Norges miljø- og biovitenskapelige universitet

Master Thesis 2023 30 EXTC
Faculty of Science and Technology (REALTEK)

Validation of Rocking Displacement for Segmented Cross Laminated Timber Shear Walls in Multi-Story Buildings Under Lateral Loads

Stian Edvardsen
Structural Engineering and Architecture

Acknowledgments

I would like to give a big thank you to my main supervisor Roberto Tomasi and external supervisor Andrew Smith for their help and guidance through out this semester. This thesis would not be possible without them. I also want to give thanks to Francesco Boggian for helping me out with the OAPI for SAP2000.

Abstract

In this thesis parametric analyses from numerical and analytical methods for one-story CLT shear walls are compared and validated for the upcoming Eurocode 5. The analytical methods are the already proposed method in Annex R and a new proposal from Arup. In addition, numerical and analytical analyses for multistory behavior are compared to see their preciseness for multistory behavior. The thesis only looks at kinematic rocking as a contribution to lateral deflection.

The parametric analysis for one-story utilizes Open Application Programming Interface with Python to manipulate the model made in the Finite Element Method program SAP2000 to do different series of analyses. The method allows for the extraction of the result made in the numerical analysis and compares the results from the analytical analyses. The method changes the different parameters such as stiffness of hold-downs, vertical joint stiffness, and the number of panels in segmented walls.

For multistory calculations Python is used for the analytical methods and hand modeling is done to make the SAP2000 models. Only certain cases are looked at in this scenario and therefore the comparison is done by hand.

The results showed very good agreement between the two analytical models from Annex R and Arup and the numerical analyses in SAP2000 when looking at single-story behavior. For multistory behavior, Annex R generally had a larger displacement, and Arup generally had a lower displacement when $K_v > K_h$. One difference between these methods is their way of calculating the rotation of the walls and that may be the reason for the discrepancy between those methods. In addition, Arup uses the moment at the top of the wall and lateral shear force as its input, while Annex R only uses the moment at the bottom of the wall. This may also be a contributor to the difference in multistory deflection. More studies are needed to further validate the analytical methods for multistory calculations.

Table of Contents

List of Tables

List of Figures

List of Symbols

θ_{i-1} Rotation at the top of the shear wall at the $i-1^{th}$ story

θ_{i-2} Rotation at the top of the $i-2^{th}$ story

$\varphi_{R,i-1}$ Rotation due to the rocking of the $i-1^{th}$ story

$\varphi_{R,i}$ The rotation at the top of the wall relative to the ground below

$\widetilde{N}_l$ Dimensionless vertical load on the shear wall

b Width of the panels

F_v Force acting in the vertical shear connectors

h_i Height of the panels

j^* First panel that is touching the ground at its trailing corner

K_h Hold-down stiffness

K_v Total vertical joint stiffness

$K_{R,i,CP}$ Rocking stiffness of the shear wall at the i^{th} story in CP mode

$K_{R,i,SW}$ Rocking stiffness of the shear wall at the i^{th} story in SW mode

K_{res} The residual stiffness

$K_{ser,anc,i}$ Vertical-tensile stiffness of the hold-down

$K_{ser,con,i}$ Stiffness of the vertical joint

l_i Length of the wall

l_j Width of a panel

l_{bot} Length not in compression at the bottom of the wall

l_{top} Length not in compression at the top of the wall

m Number of panels

$M_{i,Ed}$ Design moment at the bottom of the i^{th} story

$M_{i,N,Ed}$ Design moment acting at the top of the i^{th} story due to vertical loads, taken about the centreline of the wall

$M_{i,V,Ed}$ Design destabilizing moment acting at the top of the wall due to horizontal loads

N_A Vertical load on leading edge of wall due to vertical loads (compression positive)

N_B Vertical load on trailing edge of wall due to vertical loads (compression positive)

$N_{i,Ed}$ Design vertical force acting at the top of the wall of the i^{th} story.

$n_{v,j,i}$ Number of vertical connectors

Q Vertical force applied at top of leading wall edge (tension positive, compression negative)

q Uniformly distributed load at the top of the wall due to vertical forces

T Force in the tie-down at the bottom of the leading edge of the wall (tension positive)

T_{above} Tension force in the tie-down at the bottom of the leading edge of the wall at the floor above under the same set of loads

u_{CP} Horizontal deflection for the CP mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry

u_{IN} Lateral inter-story deflection of the wall when in INT mode calculated by using the exact number of panels that are lifting

$u_{R,i,CP}$ Inter-story lateral displacement due to the rocking of the i^{th} wall in CP mode

$u_{R,i,SW}$ Inter-story lateral displacement due to the rocking of the i^{th} wall in SW mode

$U_{R,i}$ The horizontal deflection at the top of the wall og the i^{th}

u_{SW} Horizontal deflection for the SW mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry

V_{CP} Horizontal force when going from CP to INT mode

$V_{i,Ed}$ The design horizontal shear force acting at the top of the wall at the i^{th} storey

V_{SW} Horizontal force when going from INT to SW mode

1 Introduction

1.1 Background

Cross Laminated Timber is a relatively new building material introduced in the 1990s. Its interest has become global (Brandner et al., 2016). Determining the elastic behavior of the material is quite complex because of the material characteristics and all the different contributions for lateral displacement. In today's Eurocode 5, there is no standardization on analytically calculating the Service Limit State of Cross Laminated Timber (CLT). This is now under development by several people to be implemented in the new Eurocode 5. However, before the different methods can be used, their performance must be tested and verified statically and dynamically.

Proposals of analytical methods for one-story shear walls have been created and verified. However, for multistory behavior, there has been little to none. Therefore, this thesis proposes validating two methods for calculating multistory deflections.

1.2 State of the art

A new Eurocode 5 is under production by the Technical Committee and will include analytical equations for calculating the lateral deflection of CLT walls subjected to lateral and seismic forces. Annex R.5 (Annex Y.5 in earlier drafts) in (CEN, 2023) contains the analytical equations for kinematic rocking for CLT walls already proposed for the new Eurocode 5.

A study from Casagrande et al. (2018) has examined kinematic rocking. The study uses the minimum total potential energy principle to make analytical equations to determine the mechanical behavior of a segmented CLT wall. The paper only considers a single-story wall. They find that a relatively stiff hold-down connection gives each panel an absolute center of rotation, and a relatively flexible hold-down connection allows the panels to uplift. The equations were validated by making a Finite Element model and comparing the results to each other.

A conference paper from Casagrande et al. (2023) shows an experimental verification of the same equations made in Casagrande et al. (2018) and a numerical verification for multistory. They found that the kinematic modes observed in the experimental test were the same as predicted by the analytical models. In the numerical validation, they got a maximum discrepancy of as little as 3% between the numerical and analytical results. Note that these calculations did not only consist of kinematic rocking but included the contributions from in-plane shear and rigid body sliding.

The study from Aloisio et al. (2023) looks at all the contributions for lateral displacement of single-story CLT and Light Timber Framed shear walls (LTF). The analytical equations used are the proposed equations for the new Eurocode 5. The numerical results were made in the finite element program SAP2000. 1830 parametric analyses were done and compared. The results for CLT shear walls showed excellent agreement for rigid body sliding and in-plane shear. The kinematic rocking of segmented CLT shear walls had a good agreement as long as no vertical force was present. When vertical forces were present, there was a significant bias. The comparison of in-plane bending showed little agreement between the analytical and numerical results.

A study done by D'Arenzo et al. (2021) studies how the floor-to-wall interaction between the stories in a multistory shear wall affects the rocking stiffness of segmented shear walls. The study uses an analytical model to represent the floor-to-wall interaction that calculates the lateral displacement and internal actions along the floor. The elastic analytical model is validated by the use of FEM models in SAP2000. The analytical model is used to find an equivalent spring to easily take into account the effect of the complexity that is floor-to-wall interactions. The study showed that the bending stiffness and the withdrawal stiffness of the wall-to-floor connection affected the shear walls' rocking stiffness and the kinematic behavior. An increase in those stiffnesses made an increase in rocking stiffness.

1.3 Research Questions

Considering the overall objective of the study, the specific aims were to investigate the following research questions:

1. How well does the lateral deflection due to kinematic rocking for one-story shear walls calculated by the analytical method from Arup match the results using Annex R?

2. Compared to numerical analyses, how do the analytical methods from Arup and Annex R predict lateral deflection due to kinematic rocking for two- and three-story CLT shear walls?

1.4 Research Objectives

This thesis studies the mechanical elastic behavior of single and multistory CLT shear walls. The thesis uses the analytical methods from Annex R and Arup to calculate the lateral displacement due to kinematic rocking for a one-story wall. It also uses FEM models made in SAP2000 to verify these deflections. One-story parametric analysis is done by changing the different parameters to see how they affect the displacement. It also studies how the methods work for predicting the lateral displacement of a multistory CLT shear wall and compares the results to FEM models. Finally, the thesis uses different floor stiffnesses in the two-story FEM design to see how this impacts the results.

2 Theory

In this chapter, the theoretical framework for the materials utilized will be presented. In Chapter 2.1 theory of the general properties of timber and CLT is described. Chapter 2.2 describes shear walls and their usage. Chapter 2.3 and 2.4 describe the theory of the two analytical methods used for the analytical calculation used in this thesis. Chapter 2.5 explains the Finite Element Method and the programs used.

2.1 CLT

Timber is an orthotropic material which means it has different capacities and characteristics depending on the orientation of the grain. The axis can be either parallel or perpendicular to the grain. The axes of the timber are longitudinal, tangential, or radial, as shown in Figure 2.1. The longitudinal is parallel to the grain, and the tangential and radial are perpendicular to the grain. The parallel axis is the strongest axis of the timber, and the perpendicular axes are generally much weaker. (Sanborn et al., 2019).

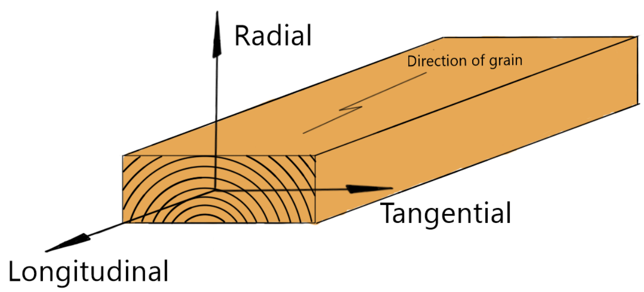


Figure 2.1: Axis of timber.

Cross-laminated timber (CLT) is made by gluing together an uneven number of timber plates consisting of side-by-side boards orthogonal to each other, as shown in Figure 2.2. This is done by using an adhesive. This gives the element rigidity in both in-plane directions because both axes will have timber parallel to the grain. CLT elements have a high weight-to-strength ratio. (Sanborn et al., 2019).

The interest in CLT has increased in the latest years due to the building industry's high impact on the emissions of climate gasses. Therefore the focus on building more with sustainable materials has increased. CLT has a much lower CO2 emission than more traditional building materials such as steel and concrete. A Life-Cycle-Analysis done by Younis and Dodoo (2022) shows a reduction in greenhouse gases by up to 40 % compared to more traditional building materials. This is due to the fact that timber stores carbon during its life cycle. Harvesting and processing timber also

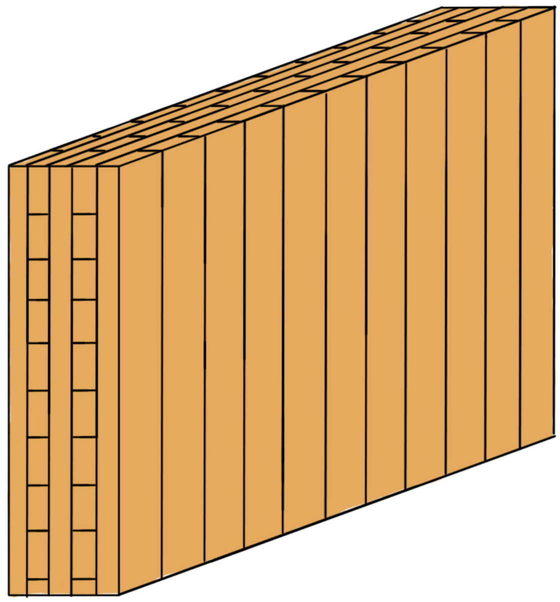


Figure 2.2: Illustration of a five-layered CLT panel.

has a lower emission and energy consumption than steel/concrete. The CLT can also be recycled and reused, lowering the impact on the climate (Younis and Dodoo, 2022).

The interest in CLT also comes from the fact that it is estimated that taller buildings are needed in cities and other urban places to house future population growth. CLT enables building larger and taller buildings than the more common light timber structures (Silva et al., 2013).

CLT products have a faster assembling time than mineral-based materials such as concrete because of their high degree of prefabrication (Brandner et al., 2016).

2.2 Shear Walls

The purpose of the shear walls is to resist horizontal loads such as wind and earthquakes. Therefore, shear walls should be orientated to resist in-plane forces along the length of the wall.

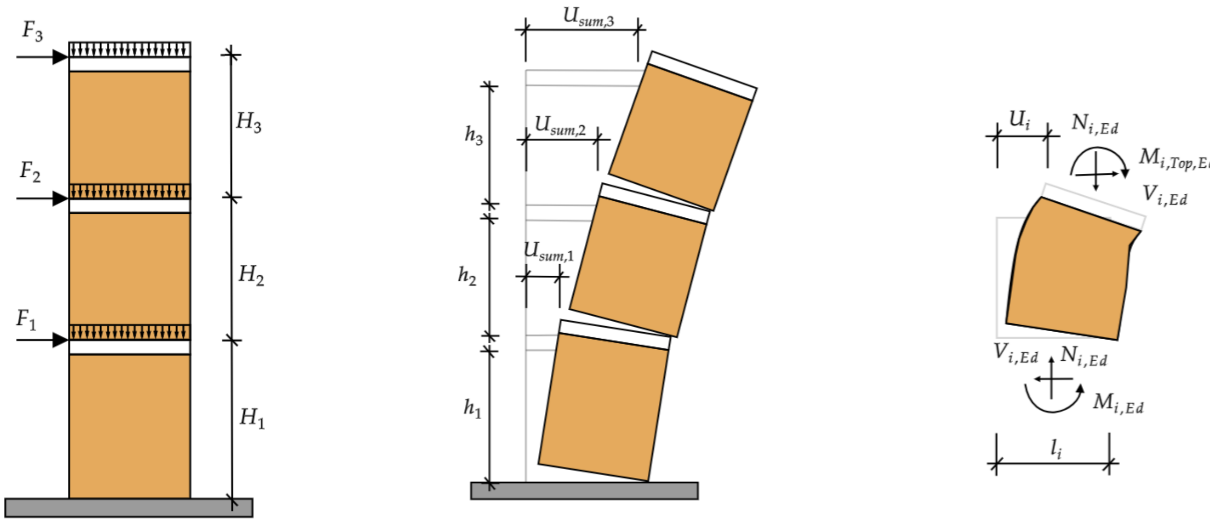
Shear walls made from Cross Laminated timber can consist of one monolithic element or be segmented into several panels. The segmented elements need to be connected in such a way that they work together. Different types of connections can be used. Typical methods are screwing together butted joints, screwing together rabbet edges, or screwing wooden boards to the face of the panels (Wallner-Novak et al., 2014).

CLT shear walls make it easy to install openings without having any super-ordinate grid. (Brandner et al., 2016).

2.3 Annex R

2.3.1 Background

Annex R is a proposed method for calculating the lateral displacement of multi-story shear walls of monolithic or segmented shear walls in the new Eurocode 5 (CEN, 2023). The method calculates the displacement due to kinematic rocking, in-plane bending deformation, rigid body sliding, and in-plane shear deformation. The method distinguishes between the wall as one monolithic panel or segmented panels.



(a) Undeformed multistory shear wall.

(b) Deformed multistory shear wall.

(c) Mechanisms in a single-story shear wall.

Figure 2.3: Lateral displacement of multistory shear walls. Drawn after Figure R.1 in CEN (2023)

For this method to be used, the walls shall have a height h_i, as presented in Figure 2.3. Figure 2.3c shows the internal actions of a single-story wall.

This thesis only looks at the kinematic rocking of segmented CLT walls. The kinematic rocking part of Annex R is based on the paper from Casagrande et al. (2018) that makes an analytical approach for the elastic behavior of one story, segmented CLT walls subjected by lateral loads. The model from Casagrande uses the relation between the stiffness of the hold-downs and the vertical shear connections between the panels. It neglects the in-plane deformation of the CLT panels by assuming they are rigid.

For segmented walls, the method predicts there are three possible modes:

- Coupled mode (CP)

- Intermediate mode (IN)

- Single wall mode (SW)

The coupled mode (CP) is when each segmented panel is in contact with the foundation below and has a local center of rotation, illustrated in Figure 2.4.

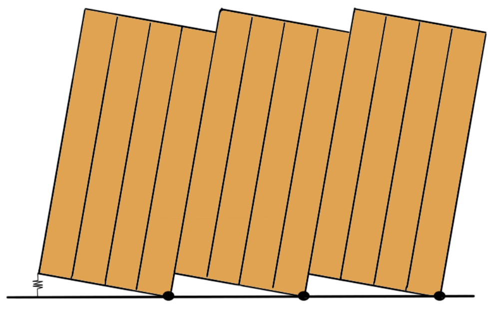


Figure 2.4: Illustration of a wall in coupled mode. Drawn after Figure R.4(a) in CEN (2023)

The single wall mode (SW) is when there is a single center of rotation on the edge of the entire wall, illustrated in Figure 2.5.

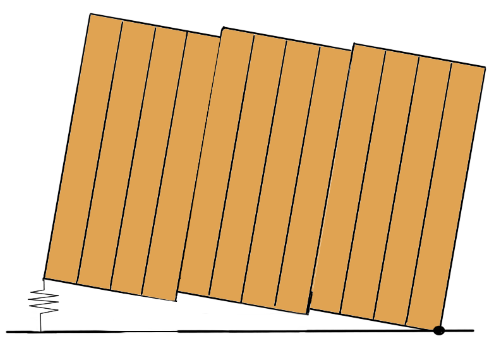


Figure 2.5: Illustration of a wall in single-wall mode. Drawn after figure R.4(c) in CEN (2023)

The intermediate mode (IN) is when only some panels are in contact with the foundation, illustrated in Figure 2.6. Some panels are in coupled mode, and some are in single-wall mode.

Annex R calculates the mode the wall is in and then calculates the deflection. Finding the mode, Formula 2.1 for CP mode, 2.3 for INT mode, and 2.2 for SW mode is used.

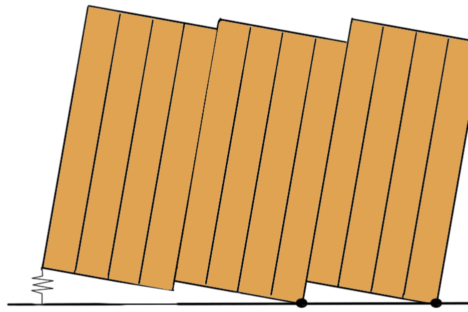


Figure 2.6: Illustration of a wall in intermediate mode. Drawn from Figure R.4(b) in CEN (2023)

$$\frac{K_{\text{ser,anc},i}}{n_{v,j,i} \cdot K_{\text{ser,con},i}} \geq \frac{1 - \widetilde{N}_l \cdot \frac{3m-2}{m^2}}{1 - \widetilde{N}_l \cdot \frac{m-2}{m^2}} \tag{2.1}$$

$$\frac{K_{\text{ser,anc},i}}{n_{v,j,i} \cdot K_{\text{ser,con},i}} \leq \frac{1 - \widetilde{N}_l}{1 + \widetilde{N}_l \cdot (m-2)} \tag{2.2}$$

$$\frac{1 - \widetilde{N}_l}{1 + \widetilde{N}_l \cdot (m-2)} < \frac{K_{\text{ser,anc},i}}{n_{v,j,i} \cdot K_{\text{ser,con},i}} < \frac{1 - \widetilde{N}_l \cdot \frac{3m-2}{m^2}}{1 - \widetilde{N}_l \cdot \frac{m-2}{m^2}} \tag{2.3}$$

Where:

m is the number of panels.

$n_{v,j,i}$ is the number of vertical connectors.

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

$K_{ser,con,i}$ is the stiffness of the vertical joint.

$\widetilde{N}_l$ is the dimensionless vertical load on the shear wall that is calculated by Formula (2.4).

$$\widetilde{N}_l = \frac{N_{i,Ed} l_i}{2M_{i,Ed}} \tag{2.4}$$

Where:

$M_{i,Ed}$ is the moment at the bottom of the i^{th} story.

$N_{i,Ed}$ is the design vertical force acting at the top of the i^{th} story.

For multi-story, the contribution from the rotation of the floor below must be included when calculating the total lateral deflection. This is done by using Formula (2.5), (2.6) and (2.7).

$$u_{\theta,i} = \frac{\theta_{i-1}}{H_i} \quad \text{for} \quad i \geq 1 \tag{2.5}$$

with

$$\theta_{i-1} = \theta_{i-2} + \varphi_{R,i-1} \tag{2.6}$$

$$\varphi_{R,i-1} = \frac{u_{R,i-1}}{H_{i-1}} \tag{2.7}$$

Where:

$\varphi_{R,i-1}$ is the rotation due to the rocking of the $i-1^{th}$ storey.

θ_{i-1} is the rotation at the top of the shear wall at the $i-1^{th}$ story.

θ_{i-2} is the rotation at the top of the $i-2^{th}$ storey.

The total lateral deflection is found by summing up the deflection of each floor.

2.3.2 Exact solutions for CP, SW and INT For One Story

For one-story shear walls, there is an exact solution for the deflection. The method calculates the deflection using the moment at the bottom of the wall. The effective moment is the shear force times the height:

$M_{1,Ed} = V_{1,Ed} h_1$

CP mode:

$$u_{R,1,CP} = \max\left\{\left(\frac{M_{1,Ed}}{K_{R,1,CP}} - \frac{N_{1,Ed} \cdot l_j}{2 \cdot K_{R,1,CP}}\right) \cdot h_1; 0\right\} \tag{2.8}$$

SW mode:

$$u_{R,1,SW} = \max\left\{\left(\frac{M_{1,Ed}}{K_{R,1,SW}} - \frac{N_{1,Ed}}{2 \cdot K_{ser,anc,1} \cdot l_1}\right) \cdot h_1; 0\right\} \tag{2.9}$$

Where:

$u_{R,1,CP}$ is the inter-story lateral displacement due to the rocking of the wall in CP mode.

$u_{R,1,SW}$ is the inter-story lateral displacement due to the rocking of the wall in SW mode.

l_j is the width of one panel.

$K_{R,1,CP}$ is the rocking stiffness of the shear wall at the 1^{st} storey in CP mode. Calculated using Formula (2.10).

$K_{R,1,SW}$ is the rocking stiffness of the shear wall at the 1^{st} storey in SW mode. Calculated using Formula (2.11).

$$K_{R,1,CP} = \frac{K_{ser,anc,1} + (m-1) \cdot n_{vj,1} \cdot K_{ser,con,1}}{m^2} l_1^2 \tag{2.10}$$

$$K_{R,1,SW} = \left[\frac{1}{K_{ser,anc,1}} + \frac{(m-1)}{n_{vj,1} \cdot K_{ser,con,1}} \right]^{-1} \cdot l_1^2 \tag{2.11}$$

To be able to calculate the lateral deflection in the intermediate mode, the number of panels that are lifting needs to be found using Formula 2.12. This value needs to be rounded up to the nearest whole number. Using this rounded number, the deflection can be found using Formula 2.14.

$$J = \frac{K_h(m-1) + K_v(1-2m)}{K_h(2m+2) - 2K_v} + \frac{\sqrt{h_1 V_{1,Ed}(K_h^2(-8m-8) + K_h K_v(8m+16) - 8K_v^2) + b^2 q_{Ed}(K_h^2(5m^2+2m+1) + K_h K_v(-8m^2-2m-2) + K_v^2(4m^2+1))}}{b\sqrt{q_{Ed}}(K_h(2m+2) - 2K_v} \tag{2.12}$$

$$j^* = Roundup(J) \tag{2.13}$$

$$u_{IN} = \frac{h_1(2h_1 K_h V_{1,Ed} j^* - 2h_1 K_h V_{1,Ed} + 2h_1 K_v V_{1,Ed} + K_h b^2 (j^*)^2 q_{Ed} - K_h b^2 j^* m q_{Ed})}{2K_v b^2 (K_h j^* m + K_h j^* - K_h m - K_v j^* + Kvm)} + \frac{h_1(-K_h b^2 j^* q_{Ed} + K_h b^2 m q_{Ed} - K_v b^2 (j^*)^2 q_{Ed} + K_v b^2 j^* q_{Ed} - K_v b^2 m q_{Ed})}{2K_v b^2 (K_h j^* m + K_h j^* - K_h m - K_v j^* + K_v m)} \tag{2.14}$$

Where:

$K_v = K_{ser,con,1} n_{v,j,1}$ is the total vertical shear stiffness of the wall.

j^* is the first panel touching the ground at its trailing corner.

u_{IN} is the lateral inter-story deflection calculated using the exact number of lifting panels.

2.3.3 Interpolation In INT For One Story

The approximation for intermediate mode can be done for one-story walls. Finding the lateral deflection $U_{R,i}$ for the intermediate mode Annex R proposes to interpolate between the deflection from CP mode and SW mode using Formula 2.8 and 2.9 and the boundaries for CP mode and SW given in Formula 2.2.

When using linear interpolation it is necessary to have the unknown x and y, and the known x_0, x_1, y_0, and y_1. Since the boundaries found in Formula 2.2 depend on the dimensionless vertical force $\widetilde{N}_l$, it can change and, therefore, not give a good interpolation. Instead, interpolating with respect to the horizontal force when the wall switches from CP to INT mode and INT to SW mode can be done. The horizontal boundary force can be found using Formula 2.15 and 2.16. Finding the lateral deflection at these boundaries using this horizontal force will give the values that can be interpolated between. Those are found by using Formula 2.17 and 2.18.

$$V_{CP} = \frac{b^2 q(K_h(m-2) + K_v(2-3m))}{2h_1(K_h - K_v)} \tag{2.15}$$

$$V_{SW} = \frac{b^2 m^2 q(K_h(2-m) - K_v)}{2h_1(K_h - K_v)} \tag{2.16}$$

$$u_{CP} = \frac{h_1(2h_1 V_{CP} - b^2 mq)}{2b^2(K_h + K_v(m-1))} \tag{2.17}$$

$$u_{SW} = \frac{h_1(h_1 V_{SW}(K_h(2m-2) + 2K_v) - K_v b^2 m^2 q)}{2K_h K_v b^2 m^2} \tag{2.18}$$

Where:

V_{CP} is the horizontal force from CP to INT mode.

V_{SW} is the horizontal force from INT to SW mode.

u_{CP} is the horizontal deflection for the CP mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry.

u_{SW} is the horizontal deflection for the SW mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry.

2.3.4 Extending To Multistory

The method can be extended to calculate the deflection due to rocking for multistory walls as well. The input load is the moment at the bottom of the wall. The method utilizes the same Formulas as for one-story walls to find the exact solutions for the deflection, however, now looking at the i^{th} story instead of the 1^{st}. Explanations of the Formulas are described in the previous Chapters.

The moment at the base of the wall is equal to the moment at the top of the wall plus the shear force times the height of the wall:

$M_{i,Ed} = M_{i,top,Ed} + V_{i,Ed} h_i$

Exact solutions are found using Formula 2.19-2.25

CP mode:

$$u_{R,i,CP} = \max\left\{\left(\frac{M_{i,Ed}}{K_{R,i,CP}} - \frac{N_{i,Ed}\cdot l_j}{2\cdot K_{R,i,CP}}\right)\cdot h_i; 0\right\} \tag{2.19}$$

SW mode:

$$u_{R,i,SW} = \max\left\{\left(\frac{M_{i,Ed}}{K_{R,i,SW}} - \frac{N_{i,Ed}}{2\cdot K_{ser,anc,i}\cdot l_i}\right)\cdot h_i; 0\right\} \tag{2.20}$$

Where:

$u_{R,i,CP}$ is the inter-story lateral displacement due to the rocking of the wall in CP mode.

$u_{R,i,SW}$ is the inter-story lateral displacement due to the rocking of the wall in SW mode.

$K_{R,i,CP}$ is the rocking stiffness of the shear wall at the i^{th} storey in CP mode. Calculated using Formula (2.21).

$K_{R,i,SW}$ is the rocking stiffness of the shear wall at the i^{th} storey in SW mode. Calculated using Formula (2.22).

$$K_{R,i,CP} = \frac{K_{ser,anc,i} + (m-1)\cdot n_{vj,i}\cdot K_{ser,con,i}}{m^2} l_i^2 \tag{2.21}$$

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

INT mode:

$$J = \frac{K_h(m-1) + K_v(1-2m)}{K_h(2m+2) - 2K_v} + \frac{\sqrt{h_i V_{i,Ed}(K_h^2(-8m-8) + K_h K_v(8m+16) - 8K_v^2) + b^2 q_{Ed}(K_h^2(5m^2+2m+1) + K_h K_v(-8m^2-2m-2) + K_v^2(4m^2+1))}}{b\sqrt{q_{Ed}}(K_h(2m+2) - 2K_v} \tag{2.23}$$

$$j^* = Roundup(J) \tag{2.24}$$

$$u_{IN} = \frac{h_i(2h_i K_h V_{i,Ed} j^* - 2h_i K_h V_{i,Ed} + 2h_i K_v V_{i,Ed} + K_h b^2 (j^*)^2 q_{Ed} - K_h b^2 j^* m q_{Ed})}{2K_v b^2(K_h j^* m + K_h j^* - K_h m - K_v j^* + Kvm)} + \frac{h_i(-K_h b^2 j^* q_{Ed} + K_h b^2 m q_{Ed} - K_v b^2 (j^*)^2 q_{Ed} + K_v b^2 j^* q_{Ed} - K_v b^2 m q_{Ed})}{2K_v b^2(K_h j^* m + K_h j^* - K_h m - K_v j^* + Kvm)} \tag{2.25}$$

Where:

$K_v = K_{ser,con,i} n_{v,j,i}$ is the total vertical shear stiffness of the wall.

j^* is the first panel touching the ground at its trailing corner.

u_{IN} is the lateral inter-story deflection calculated using the exact number of lifting panels.

The deflection in the intermediate (INT) mode can be found using interpolation for multistory as well. The values to interpolate between are found by using Formulas 2.26 - 2.29.

$$V_{CP} = \frac{b^2 q(K_h(m-2) + K_v(2-3m))}{2h_i(K_h - K_v)} \tag{2.26}$$

$$V_{SW} = \frac{b^2 m^2 q(K_h(2-m) - K_v)}{2h_i(K_h - K_v)} \tag{2.27}$$

$$u_{CP} = \frac{h_i(2h_i V_{CP} - b^2 mq)}{2b^2(K_h + K_v(m-1))} \tag{2.28}$$

$$u_{SW} = \frac{h_i(h_i V_{SW}(K_h(2m-2) + 2K_v) - K_v b^2 m^2 q)}{2K_h K_v b^2 m^2} \tag{2.29}$$

Where:

V_{CP} is the horizontal force from CP to INT mode.

V_{SW} is the horizontal force from INT to SW mode.

u_{CP} is the horizontal deflection for the CP mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry.

u_{SW} is the horizontal deflection for the SW mode based on the anchor stiffness, vertical shear connector stiffness, vertical load, and wall geometry.

2.4 Arup

Arup has proposed another method for analyzing the lateral deflection of multistory segmented CLT shear walls without openings. This whole chapter is from Smith and Lawrence (2023). The model consists of 3 steps. Step 1 simplifies the vertical loads. Step 2 identifies the response mode, and step 3 finds the solutions. The method finds the lateral deflection of the i^{th} story and the wall rotation relative to the ground or the floor below. The method does not find the rotation of a single panel but the wall as a whole. The rotation is calculated by imagining a beam parallel to the ground when the wall is not deflecting. When the panel on the leading edge start lifting, the beam will no longer be parallel to the ground, and the angle of the beam relative to the ground will be the rotation.

The actual load distribution on the top and bottom of a multi-story segmented CLT wall is very complex and hard to derive. Instead, the method calculates the summed forces at each story (red forces in Figure 2.7). From these forces, a set of simplified actions on the wall are calculated (green forces in Figure 2.7). The loads on top of the wall can be simplified to a uniformly distributed load q with a distance l_{top} from the leading edge, a tension force Q at the same edge, a downward facing point force at the trailing edge and a horizontal force $V_{i,Ed}$ at the leading edge. The loads at the bottom of the wall can be simplified to a tension force T at the leading edge, a uniformly distributed load starting from a distance l_{bot} from the leading edge, a vertical upwards force at the trailing edge and a horizontal force $V_{i,Ed}$. For calculating Q, q and l_{top} Formula (2.30), (2.33) and (2.34) may be used.

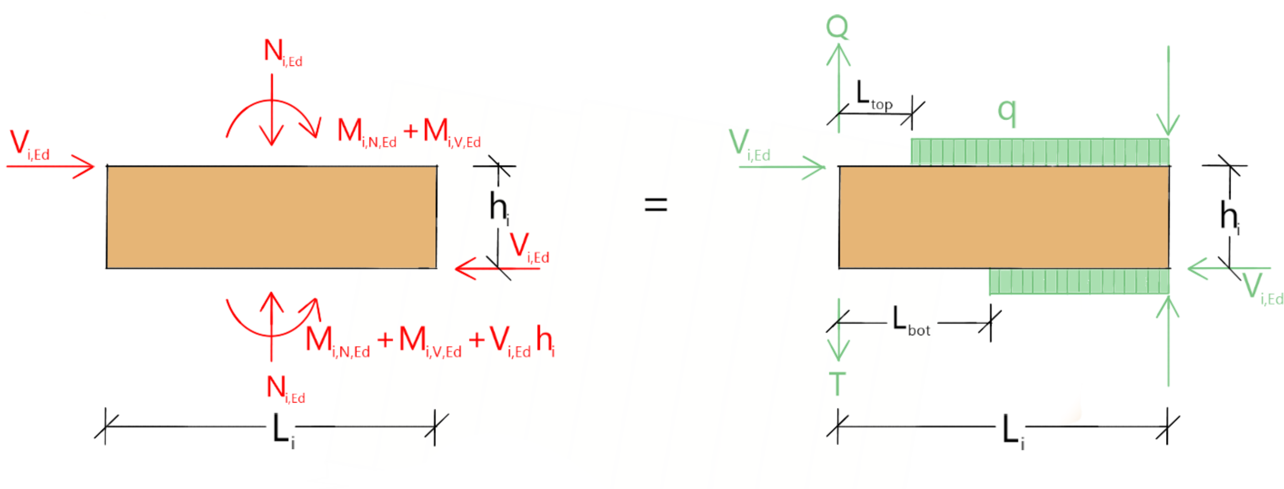


Figure 2.7: Simplified loads. Drawn after Figure 2 in Smith and Lawrence (2023)

$$Q = \begin{cases} \dfrac{M_{i,V,Ed}}{l_i} - N_A & \text{if } l_{top} = 0 \\ \dfrac{M_{i,V,Ed}}{l_i} - N_A - \dfrac{q l_i}{2} & \text{if } l_{top} = l_i \\ T_{above} & \text{otherwise} \end{cases} \tag{2.30}$$

Where:

- T_{above} is the tie-down force at the bottom of the floor's leading edge above (if present) under the same set of loads, 0 otherwise.
- N_A and N_B are calculated according to the equations below:

$$N_A = \begin{cases} -2M_{i,N,Ed}/l_i & \text{if } M_{i,N,Ed} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.31}$$

$$N_B = \begin{cases} 2M_{i,N,Ed}/l_i & \text{if } M_{i,N,Ed} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.32}$$

$$q = \frac{N_{i,Ed} - N_A - N_B}{l_i} \tag{2.33}$$

$$l_{top} = \begin{cases} 0 & \text{if } M_{i,V,Ed} \leqslant N_A l_i + T_{above} l_i \\ l_i & \text{if } M_{i,V,Ed} \geqslant N_A l_i + T_{above} l_i + \dfrac{q l_i^2}{2} \\ l_i - \sqrt{l_i^2 - \dfrac{2\left(M_{i,V,Ed} - N_A l_i - T_{above} l_i\right)}{q}} & \text{otherwise} \end{cases} \tag{2.34}$$

The method predicts that there will be different deformations depending on the stiffness of the hold-down and the vertical joint connections. It separates into two groups:

$K_h \geq K_v$: This is called the discrete panel model. This predicts the panels are in contact with the ground below but can rotate about their trailing edge. The calculation of the deflection u_R can be solved in terms of $V_{i,Ed}$.

$K_h < K_v$: This is the shear beam model, and this predicts that the panels can still rotate about their trailing edge, but some panels might lift entirely off the ground at the leading edge. In this model, the stiffness of the vertical joint is smeared out to create a continuous shear beam. To do so, the hold-down stiffness K_h is partitioned into a two-component spring acting in series. The first spring component is identical to the vertical joint stiffness K_v. The second spring component is the residual stiffness (K_{res}). It acts at the bottom of the leading edge of the shear beam. K_{res} is calculated by the Formula (2.35). This results in a shear beam which is assumed to have a large effective longitudinal bending stiffness EI, so the shear stiffness mainly governs the behavior.

$$K_{res} = \frac{1}{\frac{1}{K_h} - \frac{1}{Kv}} \tag{2.35}$$

There are identified nine response modes.

If $K_h < K_v$: Response mode A - E.

- A - No lateral deflection. Acting as stiff.
- B - Small lateral deflection, but no lift up at the leading edge.
- C - There is a lift-off at the leading edge, but the length of the lift-off l_{bot} is less than the length of the wall l_i minus the width of one panel b.
- D - There is a lift-off, and the length of the lift-off l_{bot} is larger than the entire length of the wall l_i minus the width of one panel b but lesser than the whole length of the wall.
- E - The wall is rocking about a single center of rotation, the trailing edge. $l_{bot} = l_i$.

If $K_h \geq K_v$: Response mode F - I.

- F - Acting as stiff and no deflection. The whole length of the wall is in compression ($L_{top} = 0$).
- G - The wall is deflecting, and the whole length of the wall is in compression ($L_{top} = 0$).
- H - The wall is acting stiff, and there is no deflection. A part of the wall is not in compression ($L_{top} > 0$).
- I - The wall has a deflection, and a part of the wall is not in compression ($L_{top} > 0$).

The next step is to find the response mode. To find the mode, Table 2.1 must be used if $K_h < K_v$ is true, and Table 2.2 if $K_h \geq K_v$ is true.

Table 2.1 : Response mode if $\mathbf{K_h} < \mathbf{K_v}$

Load condition	$\mathbf{V_{i,Ed}}$ **condition**	**Mode**
$Q < 0$	$V_{i,Ed} \leqslant \frac{ql_ib}{2h_i} - \frac{Qb}{h_i}$	A
	$\frac{ql_ib}{2h_i} - \frac{Qb}{h_i} < V_{i,Ed} \leqslant \frac{ql_ib}{2h_i} - \frac{Ql_i}{h_i}$	B
	$\frac{ql_ib}{2h_i} - \frac{Ql_i}{h_i} < V_{i,Ed} \leqslant \frac{qK_{res}l_i^2(l_i - b)}{2K_vbh_i} + \frac{ql_i^2}{2h} - \frac{Ql_i}{h_i}$	C
	Otherwise	D
$Q \geqslant 0$ &	$V_{i,Ed} \leqslant \frac{qb\,(l_i - l_{top})}{2h_i} - \frac{Qb}{h_i}$	A
$0 \leqslant l_{top} \leqslant l_i - b$	$\frac{qb\,(l_i - l_{top})}{2h_i} - \frac{Qb}{h_i} < V_{i,Ed}$ & $V_{i,Ed} \leqslant \frac{qK_{res}l_i(l_i - l_{top} - b)(l_i + l_{top})}{2K_vbh_i} + \frac{q(l_i - l_{top})^2}{2h_i} - \frac{Ql_i}{h_i}$	C
	Otherwise	D
$Q \geqslant 0$ &	$V_{i,Ed} \leqslant \frac{q\,(l_i - l_{top})^2}{2h_i} - \frac{Qb}{h_i}$	A
$l_i - b < l_{top} \leqslant l_i$	Otherwise	E

Table 2.2 : Response mode if $\mathbf{K_h} \geqslant \mathbf{K_v}$:

Q condition	$\mathbf{V_{i,Ed}}$ **condition**	**Mode**
$Q \leqslant 0$	$V_{i,Ed} \leqslant \frac{ql_ib}{2h_i} - \frac{Qb}{h_i}$	F
	Otherwise	G
Q > 0	$V_{i,Ed} \leqslant \frac{q\,(l_i - l_{top})\,b}{2h_i}$	H
	Otherwise	I

The length not in compression l_{bot}, tension in the hold down T, the force in the connections between the panels F_v, the horizontal deflection $u_{R,i}$ and the rotation of the top of the wall relative to the wall below $\phi_{R,i}$ are found by Tables 2.3, 2.4, 2.5, 2.6 and 2.7.

Table 2.3 : Solutions for $\mathbf{l_{bot}}$ depending on response mode

Mode	$\mathbf{l_{bot}}$
A, B	l_{top}
C	$-\frac{b\left(K_{res}l_i+2K_vl_i-K_vb\right)}{2\left(K_{res}l_i-K_vb\right)}+\frac{K_vb}{K_{res}l_i-K_vb}\times \sqrt{\left(\frac{K_{res}l_i+2K_vl_i-K_vb}{2K_v}\right)^2+\frac{K_{res}l_i-K_vb}{K_vb}\left(\frac{2(V_{i,Ed}h_i+Ql_i)}{q}-l_ib+l_{top}\left(2l_i-l_{top}\right)+\frac{K_{res}l_il_{top}\left(l_{top}+b\right)}{K_vb}\right)}$
D, E	l_i
F, G, H, I	l_{top}

Table 2.4 : Solutions for **T** depending on response mode

Mode	**T**
A, B, F	0
C	$\frac{qK_{res}}{2K_vb}\left(l_{bot}-l_{top}\right)\left(l_{bot}+l_{top}+b\right)$
D, E	$\frac{V_{i,Ed}h_i}{l_i}+Q-\frac{q\left(l_i-l_{top}\right)^2}{2l_i}$
G	$\frac{K_hb\left(V_{i,Ed}h_i+Qb-ql_ib/2\right)}{K_hb^2+K_vb(l_i-b)}$
H	Q
I	$Q+\frac{K_hb\left(V_{i,Ed}h_i-q(l_i-l_{top})b/2\right)}{K_hb^2+K_vb(l_i-b)}$

Table 2.5 : Solutions for $\mathbf{F_v}$ depending on response mode

Mode	$\mathbf{F_v}$
A, F, H	0
B	$\frac{V_{i,Ed}h_i}{l_i} - \frac{qb}{2}$
C	$\max\begin{cases} T - Q + q\left(l_{bot} - l_{top}\right) \\ Q - T \end{cases}$
D	$\max\begin{cases} T - Q + q\left(l_i - l_{top} - b\right) \\ Q - T \end{cases}$
E	$\max\begin{cases} T - Q \\ Q - T \end{cases}$
G, I	$\frac{K_v T}{K_h}$

Table 2.6 : Solutions for $\mathbf{u_{R,i}}$ depending on response mode

Mode	$\mathbf{u_{R,i}}$
A, F, H	0
B	$\frac{h_i}{K_v b}\left(\frac{V_{i,Ed}h_i}{l_i} + \frac{Qb}{l_i} - \frac{qb}{2}\right)$
C	$\frac{h_i}{K_v b}\left(T - \frac{Q\left(l_i - b\right)}{l_i} + q\left(l_{bot} - l_{top}\right)\right)$
D	$Th_i\left(\frac{1}{K_{res}l_i} + \frac{1}{K_v b}\right) - \frac{Qh_i\left(l_i - b\right)}{K_v b l_i} + \frac{qh_i\left(l_i - l_{top} - b\right)\left(l_i - l_{top}\right)}{2K_v b l_i}$
E	$Th_i\left(\frac{1}{K_{res}l_i} + \frac{1}{K_v b}\right) - \frac{Qh_i\left(l_i - b\right)}{K_v b l_i}$
G, I	$\frac{F_v h_i}{K_v b}$

Table 2.7 : Solutions for $\varphi_{\mathbf{R,i}}$ depending on response mode

Mode	$\varphi_{\mathbf{R,i}}$
A, B, C, D, E	$\dfrac{T}{K_{res}l_i}$
F, G, H, I	0

The same approach as in Annex R is used to calculate the deflection for a multistory building. The rotation of the floor or floors below multiplied by the height of the story, shown in Figure 2.8, is used to obtain the contribution from the rotation.

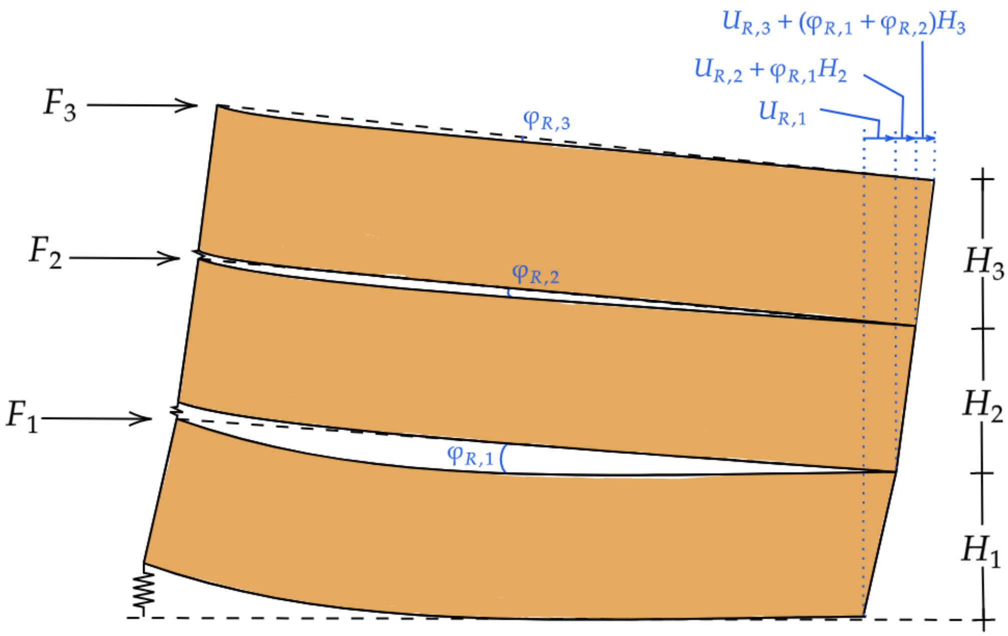


Figure 2.8: Assumed deflection of a multistory wall. Drawn after Figure 7 in Smith and Lawrence (2023).

2.5 Finite element method

The finite element method (FEM) is a numerical analysis for solving different engineering problems. FEM works by calculating approximate solutions to equations. The solution is not 100% exact but still represents the actual solution well. It divides an area into smaller areas and solves them using interpolation functions. These functions are determined based on the field variables' values at specific points referred to as nodes or nodal points. These points are often located at the element's boundaries, where it is connected to other elements. The divided area is called a mesh. The finer the mesh, the more accurate the solutions are. (Jagota et al., 2013)

2.5.1 SAP2000

SAP2000 is a program for calculating with the finite element method. This program uses an analysis engine called SAPfire. This engine allows for many different types of analysis, such as linear, nonlinear, static, dynamic, and many more. (Computers & Structures, 2017).

Area objects in SAP2000 are shell elements that can be plate, membrane, or full shell elements. They are used to model areas such as walls and floors. Area objects can be given thickness, stiffness, other material properties, and loads. (Computers & Structures, 2017).

Frames in SAP2000 are used to model beams, columns, braces, and trusses. Frames are modeled as a straight line between two points. Frame elements can be loaded with many different types of loads, such as gravity, uniformly distributed, and point loads. (Computers & Structures, 2017).

Links/support elements are used in SAP2000 to connect two joints. Links connect two joints and support elements are one-jointed springs connected to the ground. Links and support elements are defined in the same way. They are composed of six springs, one for each of its degrees of freedom. There are different types of links used for different behaviors. The gap link does only have stiffness when in compression. The hook link does only have stiffness when in tension. Linear links can have stiffness in all directions. The gap and hook links are active only when running a nonlinear analysis. (Computers & Structures, 2017).

2.5.2 Open Applications Programming Interface

Open Applications Programming Interface (OAPI) allows a third-party program to control software such as SAP2000 with a programming language. The CSI Application Programming Interface allows major programming languages like Python, MATLAB, and C# to access SAP2000. OAPI creates a link between the SAP2000 and the third-party program, allowing for a two-way exchange of information. This way results from analyses from the SAP2000 can be extracted and used in the third-party program. (Computers & Structures, 2022).

This makes it possible to make a Python script that can automate the modeling and analysis. The scripts can run many analyses with different parameters. This makes it easy to make many results.

3 Method

The following chapter will delve into the methodology of the analytic and numerical analysis employed. Rotating the different parameters makes it possible to check how they affect the displacement. In chapter 3.4, a proposal for validating the formulas used in the analytical approach from Arup is presented. This is done by comparing results from the analytical model and a numerical approach using SAP2000. Finally, in Chapter 3.5, an approach for validating the analytical method from Arup for both two- and three-story behavior is presented. This is done by comparing it with a multistory model made in SAP2000.

3.1 Python Script For Analytical Analysis

Python codes are made to easily change the different parameters used in the Formulas from chapter 2.3 and 2.4. These scripts calculate the displacement due to kinematic rocking for certain load situations. The scripts can print out graphs showing the displacement vs. horizontal load or give the displacement for a set horizontal load. Two of the Python codes used are provided in Appendix C and Appendix D.

3.2 Modeling For Numerical Analysis

For the numerical analysis, SAP2000 models are used. For the modeling of one-story walls, the FEM model is made using OAPI. This makes it possible to automate the analysis of different walls and cases and plot the results. The scripts used are based on the script in Appendix C in Bjørkedal and Saevareid (2022) and are modified for this specific project.

Table 3.1 : The different types of structural elements and what is used to model them in SAP2000.

Structural element	Model element	Parameter	Contributing
Panel	Shell element	Youngs modulus	NO
Hold-down	Hook link	Stiffness	YES
Shear connectors between panels	Linear link	Stiffness	YES
Angle brackets	Linear link	Stiffness	NO
Compression link	Gap link	Stiffness	NO

NO - The stiffness is so high that contributions to the deflection are negligible.
YES - The stiffness is given, and the element contributes to the deflection.

The modeling must be done in a certain way to make the FEM model only find the displacement due to kinematic rocking. Table 3.1 shows the different structural elements and the elements inside SAP2000 used to model them. It also presents if they contribute to the deflection due to kinematic rocking. The CLT panels are modeled with a high Youngs modulus and constraints on the top corners of the panels that give equal displacement for each point. This makes the wall act rigid, so the in-plane bending and shear can be neglected. Linear links are used to simulate angle brackets with very high stiffness and are modeled on each panel to prevent the wall from sliding. Shear connections are made by using linear links between the panels. They only contribute with stiffness in the vertical direction. Hook links are used to simulate the effect of hold-downs. Hook links do only have tensile stiffness. Gap links only resist compressive forces and are used to prevent the wall from moving downwards. They are modeled on the corners and the middle of each panel. Restraints below the wall are modeled for the Gap, hook, and linear links on the bottom of the wall to be connected to. Figure 3.1 shows how a one-story wall is modeled in SAP2000. Note that the distance between the panels is not modeled this large; it is like this to visualize the modeling.

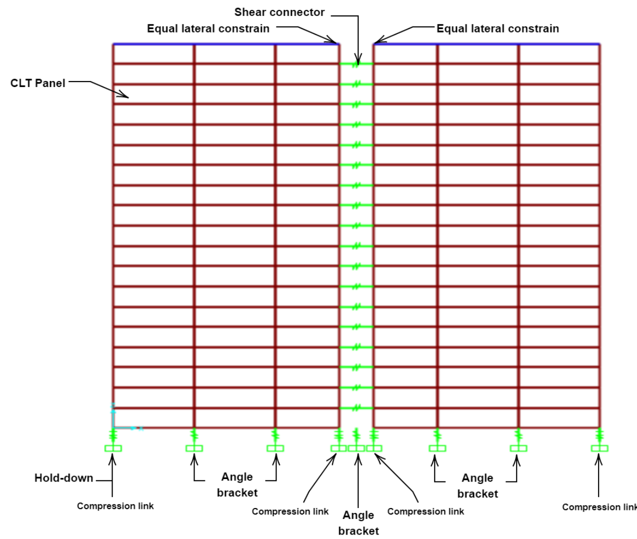


Figure 3.1: Visualisation of modeling one-story wall in SAP2000.

For a multistory wall with a floor between the stories, the compression links, hold-down and angle brackets are modeled as springs in series. This will actively affect the actual stiffness of the connections between the stories. Since the compression link and angle brackets are very stiff and only used in the FEM model and not in the analytical analysis, they are kept as is. However, it is essential to change the stiffness of the hold-down to two times what is used in the numerical analysis for the analytical analysis. How a multistory wall is modeled in SAP2000 is shown in Figure 3.2.

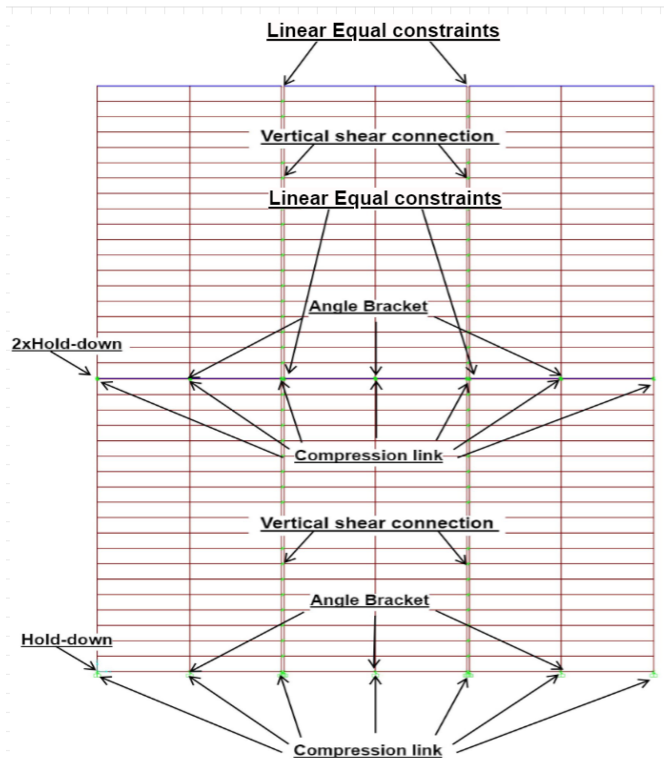


Figure 3.2: Two story model from SAP2000.

3.3 Validating Annex R For One Story

A comparison between the two proposed interpolations, calculating the exact number of panels lifting and Arup, is done to check their functionality. The parameters used in the analyses are shown in Figure3.2.

Table 3.2 : Geometry and parameters for comparison between the two Annex R methods.

Parameters	Symbol	
Hold-down stiffness	K_h (kN/m)	1,000
Vertical shear stiffness	K_v (kN/m)	10,000
Length of each panel	l_i (m)	1.4
Height of each panel	h_i (m)	2.7
Panels	m	5
Number of shear connectors	n_{con}	18
Number of floors		1

3.4 Parametric validation Of Arup For Ground Floor In Multistory Building

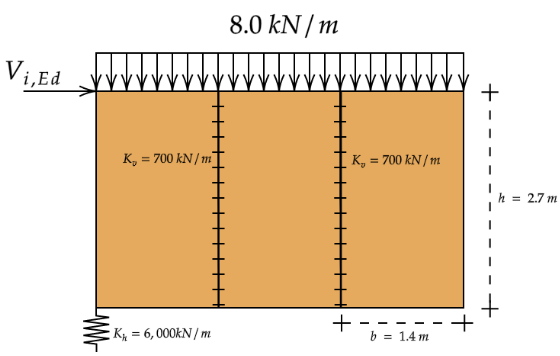
3.4.1 Ground Floor, No Wall Above

Firstly, a reconstruction of the SAP2000 model in (Casagrande et al., 2018) is made. Doing so makes it possible to ensure the model behaves as it should.

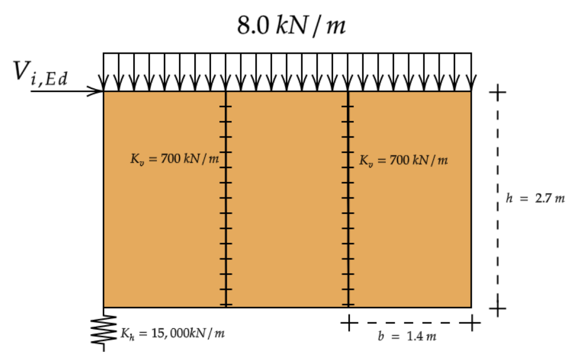
Table 3.3 : Parameters used to reconstruct Case A and B in Casagrande et al. (2018).

Paremeter	Symbol	CASE A	CASE B
Hold-down stiffness	K_h (kN/m)	6,000	15,000
Vertical shear stiffness of each connection	K_v (kN/m)	700	700
Height of panels	h (m)	2.7	2.7
Width of panels	b (m)	1.4	1.4
Number of shear connectors	n_{con}	18	18
Number of panels	m	3	3

Two cases for the validation of the method are given. Case A and B. A UDL of $8,000N/m$ along the length of the wall is applied. The horizontal load, $V_{i,Ed}$, is applied as a point load on the top corner of the left panel. This load increases from $0N$ up to $100,000N$. The hold-down is placed at the bottom of the leading edge. Angle brackets with high stiffness are placed on each panel to secure the wall from sliding. The displacement is plotted in a graph with horizontal load on the x-axis and lateral displacement on the y-axis. Figure 3.3a and Figure 3.3b show respectively CASE A and CASE B.



(a) CASE A in Casagrande et al. (2018).



(b) CASE B in Casagrande et al. (2018).

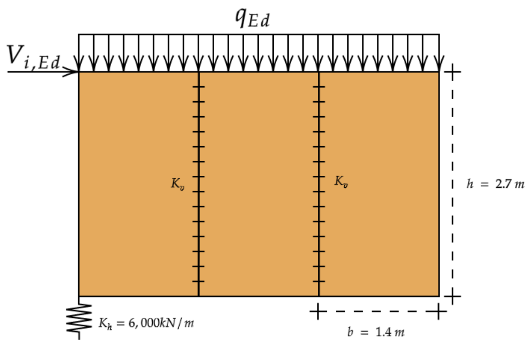
A parametric analysis is carried out to validate the Arup for one story. This analysis is very similar to the one done in Casagrande et al. (2018). However, it includes more parameters. Table 3.4 shows the parameters that stay the same, and Table 3.5 shows the parameters that are changing. Figure 3.4a - 3.4c is a visualisation of the models with respectively 3, 4, and 5 panels.

Table 3.4 : Parameters that stay the same.

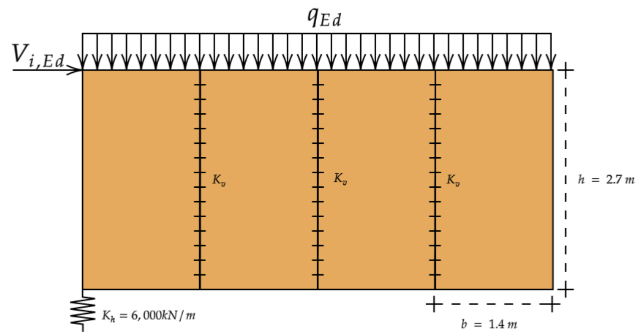
Parameters	Symbol	
Length of each panel	l_i (m)	1.4
Height of each panel	h_i (m)	2.7
Hold-down stiffness	K_h (kN/m)	6,000
Number of shear connectors	n_{con}	18

Table 3.5 : Parameters that change in both analytical and numerical analysis.

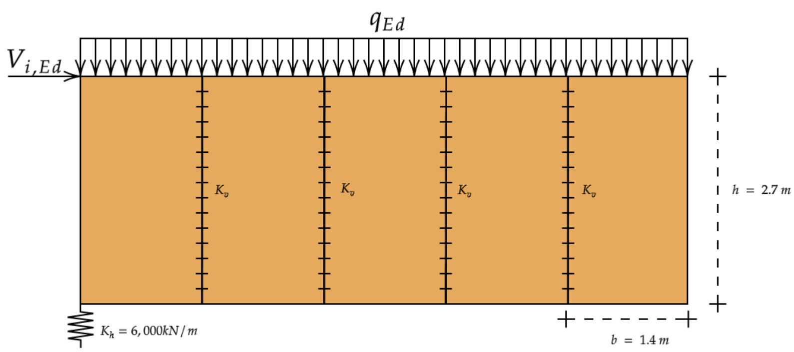
Parameters
Shear stiffness K_v:
- 3,000 kN/m
- 6,000 kN/m
- 9,000 kN/m
- 12,000 kN/m
- 15,000 kN/m
- 18,000 kN/m
Number of panels m:
- 3
- 4
- 5
Vertical load q_{Ed}:
- 1000 N/m



(a) Load case 3 panels.



(b) Load case 4 panels.



(c) Load case 5 panels.

Figure 3.4: illustration of a one-story wall.

3.4.2 Ground Floor, Wall Above

The same parameters as the one-story analysis are used for the simulated multistory analysis except for the vertical load. The vertical loads are now 1 kN/m, 4 kN/m, and 8 kN/m. Table 3.6 and 3.7 show all the parameters used.

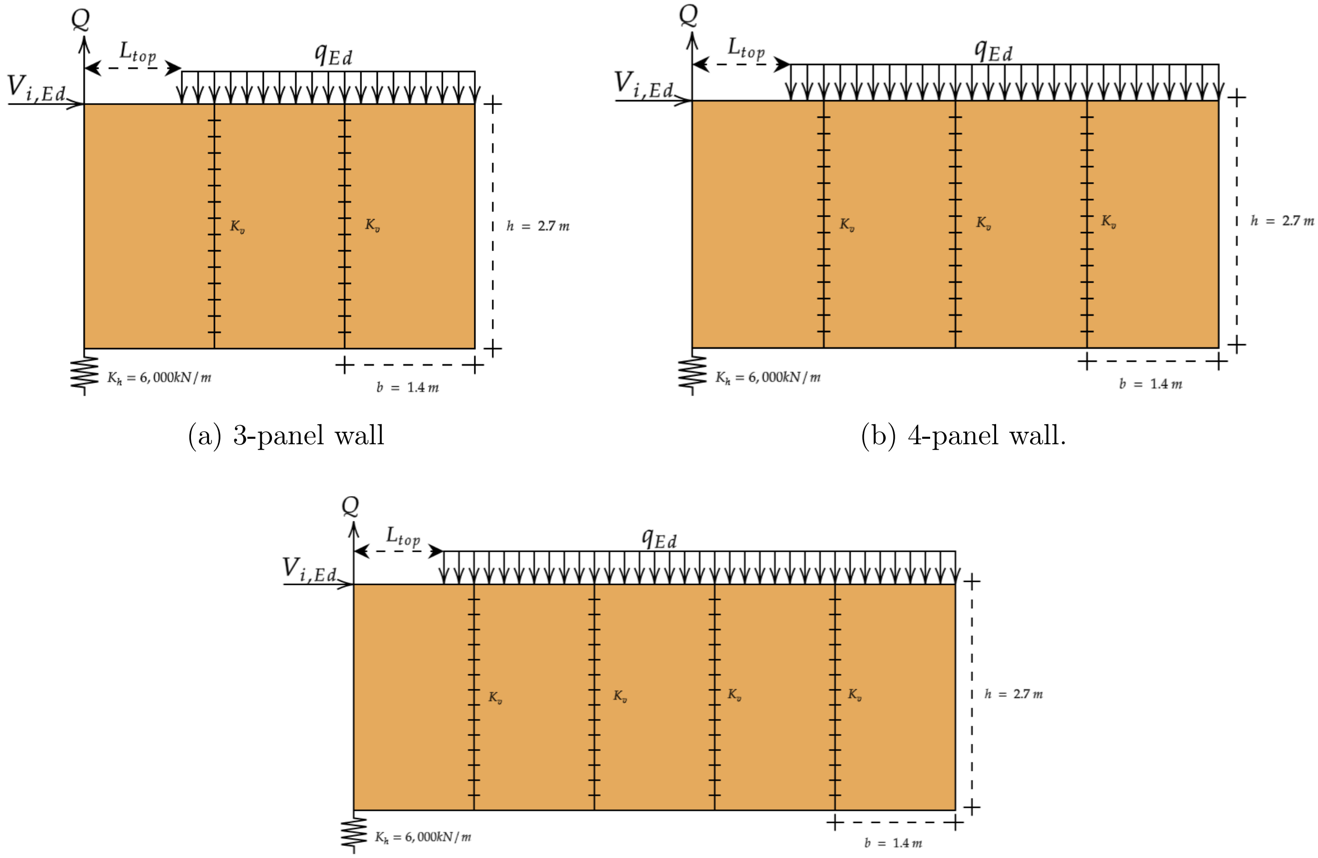
Table 3.6 : Geometry and parameters that stay the same in all calculations.

Parameters	Symbol	
Length of each panel	l_i (m)	1.4
Height of each panel	h_i (m)	2.7
Hold-down stiffness	K_h (kN/m)	6,000
Number of shear connectors	n_{con}	18

Table 3.7 : Parameters that change in both analytical and numerical analysis.

Parameters
Shear stiffness K_v:
- 3,000 kN/m
- 6,000 kN/m
- 9,000 kN/m
- 12,000 kN/m
- 15,000 kN/m
- 18,000 kN/m
Number of panels m:
- 3
- 4
- 5
Vertical load q_{Ed}:
- 1000 N/m
- 4000 N/m
- 8000 N/m

To obtain the point force Q, T_{above} needs to be calculated. T_{above} is the same as the hold down tension T from the wall above. To obtain this value, the Arup script runs one time as if it was a top story ($T_{above} = 0$ and $M_{2,V,Ed} = 0$). This returns a list with the values of T for every case of horizontal load. Then it runs again with a moment $M_{1,V,Ed} = V_{2,Ed}h_2$, which comes from the horizontal load on the wall above, and the corresponding T_{above}. The acting horizontal load will be $V_{1,Ed} = V_{i,Ed} + V_{2,Ed}$. The deflection is calculated and plotted. This also returns a list containing the wall length that is not in compression l_{top} corresponding with the horizontal load. This is needed for the SAP2000 model. A representation of the load situation that is used in the FEM model is shown in Figure 3.5a.



(a) 3-panel wall

(b) 4-panel wall.

(c) 5-panel wall.

Figure 3.5: Load situation for simulating multistory

3.5 Validation Of Arup For Multistory Behavior

3.5.1 Two-story Behavior

A walls' multistory behavior can change depending on the stiffness of the floor between the walls as explained in D'Arenzo et al. (2021). A rigid floor will give a different deflection than a very flexible floor. Neither the Arup nor Annex R model takes into account the floor's stiffness. Therefore, three cases are made for the SAP model:
- No floor between the walls.
- Flexible floor.
- Rigid floor.

The walls consist of 3 panels with a width of 1.4 m and a height of 2.7m. Table 3.8 presents the rest of the geometry and other parameters. The three load cases looked at are:
- $V_{i,Ed} = 10kN$
- $V_{i,Ed} = 20kN$
- $V_{i,Ed} = 40kN$

All cases have a vertical UDL of 1,0 kN/m on each story. For the analytical approach, the horizontal shear force $V_{i,Ed}$ must be calculated for each story based on the horizontal force acting on them. This goes for the vertical force as well. The moment due to horizontal loads needs to be found. In addition, the stiffness of the hold-down between the two stories has to be half of what's used in the numerical models with a floor since the hold-downs there are acting as springs in series.

Table 3.8 : Geometry and parameters for multistory behavior.

Parameters	Symbol	
Hold-down stiffness	K_h (kN/m)	6,000
Vertical shear stiffness	K_v (kN/m)	9,000
Length of each panel	l_i (m)	1.4
Height of each panel	h_i (m)	2.7
Panels	m	3
Number of shear connectors	n_{con}	18
Number of floors		2

For the numerical models, the horizontal force is applied to the top corner of the wall in the first and ground floor for the model without a floor, and to the top of the first floor and the floor between the first and ground floor for the model with a floor.

The two analytical methods are also analyzed by linear increasing the loads from 0 N to 40 kN to see how they behave.

3.5.2 Three-story Behavior

Four cases are made to look at the behavior of three-story walls. Case A, B, C, and D all have different parameters creating different deflections. Table 3.9 gives all the cases with their parameters. Case A, B, and C are respectively reproductions of Case 1, 2b, and 2a from (Casagrande et al., 2023). Note that not all the values given in the presentation are correct and that in Case 1, not only the kinematic rocking is contributing to the deflection. Case D is the only case where the vertical shear stiffness is higher than the hold-down stiffness.

Table 3.9 : Cases in calculating three-story behavior

		Case A			Case B			Case C			Case D		
Parameters	Symbol	GF	FF	SF	GF	FF	SF	GF	FF	SF	GF	FF	SF
Panels	m	2	2	2	3	3	3	3	3	3	3	3	3
Height of panel	h_i (m)	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5
Width of panel	b (m)	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
Floor thickness	(m)	0.1	0.1	0.1	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18
Vertical shear stiffness	K_v (kN/m)	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	9,000	9,000	9,000
Hold-down stiffness	$K_{h,i}$ (kN/m)	15,000	5,000	2,500	5,000	5,000	2,500	50,000	5,000	2,500	6,000	3,000	3,000
Vertical load	q_{Ed} (kN/m)	24	19.2	4.8	2.4	4.8	4.8	2.4	4.8	4.8	1	1	1
Horizontal load on top corner	$V_{i,Ed}$ (kN)	10	10	10	10	10	10	10	10	10	10	10	10

Note: GF = Ground floor; FF = First floor; SF = Second floor.

4 Results

This chapter will present the results from the analytical and numerical analysis. The results are generally presented as plots with the horizontal load on the x-axis and the lateral displacement on the y-axis. The exception is Figure 4.2, a reproduction of the comparison between analytical and numerical analysis given in Casagrande et al. (2018). Here the x-axis shows the displacement, and the y-axis shows the horizontal load. Other results are presented as Tables and graphs. In Chapter 4.1, the results from the comparison between the interpolation approach and the exact number of panels approach in the Annex R method are given. Chapter 4.2 shows the results from the analysis done for validating the formulas used in the analytical method made by Arup. Chapter 4.3 compares multistory behavior for the Arup and the Annex R method for two- and three-story walls.

4.1 Annex R Using Interpolation And The Exact Number Of Panels

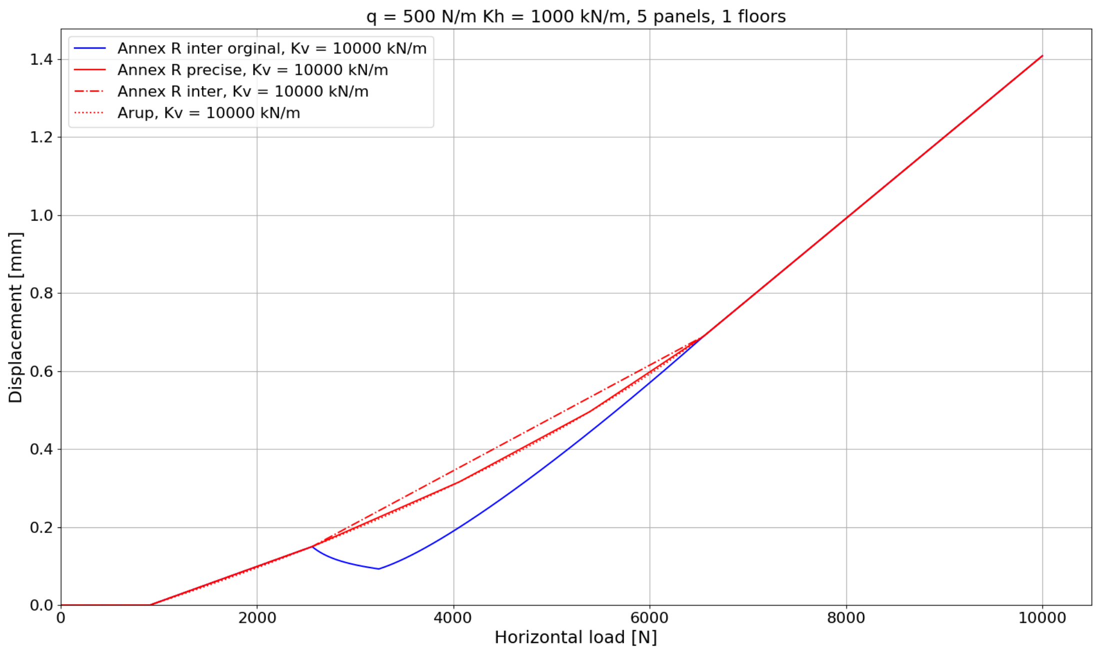


Figure 4.1: Comparison between Annex R interpolation, Annex R precise, and Arup.

Figure 4.1 shows the displacement graph of the three analytical methods. This is a one-story wall consisting of 5 panels. The hold-down stiffness is relatively flexible compared to the vertical shear stiffness. The four methods for calculating the deflection are Annex R using the originally suggested interpolation, an updated interpolation using the horizontal force, finding the exact number of panels lifting, and the Arup method. The original interpolation has a weird behavior, making the deflection go down as the horizontal force increases. The updated interpolation has a linear line when the wall is in intermediate mode. The exact number of panels lifting has small step changes when a new panel starts lifting. Arup has a smooth line throughout the whole intermediate mode. The difference between the updated interpolation and the exact number of panels lifting is very small.

4.2 Validating Formulas In Arup For One-story Wall

4.2.1 One-story, No Wall Above

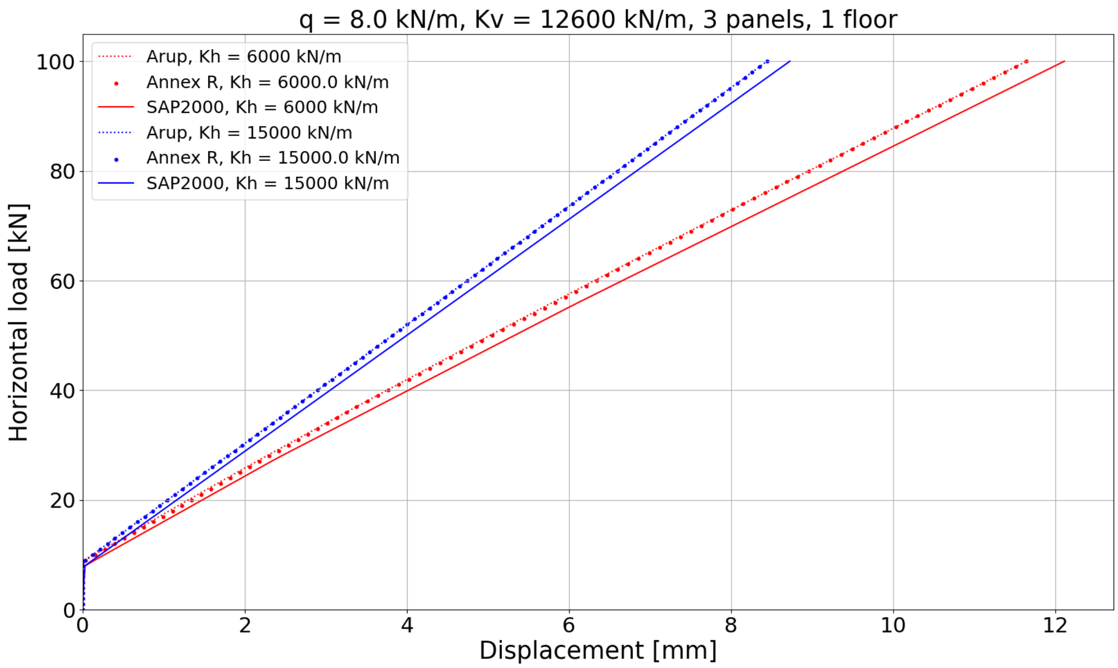
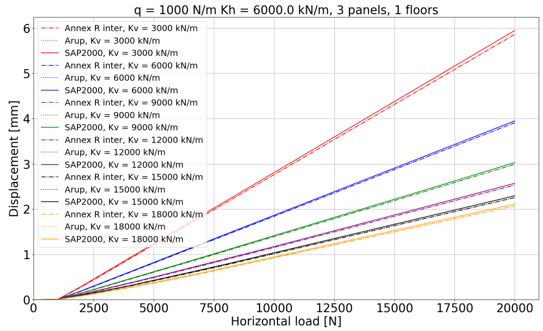
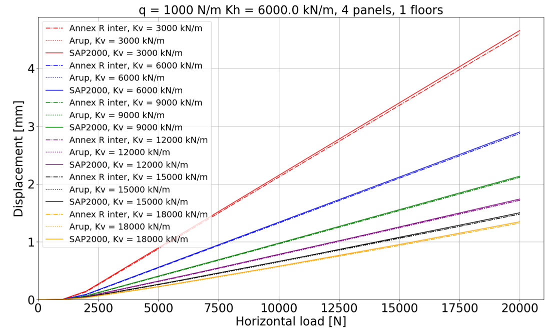


Figure 4.2: Reproduction of the analysis done in Casagrande et al. (2018).

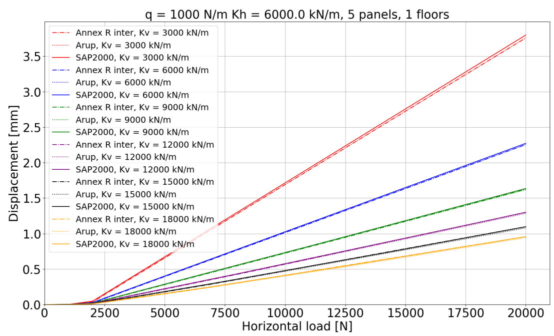
Figure 4.2 shows that a stiffer hold-down stiffness K_h gives a more rigid wall and less horizontal displacement. The wall needs the same horizontal load to start rocking, even though the higher stiffness of the hold-down gives a stiffer response. The compliance between the analytical and the numerical analysis is quite good. There is excellent agreement between the two analytical analyses' from Arup and Annex R.



(a) One story wall with 3 panels and UDL 1kN/m



(b) One story wall with 4 panels and UDL 1kN/m



(c) One story wall with 5 panels and UDL 1kN/

Figure 4.3: Lateral deflection of one-story wall with UDL along the length of the wall.

Figure 4.3a - 4.3c shows a one-story wall with panels ranging from 3 to 5 with a UDL of 1 kN/m. The graphs match up almost perfectly for the Arup, Annex R, and the numerical analysis done in SAP2000. There is a very good agreement between the different methods even though different parameters are rotated.

4.2.2 One-story, Wall Above

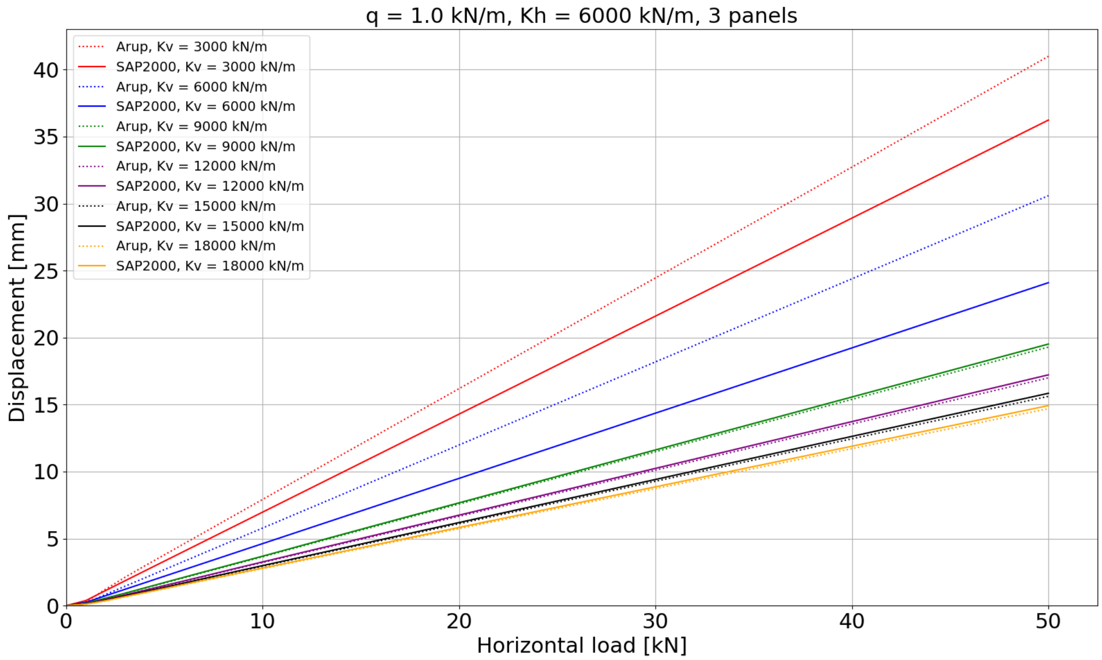


Figure 4.4: Kinematic rocking due to horizontal loads on a 3-panel wall and UDL of 1kN/m.

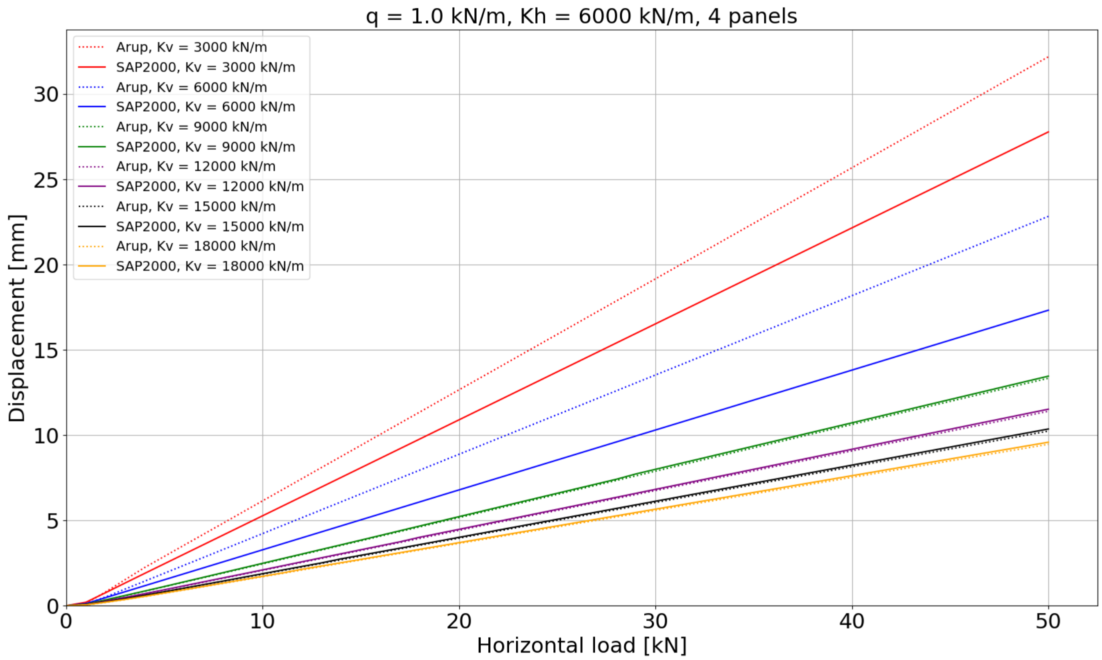


Figure 4.5: Kinematic rocking due to horizontal loads on a 4-panel wall and UDL of 1kN/m.

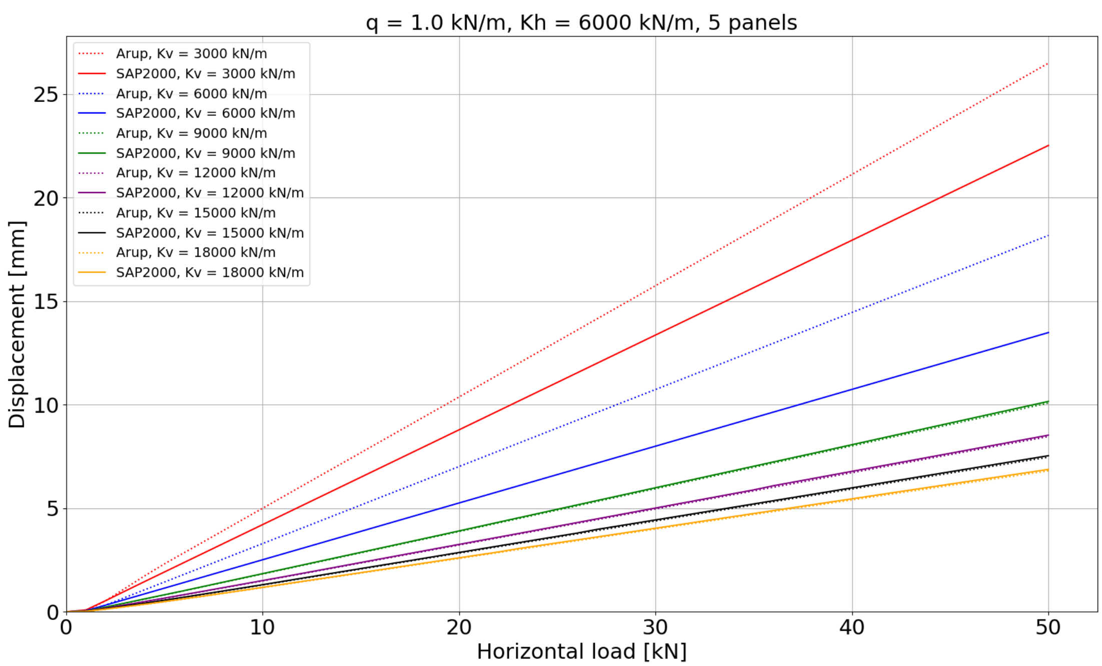


Figure 4.6: Kinematic rocking due to horizontal loads on a 5-panel wall and UDL of 1kN/m.

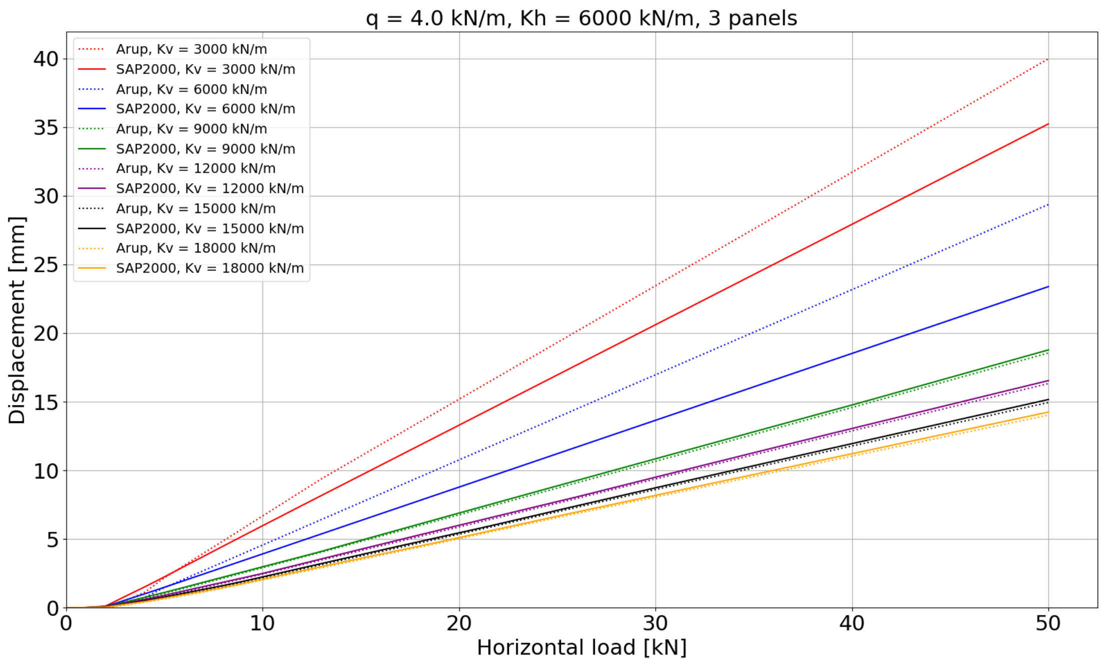


Figure 4.7: Kinematic rocking due to horizontal loads on a 3-panel wall and UDL of 4kN/m.

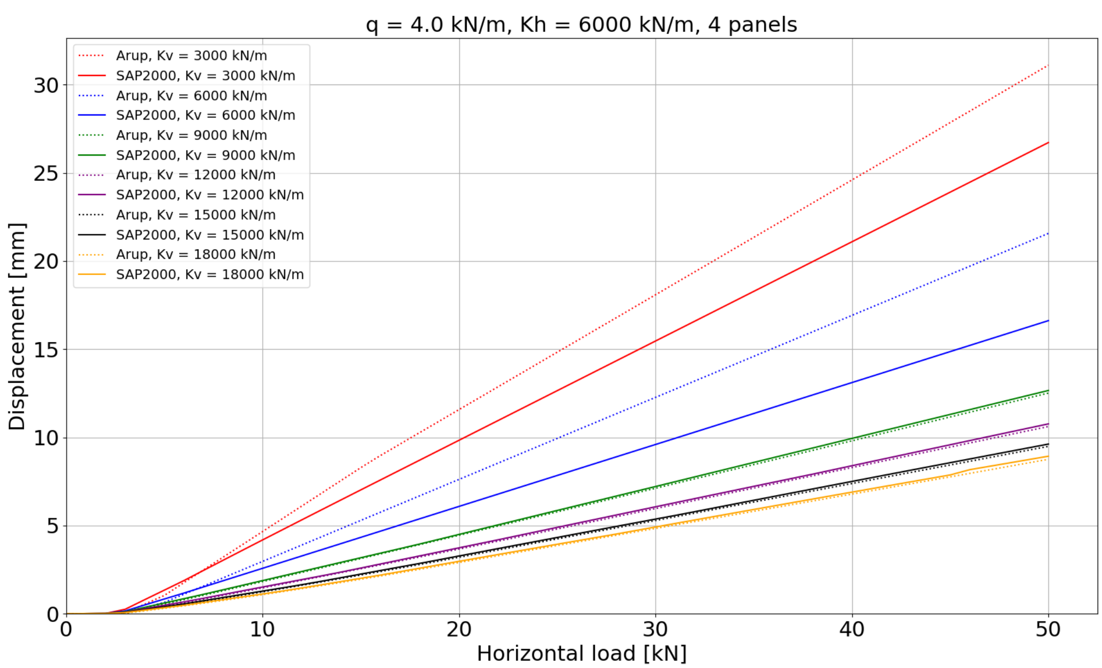


Figure 4.8: Kinematic rocking due to horizontal loads on a 4-panel wall and UDL of 4kN/m.

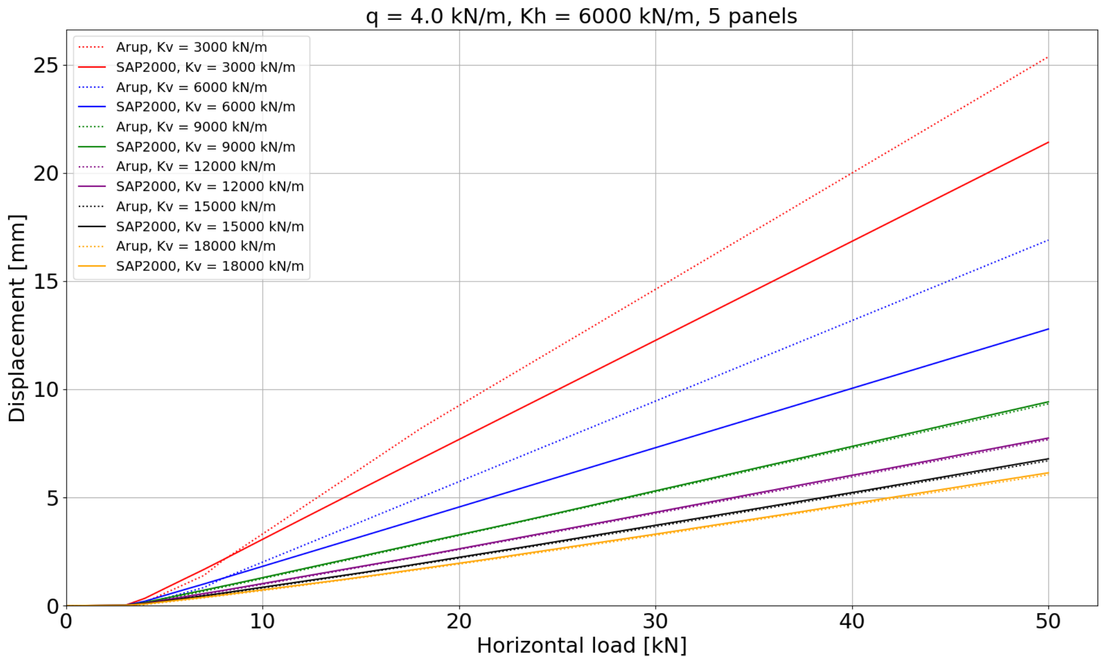


Figure 4.9: Kinematic rocking due to horizontal loads on a 5-panel wall and UDL of 4kN/m.

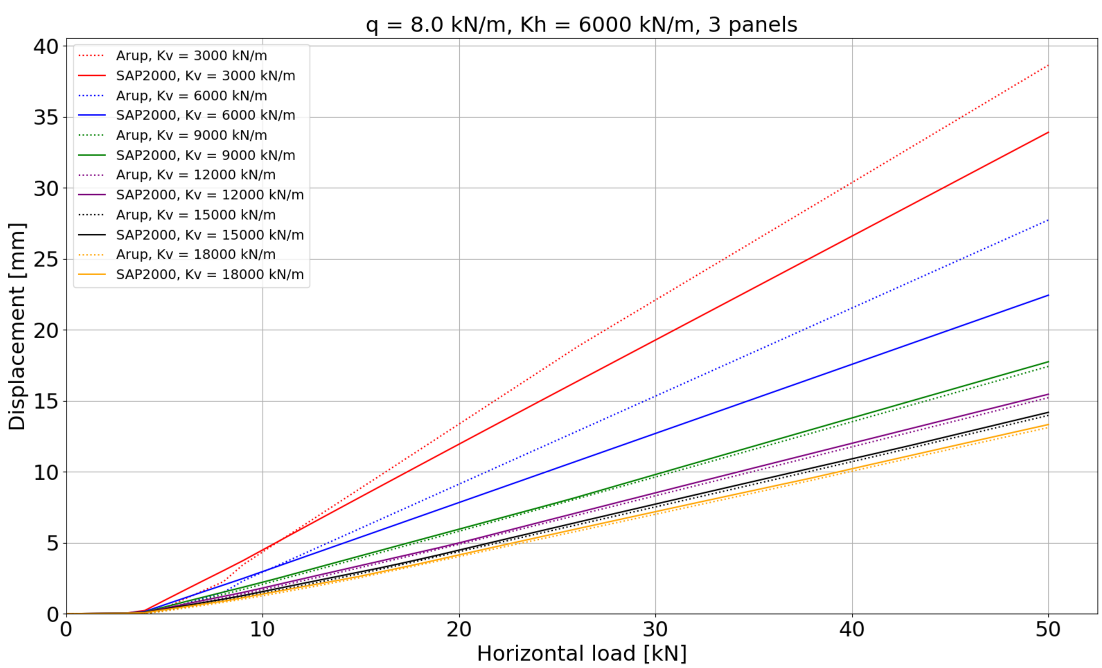


Figure 4.10: Kinematic rocking due to horizontal loads on a 3-panel wall and UDL of 8kN/m.

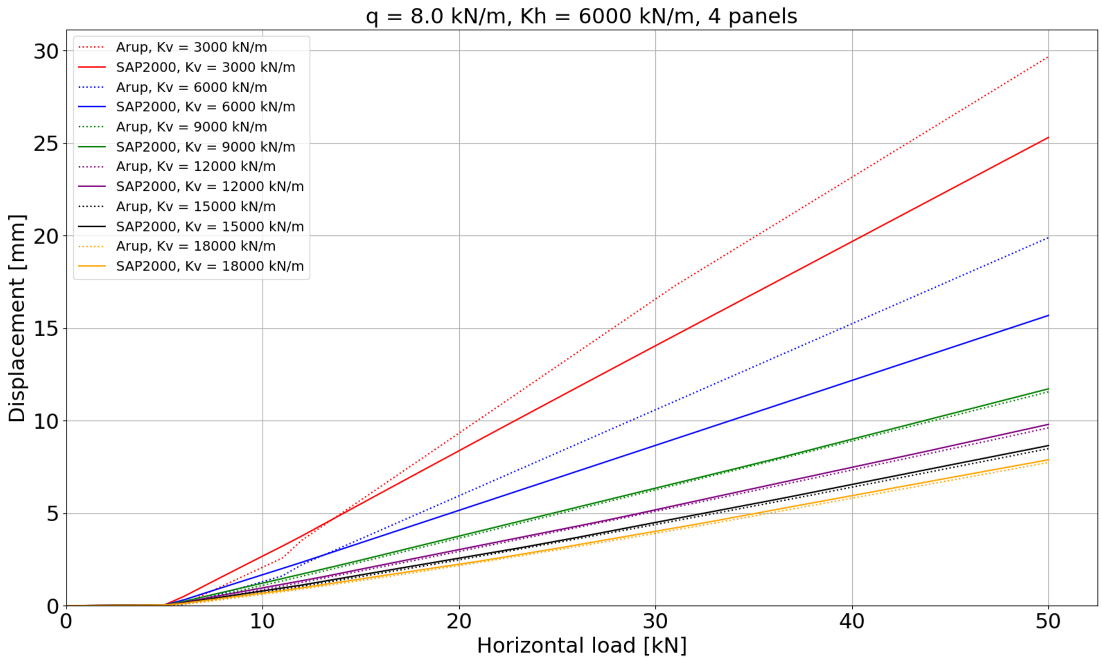


Figure 4.11: Kinematic rocking due to horizontal loads on a 4-panel wall and UDL of 8kN/m.

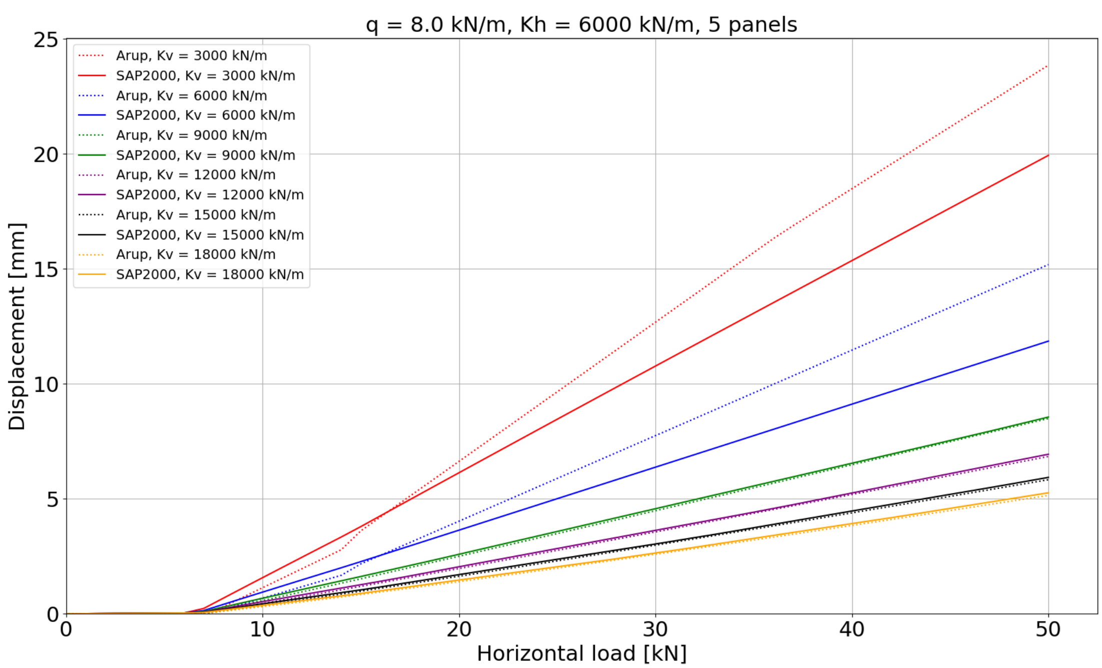


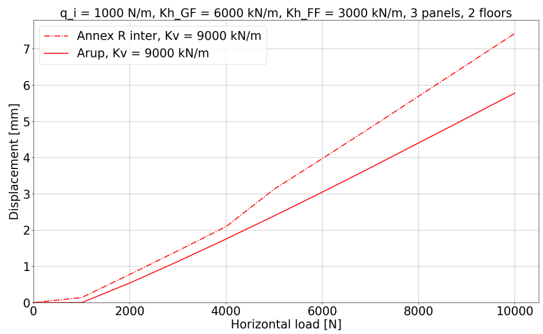
Figure 4.12: Kinematic rocking due to horizontal loads on a 5-panel wall and UDL of 8kN/m.

Figure 4.4 - 4.12 shows the displacement for a simulated multistory behavior of a wall second at the top. It shows that when $K_v \leq K_h$ (The red and blue graphs), there is quite a discrepancy between the analytical and the numerical method. When $K_v > K_h$, there is an excellent agreement between the Arup method and numerical. Increasing the stiffness of the vertical joints decreases the vertical displacement. Increasing the length of the wall also increases the stiffness of the wall.

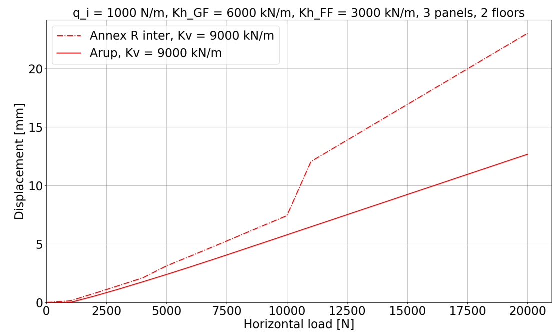
4.3 Validating Multistory Behavior

4.3.1 Two-story Behavior

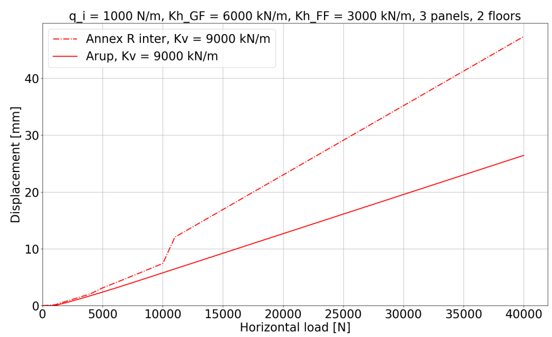
In this chapter, the results of the two-story analyses are presented. The walls consist of 3 panels with a width of 1.4 m each and a height of 2.7 m. Three load cases are studied, respectively 10 kN, 20 kN, and 40 kN. Three numerical models are studied. One without a floor, one with a flexible floor, and one with rigid floor.



(a) Horizontal force up to 10kN.



(b) Horizontal force up to 20kN.



(c) Horizontal force up to 40kN

Figure 4.13: Lateral deflection of a two-story wall, 3 panels, UDL 1 kN/m on each floor.

Figure 4.13 (a) shows the displacement of a two-story wall with 3 panels using the Arup and Annex R method. The step change when the horizontal force goes from 4 kN to 5 kN in the Annex R graph is when one of the walls goes from couple mode to intermediate. The step change when the horizontal force goes from 10 kN to 11 kN in Figure 4.13 (b) is when both walls go into single-wall mode. Figure 4.13 (c) shows no more steps in the graph, since both the walls are in single-wall mode.

Table 4.1 : Lateral displacement for a two-story wall.

		V_{Ed} = 10 kN			V_{Ed} = 20 kN			V_{Ed} = 40 kN		
Lateral displacement analysis	Unit	GF	FF	TOT	GF	FF	TOT	GF	FF	TOT
No floor	(mm)	3.5	3.2	6.7	7.4	7.8	15.2	15.2	16.6	31.8
Flexible floor	(mm)	3.1	3.7	6.8	6.8	8.3	15.1	14.3	17.3	31.6
Rigid floor	(mm)	2.7	3.7	6.4	6.2	8.2	14.4	13.1	17.3	30.4
Arup	(mm)	3.4	2.4	5.8	7.3	5.4	12.7	15.1	11.3	26.4
Annex R	(mm)	2.8	4.6	7.4	9.4	13.6	23	19.3	28	47.4

Note: GF = Ground floor; FF = First floor; TOT = GF + FF.

Table 4.1 shows the displacement for a two-story wall calculated numerically with no, flexible and rigid floor, and in addition the calculations using the analytical methods from Annex R and Arup. The displacement decreases as the floor becomes more rigid for the numerical analyses, giving this occurrence, no floor < flexible floor < rigid floor. The Arup displacement got a stiffer overall displacement than the numerical analyses, seen as the red line in Figure 4.14. The difference between the Arup slightly increases as the lateral force increases. The displacement of the ground floor agrees very well with the numerical model without a floor between the stories. The Annex R displacement has a more flexible response than the numerical models, and the discrepancy between them increases as the lateral force increases, seen as the purple line in Figure 4.14.

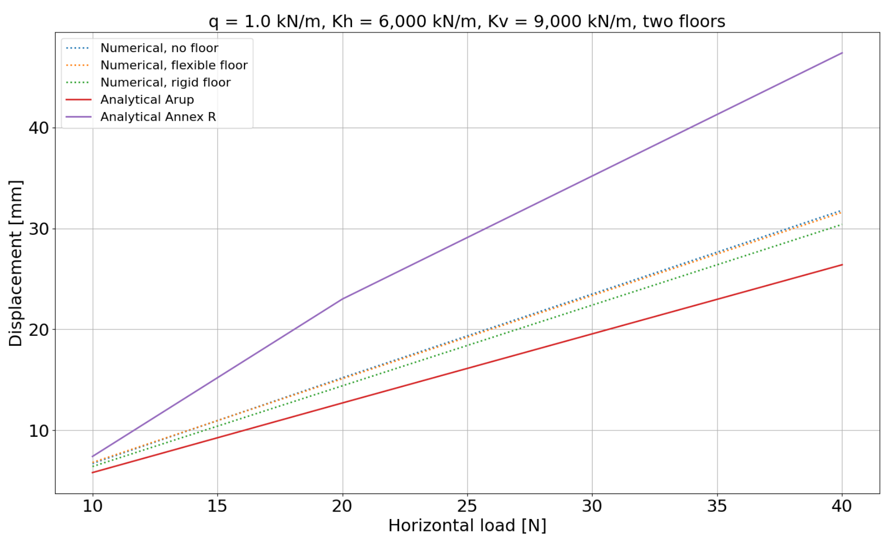


Figure 4.14: Plots of the results given in Table 4.1.

4.3.2 Three-story Behavior

The tables below show the deflection, calculated by the analytical methods from Annex R and Arup, and the numerical method, for Case A, B, C, and D given in Chapter 3.5.2. The discrepancy is between the analytical and the numerical results.

Table 4.2 : CASE A: Three stories, two panels

		Annex R				Arup				Numerical			
Parameters	Symbol	GF	FF	SF	TOT	GF	FF	SF	TOT	GF	FF	SF	TOT
Lateral displacement	Δh (mm)	7.3	8.8	6.2	22.4	3	6.3	6.2	15.5	2	4	8.1	14.1
Discrepancy	(%)	265	120	-23.5	58.9	50	57.5	-23.5	10	-	-	-	-

Note: GF = Ground floor; FF = First floor; SF = Second floor; TOT = GF + FF + SF.
$K_{h,GF} = 15kN/mm$; $K_{h,FF} = 5kN/mm$; $K_{h,SF} = 2.5kN/mm$.

Table 4.3 : CASE B: Three stories, three panels

		Annex R				Arup				Numerical			
Parameters	Symbol	GF	FF	SF	TOT	GF	FF	SF	TOT	GF	FF	SF	TOT
Lateral displacement	Δh (mm)	3.7	9.5	3.4	16.6	2.1	6.9	3.4	12.4	3.6	3.8	5.5	12.9
Discrepancy	(%)	2.7	150	-38.2	28.7	-41.7	81.6	38.2	-3.9	-	-	-	-

Note: GF = Ground floor; FF = First floor; SF = Second floor; TOT = GF + FF + SF.
$K_{h,GF} = 50kN/mm$; $K_{h,FF} = 5kN/mm$; $K_{h,SF} = 2.5kN/mm$.

Table 4.4 : CASE C: Three stories, three panels

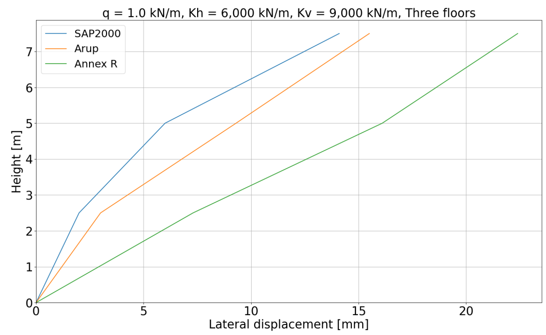
		Annex R				Arup				Numerical			
Parameters	Symbol	GF	FF	SF	TOT	GF	FF	SF	TOT	GF	FF	SF	TOT
Lateral displacement	Δh (mm)	22.2	9.5	3.4	35.1	15.5	6.9	3.4	25.8	5.9	6	7.5	19.4
Discrepancy	(%)	276	158	54.7	81	163	15	-54.7	33	-	-	-	-

Note: GF = Ground floor; FF = First floor; SF = Second floor; TOT = GF + FF + SF.
$K_{h,GF} = 5kN/mm$; $K_{h,FF} = 5kN/mm$; $K_{h,SF} = 2.5kN/mm$.

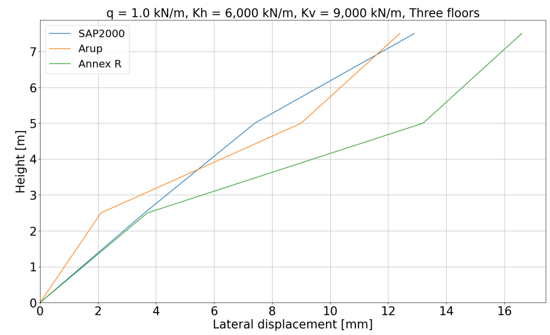
Table 4.5 : CASE D: Three stories, three panels

		Annex R				Arup				Numerical			
Parameters	Symbol	GF	FF	SF	TOT	GF	FF	SF	TOT	GF	FF	SF	TOT
Lateral displacement	Δh (mm)	10	16.6	18.7	45.3	6.7	6.8	5.7	19.3	6.8	9.2	10	26
Discrepancy	(%)	47	80.4	87	74.2	-1.5	-26.1	-43	-25.8	-	-	-	-

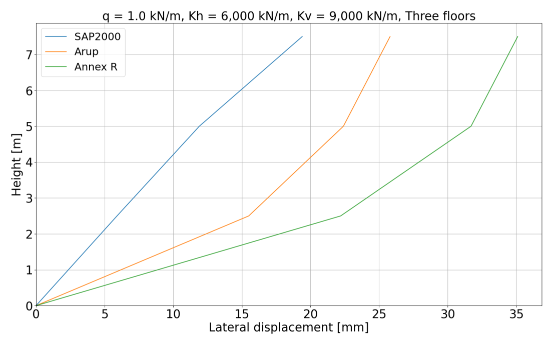
Note: GF = Ground floor; FF = First floor; SF = Second floor; TOT = GF + FF + SF.
$K_{h,GF} = 6kN/mm$; $K_{h,FF} = 3kN/mm$; $K_{h,SF} = 3kN/mm$.



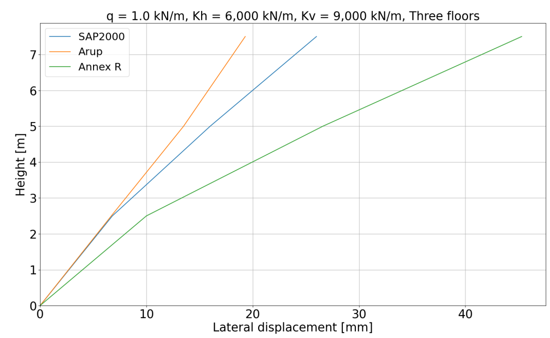
(a) CASE A - Lateral displacement in relation to the height above ground.



(b) CASE B - Lateral displacement in relation to the height above ground.



(c) CASE C - Lateral displacement in relation to the height above ground.



(d) CASE D - Lateral displacement in relation to the height above ground.

Figure 4.15: The four cases in the three-story analysis show the lateral displacement for each story.

Looking at Case A in Table 4.2, the horizontal displacement is higher for the analytical calculations than the numerical. The Arup got a discrepancy of 10 % more total deflection than the SAP2000 model. The discrepancy for each floor varies from almost 60 % more deflection to 23 % less. Annex R got a total discrepancy of 59 % more deflection. However, the ground floor is the main contributor, with a discrepancy of 265 %. Observing Figure 4.15 (a), the Arup calculations (blue line) are closer to the numerical (orange line) than the Annex R (green line) is.

For Case B, in Table 4.3, the discrepancy for the total deflection between the Arup and FEM model is down to just below 4% less. The displacement for each story also varies greatly, with the Arup giving both less and more displacement than the FEM analysis. Annex R provides a higher deflection overall than the FEM analysis but is close to the SAP2000 for the first floor. This can be observed in Figure 4.15 (b).

In Case C, the displacement increases for the Arup method giving it a discrepancy of 33 % compared to the numerical. Again, the displacement varies a lot for each story. Annex R gives an 81 % higher total displacement than the numerical model. Both the analytical models produce higher deflections than the numerical. Looking at Figure 4.15 (c) both the analytical methods have similar responses, only Arup is more stiffer than Annex R.

Case D is the only case where the K_v is higher than the K_h, giving the Arup a stiffer response resulting in 26 % less deflection than the SAP2000. Annex R provides a 74 % higher deflection. Figure 4.15 (d) shows good agreement between Arup and numerical for the ground floor with an increasing difference as floors are added. The difference between Annex R increases as more floors are added.

5 Discussion

In this chapter, we will look at the results given in Chapter 4. Chapter 5.1 discusses what source of error might be present in the analyses done. Chapter 5.2 compares the two methods from Annex R and, in addition, discusses the result of Arup and the two Annex R methods for one-story shear walls. In Chapter 5.3, The results from the simulated multistory behavior are discussed. Chapter 5.4 and 5.5 talk about the two- and three-story comparison between numerical and analytical results.

5.1 Source Of Errors

When modeling shear walls in FEM, some assumptions have to be made for it to simulate the real world as well as possible. Imperfections in the modeling will give some deflating results.

5.2 Interpolation vs. The Exact Number Of Panels

Since there are different ways to calculate the displacement with Annex R when the wall is in SW mode, it is interesting to look at the difference between them. Looking at Figure 4.1 in Chapter4.1, the difference between the interpolation using the horizontal load V_{CP} and V_{SW} and the exact number of panels lifting is so small that it can be neglected. The original interpolation does not work since the values interpolated between are not constant. This, in turn, makes the wall deflection decrease, even though there is an increase in horizontal load. The formulas' complexity is examined to decide whether to use the new interpolation or the exact number of panels lifting. Interpolation is much more manageable than calculating the exact number of panels lifting with Formula 2.12. Therefore, only the interpolation is used in further calculations. Both methods will give the same displacement as long as the wall does not consist of more than three panels. That is because when calculating the exact number of panels lifting j, it is rounded up to the nearest whole number.

5.3 Arup Formulas

The formulas in the Arup proposal are validated to see how well they work compared to a FEM model. The first thing was to make a FEM model that works as it should. Figure 4.2 shows the reproduction of the case given in Casagrande et al. (2018). The Figure also includes the result using the Arup method in that case. The difference between the analytical and numerical analyses is small, so the FEM model behaves as expected. The result is very close to that in Casagrande et al. (2018).

A simulated multistory behavior is done to check how the formulas in the Arup work. A two-story wall is simulated by calculating the tension force by the hold-down from the story above. This, again with a moment equal to that of a wall above, is used in calculating the wall looked at. Looking at Figure 4.4, when $K_v \leq K_h$, the Arup lateral displacement is higher than the displacement from the FEM analysis. The reason for this in the Arup formula can be explained by how the equations are made. The T is derived from the force equilibrium equation. And in that equation, Q is just added together when in reality there would, the force would most likely be distributed in some way in the wall. This is done since hold-downs often are crisp, and therefore it is preferable to be conservative in calculating the forces present when calculating in the ultimate limit state. The graphs where $K_v > K_h$ show an excellent agreement between Arup and the FEM analysis. Changing the number of panels, how much stiffer K_v is than K_h or the UDL on top still gives the same results for Arup and FEM.

5.4 Two-story Behavior

Looking at the results from the numerical analyses of a two-story wall, it can be observed that the lateral deflection varies depending on the stiffness of the floor. The floor contributes stiffness to the system, which affects the comparison with analytical analyses that do not consider the floor's stiffness. Looking at Table 4.1, it can be observed that the wall's total stiffness is like this, No floor < Flexible floor < Rigid floor. Comparing the lateral displacement for just the ground floor from the Arup with the deflection for the three numerical analyses, it can be observed that it is very close to the case with no floor. The first floor is the floor that differs from the numerical analysis. The results make sense if compared to the results in Chapter 4.2.2. The Arup deflection in the simulated two-story cases agrees with the numerical deflection of those cases. For the first floor, Arup has less deflection. Looking at just that floor would be like looking at a one-story wall. Looking at the results for a one-story wall given in Figure 4.3a - 4.3c, there is an excellent agreement for one-story deflection. Therefore the discrepancy for multistory deflection might come from how the Arup method calculates the rotation of the wall.

For Annex R, the total lateral displacement is larger than the numerical displacement for all the cases. This can be explained by the fact that Annex R calculates the rotation by dividing the lateral displacement by the panel's height when in either intermediate or single-wall mode. In Figure 4.13b, a "jump" can be observed in the deflection when the wall goes from coupled mode to intermediate/single wall mode. This is because the wall would already have an angle since the panels would rotate before they start lifting. This will give that sudden increase in deflection.

5.5 Three-story Behavior

In Case D, when $K_v > K_h$, the lateral displacement discrepancy was -26% for the Arup method and 74% for the Annex R method when compared with the FEM analysis. Arup had a stiffer response, whereas Annex R was more flexible. Looking at the ground floor, the discrepancy is just -1.5%, which is consistent with the results for two-story and simulated multistory behavior in Chapters 4.2 and 4.3.1. On the first and second floors, the walls were stiffer. This, again, may be due to the calculation of the rotation, as in the case of the two stories.

For Cases A, B, and C, the $K_v \leq K_h$ meaning there is a different response mode using the Arup method compared with Case D. In Cases A and C, the displacement was more significant for both analytical methods. Arup is only 10% in Case A and 33% in Case C. This extra deflection can be due to the T as mentioned in the discussion of the validation of the Arup formulas in Chapter 5.3. This corresponds to the results from the simulated multistory behavior in Chapter 4.2.2. Annex R showed a higher discrepancy. This is probably because of the rotation that the method uses. The input forces can also be a contributor to this discrepancy. The Annex R method uses the equivalent moment at the bottom of the wall as input, while Arup uses the moment at the top of the wall and the lateral shear force. For one-story walls, this produces the same results. However, for multistory, this can give different deflections.

Overall, sometimes Arup gives the most accurate lateral displacement, and sometimes Annex R does. The Arup method is generally the most accurate for the total deflection, observed in Figure 4.15, and precise for the ground floor when $K_v > K_h$.

6 Conclusion

The overall purpose of this thesis was to investigate kinematic rocking of single- and multistory segmented CLT shear walls. The study adds to previous research by adding empirical evidence for the research questions under scope. Pertaining to RQ1, how well does the lateral deflection for one-story shear walls calculated by the analytical method from Arup match the results using Annex R? The study found to support that they provide the same results. Minimal differences can be observed if interpolation is used, though these differences are negligible.

Regarding RQ2, compared to numerical analyses, how do the analytical methods from Arup and Annex R predict lateral deflection due to kinematic rocking for two- and three-story CLT shear walls, the study showed that Annex R produced excessively more flexible deflection in some cases than those produced by the FEM models. This is probably due to the way rotation is calculated. The displacement is generally stiffer for Arup when $K_v \leq K_h$. Again, this probably comes from how this method calculates its rotation.

The analytical methods from Annex R and Arup produce good results for one-story walls compared to FEM models. They both produce the same results for different cases of one-story shear walls. Using interpolation to find the displacement when the wall is in the intermediate mode is a reasonable approach since the difference is minimal compared to the method when calculating the exact number of panels that lift. When $K_v \leq K_h$, the Arup methods formulas work very well when simulating multistory behavior. However, when $K_v > K_h$, the displacement is much higher than the numerical results. Sometimes Annex R is more accurate, and sometimes the Arup is.

Arup generally gives less discrepancy to the numerical than the Annex R when looking at the total deflection, however, both methods vary a lot for each story. More studies should be done to further validate the methods.

6.1 Further work

Making a parametric analysis for a three-story wall in a FEM program that runs many analyses where the horizontal force increases, with the result plotted, would be very helpful. This would make a good visualization of how it behaves. In this study, only one or a few loads are studied for multistory, so it is hard to see the behavior.

Doing an experimental analysis and comparing it to the numerical and analytical analysis to see how well they calculate the displacement versus the actual, real-world multistory wall.

References

Aloisio, A., Boggian, F., Sævareid, H. O., Bjørkedal, J. and Tomasi, R. (2023), 'Analysis and enhancement of the new Eurocode 5 formulations for the lateral elastic deformation of LTF and CLT walls', *Structures* **47**, 1940–1956.
URL: *http://dx.doi.org/10.1016/j.istruc.2022.11.129*

Bjørkedal, J. and Saevareid, H. O. (2022), Validation of Serviceability Limit State Calculation Models for Timber Shear Walls in Multi-Storey Buildings, Master's thesis, Norwegian University of Life Science, Ås.
URL: *https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/3032920*

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

Casagrande, D., Arenzo, G. D. ., Doudak, G. and Masroor, M. (2023), CEN/TC 250/SC 5/WG 1 N 315 - Validation of the analytical formulations for the calculation of lateral displacements of Segmented CLT shearwalls, Technical report.

Casagrande, D., Doudak, G., Mauro, L. and Polastri, A. (2018), 'Analytical Approach to Establishing the Elastic Behavior of Multipanel CLT Shear Walls Subjected to Lateral Loads', *Journal of Structural Engineering* **144**(2).
URL: *http://dx.doi.org/10.1061/(asce)st.1943-541x.0001948*

CEN, T. C. (2023), CEN/TC 250/SC 5 N 1729 - Eurocode 5-Design of timber structures-Part 1-1: General rules and rules for buildings, Technical report.

Computers & Structures, I. (2017), CSI Analysis Reference Manual, Technical report.
URL: *www.csiamerica.com*

Computers & Structures, I. (2022), 'CSI OAPI Documentation'.

D'Arenzo, G., Schwendner, S. and Seim, W. (2021), 'The effect of the floor-to-wall interaction on the rocking stiffness of segmented CLT shear-walls', *Engineering Structures* **249**.
URL: *http://dx.doi.org/10.1016/j.engstruct.2021.113219*

Jagota, V., Preet, A., Sethi, S. and Kumar, K. (2013), Finite Element Method: An Overview, Technical Report 1.
URL: *http://wjst.wu.ac.th*

Sanborn, K., Gentry, T. R., Koch, Z., Valkenburg, A., Conley, C. and Stewart, L. K. (2019), 'Ballistic performance of Cross-laminated Timber (CLT)', *International Journal of Impact Engineering* **128**, 11–23.
URL: *http://dx.doi.org/10.1016/j.ijimpeng.2018.11.007*

Silva, C., Branco, J. M. and Lourenco, P. B. (2013), A project contribution to the development of sustainable multistorey timber buildings of Timber Structures View project Seismic Behavior of Concrete Block Masonry Buildings View project, Technical report, Guimarães.
URL: *https://www.researchgate.net/publication/310160381*

Smith, A. and Lawrence, A. (2023), Proposed method for analysing deections of multi-storey segmented CLT shear walls without openings, Technical report.

Wallner-Novak, M., Koppelhuber, J. and Pock, K. (2014), Cross-Laminated Timber Structural Design Basic design and engineering principles according to Eurocode, Technical report.
URL: *www.proholz.at*

Younis, A. and Dodoo, A. (2022), 'Cross-laminated timber for building construction: A life-cycle-assessment overview', **52**.
URL: *http://dx.doi.org/10.1016/j.jobe.2022.104482*

Appendix

	Description
Appendix A	Annex R from the newest proposal for the Eurocode 5
Appendix B	The proposed method from Arup.
Appendix C	Python script used for the parametric analysis for a multistory wall. The script calculates the displacement with the analytical methods from Annex R and Arup, opens the FEM program SAP2000 with OAPI, and runs the numerical analysis.
Appendix D	Python script used for validating the formulas used in the Arup method.

Table 6.1 : List of Appendixes.

Appendix A - Annex R

Annex R
(informative)

Lateral displacement of multi storey monolithic shear walls and single-storey segmented shear walls

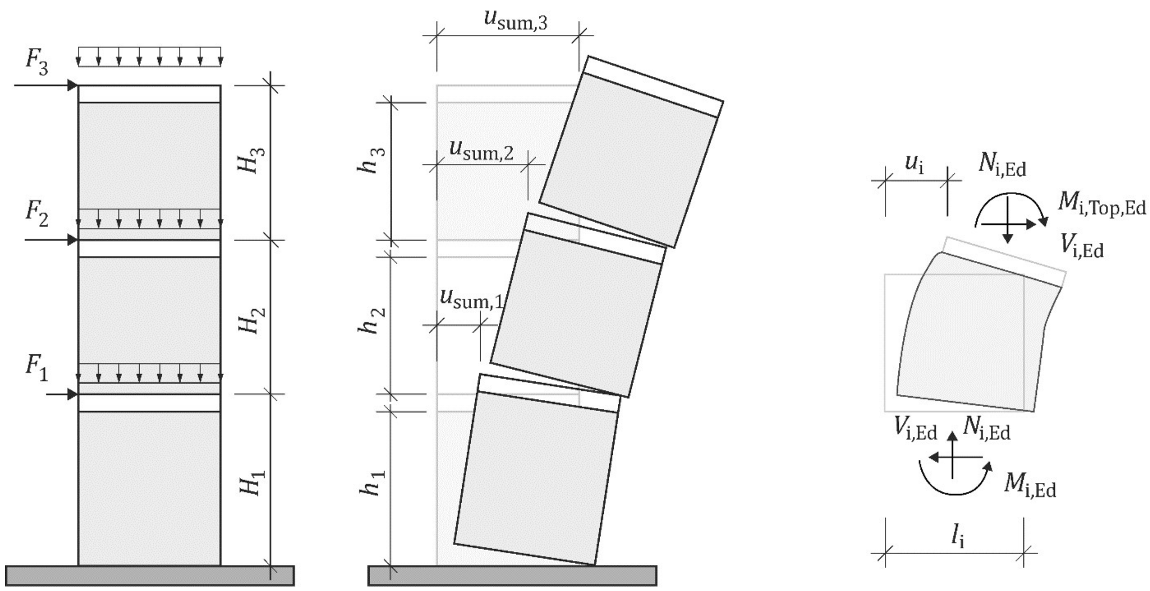
R.1 Use of this annex

(1) This Informative Annex provides supplementary guidance to specific provision given in 13.3.3 for Light Timber Frame shear walls (LTF), and in 13.6 for shear walls built out of massive timber (CLT and GLVL-C).

NOTE National choice on the application of this Informative Annex can be given in the National Annex. If the National Annex contains no information on the application of this Informative Annex, it can be used.

R.2 Scope and field of application

(1) This Informative Annex may be used for the calculation of lateral displacements of multi-storey monolithic shear walls, connected at each inter-storey, and single-storey segmented shear walls.



a) undeformed multi-storey shear wall **b) deformed multi-storey shear wall** **c) mechanism of a single-storey shear wall**

Figure R.1 — Lateral displacements of a multi-storey shear wall

(2) To apply the provisions of this subsection, the shear walls shall have the height h_i according to Figure R.1 and

— shall have no horizontal splices throughout the elements;

— should have lengths with a variation not exceeding 10% along the height of the building;

— may only have horizontal splices of sheathings which are backed by battens with adequate connections, see Figure 13.3 b) in the case of LTF.

a.1

NOTE If wall length varies by more than 10% over the height of the building, the deflection due to rocking and bending effects can be calculated with alternative models.

R.3 Method of calculation of lateral displacement

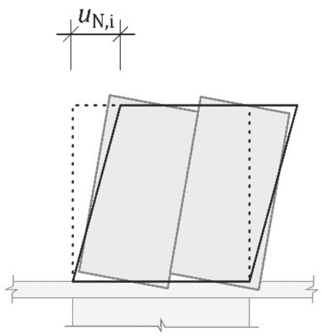
(1) The total lateral displacement of the shear wall at the top of the jth storey $u_{\mathrm{sum,j}}$ may be calculated as the sum of the inter-storey lateral displacements u_{i} from the 1st to the jth storey:

$$u_{\mathrm{sum,j}} = \sum_{i=1}^{j} u_{\mathrm{i}} \tag{R.1}$$

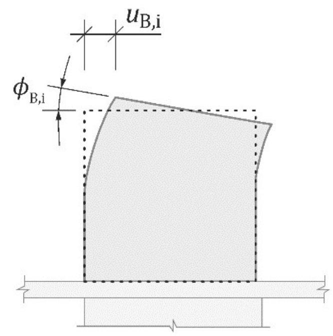
(2) The inter-storey lateral displacement u_{i} of the shear wall at the top of ith storey may be taken as the sum of the relevant inter-storey displacement contributions as given in Figure 13.10, Figure R.2, and Table R.1.

Table R.1 — Inter-storey displacement contributions for multi-storey shear walls

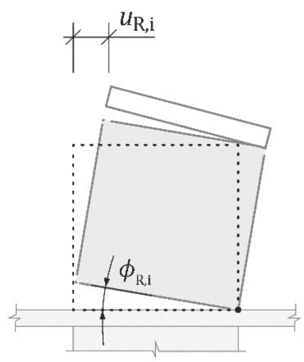
			Fully anchored and monolithic walls	
Inter-storey displacement contribution		**Figure**	**Formula for LTF**	**Formula for CLT and GLVL-C**
$u_{\mathrm{N,i}}$	Inter-storey lateral displacement from the deformation of the fasteners connecting sheathing to frame in LTF walls	R.2 a)	(13.16)	Not relevant
$u_{\mathrm{B,i}}$	Inter-storey lateral displacement due to the in-plane bending deformation	R.2 b)	(R.2) (R.3) and (13.18)	(R.2), (R.4) and (R.5)
$u_{\mathrm{R,i}}$	Inter-storey lateral displacement due to the rocking of the shear wall related to the vertical-shear flexibility of vertical joints (only for segmented walls) and the vertical-tensile flexibility of the mechanical anchors	R.2 c)	(R.6) and (R.7)	(R.6) and (R.7)
$u_{\mathrm{A,i}}$	Inter-storey lateral displacement due to the rigid body sliding of the shear wall related to the horizontal-shear flexibility of the mechanical anchors	R.2 d)	(R.8)	(R.8)
$u_{\mathrm{C,i}}$	Inter-storey lateral displacement from the deformation of the bottom rail perpendicular to grain in LTF walls	R.2 e)	(R.9)	Not relevant
$u_{\mathrm{S,i}}$	Inter-storey lateral displacement due to the in-plane shear deformation	R.2 f)	(13.24)	(R.10)
$u_{\theta\mathrm{,i}}$	Inter- storey lateral displacement due to the rotation at the top of the shear wall underneath – namely, the shear wall at the (i-1)th storey	R.2 g)	(R.11) – (R.15)	(R.11) – (R.15)



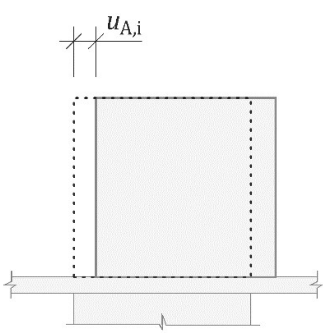
a) Inter-storey lateral displacement from the deformation of the sheathing-to-framing connection in LTF walls



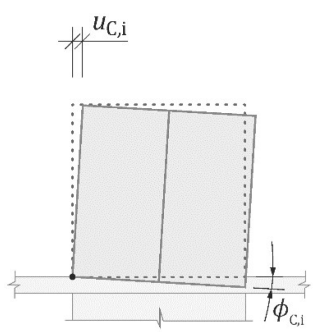
b) Inter-storey lateral displacement due to the in-plane bending deformation



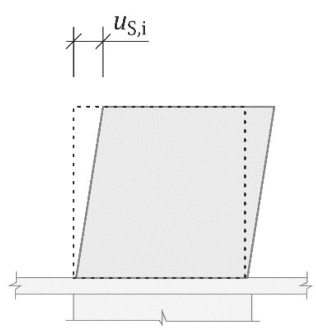
c) Inter storey lateral displacement due to the rocking of the shear wall related to the vertical-shear flexibility of vertical joints (only for segmented walls) and the vertical-tensile flexibility of the mechanical anchors



d) Inter-storey lateral displacement due to the rigid body sliding of the shear wall related to the horizontal-shear flexibility of the mechanical anchors

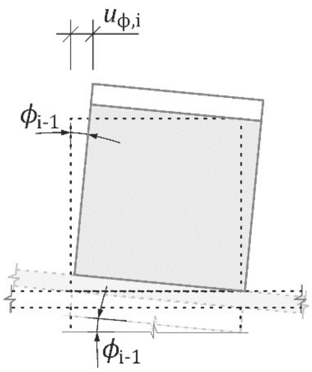


e) Inter-storey lateral displacement from the deformation of the bottom rail perpendicular to grain in LTF walls



f) Inter-storey lateral displacement due to the in-plane shear deformation

a.3



g) Inter-storey lateral displacement due to the rotation at the top of the shear wall underneath – namely, the shear wall at the (***i***-1)th storey

Figure R.2 — Inter-storey displacement contributions for multi-storey shear walls

R.4 Displacement contributions for LTF, fully anchored- and monolithic CLT- and GLVL-C shear walls without openings

(1) The contributions to the inter-storey displacement u_{i} of the i^{th} storey for a LTF fully anchored- or a monolithic CLT- or a monolithic GLVL-C shear wall with no openings may be calculated according to R.4(2) – (7), where the following symbols apply, see Figure R.1:

H_{i} is the inter-storey height of the i^{th} storey;

h_{i} is the height of the shear wall at the i^{th} storey;

l_i is the length of the shear wall at the i^{th} storey;

$V_{\mathrm{i,Ed}}$ is the design shear load acting at the i^{th} storey;

$N_{\mathrm{i,Ed}}$ is the design vertical force assumed to be on the centreline of the shear wall of the i^{th} storey;

$M_{\mathrm{i,top,Ed}}$ is the design moment acting at the top of the shear wall of the i^{th} storey;

$M_{\mathrm{i,Ed}}$ is the total design moment acting at the bottom of the shear wall of the i^{th} storey.

(2) The inter-storey lateral displacement $u_{\mathrm{N,i}}$ of a LTF shear wall due to the deformation of the sheathing-to-framing connections of a wall consisting of consecutive sheathing panels of varying width $l_{\mathrm{per,j}}$ fixed to one or both sides of the framing, may be calculated as given by Formula (13.16).

(3) The inter-storey lateral displacement due to the in-plane bending deformation $u_{\mathrm{B,i}}$ may be taken as follows:

$$u_{\mathrm{B,i}} = \frac{M_{\mathrm{i,top,Ed}}\ h_{\mathrm{i}}^{2}}{2\,(EI)_{\mathrm{ef,i}}} + u_{\mathrm{B,V,i}} \tag{R.2}$$

where

$(EI)_{\mathrm{ef,i}}$ is the effective in-plane bending stiffness of the shear wall at the i^{th} storey taken from Formula (R.3) for LTF and to Formula (R.4) for CLT or GLVL-C.

$$(EI)_{\mathrm{ef,i}} = \frac{E_{\mathrm{m,0,mean}}\ A_{\mathrm{stud}}\ l_i^{2}}{2} \tag{R.3}$$

where

$E_{m,0,mean}$ is the mean modulus of elasticity parallel to grain of the external studs;

A_{stud} is the average cross-section area of the leading and trailing studs.

$$(EI)_{ef,i} = \frac{E_{0,mean}\, t_{z,i}\, l_i^{\,3}}{12} \quad (R.4)$$

where

$E_{0,mean}$ is the mean modulus of elasticity parallel to grain of the vertical laminations for CLT or the mean modulus of elasticity parallel to grain for GLVL-C elements;

$t_{z,i}$ is the total thickness of the vertical layers for CLT or the overall thickness for GLVL-C shear walls used at the i^{th} storey;

$u_{B,V,i}$ is the contribution of the in-plane bending deformation due to the lateral force $V_{i,Ed}$ taken from Formula (13.18) for LTF and from Formula (R.5) for CLT or GLVL-C.

$$u_{B,V,i} = \frac{V_{i,Ed}\ h_i^{\,3}}{3(EI)_{ef,i}} \quad (R.5)$$

(4) The inter-storey lateral displacement due to the rocking kinematic mode of the shear wall $u_{R,i}$ may be taken as follows:

$$u_{R,i} = \max\left\{\left(\frac{M_{i,Ed}}{K_{R,i}} - \frac{N_{i,Ed}\ (l_i/2 - l_c)}{K_{R,i}}\right) H_i; 0\right\} \quad (R.6)$$

with

$$K_{R,i} = \sum_j \left[K_{a,z,j}\ (s_{a,j} - l_c)^2\right] \quad (R.7)$$

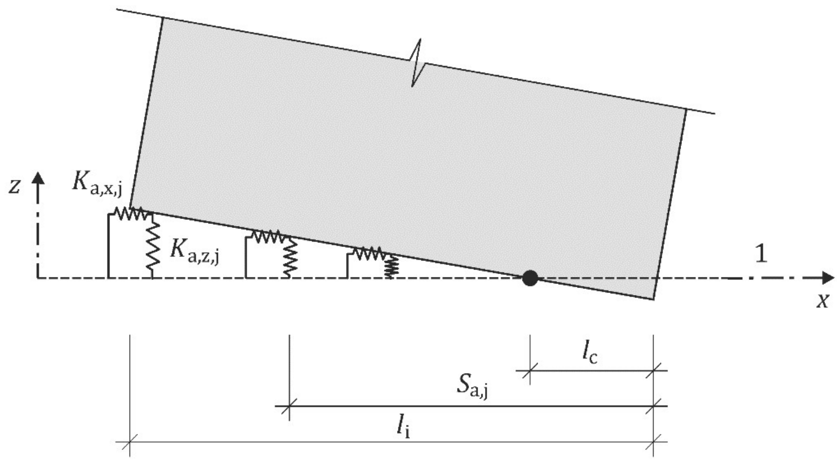
where

l_c is the distance between the centre of rotation and the end of the wall (Figure R.3);

l_c may be taken as 0,1 l_i. Values different from l_c = 0,1l_i can be calculated from considering the deformation contribution related to the contact of the panel with either the foundation or the timber floor panel underneath.

$K_{R,i}$ is the rocking stiffness of shear wall at the i^{th} storey due to the vertical-tensile stiffness $K_{a,z,j}$ for serviceability limit state of the parts of the j^{th} mechanical connections subjected to tension (e.g. bolts, screws, hold-downs, tie-downs, foundation tie-downs, shear plates, etc.) accounting for the connections between the bottom of the wall and the floor, and the connections between the floor and the top of the wall below. For LTF shear walls the stiffness of the fasteners between the sheathing panels and bottom rail should not be included in Formula (R.7).

$s_{a,j}$ is the distance of the j^{th} mechanical anchor from the shear wall edge.



Key

1 upper surface of floor

Figure R.3 — Static model for rocking kinematic mode

(5) The inter-storey lateral displacement due to the rigid body sliding of the shear wall $u_{A,i}$ may be taken as follows:

$$u_{A,i} = \frac{V_{i,Ed}}{K_{A,i}} \tag{R.8}$$

where

$K_{A,i}$ is the sliding stiffness of the shear wall at the i^{th} storey accounting for all horizontal interfaces (e.g. at the bottom of the wall, at the top of the wall and any other interfaces within the floor).

(6) The storey lateral displacement $u_{C,i}$ of a LTF wall due to the deformation of the bottom and top rails perpendicular to grain at the trailing stud, may be taken as follows:

$$u_{C,i} = w_{SLS,z} \frac{H_i}{l_i} \tag{R.9}$$

where

$w_{SLS,z}$ is the compressive deformation of the bottom and top rails perpendicular to grain according to 9.4.

(7) The inter-storey lateral displacement due to the in-plane shear deformation $u_{S,i}$ may be taken as follows for LTF and Formula (R.10) for CLT or GLVL-C:

$$u_{S,i} = \frac{V_{i,Ed}\ h_i}{G_{xy,mean,i}\ t_i\ l_i} \tag{R.10}$$

where

t_i is the total thickness of the CLT or GLVL-C shear wall at the i^{th} storey;

$G_{xy,mean,i}$ is the mean effective in-plane shear modulus of the CLT or GLVL-C shear wall at the i^{th} storey.

NOTE For GLVL-C $G_{xy,mean}$ can alternatively be named as $G_{0,edge,mean}$, see Table 3.1.

(8) The storey lateral displacement due to the rotation at the top of the shear wall underneath $u_{\theta,i}$ may be taken as follows:

$$u_{\theta,\mathrm{i}} = \theta_{i-1}\, H_{\mathrm{i}} \text{ for } i \geq 1 \tag{R.11}$$

with

$$\theta_{\mathrm{i}-1} = \theta_{\mathrm{i}-2} + \varphi_{\mathrm{B,i}-1} + \varphi_{\mathrm{R,i}-1} + \varphi_{\mathrm{C,i}-1} \tag{R.12}$$

where

$\theta_{\mathrm{i}-1}$ is the rotation at the top of the shear wall at the (i-1)th storey calculated as given by Formula (R.12);

$\theta_{\mathrm{i}-2}$ is the rotation at the top of the shear wall at the (i-2)th storey.

NOTE θ_0 accounts for the rotation of any superstructure / substructure below the bottom of the wall caused by $M_{0,\mathrm{Ed}}$. In most situations this rotation can be considered negligible.

$\varphi_{\mathrm{B,i-1}}$ is the rotation due to the panel bending deformation at the top of shear wall at the (i-1)th storey calculated as given by Formula (R.13):

$$\varphi_{\mathrm{B,i-1}} = \frac{M_{\mathrm{i-1,top,Ed}}\, h_{\mathrm{i}-1}}{(EI)_{\mathrm{ef,i}-1}} + \frac{V_{\mathrm{i-1,Ed}}\, {h_{\mathrm{i}-1}}^2}{2\,(EI)_{\mathrm{ef,i}-1}} \tag{R.13}$$

$\varphi_{\mathrm{B,i-1}}$ is the rotation contribution due to the rocking of the shear wall at the (i-1)th storey calculated as given by Formula (R.14):

$$\varphi_{\mathrm{B,i-1}} = \frac{u_{\mathrm{R,i}-1}}{H_{\mathrm{i}-1}} \tag{R.14}$$

$\varphi_{\mathrm{C,i-1}}$ is the rotation contribution due to the compression perpendicular to grain of the shear wall at the (i-1)th storey calculated as given by Formula (R.15):

$$\varphi_{\mathrm{C,i-1}} = \frac{u_{\mathrm{C,i}-1}}{H_{\mathrm{i}-1}} \tag{R.15}$$

R.5 Displacement contributions for single-storey segmented CLT- and GLVL-C shear walls without opening

(1) The contributions to displacement of single-storey for a CLT or GLVL-C segmented shear wall composed of m panels with length l_{j} may be calculated according to R.5(2) – (8).

(2) The storey lateral displacement due to the panel shear deformation $u_{\mathrm{S},1}$ may be calculated from Formula (R.10) where $l_1 = m\, l_{\mathrm{j}}$.

(3) The inter-storey lateral displacement due to the panel bending deformation $u_{\mathrm{B,i}}$ may be calculated as given by Formula (R.2) where the bending stiffness $(EI)_{\mathrm{ef},1}$ is taken as follows:

$$(EI)_{\mathrm{ef},1} = m\, \frac{E_{0,\mathrm{mean}}\, t_{\mathrm{z},1}\, {l_{\mathrm{j}}}^3}{12} \tag{R.16}$$

(4) The inter-storey lateral displacement due to the rigid body sliding of the wall $u_{\mathrm{A},1}$ may be calculated like for a monolithic shear wall taken from Formula (R.8).

(5) If the segmented shear wall is anchored against uplift at the end corners and the vertical-tensile stiffness of the shear connections are neglected, three different rocking kinematic modes may occur depending on the relative stiffness of the hold-down as defined from a) – c):

a) Coupled-panel (CP) kinematic mode is a mode in which each panel is in contact with the foundation (or the floor underneath) having a centre of rotation according to Figure R.4 a). To achieve a coupled-panel kinematic mode, the following should be applied:

a.7

$$\frac{K_{\text{SLS,anc}}}{K_{\text{SLS,con}}} \geq \frac{1-\widetilde{N_i}\,(3\text{m}-2)/m^2}{1-\widetilde{N_i}\,(\text{m}-2)/\text{m}^2} \quad \text{(R.17)}$$

NOTE The coupled-panel (CP) kinematic mode appears when the hold-down is relatively stiff.

b) Intermediate (IN) kinematic mode in which only some panels are in contact with the foundation (or the floor underneath), see Figure R.4 b).

c) Single-wall (SW) kinematic mode in which a single centre of rotation at one of the ends of the entire shear wall, see Figure R.4 c). To achieve a single-wall (SW) kinematic mode, the following should be applied:

$$\frac{K_{\text{SLS,anc}}}{K_{\text{SLS,con}}} \leq \frac{1-\widetilde{N_i}}{1+\widetilde{N_i}\,(m-2)} \quad \text{(R.18)}$$

NOTE The single-wall (SW) kinematic mode appears when the hold-down is relatively flexible.

where

$K_{\text{SLS,anc}}$ is the vertical-tensile stiffness of the hold-down placed at the corner of the shear wall;

$K_{\text{SLS,con}}$ is the stiffness of the vertical joint;

$\widetilde{N_i}$ is the dimensionless vertical load on the shear wall that may be taken from Formula (R.19).

$$\widetilde{N_i} = \frac{N_{1,\text{Ed}}\,l_i}{2\,M_{1,\text{Ed}}} \quad \text{(R.19)}$$

(6) The inter-storey lateral displacement due to the rocking of the wall $u_{\text{R},1}$ may be taken as follows for the CP and SW kinematic mode, respectively:

$$\text{u}_{\text{R},1} = \max\left\{\left(\frac{M_{1,\text{Ed}}}{K_{\text{R},1,\text{CP}}} - \frac{N_{1,\text{Ed}}l_j}{2\,K_{\text{R},1,\text{CP}}}\right)H_1;0\right\} \text{ for CP kinematic mode} \quad \text{(R.20)}$$

$$u_{\text{R},1} = \max\left\{\left(\frac{M_{1,\text{Ed}}}{K_{\text{R},1,\text{SW}}} - \frac{N_{1,\text{Ed}}}{2\,K_{\text{SLS,anc}}\,l_1}\right)H_1;0\right\} \text{ for SW kinematic mode} \quad \text{(R.21)}$$

where

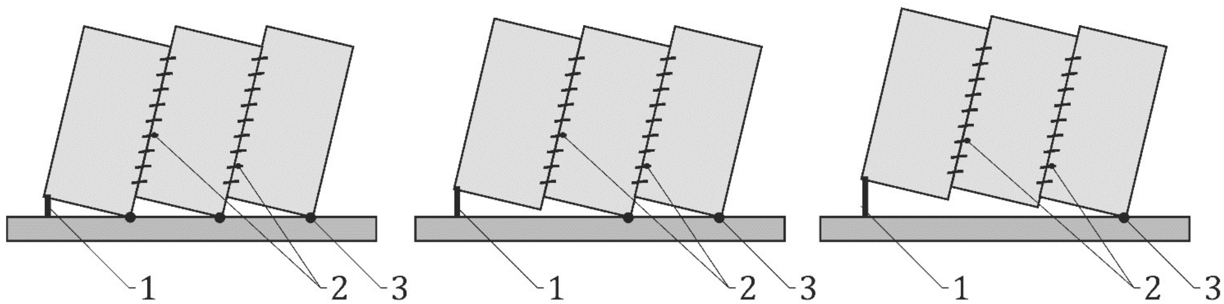
$K_{\text{R},1,\text{CP}}$ is the rocking stiffness of the shear wall in the case of CP kinematic mode, calculated from Formula (R.22);

$K_{\text{R,i,SW}}$ is the rocking stiffness of the shear wall in the case SW kinematic mode, calculated from Formula (R.23);

$$K_{\text{R},1,\text{CP}} = \frac{[K_{\text{SLS,anc}}+(m-1)\,K_{\text{SLS,con}}]}{m^2}\,{l_1}^2 \quad \text{(R.22)}$$

$$K_{\text{R},1,\text{SW}} = \left[\frac{1}{K_{\text{SLS,anc}}} + \frac{(m-1)}{K_{\text{SLS,con}}}\right]^{-1} {l_1}^2 \quad \text{(R.23)}$$

(7) Values of $u_{\text{R},1}$ for the IN kinematic mode may be obtained by linear interpolation between the ones obtained from Formulae (R.20) and (R.21).



a) Coupled-panel (CP) kinematic mode **b) Intermediate (IN) kinematic mode** **c) Single-wall (SW) kinematic mode**

Key

1 hold-down

2 rotation centre

3 deformed vertical joint between two panels

Figure R.4 — Rocking kinematic modes for segmented CLT or GLVL-C shear walls

a.9

Appendix B - Arup

Proposed method for analysing deflections of multi-storey segmented CLT shear walls without openings

Andrew Smith & Andrew Lawrence
Arup
Issue 5

May 2, 2023

Contents

1 Introduction

The current approach for analysing the deflections of segmented CLT shear walls takes a discrete panel approach, based on the paper [1]. Unfortunately, as discussed in the combined meeting with SC 5/WG 1 (CLT) and SC 5/WG 3/SG 5 (timber diaphragms) on Monday 6th June 2022, this approach is not applicable to multi-storey walls. This is because it assumes there is no moment at the top of the wall:

- The method assumes that a vertical load is applied uniformly at the top of the wall along its whole length.
- The method does not allow for a tie-down force applied at the top of the wall from the storey above.

Incorporating these effects into the solutions in [1] would be highly complex due to the mathematics involved.

In the meeting on Monday 6th June 2022, it was agreed that Arup would attempt to derive a new method for analysing segmented walls which would be applicable to multi-storey construction. Arup understands that there is an informal deadline of 8th July 2022 for changes to Annexes, and that after this it will be difficult to make new additions to the code.

Following successful research in the past 3 weeks, Arup has assembled a proposal for a new design method. This document sets out this method, which:

- Allows engineers to calculate deflections of segmented walls;
- Is applicable to single- and multi-storey walls;
- Is compatible with the combined deflection proposal for the revised version of Eurocode 5 (new Annex Y);
- Allows engineers to calculate the force in the tie-down and the force in the vertical fasteners between panels, and could therefore also be useful for ULS design.

Neither this proposed method nor the method in [1] (which is already incorporated into Clause Y.5 of Annex Y) have been calibrated for multi-storey walls. This will need to be carried out before either method can be published in the final code. However, given the tight timescales, it is proposed that the method in this document be incorporated into the draft in time for the Formal Enquiry, in addition to the method in [1] (currently Clause Y.5 of Annex Y).

Numerical checks will then be carried out on both this method and the method in [1] while the formal enquiry is ongoing. These would be based on finite element models of multi-storey walls and physical testing (if available). Following this, the methods can be updated if required. If either method is found to be invalid then it can be removed from Annex Y and not included in the final version of the code.

[1] Casagrande, D., Doudak, G., Mauro, L., Polastri, A. (2018) "Analytical Approach to Establishing the Elastic Behavior of Multipanel CLT Shear Walls Subjected to Lateral Loads" J. Struct. Eng., 144(2): 04017193

The proposed method is set out in Section 4 of this document. The method has three Clauses:

- Clause 1 calculates a simplified loading on the top of the wall;
- Clause 2 selects the response mode of the wall (i.e. how it's behaving);
- Clause 3 calculates the wall's deflection.

It is envisaged that Section 4 of the document could be incorporated into Annex Y.5 (along with symbol definitions in Section 3), in addition to the current method.

2 Background and assumptions

This section sets out the background to the new proposed method for analysing segmented CLT walls and the key assumptions made.

Due to the time constraints, it has not yet been possible to type up the derivations behind the method. These will be provided in due course.

2.1 Wall layout

Figure 1 shows the layout of a typical wall between two floors at the i^{th} storey:

- The wall is made up of a number of solid discrete panels (e.g. constructed from CLT), each assumed to be equal width b.
- These panels are connected together at the vertical joints (e.g. via screwed half-lap joints).
- The wall length is l_i, and it has a height of h_i.
- A tie-down is present at the bottom of the wall's leading edge.
- A horizontal load is applied at the top of the wall. It is assumed that the floor zone above can distribute this load along the wall's length as required, and that the floor zone below can provide a reaction to this load.
- A tie-down may be present at the top of the wall's leading edge. This tie-down may transfer a tension force from the floor above.
- Vertical loads are applied at the top of the wall (shown as a non-uniform distributed load in red in Figure 1). There may be a length on the top of the wall at the leading edge where there is no compressive load. The exact distribution of the loading will depend on many factors, such as where along the wall the loads are applied, the destabilising moment from the horizontal loads, the behaviour of any walls above, and the behaviour of the floor above.
- There are vertical reactions at the bottom of the wall (again shown as a non-uniform distributed load in red in Figure 1). These will be different from the vertical loads at the top of the wall. There may be a length on the bottom of the wall at the leading edge where there is no compressive load. This length will be longer than or equal to the length in tension on the top of the wall. The exact distribution of the reaction will depend on the loading on top of the wall, the horizontal load, and the mechanical behaviour of the wall.

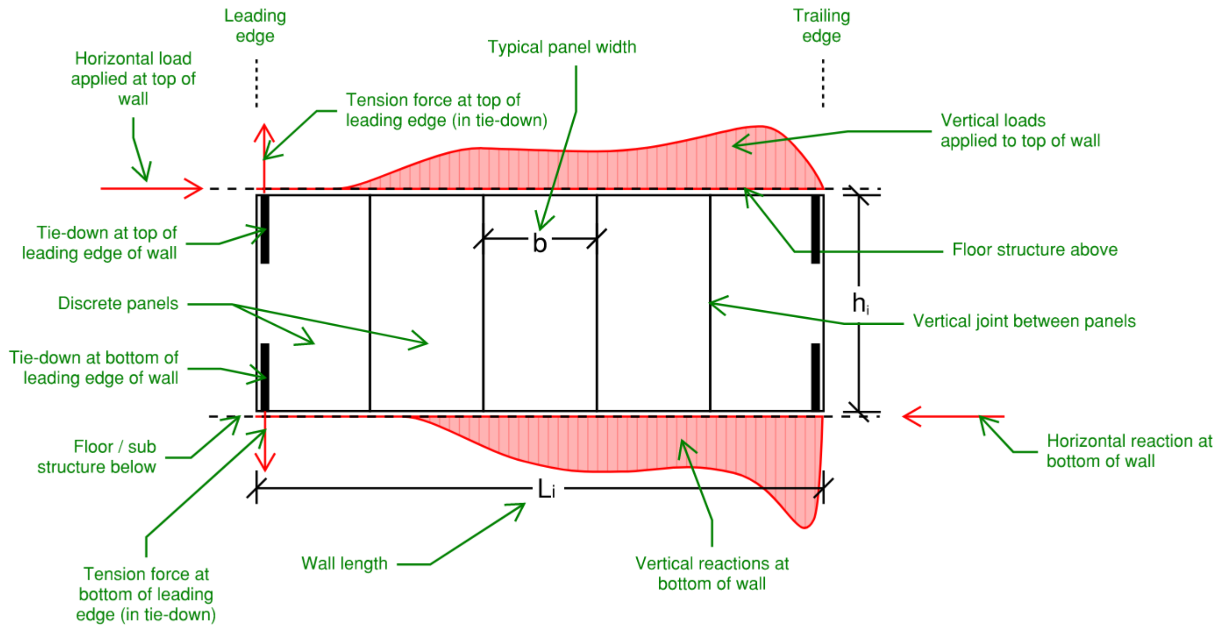


Figure 1: The layout, features and actions applied to a segmented wall between two storeys.

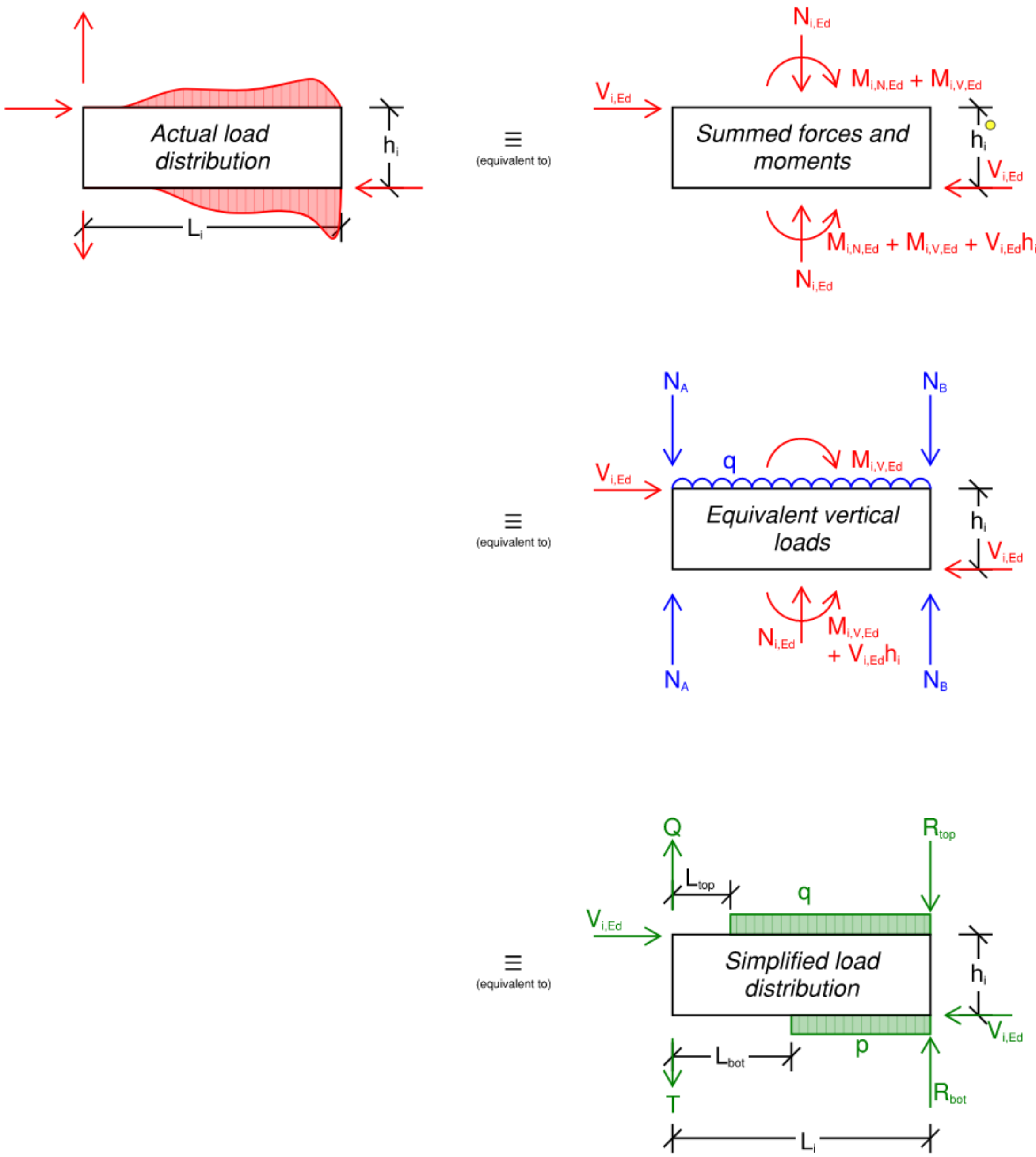


Figure 2: Simplification of forces acting on a wall between two storeys.

2.2 Simplification of vertical loads (step 1 of method)

In order to have an analytical method which can be carried out by hand / spreadsheet, it is necessary to simplify the vertical actions on the top and bottom of the wall. Due to the complex nature of multi-storey segmented CLT shear walls it is very hard to derive the actual load distribution on the top and bottom of the wall (top left in Figure 2), but it is easy to sum up the actions (top right in Figure 2) to calculate:

- $\mathbf{N_{i,Ed}}$: The design vertical force acting at the top of the wall at the i^{th} storey.
- $\mathbf{M_{i,N,Ed}}$: The design moment acting at the top of the wall at the i^{th} storey due to vertical loads only, taken about centreline of wall (destabilising positive, stabilising negative).
- $\mathbf{V_{i,Ed}}$: The design horizontal shear force acting at the top of the wall at the i^{th} storey.
- $\mathbf{M_{i,V,Ed}}$: The design destabilising moment acting at the top of the wall at the i^{th} storey due to horizontal loads only.

Throughout this document, these summed forces are shown in **red**.

The method uses these summed forces to calculate a simplified set of actions on the wall at the i^{th} storey (bottom right in Figure 2). These are shown in **green** throughout this document and are calculated in step 1 of the method:

- $\mathbf{V_{i,Ed}}$: The horizontal force is taken as the summed value.
- **Q**: A vertical force at the top of the leading wall edge. This may be tensile (e.g. a force from a tie-down above, taken as positive), or compressive (e.g. due to a stabilising load at the wall's leading edge, taken as negative).
- **T**: A tension force in the tie-down at the bottom of the leading wall edge.
- $\mathbf{R_{top}}$ & $\mathbf{R_{bot}}$: Vertical forces at the top and bottom of the wall's trailing edge. These forces are not required for calculating the deflection and so their equations have not been included in the proposal.
- **q**: A UDL on the top of the wall, starting a distance l_{top} from the leading wall edge.
- **p**: A UDL on the bottom of the wall, starting a distance l_{bot} from the leading wall edge. Due to the stiffening effect (see Section 2.6), $p \leqslant q$. The method does not require p to be explicitly calculated in order to calculate deflections, and so its equations have not been included in the proposal.

Segmented walls generally stabilise themselves against overturning by progressively shifting vertical loads from the leading wall edge to the trailing wall edge. Therefore, the method initially calculates an intermediate set of loads (shown in **blue** in the middle right of Figure 2) comprising the UDL q and two downwards points loads N_A and N_B on the wall's leading and trailing ends. These are calculated based on the vertical loads only ($N_{i,Ed}$ and $M_{i,N,Ed}$). The method then incorporates the horizontal loads ($V_{i,Ed}$ and $M_{i,V,Ed}$), giving the green set of forces shown bottom right in Figure 2.

2.3 Relative stiffness of tie-down & vertical joint between panels

The deformations of a segmented wall depend on the stiffness of two connections:

- $\mathbf{K_h}$: the stiffness of the tie-down at the bottom of the wall's leading edge.
- $\mathbf{K_v}$: the total stiffness of the vertical joint between two panels.

When horizontal loads are applied to segmented walls, each panel rotates about its lower trailing corner. Depending on the relative magnitudes of K_h and K_v, some panels at the leading end of the wall may lift up entirely off the floor below. The proposed method therefore sorts segmented walls into two categories:

- Those where $\mathbf{K_h} \geqslant \mathbf{K_v}$: In these walls it is assumed that all panels remain in contact with the floor below, but the panels may rotate about their lower trailing corner. The proposed method analyses these walls as a series of discrete panels (see Section 2.4).
- Those where $\mathbf{K_h} < \mathbf{K_v}$: In these walls, the panels may rotate about their lower trailing corner, but some panels at the leading edge may lift up entirely off the floor below. To accommodate this effect, the proposed method approximates these walls to a shear beam (see Section 2.5).

For walls with one panel, K_v should be taken as 0.

2.4 Discrete panel model (for walls where $\mathbf{K_h} \geqslant \mathbf{K_v}$)

For walls where $K_h \geqslant K_v$ the method assumes that all the panels remain in contact with the floor below, giving the model show in Figure 3. This simple model can be easily solved for the deflection u_R in terms of $V_{i,Ed}$.

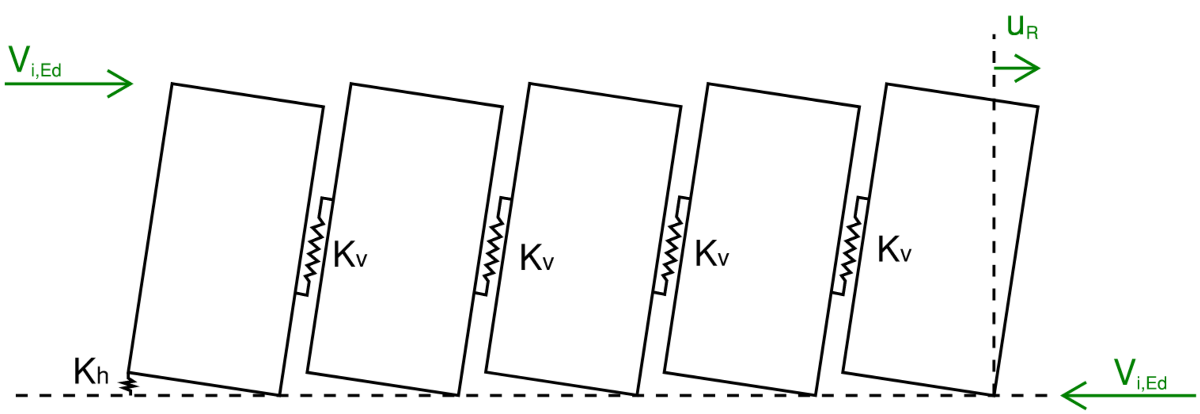


Figure 3: Discrete model for walls where $K_h \geqslant K_v$ (vertical loads omitted for clarity).

2.5 Shear beam model (for walls where $\mathbf{K_h < K_v}$)

For walls where $K_h < K_v$, the method smears out the stiffness of the vertical panel joints to get a continuous shear beam. The standard deflection differential equation for a shear beam is as follows:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{S}{kAG}$$

where z is the vertical deflection, x is the distance along the beam, S is the vertical shear force (which varies with x), and kAG is the shear stiffness multiplied by the shape factor. For a segmented wall, the method substitutes $kAG = K_v b$ to get:

$$\frac{\mathrm{d}z}{\mathrm{d}x} = \frac{S}{K_v b}$$

where K_v is the stiffness of the vertical joint between panels, and b is the panel width.

The tie-down spring K_h is separated into two component springs acting in series as shown in Figure 4:

- The first component spring is taken as equal to the stiffness of the joint between panels (K_v). This component gets smeared over the first panel as part of the shear beam analogy explained above.
- The second component spring (K_{res}) includes the residual stiffness. This spring acts at the bottom of the shear beam's leading edge – see Figure 5 – and is calculated as follows:

$$K_{res} = \frac{1}{\dfrac{1}{K_h} - \dfrac{1}{K_v}}$$

The resulting shear beam is shown in Figure 5. It is assumed that the effective longitudinal bending stiffness EI of this beam is very large, and so the overall behaviour is governed by the shear stiffness.

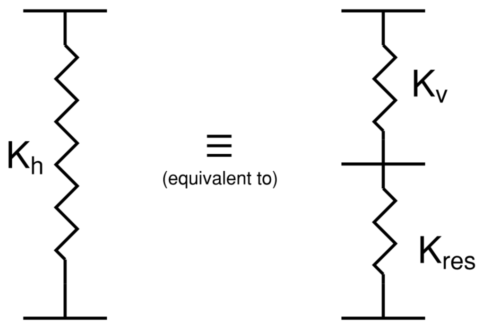


Figure 4: Tie-down spring broken down into two springs acting in series.

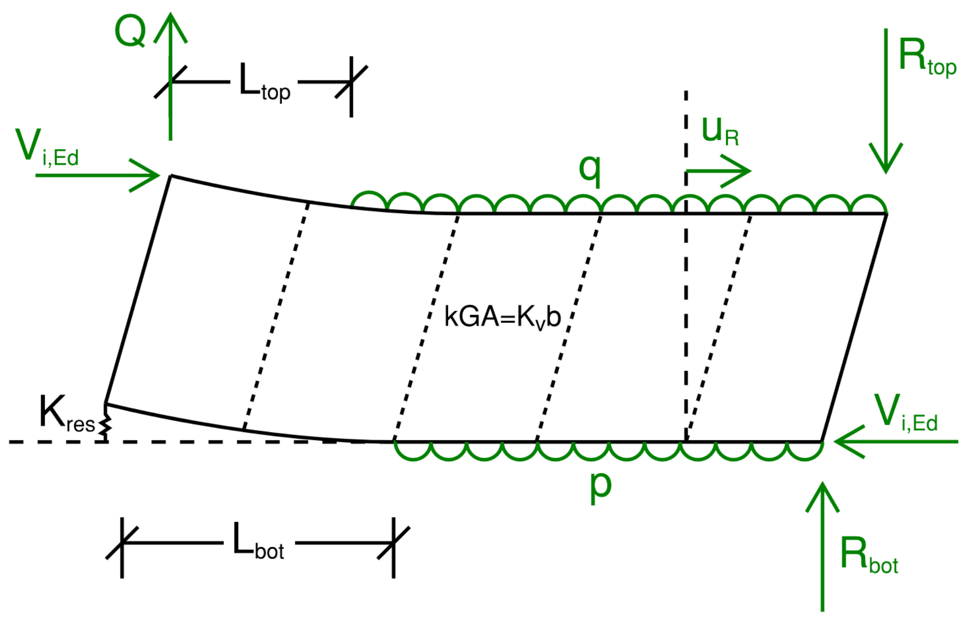


Figure 5: Shear beam model for walls where $K_h < K_v$.

2.6 Initial "stiff" response due to downwards vertical loads

When downwards vertical loads are applied to segmented CLT walls, they have a stiffening effect on its behaviour. The CLT panels are considered as rigid bodies on a rigid foundation, and so if a vertical load is applied on the top of a panel then it takes a certain amount of horizontal load before the panel can start rotating.

Consider the three panels shown in Figure 6, each of which have a UDL q applied to the top:

A) In this panel there is no horizontal force, and so the reaction under the panel is the UDL q. The deflection of the panel $u_R = 0$

B) As H increases, the compression block becomes trapezoidal and then triangular. In reality there will be some small embedment of the CLT panel into the structure below, but this is assumed to be small. It is therefore assumed that the deflection $u_R \approx 0$.

C) When $H = qb^2/2h$, the bottom reaction becomes a point load at the panel's trailing end. Again, any embedment under the trailing corner is assumed to be small. Therefore, if $H \leqslant qb^2/2h$ then it is assumed that $u_R \approx 0$. At this point the panel is able to freely rotate. If H is increased above $qb^2/2h$ then any further movement needs to be resisted by a tie-down and/or connections between the panels.

The method therefore assumes that the wall initially has a very stiff horizontal response (i.e. $u_R = 0$) until this effect is overcome. The amount of stiffening will depend on the extent of the vertical loads applied to the top of the wall. This applies to both the discrete panel model (for walls where $K_h \geqslant K_v$, see Section 2.4) and the shear beam model (for walls where $K_h < K_v$, see Section 2.5).

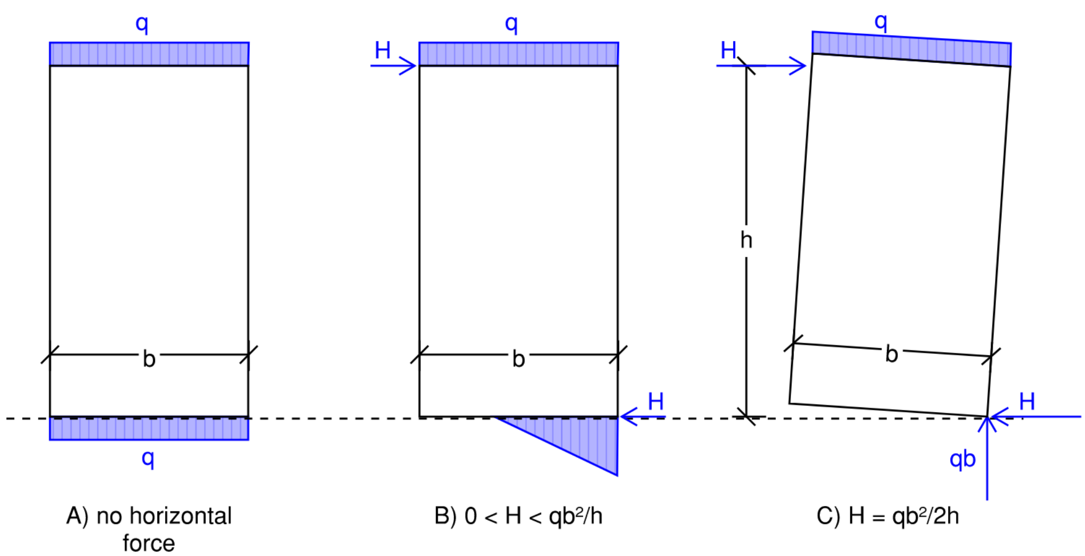


Figure 6: Stiffening effect due to vertical loads.

2.7 Multi-storey behaviour

The proposed method analyses multi-storey walls on a storey-by-storey basis. This is similar to the accepted methods for fully-anchored framed walls, partially-anchored framed walls and solid CLT walls, and is compatible with the new multi-storey deflections Annex Y. For a wall at a given storey i, the proposed method outputs:

- $\mathbf{u_{R,i}}$ — the horizontal deflection of the top of the wall relative to the bottom of the wall
- $\varphi_{\mathbf{R,i}}$ — the rotation of the top of the wall relative to the bottom of the wall

The rotations are cumulative up the building, as per Equations Y.11 & Y.12 of Annex Y. This is shown indicatively in Figure 7.

The proposed method ignores interactions between storeys, again similar to the accepted methods for fully-anchored framed walls, partially-anchored framed walls and solid CLT walls. All these methods effectively assume that the wall at each storey sits on a straight surface, shown by the dashed lines in Figure 7. These straight lines may not be parallel to each other. The white gaps between the walls at each storey demonstrate the potential lack of compatibility between one wall and the wall above.

In reality, if a wall at one floor tries to peel up over a certain length (e.g. as shown in Figure 7), it will also push up the walls on the storeys above. However, the exact behaviour is very hard to predict (even with finite element analysis) because it depends on the flexibility of the floor zone and the connections which are often not known in sufficient detail.

The adopted approach is generally believed to be conservative. If the walls at different storeys do interact, then the likely effects would be that:

- Overall deflections decrease,
- Tie-down forces decrease,
- The forces between panels at lower storeys decrease,
- The forces between panels at upper storeys increase.

All of these effects are expected to be small and so the results should not change significantly, however it would be good to verify this with finite element analysis and/or physical testing.

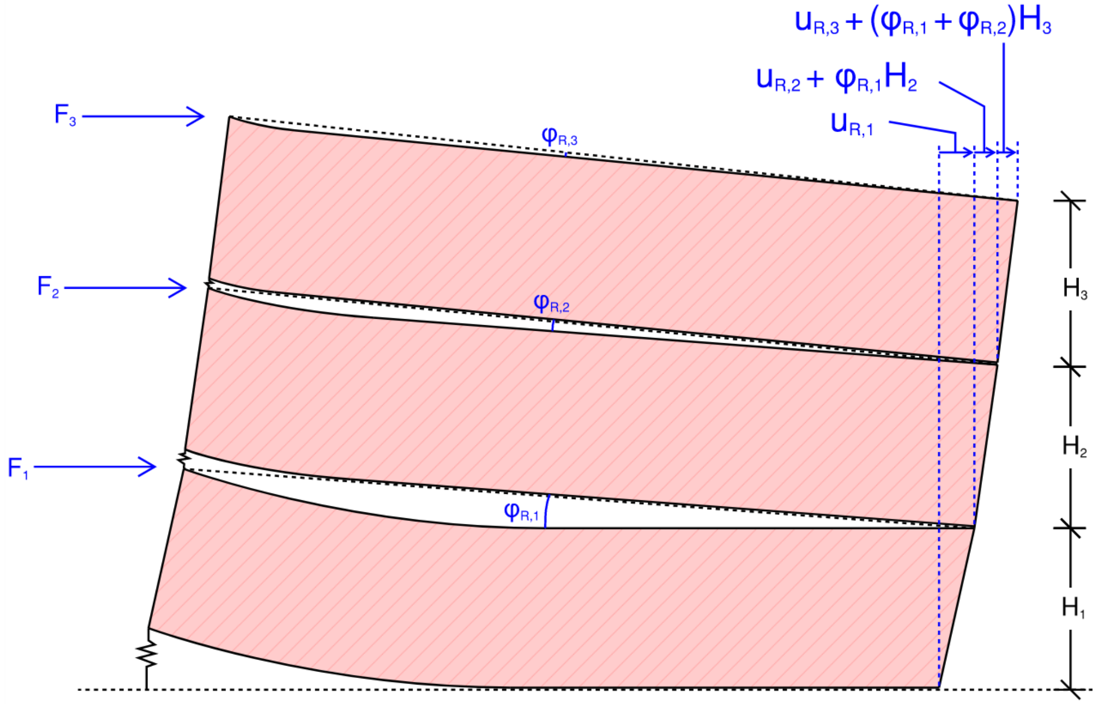


Figure 7: Deflections of a multi-storey wall as assumed by the proposed method (floor zone omitted for clarity).

2.8 Response mode and solutions (steps 2 and 3 of method)

To be updated

3 Symbols

b — Width of one panel

h_i — Height of the shear wall at the i^{th} storey

l_i — Length of the shear wall at the i^{th} storey

l_{bot} — Length not in compression at the bottom of the wall

l_{top} — Length not in compression at the top of the wall

q — Uniformly distributed load along the top of the wall due to vertical loads

$u_{R,i}$ — The inter-storey lateral displacement due to the rocking kinematic mode of the shear wall at the i^{th} storey

K_{res} — Spring accounting for the difference in stiffness between the tie-down and the vertical joints between the panels

K_h — Vertical K_{ser} stiffness of the tie-down at the wall's leading edge

K_v — Total vertical K_{ser} stiffness of the joint between two panels

$M_{i,N,Ed}$ — Design moment acting at the top of the wall at the i^{th} storey due to vertical loads only, taken about centreline of wall (destabilising positive, stabilising negative).

$M_{i,V,Ed}$ — Design destabilising moment acting at the top of the wall at the i^{th} storey due to horizontal loads only.

$N_{i,Ed}$ — The design vertical force acting at the top of the wall at the i^{th} storey (assumed ≥ 0)

N_A — vertical load on leading edge of wall due to vertical loads (compression positive)

N_B — vertical load on trailing edge of wall due to vertical loads (compression positive)

Q — vertical force applied at top of leading wall edge (tension positive, compression negative)

R_{bot} — vertical upwards reaction on bottom of trailing wall edge (assumed ≥ 0)

R_{top} — vertical downwards force applied on top of trailing wall edge (assumed ≥ 0)

T — force in the tie-down at the bottom of the leading edge of the wall (tension positive)

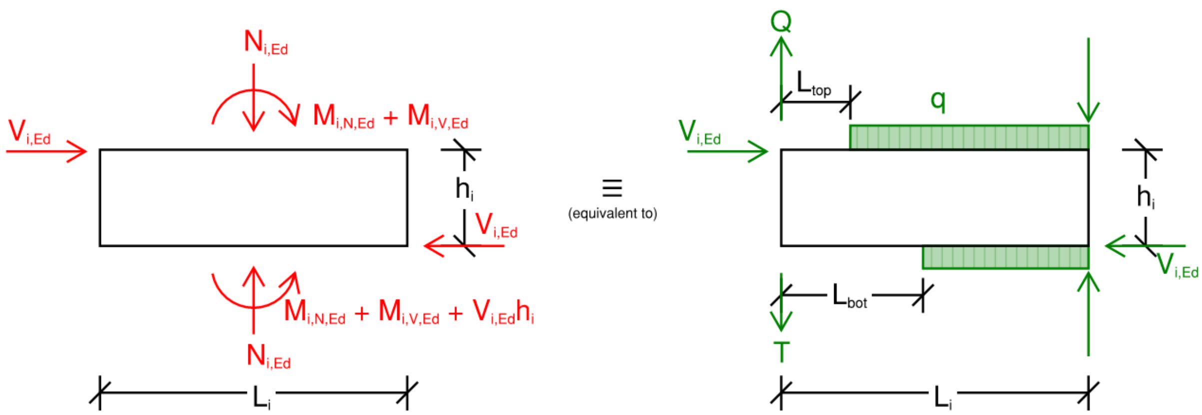
T_{above} — tension force in the tie-down at the bottom of the leading edge of the wall at the floor above (if present) under the same set of loads, 0 otherwise

$V_{i,Ed}$ — The design horizontal shear force acting at the top of the wall at the i^{th} storey

$\varphi_{R,i}$ — Rotation contribution due to the rocking kinematic mode of the shear wall at the i^{th} storey

4 Proposed code clauses

(Clause 1) The loads acting on the wall at the i^{th} storey may be simplified to an equivalent set of loads as shown in the Figure below. The loads on the top of the wall may be simplified to a uniformly distributed load q starting a distance l_{top} from the leading wall edge, a vertical point force Q at the leading wall edge, a downwards vertical point force at the trailing wall edge, and a horizontal force $V_{i,Ed}$. The loads acting on the bottom of the wall may be simplified to a uniformly distributed load starting a distance l_{bot} from the leading wall edge, a tension force T at the leading wall edge, an upwards vertical point force at the trailing wall edge, and a horizontal force $V_{i,Ed}$. Q, q and l_{top} may be calculated according to Equations (1), (4) and (5) respectively.



$$Q = \begin{cases} \dfrac{M_{i,V,Ed}}{l_i} - N_A & \text{if } l_{top} = 0 \\ \dfrac{M_{i,V,Ed}}{l_i} - N_A - \dfrac{ql_i}{2} & \text{if } l_{top} = l_i \\ T_{above} & \text{otherwise} \end{cases} \tag{1}$$

where:

- T_{above} is the tie-down force at the bottom of the leading edge of the floor above (if present) under the same set of loads, 0 otherwise.
- $M_{i,V,Ed}$ is the design destabilising moment acting at the top of the wall at the i^{th} storey due to horizontal loads only.
- N_A and N_B are calculated according to the equations below:

$$N_A = \begin{cases} -2M_{i,N,Ed}/l_i & \text{if } M_{i,N,Ed} < 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$N_B = \begin{cases} 2M_{i,N,Ed}/l_i & \text{if } M_{i,N,Ed} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where:

- $M_{i,N,Ed}$ is the design moment acting at the top of the wall at the i^{th} storey due to vertical loads only, taken about centreline of wall (destabilising positive, stabilising negative). It is assumed that the effective line of actions of $N_{i,Ed}$ sits within the length of the wall (i.e. $-l_i/2 \leq M_{i,N,Ed}/N_{i,Ed} \leq l_i/2$).

$$q = \frac{N_{i,Ed} - N_A - N_B}{l_i} \tag{4}$$

$$l_{top} = \begin{cases} 0 & \text{if } M_{i,V,Ed} \leqslant N_A l_i + T_{above} l_i \\ l_i & \text{if } M_{i,V,Ed} \geqslant N_A l_i + T_{above} l_i + \dfrac{q l_i^2}{2} \\ l_i - \sqrt{l_i^2 - \dfrac{2\left(M_{i,V,Ed} - N_A l_i - T_{above} l_i\right)}{q}} & \text{otherwise} \end{cases} \tag{5}$$

(Clause 2) The response mode of the wall at the i^{th} storey may be derived from Table 1 if $K_h < K_v$ or Table 2 if $K_h \geqslant K_v$, where K_h is the vertical K_{ser} stiffness of the tie-down at the wall's leading edge and K_v is the total vertical K_{ser} stiffness of the joint between two panels. For walls consisting of only one panel, K_v should be taken as 0.

Table 1: Response mode if $\mathbf{K_h < K_v}$ (see definitions of $\mathbf{l_{bot,Q}}$ & $\mathbf{K_{res}}$ below):

Load condition	$\mathbf{V_{i,Ed}}$ **condition**	**Mode**
$Q < 0$	$V_{i,Ed} \leqslant \dfrac{ql_ib}{2h_i} - \dfrac{Qb}{h_i}$	A
	$\dfrac{ql_ib}{2h_i} - \dfrac{Qb}{h_i} < V_{i,Ed} \leqslant \dfrac{ql_ib}{2h_i} - \dfrac{Ql_i}{h_i}$	B
	$\dfrac{ql_ib}{2h_i} - \dfrac{Ql_i}{h_i} < V_{i,Ed} \leqslant \dfrac{qK_{res}l_i^2(l_i - b)}{2K_vbh_i} + \dfrac{ql_i^2}{2h} - \dfrac{Ql_i}{h_i}$	C
	Otherwise	D
$Q \geqslant 0$ **&**	$V_{i,Ed} \leqslant \dfrac{qb\,(l_i - l_{top})}{2h_i} - \dfrac{Qb}{h_i}$	A
$0 \leqslant l_{top} \leqslant l_i - b$	$\dfrac{qb\,(l_i - l_{top})}{2h_i} - \dfrac{Qb}{h_i} < V_{i,Ed}$ **&** $V_{i,Ed} \leqslant \dfrac{qK_{res}l_i(l_i - l_{top} - b)(l_i + l_{top})}{2K_vbh_i} + \dfrac{q(l_i - l_{top})^2}{2h_i} - \dfrac{Ql_i}{h_i}$	C
	Otherwise	D
$Q \geqslant 0$ **&**	$V_{i,Ed} \leqslant \dfrac{q\,(l_i - l_{top})^2}{2h_i} - \dfrac{Qb}{h_i}$	A
$l_i - b < l_{top} \leqslant l_i$	Otherwise	E

Table 2: Response mode if $\mathbf{K_h \geqslant K_v}$:

Q condition	$\mathbf{V_{i,Ed}}$ **condition**	**Mode**
$Q \leqslant 0$	$V_{i,Ed} \leqslant \frac{ql_i b}{2h_i} - \frac{Qb}{h_i}$	F
	Otherwise	G
Q $>$ 0	$V_{i,Ed} \leqslant \frac{q\left(l_i - l_{top}\right)b}{2h_i}$	H
	Otherwise	I

where:

$$K_{res} = \frac{1}{\frac{1}{K_h} - \frac{1}{K_v}} \tag{6}$$

(Clause 3) The length not in compression on the bottom of the wall (l_{bot}), the tie-down force (T), the force in the connection between the panels (F_v), the horizontal deflection ($u_{R,i}$) and the rotation of the top of the wall relative to the wall below ($\varphi_{R,i}$), all at the i^{th} storey may be calculated from Tables 3 to 7.

Table 3: Solutions for $\mathbf{l_{bot}}$ depending on response mode:

Mode	$\mathbf{l_{bot}}$
A, B	l_{top}
C	$-\frac{b\left(K_{res}l_i+2K_vl_i-K_vb\right)}{2\left(K_{res}l_i-K_vb\right)}+\frac{K_vb}{K_{res}l_i-K_vb}\times \sqrt{\left(\frac{K_{res}l_i+2K_vl_i-K_vb}{2K_v}\right)^2+\frac{K_{res}l_i-K_vb}{K_vb}\left(\frac{2(V_{i,Ed}h_i+Ql_i)}{q}-l_ib+l_{top}\left(2l_i-l_{top}\right)+\frac{K_{res}l_il_{top}\left(l_{top}+b\right)}{K_vb}\right)}$
D, E	l_i
F, G, H, I	l_{top}

Table 4: Solutions for T depending on response mode:

Mode	**T**
A, B, F	0
C	$\frac{qK_{res}}{2K_vb}\left(l_{bot}-l_{top}\right)\left(l_{bot}+l_{top}+b\right)$
D, E	$\frac{V_{i,Ed}h_i}{l_i}+Q-\frac{q\left(l_i-l_{top}\right)^2}{2l_i}$
G	$\frac{K_hb\left(V_{i,Ed}h_i+Qb-ql_ib/2\right)}{K_hb^2+K_vb(l_i-b)}$
H	Q
I	$Q+\frac{K_hb\left(V_{i,Ed}h_i-q(l_i-l_{top})b/2\right)}{K_hb^2+K_vb(l_i-b)}$

Table 5: Solutions for $\mathbf{F_v}$ depending on response mode:

Mode	$\mathbf{F_v}$
A, F, H	0
B	$\frac{V_{i,Ed}h_i}{l_i} - \frac{qb}{2}$
C	$\max \begin{cases} T - Q + q\left(l_{bot} - l_{top}\right) \\ Q - T \end{cases}$
D	$\max \begin{cases} T - Q + q\left(l_i - l_{top} - b\right) \\ Q - T \end{cases}$
E	$\max \begin{cases} T - Q \\ Q - T \end{cases}$
G, I	$\frac{K_v T}{K_h}$

Table 6: Solutions for $\mathbf{u_{R,i}}$ depending on response mode:

Mode	$\mathbf{u_{R,i}}$
A, F, H	0
B	$\frac{h_i}{K_v b}\left(\frac{V_{i,Ed}h_i}{l_i} + \frac{Qb}{l_i} - \frac{qb}{2}\right)$
C	$\frac{h_i}{K_v b}\left(T - \frac{Q\left(l_i - b\right)}{l_i} + q\left(l_{bot} - l_{top}\right)\right)$
D	$Th_i\left(\frac{1}{k_{res}l_i} + \frac{1}{K_v b}\right) - \frac{Qh_i\left(l_i - b\right)}{K_v b l_i} + \frac{qh_i\left(l_i - l_{top} - b\right)\left(l_i + l_{top}\right)}{2K_v b l_i}$
E	$Th_i\left(\frac{1}{k_{res}l_i} + \frac{1}{K_v b}\right) - \frac{Qh_i\left(l_i - b\right)}{K_v b l_i}$
G, I	$\frac{F_v h_i}{K_v b}$

Table 7: Solutions for $\varphi_{\mathbf{R,i}}$ depending on response mode:

Mode	$\varphi_{\mathbf{R,i}}$
A, B, C, D, E	$\dfrac{T}{K_{res} l_i}$
F, G, H, I	0

Appendix C - Multistory comparison

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 24 15:04:53 2023

@author: stian


Arup proposal
"""

import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import math


def find_Na_Nb(M_Ned, li):
    if M_Ned < 0:
        Na = -2 * M_Ned / li
        Nb = 0
        print("Na is", Na, "N")
        print("Nb is", Nb, "N")
    else:
        Na = 0
        Nb = 2 * M_Ned / li
        print("Na is", Na, "N")
        print("Nb is", Nb, "N")
    return [Na, Nb]


def find_q(Na, Nb, li, Nied):
    q = (Nied - Na - Nb) / li
    print("q is",q, "N/m")
    return q


def find_l_top(M_Ved, Na, li, T_above, q):
    if M_Ved <= Na * li + T_above * li:
        l_top = 0
    elif M_Ved >= (Na * li + T_above * li + (q * (li) ** 2) / 2):
        l_top = li
    else:
        l_top = li - (li ** 2 - (2 * (M_Ved - Na * li - T_above * li)) /
q) ** 0.5
    print("L_top is",l_top, "m")
```

```
    return l_top


def find_Q(M_Ved, Na, li, T_above, q, l_top):
    if l_top == 0:
        Q = M_Ved / li - Na
    elif l_top == li:
        Q = M_Ved / li - Na - q * li / 2
    else:
        Q = T_above
    print("Q", Q)
    return Q


def find_l_bot_Q(li, l_top, b, Q, q, Ka, Kv, ):
    if Q <= 0:
        l_bot_Q = l_top
    elif Q >= q * (((Ka * (li + l_top)) / (2 * Kv * b)) + 1) * (li -
l_top):
        l_bot_Q = li
    else:
        l_bot_Q = (((Kv * b) / Ka + l_top) ** 2 + (2 * Q * Kv * b) / (Ka
 * q)) ** (0.5) - (Kv * b) / Ka
    print("l_bot_Q is", l_bot_Q, "m")
    return l_bot_Q


def find_mode(V_ed, q, Q, li, l_bot_Q, l_top, hi, b, Ka, Kv,Kt):
    if Kv > Kt:
        if Q < 0:
            # print("l_bot_Q == 0")
            if V_ed <= (q * li * b) / (2 * hi) - (Q * b) / hi:
                print("Mode A")
                info = Mode_A(Ka, li, l_bot_Q)
            elif ((V_ed > (q * li * b) / (2 * hi) - (Q * b) / hi) and (
                    V_ed <= (q * li * b) / (2 * hi) - (Q * li) / hi)):
                print("Mode B")
                info = Mode_B(q, Q, hi, b, li, Kv,l_top, V_ed)
            elif ((V_ed > (q * li * b) / (2 * hi) - (Q * li) / hi) and (
                    V_ed <= (q * Ka * li ** 2 * (li - b)) / (2 * Kv * b
 * hi) + (q * li ** 2) / (2 * hi) - (
                    Q * li) / hi)):
                print("Mode C")
                info = Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
            else:
                print("Mode D")
                info = Mode_D(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv)

        elif ((Q >= 0) and (0 <= l_top <= li - b)):
            # print("(l_bot_Q > 0) and (l_bot_Q <= li - b)")
            if V_ed <= ((q*b*(li-l_top))/(2*hi)-(Q*b/hi)):
```

```
                print("Mode A")
                info = Mode_A(Ka, li, l_bot_Q)
            elif ((q*b*(li-l_top))/(2*hi)-(Q*b/hi) < V_ed <= ((q * Ka *
li * (li - l_top - b) * (li + l_top)) / (2 * Kv * b * hi) + (q * (li -
l_top) ** 2) / (2 * hi) - (Q * li) / hi)):
                print("Mode C")
                info = Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
            else:
                print("Mode D")
                info = Mode_D(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv)
        elif ((Q >= 0) and (li-b < l_top <= li)):
            # print("(l_bot_Q > li - b) and (l_bot_Q < li)")
            if V_ed <= (q * (li - l_top)**2) / (2 * hi) - (Q*b)/hi:
                print("Mode A")
                info = Mode_A(Ka, li, l_bot_Q)
            else:
                print("Mode E")

                info = Mode_E(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv)
    else:
        if Q <= 0:
            if V_ed <= (q * li * b) / (2 * hi) - (Q * b) / hi:
                print("Mode F")
                info = Mode_F(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv)
            else:
                print("Mode G")
                info = Mode_G(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv, Kt)
        else:
            if V_ed <= (q * (li - l_top) * b) / (2 * hi):
                print("Mode H")
                info = Mode_H(l_top, Q)

            else:
                print("Mode I")
                info = Mode_I(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka,
Kv, Kt)

    return info


def Mode_A(Ka, li, l_top):
    l_bot = l_top
    T = 0
    Fv = 0
    U_ri = 0
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]
```

```
def Mode_B(q, Q, hi, b, li, Kv, l_top, V_ed):
    l_bot = l_top
    T = 0
    Fv = (V_ed*hi)/li - (q*b)/2
    U_ri = hi/(Kv*b)*((V_ed*hi)/li+(Q*b)/li-(q*b)/2)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv):
    l_bot = -(b*(Ka*li+2*Kv*li-Kv*b))/(2*(Ka*li-Kv*b))+((Kv*b)/(Ka*li-Kv*
b))*(((Ka*li+2*Kv*li-Kv*b)/(2*Kv))**2+((Ka*li-Kv*b)/(Kv*b))*((2*(V_ed*hi+
Q*li))/q -li*b+l_top*(2*li-l_top)+(Ka*li*l_top*(l_top+b))/(Kv*b)))**(0.5)
    # print("lbot",l_bot,"m")
    T = ((Ka * q) / (2 * Kv * b)) * (l_bot - l_top) * (l_bot + l_top+b)
    # print("T",T,"N")
    Fv = max(T-Q+q*(l_bot-l_top),Q-T)
    # print("Fv",Fv,"N")
    U_ri = hi/(Kv*b)*(T-(Q*(li-b)/li)+q*(l_bot-l_top))
    # print("U",U_ri,"m")
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_D(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka, Kv):
    l_bot = li
    # print(l_bot)
    T = (V_ed*hi)/li + Q - (q*(li-l_top)**2)/(2*li)
    # print(T)
    Fv =  max(T-Q+q*(li-l_top-b),Q-T)
    # print(Fv)
    U_ri = T*hi*(1/(Ka*li)+1/(Kv*b))-(Q*hi*(li-b))/(Kv*b*li)+(q*hi*(li-
l_top-b)*(li-l_top))/(2*Kv*b*li)
    # print(U_ri)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_E(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka, Kv):
    l_bot = li
    T = (V_ed * hi) / li + Q - (q * (li - l_top) ** 2) / (2 * li)
    Fv = max(T-Q,Q-T)
    U_ri = T*hi*(1/(Ka*li)+1/(Kv*b))-(Q*hi*(li-b))/(Kv*b*li)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_F(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka, Kv):
    l_bot = l_top
    T = 0
    Fv = 0
```

```
    U_ri = 0
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_G(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka, Kv, Kt):
    l_bot = l_top
    T = (Kt * b * (V_ed * hi - Q * b - q * li * b / 2)) / (Kt * b ** 2 +
Kv * b * (li - b))
    Fv = (Kv * T) / (Kt)
    U_ri = (Fv * hi) / (Kv * b)
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_H(l_top, Q):
    l_bot = l_top
    T = Q
    Fv = 0
    U_ri = 0
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_I(q, Q, V_ed, l_bot_Q, l_top, li, b, hi, Ka, Kv, Kt):
    l_bot = l_top
    T = Q + (Kt * b * (V_ed * hi - q * (li - l_top) * b / 2)) / (Kt * b
 ** 2 + Kv * b * (li - b))
    Fv = (Kv * T) / (Kt)
    U_ri = (Fv * hi) / (Kv * b)
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def find_displacement(n_story):
    Nied = []
    T_above = 0
    Ui = []
    φi = []
    phii = []
    deflectionEachFloor = []
    for i in range(n_story):
        Kh = Kh_arup[i]
        Kt = Kh

        Ka = 1 / ((1 / Kh) - (1 / Kv))  # N/m
        Nied.append(qed[i] * li)
        print("floor", n_story - i)
        print("Kh", Kh)
        M_ed_i = []
        teller = n_story-i
        for k in range(n_story-(teller)):
```

```
            M_ed_i.append(V_ed_i*(hi*(k+1)+0.2*k))
        M_Ved = sum(M_ed_i)
        V_ed = V_ed_i*(i+1)
        print("M_Ved", M_Ved)
        print("V_ed",V_ed)
        NaNb = find_Na_Nb(M_Ned, li)
        Na = NaNb[0]
        Nb = NaNb[1]
        q = find_q(Na, Nb, li, sum(Nied))
        l_top = find_l_top(M_Ved, Na, li, T_above, q)
        Q = find_Q(M_Ved, Na, li, T_above, q, l_top)
        l_bot_Q = find_l_bot_Q(li, l_top, b, Q, q, Ka, Kv, )
        info = find_mode(V_ed, q, Q, li, l_bot_Q, l_top, hi, b, Ka, Kv,Kt
)
        T_above = info[1]
        Ui.append(info[3])
        φi.append(info[4])
        print("Ka",Ka)
        print("T", info[1])
        print("Ui",Ui)
        print("phii",φi)


    for n in range(len(Ui)):
        phii = φi[n + 1:]
        phi = 0
        print("phii",phii)
        for m in phii:
            phi += m
            print("phi",phi)
        deflectionEachFloor.append(phi * hi + Ui[n])
    print("Displacement each floor",deflectionEachFloor)
    deflection = sum(deflectionEachFloor) * 10 ** 3
    return deflection

    for n in range(len(Ui)):
        phii = φi[n+1:]
        phi = 0
        #print("phii",phii)
        for m in phii:
            phi += m
            #print("phi",phi)
        deflectionEachFloor.append(phi*hi + Ui[n])

    deflection = sum(deflectionEachFloor)*10**3
    print("Ui", Ui)
    print(deflection)
    return deflection


"Annex Y"
def Annex_y(Kh,Kv,hi,li,lj,m,N_ed,qed,M_ed,N_dim,V_ed):
```

```
    #INTERPOLATION
    print("V_ed",V_ed)
    K_CP = ((Kh + (m-1)*Kv)/(m**2))*li**2
    K_SW = (1/(Kh)+(m-1)/(Kv))**(-1)*li**2

    if (Kh/Kv) >= ((1-N_dim*(3*m-2)/(m**2))/(1-N_dim*(m-2)/(m**2))):
        #Coupled mode
        Ur_CP = max(((M_ed/K_CP)-(N_ed*lj)/(2*K_CP))*hi, 0)
        Ur_CP = Ur_CP*10**3
        print("UR_CP",Ur_CP)
        return Ur_CP

    elif(Kh/Kv) <= ((1-N_dim)/(1+N_dim*(m-2))):
        #Single wall mode
        Ur_SW = max(((M_ed/K_SW)-(N_ed/(2*Kh*li)))*hi,0)
        Ur_SW = Ur_SW*10**3
        print("UR_SW",Ur_SW)
        return Ur_SW

    else:
        #intermediate mode
        print("interpolation")
        V_CP = (b**2*qed*(Kh*(m - 2) + Kv*(2 - 3*m)))/(2*hi*(Kh - Kv))
        Ur_CP = (hi*(2*hi*V_CP - b**2*m*qed))/(2*b**2*(Kh + Kv*(m - 1)))

        V_SW = (b**2*m**2*qed*(Kh*(2-m)-Kv))/(2*hi*(Kh-Kv))
        Ur_SW = (hi*(hi*V_SW*(Kh*(2*m - 2) + 2*Kv) - Kv*b**2*m**2*qed))/(
2*Kh*Kv*b**2*m**2)


        Ur_IN = max((Ur_CP*(V_SW - V_ed) + Ur_SW*(V_ed - V_CP))/(V_SW -
V_CP),0)
        Ur_IN = Ur_IN*10**3
        print("UR_IN",Ur_IN)
        return Ur_IN


def Tot_def_annex_y(n_story):
    print("-----Annex y interpolation-----")
    qed_i = []
    Uri = []
    φri = []
    theta_i = []
    deflection_i = []
    for n in range(n_story) :
        if n == (n_story-1):
            Kh = Kh_3
        elif n == 0:
            Kh = Kh_1
        else:
            Kh = Kh_2
        print("Kh",Kh)
```

```
        M_ed_i = []
        for k in range(n_story-n):
            M_ed_i.append(V_ed_i*((hi)*(k+1)+0.2*k))

        M_ed = sum(M_ed_i)
        V_ed = V_ed_i*(n_story-n)

        qed_i = vertical_load[n]
        print("qed",qed_i)
        N_ed=qed_i*li
        N_dim = ((N_ed)*li)/(2*M_ed)
        print("Med",M_ed)
        Uri.append(Annex_y(Kh,Kv,hi,li,lj,m,N_ed,qed_i,M_ed,N_dim,V_ed))

        if (Kh/Kv) >= ((1-N_dim*(3*m-2)/(m**2))/(1-N_dim*(m-2)/(m**2))):
            φri.append(0)
        else:
            φri.append(Uri[n]/hi)


    for n in range(len(Uri)):
        thetai = φri[:n]
        #print("thetai",thetai)
        theta = 0
        for j in thetai:
            theta += j
            #print("theta",theta)
        deflection_i.append(Uri[n] + theta*hi)
        #print("deflection_i",deflection_i)
    print("φri",φri)
    print("Uri",Uri)
    print("Deflection each floor",deflection_i)
    deflection = sum(deflection_i)
    print("Delfection",deflection)
    return deflection




Kh_1 = 6000000
Kh_2 = 3000000
Kh_3 = 3000000

Kvert=[9000000]
Kh_arup = [3000000+0.001, 6000000+0.001, 6000000]


M_Ned = 0 #Nm
qed = [1000,1000]
vertical_load=[2000,1000]
panels = [3]
```

```
colours = ['red','blue','green', 'yellow', 'black','orange']


for i in range(len(panels)):
    for a in range(len(vertical_load)):
        fig = plt.figure(figsize=(1920 / 100, 1080 / 100))
        plt.xticks(fontsize=25)
        plt.yticks(fontsize=25)
        for x in range(len(Kvert)):
            # Property Arup and Annex Y
            m = panels[i]  # number of panels
            b = 1.4   # m
            li = m * b   # m
            lj = 1.4   # m, Same as b in Arup method.
            hi = 2.7 # m
            dens = 420   # kg/m3
            Kv = Kvert[x]

            # number of storys
            n_story = 2

            V_ed_i = 1
            plots = 11
            x1 = []
            y1 = []
            y2 = []
            #Nied = qed*li #N
            for n in range(plots):
                displacement_arup = find_displacement(n_story)
                displacement_annex_y = Tot_def_annex_y(n_story)
                x1.append(V_ed_i)
                y1.append(displacement_arup)
                y2.append(displacement_annex_y)
                V_ed_i += 1000
                print("V_ed",V_ed_i)
                print("-----")
                print("displacement annex y interpolation",
displacement_annex_y)
                print("displacement Arup",displacement_arup)
            plt.plot(x1, y2,linewidth = 2, color=colours[x], linestyle=
'-.', label = "Annex R inter, Kv = {} kN/m".format(round(Kv*10**(-3))))
            plt.plot(x1, y1,linewidth = 2, color = colours[x],label = "
Arup, Kv = {} kN/m".format(round(Kv*10**(-3))))

        plt.xlim(0)
        plt.ylim(0)
        plt.grid(True)
        plt.xlabel('Horizontal load [N]',fontsize = 25)
        plt.ylabel('Displacement [mm]',fontsize = 25)
        plt.title("q_i = {} N/m, Kh_GF = {} kN/m, Kh_FF = {} kN/m, {}
panels, {} floors".format(qed[0], round(Kh_arup[1]*10**(-3)),round(
```

```
Kh_arup[0]*10**(-3)), m,n_story),fontsize = 25)
        plt.legend(fontsize = 25)
        plt.savefig("//Client/C$/Result/Two_story/Two story annex y and
arup 10k"+str(m)+"_qed"+str(vertical_load[a]))
        #plt.show()
```

Appendix D

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 24 15:04:53 2023

@author: stian


Arup proposal
"""
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import math


def find_Na_Nb(M_Ned, li):
    if M_Ned < 0:
        Na = -2 * M_Ned / li
        Nb = 0
    else:
        Na = 0
        Nb = 2 * M_Ned / li
    return [Na, Nb]


def find_q(Na, Nb, li, Nied):
    q = (Nied - Na - Nb) / li
    print("q is",q, "N/m")
    return q


def find_l_top(M_Ved, Na, li, T_above, q):
    if M_Ved <= Na * li + T_above * li:
        l_top = 0
    elif M_Ved >= (Na * li + T_above * li + (q * (li) ** 2) / 2):
        l_top = li
    else:
        l_top = li - (li ** 2 - (2 * (M_Ved - Na * li - T_above * li)) /
q) ** 0.5
    print("L_top is",l_top, "m")
    return l_top


def find_Q(M_Ved, Na, li, T_above, q, l_top):
    if l_top == 0:
        Q = M_Ved / li - Na
    elif l_top == li:
        Q = M_Ved / li - Na - q * li / 2
```

```
    else:
        Q = T_above
    print("Q", Q)
    return Q


def find_l_bot_Q(li, l_top, b, Q, q, Ka, Kv, ):
    if Q <= 0:
        l_bot_Q = l_top
    elif Q >= q * (((Ka * (li + l_top)) / (2 * Kv * b)) + 1) * (li -
l_top):
        l_bot_Q = li
    else:
        l_bot_Q = (((Kv * b) / Ka + l_top) ** 2 + (2 * Q * Kv * b) / (Ka
 * q)) ** (0.5) - (Kv * b) / Ka
    return l_bot_Q


def find_mode(V_ed, q, Q, li, l_top, hi, b, Ka, Kv):
    if Kv > Kt:
        if Q < 0:
            # print("l_bot_Q == 0")
            if V_ed <= (q * li * b) / (2 * hi) - (Q * b) / hi:
                print("Mode A")
                info = Mode_A(Ka, li, top)
            elif (V_ed > (q * li * b) / (2 * hi) - (Q * b) / hi) and (
                    V_ed <= (q * li * b) / (2 * hi) - (Q * li) / hi):
                print("Mode B")
                info = Mode_B(q, Q, hi, b, li, Kv,l_top, V_ed)
            elif (V_ed > (q * li * b) / (2 * hi) - (Q * li) / hi) and (
                    V_ed <= (q * Ka * li ** 2 * (li - b)) / (2 * Kv * b
 * hi) + (q * li ** 2) / (2 * hi) - (
                    Q * li) / hi):
                print("Mode C")
                info = Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
            else:
                print("Mode D")
                info = Mode_D(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)

        elif ((Q >= 0) and (0 <= l_top <= li - b)):
            # print("(l_bot_Q > 0) and (l_bot_Q <= li - b)")
            if V_ed <= (q * b * (li - l_top))/(2 * hi) - (Q * b / hi):
                print("Mode A")
                info = Mode_A(Ka, li, l_top)
            elif (q * b * (li - l_top)) / (2 * hi) - (Q * b / hi) < V_ed
 <= (q * Ka * li * (li - l_top - b) * (li + l_top)) / (2 * Kv * b * hi
) + (q * (li - l_top) ** 2) / (2 * hi) - (Q * li) / hi:
                print("Mode C")
                info = Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
            else:
                print("Mode D")
                info = Mode_D(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
```

```python
        elif ((Q >= 0) and (li-b < l_top <= li)):
            # print("(l_bot_Q > li - b) and (l_bot_Q < li)")
            if V_ed <= (q * (li - l_top) ** 2) / (2 * hi) - (Q * b) / hi:
                print("Mode A")
                info = Mode_A(Ka, li, l_top)
            else:
                print("Mode E")

                info = Mode_E(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
    else:
        if Q <= 0:
            if V_ed <= (q * li * b) / (2 * hi) - (Q * b) / hi:
                print("Mode F")
                info = Mode_F(q, Q, V_ed, l_top, li, b, hi, Ka, Kv)
            else:
                print("Mode G")
                info = Mode_G(q, Q, V_ed, l_top, li, b, hi, Ka, Kv, Kt)
        else:
            if V_ed <= (q * (li - l_top) * b) / (2 * hi):
                print("Mode H")
                info = Mode_H(l_top, Q)

            else:
                print("Mode I")
                info = Mode_I(q, Q, V_ed, l_top, li, b, hi, Ka, Kv, Kt)

    return info


def Mode_A(Ka, li, l_top):
    l_bot = l_top
    T = 0
    Fv = 0
    U_ri = 0
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_B(q, Q, hi, b, li, Kv, l_top, V_ed):
    l_bot = l_top
    T = 0
    Fv = (V_ed*hi)/li - (q*b)/2
    U_ri = hi/(Kv*b)*((V_ed*hi)/li+(Q*b)/li-(q*b)/2)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_C(q, Q, V_ed, l_top, li, b, hi, Ka, Kv):
    l_bot = - (b * (Ka * li + 2 * Kv * li - Kv * b)) / (2 * (Ka * li - Kv
 * b)) + ((Kv * b) / (Ka * li - Kv * b)) * (((Ka * li + 2 * Kv * li - Kv
 * b) / (2 * Kv)) ** 2 + ((Ka * li - Kv *b) / (Kv * b)) * ((2 * (V_ed *
hi + Q * li)) / q - li * b + l_top * (2 * li - l_top) + (Ka * li * l_top
```

```
 * (l_top + b)) / (Kv * b))) ** (0.5)
    # print("lbot",l_bot,"m")
    T = ((q * Ka) / (2 * Kv * b)) * (l_bot - l_top) * (l_bot + l_top + b)
    # print("T",T,"N")
    Fv = max(T - Q + q * (l_bot - l_top), Q - T)
    # print("Fv",Fv,"N")
    U_ri = (hi / (Kv * b )) * (T - (Q * (li - b)) / li + q * (l_bot -
l_top))
    # print("U",U_ri,"m")
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_D(q, Q, V_ed, l_top, li, b, hi, Ka, Kv):
    l_bot = li
    # print(l_bot)
    T = (V_ed * hi) / li + Q - (q * (li - l_top) ** 2) / (2 * li)
    # print(T)
    Fv =  max(T - Q + q * (li - l_top - b), Q - T)
    # print(Fv)
    U_ri = T * hi * (1 / (Ka * li) + 1 / (Kv * b)) - (Q * hi * (li - b
)) / (Kv * b * li) + (q * hi * (li - l_top - b) * (li - l_top)) / (2 * Kv
 * b * li)
    # print(U_ri)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_E(q, Q, V_ed, l_top, li, b, hi, Ka, Kv):
    l_bot = li
    T = (V_ed * hi) / li + Q - (q * (li - l_top) ** 2) / (2 * li)
    Fv = max(T-Q,Q-T)
    U_ri = T*hi*(1/(Ka*li)+1/(Kv*b))-(Q*hi*(li-b))/(Kv*b*li)
    φ_ri = T / (Ka * li)
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_F(q, Q, V_ed, l_top, li, b, hi, Ka, Kv):
    l_bot = l_top
    T = 0
    Fv = 0
    U_ri = 0
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_G(q, Q, V_ed, l_top, li, b, hi, Ka, Kv, Kt):
    l_bot = l_top
    T = (Kt * b * (V_ed * hi - Q * b - q * li * b / 2)) / (Kt * b ** 2 +
Kv * b * (li - b))
    Fv = (Kv * T) / (Kt)
    U_ri = (Fv * hi) / (Kv * b)
```

```
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_H(l_top, Q):
    l_bot = l_top
    T = Q
    Fv = 0
    U_ri = 0
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]


def Mode_I(q, Q, V_ed, l_top, li, b, hi, Ka, Kv, Kt):
    l_bot = l_top
    T = Q + (Kt * b * (V_ed * hi - q * (li - l_top) * b / 2)) / (Kt * b
 ** 2 + Kv * b * (li - b))
    Fv = (Kv * T) / (Kt)
    U_ri = (Fv * hi) / (Kv * b)
    φ_ri = 0
    return [l_bot, T, Fv, U_ri, φ_ri]



def Annex_y(Kh, Kv, hi, li, lj, m, V_ed, qed, M_ed):
    # INTERPOLATION
    V_ed = M_ed / hi
    N_ed = qed * li
    N_dim = ((N_ed * li) / (2 * M_ed))
    K_CP = ((Kh + (m - 1) * Kv) / (m ** 2)) * li ** 2
    K_SW = (1 / (Kh) + (m - 1) / (Kv)) ** (-1) * li ** 2

    if (Kh / Kv) >= ((1 - N_dim * (3 * m - 2) / (m ** 2)) / (1 - N_dim
 * (m - 2) / (m ** 2))):
        # Coupled mode
        Ur_CP = max(((M_ed / K_CP) - (N_ed * lj) / (2 * K_CP)) * hi, 0)
        return Ur_CP

    elif (Kh / Kv) <= ((1 - N_dim) / (1 + N_dim * (m - 2))):
        # Single wall mode
        Ur_SW = max(((M_ed / K_SW) - (N_ed / (2 * Kt * li))) * hi, 0)
        return Ur_SW

    else:
        # intermediate mode
        V_CP = (b ** 2 * qed * (Kh * (m - 2) + Kv * (2 - 3 * m))) / (2 *
hi * (Kh - Kv))
        Ur_CP = (hi * (2 * hi * V_CP - b ** 2 * m * qed)) / (2 * b ** 2
 * (Kh + Kv * (m - 1)))

        V_SW = (b ** 2 * m ** 2 * qed * (Kh * (2 - m) - Kv)) / (2 * hi
 * (Kh - Kv))
```

```python
        Ur_SW = (hi * (hi * V_SW * (Kh * (2 * m - 2) + 2 * Kv) - Kv * b
 ** 2 * m ** 2 * qed)) / (
                    2 * Kh * Kv * b ** 2 * m ** 2)

        Ur_IN = max((Ur_CP * (V_SW - V_ed) + Ur_SW * (V_ed - V_CP)) / (
V_SW - V_CP), 0)
        return Ur_IN




Kh = 6000000
Kt = 6000000
Kvert = [0.5*Kh,Kh-1,1.5*Kh,2*Kh,2.5*Kh,3*Kh]
#Kvert = [1.5*Kh]

Vertical_load = [1000]

hi = 2.7 #m
b = 1.4 #m
n_con = 18
panels = [3,4,5]

colours = ['red','blue','green', 'purple', 'black','orange']

for y in range(len(panels)):
    fig = plt.figure(figsize=(1920 / 100, 1080 / 100))
    plt.xticks(fontsize=22)
    plt.yticks(fontsize=22)
    for x in range(len(Kvert)):

        def find_displacement(n_story):
            T_above = 0
            T_above1 = []
            Ui = []
            φi = []
            phii = []
            deflectionEachFloor = []
            for i in range(n_story):
                print("floor", n_story - i)
                M_Ned = 0
                print("M_Ned",M_Ned)
                M_Ved = V_ed * hi * i
                print("M_Ved", M_Ved)
                NaNb = find_Na_Nb(M_Ned, li)
                Na = NaNb[0]
                Nb = NaNb[1]
                q = find_q(Na, Nb, li, Nied)
                l_top = find_l_top(M_Ved, Na, li, T_above, q)
                Q = find_Q(M_Ved, Na, li, T_above, q, l_top)
```

```python
        l_bot_Q = find_l_bot_Q(li, l_top, b, Q, q, Ka, Kv, )
        info = find_mode(V_ed, q, Q, li, l_top, hi, b, Ka, Kv)
        T = info[1]
        T_above1.append(info[1])
        print("l_bot", info[0])
        print("T", info[1])
        print("Fv", info[2])
        print("Uri, ", info[3])
        print("Phi", info[4])
        print("T_above",T_above1)

    return T



# Property Arup and Annex Y
m = panels[y]  # number of panels
li = m * b  # m
lj = b # m, Same as b in Arup method.
dens = 420  # kg/m3

# number of storys
n_story = 1


n_con = 18  # pcs, number of connectors

Kv = Kvert[x]


# Stiffness
if Kt == Kv:
    Ka = 1
else:
    Ka = 1 / ((1 / Kt) - (1 / Kv))  # N/m
# print(Ka,"Ka")



qed = Vertical_load[0]  # N/m

nr_graphs = 1
plots = 51
x1 = []
y1 = []

for i in range(nr_graphs):
    x1 = []
    y1 = []
    y2 = []
    y3 = []
    V_ed = 1
```

```
        T_above2 =[]
        Nied = qed * li  # N
        for n in range(plots):
            T_above2.append(find_displacement(n_story))
            print("V_ed", V_ed)
            V_ed = V_ed + 1000
            print("-----")

    Ui = []
    φi = []
    l_top_sap = []
    Q_sap = []
    V_ed_i = 1
    for n in range(len(T_above2)):
        Nied = qed*li
        T_above = T_above2[n]
        print("T_above",T_above)
        M_Ned = 0
        M_Ved = V_ed_i*hi
        M_Ved_y = V_ed_i * hi * 2 + V_ed_i * hi
        V_ed = V_ed_i + M_Ved/hi
        print("M_Ved", M_Ved)
        print("V_ed", V_ed)
        NaNb = find_Na_Nb(M_Ned, li)
        Na = NaNb[0]
        Nb = NaNb[1]
        q = find_q(Na, Nb, li, Nied)
        l_top = find_l_top(M_Ved, Na, li, T_above, q)
        l_top_sap.append(l_top)
        Q = find_Q(M_Ved, Na, li, T_above, q, l_top)
        Q_sap.append(Q)
        l_bot_Q = find_l_bot_Q(li, l_top, b, Q, q, Ka, Kv, )
        info = find_mode(V_ed, q, Q, li, l_top, hi, b, Ka, Kv)

        y1.append(info[3])
        y2.append(Annex_y(Kh, Kv, hi, li, lj, m, V_ed, qed, M_Ved_y))
        x1.append(V_ed_i)
        print("l_bot",info[0])
        print("T", info[1])
        print("Fv",info[2])
        print("Uri, ", info[3])
        print("Phi",info[4])
        print(V_ed_i)
        V_ed_i += 1000
    x1 = [n*10**(-3) for n in x1]
    y1 = [n * 1000 for n in y1]
    y2 = [n * 1000 for n in y2]
    plt.plot(x1, y1, color = colours[x], linestyle = 'dotted', label
 = "Arup, Kv = {} kN/m".format(round(Kvert[x]*10**(-3))))
    #plt.plot(x1, y2, color=colours[x], linestyle='-.', label = "
Annex Y, Kv = {} kN/m".format(round(Kvert[x] * 10 ** (-3))))
```

```
# -*- coding: utf-8 -*-
"""
Created on Wed Mar  1 12:55:14 2023

@author: stian
"""

import os
import sys
import comtypes.client
import matplotlib
import matplotlib.pyplot as plt
import pandas as pd
import math

# 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
24\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()

        ret = SapModel.SetPresentUnits(10)


        h = hi # Height m
        m = panels[y]  # Panels
        L = 3  # Layers
        s = 5  # number of angle brackets

        dens = 420  # kg/m3

        Kv = Kvert[x]/n_con
        Kab = 10 ** 15  # N/m. Angle Brackets
```

```
        Fh = []  # N
        steps = 51
        load = 1  # N
        for i in range(steps):
            Fh.append(load)
            load = load + 2000



        Fv = Vertical_load  # N/m


        def E(L):
            E_0 = 11000 * 10 ** 20  # N/m2
            E_90 = 370 * 10 ** 20  # N/m2
            G = 690 * 10 ** 20  # N/m2
            E1 = 0
            E2 = 0
            E3 = 0

            for n in range(L):
                if n % 2 == 0:
                    E1 = E1 + E_0
                    E2 = E2 + E_90
                elif n % 2 != 0:
                    E1 = E1 + E_90
                    E2 = E2 + E_0
            E1 = E1 / L
            E2 = E2 / L
            E3 = E_90
            G12 = G * 0.75
            G23 = G12 / 10
            G13 = G12 / 10
            return E1, E2, E3, G12, G23, G13, E_0


        # define material property
        MATERIAL_CLT = 3  # Change to CLT

        ret = SapModel.PropMaterial.SetMaterial('CLT', MATERIAL_CLT)

        # assign isotropic mechanical properties to material
        ret = SapModel.PropMaterial.SetWeightAndMass('CLT', 2, dens / 9.
81)
        ret = SapModel.PropMaterial.SetMPOrthotropic('CLT', [E(L)[0], E(L
)[1], E(L)[2]], [0.3, 0.3, 0.3],
                                                     [E(L)[3], E(L)[4], E
(L)[5]])

        ret = SapModel.PropArea.SetShell_1('Wall A', 5, True, 'CLT', 0, 0
.1, 0.1, -1)
```

```
        ret = SapModel.PropMaterial.SetMaterial('L', 3, -1)
        # ret = SapModel.PropMaterial.SetWeightAndMass('L',2,())
        ret = SapModel.PropFrame.SetRectangle('Load', 'L', 0.05, L)

        # Walls
        # Creating the wall
        for i in range(m):
            if i == 0:
                ret = SapModel.AreaObj.AddByCoord(4, [b * i, b * i + b -
0.01, b * i + b - 0.01, b * i], [0, 0, 0, 0],
                                                  [0, 0, h, h ], "Panel_"
 + str(i + 1), 'Wall A', "Panel_" + str(i + 1),
                                                  "Global")
                ret = SapModel.FrameObj.AddByCoord(i * b, 0, h, i * b + b
 - 0.01, 0, h, 'FF' + str(i), 'Load', 'FF' + str(i))
                #ret = SapModel.FrameObj.AddByCoord(i * b, 0, 0, i * b +
b - 0.01, 0, 0, 'FB' + str(i), 'Load', 'FB' + str(i))
            elif i == m - 1:
                ret = SapModel.AreaObj.AddByCoord(4, [b * i + 0.010, b *
i + b, b * i + b, b * i + 0.010], [0, 0, 0, 0],
                                                  [0, 0, h, h ], "Panel_"
 + str(i + 1), 'Wall A', "Panel_" + str(i + 1),
                                                  "Global")
                ret = SapModel.FrameObj.AddByCoord(i * b + 0.01, 0, h, i
 * b + b, 0, h, 'FF' + str(i), 'Load', 'FF' + str(i))
                #ret = SapModel.FrameObj.AddByCoord(i * b + 0.01, 0, 0, i
 * b + b, 0, 0, 'FB' + str(i), 'Load', 'FB' + str(i))
            else:
                ret = SapModel.AreaObj.AddByCoord(4, [b * i + 0.01, b * i
 + b - 0.010, b * i + b - 0.010, b * i + 0.010],
                                                  [0, 0, 0, 0], [0, 0, h
, h ], "Panel_" + str(i + 1), 'Wall A',
                                                  "Panel_" + str(i + 1),
"Global")
                ret = SapModel.FrameObj.AddByCoord(i * b + 0.01, 0, h, i
 * b + b - 0.01, 0, h, 'FF' + str(i), 'Load','FF' + str(i))
                #ret = SapModel.FrameObj.AddByCoord(i * b + 0.01, 0, 0, i
 * b + b - 0.01, 0, 0, 'FB' + str(i), 'Load
',                                  'FB' + str(i))

        # links
        ret = SapModel.Proplink.SetHook('Hold Down', [True, False, False
, False, False, False],
                                        [False, False, False, False,
False, False], [True, False, False, False, False, False],
                                        [Kh, 0, 0, 0, 0, 0], [0, 0, 0, 0
, 0, 0], [Kh, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], 0, 0)
        ret = SapModel.Proplink.SetLinear('Shear', [False, True, False,
False, False, False],
                                          [False, False, False, False,
False, False], [0, Kv, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0],
                                          0.02, 0.02, False, False)
```

```
        ret = SapModel.PropLink.SetGap('Compression', [True, False, False
, False, False, False],
                                       [False, False, False, False, False
, False], [True, False, False, False, False, False],
                                       [10000000000000, 0, 0, 0, 0, 0], [
0, 0, 0, 0, 0, 0], [10000000000000, 0, 0, 0, 0, 0],
                                       [0, 0, 0, 0, 0, 0], 0.01, 0.01)
        ret = SapModel.Proplink.SetLinear('AngleBracket', [False, True,
False, False, False, False],
                                          [False, False, False, False,
False, False], [0, Kab, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0],
                                          0.01, 0.01, False, False)

        # Shear connection
        for i in range(m):
            for n in range(n_con + 2):
                if i != 0 and n != 0 and n != n_con + 1:
                    ret = SapModel.LinkObj.AddByCoord(i * b - 0.010, 0, n
 * h / (n_con + 1), i * b + 0.010, 0,
                                                      n * h / (n_con + 1
), 'SHEAR CON_' + str(i) + '_' + str(n), False, 'Shear',
                                                      'SHEAR CON_' + str(
i) + '_' + str(n))
                    ret = SapModel.PointObj.AddCartesian(i * b - 0.01, 0
, h, 'PL_m' + str(i) + '_' + 'l ',
                                                         'Pl_ m' + str(i
) + '_' + 'l')
                    ret = SapModel.PointObj.AddCartesian(i * b + 0.01, 0
, h, 'PL_m' + str(i) + '_' + 'r',
                                                         'Pl_ m' + str(i
) + '_' + 'r')
                    ret = SapModel.ConstraintDef.SetEqual('Cons_m' + str(
i), [True, False, False, False, False, False])
                    ret = SapModel.PointObj.SetConstraint('PL_m' + str(i
) + '_' + 'l', 'Cons_m' + str(i))
                    ret = SapModel.PointObj.SetConstraint('PL_m' + str(i
) + '_' + 'r', 'Cons_m' + str(i))
                elif i != 0:
                    ret = SapModel.PointObj.AddCartesian(i * b - 0.01, 0
, h, 'PL_m' + str(i) + '_' + 'l',
                                                         'PL_m' + str(i
) + '_' + 'l')
                    ret = SapModel.PointObj.AddCartesian(i * b + 0.01, 0
, h, 'PL_m' + str(i) + '_' + 'r',
                                                         'PL_m' + str(i
) + '_' + 'r')
                    ret = SapModel.ConstraintDef.SetEqual('Cons_m' + str(
i), [True, False, False, False, False, False])
                    ret = SapModel.PointObj.SetConstraint('PL_m' + str(i
) + '_' + 'r', 'Cons_m' + str(i))
                    ret = SapModel.PointObj.SetConstraint('PL_m' + str(i
) + '_' + 'l', 'Cons_m' + str(i))
```

```
        # Hold down connection and compression link
        for i in range(m):
            if i == 0:
                ret = SapModel.LinkObj.AddByCoord(0, 0, 0, 0, 0, -0.01, '
HD_' + str(i), False, 'Hold down', 'HD_' + str(i))
                ret = SapModel.PointObj.AddCartesian(0, 0, -0.01, 'PC_SL'
 + str(i), 'PC_SL' + str(i))
                ret = SapModel.LinkObj.AddByCoord(0, 0, 0, 0, 0, -0.01, '
LC_SL_' + str(i), False, 'Compression',
                                                  'LC_SL_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL' + str(i), [
True, True, True, True, True, True])
                ret = SapModel.LinkObj.AddByCoord(b - 0.01, 0, 0, b - 0.
01, 0, -0.01, 'LC_ST_' + str(i), False, 'Compression',
                                                  'LC_ST_' + str(i))
                ret = SapModel.PointObj.AddCartesian(b - 0.01, 0, -0.01,
'PC_ST_' + str(i), 'PC_ST_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_' + str(i), [
True, True, True, True, True, True])


            elif i == m - 1:
                ret = SapModel.LinkObj.AddByCoord((i + 1) * b, 0, 0, (i
 + 1) * b, 0, -0.01, 'HD_' + str(i), False, 'Hold down',
                                                  'HD_' + str(i))
                ret = SapModel.LinkObj.AddByCoord(i * b + 0.01, 0, 0, i
 * b + 0.01, 0, -0.01, 'LC_SL_' + str(i), False,
                                                  'Compression', 'LC_SL_'
 + str(i))
                ret = SapModel.PointObj.AddCartesian(i * b + 0.01, 0, -0.
01, 'PC_SL_' + str(i), 'PC_SL_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL_' + str(i), [
True, True, True, True, True, True])
                ret = SapModel.LinkObj.AddByCoord((i + 1) * b, 0, 0, (i
 + 1) * b, 0, -0.01, 'LC_ST_' + str(i), False,
                                                  'Compression', 'LC_ST_'
 + str(i))
                ret = SapModel.PointObj.AddCartesian((i + 1) * b, 0, -0.
01, 'PC_ST_' + str(i), 'PC_ST_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_' + str(i), [
True, True, True, True, True, True])
            else:
                ret = SapModel.LinkObj.AddByCoord(i * b + 0.01, 0, 0, i
 * b + 0.01, 0, -0.01, 'LC_SL_' + str(i), False,
                                                  'Compression', 'LC_SL_'
 + str(i))
                ret = SapModel.PointObj.AddCartesian(i * b + 0.01, 0, -0.
01, 'PC_SL_' + str(i), 'PC_SL_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_SL_' + str(i), [
True, True, True, True, True, True])
                ret = SapModel.LinkObj.AddByCoord((i + 1) * b - 0.01, 0,
```

```
0, (i + 1) * b - 0.01, 0, -0.01, 'LC_ST_' + str(i),
                                                  False, 'Compression', '
LC_ST_' + str(i))
                ret = SapModel.PointObj.AddCartesian((i + 1) * b - 0.01,
0, -0.01, 'PC_ST_' + str(i), 'PC_ST_' + str(i))
                ret = SapModel.PointObj.SetRestraint('PC_ST_' + str(i), [
True, True, True, True, True, True])

        a = (b * m) / (s + 1)
        for i in range(s):
            ret = SapModel.LinkObj.AddByCoord(a * (i + 1), 0, 0, a * (i
 + 1), 0, -0.01, 'LAB_S' + str(i), False, 'AngleBracket',
                                                     'LAB_S' + str(i))
            ret = SapModel.PointObj.AddCartesian(a * (i + 1), 0, -0.01, '
PAB_S' + str(i), 'PAB_S' + str(i))
            ret = SapModel.PointObj.SetRestraint('PAB_S' + str(i), [True
, True, True, True, True, True])
        ret = SapModel.View.RefreshView(0, False)

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

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

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

        ret = SapModel.AreaObj.SetAutoMesh("GroupCLT", 2, MaxSize1=5,
MaxSize2=5, ItemType=1)

        # Horizontal load
        ret = SapModel.PointObj.AddCartesian(0, 0, h, 'PF', 'PF')
        for i in range(len(Fh)):
            ret = SapModel.LoadPatterns.Add('Fh' + str(i + 1), 3, 0,
False)
            ret = SapModel.PointObj.SetLoadForce('PF', 'Fh' + str(i + 1
), [Fh[i], 0, 0, 0, 0, 0])
```

```
        ret = SapModel.PointObj.AddCartesian(m * b / 2, 0, h, 'PMO', 'PMO
')

        # Creating load pattern for vetrical load
        #for i in range(len(Fv)):
        #    ret = SapModel.LoadPatterns.Add("Fv" + str(i + 1), 3, 0,
False)
            #ret = SapModel.PointObj.SetLoadForce('PMO', "Fv" + str(i +1
), [0,0,T_above2[i]-Fv[0]*m*b/2,0,0,0])
            #UDL
        for i in range(len(Fh)):
            ret = SapModel.LoadPatterns.Add("Fv" + str(i + 1), 3, 0,
False)
            for n in range(m):
                if n == 0:
                    if l_top_sap[i] <= b-0.01:
                        ret = SapModel.FrameObj.SetLoadDistributed('FF'
 + str(n), "Fv" + str(i + 1), 1, 10, l_top_sap[i], b-0.01, Fv[0], Fv[0],"
Global", False, False)


                elif n == m-1:
                    if l_top_sap[i]>=n*b+0.01:
                        ret = SapModel.FrameObj.SetLoadDistributed('FF'
 + str(n), "Fv" + str(i + 1), 1, 10, l_top_sap[i] - (2*b + 0.01),b, Fv[0
], Fv[0], "Global", False, False)
                    else:
                    #elif l_top_sap[i]<n*b+0.01:
                        ret = SapModel.FrameObj.SetLoadDistributed('FF'
 + str(n), "Fv" + str(i + 1), 1, 10, 0,1, Fv[0], Fv[0], "Global", True,
False)


                else:
                    if b + 0.01 < l_top_sap[i] <= n * b + b - 0.01:
                        ret = SapModel.FrameObj.SetLoadDistributed('FF'
 + str(n), "Fv" + str(i + 1), 1, 10,

l_top_sap[i] - (n * b + 0.01), b - 0.01, Fv[0],
                                                                   Fv[0
], "Global", False, False)
                    elif l_top_sap[i] <= n * b + 0.01:
                        ret = SapModel.FrameObj.SetLoadDistributed('FF'
 + str(n), "Fv" + str(i + 1), 1, 10, 0, 1, Fv[0],
                                                                   Fv[0
], "Global", True, False)

        for i in range(len(T_above2)):
            ret = SapModel.LoadPatterns.Add('Q'+str(i+1),3,0,False)
            ret = SapModel.PointObj.SetLoadForce('PF','Q'+str(i+1),[0,0,
Q_sap[i],0,0,0])
```

```
        # Load combos
        for i in range(len(Fh)):
            for n in range(len(Fv)):
                ret = SapModel.LoadCases.StaticNonlinear.SetCase('H' +
str(Fh[i]) + 'V' + str(Fv[n]) + 'Q' + str(round(Q_sap[i])))
                ret = SapModel.LoadCases.StaticNonlinear.SetLoads('H' +
str(Fh[i]) + 'V' + str(Fv[n]) + 'Q' + str(round(Q_sap[i])), 3, ["Load", "
Load", "load"], ['Fh' + str(i + 1), 'Fv' + str(i + 1), 'Q'+str(i+1)], [ 1
, 1,1])

        # Setting only the loadcases we want to run, to 'run'
        ret = SapModel.File.Save(ModelPath)
        ret = SapModel.Analyze.SetActiveDOF([True, False, True, False,
True, False])
        ret = SapModel.Analyze.SetRunCaseFlag('', False, True)
        Combinations = []
        for i in range(len(Fh)):
            for n in range(len(Fv)):
                ret = SapModel.Analyze.SetRunCaseFlag('H' + str(Fh[i]) +
'V' + str(Fv[n]) +'Q' + str(round(Q_sap[i])), True, False)
                Combinations.append('H' + str(Fh[i]) + 'V' + str(Fv[n
]) + 'Q' + str(round(Q_sap[i])))

        Displa = []
        Contributions = []
        Analytical = []
        ret = SapModel.PointObj.AddCartesian(m * b, 0, h, 'PA', 'PA')

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

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

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

```
            ObjectElm = 0

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

                                                   ObjectElm,

                                                   NumberResults,

                                                   Obj,

                                                   Elm,

                                                   ACase,

                                                   StepType,

                                                   StepNum,

                                                   U1,

                                                   U2,

                                                   U3,

                                                   R1,

                                                   R2,

                                                   R3)

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


        ret = mySapObject.ApplicationExit(False)
        SapModel = None

        mySapObject = None

        Fh = [n*10**(-3) for n in Fh]
        Dis = [n * 1000 for n in Dis]
        plt.plot(x1, Dis,color = colours[x], label="SAP2000, Kv = {} kN/m
".format(round(Kvert[x]*10**(-3))))

    plt.xlim(0)
    plt.ylim(0)
    plt.xlabel('Horizontal load [kN]',fontsize = 22)
    plt.ylabel('Displacement [mm]',fontsize = 22)
    plt.title("q = {} kN/m, Kh = {} kN/m, {} panels".format(qed*10**(-3),
```

```
round(Kh*10**(-3)), m), fontsize = 22)
    plt.legend(fontsize = 14)
    #plt.savefig("c:/result/"+ str(m) + " panels")
    plt.savefig("//Client/C$/result/ny0606_"+ str(m)+ "panels_qed"+str(Fv
[0]))
    plt.clf()
```

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

Postboks 5003
NO-1432 Ås
Norway