [ 0.0417125 ,  0.02085625,  0.06256875],
            [ 7.46060274,  3.72833333, 11.185     ],
            [ 3.91443731,  1.95666667,  5.87      ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.44055026,  0.18641667,  0.55925   ],
            [ 0.03728333,  0.01864167,  0.055925  ],
            [ 0.03728333,  0.01864167,  0.055925  ],
            [ 6.7500431 ,  3.374     , 10.122     ],
            [ 4.62222073,  2.311     ,  6.933     ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.3503614 ,  0.1687    ,  0.5061    ],
            [ 0.03374   ,  0.01687   ,  0.05061   ],
            [ 0.03374   ,  0.01687   ,  0.05061   ],
            [ 7.4574407 ,  3.72833333, 11.185     ],
            [ 3.91346507,  2.311     ,  5.87      ],
            [ 0.3       ,  0.15      ,  0.45      ],
            [ 0.38303263,  0.1687    ,  0.55925   ],
            [ 0.03728333,  0.01687   ,  0.055925  ],
            [ 0.03728333,  0.01687   ,  0.055925  ],
            [ 4.54235651,  2.25      ,  6.75      ],
            [ 2.66726807,  1.275     ,  3.825     ],
            [ 0.19336342,  0.084     ,  0.252     ],
            [ 5.06587119,  2.5305    ,  7.5915    ],
            [ 3.46555758,  1.73325   ,  5.19975   ],
            [ 0.225     ,  0.1125    ,  0.3375    ],
            [ 0.05061   ,  0.05061   ,  0.45549   ],
            [ 0.025305  ,  0.0126525 ,  0.0379575 ],
            [ 0.025305  ,  0.0126525 ,  0.0379575 ],
            [ 5.19628085,  2.59798   ,  7.79394   ],
            [ 3.55896399,  1.77947   ,  5.33841   ],
            [ 0.231     ,  0.1155    ,  0.3465    ],
            [ 0.20338306,  0.0519596 ,  0.4676364 ],
            [ 0.0259798 ,  0.0129899 ,  0.0389697 ],
            [ 0.0259798 ,  0.0129899 ,  0.0389697 ],
            [ 5.75462719,  2.87081667,  8.61245   ],
            [ 3.01233337,  1.50663333,  4.5199    ],
            [ 0.231     ,  0.1155    ,  0.3465    ],
            [ 0.30001279,  0.05741633,  0.516747  ],
            [ 0.02870817,  0.01435408,  0.04306225],
            [ 0.02870817,  0.01435408,  0.04306225],
            [ 5.8042818 ,  2.89846154,  8.69538462],
            [ 2.72898192,  1.36528846,  4.09586538],
```

```
        [ 0.225     ,  0.1125    ,  0.3375    ],
        [ 0.05796923,  0.05796923,  0.52172308],
        [ 0.02898462,  0.01449231,  0.04347692],
        [ 0.02898462,  0.01449231,  0.04347692],
        [ 6.25802976,  3.1284375 ,  9.3853125 ],
        [ 2.27042119,  1.1353125 ,  3.4059375 ],
        [ 0.225     ,  0.1125    ,  0.3375    ],
        [ 0.06256875,  0.06256875,  0.56311875],
        [ 0.03128438,  0.01564219,  0.04692656],
        [ 0.03128438,  0.01564219,  0.04692656],
        [ 0.43222457,  0.15      ,  0.45      ]])

x_label = ['CLT100_E1','CLT100_E2','CLT100_E3','CLT100_G1','CLT100_G2','CLT100_G3',
           'CLT120_E1','CLT120_E2','CLT120_E3','CLT120_G1','CLT120_G2','CLT120_G3'
,
           'CLT130_E1','CLT130_E2','CLT130_E3','CLT130_G1','CLT130_G2','CLT130_G3'
,
           'CLT140_E1','CLT140_E2','CLT140_E3','CLT140_G1','CLT140_G2','CLT140_G3'
,
           'CLT160_E1','CLT160_E2','CLT160_E3','CLT160_G1','CLT160_G2','CLT160_G3'
,
           'CLT180_E1','CLT180_E2','CLT180_E3','CLT180_G1','CLT180_G2','CLT180_G3'
,
           'CLT200_E1','CLT200_E2','CLT200_E3','CLT200_G1','CLT200_G2','CLT200_G3'
,
           'CLT90_E1' ,'CLT90_E2' ,'CLT90_E3' ,'CLT90_G1' ,'CLT90_G2' ,'CLT90_G3'
,
           'Sf_CLT'   ,'Live_floor','Live_roof',
           'CLT100_E1_d','CLT100_E2_d','CLT100_E3_d','CLT100_G1_d','CLT100_G2_d','
CLT100_G3_d',
           'CLT100_E1_v','CLT100_E2_v','CLT100_E3_v','CLT100_G1_v','CLT100_G2_v','
CLT100_G3_v',
           'CLT120_E1_v','CLT120_E2_v','CLT120_E3_v','CLT120_G1_v','CLT120_G2_v','
CLT120_G3_v',
           'CLT130_E1_d','CLT130_E2_d','CLT130_E3_d','CLT130_G1_d','CLT130_G2_d','
CLT130_G3_d',
           'CLT160_E1_d','CLT160_E2_d','CLT160_E3_d','CLT160_G1_d','CLT160_G2_d','
CLT160_G3_d',
           'Poisson']

vCLT = np.array([['CLT_90',0],['CLT_100',0],['CLT_120',0],['CLT_130',0],['CLT_140'
,0],['CLT_160',0],
                ['CLT_180',0],['CLT_200',0],['CLT_100_Dor',0],['CLT_100_Vindu',0],
                ['CLT_120_Vindu',0],['CLT_130_Dor',0],['CLT_160_Dor',0]]);

#Set masssource
ret = SapModel.SourceMass.SetMassSource("M1", False, False, True, True, 1, ["DEAD"
], [1])
ret = SapModel.SetPresentUnits(kN_m_C)
#Defines groups
ret = SapModel.GroupDef.SetGroup("Aroof")
ret = SapModel.GroupDef.SetGroup("Aint")

"""
---------------------------------------------------------------------------
Input for Geometry-1:
---------------------------------------------------------------------------
"""
h  = 2.95
```

```
    A  = 345
    L3 = L3it[hei]
    L1 = 4*L3
    L4 = 1.67
    L2 = ((-2)*L1 + ((2*L1)**2 + 4*A)**0.5)/2
    L5 = (L2-L4)/2

    x1 = L5/2
    x2 = L5
    x3 = L5+L4
    x4 = x3 + (L5 / 2)
    x5 = L2
    x6 = L2 + L3
    x7 = L2 + 2*L3
    x8 = L2 + 3*L3
    x9 = L2 + L1

    #Grid in mm
    x_ = np.array([0,x1,x2,x3,x4,x5,x6,x7,x8,x9])
    y_ = np.array([0,x1,x2,x3,x4,x5,x6,x7,x8,x9])
    z_ = np.array([0,h,2*h,3*h,4*h,5*h,6*h,7*h,8*h])

    wallxtype = np.array([[["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["","","","","","","","",""],
                           ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","","","",""],
                           ["CLT_140","CLT_140","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["CLT_90","","","","CLT_90","","","",""],
                           ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                           ["CLT_90","","","","CLT_90","","","",""],
                           ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                          [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["","","","","","","","",""],
                           ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","","","",""],
                           ["CLT_140","CLT_140","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["CLT_90","","","","CLT_90","","","",""],
                           ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                           ["CLT_90","","","","CLT_90","","","",""],
                           ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                          [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["","","","","","","","",""],
                           ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
```

```
                              ["","","","","","","","",""],
                              ["CLT_140","CLT_140","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                             [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["","","","","","","","",""],
                              ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                             [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["","","","","","","","",""],
                              ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                             [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["","","","","","","","",""],
                              ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                             [["CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","","","","","","",""],
                              ["","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","
```

```
CLT_100_Dor"],
                       ["","","","","","","","",""],
                       ["CLT_100","CLT_100","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]],

                      [["CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                       ["","","","","","","","",""],
                       ["","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                       ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                       ["","","","","","","","",""],
                       ["CLT_100","CLT_100","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]]])

wallytype = np.array([[["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                       ["","","","","","","","",""],
                       ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                       ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                       ["","","","","","","","",""],
                       ["CLT_160","CLT_160","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                      [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                       ["","","","","","","","",""],
                       ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                       ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                       ["","","","","","","","",""],
                       ["CLT_160","CLT_160","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                       ["CLT_90","","","","CLT_90","","","",""],
                       ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                      [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
```

```
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                          ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","CLT_140","CLT_140","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                         [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                         [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                         [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                          ["","","","","","","","",""],
                          ["CLT_160","CLT_160","","CLT_120","CLT_120","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                          ["CLT_90","","","","CLT_90","","","",""],
                          ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                          ["CLT_90","","","","CLT_90","","","",""],
```

```
                        ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","
    CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","
    CLT_100_Vindu"],
                        ["","","","","","","","",""],
                        ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor","
    CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","
    CLT_100_Dor"],
                        ["","","","","","","","",""],
                        ["CLT_120","CLT_120","","CLT_100","CLT_100","CLT_100_Vindu",
    "CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","","","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                        ["CLT_90","","","","CLT_90","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","
    CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","
    CLT_100_Vindu"],
                        ["","","","","","","","",""],
                        ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor","
    CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","
    CLT_100_Dor"],
                        ["","","","","","","","",""],
                        ["CLT_120","CLT_120","","CLT_100","CLT_100","CLT_100_Vindu",
    "CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","","","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","","","",""],
                        ["CLT_90","","","","CLT_90","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]]])

    len(z_)-1 == len(wallxtype)
    len(y_)   == len(wallxtype[0,:,0])
    len(x_)-1 == len(wallxtype[0,0,:])

    len(y_)-1 == len(wallxtype[0,0,:])
    len(x_)   == len(wallxtype[0,:,0])
    len(z_)-1 == len(wallxtype)

    #Makes grid name:
    x_name = np.array(list(range(1,len(x_)+1))).astype('str')
    y_name = np.array(list(string.ascii_uppercase)[0:len(y_)])
    z_name = np.array(list(range(1,len(z_)+1))).astype('str')

    x = np.transpose(np.array([x_,x_name]))
    y = np.transpose(np.array([y_,y_name]))
    z = np.transpose(np.array([z_,z_name]))
    Unused  = ["4B","5B","7G","7H","7I","7J","8G","8H","8I","8J","9G","9H","9I","9J","
    10G","10H","10I","10J"];
    Undekke = ["3A","4A","5A","3B","4B","5B"] #First x and y coordinat of a dekke that
    is not in use.

    """
    ----------------------------------------------------------------------------
    Input for Geometry-2:
    ----------------------------------------------------------------------------
```

```
    """
    h = 2.950
    A = 345

    L3 = L3it[hei]
    L1l = 6*L3
    L1s = 2*L3
    L4 = 1.67
    L2 = (-((4/3)*L1l)+(((4/3)*L1l)**2 +4*A)**(0.5)) / 2
    L5 = (L2 - L4)/2

    x1 = L5/2
    x2 = L5
    x3 = L5+L4
    x4 = x3 + (L5 / 2)
    x5 = L2
    x6 = L2 + 1*L3
    x7 = L2 + 2*L3
    x8 = L2 + 3*L3
    x9 = L2 + 4*L3
    x10= L2 + 5*L3
    x11= L2 + L1l

    y1 = L5/2
    y2 = L5
    y3 = L5+L4
    y4 = y3 + (L5 / 2)
    y5 = L2
    y6 = L2 + L3
    y7 = L2 + L1s


    #Grid in mm
    x_ = np.array([0,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11])
    y_ = np.array([0,y1,y2,y3,y4,y5,y6,y7])
    z_ = np.array([0,h,2*h,3*h,4*h,5*h,6*h,7*h,8*h])


    wallxtype = np.array([[["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["","","","","","","","","","",""],
                           ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                           ["","","","","","","","","","",""],
                           ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["CLT_90","","","","CLT_90","","","","","",""],
                           ["CLT_140","CLT_140","","CLT_140","CLT_140","","","","","",""]],

                          [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                           ["","","","","","","","","","",""],
                           ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
```

```
                    ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                    ["","","","","","","","","","",""],
                    ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["CLT_90","","","","CLT_90","","","","","",""],
                    ["CLT_140","CLT_140","","CLT_140","CLT_140","","","","","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","","","",""],
                    ["","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                    ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                    ["","","","","","","","","","",""],
                    ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["CLT_90","","","","CLT_90","","","","","",""],
                    ["CLT_140","CLT_140","","CLT_140","CLT_140","","","","","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","","","",""],
                    ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","","","","","",""],
                    ["CLT_130","CLT_130","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["CLT_90","","","","CLT_90","","","","","",""],
                    ["CLT_120","CLT_120","","CLT_120","CLT_120","","","","","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","","","",""],
                    ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","","","","","",""],
                    ["CLT_130","CLT_130","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["CLT_90","","","","CLT_90","","","","","",""],
                    ["CLT_120","CLT_120","","CLT_120","CLT_120","","","","","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","","","",""],
                    ["","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","
```

```
CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                        ["","","","","","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor","CLT_130_Dor"],
                        ["","","","","","","","","","",""],
                        ["CLT_130","CLT_130","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                        ["CLT_90","","","","CLT_90","","","","","",""],
                        ["CLT_120","CLT_120","","CLT_120","CLT_120","","","","","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["","","","","","","","","","",""],
                        ["","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","","","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","","","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","","","","","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["","","","","","","","","","",""],
                        ["","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","","","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","","","","","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","","","","","",""]]])


wallytype = np.array([[["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","","","","",""],
                        ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor"],
                        ["","","","","","CLT_160_Dor","CLT_160_Dor"],
                        ["","","","","","",""],
                        ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                       [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","","","","",""],
```

```
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                           [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                           [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor"],
                            ["","","","","","CLT_130_Dor","CLT_130_Dor"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                           [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor"],
                            ["","","","","","CLT_130_Dor","CLT_130_Dor"],
                            ["","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                            ["CLT_90","","","","CLT_90","",""],
                            ["CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                           [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
```

```
                        ["","","","","","",""],
                        ["CLT_160","CLT_160","","","","CLT_130_Dor","CLT_130_Dor"],
                        ["","","","","","CLT_130_Dor","CLT_130_Dor"],
                        ["","","","","","",""],
                        ["CLT_160","CLT_160","","CLT_130","CLT_130","CLT_120_Vindu","CLT_120_Vindu"],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["","","","","","",""],
                        ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","",""],
                        ["CLT_120","CLT_120","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","",""]],

                       [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                        ["","","","","","",""],
                        ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","CLT_100_Dor","CLT_100_Dor"],
                        ["","","","","","",""],
                        ["CLT_120","CLT_120","","CLT_100","CLT_100","CLT_100_Vindu","CLT_100_Vindu"],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["CLT_90","","","","CLT_90","",""],
                        ["CLT_100","CLT_100","","CLT_100","CLT_100","",""]]])

    #Makes grid name:
    x_name = np.array(list(range(1,len(x_)+1))).astype('str')
    y_name = np.array(list(string.ascii_uppercase)[0:len(y_)])
    z_name = np.array(list(range(1,len(z_)+1))).astype('str')

    x = np.transpose(np.array([x_,x_name]))
    y = np.transpose(np.array([y_,y_name]))
    z = np.transpose(np.array([z_,z_name]))
    Unused =  ["4B","5B","7G","8G","9G","10G","11G","12G","7H","8H","9H","10H","11H","12H"];
    Undekke = ["3A","4A","5A","3B","4B","5B"] #First x and y coordinat of a dekke that is not in use.

    """
    ---------------------------------------------------------------------------
    Input for Geometry-3:
```

```
-----------------------------------------------------------------------------
"""
A = 345
L4 = 1.67

L3 = L3it[hei]
L5 = ((-8*L3+((8*L3)**2 + 4*A)**0.5)/2 - L4)/2

x1 = L3
x2 = x1 + L3
x3 = x2 + L5/2
x4 = x3 + L5/2
x5 = x4 + L4
x6 = x5 + L5/2
x7 = x6 + L5/2
x8 = x7 + L3
x9 = x8 + L3

y1 = L5/2
y2 = y1 + L5/2
y3 = y2 + L4
y4 = y3 + L5/2
y5 = y4 + L5/2
y6 = y5 + L3
y7 = y6 + L3
y8 = y7 + L3
y9 = y8 + L3

#Grid in mm
x_ = np.array([0,x1,x2,x3,x4,x5,x6,x7,x8,x9])
y_ = np.array([0,y1,y2,y3,y4,y5,y6,y7,y8,y9])
z_ = np.linspace(0,2.95*8,9)

wallxtype = np.array([[["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","
CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","","","","","","",""],
                        ["CLT_160_Dor","CLT_160_Dor","","","CLT_180","CLT_180","
CLT_180","CLT_160_Dor","CLT_160_Dor"],
                        ["CLT_160_Dor","CLT_160_Dor","","","","","","CLT_160_Dor","
CLT_160_Dor"],
                        ["","","","","","","","",""],
                        ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","
CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","CLT_90","","","","CLT_90","",""],
                        ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                        ["","","CLT_90","","","","CLT_90","",""],
                        ["","","CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                       [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","
CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","","","","","","",""],
                        ["CLT_160_Dor","CLT_160_Dor","","","CLT_180","CLT_180","
CLT_180","CLT_160_Dor","CLT_160_Dor"],
                        ["CLT_160_Dor","CLT_160_Dor","","","","","","CLT_160_Dor","
CLT_160_Dor"],
                        ["","","","","","","","",""],
                        ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","
CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
                        ["","","CLT_90","","","","CLT_90","",""],
```

```
                    ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","",""],
                    ["CLT_160_Dor","CLT_160_Dor","","","CLT_180","CLT_180","CLT_180","CLT_160_Dor","CLT_160_Dor"],
                    ["CLT_160_Dor","CLT_160_Dor","","","","","","CLT_160_Dor","CLT_160_Dor"],
                    ["","","","","","","","",""],
                    ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_140","CLT_140","","CLT_140","CLT_140","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","",""],
                    ["CLT_130_Dor","CLT_130_Dor","","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor"],
                    ["CLT_130_Dor","CLT_130_Dor","","","","","","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","","","",""],
                    ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","",""],
                    ["CLT_130_Dor","CLT_130_Dor","","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor"],
                    ["CLT_130_Dor","CLT_130_Dor","","","","","","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","","","",""],
                    ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                    ["","","CLT_90","","","","CLT_90","",""],
                    ["","","CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                   [["CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_160","CLT_160","CLT_160","CLT_120_Vindu","CLT_120_Vindu"],
                    ["","","","","","","","",""],
                    ["CLT_130_Dor","CLT_130_Dor","","","CLT_160","CLT_160","CLT_160","CLT_130_Dor","CLT_130_Dor"],
                    ["CLT_130_Dor","CLT_130_Dor","","","","","","CLT_130_Dor","CLT_130_Dor"],
                    ["","","","","","","","",""],
                    ["CLT_120_Vindu","CLT_120_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu"],
```

```
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_120","CLT_120","","CLT_120","CLT_120","",""]],

                             [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_100_Dor","CLT_100_Dor","","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor"],
                              ["CLT_100_Dor","CLT_100_Dor","","","","","","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_100_Vindu","CLT_100_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_100","CLT_100","","CLT_100","CLT_100","",""]],

                             [["CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_120","CLT_120","CLT_120","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_100_Dor","CLT_100_Dor","","","CLT_120","CLT_120","CLT_120","CLT_100_Dor","CLT_100_Dor"],
                              ["CLT_100_Dor","CLT_100_Dor","","","","","","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_100_Vindu","CLT_100_Vindu","CLT_90","CLT_90","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_90","CLT_90","","CLT_90","CLT_90","",""],
                              ["","","CLT_90","","","","CLT_90","",""],
                              ["","","CLT_100","CLT_100","","CLT_100","CLT_100","",""]]])

    wallytype = np.array([[["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                              ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                             [["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                              ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
```

```
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                           [["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_140","CLT_140","","CLT_140","CLT_140","","","",""]],

                           [["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                           [["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                           [["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""],
                            ["CLT_90","","","","CLT_90","","","",""],
                            ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                            ["","","","","","","","",""],
                            ["CLT_160","CLT_160","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor"],
                            ["","","","","","CLT_160_Dor","CLT_160_Dor","CLT_160_Dor","
```

```
CLT_160_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_160","CLT_160","","CLT_90","CLT_90","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu","CLT_120_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_120","CLT_120","","CLT_120","CLT_120","","","",""]],

                             [["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]],

                             [["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_90","CLT_90","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor","CLT_100_Dor"],
                              ["","","","","","","","",""],
                              ["CLT_120","CLT_120","","CLT_90","CLT_90","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu","CLT_100_Vindu"],
                              ["CLT_90","","","","CLT_90","","","",""],
                              ["CLT_100","CLT_100","","CLT_100","CLT_100","","","",""]]])

len(z_)-1 == len(wallxtype)
len(y_)   == len(wallxtype[0,:,0])
len(x_)-1 == len(wallxtype[0,0,:])

len(y_)-1 == len(wallytype[0,0,:])
len(x_)   == len(wallytype[0,:,0])
len(z_)-1 == len(wallytype)

#Makes grid name:
x_name = np.array(list(range(1,len(x_)+1))).astype('str')
y_name = np.array(list(string.ascii_uppercase)[0:len(y_)])
z_name = np.array(list(range(1,len(z_)+1))).astype('str')

x = np.transpose(np.array([x_,x_name]))
y = np.transpose(np.array([y_,y_name]))
z = np.transpose(np.array([z_,z_name]))
Unused =  ["1G","1H","1I","1J","2G","2H","2I","2J","6B","7B","9G","9H","9I","9J","10G","10H","10I","10J"];
Undekke = ["5A","6A","7A","5B","6B","7B"] #First x and y coordinat of a dekke that is not in use.

"""
```

```
-----------------------------------------------------------------------------
Running of functions:
-----------------------------------------------------------------------------
"""
GridPoints(x,y,z,Unused)
WallX(wallxtype,x,y,z,vCLT)
WallY(wallytype,x,y,z,vCLT)
aDekke = Dekke(x,y,z,Undekke,vCLT)
SapModel.AreaObj.SetEdgeConstraint("ALL", True, 1)

"""
-----------------------------------------------------------------------------
Apply rigid diaphragm constraint:
-----------------------------------------------------------------------------
"""

NumberNames = 0; MyName = [];
ud1, navn, ud2 = SapModel.PointObj.GetNameList(NumberNames, MyName)

SapModel.GroupDef.SetGroup("1")
SapModel.ConstraintDef.SetDiaphragm("1",3)
SapModel.GroupDef.SetGroup("2")
SapModel.ConstraintDef.SetDiaphragm("2",3)
SapModel.GroupDef.SetGroup("3")
SapModel.ConstraintDef.SetDiaphragm("3",3)
SapModel.GroupDef.SetGroup("4")
SapModel.ConstraintDef.SetDiaphragm("4",3)
SapModel.GroupDef.SetGroup("5")
SapModel.ConstraintDef.SetDiaphragm("5",3)
SapModel.GroupDef.SetGroup("6")
SapModel.ConstraintDef.SetDiaphragm("6",3)
SapModel.GroupDef.SetGroup("7")
SapModel.ConstraintDef.SetDiaphragm("7",3)
SapModel.GroupDef.SetGroup("8")
SapModel.ConstraintDef.SetDiaphragm("8",3)
SapModel.GroupDef.SetGroup("9")
SapModel.ConstraintDef.SetDiaphragm("9",3)

for i in range(ud1):
    if navn[i][-1] == "1":
        SapModel.PointObj.SetGroupAssign(navn[i],"1",False,0)
    if navn[i][-1] == "2":
        SapModel.PointObj.SetGroupAssign(navn[i],"2",False,0)
    if navn[i][-1] == "3":
        SapModel.PointObj.SetGroupAssign(navn[i],"3",False,0)
    if navn[i][-1] == "4":
        SapModel.PointObj.SetGroupAssign(navn[i],"4",False,0)
    if navn[i][-1] == "5":
        SapModel.PointObj.SetGroupAssign(navn[i],"5",False,0)
    if navn[i][-1] == "6":
        SapModel.PointObj.SetGroupAssign(navn[i],"6",False,0)
    if navn[i][-1] == "7":
        SapModel.PointObj.SetGroupAssign(navn[i],"7",False,0)
    if navn[i][-1] == "8":
        SapModel.PointObj.SetGroupAssign(navn[i],"8",False,0)
    if navn[i][-1] == "9":
        SapModel.PointObj.SetGroupAssign(navn[i],"9",False,0)

SapModel.PointObj.SetConstraint("1", "1", 1)
```

```
    SapModel.PointObj.SetConstraint("2", "2", 1)
    SapModel.PointObj.SetConstraint("3", "3", 1)
    SapModel.PointObj.SetConstraint("4", "4", 1)
    SapModel.PointObj.SetConstraint("5", "5", 1)
    SapModel.PointObj.SetConstraint("6", "6", 1)
    SapModel.PointObj.SetConstraint("7", "7", 1)
    SapModel.PointObj.SetConstraint("8", "8", 1)
    SapModel.PointObj.SetConstraint("9", "9", 1)

    """
    ---------------------------------------------------------------------------
    Calculate eccentricity between center of mass and center of rigidity:
    ---------------------------------------------------------------------------
    """
    SapModel.SetModelIsLocked(False)
    SapModel.Analyze.RunAnalysis()

    NumberResults = 0 #ukjent
    GroupElm = 2
    PointElm = []
    U1 = []
    U2 = []
    U3 = []
    R1 = []
    R2 = []
    R3 = []
    [antall,label,U1,U2,U3,R1,R2,R3,null] = SapModel.Results.AssembledJointMass("ALL",
    GroupElm, NumberResults, PointElm, U1, U2, U3, R1, R2, R3)
    xm = 0
    ym = 0
    mx  = 0
    my = 0

    for i in range(antall):
        asdf = 0
        x = 0
        y = 0
        z = 0
        [x,y,z,asdf] = SapModel.PointObj.GetCoordCartesian(navn[i], x, y, z)

        xm += x*U1[i]
        ym += y*U2[i]
        mx +=   U1[i]
        my +=   U2[i]

    Xcom = xm/mx
    Ycom = ym/my

    SapModel.SetModelIsLocked(False)

    for i in range(len(x_name)):
        if Xcom > x_[i-1] and Xcom < x_[i]:
            Acomx = x_name[i-1]+"-"+x_name[i]
            indX  = i

    for i in range(len(y_name)):
        if Ycom > y_[i-1] and Ycom < y_[i]:
            Acomy = y_name[i-1]+"-"+y_name[i]
            indY  = i
```

```

    Floor = "9"
    Acom = Acomx+"_"+Acomy+"_"+Floor+"-"+Floor

    Name = ""
    SapModel.PointObj.AddCartesian(Xcom, Ycom, 0, Name, "COM")
    NumberAreas = 4  # The number of area objects created when the specified area object is divided.
    AreaName = []    # This is an array of the name of each area object created when the specified area object
                     # is divided.
    LocalAxesOnEdge = True #If this item is True, and if both points along an edge of the original area object
                     # have the same local axes, the program makes the local axes for added points along the edge
                     # the same as the edge end points.
    LocalAxesOnFace = True # If this item is True, and if both points along an edge of the original area object
                     # have the same local axes, the program makes the local axes for added points along the edge the
                     # same as the edge end points.
    RestraintsOnEdge = False  # If this item is True, and if both points along an edge of the original area object have
                     # the same restraint/constraint, then, if the added point and the adjacent corner points have the
                     # same local axes definition, the program includes the restraint/constraint for added points along
                     # the edge.
    RestraintsOnFace = False #If this item is True, and if all points around the perimeter of the original area object
                     # have the same restraint/constraint, then, if an added point and the perimeter points have the
                     #same local axes definition, the program includes the restraint/constraint for the added point.
    n1 = n2 = n3 = n4 = n6 = 0
    n5 = 0           # This item applies when MeshType = 5. MeshType = 5 provides cookie cut meshing based on two
                     # perpendicular lines passing through SELECTED point objects. By default these lines align with
                     # the area object local 1 and 2 axes. The Rotation item is an angle in degrees that the meshing
                     # lines are rotated from their default orientation. [deg]

    SapModel.PointObj.SetSelected("COM", True, 0)
    SapModel.EditArea.Divide(Acom, 5, NumberAreas, AreaName,n1,n1,n2,n2,n3,n3,n3,n4,n5,n6,LocalAxesOnEdge,LocalAxesOnFace,RestraintsOnEdge,RestraintsOnFace)
    SapModel.PointObj.SetGroupAssign("740",Floor,False,0)
    SapModel.PointObj.SetGroupAssign("741",Floor,False,0)
    SapModel.PointObj.SetGroupAssign("742",Floor,False,0)
    SapModel.PointObj.SetGroupAssign("743",Floor,False,0)
    SapModel.PointObj.SetGroupAssign("744",Floor,False,0)
    SapModel.PointObj.SetGroupAssign("COM",Floor,False,0)
    SapModel.PointObj.SetConstraint("9", "9", 1)

    SapModel.SetPresentUnits(10)

    SapModel.LoadPatterns.Add('X', 1, 0, True)
    SapModel.LoadPatterns.Add('Y', 1, 0, True)
    SapModel.LoadPatterns.Add('Z', 1, 0, True)
```

```

name = "COM"

ret = SapModel.PointObj.SetLoadForce(name, 'X',[1,0,0,0,0,0])
ret = SapModel.PointObj.SetLoadForce(name, 'Y',[0,1,0,0,0,0])
ret = SapModel.PointObj.SetLoadForce(name, 'Z',[0,0,0,0,0,1])

#Center of rigidity: ----------------------------------------------
SapModel.Analyze.RunAnalysis()
GroupElm = 0
NumberResults = 0
Obj = []
Elm = []
LoadCase = "X"
StepType = "X"
StepNum = []
U1 = []
U2 = []
U3 = []
R1 = []
R2 = []
RX = []
ret = SapModel.Results.Setup.SetCaseSelectedForOutput("X")
[NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,RX,ret] = SapModel.
Results.JointDispl(name, GroupElm, NumberResults, Obj, Elm, LoadCase, StepType,
StepNum, U1, U2, U3, R1, R2, RX)
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

GroupElm = 0
NumberResults = 0
Obj = []
Elm = []
LoadCase = "Y"
StepType = "Y"
StepNum = []
U1 = []
U2 = []
U3 = []
R1 = []
R2 = []
RY = []
ret = SapModel.Results.Setup.SetCaseSelectedForOutput("Y")
[NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,RY,ret] = SapModel.
Results.JointDispl(name, GroupElm, NumberResults, Obj, Elm, LoadCase, StepType,
StepNum, U1, U2, U3, R1, R2, RY)
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

GroupElm = 0
NumberResults = 0
Obj = []
Elm = []
LoadCase = "Z"
StepType = "Z"
StepNum = []
U1 = []
U2 = []
U3 = []
R1 = []
R2 = []
```

```
    RZ = []
    ret = SapModel.Results.Setup.SetCaseSelectedForOutput("Z")
    [NumberResults,Obj,Modal,Mode,StepNum,Period, U1,U2,U3,R1,R2,RZ,ret] = SapModel.
    Results.JointDispl(name, GroupElm, NumberResults, Obj, Elm, LoadCase, StepType,
    StepNum, U1, U2, U3, R1, R2, RZ)
    ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput()

    RX = RX[0];RY = RY[0];RZ = RZ[0]
    Xcor = Xcom - (RX/RZ)
    Ycor = Ycom + (RY/RZ)
    dist = ((Xcom-Xcor)**2 + (Ycom-Ycor)**2)**(0.5)
    e_verdi[hei] = dist

    print("L3: ",L3)
    print("Center of mass:      [",np.round(Xcom,3)," , ",np.round(Ycom,3),"]")
    print("Center of rigidity:  [",np.round(Xcor,3)," , ",np.round(Ycor,3),"]")
    print("Distance between com and cor: ",np.round(dist,3))

    """
    ---------------------------------------------------------------------------
    Run analysis and remove diaphragm constraints:
    ---------------------------------------------------------------------------
    """
    SapModel.SetModelIsLocked(False)
    SapModel.ConstraintDef.Delete("1")
    SapModel.ConstraintDef.Delete("2")
    SapModel.ConstraintDef.Delete("3")
    SapModel.ConstraintDef.Delete("4")
    SapModel.ConstraintDef.Delete("5")
    SapModel.ConstraintDef.Delete("6")
    SapModel.ConstraintDef.Delete("7")
    SapModel.ConstraintDef.Delete("8")
    SapModel.ConstraintDef.Delete("9")

    float_formatter = "{:.2f}".format
    np.set_printoptions(formatter={'float_kind':float_formatter})

    F,MassPar,MassRea,ModeType,ModeAndel = run_model(inp1)
    print("F: ",np.round(F,3))
    print("ModeType:  ",np.round(ModeType,3))
    print("ModeAndel: ",np.round(ModeAndel,3))
    print("Sum F:     ",np.round(MassRea,3))

    Masse_tot[hei] = MassRea
    j    = 0
    full = 0
    while full < 5:
        if ModeType[j] < 4:
            f_verdi[full,0 + 2*hei] = F[j]
            f_verdi[full,1 + 2*hei] = ModeType[j]
            tra_andel[full,hei] = ModeAndel[0,j]
            tor_andel[full,hei] = ModeAndel[1,j]
            vinkel_a[full,hei]  = ModeAndel[2,j]
            full += 1
        j += 1

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

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

Postboks 5003
NO-1432 Ås
Norway