Urban quay walls

A numerical study to recognize foundation defects via masonry damage patterns

M. Grund

TUDelft

SWECO

Urban quay walls: A numerical study to recognize foundation defects via masonry damage patterns

by

Mitchel Grund
4219678

in fulfillment of the requirements for the degree of

Master of Science

at the Delft University of Technology,
to be defended on 24 April 2020 at 11:00.

Thesis committee:

Prof. dr. ir. J.G. Rots	TU Delft, Faculty of Civil Engineering and Geosciences
Dr. R. Esposito	TU Delft, Faculty of Civil Engineering and Geosciences
Dr. ir. J.G. de Gijt	TU Delft, Faculty of Civil Engineering and Geosciences
Ir. R. Roggeveld	Sweco

Faculty of Civil Engineering and Geosciences
Delft University of Technology

TU Delft

SWECO

Summary

The city of Amsterdam has a large number of old quay walls with rotten foundation piles. These foundation piles need to be identified and measures need to be taken. The urban quay walls are supported by two types of foundation: pinewood piles, which are easily affected by bacterial decay and spruce piles. To understand the mechanical behaviour of quay walls better, it is needed to know the type of wood used for each pile foundation along the 200 km of quay walls currently showing signs of damage. For that reason, specialized diving teams are hired to identify the rotten piles and foundation defects, to know which foundation piles need replacement.

Since the area is very large and diving inspections are costly and lengthy in time, there is a need to correlate the foundation defect to the masonry damage above the water level. The masonry above the water level could give lots of information about the condition of the foundation, due to cracks or deformations in the masonry. This research could help to relate foundation defects with damage patterns in the masonry. Understanding this relation helps identifying foundation defects at an earlier stage and helps the municipality to prioritize the replacement of foundation piles.

The thesis aims to find indicators above the water line to identify foundation problems by studying the crack patterns in a typical unreinforced masonry quay in Amsterdam. From the point of view of the masonry structures, failure of foundation piles results in a settlement deformation causing cracking. This research will support the current work by Sweco in helping to find foundation defects from above the waterline via masonry damage patterns in quay walls.

This will be achieved by performing a parametric study, bases on 2D nonlinear finite element analyses, varying the extent of the pile defects, the material properties of the masonry and lateral boundary conditions for a selected representative base case. To simulate the damage in masonry, a smeared crack approach was used. The foundation defects were simulated by applying a settlement deformation to the quay wall. A Gaussian settlement deformation profile was imposed and the ratio between the length of the profile and the length of the quay wall was varied to simulate the failure of single or multiple piles. To capture the influence of the material properties of masonry (especially related to tension failure), three types of masonry were defined: weak, average and strong. The influence of the boundary conditions at the edges was checked by performing analyses with horizontally free lateral sides and with horizontally fixed lateral side. This is done to simulate the effect of arching in the structure. Eventually, the influence of the location of the foundation defect was analyzed by comparing a symmetric Gaussian settlement deformation with an asymmetric settlement.

The analyses show correlations between the vertical displacement at the top of the structure and the length of the settlement profile. As expected, this can be interpreted with the fact that if several piles are damaged simultaneously, a larger portion of the quay wall is cracking. The material properties of the masonry influence the development of crack patterns. The stronger is the masonry, meaning increasing the values of Young's modulus, tensile strength and fracture energy, the larger is the settlement displacement needed to obtain the same crack pattern. The lateral constraints contribute mostly to the development of the horizontal crack since no horizontal cracks appeared in situations without these constraints. Since the influence of additional loads is not considered in the analysis and the model is modelled in 2D it is recommended to analyze the influence of both in further studies. It is also recommended to validate the model against field measurements since no verification has been done.

"No matter what you've done for yourself or for humanity, if you can't look back on having given love and attention to your own family, what have you really accomplished?"

LEE IACOCCA

Preface

This report is written as the final work for my master program Structural Engineering at the Delft University of Technology. In this program, I chose the specialization Hydraulic Structures since I have always been very interested in these types of structures. When I got the chance to do my research about the urban quay walls in Amsterdam, I was very excited. Mostly, because this subject contained a real-life problem in which I could contribute. I learned a lot about finite element models, quay wall structures and "doing research". However, I think the most important thing that I learned is to set priorities: sometimes important things are not that important at all.

This thesis was conducted at Sweco, in agreement with Ingenieurs Bureau Amsterdam. I want to thank Ingenieurs Bureau Amsterdam and Sweco for their help with this thesis and especially Arjan Frens and all my colleagues for the pleasant working environment at the office. A special thanks to my thesis committee, Rita Esposito, Jarit de Gijt, Richard Roggeveld and Jan Rots, for their knowledge and guidance through this process.

I was lucky to be surrounded by friends who supported me during the good and the tough times but also to make my student days memorable. Last but not least, I want to thank my father Arno, my sister Daisy and the rest of the family, for all their support and believe in me.

A special notice to my mum: without your love and support it would be impossible for me to be where I am right now. I love you.

Mitchel Grund
Delft, March 30, 2020

Contents

1

Introduction

1.1 Background

In the cities of the Netherlands, a large number of quay walls are built alongside the rivers and canals. Most of the quay walls were built more than 100 years ago and are therefore very old. The quay walls of Amsterdam are quite famous in the whole world and became part of the image of the city.

A lot of the urban quay walls in Amsterdam are in poor condition. The engineering department of the municipality of Amsterdam (IBA) has to manage around 200 kilometres of quay wall structures, which suffer from overdue maintenance. Basic cuts have been made since the 1980s because the focus of investments was more on the quality of life and social affairs than in maintenance (Dijksma, 2019). Since that time, the urban quay walls of Amsterdam have been inspected superficial and only above the water level. This means that the foundation of the quay walls has not been checked for almost forty years, while pile rot is one of the significant problems for foundations in Amsterdam (Boshoff, 1990). This is also proven by the professional diving team (Baars-Cipro), who inspect the foundation of the quay walls in Amsterdam (Baars-CIPRO, 2019).

(a)   (b)

Figure 1.1: The difference in how quay walls were used 100 years ago (a) and nowadays (b).

Furthermore, the old urban quay walls in Amsterdam were not designed for their present-day use. In the early days only small and light vehicles, sometimes with horses, were used in Amsterdam. These smaller vehicles were accounted for in the design of the quay walls, while nowadays more vehicles drive through the city centre, which has also increased significantly in size. This results in a significant increase in loads. Additionally, quay walls are often used as a storage location, e.g. heavy cranes are placed on top of them, which increases the vertical load acting on the structures. Figure 1.1 shows the difference between the use of the quay walls over the years.
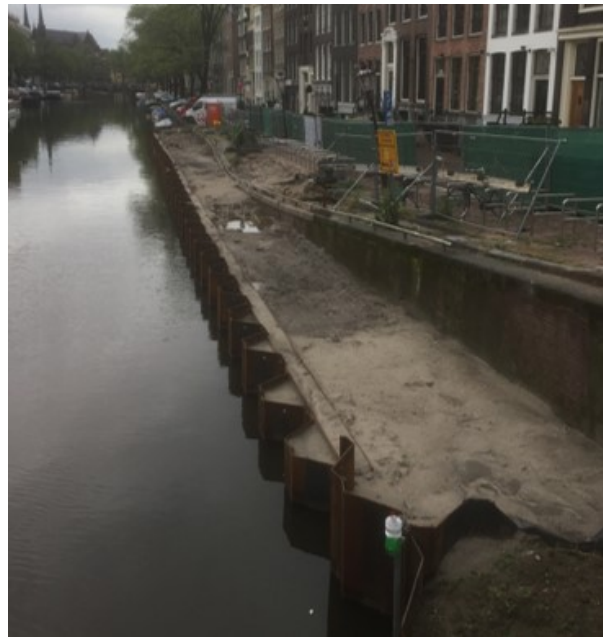

Figure 1.2: A temporary solution: a sheet pile construction strengthens the quay wall.

Due to recent collapses and near-collapses of quay walls, it has become clear that the current maintenance regime of the municipality of Amsterdam does not result in sufficiently accurate insight into the structural condition of the quay walls. A new asset management approach with an emphasis on safety is being developed. This program consists of three phases:

- Researching

- Enhancing, decreasing loads and monitoring

- Replacing

In the first phase (research), IBA investigates the condition of the (pile)foundation, capping beams and all the connections beneath the water level to identify the quay walls with inadequate bearing capacity. Boshoff (1990) stated that buildings and quay walls in Amsterdam suffer from failing pile foundations. One reason is the difference in sapwood thickness of spruce pines and pinewood. Both materials are used as foundations for buildings and quay walls in Amsterdam. Bacteriological attacks have affected the sapwood of the wooden piles over the years, which leads to a decrease in the effective cross-section. In the case of pinewood, the sapwood is relatively big (4-5 cm), while the sapwood area for spruce pines are relatively small (1 cm) (Klaassen, 2008). The chances that pinewood piles fail due to cross-section reduction are relatively big.

Some quay walls are very close to collapse, so emergency measures must be taken. This process is defined in phase two of the program. The weak quay walls are usually supported by placing sheet pile walls and fill the gap in between with sand (Figure 1.2). In this way, the quay wall structures are strengthened and prevented from collapsing. However, this is just a temporary solution, since this affects the image of the city. Therefore, in the third stage, the structures are renovated or replaced. However, only two kilometres of quay walls are replaced each year, which means it takes 100 years to replace all the existing quay wall structures. IBA strives to increase this to ten kilometres each year, whereafter they want to maintain it to five kilometres each year.

1.2 Research scope

The current inspection and maintenance regime does not provide a systemic approach to identify quay walls at risk. This research aims to find masonry damage patterns which can be utilized as indicators or telltales to predict foundation defects accurately. Such insights can contribute to accelerating the condition assessment program and can help to allocate the inspection budgets effectively, primarily by limiting the number of (expensive) diving inspections. According to Boshoff (1990), foundation problems can be revealed by deformations and cracks in masonry of buildings. This leads to the main research question:

"***How to recognize foundation defects via masonry damage patterns in quay walls?***"

To answer the main research question, other questions are formulated:

- Which are the common foundation defects encountered in the typical Amsterdam quay wall?
- Which boundary conditions apply to a typical Amsterdam quay wall, and what is their influence?
- What is the influence of the material properties of masonry on the development of damage patterns?
- How do foundation defects alter the static system and structural response of the quay wall?
- Which damage patterns may indicate foundation defects?

My research supports the current work by IBA in assisting to find foundation defects from above the waterline. The main problem is that IBA does not know the condition of the foundation of the quay wall structures since this is not visible from above the waterline. Therefore, inspections done by divers are deployed to investigate and assess the foundations of the quay walls. This is an expensive and time-consuming method which has to be done faster and cheaper.

This research could help in the first step to identify foundation defects by looking at the masonry. The masonry could give much information about the foundation due to cracks or deformations in the masonry. Knowing the relation between those can help to identify foundation defects at an earlier stage and help the municipality to map the quay walls which are in poor condition.

1.3 Research method

The first step is to get information about quay walls, masonry, and how to perform non-linear finite element analyses from literature. After that, non-linear finite element analyses of various cases are done in terms of material properties, pile defects and boundary conditions. A base case is defined, which is the starting point of the research. Then, to capture the influence of the stiffness and strength of the masonry, three types of masonry are formulated: weak, average and strong. Analyses are performed with the three types of masonry, combined with various lengths of foundation defects. The foundation defects are described as the length of the defect, divided by the length of the quay wall section. The influence of the boundary conditions at the edges is checked by performing analyses with free edges and with horizontal fixed edges.

1.4 Thesis outline

This section describes the structure of the report. In the first chapters, the literature used in this report is elaborated and summarized. Starting in Chapter 2, which describes the typical Amsterdam quay wall. In this chapter also the loads acting on the quay wall structures, boundary conditions and the pile foundations itself are discussed. Chapter 3 describes the theory about masonry and its mechanical behaviour. This chapter also describes two material models: the Total Strain Cracking model and the Engineering Masonry model.

The model is presented in Chapter 4. The model is described with the associated input parameters and boundary and loading conditions. Also, a sensitivity analysis is performed. Chapter 5 displays the results from the model and several variations. The influence of the various parameters, such as the material parameters and the foundation defects are also presented. Chapter 6 reviews all the results and the influence of the parameters are discussed. In Chapter 7 the conclusions and recommendations are presented.

2

Quay walls in Amsterdam

This chapter describes the geometry and loading conditions of the typical Amsterdam quay wall, together with the currently identified foundation problems. Amsterdam contains multiple types of quay walls, which can be distinguished in four primary forms (De Gijt and Roubos, 2013):

- gravity wall on spread foundations,

- gravity wall on pile foundations,

- L-wall on pile foundations,

- sheet pile wall from steel or concrete.

Currently, the gravity wall on pile foundations is expiring the most extensive damage (Baars-CIPRO, 2019). The quay wall consists of a masonry gravity wall with natural stones on top. The quay wall is based on a wooden pile foundation, made of spruce or pinewood. The capping beams connect the four piles in the cross-section and support the wooden floor on top of it.
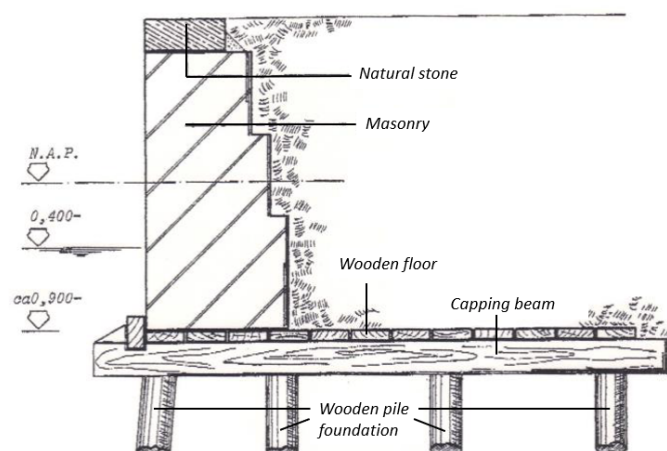
Figure 2.1: A gravity wall on a wooden pile foundation (IBA).

Information about the properties of the pile foundation is significantly relevant since spruce and pinewood react differently to bacterial decay. This aspect is described in Section 2.1. However, information on the type of wood used at which place for the foundation is currently unavailable.

As mentioned in Chapter 1, loads have increased substantially since 1900. Therefore an overview of all the loads acting on the quay wall structures is given in Section 2.2. Although the loads have been significantly increased and the quay walls are not designed to withstand these loads, most of the quay walls were still observed to maintain stable after exceeding the design load. There seems to be a hidden resistance in the quay walls, according to De Gijt and Roubos (2013):

> "It is notable that, according to the assessment of certain quay walls, some collapsing should have already occurred whereas, in practice, the quay walls are still functioning despite dealing with far greater loads compared to the loads these walls were originally designed for."

This hidden strength can be composed by the effect of arching in the quay wall. Therefore it is essential to look at the boundary conditions of the quay wall sections. It could be possible that the boundary conditions are such, that the quay walls encounter horizontal confinement, which gives the structures extra strength. This phenomenon is explored in Section 2.3. Lastly, it is necessary to describe the failure mechanisms of gravity walls with a pile foundation. The most relevant ones are reported and discussed in Section 2.4.

2.1 Degradation of pile foundation

In this section, the biological degradation of the wooden pile foundation is described. A distinction is made between continuous decay and non-continuous decay.

Continuous decay

A wooden pile foundation is exposed to continuous decay, which is mainly caused by bacteria. This type of decay occurs continuously over time and the entire length of the pile. These bacteria become active when water penetrates within the pile. Differences in the water flow are caused by pressure differences, but also by the permeability of the wood (De Gijt and Roubos, 2013). Klaassen (2008) found out that the permeability of the wood plays a vital role in the water flow in a pile. The higher the permeability of the wooden pile, the higher the flow velocities in a pile are, which results in faster decay by erosion bacteria.

Differences in permeability can be found between the two most used types of wood for foundations: spruce and pinewood. Both materials were common to use as a wooden foundation. Pinewood has an open wood structure, which shows a higher water flow than spruce. Therefore, pinewood is more vulnerable to decay by erosion bacteria than spruce woodpiles. Klaassen (2008) found out that water flow through pinewood stems can be up to four times higher than in spruce stems. On top of that, flow rates through spruce are less variable than through pinewood, which might contribute to the generally low water flow in spruce.

Another aspect to take into account is the stem structure of the pile. A foundation pile consists of sapwood, the external portion of the stem and heartwood, the inner portion (Figure 2.2). Pine sapwood is vulnerable to bacterial decay, while in pine heartwood the flow velocity is almost zero. For spruce, it is believed that there is a difference between the sap- and heartwood. Nevertheless, after several decades in function, the boundary between sapwood and heartwood cannot be distinguished easily anymore. This results in the decay rate, which is found in spruce, is a combination of the velocity in sapwood and heartwood (Klaassen and van Overeem, 2012).
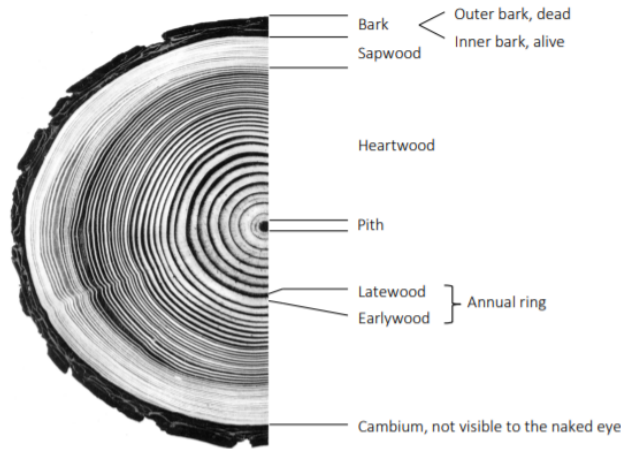
Figure 2.2: Cross-section through a stem (Blaß and Sandhaas, 2017).

Lastly, the origin of the foundation piles, the differences in harvesting and the tree growth dynamics affect the quality of the wood. These factors can lead to differences in material properties and the amount of sapwood in a tree. Consequently, it is not possible to assume that every foundation pile has the same material properties.

Non-continuous decay

Non-continuous decay occurs when foundation wood comes into contact with air as a result of a dry period (De Gijt and Roubos, 2013). Due to the sufficient oxygen in the air and insufficient oxygen in the water, the wood-degrading fungi become active when it gets in contact with air. The moisture level of the wood determines the type of fungi.

Among others, soft rot fungi are capable of breaking down the wood with even a limited amount of oxygen. The speed of decay depends upon the available oxygen and type of wood (De Gijt and Roubos, 2013). When the dry period is more extended, white rot and brown rot can occur, which are highly efficient wood-decaying fungi. The speed of the decay depends on the moisture level of wood and the species and type of wood. Wood degraded by these fungi has lost its texture.

| (a) | (b) |

Figure 2.3: An increment borer (a) is used to take a wood sample (b) from the pile (De Gijt and Roubos, 2013), (Baars-CIPRO, 2019).

When the decay of the pile foundation is less advanced, it can be challenging to recognize the fungi visually. The pile can be dehydrated via the top layer around the heartwood, which means the decay is

not visual on the outside. A drill sample assessment (Figure 2.3) into the heart of the pile can provide sufficient information about the presence of rot decay. The drill sample can also see the presence of blue hyphae. Blue fungi do not deteriorate wood but give right information about how far the wooden pile head has dried in the past (De Gijt and Roubos, 2013).

Inspections to determine degradation in pile foundation

Inspections are required to get adequate insight into the condition of the quay wall structures. In the case of the brickwork, it is relatively easy to execute such an inspection. The more significant part of the masonry lies above the water level and is therefore relatively easy to access. Nevertheless, there are some pitfalls. In the past it was a standard routine to renew the brickwork from the waterline, meaning that this visible part is relatively new compared with the brickwork beneath the waterline. The brickwork beneath the waterline may be very damaged, while above water everything seems fine. It is more complicated to get insight into the conditions of the structure beneath the water level. A specialized diving inspection team is needed to assess the foundation accurately. The initial inspection costs for such an inspection are relatively high.
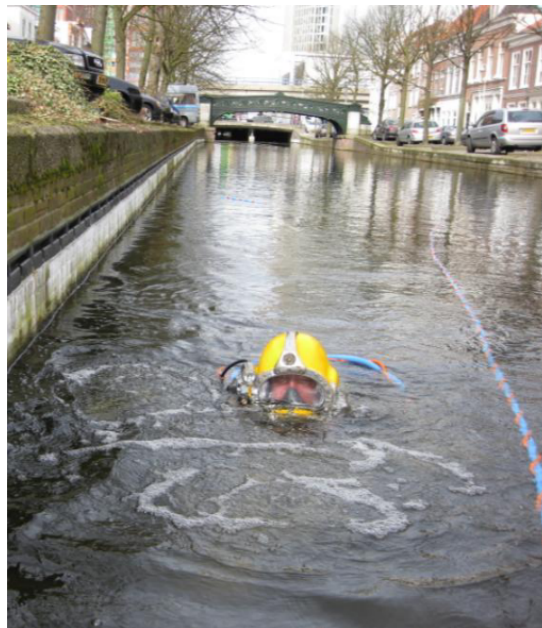

Figure 2.4: A diver inspecting a typical urban quay wall (De Gijt and Roubos, 2013).

According to the inspection guidelines of SBRCURnet (2014), several measurement methodologies are used to determine the condition of foundation piles, such as: consulting the archives for relevant information, visual inspections, measurements, and environmental factors. These kind of inspections are more superficial and do not give more detailed information. Nevertheless, it is more complicated to obtain this kind of information because a specialized diving inspection team is needed. They perform a technical inspection, resulting in a rough indication of the condition of the piles.

The soft skin is the deteriorated part of the foundation pile and does not contribute to the strength of the pile. Knowing the thickness of the soft skin gives a good insight into the status of the pile. To know this thickness, the divers take penetration measurements with the Specht (Figure 2.5). The Specht is an instrument which can penetrate the wooden pile underwater and give an estimation of the degree of

Figure 2.5: Penetration meter: Specht De Gijt and Roubos (2013).

decay of the pile. This is done by taking measurements from four locations at one single pile (left, right, front and back of the pile).
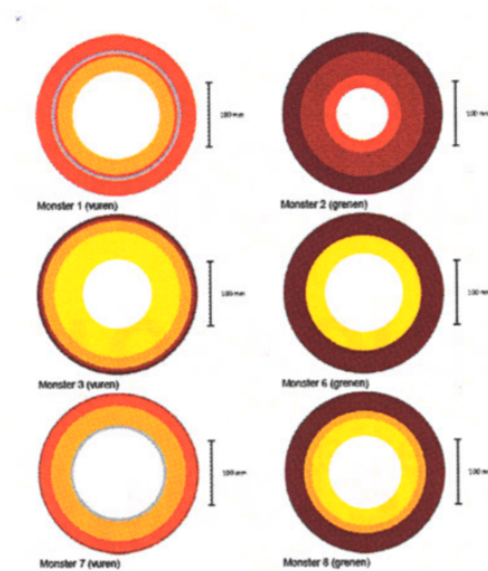

Figure 2.6: laboratory results: left: spruce, right: pinewood (IBA).

To get the core samples, the divers use an increment borer ⌀ 10/12 mm. It is essential to prevent the sample from being pulverized or breaking into many pieces, so the borer must be sharp enough to penetrate the pile. When the samples are taken, they must be stored in a tube submerged in water from the environment to maintain their properties. It is also advised to take two wood samples every 15 running metres to get sufficient information. The objective of examining core samples is to track the cause and the progress of the decay of a woodpile. Additionally, the thickness of the sapwood and the type of degradation are determined. With this information, an estimation of the development of the degradation and the loss of strength for the coming 25-50 years can be established (Figure 2.6). Knowing the intrusion values from the penetration measurements and the results of the examined core samples, the average thickness of the soft skin is determined. Combined with the diameter of the pile, the residual bearing surface can be determined.

2.2   Loads

This section describes the various kinds of loads which can act on a quay wall structure. During the design of the quay walls, they did not consider some loads since these loads did not exist. Essential loads are described shortly in this section.

Modern traffic is the leading cause of the increase in the load on quay wall structures. In the early days, people did not have cars or trucks and moved from place to place with horse and wagon. On top of that, the intensity of the traffic increased as well. Nowadays almost everyone owns a car, which they park in front of their home. The municipality of Amsterdam arranged a lot of parking space on the quay wall structures itself. The tram is also considered as an additional traffic load.

A type of load which must be accounted for is the temporary load. Quay wall structures are also often used as a storage place, where heavy equipment such as cranes or containers is stored. Another temporary load one may think of are large crowds at events like the Canal Parade or Kingsday.

| (a) | (b) |

Figure 2.7: Two types of extra temporary loads acting on the quay walls: (a) heavy cranes, (b) large crowds during Kingsday.

The presence of trees affects the quay wall structure in multiple ways. First of all, the weight of the tree gives a vertical load and a horizontal load when the weight urges the soil behind the quay wall against the structure itself. Additionally, the root system of the trees can affect the quay wall by pushing against the quay wall structure. Also, the wind load on the tree transferred to the surface via the root system can affect the structure.

2.3   Boundaries

A quay wall exists of multiple sections, which are separated with a dilatation. Dilatation is a small opening, which is sometimes filled with mortar. The dilatation is defined as the lateral end of the quay wall section, which means that boundary conditions have to be formulated there. First of all, it can be assumed that the quay wall section is not able to move in the lateral direction since multiple sections are positioned next to each other. By this assumption, it is possible that arching of internal compressive membrane forces occurs in the structure. This phenomenon is called Compressive Membrane Action (CMA). CMA occurs in a structure whose edges are restrained against lateral movement by stiff boundary elements. This lateral movement is caused by the load on top of the structure. As the structure deforms, changes of geometry cause the lateral ends to move outward. This movement is restricted, resulting in compression forces in the structure. Just like concrete, masonry has lower strengths for tension than for compression, meaning that CMA can develop only after cracking occurs in a structure.
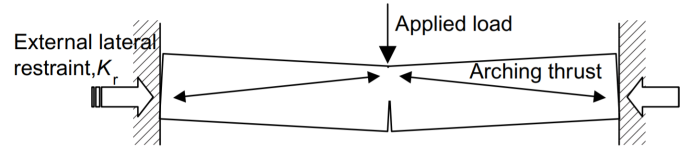
Figure 2.8: Compressive Membrane action (Ben-Gera, 2019).

CMA leads to an increase in the bearing capacity, and it fails at a load much higher than predicted by the standard yield line theory. Moreover, CMA is not taken into account when designing a structure in Europe (Ben-Gera, 2019). Therefore, quay wall structures may have a higher strength capacity than the designed strength.

Ben-Gera (2019) found out that CMA can be modelled by implying a lateral restraint in the form of horizontal support. It is essential to have sufficient lateral stiffness at the edges, which is obtained by this restraint since it implies infinite stiffness.

2.4 Failure mechanisms

This section describes the failure mechanisms for gravity walls on a wooden foundation. The failure mechanisms regarding the pile foundation itself are pointed out and elaborated. The most critical failure mechanisms, according to De Gijt and Roubos (2013) for a retaining gravity wall on pile foundations, can be found in figure 2.9. These are:

1. Exceeding of the vertical pile (pressure) bearing capacity.
2. Exceeding the tension bearing capacity.
3. Failure of the surface due to a horizontal load on the pile foundation.
4. Exceeding the general stability.
5. Structural failure of the retaining wall.
6. Structural failure of the piles due to pressure, tension, bending, buckling or shearing.
7. Collapsing of the structure due to the large displacement of the foundation.
8. Collapsing caused by internal erosion or piping.

Four of these failure mechanisms are related to foundation defects and are elaborated.

The first failure mechanism is the exceedance of the vertical bearing capacity of one of the piles. This failure mechanism can occur as a result of a load increase, or as a decrease of the pile strength. Since both aspects seem to occur, these failure mechanisms are not unlikely to happen. As can be seen in figure 2.9 (1), the quay wall tilts over when the bearing capacity of the pile is exceeded. Tilt could be a good indication that one of the piles in defected or collapsed. Note that this illustration gives just the cross-section. The length of the structure may give the quay wall an extra-strength capacity.

Failure of the surface due to a horizontal load on the pile foundation can occur when the horizontal pressure behind the structure is large enough. Overloading could be one of the causes of substantial horizontal pressures on the foundation. When too much load is acting behind the structure, the subsoil converts this load into horizontal loads acting on the quay wall and foundation.

Structural failure of the piles due to out-of-plane pressure, tension, bending, buckling or shearing can occur for large vertical loads on the pile. This failure mechanism looks like the exceedance of the vertical
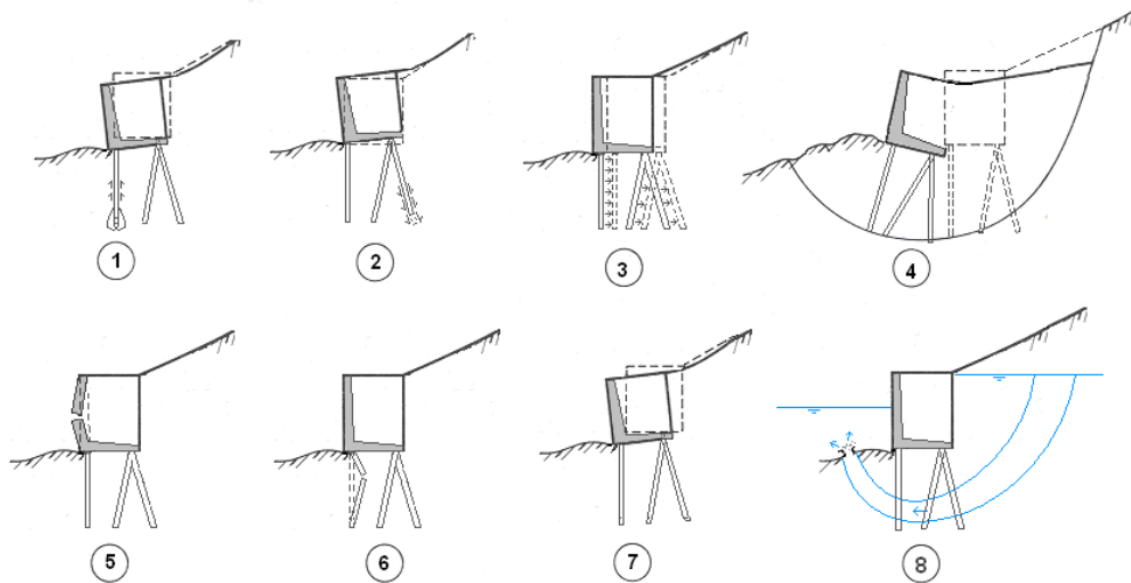
Figure 2.9: Failure mechanisms of a retaining wall on pile foundations (De Gijt and Roubos, 2013).

pile pressure since it is dependent on the vertical load. This failure mechanism is likely to occur with an initial horizontal movement of the structure and vertically overloading.

Collapsing of the structure due to large displacements of the foundation occurs when the soil pressure behind the structure is too big, resulting in the quay wall to start moving. The load acting on the structure is mainly a horizontal load, which results in a rotational moment.

3

Masonry

This chapter outlines the information about the material masonry and its behaviour. First, the symptoms of structural damage in masonry is described. Thereafter, the nonlinear behaviour of masonry in tension, compression and shear is explained. Eventually, two material models are presented: the Total Strain Crack model and the Engineering Masonry model.

Masonry has been used as a building material for many years by many populations. The Great Wall of China, the old Inca cities or the coliseum in the Roman empire are just a few of the many examples of the use of masonry throughout history. Lourenço (1996) explained this popularity by the most important characteristic of masonry: "its simplicity". He pointed out that stacking pieces of stone or bricks on top of each other, is a "simple, though adequate technique".

Although the concept of stacking units has not changed much, the properties of masonry structures itself have evolved over the years (figure 3.1). This has to do with three changes. First of all the type units have changed throughout the years. Stonework masonry has been replaced by calcium silicate bricks, resulting in different properties for the units. Also, mortar was introduced as a cohesion material between the unit bricks, causing them to be 'glued' together. Lastly, the properties of the mortar altered from a lime-based mortar to a cement-based mortar. This change caused the mortar to be less ductile and therefore more stiff, which indicates that the older masonry structures are structures with lower strength and stiffness than masonry structures built nowadays.
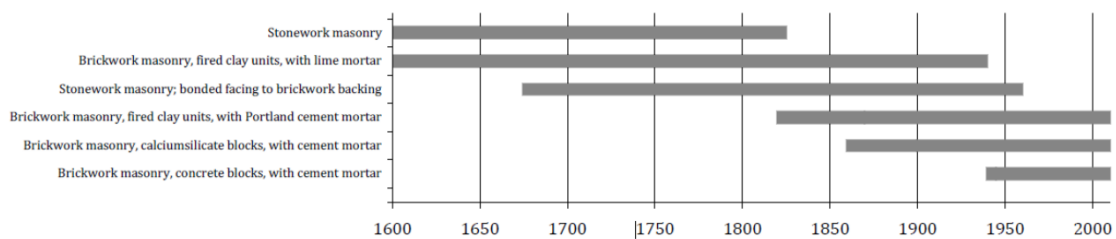

Figure 3.1: Types of Masonry throughout the years Van Noort (2012).

Masonry is a composite material, which means that it consists of two or more different components. In the case of masonry, it contains units and mortar. The properties of masonry are strongly dependent

upon the properties of its constituents (Lourenço, 1996). Likewise, masonry is an anisotropic material, meaning the physical properties are different in various directions. Van Noort (2012) mentioned that mortar with a lower stiffness increases the relative differences of the Young's modulus of the masonry in both directions: the stiffness in the horizontal direction was bigger than the stiffness in the vertical direction. Due to the ductile behaviour of mortar, the differences in material properties between the vertical- and horizontal direction is bigger for older quay walls than for the newer, more stiff ones.

3.1   Symptoms of structural damage

The structural behavior of masonry can be distinguished into three categories: deformation, crack development and tilt, reported by De Vent (2011): "Deformations indicate a lack of stiffness, cracks point to a lack of strength, and tilt designates a lack of external stability."

It is especially hard to make a distinction between deformations and cracks since cracks can be seen as discontinue deformation. When masonry starts to deform, microcracks arise in the material. When cracking occurs, these microcracks start to localize and a macrocrack occurs (Hordijk, 1992). Cracks can be seen as a discontinuous deformation and is defined by the strength of the material. The transition from a continuous element to cracks is therefore dependent on the material. For quasi-brittle materials, it is not an abrupt transition, because the process develops more gradually.

Five basis load situations for deformation can be distinguished: compression, tension, shearing, bending and torsion. Figure 3.2 shows how deformation develops for these load situations and how cracking occurs for quasi-brittle material. Compression will cause the element to be shortened, while tension will result in an elongation of the element. For tension, the crack direction occurs perpendicular to the load direction. Under compression, the crack arises parallel to the load direction. An explanation for this phenomenon is that the compressive strength is much larger than the tensile strength.
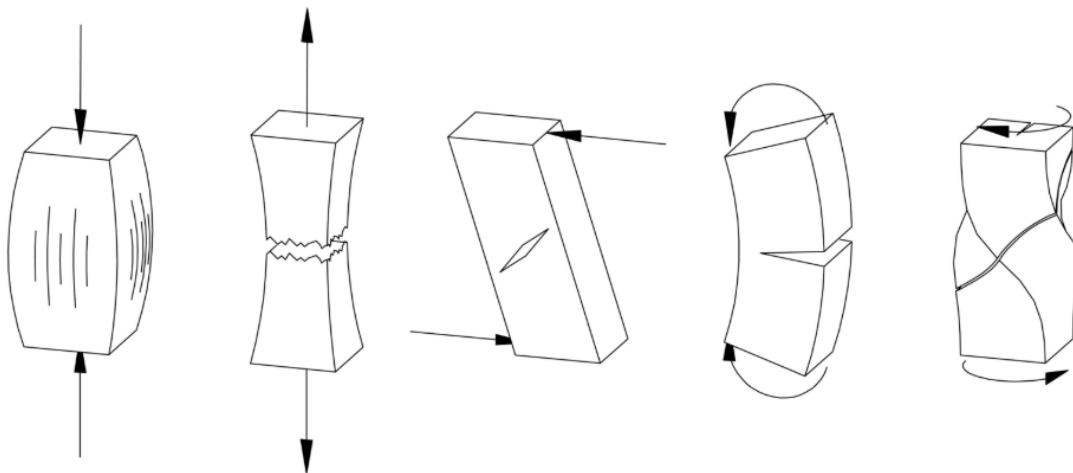

Figure 3.2: Five basic load situations for deformation and cracks: compresion, tension, shear, bending, and torsion (De Vent, 2011).

Shearing occurs when two loads are in opposition and parallel direction, given that they do not act at the same working line. Shearing results in a combination of compressive and tensile stresses, resulting in a crack occurring at an angle of 45°. While during shearing the cross-section remains parallel to each other, bending makes the cross-sections rotate to each other. The crack starts at the point where the element experiences the biggest tensile stress. Torsion occurs when the force causes a twist in the structure.

The load situations of crack result in three modes of failure, mode I, II and III (figure 3.3). Mode I, the pure opening mode, is regarding cracks caused by normal stress, such as tension and compression. Mode II is called the shearing mode and is caused by shear. Mode III is the tearing mode and is caused by out-of-plane stresses.
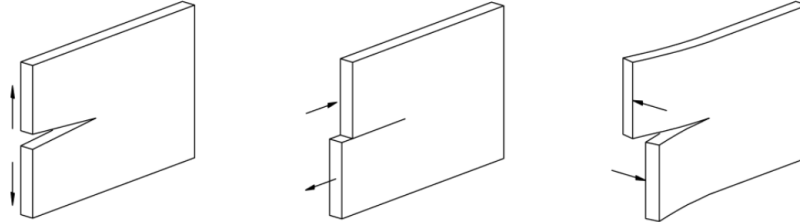

Figure 3.3: Mode I: pure opening, Mode II: shearing mode, Mode III: tearing mode.

3.2   Nonlinear behaviour of masonry

The material properties described in this section are the Young's modulus, Poisson's ratio and, the mass density. The Young's modulus is also characterized as the modulus of elasticity and can be determined by dividing the stress over the strain or by considering the slope of the linear elastic part of the stress-strain diagrams. Because masonry is a composite material, the Young's modulus of masonry is dependent on the characteristics of the two components: brick units and mortar. The brick units are in principle more stiff than mortar. The formula's of Pande et al. (1994) give good estimations for the values of the Young's modulus of masonry. Van Noort (2012) stated that the stiffness in the horizontal and vertical direction is approximately the same, but that in some cases the stiffness in the horizontal direction seemed to be higher. This is related to the presence of mortar in this direction, which is less compared with the vertical direction. Likewise, Poisson's ratio of masonry can be formulated by using the properties of both components. Nevertheless, the values calculated by the formula's of Pande et al. (1994) differ from the numerical values found by Van Noort (2012), which are a bit larger. However, it can still be assumed that the formulas are accurate enough, hence they can still be used. Another approach to determine the material properties of masonry is to use the tables given in [NPR998:2018]. This assessment gives guidelines for values to use for masonry structures.

Behavior in tension

The tension behavior of masonry shows a great similarity with the behaviour of concrete (Van der Pluijm, 1997). Concrete and masonry are both considered quasi-brittle, since the brick units, mortar and concrete are stone-like materials.

A typical stress-displacement curve of masonry under tension regime is shown in figure 3.4. The graph consists of a linear elastic part, a peak, and a softening area. The Young's modulus reflects the stiffness of the masonry and applies to the linear elastic part of the diagram. In this part of the stress-strain-diagram, only microcracks may occur. At the moment the maximum tensile strength is reached, these microcracks start to localize within a narrow zone and a macro crack is formed (Hordijk, 1992). At this point, cracking has occurred. The post-peak behavior of masonry is described by the second part of the graph, the softening part. Lourenço (1996) mentioned the following about softening:
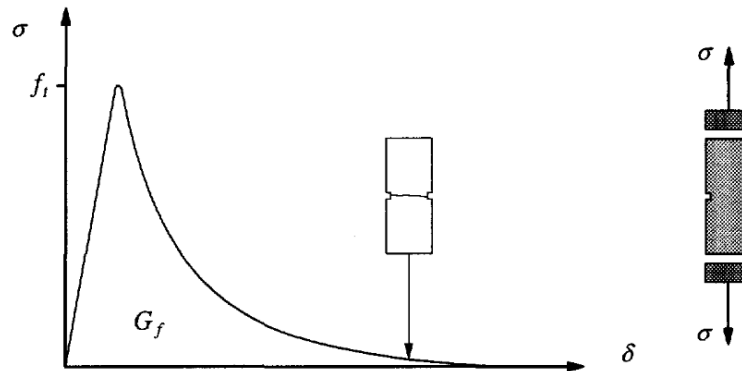
Figure 3.4: Tension behavior of masonry (Lourenço, 1996).

"Softening behavior is a gradual decrease of mechanical resistance under a continuous increase of deformation forced upon a material specimen or structure [...]. Initially, the microcracks are stable which means that they grow only when the load is increased. Around the peak load, an acceleration of crack formation takes place and the formation of macrocracks starts. The macrocracks are unstable, which means that the load has to decrease to avoid an uncontrolled growth." (Lourenço, 1996)

To describe the softening behaviour, the fracture energy is considered as the amount of energy by unit area that is needed to create a crack in which no tensile stresses can be transferred (Van der Pluijm, 1997). The fracture energy can be found by taking the integral of the stress-displacement graph, thus by taking the area underneath it.

Behavior in compression

The stress-displacement curve (Figure 3.5) for compression has some similarities with the tension one. The graph is also defined by a linear elastic part, a peak at the maximum compression strength and softening. The main differences are that non-linearity starts before the peak is reached. Before the peak is reached, the stiffness of the material starts to decrease while the load is still able to increase.
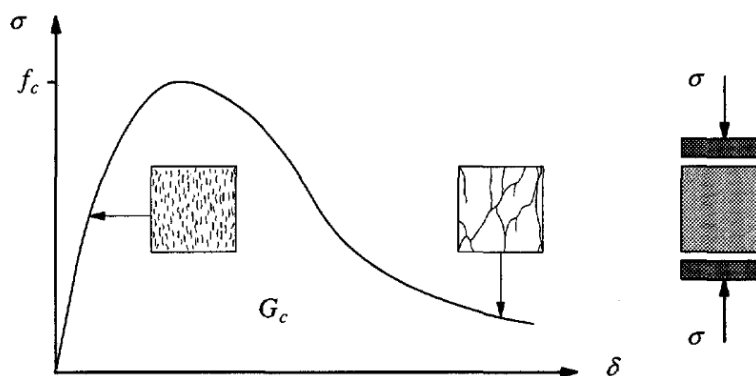
Figure 3.5: Compression behaviour of masonry (Lourenço, 1996).

The compression graph is indicated by the compression parameters: Young's modulus, maximum compressive strength, softening and the compressive fracture energy. These parameters determine the values and the shape of the graph.

Behavior in shear

Shear is part of the failure mechanism, generally identified as mode II. It consist of slip of the unit-mortar interface under shear loading. Figure 3.6 shows the behaviour of masonry in shear. Two lines can be distinguished in this graph: a curve without the contribution of compression ($\sigma = 0$) and a curve with this contribution ($\sigma > 0$). For the first one, the maximum shear strength is indicated by the cohesion parameter (c), while the second curve shows an extra shear capacity due to the compressive stress.
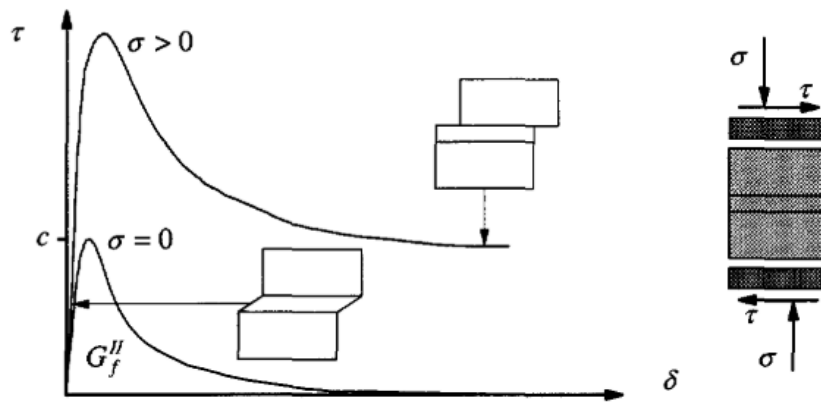
Figure 3.6: Shear behaviour of masonry (Lourenço, 1996).

Lourenço (1996) stated that it is assumed that the inelastic behaviour in shear can be described by the mode II fracture energy G_f^{II}. The fracture energy is defined by the integral of the $\tau - \delta$ diagram in the absence of normal load.

3.3   Material models

Masonry is a composite material, which means that it consists of two or more different components. It contains unit bricks and mortar. Each component contains its own material properties, such as Young's modulus, Poisson's ratio and, the Shear modulus. This must be considered in the model.

Rots (1991) pointed out that there are various strategies to model masonry in a finite element software package, depending on the degree of simplicity versus accuracy required (Figure 3.7). In the detailed micro-model, the material properties of both the brick units as the mortar are taken into account and represented by continuum elements. These elements are then connected by an interface element, which represents a potential crack and slip plane. A dummy stiffness is given to this interface element to avoid interpenetration of the continuum ((Lourenço et al., 1995). In the second approach, the simplified micro-model, the properties of the mortar and the unit-mortar interface are defined into an average interface. The brick units are expanded to keep the same geometry. The macro-model describes the composite behavior in terms of average stresses and strains, smeared out over the elements. For that reason, masonry can be assumed to be a homogeneous, anisotropic material (Lourenço, 1996).
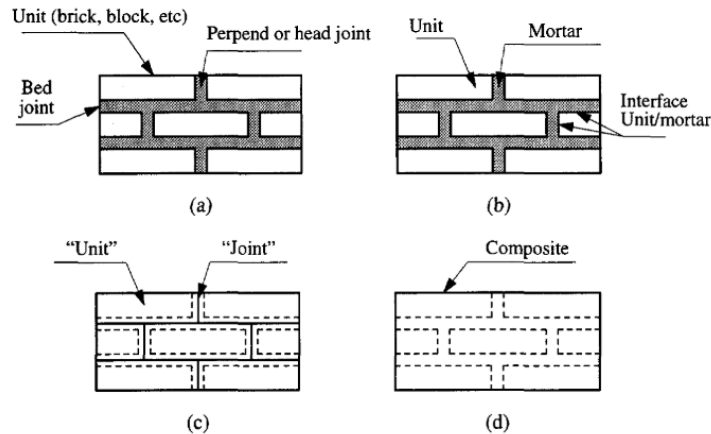
SWECO

Figure 3.7: Modeling strategies for masonry structures: (a) masonry sample; (b) detailed micro-modeling; (c) simplified micro-modeling; (d) macro-modeling. (Lourenço et al., 1995).

Smeared cracking

In the smeared crack approach, cracks are imagined to be a continuum instead of a discrete crack. The smeared crack approach is a macro model, where all the properties of the different components are smeared over the element. For masonry, this means that the properties of the mortar and the brick units are combined and converted into the properties of one material. Rots and Blaauwendraad (1989) mentioned that the procedure is attractive because it it does not impose restrictions with respect to the orientation of the crack planes. In other words, the orientation is not fixed by the interface elements. An other advantage of this approach is the relatively short computational time. Two material models that make use of the smeared concept are elaborated in this section: the Total Strain Cracking model (TSRC) and the Engineering Masonry model (EMM).

Total Strain Cracking model

The Total Strain Crack model, originally proposed by Vecchio and Collins (1986), is a constitutive model based on total strain. It makes use of a smeared approach for the fracture energy [CITE] [DIANA MANUAL], with plane stress (membrane) elements. The model was first intended to numerically model concrete but appeared also to be applicable to masonry structures. The model considers the material to be isotropic by using the averaged total stress-strain relationships of the material in all directions.
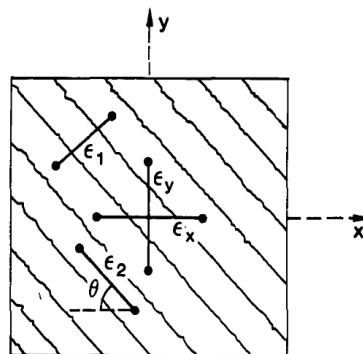
Figure 3.8: Average strains in cracked element (Vecchio and Collins, 1986).

The membrane element represents a component of the masonry structure. The loads, acting on the element, are assumed to consist of two uniform axial stresses (f_x and f_y) and an uniform shear stress (v_{xy}). These three stresses correspond each with the three in-plane strains: ϵ_x, ϵ_y and γ_{xy} (Vecchio and Collins, 1986). Knowing the values for the three strain direction, any direction can be determined with help of the Mohr's circle of strain. The principle strains (ϵ_1 and ϵ_2) (Figure 3.8) and corresponding stresses are obtained, which means, when cracking occurs, the direction of the cracks can be obtained as well. This only hold for the rotating model, where the direction of the crack can rotate each time the stress-strain relation is evaluated (Van Noort, 2012). In the fixed model, the direction of the crack is assumed beforehand and therefore can not change. When cracking occurs, the crack is smeared over the area of an integration point (Figure 3.9).
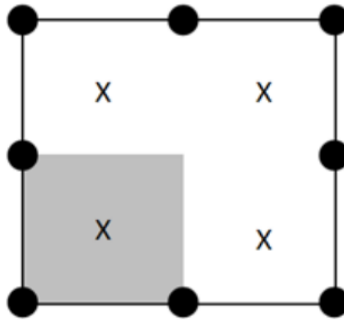
Figure 3.9: An element with one cracked integration point. (Van Noort, 2012).
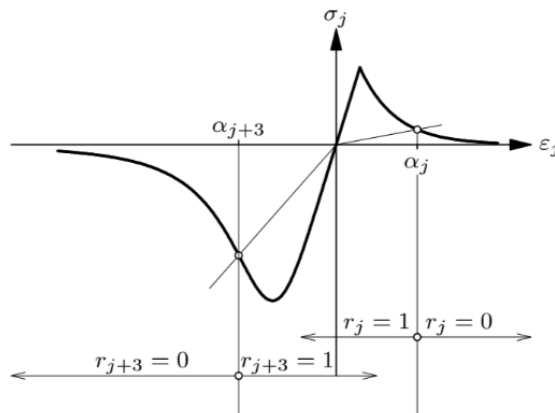
Figure 3.10: The loading and unloading path of the Total Strain Crack model.

The loading and unloading path of the Total Strain Crack model is presented in Figure 3.10. The tension regime is plotted on the right side of the vertical axis, while the compression regime can be seen on the left side. This graph is only valid for the principle directions. The graph consist of a couple important phases. In the tension regime, a linear part, a peak and a softening part can be distinguished, while in the compression regime a extra hardening part can be noted. This corresponds with the graphs Figure 3.4 and Figure 3.5. The thinner line represents the unloading path of the material. When the softening region is reached, the unloading part will follow this path back to the origin of the axes.

There are several ways to model the softening part of the curve (the post-peak behaviour). A distinction have to made between tension and compression, since the behaviour is different and therefore various

theories exist. These methods are summarized and predefined for the use of FEM.

Engineering Masonry model

The Engineering Masonry model is, like the Total Strain Cracking model, a smeared failure model. Regular plane stress membrane and curved shell elements can be used. Total strain-based continuum model. According to Schreppers et al. (2016), the Engineering Masonry model overcomes the two shortcomings of the Total Strain Crack model. First of all the Engineering Masonry model includes orthotropy. This means that the elasticity, strength and softening have to be formulated in the x- and y-direction and can have different values for both axes. In this case, the x-direction is aligned with the bed joints and normal to the head joints, which are oriented in the y-direction. Secondly, the unloading-reloading behavior is described more realistically compared with the Total Strain Crack model, where the unloading-reloading curves strongly underestimate the dissipation under cyclic loading. Both the models assume linear unloading for compressive stresses. However, the Engineering Masonry model assumes this behavior with initial elastic stiffness, which is more realistic compared with the Total Strain Crack model. Figure 3.11 shows the constitutive behaviour in compression, tension and shear for the Engineering Masonry model.

(a) Compression

(b) Tension

(c) Shear
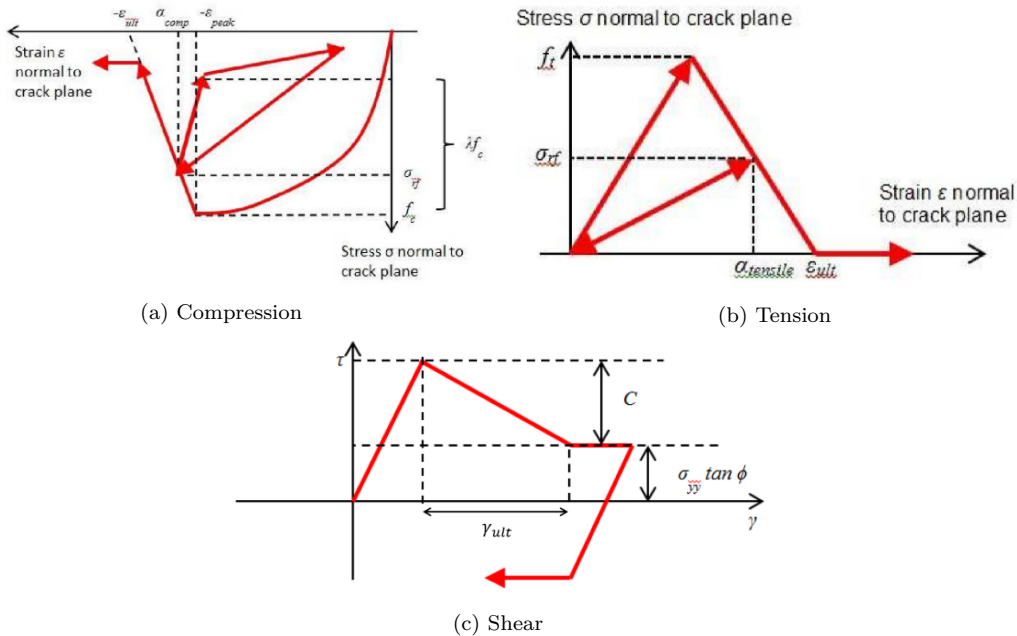
Figure 3.11: Predefined behaviour for the Engineering masonry model (Schreppers et al., 2016).

4

The base case

This chapter describes the base case, that is the starting point for the parametric study performed in Chapter 5. First, the physical model and the mechanical model are presented, where the base case is described in terms of geometry, material properties, boundary and loading conditions, and foundation properties. After that, the choices regarding the Finite Element Model are clarified. A sensitivity analysis is performed to check the mesh size, load step size, convergence norm and the material model.

4.1 Physical model

Urban quay walls are long structures which are divided into sections of various lengths. From data obtained from IBA, it can be assumed that one section has a length of twenty meters and a height of two meters. Vertical dilatation joints filled by mortar are placed between sections. The masonry of the old quay wall structures consists of red clay bricks and cement-based mortar. Figure 4.1 shows an illustration of the physical model of the quay wall.
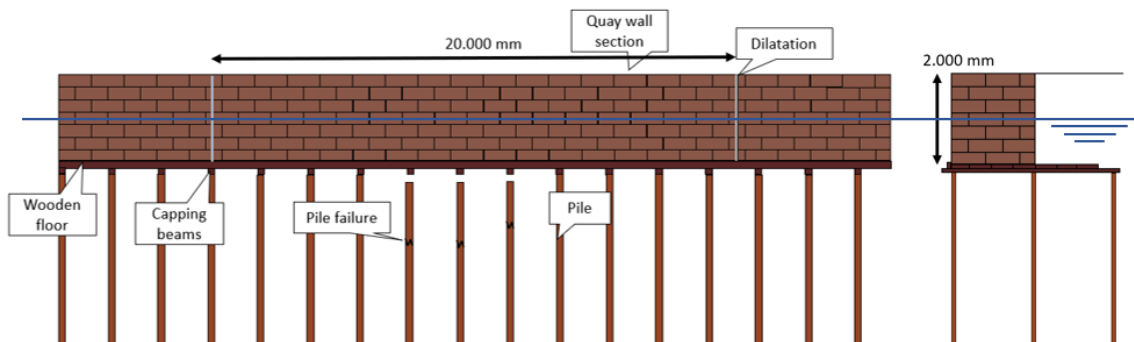
Figure 4.1: Physical model of the basecase.

The foundation of the quay walls mainly consists of wooden piles, connected with capping beams. A wooden floor is placed on top of the capping beams. On top of this table, the masonry wall is placed, which act as a gravity wall. It is assumed that there is no other connection between the masonry and the foundation since no evidence of such a connection exist. When one or more piles are failing, the masonry structure is not supported at that position anymore.

4.2 Mechanical model

A 2D mechanical model is considered, meaning that a uniform stress distribution over the thickness of the wall (plane stress condition) is assumed. Cross-sectional differences and out-of-plane deformation are not considered since pile failure has a significant influence in the vertical in-plane direction. Both cross-sectional differences and out-of-plane loading influence the horizontal movement of the structure, which is not in the scope of this thesis. However, it could be of significant influence in other failure mechanisms. Only one section of the urban quay wall is considered for the mechanical model. This section has a length of 20 metres and a height of 2 metres. Figure 4.2 shows an illustration of the mechanical model of the quay wall section.
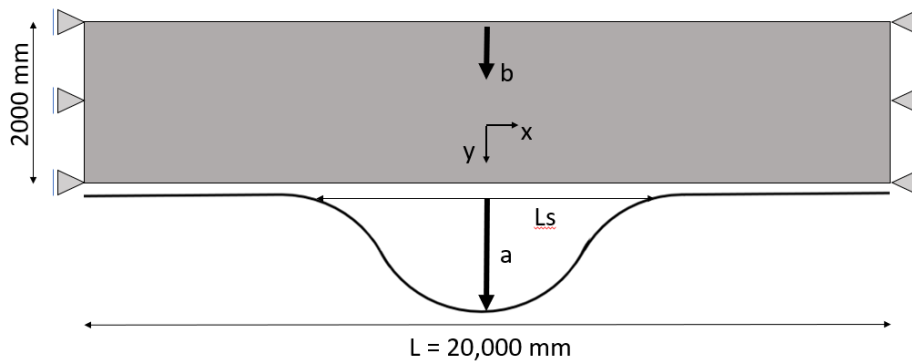
Figure 4.2: Mechanical model of the basecase.

Since no information on the material properties of the masonry is available, the properties suggested by NPR 9998 (NEN, 2018) are adopted. A distinction has to be made for the properties in the x- and y-direction since masonry is orthotropic. Nevertheless, an isotropic material model is used for the basecase. The influence of orthotropic behaviour is analysed in Chapter 5. An overview of the material parameters is given in Table 4.1.

Material parameter	Value
Young's modulus	5000 N/mm^2
Mass density	1800 kg/m^3
Tensile strength	0.1 N/mm^2
Tensile fracture energy	0.01 N/mm
Residual tensile strength	0.001N/mm^2
Compressive strength	8.5 N/mm^2
Compressive fracture energy	20 N/mm
Shear modulus	2000 N/mm^2
Shear strength	0.35 N/mm^2
Shear fracture energy	0.1 N/mm
Cohesion	0.3 N/mm^2

Table 4.1: Overview of the material properties of masonry.

The boundary conditions deal with the lateral sides of the structure. Because there is no interlocking between the bricks at the dilatations, the assumption is made that there is no interaction at all. However, some forces are present in these dilatations because the quay wall is not able to move in the lateral direction, due to the presence of other quay walls sections. Therefore, horizontal constraints are implemented,

so the lateral sides are not able to move in the horizontal x-direction. The lateral sides can move freely in the vertical y-direction since it is assumed that the frictional forces in the dilatations are negligible. It is assumed that Compressive Membrane Action (CMA) occurs at the structure since the edges are restrained against lateral movement by the horizontal supports. The influence of the presence of the horizontal supports is analysed in Chapter 5.

Material parameter	Value
Length	20,000 mm
Height	2000 mm
L_s/L	0.3

Table 4.2: Overview of the geometrical parameters.

Figure 4.2 shows two deformations: *deformation a*, which represents the imposed deformation and *deformation b*, which represents the deformation on top of the structure. To obtain *deformation a* an imposed Gaussian function is used. It is assumed that failure of one or more of these piles leads to deformations in the capping beams and therefore deformations in the wooden floor. According to Giardina (2013), deformations in the subsurface can be modelled as a steel beam, with an imposed displacement. For this model, a line with the properties of a steel beam and an imposed deformation is implemented. The vertical displacement is indicated with a, and the length of the displacement is called L_s. This length represents the number of foundation piles failing underneath the quay wall section. The imposed deformation in the subsurface will lead to a reaction in the structure on top of it. The vertical deformation on top of the structure is indicated with b. Because the masonry is not connected to the foundation, no tension is assumed between the structure and the imposed line beneath it. For the base case, it is assumed that the length of the foundation defect is equal to 6 meters ($L_s/L = 0.3$). Table 4.2 shows an overiew of the geometrical parameters.

4.3 Finite element model

The model is a 2D-model, which means that a uniform stress distribution over the thickness of the wall is considered. The smeared crack approach is used to represent the masonry. Figure 4.3 shows an illustration of the finite element model. An overview of the elements used in the model is presented in Table 4.3. The elements are 8-noded plane stress elements, 3x3-noded interface elements and 3-noded beam elements.
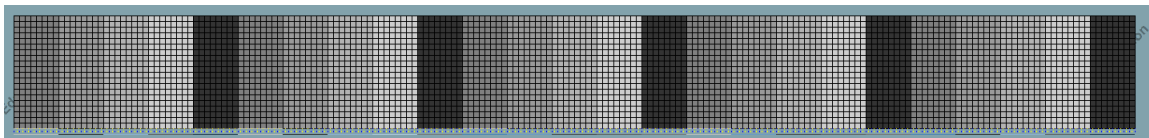

Figure 4.3: An illustration of the Finite Element Model.

The chosen model parameters used for the base case can be found in Table 4.4. This table shows the results of the choices for the mesh size, load step size and the material model. Moreover, it shows that an integration scheme of 3 x 3 Gauss is used and which convergence norm is used for all analyses. When nothing is mentioned about these parameters, these inputs are used for these parameters.

The interface conditions characterize the interface between the quay wall structure and the foundation. The interface can be seen as a dummy since it has no meaning physically and only describes the interaction between the components. Assumptions are made to implement the conditions into the model. The
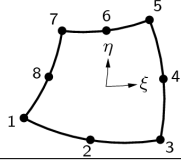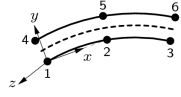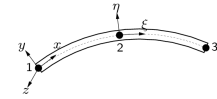
Element type	Parameters	
CQ16M	DOF	u_x, u_y, 8-noded
	Interpolation Scheme	Quadratic
	Integration scheme	3×3 Gauss integration
	Shape dimension	2D
	Topological dimension	2D quadrilateral
CL12I	DOF	u_x, u_y, 6-noded
	Interpolation Scheme	Quadratic
	Integration scheme	3-point Newton-Cotes
	Shape dimension	2D
	Topological dimension	1D
CL9BE	DOF	u_x, u_y,ψ_z, 3-noded
	Interpolation Scheme	Quadratic
	Integration scheme	2-point Gauss integration
	Shape dimension	2D
	Topological dimension	1D

Table 4.3: Overview of the elements used in the model.

Mesh size	100 mm
Integration scheme	3×3 Gauss
Step size	0.1 mm (200)
Material model	Engineering Masonry model
Convergence norm	Displacement = 0.01 Force = 0.01 Satisfy both norms = Yes

Table 4.4: An overview of the model parameters used for the analyses.

masonry quay wall is gravity-based, which means that there is no clear connection between the two components. Therefore, it is assumed that no tension occurs. Secondly, it is assumed that the interface has no shear stiffness because the horizontal edges will prevent the structure from shearing. At the lateral end, supports fixed in the x-direction are applied. These supports will constrain the structure to move in the x-direction. It is assumed that there is no interaction in the vertical direction; thus, the supports are able to move vertically.

4.4 Sensitivity analysis

In this section, a sensitivity analysis is performed to check the influence of the mesh size, load step size, convergence norm and material model.
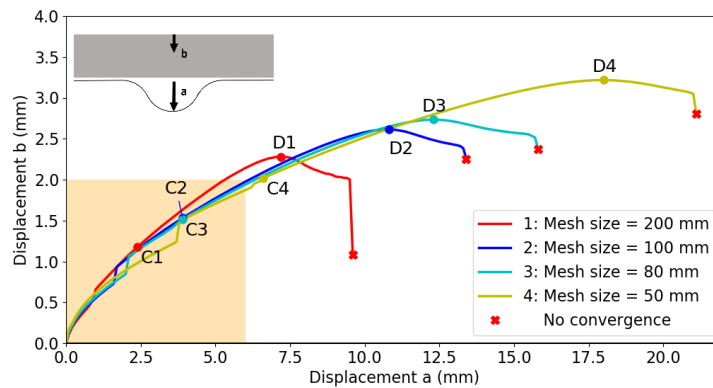
Mesh size

The influence of mesh size on the results is checked by adopting models with mesh sizes of 200, 100, 80 and 50. A coarse mesh size will result in a short computational time, while a finer mesh will converge to a more realistic result. Figure 4.4 shows the displacement graphs for the four mesh sizes. The displacement on top of the structure (*displacement b*) is plotted as a function of the deformation of the bottom support (*displacement a*). The graph also shows markers for four representative load levels: A, B, C and D.

Figure 4.4 shows that all the analyses give different results, especially regarding the crack development. The initiation of the first vertical crack, indicated by point A in Figure 4.4a, occurs at different moments
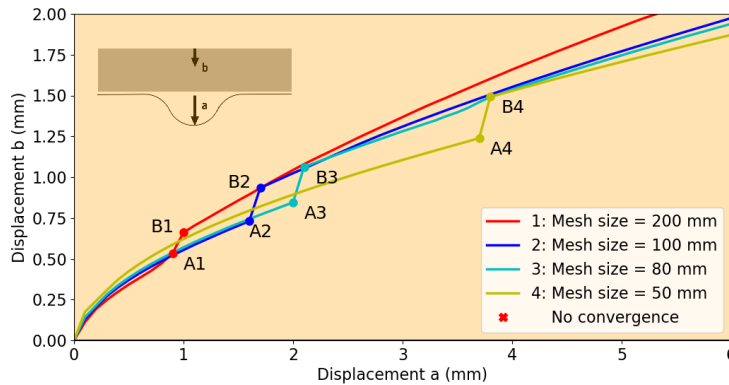
(a)

(b)

Figure 4.4: Influence of mesh size.

for all four mesh sizes. The same holds for the ultimate displacement on top of the structure, indicated by point D. Besides the ultimate displacement is reached at a later moment, it also reaches a higher value for a finer mesh. The differences in the graphs can be explained by the stress-strain relationships in tension, given in Figure 4.5. This figure shows that the elastic tension part for all the mesh sizes is the same. Looking at the post-peak behaviour, differences can be observed: A finer mesh results in a milder slope. The reason that the slope changes with different mesh sizes is because of the definition of the crack bandwidth. Since the smeared crack approach is used, the model needs input for the crack bandwidth. (Rots, 1988) proposed a method, which is eventually implemented in Diana FEA as default for determining the crack bandwidth. This method considers the crack bandwidth to be equal to the mesh size. This means that the crack bandwidth changes when the mesh size is adjusted. When the crack bandwidth is dependent on the mesh size, this value changes with different mesh sizes. The ultimate strain will increase with a decreasing mesh size, resulting in a milder slope. A milder slope also means that the material acts more ductile. The area underneath the graph is defined by the fracture energy in tension (G_{ft}) divided by the crack bandwidth (h). This means that a finer mesh produces a larger area and therefore a milder slope after the peak.

This behaviour can also be observed in the crack patterns of the model. The principal strains are presented in Figure 4.6. The red colour indicates the places where fully developed crack occurs. The cracks for the models with a finer mesh are smaller than the cracks for models with a coarser mesh. This is due to the
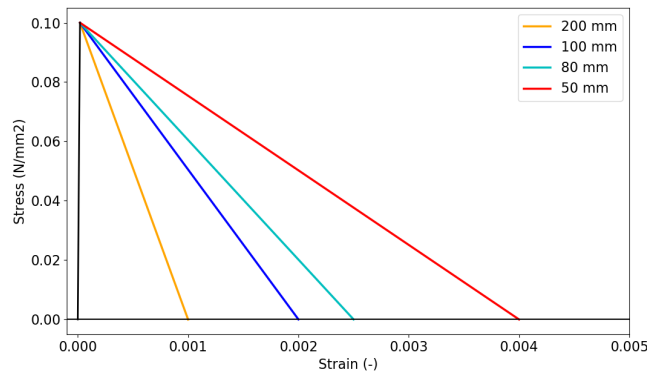
Figure 4.5: Stress-strain relationships of four models: comparison between the mesh sizes with corresponding crack bandwidth according to the theory of Rots (1988).

fact that the ultimate strain is smaller for a bigger mesh size and therefore reached at an earlier stage. This indicates that in models with a big mesh size, cracking occurs more brittle than in models with a fine mesh size.

Although the cracks do not occur at the same time, the crack patterns are more or less the same. Taking this into account and considering the computational time of the model, a mesh size of 100 mm is chosen for the base case. By doing this, it is possible to compare the results to each other and determine the influences of the adjustments. Furthermore, it is also a conservative choice, since damage patterns for this mesh size occur at an earlier stage than for a finer mesh. However, it must be noticed that the absolute values of the analyses can differ from reality.
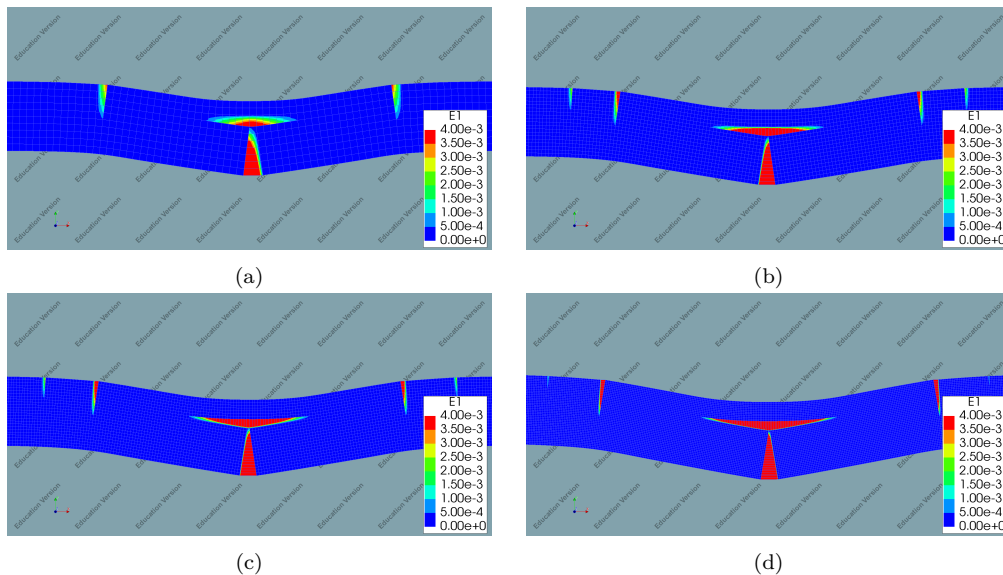
Figure 4.6: Crack patterns at a maximum displacement b (point D): comparison between the mesh sizes (a) 200 mm at D1, (b) 100 mm, at D2, (c) 80 mm at D3, and (d) 50 mm at D4, with corresponding crack bandwidth, according to the theory of Rots (1988).

Load step size

Two analyses, both with mesh size 100 millimetres, are performed to check the influence of the load step size. The first analysis is executed with load step size 0.1 millimetre, while the second analysis is executed with load step size 0.05 millimetre. The results are shown in Figure 4.7.
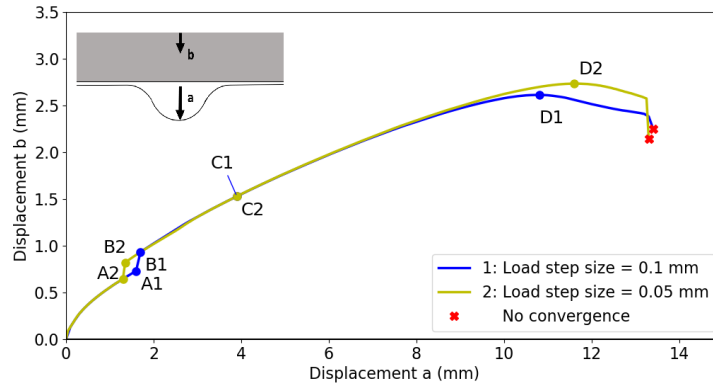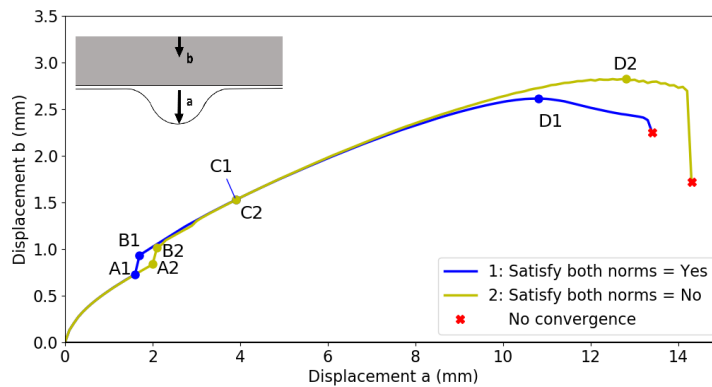

Figure 4.7: Vertical displacement in point B as a function of the applied settlement in point A: comparison between a load step size of 0.1 mm and 0.05 mm.

The graphs do not show large differences between each other, except for the location of the jump and the position of point D. As mentioned before, this jump represents the transition between the partially open crack to fully open crack, while point D indicates the maximum displacement on top of the structure. Looking at the applied displacement when the first crack occurs, the difference is 20%, which is quite extensive. On the other hand, the difference for point D is on 7%. This difference is only important for the applied displacement where the cracking appears. As illustrated in Figure 4.8, the crack patterns at point D are similar, which indicate that this difference has a limited influence on the estimation of the final crack pattern.
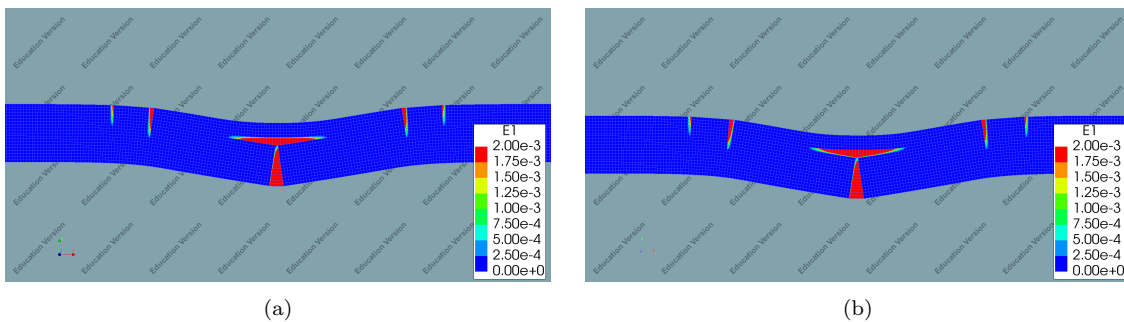
Figure 4.8: Influence of the load step size (a) load step of 0.1 mm, (b) load step of 0.05mm.

Considering the computational time and the fact that the crack pattern does not differ very much, Load step size of 0.1 mm is chosen for the base case. Nevertheless, it is important to keep in mind that the model gives differences regarding the applied displacement, and therefore also for the values of the displacement on top of the structure.

Convergence norm

Two analyses are performed to check the influence of different setups for the convergence criteria. Both analyses are performed with mesh size 100 and load step size 0.1 millimetres. The first analysis satisfies both the force as the displacement norm, while in the second analysis, one norm is satisfied. Satisfying only one norm saves computational time.

Figure 4.9 shows different graphs for the two analyses. The main differences between the graphs can be seen at the moment of initial cracking and the maximum applied displacement. The difference at first cracking is 24%, while the differences in the maximum applied displacement is 19%. The crack patterns in Figure 4.10 are almost the same. Nevertheless, a small difference can be noticed at the vertical crack in the middle of the structure. When satisfying both norms, this crack appears in one element only, while for the other analysis, this crack occurs in two elements.

Figure 4.9: Vertical displacement in point B as a function of the applied settlement in point A: comparison between an analysis which satisfies both norms and an analysis which only satisfies one norm.
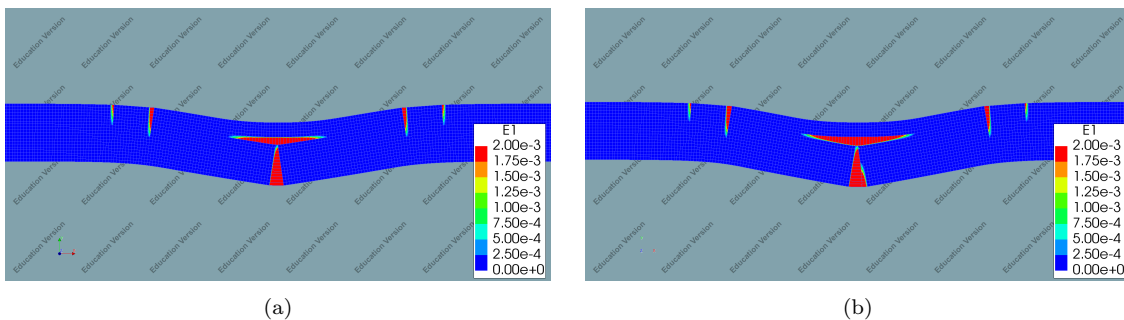
Figure 4.10: Influence of the convergence norm: (a) Satisfies both norms = yes (b) Satisfy both norms = no.

The choice is made to let the model satisfy both norms simultaneously since this gives more representative results regarding both the values for the applied displacement as the crack patterns.

Material model

The last decision that has to be made is which material model to use. section 3.3 describes the differences between the two material models, used in a smeared crack approach: The Total Strain Rotating Crack model (TSRM) and the Engineering Masonry model (EMM). Analyses with both material model are performed, with the same mesh size and the same load step size and isotropic material properties. However,

when adopting the Total Strain Rotating Crack model, the convergence issue arises, and convergence of only one of the two norms is allowed. For the Engineering Masonry model, convergence for both the norms where allowed. The results are illustrated in Figure 4.11.
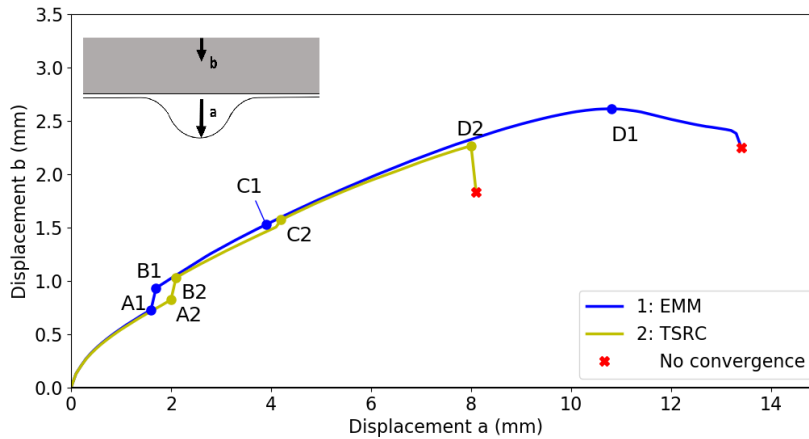

Figure 4.11: Vertical displacement in point B as a function of the applied settlement in point A: comparison between the Engineering Masonry model (EMM) and Total Strain Rotating Crack model (TSRC).

The results are quite similar, but differences can be noticed. The figure shows that at the imposed displacement (A), the Total Strain Cracking model was not able to satisfy either the displacement or the force norm. On the contrary, the Engineering Masonry model is able to satisfy both norms. Also a difference in the graph at the moment the first crack appears can be observed, with a difference of 24%. Looking at point D, this is difference is 26%.

(a)   (b)
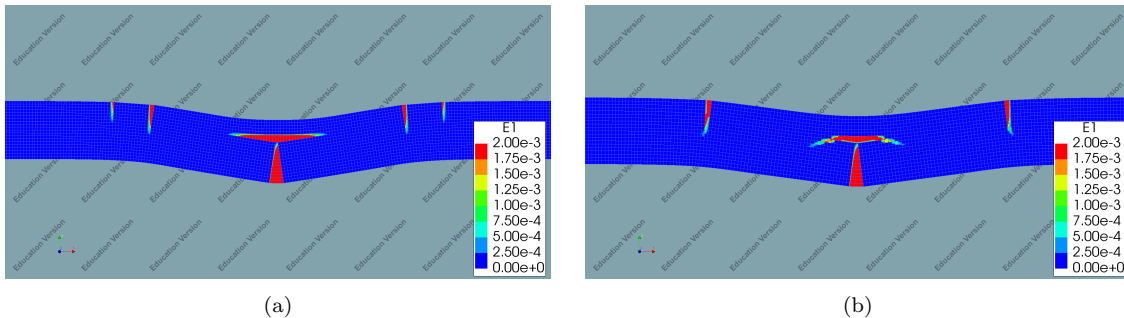
Figure 4.12: Influence of the material model: (a) EMM (b) TSRC.

The Engineering Masonry model is chosen for the base case, since this model is more stable than the Total Strain Crack model, and give more accurate results. Although the computational time is larger than for the Total Strain Crack model, this disadvantage is overcome by the accuracy of the model. Moreover, the EMM allows accounting for orthogonality.

5

Numerical results

This chapter shows the results of the analyses. These analysis are done with the help of a Python script (Appendix B). First, an analysis for the base case is performed to outline a reference for the parametric analyses. Four analyses are performed: The first analysis is performed in terms of the properties of the material. Three types of masonry (weak, average and strong) are compared with each other. Differences are made regarding Young's modulus, tensile strength and fracture energy. The second analysis shows the influence of constraints at the lateral ends of the structure. In the third analysis, the influence of the location of the foundation defect is considered. The foundation defect is moved to the left side of the structure, making the situation asymmetric. In the last analysis, the masonry is given different values for the x- and y-direction. Differences between the orthotropic and isotropic approach are illustrated. All the analyses are performed with isotropic material, except in the last section.

5.1   Results base case

To outline a reference for the parametric analyses, the results of the base case are presented. The input parameters for this base case can be found in Chapter 4. First, vertical equilibrium is checked. This is done for every load step by checking the sum of the vertical reaction forces in the model and compare it with the self-weight of the structure, calculated as:

$$(h \cdot l \cdot t \cdot \rho_{masonry} + h \cdot w \cdot t \cdot \rho_{steelbeam}) \cdot g = 85347N. \tag{5.1}$$

The sum of the reaction forces shown by DIANA give for every load step a force of 85347 N as well, meaning a vertical equilibrium.

Figure 5.1 shows the stress distribution in the model. The vectors show the direction and the magnitude of the stresses. In this illustration, the blue indicates maximum compressive stresses, which is located at the top in the middle and at the bottom above the inflexion points. The arch formed by Compressive Membrane Action is visible through the green compressive vectors. Red indicated the tensile stresses, which can be observed at the bottom in the middle and at the top above the inflexion points. It is expected that cracking occurs at these places.

Figure 5.2d shows that three types of crack can be distinguished: the vertical crack starting from the bottom in the middle of the structure, cracks above both the inflexion points of the imposed Gauss curve and a horizontal crack in the middle of the structure. Since it is possible that these cracks indicate
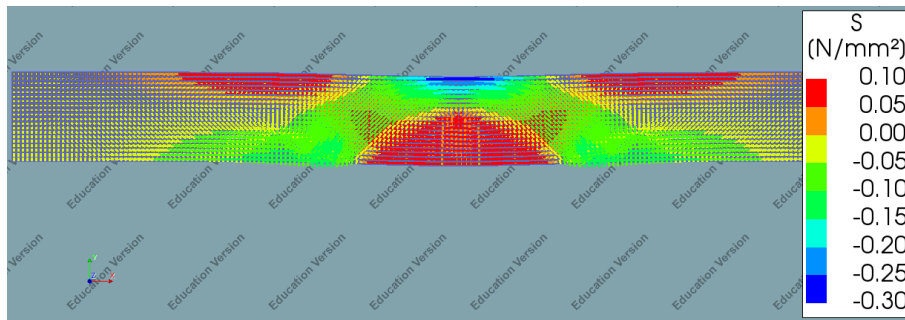
Figure 5.1: Stresses as a vector in the structure.

foundation defects in quay walls, the load level at which these cracks are formed are considered as reference for the comparison in the following parametric study.
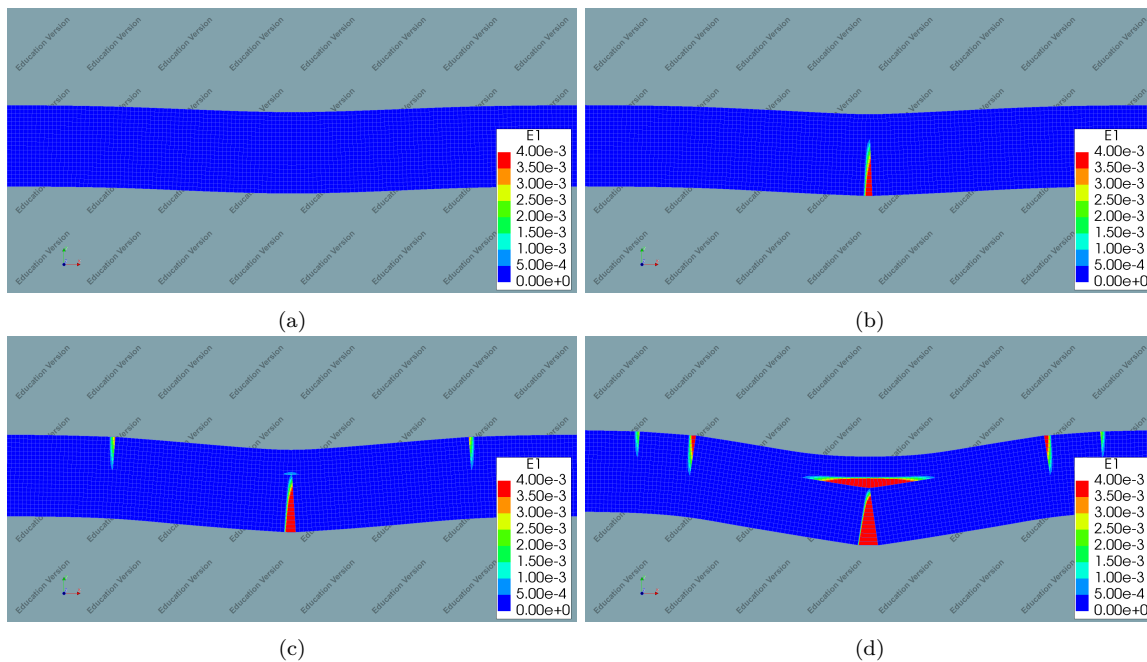

| (a) | (b) |
| (c) | (d) |

Figure 5.2: Crack patterns for average masonry at (a) Onset of cracking, (b) Vertical crack in the middle, (c) Vertical cracks above the inflexion point and (d) Horizontal crack in the middle of the structure.

Considering these types of cracks, four main load levels are described which are shown in Figure 5.3:

- Point A: Onset of cracking
- Point B: Abruptly formation of the vertical crack in the middle of the structure.
- Point C: Onset of formation of the vertical cracks above the inflexion points.
- Point D: Onset of formation of the horizontal crack in the middle of the structure.
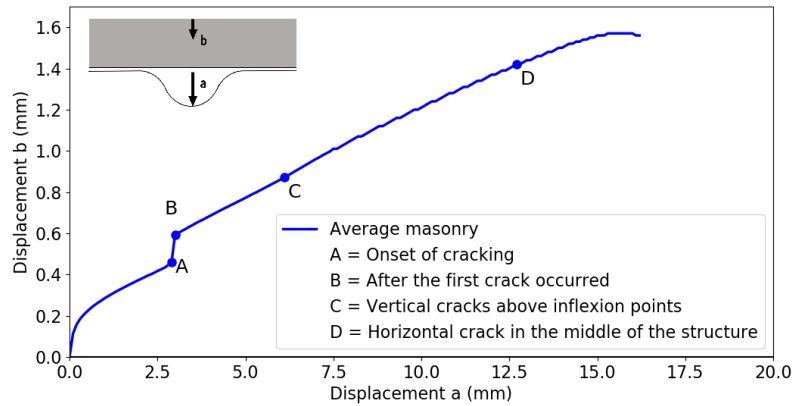
Figure 5.3: *Displacement b* as a function of the imposed *displacement a.*

5.2 Influence of material properties

This section is about the difference in material properties regarding the tension regime. For this analysis, three types of masonry are defined: weak, average and strong masonry. These three types of masonry have different values for Young's modulus, tensile strength and fracture energy. The influence of these parameters can be found in the stress strain relation in Figure 5.4. Young's modulus influences the steepness of the linear part and the tensile strength defines the maximum of the graph. The fracture energy influences the area and has therefore an effect on the second part of the graph. An overview of the values for these properties can be found in Table 5.1.
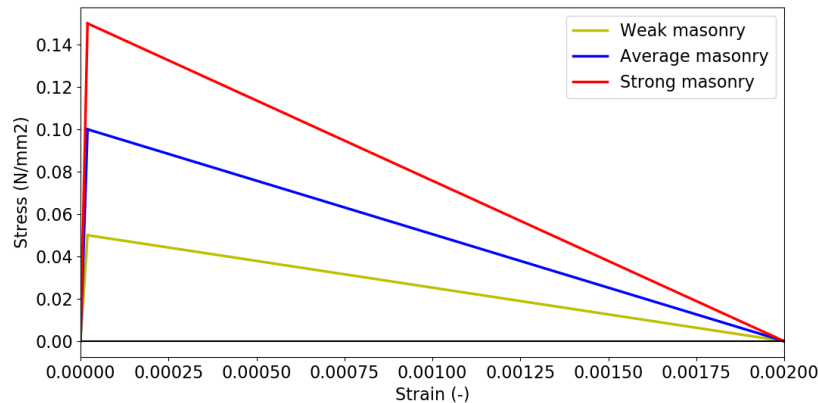
Figure 5.4: Stress-strain relationship of the three types of masonry: Weak, Average and Strong.

Material parameter	Weak masonry	Average masonry	Strong masonry
Young's modulus	2500 N/mm^2	5000 N/mm^2	7500 N/mm^2
Tensile strength	0.05 N/mm^2	0.1 N/mm^2	0.15 N/mm^2
Fracture energy	0.005 N/mm	0.01 N/mm	0.015 N/mm

Table 5.1: Overview of the parameters in the tension regime for the three predefined types of masonry.

The crack patterns happen for all the three materials in the same way: first, a vertical crack in the middle occurs due to tension in the bottom of the structure. At the end of this crack, a horizontal crack appears. Two vertical cracks can be noticed above the inflexion points of the imposed *displacement a.* These cracks happen at the same time since the situation is symmetric. Figure 5.5 shows *displacement b* at the top of the structure as a function of imposed *displacement a* for the three defined material types. In this graph, the four load levels can be distinguished.

The situation at these points is compared with each other in this analysis. The comparison of the crack patterns for the three types of masonry can be found in Appendix A. Although the crack patterns are the same, some variations are noticed. Figure 5.5 shows that the cracks do not appear at the same imposed deformations for the different materials. This means that the type of masonry has an influence.
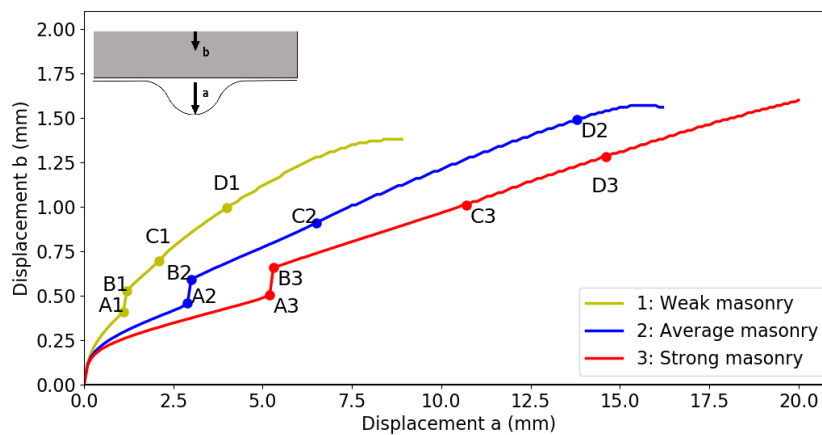
Figure 5.5: *Displacement b* as a function of *displacement a*: variations for weak, average and strong masonry at the predefined points A, B, C and D.

The onset of cracking is the situation where no cracks have appeared yet, but some evidence for future cracking can be found. Tension is observed at the bottom of the structure. This point is denoted as point A in the graph. Figure 5.6a shows the imposed deformation (*deformation a*) as a function of the ratio of the foundation defect (L_s/L) at the onset of cracking. For larger values of Ls (and Ls/L ratio), the onset of cracking occurs for a lower maximum imposed displacement. For stronger masonry, a larger maximum imposed displacement at the foundation is needed to trigger cracking with respect to weaker masonry; this difference decreases when the failure length of the foundation increases. How this effect can be observed at the top of the structure is presented in Figure 5.6b. The differences between the materials are not that large for small values of L_s. For all three masonry types, the same trend can be seen: an increase of the deformation length L_s increases *deformation b.* For the weak masonry, a sort of plateau is spotted in the graph. This plateau can be the consequence of a different type of crack pattern in the masonry. Where the vertical crack occurs in only one element, the crack occurs in two elements for values of L_s/L between 0.4 and 0.55. This can influence the deformation on top of the structure.

The second situation is at the moment the first crack is fully developed. This crack occurs vertically in the middle of the structure from the bottom to the top. An illustration of this crack is given in Appendix A. The figures show that the crack patterns for the three masonry types are almost the same. Figure 5.6c shows the *displacement a*, plotted against the length of the foundation defect, indicated by L_s/L at the moment the vertical crack appeared. The graph shows that a more significant length of the foundation defect (L_s) is needed to let the structure crack at a smaller deformation a. In other words, when the length of the imposed deformation is increased, the structure cracks at an earlier stage. The

graph also presents the three types of masonry and differences can be noticed. The structure with strong masonry, given a particular length for the foundation defect, shows the first crack at a more significant deformation a than the structure with weak masonry. Nevertheless, the differences decrease when L_s is increased. In Figure 5.6d, the deformation b at the top of the structure is plotted against the length of the foundation defect (Ls) for the three types of masonry. For small values of Ls, deformation b is also relatively small, while the imposed deformation is relatively large. At the moment the first crack occurs, at a large imposed displacement, the top of the masonry has not deformed that much. The bigger the length of the foundation defect gets, the more the structure itself deform. The crack occurs at an earlier stage. Also, the types of masonry show different behaviour in this graph. This is especially noticeable for large values of Ls. Stronger masonry deforms more before the first vertical crack can be noticed. This can be seen in Figure 5.6d, where the red line, indicating the strong masonry, is for all values of L_s above the blue and yellow line. An increase of L_s results in bigger differences between the graphs. For small values of Ls, the deformation on top of the structure is very small, and also the differences between the masonry types are very small.

The third situation is the moment that two vertical cracks occur above the inflexion point. These cracks start at the top of the structure and develop downwards. The cracks do not always occur abruptly; sometimes, they develop more slowly. Therefore, the moment the ultimate strain is reached at some point in the graph is taken as a reference. Figure 5.6e shows the imposed deformation A plotted against L_s/L. This graph shows significant similarities with the graph at the moment that the first crack occurs. The main difference is that the vertical cracks occur at a later moment. The graph also shows that for a value of $L_s/L = 0.2$, strong masonry does not show vertical cracks above the inflexion points, while the weak and average masonry do. Looking at Figure 5.6f, the occurrence of the two vertical cracks happens more or less at the same time. Small differences can be noticed for small values of Ls. For larger values, the differences seem to be smaller. This has to do with the development of these cracks. For lower values of L_s, the cracks develop more slowly than for larger values, where these cracks occur more abruptly.

The horizontal crack occurs in the middle of the structure, on top of the middle vertical crack. Like the vertical cracks above the inflexion points, the reference of the horizontal crack is chosen to be at the moment the crack starts to develop. Figure 5.6g and Figure 5.6h show deformations a and b as functions of the length of the foundation defect. As can be seen in the graphs, the strong masonry needs a larger displacement of both a and b to occur, before a horizontal crack appears.
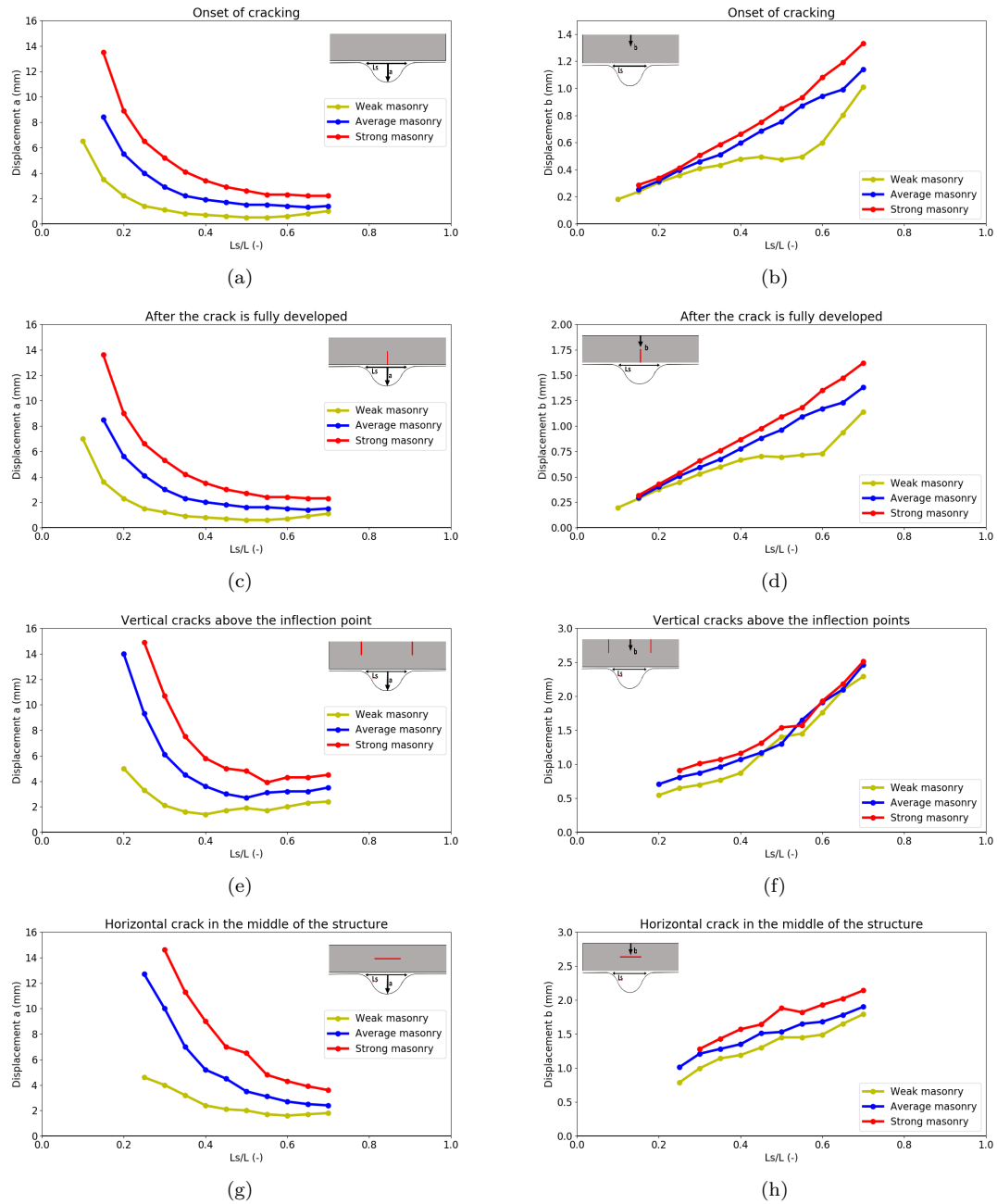
Figure 5.6: Displacements a and b plotted as a function of L$_s$/L, displayed at the moment of (a)-(b) onset of cracking, (c)-(d) Vertical crack in the middle of the structure occurs, (e)-(f) Vertical cracks occur above the inflexion points, (g)-(h) Horizontal crack occurs in the middle of the structure.

Another thing that can be noticed from the graphs is that the horizontal crack does not always appear after the vertical cracks above the inflexion points appeared. For particular values of L$_s$/L, the horizontal crack occurs earlier. This can be seen in Figure 5.7, where the intersection of the graphs with the corresponding colour indicates that there is a change in which cracks occurs earlier. It can be seen that large values of L$_s$ will result in a horizontal crack in an earlier stage than vertical cracks.
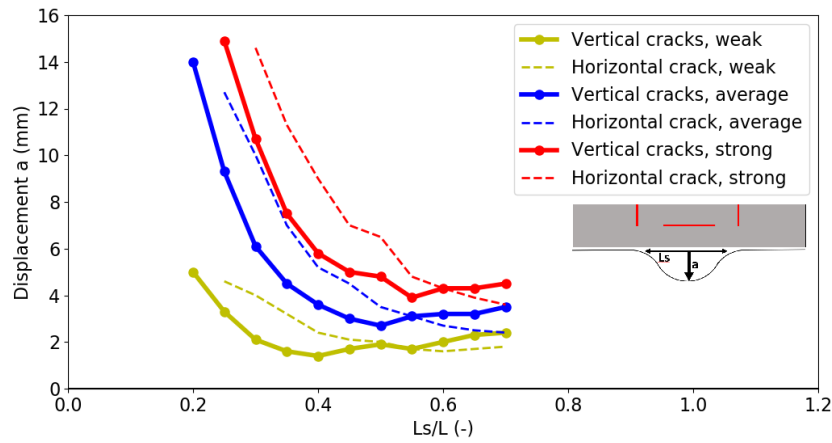
Figure 5.7: *Displacement a* as a function of L_s/L: comparison between the occurrence of vertical cracks above the inflexion points and horizontal crack in the middle of the structure.

5.3   Influence of lateral constraints

The influence of the horizontal constraints at the edges of the structure is determined in this section. Two analyses are performed, one analysis with horizontal constraints at the edges, which is the same case as the base case, and one analysis without the constraints. This means that the structure can move freely in the horizontal x-direction.
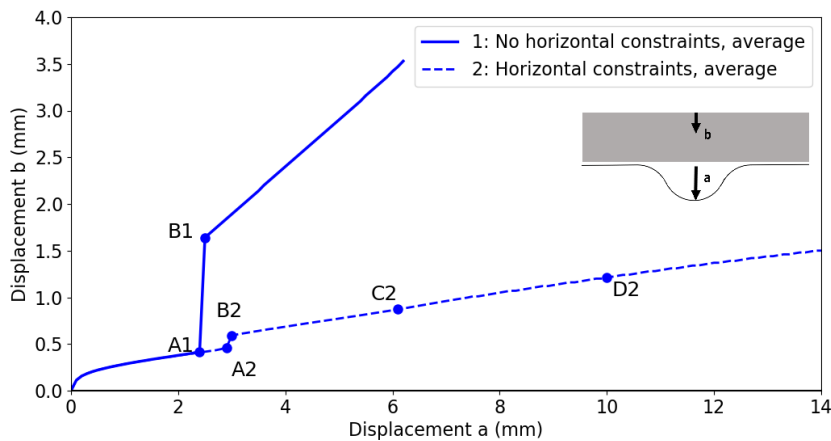
Figure 5.8: Comparison between laterally constrained and laterally free quay wall section (average material properties): *Displacement a* as a function of *displacement b*.

Figure 5.8 shows the comparison of the cases with and without lateral constraints in terms of displacements. In this graph, the imposed *displacement a* is plotted on the x-axis, and *displacement b* on top of the structure is plotted on the y-axis. The first part of the graph is almost precisely the same; however, after an imposed displacement of 2.4 millimetres, the graph shows some big differences between the two analyses. The first real difference can be seen at the onset of cracking, indicated by point A. The onset of cracking occurs at an earlier stage for the situation without the lateral constraints, which is also observed in Figure 5.9a. Secondly, it also shows that the displacement on top of the structure

increases significantly for the cases without horizontal constraints. This is especially noticeable after the first crack, indicated by point B. After this moment, larger deformations on top of the structure can be observed for the structure without the horizontal constraints.
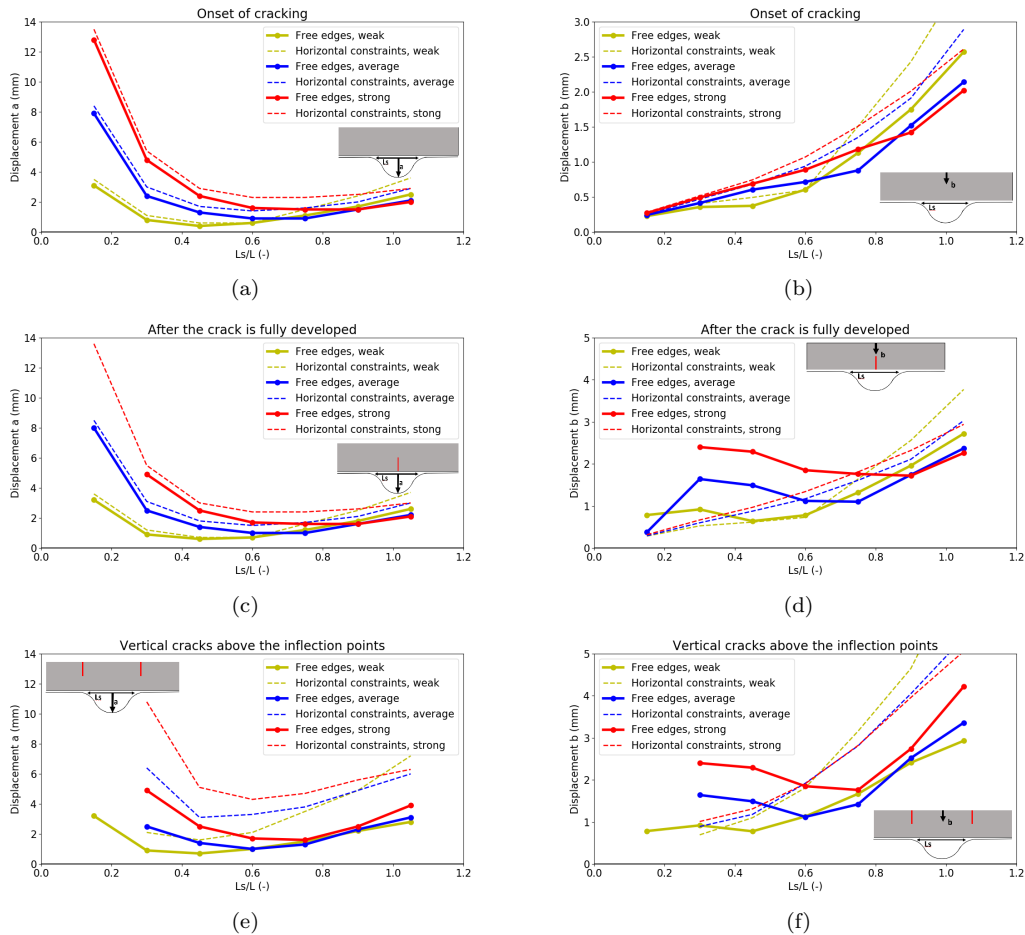

Figure 5.9: Displacements a and b plotted as a function of Ls/L, displayed at the moment of (a)-(b) onset of cracking,(c)-(d) Vertical crack in the middle of the structure occurs, (e)-(f) Vertical cracks occur above the inflexion points,(g)-(h) Horizontal crack occurs in the middle of the structure, cases with and without lateral constraints.

Figure 5.9c and Figure 5.9d show graphs of *displacement a* and *b* respectively, as a function of L_s/L, for the cases with and without the horizontal constraints. It can be noticed that not many differences can be noted for the imposed *displacement a*, except that the first fully developed crack occurs at a slightly larger displacement for the case without horizontal constraints. Nevertheless, the displacement on top of the quay wall structure shows some large differences between the two cases. The structure with horizontal constraints shows an increase in deformation for an extension of the foundation defect. The structure without the horizontal constraint does not show a clear relation between these parameters. This can be explained by looking at the crack pattern development of various cases (Figure 5.10). This illustration shows the crack pattern, for $L_s/L = 0.15$, 0.30 and 0.75, respectively, at the moment the first fully developed crack appears in the structure. The crack patterns for the cases with of $L_s/L = 0.30$, (and the same for 0.45 and 0.60) show that cracking occurs not only in the middle but also above the inflexion points of the imposed deformation simultaneously. This is also noticed in Figure 5.9e and Figure 5.9f.

From the moment that L_s/L becomes larger than 0.6, the vertical cracks above the inflexion points start to appear at a later stage than the central vertical crack. The graphs also show that a smaller displacement of a and b is needed for the structure to crack at these points. The structure with the horizontal constraints shows eventually horizontal cracking in the middle of the structure. Nevertheless, in the case without these constraints, no horizontal cracking occurs.
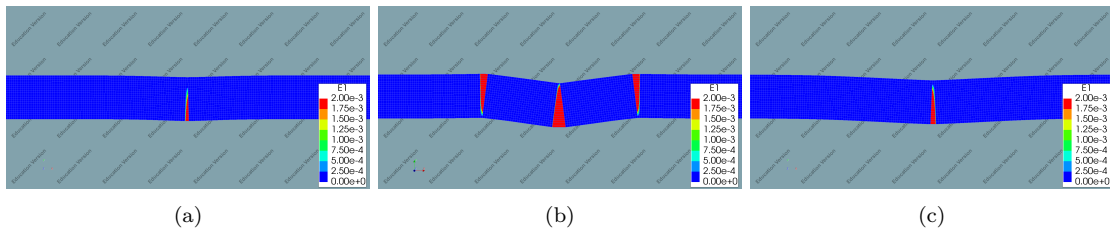

(a) (b) (c)

Figure 5.10: Illustrations of the moment the first crack occurs for a L_s/L = (a) 0.15, (b) 0.30 and (c) 0.75.

The absence of the horizontal crack can be explained by the influence of the lateral constraints since these constraints cause Compressive Membrane Action to occur. Figure 5.11 shows the in-plane principal stresses in the structure for the cases with and without the lateral constraints. In the case with the lateral constraints, the a compressive arch is visible in the structure after the first cracks occurs, while the case without the constraints does not show this arch. An extra region with tensile stresses occurs in the middle of the structure causing a horizontal crack. Compressive Membrane Action also restricts the structure to have large deformation and therefore gives the quay wall extra strength.

(a) (b)

(c) (d)

(e) (f)
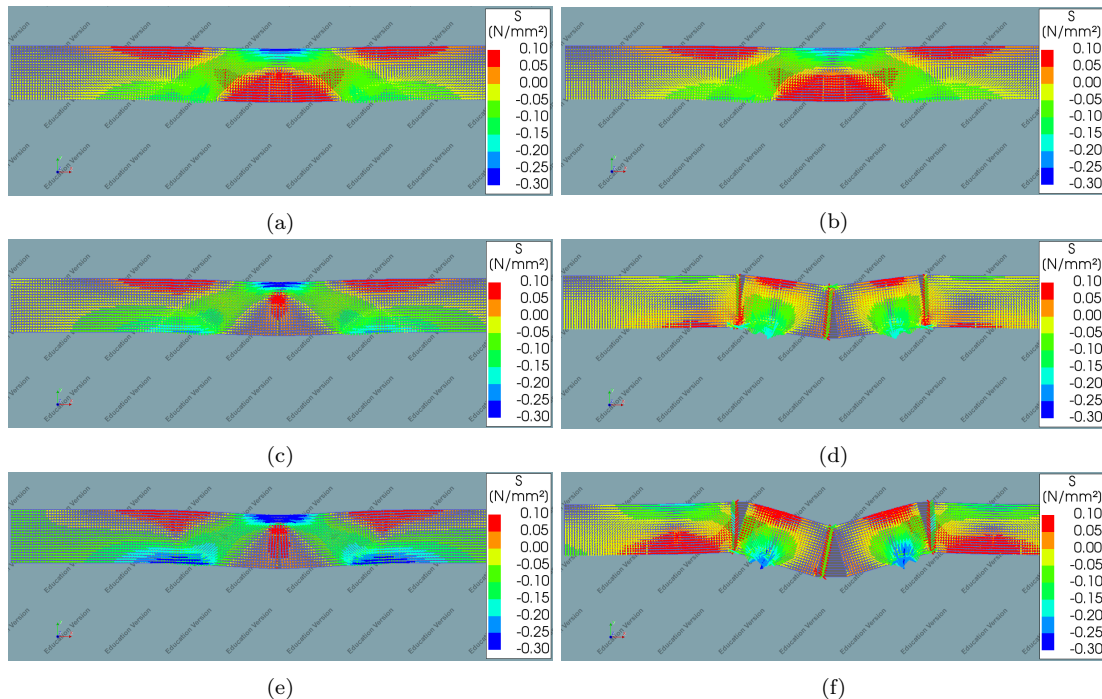
Figure 5.11: In-plane principal stresses of the laterally constrained (a)-(c)-(e) and laterally free (b)-(d)-(f) quay wall section (average material properties): (a)-(b) load level A; (c)-(d) load level B; (e)-(f) imposed *displacement* $a = 6.3$ mm.

5.4   Influence of the location of the foundation defects

In this section, an asymmetric settlement profile is considered to evaluate the influence of the location of the foundation defect (Figure 5.12). Like the previous sections, four situations are discussed: Onset of cracking, bottom vertical crack, vertical cracks above inflexion points and the horizontal crack. Since the position of the foundation defect is moved to the lateral end, only one inflexion point in the Gauss curve beneath the structure is present, resulting in only one vertical crack from the top of the structure. Also, the central vertical crack and the horizontal crack do not longer occur in the middle of the structure, but just above the imposed *displacement a*.
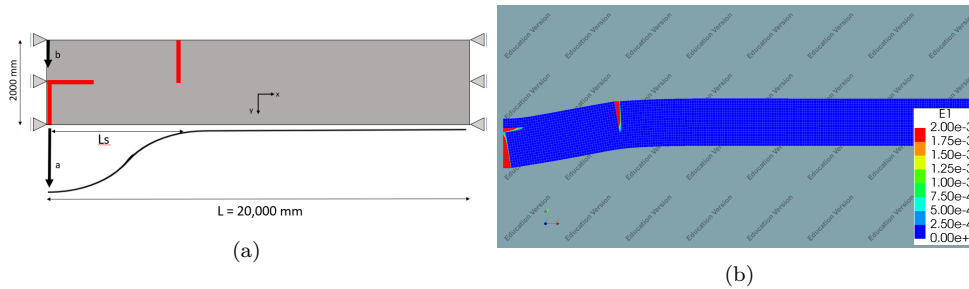
Figure 5.12: Illustrations of the situation where the foundation defect is moved to the left: (a) mechanical model and (b) Plot of the principal strain in the finite element model.

Figure 5.13 shows that material properties of the masonry effects the structural behaviour. Like the previous analyses, the strong masonry cracks after larger imposed displacements. The weak masonry cracks after relative small imposed displacements.
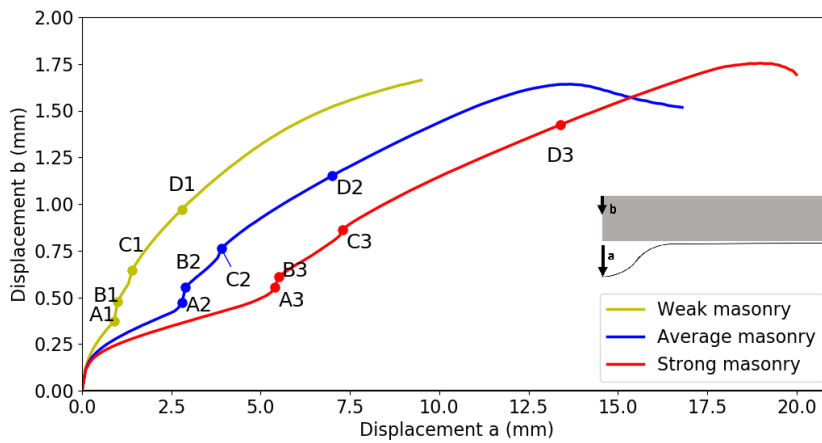
Figure 5.13

Figure 5.14a and figure 5.14d show the imposed displacement as a function of the length of the foundation defect. These four graphs show that stronger masonry needs a more significant displacement to show crack patterns than weaker masonry. Looking at Figure 5.14e, the absence of the first points for average and strong masonry stands out. In the analysis, no vertical crack above the inflexion point occurred for these types of masonry. This also holds for the horizontal crack in Figure 5.14g, wherein all three types of masonry no horizontal crack appeared.
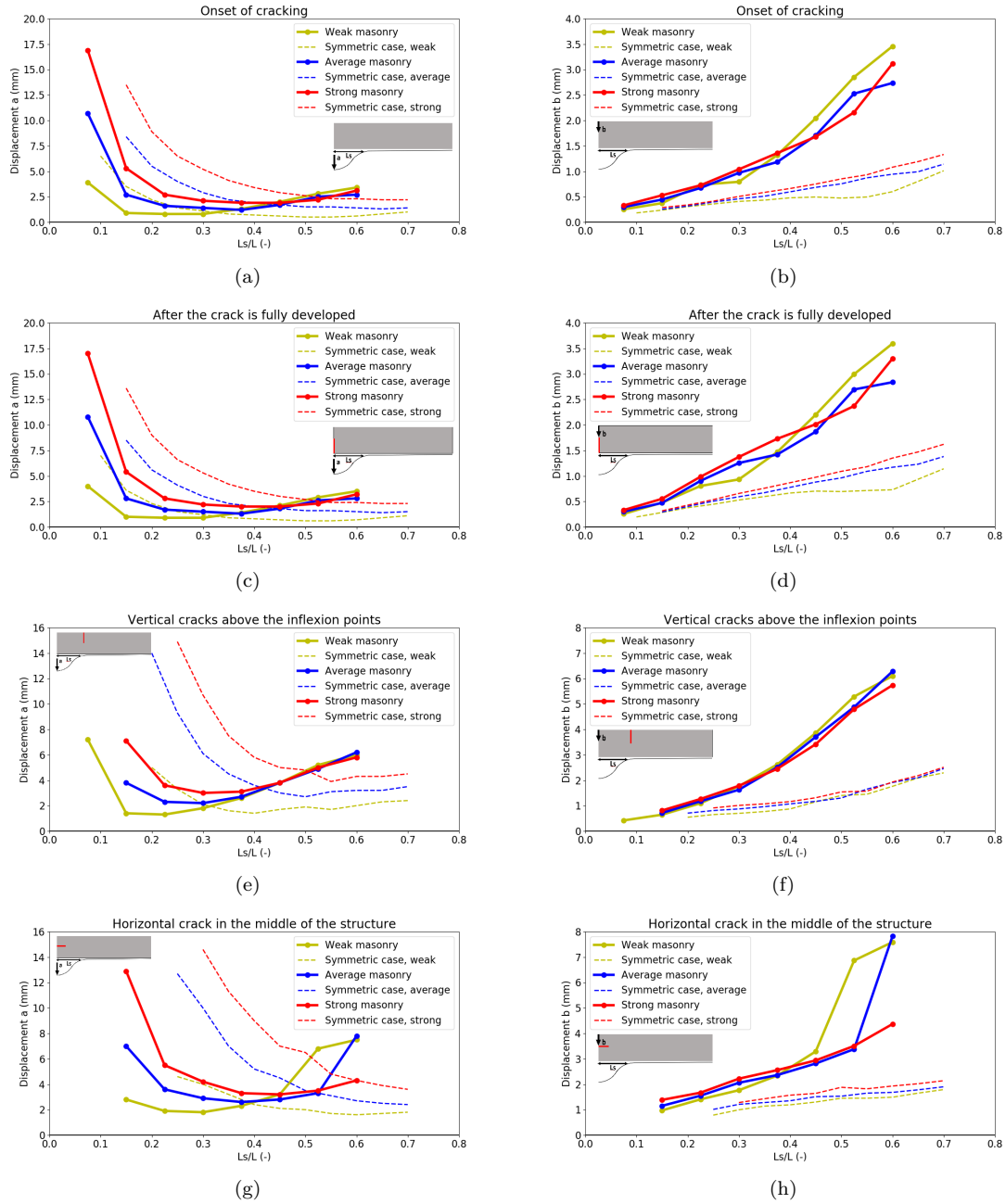
Figure 5.14: Displacements a and b plotted as a function of L_s/L, displayed at the moment of (a)-(b) onset of cracking, (c)-(d) Vertical crack in the middle of the structure occurs, (e)-(f) Vertical cracks occur above the inflexion points (g)-(h) The horizontal crack occurs in the middle of the structure. Case where the foundation defect is positioned at the left side of the structure:

As Figure 5.14 shows, the displacement on top of the structure is very much the same for all types of masonry at the given crack patterns and small lengths of foundation defects. More differences can be noticed after $L_s/L = 0.4$. The largest difference can be seen in Figure 5.14h, where the graph shows a large difference for the weak and average masonry with respect to the strong masonry at $L_s/L \geq 0.5$. This can be due to the difference in crack patterns in the structure. The first vertical crack occurs for the strong masonry at the far left between the masonry and the support, while the weak masonry shows a crack at a small distance from the side (Figure 5.15).

(a)   (b)
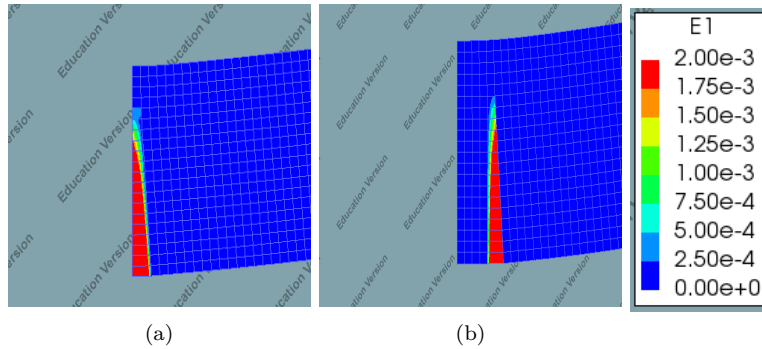
Figure 5.15: Crack pattern obtained from analyses with an asymmetric settlement profile with Ls/L = 0.5: (a) strong masonry (a = 3 mm); (b) weak masonry (a = 3 mm).

5.5   Influence of orthotropic behaviour of masonry

The previous analyses are all performed with isotropic material properties for the masonry. In this section, an orthotropic material model is considered. Table 5.2 shows that different values for Young's modulus and the tensile strength are given along the x- and the y-direction, while all the other properties are the same in the two directions. For Young's modulus, the same properties adopted for the isotropic material model are used along the y-direction, while The Young's modulus along the x-direction is decreased by 50%. For the tensile strength, the same value adopted from the isotropic material model is used for the x-direction, while the value for the y-direction is three times smaller (Schreppers et al., 2016).

Material parameter	Weak masonry	Average masonry	Strong masonry
Young's modulus	$E_x = 1250$ N/mm^2 $E_y = 2500$ N/mm^2	$E_x = 2500$ N/mm^2 $E_y = 5000$ N/mm^2	$E_x = 3750$ N/mm^2 $E_y = 7500$ N/mm^2
Tensile strength	$f_{tx} = 0.05$ N/mm^2 $f_{ty} = 0.0167$ N/mm^2	$f_{tx} = 0.1$ N/mm^2 $f_{ty} = 0.033$ N/mm^2	$f_{tx} = 0.15$ N/mm^2 $f_{ty} = 0.05$ N/mm^2
Fracture energy	$G_{ft} = 0.005$ N/mm	$G_{ft} = 0.01$ N/mm	$G_{ft} = 0.015$ N/mm
Compressive strength Compressive fracture energy	$f_c = 8.5$ N/mm^2 $G_{fc} = 20$ N/mm	$f_c = 8.5$ N/mm^2 $G_{fc} = 20$ N/mm	$f_c = 8.5$ N/mm^2 $G_{fc} = 20$ N/mm
Shear modulus Shear strength Shear fracture energy Cohesion	$G = 2000$ N/mm^2 $f_s = 0.35$ N/mm^2 $G_{fs} = 0.1$ N/mm $c = 0.3$ N/mm^2	$G = 2000$ N/mm^2 $f_s = 0.35$ N/mm^2 $G_{fs} = 0.1$ N/mm $c = 0.3$ N/mm^2	$G = 2000$ N/mm^2 $f_s = 0.35$ N/mm^2 $G_{fs} = 0.1$ N/mm $c = 0.3$ N/mm^2

Table 5.2: Overview of the parameters in the tension regime for the three predefined types of orthotropic masonry.

Figure 5.16 shows the comparison between the analyses obtained with the orthotropic and the isotropic material model for masonry. Comparing the analysis with the orthotropic and isotropic material model,

the vertical crack in the middle of the structure occurs for larger imposed displacement in the case of the orthotropic material model. This happens as a result of the difference in Young's modulus in the x-direction. Since this value is decreased by 50% and the tensile strength in x-direction remained the same, the cracking strain results in two times larger in the case of the orthotropic material model. In the orthotropic model, the horizontal crack (D1) occurs simultaneously to the formation of the vertical crack (B1) in the middle of the structure. This is because the tensile strength in the y-direction is decreased significantly, resulting in the masonry to crack in the horizontal direction at an earlier stage. This can also be seen in the crack patterns in Figure 5.17. Additionally, the analysis adopting the orthotropic material model does not converge anymore after the imposed displacement of 5.4 mm. It is possible that the structure undergoes extensive damage at this point.
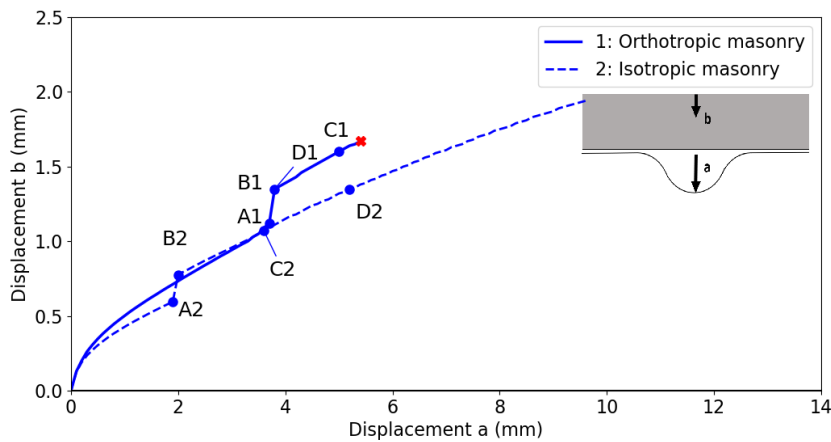
Figure 5.16: *Displacement b* as a function of *displacement a*: differences between orthotropic average masonry and isotropic average masonry.
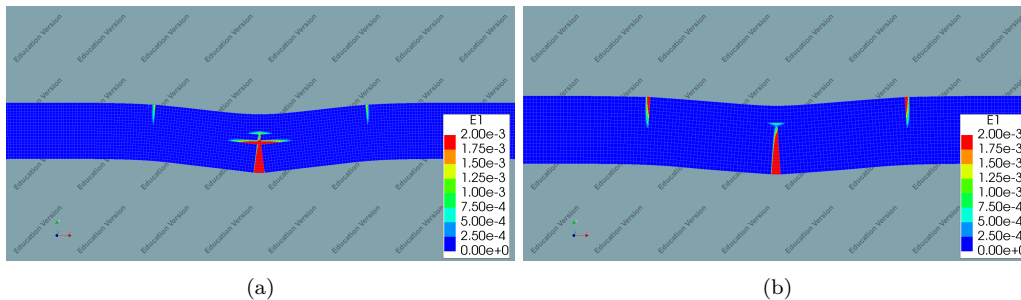
Figure 5.17: The crack patterns of two cases: (a) orthotropic material model; (b) isotropic material model

Figure 5.18 shows the another difference compared with the isotropic masonry: a new horizontal crack appears at the bottom of the structure. The dependency of this crack pattern to the length of the foundation defect is plotted in Figure 5.19a and Figure 5.19b. These graphs show that there is not a clear correlation between the length of the foundation defect (L_s) and imposed *displacement a*. Nevertheless, the influence of the properties of the masonry is evident. Strong masonry needs a far larger displacement to let the horizontal crack occur than weaker masonry. The influence of the masonry can not be seen on top of the structure, where the displacement is similar for all three materials, given a small length of the foundation defect. Differences can be noticed for larger values. However, when L_s/L reaches values over 0.25, the crack does not occur anymore.
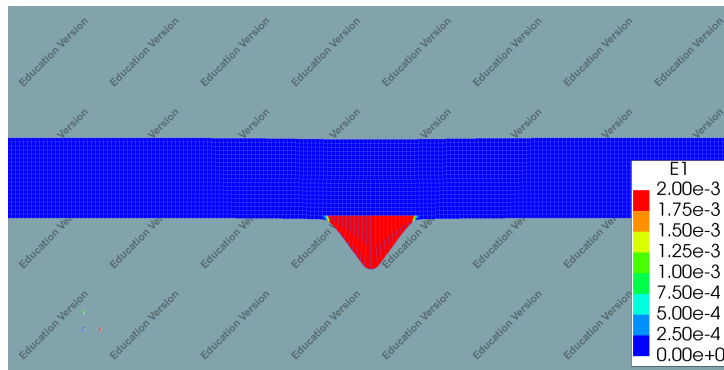
Figure 5.18: Illustration of the quay wall where cracking occurs at the bottom of the structure.

Like the isotropic masonry, a vertical crack occurs in the middle of the structure. Nevertheless, compared with the isotropic masonry, this crack pattern occurs at a later stage. The same trend can be seen in Figure 5.19d. The top of the structure has larger deformations for the three types of masonry when the vertical crack occurs. Also, the influence of the type of masonry can be seen. Like in all the previous analyses, the stronger masonry needs larger displacements of the foundation to let cracking occur.

The results for the vertical cracks above the inflexion point and the horizontal crack in the middle of the structure look a bit arbitrary. The main reason for this is because the analysis shows different crack patterns and crack developments for different values of L_s. For example, the lower peak for the average masonry at $L_s/L = 0.6$ can be explained by the fact that first, a vertical crack develops above the right inflexion point and then at the left side. The horizontal crack in the middle of the structure seemed to occur for smaller imposed displacements, according to Figure 5.16. The same is confirmed by Figure 5.19g, especially for small values of L_s. Nevertheless, the difference with the isotropic material decreases with an increase of L_s.

Compared with the isotropic material model, the orthotropic material model gives some problems with convergence at an early stage. It is possible that the structure undergoes extensive damage at the moment the convergence criteria are not met, which will indicate that the structure or the analysis is less stable with orthogonal properties.
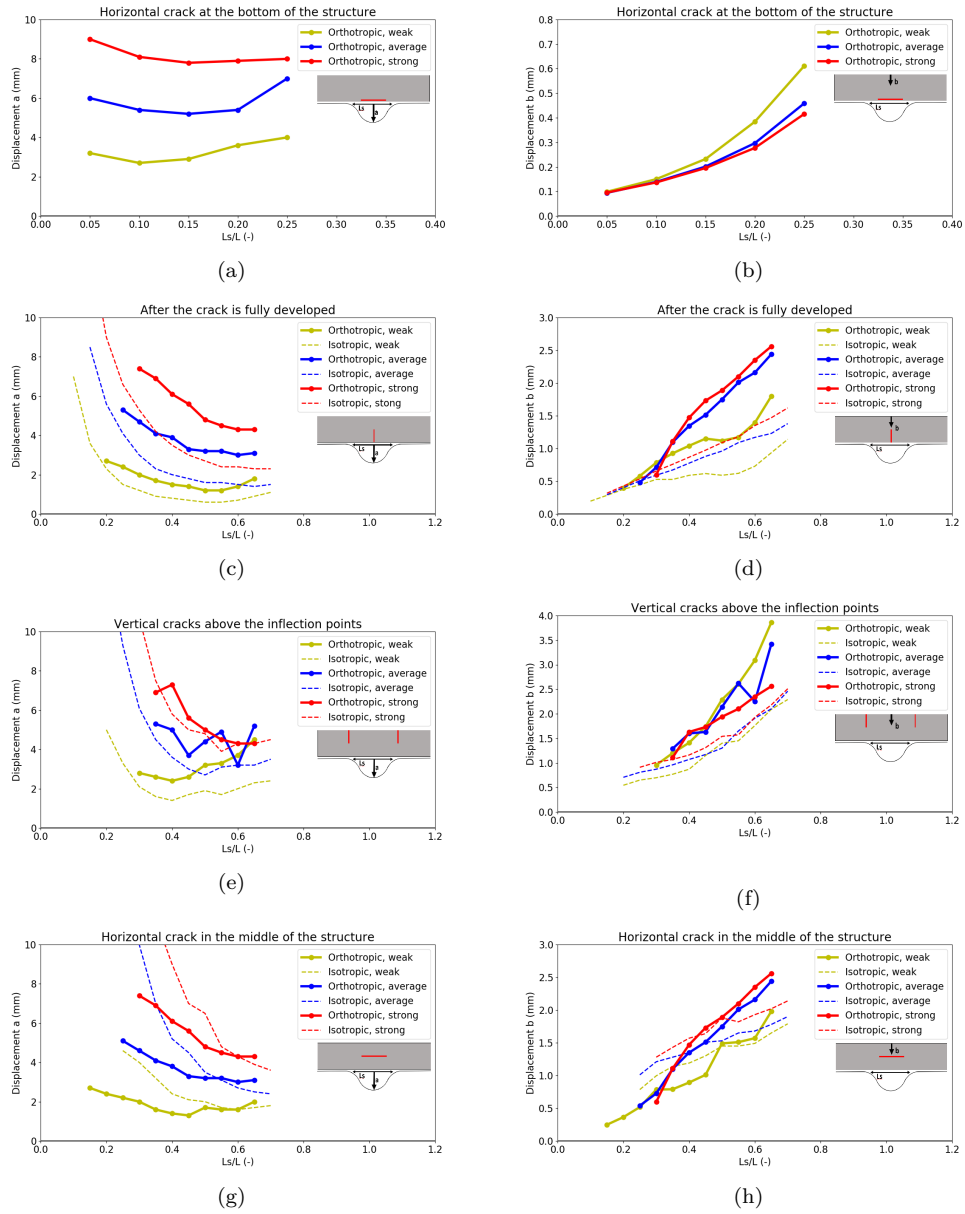
Figure 5.19: Displacements a and b plotted as a function of Ls/L, displayed at the moment of (a)-(b) Horizontal cracking at the bottom of the structure, (c)-(d) Vertical crack in the middle of the structure occurs, (e)-(f) Vertical cracks occur above the inflexion points (g)-(h) Horizontal crack occurs in the middle of the structure: The orthotropic case is compared with the isotropic case.

6

Discussion

In this chapter, the results are discussed to answer the main research question: How to recognize foundation defects via masonry damage patterns in quay walls? The influence of the type of masonry, the horizontal constraints, the defects in the pile foundation and orthotropic behaviour are discussed. Thereafter, the damage patterns indicating foundation defects are discussed. Eventually, the limitations of the analyses were given.

6.1 Influence of material properties

To answer the question: "What is the influence of the material properties of masonry on the development of damage patterns?" an analysis with a variation in material parameters is performed. The data from this analysis suggests that there is an influence of the material parameters in the tension regime. The stronger the masonry, the larger the imposed displacement must be to let cracking occur.

In line with the theory, strong masonry cracks at higher values for the imposed displacement than weak masonry. This was also confirmed by Figure 5.6. Strong masonry has a higher Young's modulus and tensile strength. The higher tensile strength ensures an extra capacity for tension, resulting in the masonry to crack for higher levels of tension forces and therefore larger strains. The increase of Young's modulus ensures the masonry to act more brittle. A smaller strain is needed to crack the masonry. The combination of both provided the masonry to crack at the same strain, but at higher tensile strengths. This means that cracking occurs at a later stage or at a larger imposed displacement. The results of the analysis show the same conclusion: stronger masonry acts stiffer than weak masonry and cracks for larger imposed displacements. This was also presented by Figure 5.6. Stronger masonry needs larger imposed displacements to initiate a particular crack pattern in the structure.

Since this analysis is done with the isotropic material model, the reactions in the x- and y-direction differ from the orthotropic material model. The differences between these analyses are discussed in Section 6.4. Also, the choice for the mesh size and load step size is affecting the results. The ultimate strain is dependent on the mesh size and load step size, which means that other values could be obtained from the analyses when these parameters are altered. Nevertheless, comparison between analyses is possible, since the values are kept the same for all analyses. Although no absolute values can be given, the results can be interpreted qualitatively.

6.2 Influence of lateral constraints

To answer the question: "Which boundary conditions apply to a typical Amsterdam quay wall and what is their influence?" an analysis with variation in boundary conditions at the lateral ends is performed. It was assumed that the lateral ends were not able to move in the lateral direction, for the reason another quay wall section is situated there. It is also assumed that the lateral ends can move freely in the vertical direction while some friction forces could be present between the sections, decreasing the ability of the structure to move vertically. Typically, the pile foundation restricts vertical movement, but because analyses are performed for failing pile foundations, this can not be assumed for any case.

Figure 5.9 shows that the horizontal constraints at the lateral ends of the structure ensure that cracking occurs for larger settlement loads than a structure without these constraints. The situation without the constraints shows a large displacement on top of the structure and can be observed when the first vertical crack is fully developed. This statements only holds for the ratio $L_s/L \leq 0.6$. When the foundation defect length is larger than that ratio, the displacement on top of the structure due to the first vertical crack will be relatively smaller. The analysis also shows that vertical cracks above the inflexion point of the imposed deformation occur simultaneously or shortly after the first vertical crack is initiated. The horizontal constraints ensure these cracks to occur at larger imposed displacements. The horizontal crack does not appear at all when the lateral ends of the structure are horizontally free. This can be explained by the influence of Compressive Membrane Action (CMA) in the structure. Due to CMA, an compressive arch is formed in the quay wall causing tensile stresses in the vertical direction. These tensile stresses result in a horizontal crack.

The analyses show that horizontal constraints alter the structural behaviour of the quay wall section. When the structure is not able to move in the horizontal x-direction due to the presence of other section, the structure has a small amount of extra strength. This amount of extra stress can be related with the compressive arch formed by the effect of Compressive Membrane Action. Deformations at the top of the structure are significantly smaller when the structure is horizontally fixed. Moreover, multiple vertical cracks from the top of the structure allow the structure to move with the imposed deformation indicating failure. Secondly, the horizontal constraints provide the structure with a new horizontal crack. This horizontal crack can appear due to the arching force in the masonry. When this force is absent, the structure is allowed to move as a whole in the vertical direction. Due to this force, some parts will not move along the y-axis, causing a tension force in the bed joints, which will cause a horizontal crack.

The assumption is made that the structure is completely fixed in the horizontal direction, while the lateral ends do not have to be fully fixed. It is possible that some spacing is present, and thus some movement is possible.

6.3 Influence of foundation defects

In this section, the influence of pile failure is discussed. Analyses have been performed regarding the number of piles in a pile group failing (the length of the foundation defect) and the location of the failure. The question to be answered is: "How do foundation defects alter the static system and structural response of the quay wall?". First of all, the analysis demonstrates a correlation between the length of the foundation defect, indicated by L_s and the imposed deformation, indicated by *deformation a*. An increase in the length of the foundation defect results in a decrease of the imposed deformation needed to let a specific crack occur. The second aspect is the location of the foundation defect. Analyses have been performed for the symmetric case and asymmetric case. In the symmetric case, the foundation defect was situated in the middle, while for the asymmetric case, the foundation defect was positioned underneath the left edge of the structure. The analysis shows that asymmetry does not alter the results very much in terms of the type of cracks. For both the symmetric as the asymmetric situation, a vertical crack,

horizontal crack and a crack above the inflexion point occurred. For larger values of L_s, the structure shows different crack patterns, which influences the results. The vertical crack from the bottom of the structure occurs at a small distance from the side (Figure 5.15).

The ratio L_s/L has a significant influence in the reaction of the masonry on top of it. When this value increases, larger deformations at the top of the structure can be observed. A much smaller imposed deformation is needed when this ration is larger, meaning that the piles only need to lose a small amount of strength. In other words, the piles do not have to fail completely to have an impact on the structural behaviour. When the ratio L_s/L is relatively small, the impact is less substantial than the impact of larger ratios with small imposed deformations. This can also be observed from above the waterline. The deformation on top of the structure is smaller for smaller values of L_s.

The reliability of this data is limited by the assumption made for friction at the lateral ends. It is assumed that no vertical restrictions are present at these places, which enables the structure to move freely in the vertical direction. When this movement is restricted by friction forces present between the quay wall sections, the response of the structure can be different, as discussed in Section 6.2.

6.4   Influence of orthotropic behaviour

All the analyses were done using the isotropic material model. In this section, the influence of orthotropic behaviour is discussed. The analyses show that the orthotropic material model alters the crack pattern for small values of L_s since a new horizontal crack appears at the bottom of the structure. Secondly, it is observed that the initial vertical crack occurs at a larger imposed displacement than for isotropic masonry. On the other hand, the horizontal crack occurs at a smaller imposed displacement. The difference with the isotropic material decreases with an increase of the length of the foundation defect. It was also observed that the model is not able to find convergence at an earlier stage of the analyses.

The analyses show a new horizontal crack on the bottom of the structure when using the orthotropic material model. Moreover, the already observed horizontal crack at the middle of the structure appears for smaller imposed displacements compared with the isotropic masonry. The fact that horizontal cracking seems to occur more frequently is in line with the expectations since the tensile strength of the bed joints is decreased. The fact that a vertical crack occurs at a larger imposed displacement can be explained by the decrease of Young's modulus in the horizontal x-direction. This decrease ensures the vertical crack to appear at larger strains, leading to more ductile behaviour in this direction. Lastly, the model does not find convergence at a relatively early stage for many of the analyses of the orthotropic material. It could be possible that the model does not find convergence anymore, because the structure undergoes extensive damage, leading to an imbalance of forces. It is observed that horizontal cracking occurred at an early stage as well, which could indicate a relation between this horizontal crack and the failure of the structure.

The fact that many analyses do not converge at an early stage should be investigated. It is possible that there is a relation between the occurrence of a horizontal crack and the failure of the structure. However, it is also possible that there is an imperfection in the model. A sensitivity analysis regarding the mesh size and load step size should be performed to investigate this topic.

6.5   Damage patterns indicating foundation defects

The crack patterns in the masonry and the deformation on top of the structure can be observed from above the water level as damage patterns. To answer the question: "How to recognize foundation defects via masonry damage patterns in quay walls?", these damage patterns are investigated to have a relation

with foundation defects. Figure 6.1 shows an overview of these damage patterns.
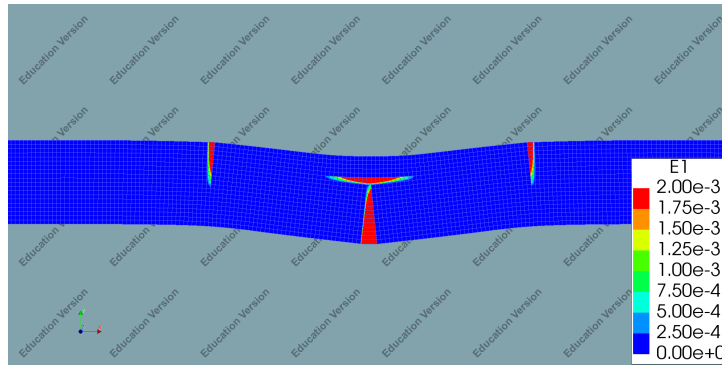

Figure 6.1: Illustration of the main crack patterns in a quay wall section.

Regarding the deformation on top of the structure, a large deformation on top of the structure can indicate that foundation defects are present. When a large deformation is observed, but no cracking occurred in the masonry, this can indicate that a large number of piles have lost strength and need to be investigated. When no cracking occurred yet, the properties of the masonry are probably strong. Nevertheless, high stresses can be present in the quay wall section, which can soon lead to cracking.

When a vertical crack is observed in combination with large displacements at the top of the structure, this is a good indication that one or more piles around this crack have defects. The tensile strength of the masonry at the bottom of the structure is exceeded, causing this crack. The type of masonry defines this tensile crack. When the masonry is very old and therefore very weak, it is possible that the crack occurred, without a large displacement noticeable at the top of the structure. At this situation, it is still possible that the strength of a large number of piles is exceeded. However, this vertical crack does not have to be visible from above the waterline since the crack originates from the bottom of the structure.

Vertical cracking, downwards from the top of the quay wall section can indicate foundation defects in the quay wall section. Nevertheless, the foundation defect is not expected to be beneath this crack, since this crack occurs above the inflexion points of the imposed displacement. In combination with a second vertical crack, this crack can indicate the boundary of the foundation defect.

A horizontal crack is a good indicator for foundation defects. The lateral constraints have a significant influence in the occurrence of the horizontal crack. Due to the constraints, a compressive arch is formed in the structure due to Compressive Membrane Action. This arch leads to tensile strength resulting in a horizontal crack. When a horizontal crack is observed, the foundation of the quay wall is probably in an advance stage of failure. The leading cause is probably that a large number of piles have (partially) failed.Mainly when a vertical crack is observed from to bottom of the structure to the horizontal crack, inspections should be performed.

In addition to the cracks described, some variations are also observed. For example, two vertical cracks in the middle were observed. This crack pattern was only detected for large values of L_s. In many cases, multiple vertical cracks, initiated from the top of the structure were observed. In most of the cases, these cracks appeared at extensive imposed deformations. These cracks were left out of the scope since other cracks occurred at an earlier stage.
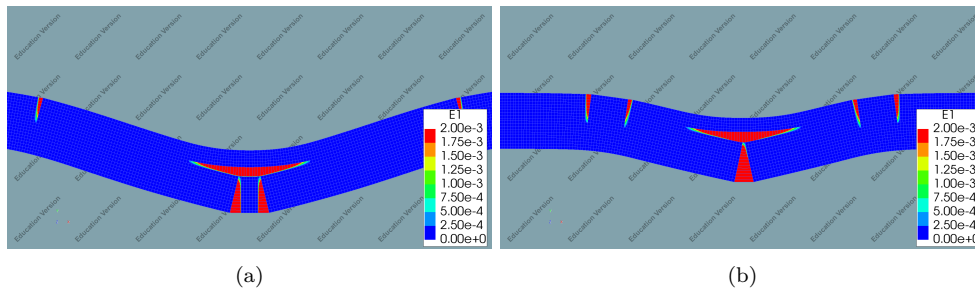
Figure 6.2: Illustrations of two examples with new crack patterns.

6.6 Limitations

In this section, the limitations of the study are discussed. The reliability of the data is impacted by the choices made regarding the model.

First of all, the influence of additional loads, such as traffic and trees, is not considered in the analysis. The background problem stated that an increase of load, due to traffic, for example, is one of the reasons that the load capacity of quay walls is exceeded. However, in the model, only the dead weight of the structure is considered, leaving the other loads out of scope. Additional loads on the structure can lead to different crack patterns. For example, the influence of trees can be modelled as point loads, which can contribute to other damage patterns. The same holds for loads out-of-plane, such as the pressure of the soil. Loading out-of-plane could lead to different crack patterns in the masonry or influence cracks to occur at a different moment.

The model is modelled in 2D, which means that a uniform cross-section and stress distribution over the thickness of the wall is considered. This assumption can be made, since only in-plane dead weight is considered in this research and out-of-plane loading is not. When taking out-of-plane loads into account, the stress distribution over the thickness can not be considered uniform anymore and cross-sectional influences become important.

A continuum damage model, in which masonry is assumed as a homogeneous material, was used. Therefore, the knowledge of the interaction between bricks and mortar were negligible. The use of such a model ensures a limited computational time. However, the use of a more detailed model in which the bricks, joints and their interfaces are explicitly modelled could provide more accurate information, especially in terms of crack pattern.

The adopted model was not validated against field measurements, such us deformation of the structure, inspection of crack pattern in masonry and foundation piles. This will be possible, once the field measurement planned by the IBA will be available.

7

Conclusions and recommendations

The city of Amsterdam has a large number of old quay wall section with rotten foundation piles. Since the area is very large and diving inspections are costly and lengthy in time, there is a need to correlate the foundation defect to the masonry damage above the water level. This thesis aimed to find indicators above the water line to identify foundation problems by studying the crack patterns in a typical unreinforced masonry quay in Amsterdam. A parametric study, based on 2D nonlinear finite element analyses is performed. The material properties of masonry, lateral boundary conditions and the extend of pile defects are varied. The following conclusions and recommendations are defined:

7.1 Conclusions

- The material properties of the masonry influence the development of damage patterns. By increasing the values of the tension parameters, such as Young's modulus, tensile strength and fracture energy, a more stiff masonry is obtained. A more stiff masonry needs a larger settlement displacement to obtain the same crack pattern. Nevertheless, the evolution of damage and the observed crack types result the same.

- The lateral constraints alter the structural behaviour of the quay wall section since they allow for arching of the internal forces. Due to this effect, it is more likely that horizontal cracks occur in the case of horizontally fixed lateral edges during the failure of the pile foundation.

- There is a correlation between the applied displacement and the length of the foundation defect. When the length of the foundation defect is relatively large, the value of the applied displacement is of less importance.

- There is a correlation between the deformations at the top of the quay wall section and the length of the foundation defect. An increase of the number of neighbouring piles failing in a quay wall section results in larger deformations on top of the structure.

- The positioning of the foundation defect does not alter the type of cracks.

- By assuming an orthotropic behaviour of the masonry instead of an isotropic behaviour, horizontal cracks occur at smaller settlement displacements. Moreover, horizontal cracks occur in a different location.

- A vertical crack from the bottom of the structure in combination with a horizontal crack is a good indicator for foundation defects. Vertical cracks from the top of the structure indicate the location of the inflexion points of the imposed deformation.

7.2 Recommendations

- The influence of external loads, such as traffic and soil pressure, are not considered in this thesis but can have a significant effect on the quay walls. Primarily since the loads increased the past year, it is recommended to analyze this influence. If combinations of different loading are considered a revision of the mechanical and finite element model should be made, i.e if the effect of soil pressure is included a 3D model is necessary.

- To save computational time, the macro model is used to perform the analyses. Nevertheless, to capture detailed information such as the interaction between the unit bricks and the mortar it is advised to use the micromodel.

- No verification has been done regarding the lateral ends of the quay walls and the material properties of both the masonry as the foundation piles. To show the actual behaviour of the quay walls, it is recommended to validate the model against field measurements.

References

Baars-CIPRO (2019). Inspection reports. Technical report, Baars-CIPRO.

Ben-Gera, T. (2019). *Compressive Membrane Action in Immersed Tubes.* PhD thesis, Delft University of Technology.

Blaß, H. and Sandhaas, C. (2017). *Timber engineering: principles for design.* Scientific publishing.

Boshoff, F. (1990). Funderingen 20-40 Gordel. Technical report.

De Gijt, J. G. and Roubos, A. (2013). *SBRCURnet publicatie Binnenstedelijke kademuren.* Joshua Tourrich.

De Vent, I. (2011). *Structural damage in masonry; Developing diagnostic decision support.* PhD thesis, Delft University of Technology.

Dijksma, S. (2019). Brief aan gemeenteraad over aanpak civiele constructies.

Giardina, G. (2013). *Modelling of settlement induced building damage.* PhD thesis, Delft University of Technology.

Hordijk, D. A. (1992). Tensile and tensile fatigue behaviour of concrete; experiments, modelling and analyses.

Klaassen, R. K. (2008). Water flow through wooden foundation piles: A preliminary study. *International Biodeterioration and Biodegradation*, 61(1):61–68.

Klaassen, R. K. and van Overeem, B. S. (2012). Factors that influence the speed of bacterial wood degradation. *Journal of Cultural Heritage*, 13(3 SUPPL.):S129–S134.

Lourenço, P. (1996). Computational strategies for masonry structures.

Lourenço, P., Rots, J., and Blaauwendraad, J. (1995). Two approaches for the analysis of masonry structures: micro and macro-modeling. *Heron (Delft)*, 40(4):313–340.

Pande, G., Kralj, B., and Middleton, J. (1994). Analysis of the compressive strength of masonry given by the equation fk = K fbˆ α fmˆβ. *The Structural Engineer*, 71(1):7–13.

Rots, J. (1988). *Computational modeling of concrete fracture.* PhD thesis, TU Delft.

Rots, J. (1991). Numerical simulation of cracking in structural masonry.

Rots, J. and Blaauwendraad, J. (1989). Crack models for concrete: discrete or smeared? Fixed multi-directional or rotatin? *Heron*, 34(1):3–59.

SBRCURnet (2014). Richtlijn onderzoek en beoordeling vanhouten paalfunderingen onder gebouwen. Technical report, SBRCURnet.

Schreppers, G., Garofano, A., Messali, F., and Rots, J. (2016). DIANA FEA report 2016-DIANA-R1601: DIANA, TU Delft Structural Mechanics CiTG report CM-2016-17: Validation report for Masonry modelling. Technical report, DIANA FEA B.V.

Van der Pluijm, R. (1997). Nonlinear behavior of masonry under tension. *Heron*, 42(1):25–54.

Van Noort, J.-R. (2012). *Computational modelling of masonry structures.* PhD thesis, Delft University of Technology.

Vecchio, F. J. and Collins, M. P. (1986). Modified Compression-Field Theory for Reinforced Concrete Elements Subjected To Shear. *Journal of the American Concrete Institute*, 83(2):219–231.

List of Figures

List of Tables

Appendices

A

Additional graphs and illustrations

In this Appendix, the crack patterns for the weak, average and strong masonry are given for the four load levels. This Appendix also shows the results of the performed analyses in Diana. The graphs shows *displacement b* as a function of *displacement a* for the three types of masonry. First the graphs regarding the influence of the material are shown, then the graphs regarding the influence of the lateral constraints. There after, the graphs regarding the asymmetric foundation defect and the influence of the orthotropic material model are presented.

The four load levels are:

- Point A: Onset of cracking

- Point B: Abruptly formation of the vertical crack in the middle of the structure.

- Point C: Onset of formation of the vertical cracks above the inflexion points.

- Point D: Onset of formation of the horizontal crack in the middle of the structure.

A.1 Crack patterns

This section shows the crack patterns for the four load levels. Three types of masonry are presented in illustrations: weak, average and strong masonry.
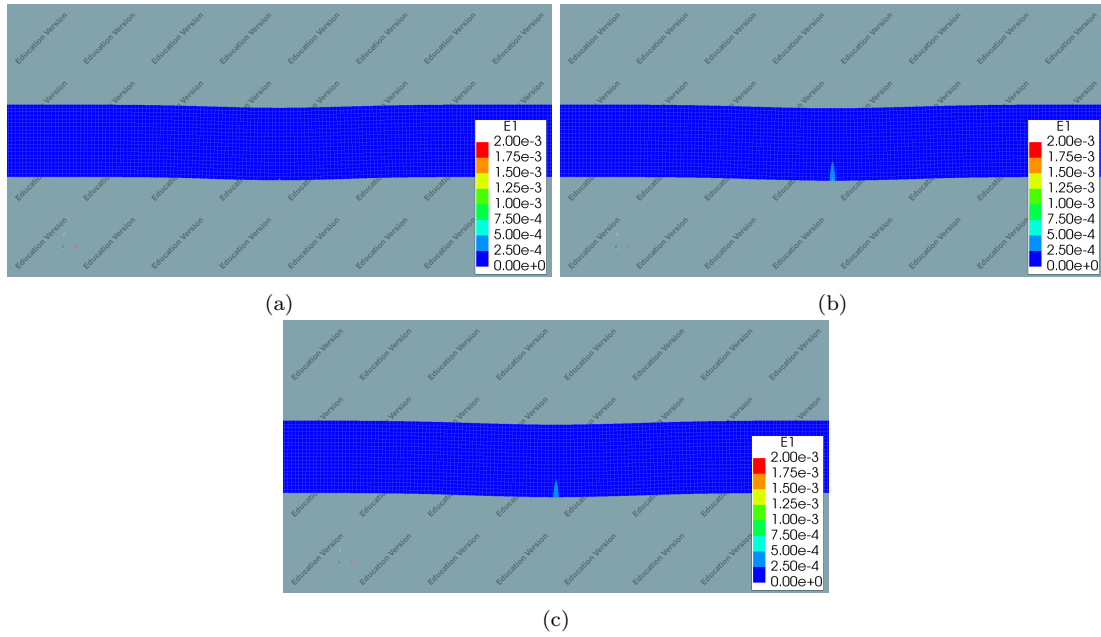
Figure A.1: Crack patterns at load level A: Onset of cracking, (a) weak masonry, (b) average masonry, (c) strong masonry.

A.2 Influence of material properties
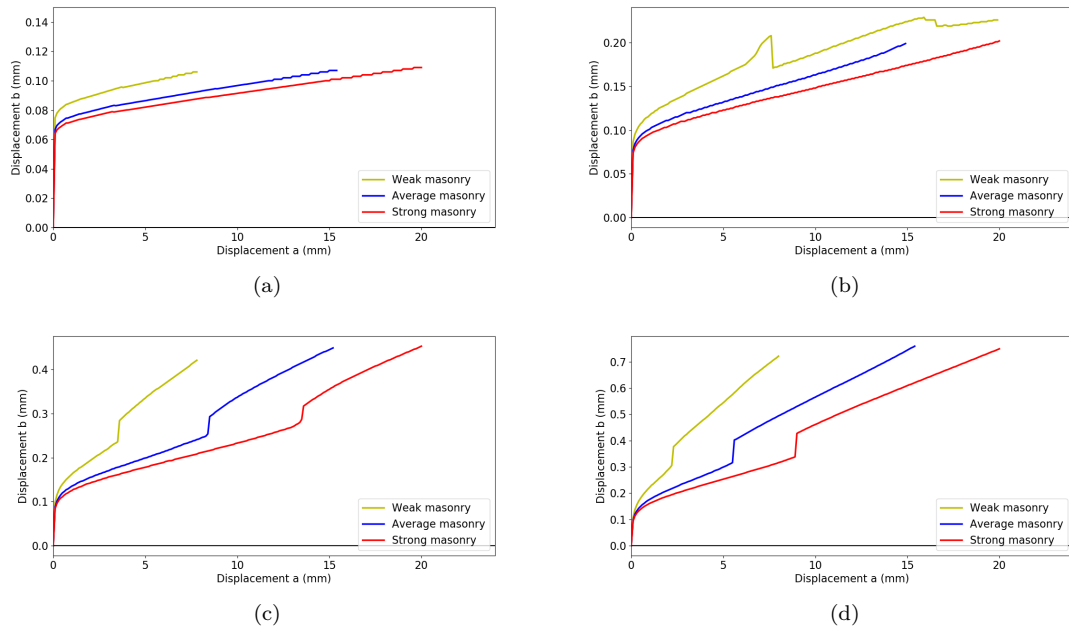
Figure A.5: *Displacement b* plotted as a function of *displacement a* for load level $L_s/L =:$ (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20.

(a)

(b)

(c)
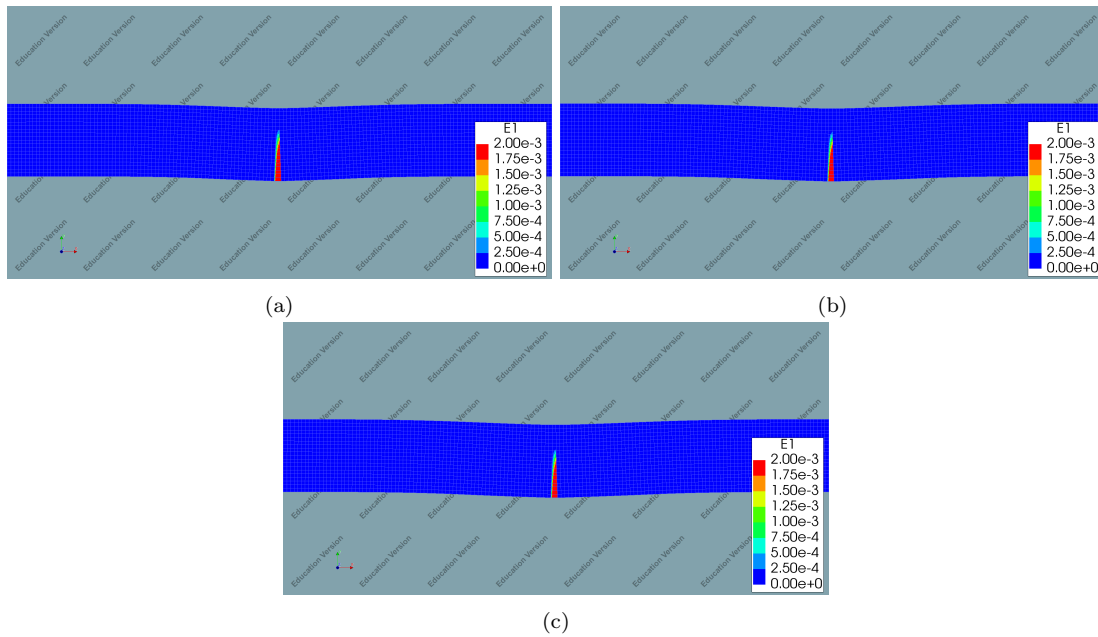
Figure A.2: Crack patterns at load level B: Vertical crack in the middle of the structure, (a) weak masonry, (b) average masonry, (c) strong masonry.

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(a)

(b)

(c)
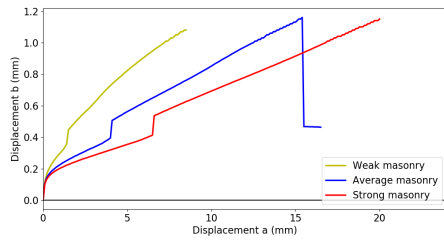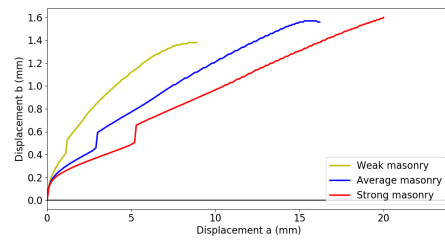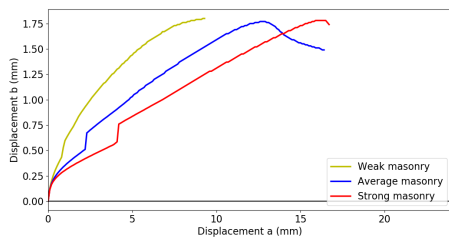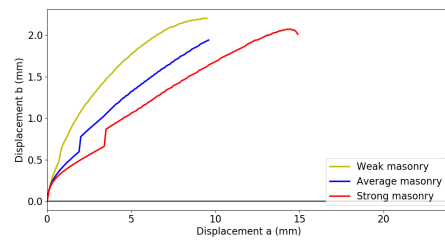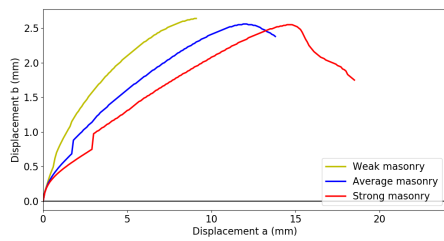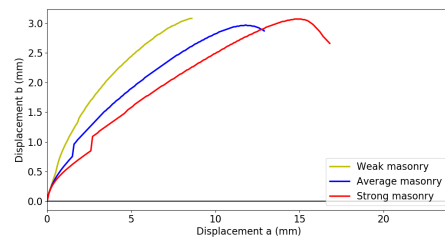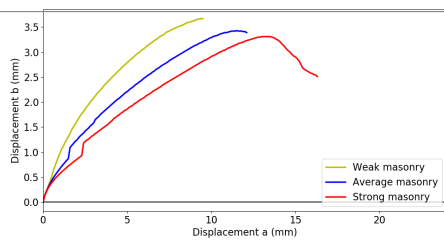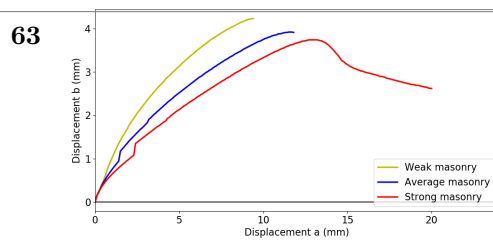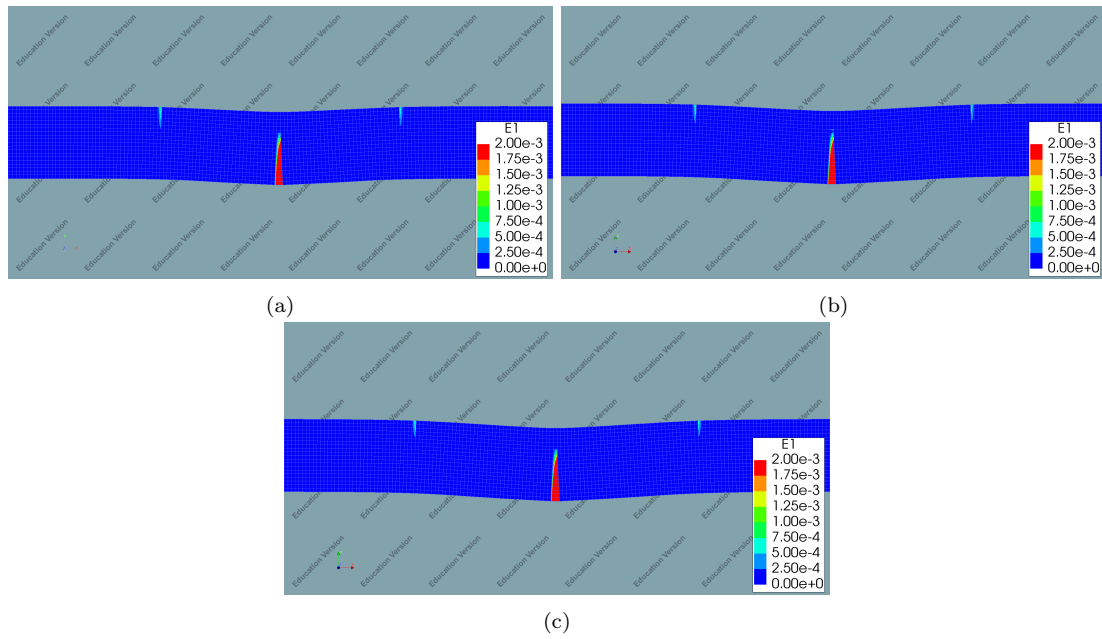
Figure A.3: Crack patterns at load level C: Vertical cracks above the inflexion points, (a) weak masonry, (b) average masonry, (c) strong masonry.

(a)

(b)

(c)
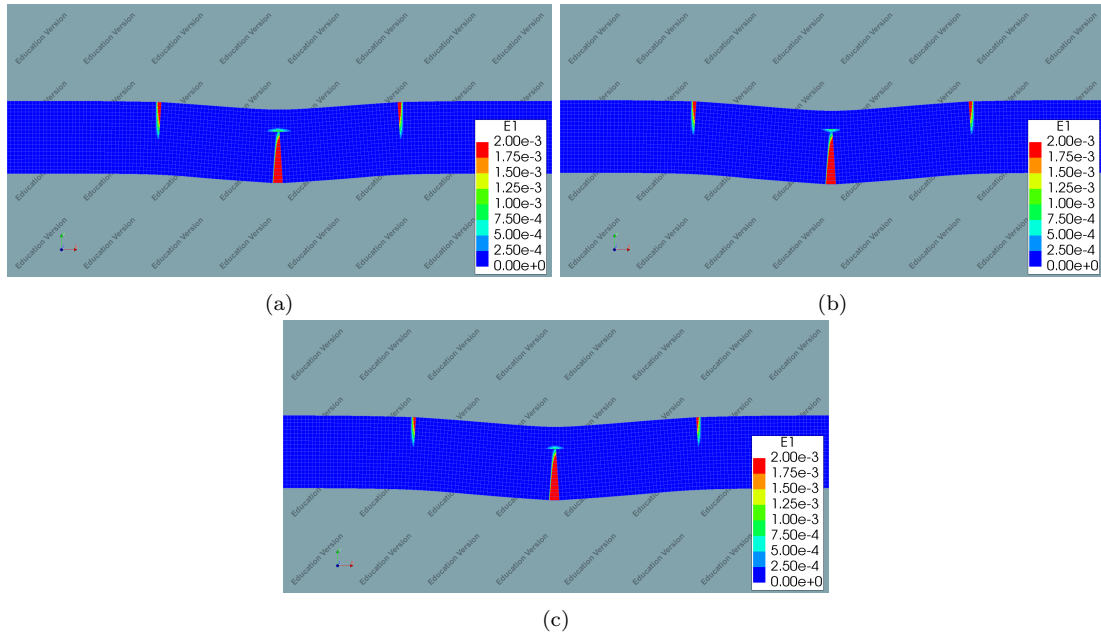
Figure A.4: Crack patterns at load level D: Horizontal crack in the middle of the structure, (a) weak masonry, (b) average masonry, (c) strong masonry.

A.3 Influence of lateral constraints

(a)

(b)

(c)

(d)

(e)

(f)
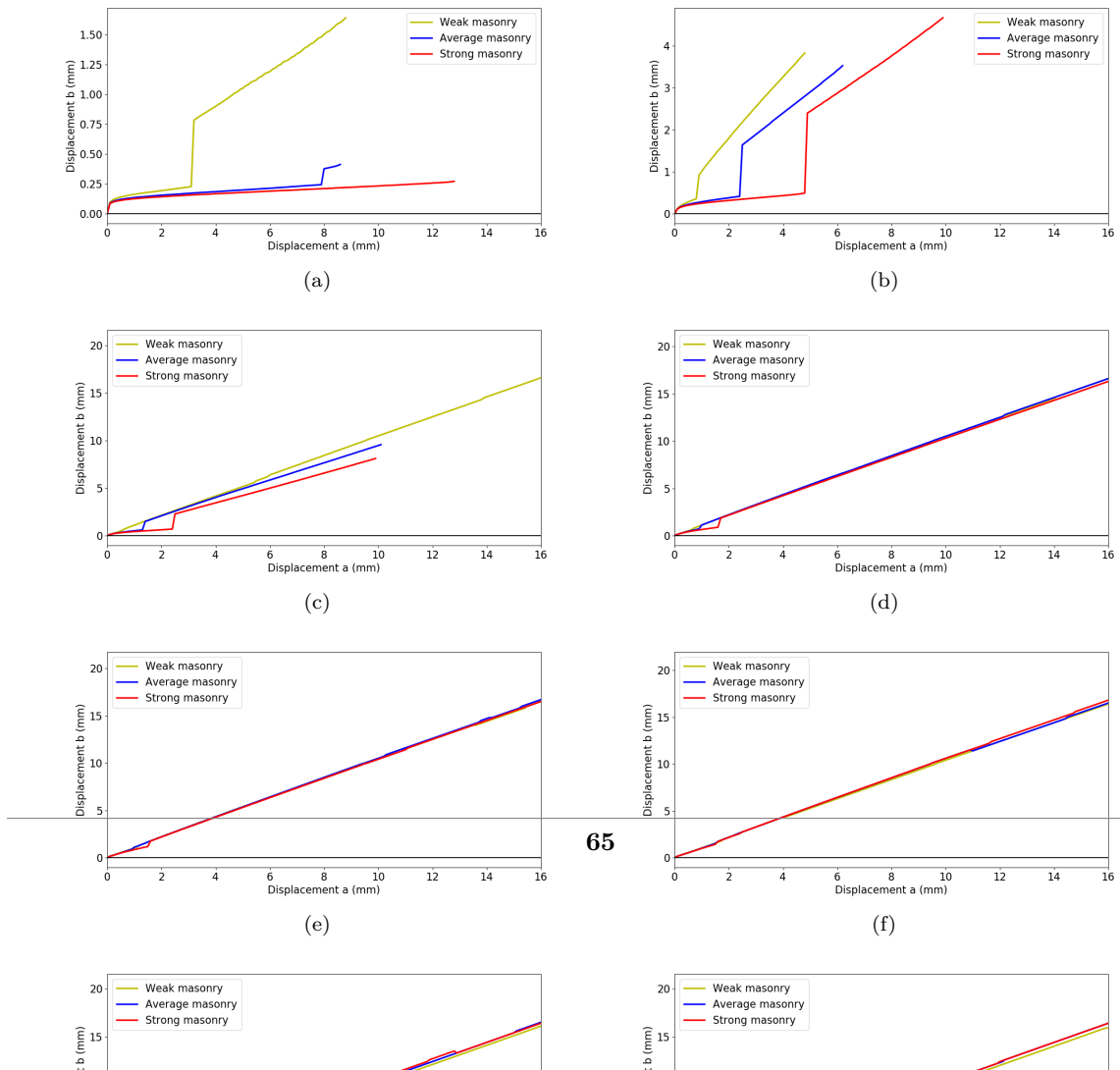
A.4   Influence of the location of foundation defects
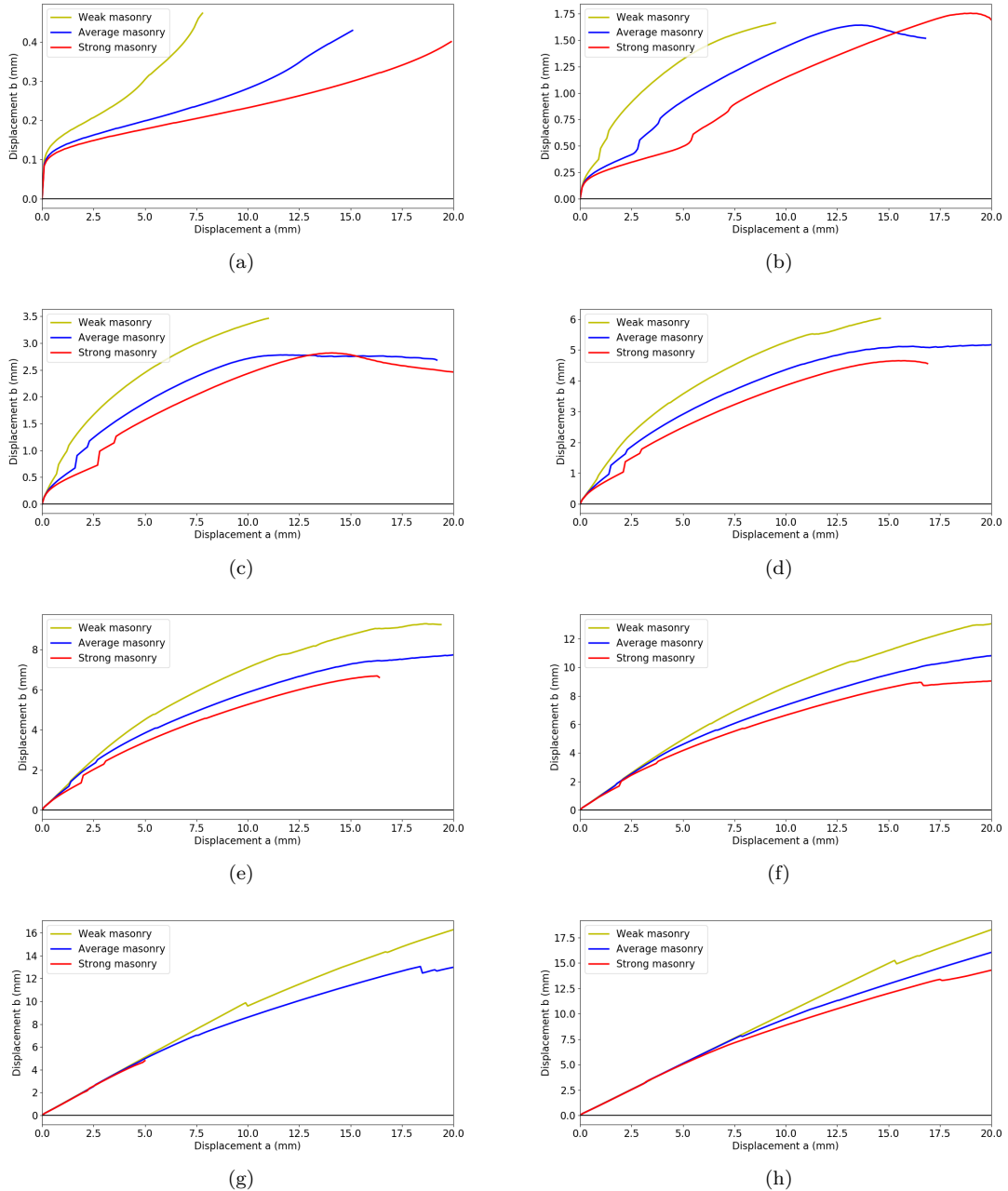
Figure A.8: *Displacement b* plotted as a function of *displacement a* for load level L$_s$/L =: (a) 0.075, (b) 0.15, (c) 0.225, (d) 0.30, (e) 0.0.375, (f) 0.45, (g) 0.525, (h) 0.60. The base case with the asymmetric load level.

A.5   Influence of orthotropic behaviour of masonry
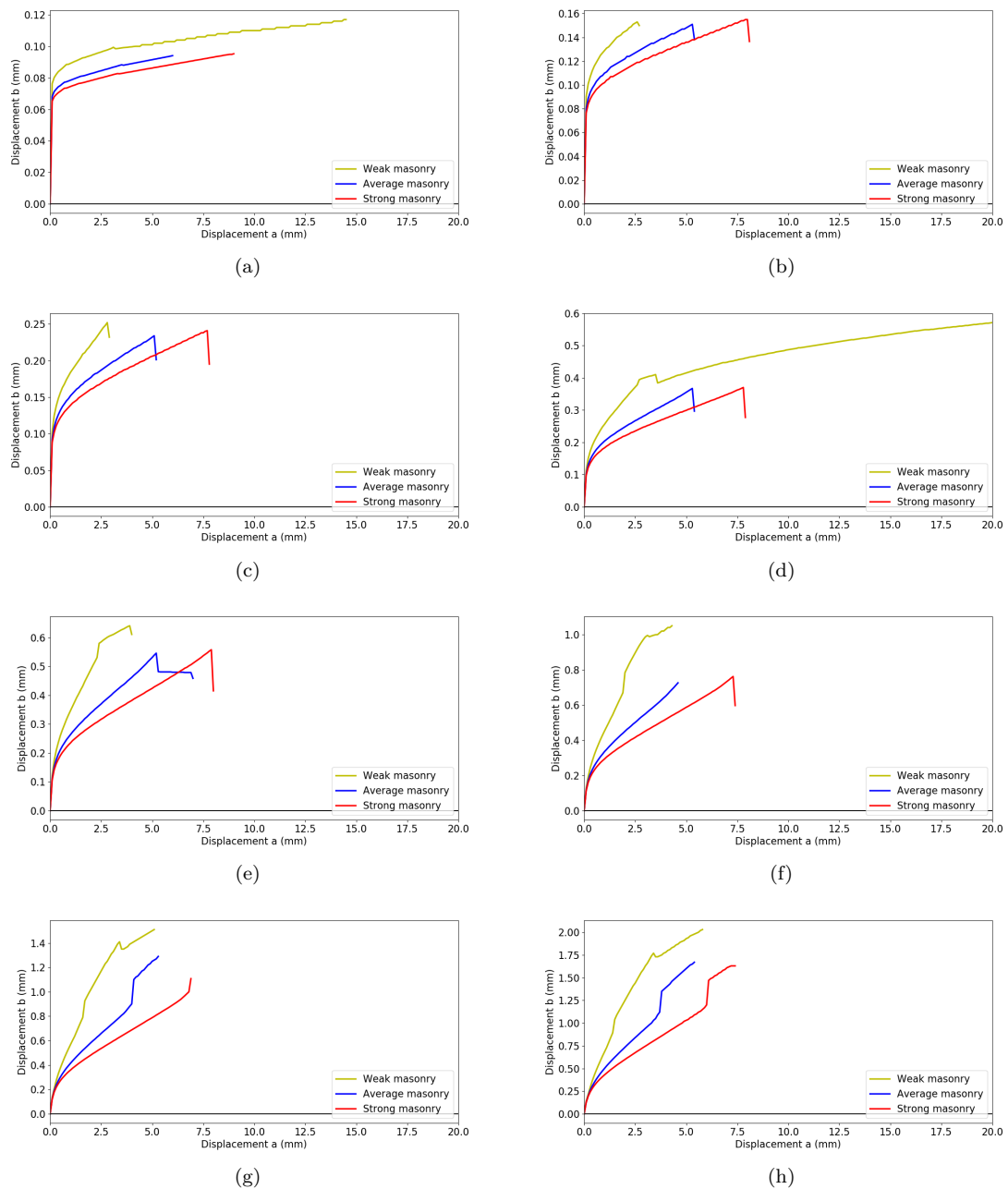
Figure A.9: *Displacement b* plotted as a function of *displacement a* for load level L$_s$/L =: (a) 0.05, (b) 0.10, (c) 0.15, (d) 0.20, (e) 0.25, (f) 0.30, (g) 0.35, (h) 0.40. The base case with the orthotropic material model.
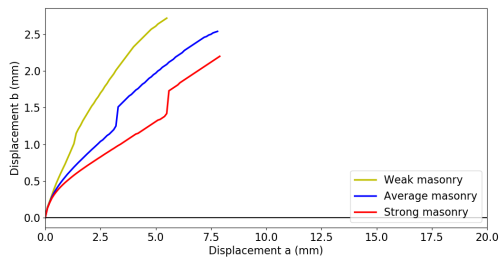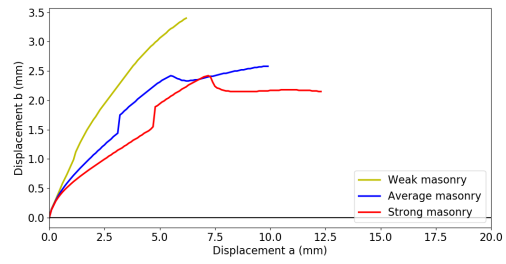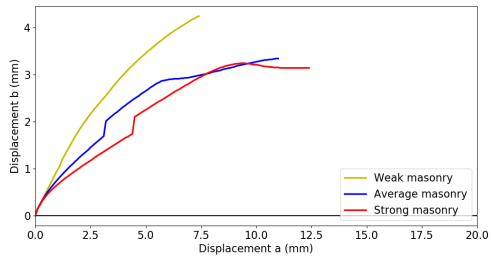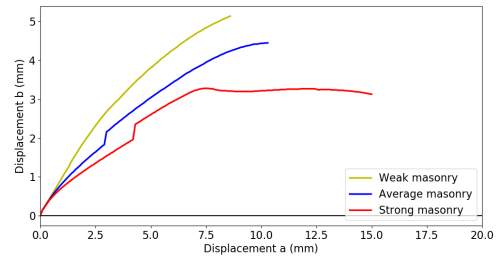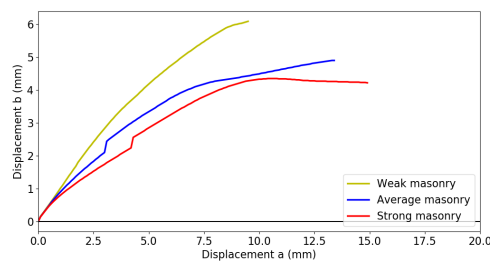
Figure A.10: *Displacement b* plotted as a function of *displacement a* for load level $L_s/L =$: (a) 0.45, (b) 0.50, (c) 0.55, (d) 0.60, (e) 0.65. The base case with the orthotropic material model.

B

Python script

This Appendix shows the Python script (Python 3.7) used to perform the analyses in DIANA 10.3.

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Jan 22 08:23:28 2020

@author: NLMUND
"""

import math
class env():
    def __init__(self): return

config=env()

#################################    Convert input
############################################################################################


def convert(config):
#Geometry
    config.block_h         = config.block[0]
    config.block_w         = config.block[1]
    config.block_elements  = config.block[2]
    config.block_name      = config.block[3]

#Material (variations)
    config.masonry_E_x    = config.masonry[0]
    config.masonry_E_y    = config.masonry[1]
    config.masonry_ft_x   = config.masonry[2]
    config.masonry_ft_y   = config.masonry[3]
    config.masonry_Gft_x  = config.masonry[4]
    config.masonry_Gft_y  = config.masonry[5]

#Material (constants)
```

```
#general
config.masonry_rho          = config.masonry_constant[0]
config.masonry_poisson      = config.masonry_constant[1]
#compression
config.masonry_fc           = config.masonry_constant[2]
config.masonry_Gfc          = config.masonry_constant[3]
#shear
config.masonry_G            = config.masonry_constant[4]
config.masonry_fs           = config.masonry_constant[5]
config.masonry_c            = config.masonry_constant[6]
config.masonry_Gfs          = config.masonry_constant[7]
config.masonry_res_strength = config.masonry_constant[8]


#Steelbeam
config.steelbeam_h    = config.steelbeam[0]
config.steelbeam_E    = config.steelbeam[1]
config.steelbeam_v    = config.steelbeam[2]
config.steelbeam_rho  = config.steelbeam[3]


#Q-last
config.load_all_name      = config.load_all[0]
config.load_all_magnitude = config.load_all[1]
config.load_all_direction = config.load_all[2]


#loadsingle
config.load_single_name      = config.load_single[0]
config.load_single_magnitude = config.load_single[1]
config.load_single_direction = config.load_single[2]
config.load_single_place     = config.load_single[3]
config.load_single_n         = config.load_single[4]


#function
config.function_mu       = config.function[0]
config.function_sigma    = config.function[1]
config.function_stepsize = config.function[2]
config.function_factor   = 100
config.function_elements = config.block[2]


#output
config.boundaryvalues_deform_xmin  = config.boundaryvalues_deform[0]
config.boundaryvalues_deform_xmax  = config.boundaryvalues_deform[1]
config.boundaryvalues_deform_ymin  = config.boundaryvalues_deform[2]
config.boundaryvalues_deform_ymax  = config.boundaryvalues_deform[3]

config.boundaryvalues_stress_xmin  = config.boundaryvalues_stress[0]
config.boundaryvalues_stress_xmax  = config.boundaryvalues_stress[1]
config.boundaryvalues_stress_ymin  = config.boundaryvalues_stress[2]
config.boundaryvalues_stress_ymax  = config.boundaryvalues_stress[3]

config.boundaryvalues_strain_xmin  = config.boundaryvalues_strain[0]
config.boundaryvalues_strain_xmax  = config.boundaryvalues_strain[1]
config.boundaryvalues_strain_ymin  = config.boundaryvalues_strain[2]
config.boundaryvalues_strain_ymax  = config.boundaryvalues_strain[3]

############################   Define the project
#######################################################
```

```python
def createNewModel(config):
    print("CREATING_NEW_MODEL")
    mystring = newProject("%s/%s/%s" % (config.home, config.map, config.filename),
        100)
    setModelAnalysisAspects( [ 'STRUCT' ] )
    setModelDimension( '2D' )
    setDefaultMeshOrder( 'QUADRATIC' )
    setDefaultMesherType( 'HEXQUAD' )
    setDefaultMidSideNodeLocation( 'ONSHAP' )
    setUnit( 'LENGTH', 'MM' )
    setUnit( 'FORCE', 'N' )
    return(mystring)


def saveproject(config):
    saveProjectAs("%s/%s/%s" % (config.home, config.map, config.filename))


def closeproject(config):
    mystring = closeProject()
    return(mystring)

###############################        Define  Geometry
    ################################################################

def geometry(config):
    print("CREATING_GEOMETRY")
    createSheet (config.block_name,   [[0,0,0],[0,config.block_h,0],[config.block_w
        ,config.block_h,0],[config.block_w,0,0]])
    createLine("Line1", [0, -100, 0], [config.block_w, -100, 0])

    arrayCopy (config.block_name, [ config.block_w, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0
        ], (config.block_elements-1))
    arrayCopy ("Line1", [ config.block_w, 0, 0 ], [ 0, 0, 0 ], [ 0, 0, 0 ], (
        config.block_elements-1))


###############################        Define  Material
    ################################################################


###############################        Masonry
def creatematerial_TotalStrain(config):
    addMaterial("TotalStrain", "CONCR", "TSCR", [])
    setParameter(MATERIAL, "TotalStrain", "LINEAR/ELASTI/YOUNG", config.
        masonry_E_x)
    setParameter(MATERIAL, "TotalStrain", "LINEAR/ELASTI/POISON", config.
        masonry_poisson)
    setParameter(MATERIAL, "TotalStrain", "LINEAR/MASS/DENSIT", config.masonry_rho
        )
    setParameter(MATERIAL, "TotalStrain", "MODTYP/TOTCRK", "ROTATE")
    setParameter(MATERIAL, "TotalStrain", "TENSIL/TENCRV", "LINEAR")
    setParameter(MATERIAL, "TotalStrain", "TENSIL/TENSTR", config.masonry_ft_x)
    setParameter(MATERIAL, "TotalStrain", "TENSIL/GF1", config.masonry_Gft_x)
    setParameter(MATERIAL, "TotalStrain", "TENSIL/RESTST", config.
        masonry_res_strength)
```

```python
        setParameter(MATERIAL, "TotalStrain", "COMPRS/COMCRV","ELASTI")

def creatematerial_Engineering(config):
    addMaterial( "Engineering", "CONCR", "MASONR", [] )
    setParameter( MATERIAL, "Engineering", "ELASTI/YOUNG", [ config.masonry_E_x,
        config.masonry_E_y ] )
    setParameter( MATERIAL, "Engineering", "ELASTI/SHRMOD", [ config.masonry_G ] )
    setParameter( MATERIAL, "Engineering", "ELASTI/MASS/DENSIT", config.
        masonry_rho )
    setParameter( MATERIAL, "Engineering", "CRACKI/HEADTP", "EXPLIC" )
    setParameter( MATERIAL, "Engineering", "CRACKI/TENSI2/TENSTR", [ config.
        masonry_ft_x, config.masonry_ft_y ] )
    setParameter( MATERIAL, "Engineering", "CRACKI/GF1", config.masonry_Gft_x )
    setParameter( MATERIAL, "Engineering", "CRACKI/RESTST", config.
        masonry_res_strength )
    setParameter( MATERIAL, "Engineering", "CRUSHI/COMSTR", config.masonry_fc )
    setParameter( MATERIAL, "Engineering", "CRUSHI/GC", config.masonry_Gfc )
    setParameter( MATERIAL, "Engineering", "SHEARF/COHESI", config.masonry_c )
    setParameter( MATERIAL, "Engineering", "SHEARF/GFS", config.masonry_Gfs )

def creategeometry(config):
    addGeometry("Element geometry 1", "SHEET", "MEMBRA", [])
    setParameter(GEOMET, "Element geometry 1", "THICK", 100)       #THICKNESS =
        ELEMENTSIZE
    setParameter( GEOMET, "Element geometry 1", "LOCAXS", True )
    setParameter( GEOMET, "Element geometry 1", "LOCAXS/XAXIS", [ 1, 0, 0 ] )

def setelementclasstype(config):
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Masonry"+str(x)
        setElementClassType(SHAPE, [name], "MEMBRA")

def assignmaterial(config):
    if config.input_material_model == 1:
        config.material_model = "TotalStrain"
    else:
        config.material_model = "Engineering"
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Masonry"+str(x)
        assignMaterial(config.material_model, SHAPE, [name])


def Iteration_scheme(config):
    addElementData("Gauss3")
    setParameter("DATA","Gauss3", "NINTEG", [3,3])
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Masonry"+str(x)
        assignElementData("Gauss3", SHAPE, [name])

def assigngeometry(config):
```

```
        x=0
        for i in range (0,config.block_elements):
            x= x+1
            name = "Masonry"+str(x)
            assignGeometry("Element␣geometry␣1", SHAPE, [name])
```

```
########################## Steel beam
```

```
def creatematerial_beam(config):
    addMaterial( "Steel␣Beam", "MCSTEL", "ISOTRO", [] )
    setParameter( MATERIAL, "Steel␣Beam", "LINEAR/ELASTI/YOUNG", config.
        steelbeam_E )
    setParameter( MATERIAL, "Steel␣Beam", "LINEAR/ELASTI/POISON", config.
        steelbeam_v )
    setParameter( MATERIAL, "Steel␣Beam", "LINEAR/MASS/DENSIT", config.
        steelbeam_rho )
```

```
def creategeometry_beam(config):
    addGeometry( "Element␣geometry␣beam", "LINE", "CLS3B2", [] )
    setParameter( GEOMET, "Element␣geometry␣beam", "SHAPE/RECTAN", [ config.
        steelbeam_h, config.mesh_elementsize ] )
```

```
def setelementclasstype_beam(config):
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Line"+str(x)
        setElementClassType(SHAPE, [name], "CLS3B2")
```

```
def assignmaterial_beam(config):
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Line"+str(x)
        assignMaterial("Steel␣Beam", SHAPE, [name])
```

```
def assigngeometry_beam(config):
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Line"+str(x)
        assignGeometry("Element␣geometry␣beam", SHAPE, [name])
```

```
################################    Supports
    ###########################################################
```

```
### Edgesupports
```

```
def addsupportsets(config):
    addSet( GEOMETRYSUPPORTSET, "Edge␣support" )
    addSet( GEOMETRYSUPPORTSET, "Bottom␣support" )
```

```python
def edgesupport_left(config):
    print("CREATING EDGESUPPORT LEFT")

    createLineSupport( "Edge support left", "Edge support" )
    setParameter( GEOMETRYSUPPORT, "Edge support left", "AXES", [ 1, 2 ] )
    setParameter( GEOMETRYSUPPORT, "Edge support left", "TRANSL", [ 1, 0, 0 ] )
    setParameter( GEOMETRYSUPPORT, "Edge support left", "ROTATI", [ 0, 0, 0 ] )
    attach( GEOMETRYSUPPORT, "Edge support left", "Masonry1", [[ 0, 10, 0 ]] )

def edgesupport_right(config):
    print("CREATING EDGESUPPORT RIGHT")
    createLineSupport( "Edge support right", "Edge support" )
    setParameter( GEOMETRYSUPPORT, "Edge support right", "AXES", [ 1, 2 ] )
    setParameter( GEOMETRYSUPPORT, "Edge support right", "TRANSL", [ 1, 0, 0 ] )
    setParameter( GEOMETRYSUPPORT, "Edge support right", "ROTATI", [ 0, 0, 0 ] )
    attach( GEOMETRYSUPPORT, "Edge support right", "Masonry"+str(config.
        block_elements), [[ config.block_elements*config.block_w, 10, 0 ]] )


### Bottomsupport

def bottomsupport(config):
    print("CREATING BOTTOMSUPPORT")
    createBodySupport( "Bottomsupport", "Bottom support" )
    setParameter( GEOMETRYSUPPORT, "Bottomsupport", "AXES", [ 1, 2 ] )
    setParameter( GEOMETRYSUPPORT, "Bottomsupport", "TRANSL", [ 1, 1, 0 ] )
    setParameter( GEOMETRYSUPPORT, "Bottomsupport", "ROTATI", [ 0, 0, 0 ] )
    x=0
    for i in range (0,config.block_elements):
        x= x+1
        name = "Line"+str(x)
        attach( GEOMETRYSUPPORT, "Bottomsupport", [name])


############################### Interface
    #######################################################################

def creatematerial_interface(config):
    print("CREATING INTERFACE")
    addMaterial( "Interface nonlin", "INTERF", "NONLIF", [] )
    setParameter( MATERIAL, "Interface nonlin", "LINEAR/IFTYP", "LIN2D" )
    setParameter( MATERIAL, "Interface nonlin", "LINEAR/ELAS2/DSNY", .7 )       #
        Value?
    setParameter( MATERIAL, "Interface nonlin", "LINEAR/ELAS2/DSSX", 1e-9 )     #
        Value?
    setParameter( MATERIAL, "Interface nonlin", "NONLIN/IFNOTE", "NOTENS" )
    setParameter( MATERIAL, "Interface nonlin", "NONLIN/NLEL7/NOTENS", [ 0.01,
        0.01 ] ) #Value?

def creategeometry_interface(config):
    addGeometry( "Element geometry interface", "LINE", "STLIIF", [] )
    setParameter( GEOMET, "Element geometry interface", "LIFMEM/THICK", config.
        mesh_elementsize )

def createconnection_interface(config):
```

```python
    x=0
    for i in range (0, config.block_elements):
        x= x+1
        name = "Interface"+str(x)
        Line = "Line"+str(x)
        Masonry = "Masonry"+str(x)
        createConnection( name, "INTER", SHAPEEDGE, SHAPEEDGE )
        setParameter( GEOMETRYCONNECTION, name, "MODE", "MANUAL" )
        setElementClassType( GEOMETRYCONNECTION, name, "STLIIF" )
        assignMaterial( "Interface nonlin", GEOMETRYCONNECTION, name )
        assignGeometry( "Element geometry interface", GEOMETRYCONNECTION, name )
        setParameter( GEOMETRYCONNECTION, name, "FLIP", False )
        attachTo( GEOMETRYCONNECTION, name, "SOURCE", Line, [[ config.block_w/2.0,
            -100, 0 ]] )
        attachTo( GEOMETRYCONNECTION, name, "TARGET", Masonry, [[ config.block_w
            /2.0, 0, 0 ]] )
        attachTo( GEOMETRYCONNECTION, name, "SRCANC", Line, [[ (x-1)*config.
            block_w, -100, 0 ]] )
        attachTo( GEOMETRYCONNECTION, name, "TRGANC", Masonry, [[ (x-1)*config.
            block_w, 0, 0 ]] )


############################## Loads
    ##################################################################

def addset_loads(config):
    addSet(GEOMETRYLOADSET, "Deadweight")
    addSet( GEOMETRYLOADSET, "External loads" )
    addSet( GEOMETRYLOADSET, "Deform" )

def createdeadweight(config):
    createModelLoad( "Deadweight", "Deadweight" )


def externalload_edge_all(config):
    x=0
    for i in range(0, config.block_elements):
        x=x+1
        name = config.load_all_name+str(x)
        Masonry = "Masonry"+str(x)
        createLineLoad( name, "External loads" )
        setParameter( GEOMETRYLOAD, name, "FORCE/VALUE", config.load_all_magnitude
            ) #value
        setParameter( GEOMETRYLOAD, name, "FORCE/DIRECT", config.
            load_all_direction) #value
        attach( GEOMETRYLOAD, name, Masonry, [[ (x-1)*config.block_w+config.
            block_w/2.0, config.block_h, 0 ]] )

def externalload_edge_single(config):
    x=0
    for i in range(0, config.load_single_n):
        x=x+1
        name = config.load_single_name+str(x)
        Masonry = "Masonry"+str(config.load_single_place+x-1)
        createLineLoad( name, "External loads" )
```

```
        setParameter( GEOMETRYLOAD, name, "FORCE/VALUE", config.
            load_single_magnitude ) #value
        setParameter( GEOMETRYLOAD, name, "FORCE/DIRECT", config.
            load_single_direction) #value
        attach( GEOMETRYLOAD, name, Masonry, [[ (config.load_single_place-1+(x-1))
            *config.block_w+config.block_w/2.0, config.block_h, 0 ]] )


################################    Create deformation functions
    ################################################################################


def makegauss(config):
    print("CREATING_A_GAUSSFUNCTION")
    config.maximum = config.block_w * config.function_elements
    config.numberstep = config.maximum/config.function_stepsize
    config.a = []
    for i in range (0, int(config.numberstep+1)):  #Creating an array of zeros
        config.a.append(0)
    config.b = []
    for i in range(0, int(config.numberstep+1)):#Creating an array of zeros
        config.b.append(0)
    config.c = []
    for i in range(0, int(config.numberstep+1)):#Creating an array of zeros
        config.c.append(0)
    f=0
    config.a[0] = 0 #0.5*config.block_elements-(0.5*config.function_elements))*
        config.block_w
    for i in range (1, int(config.numberstep+1)):
        config.a[f+1] = config.a[f]+config.function_stepsize
        f=f+1
    config.b[0] = 0
    g=0
    for i in range (1, int(config.numberstep+1)):
        config.b[g+1] = config.b[g]+config.function_stepsize
        g=g+1
    for i in range (1,int(config.numberstep+1)):
        config.c[i] = -config.function_factor*math.exp((-(config.b[i]-config.
            function_mu)**2)/(2*config.function_sigma**2))
        config.c[i] = round(config.c[i],5)

def createfunction(config):
    makegauss(config)
    setFunctionValues("Gauss", config.a,[   ], [   ], config.c )

def deformation(config):
    createLineLoad( "Deform", "Deform" )
    setParameter( GEOMETRYLOAD, "Deform", "LODTYP", "DEFORM" )
    setParameter( GEOMETRYLOAD, "Deform", "DEFORM/SUPP", "Bottomsupport" )
    setParameter( GEOMETRYLOAD, "Deform", "DEFORM/TR/VALUE", 1 )   #mm
    setParameter( GEOMETRYLOAD, "Deform", "DEFORM/TR/DIRECT", 2 )  # y-direction
    x=-0.5*config.block_w
    y = 0
    for i in range (0,config.block_elements):
        x= x+config.block_w
```

```
            y = y+1
            Line = "Line"+str(y)
            attach( GEOMETRYLOAD, "Deform", Line, [[ x, -100, 0 ]] )
        setValueFunction( GEOMETRYLOAD, "Deform", "Gauss" )    ###adjust Parabool




############################## Mesh
    #######################################################
def makearray_mesh(config):
    config.maximum = config.block_w * config.block_elements
    config.numberstep   = config.maximum/config.function_stepsize
    config.Masonry = []
    for i in range (0, config.block_elements):
        config.Masonry.append(0)
    config.Line = []
    for i in range(0, config.block_elements):
        config.Line.append(0)
    x=0
    config.Masonry[0] = 0
    for i in range (1, config.block_elements+1):
        x = x+1
        config.Masonry[x-1] = "Masonry"+str(x)
        config.Line[x-1] = "Line"+str(x)


def create_mesh(config):
    print("CREATING MESH")
    makearray_mesh(config)
    setElementSize(config.Masonry, config.mesh_elementsize, -1, True)
    setElementSize(config.Line, config.mesh_elementsize, -1, True)
    clearMesherType(config.Masonry)
    clearMesherType(config.Line)
    clearMidSideNodeLocation(config.Masonry)
    clearMidSideNodeLocation(config.Line)


def generate_mesh(config):
    generateMesh([])
    hideView("GEOM")
    showView("MESH")

############################## Analysis
    #######################################################

def addanalysis(config):
    addAnalysis(config.analysis)
    addAnalysisCommand(config.analysis, "NONLIN", "Structural nonlinear")

def executeblock_deadweight(config):
    addAnalysisCommandDetail( config.analysis, "Structural nonlinear", "EXECUT(1)/
        LOAD/LOADNR" )
    setAnalysisCommandDetail( config.analysis, "Structural nonlinear", "EXECUT(1)/
        LOAD/LOADNR", 1 )    # 2 loadsteps?

def executeblock_deform1(config):
```

```
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        LOAD/LOADNR" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        LOAD/LOADNR", 3 )#3=deformation
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        LOAD/STEPS/EXPLIC/SIZES", config.steps_1 )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        ITERAT/MAXITE", config.max_iterations ) #max iterations
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        ITERAT/CONVER/DISPLA/NOCONV", "TERMIN" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        ITERAT/CONVER/FORCE/NOCONV", "TERMIN" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(2)/
        ITERAT/CONVER/SIMULT", True )


def executeblock_deform2(config):
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        LOAD/LOADNR" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        LOAD/LOADNR", 3 )#3=deformation
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        LOAD/STEPS/EXPLIC/SIZES", config.steps_2 )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        ITERAT/MAXITE", config.max_iterations ) #max iterations
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        ITERAT/CONVER/DISPLA/NOCONV", "TERMIN" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        ITERAT/CONVER/FORCE/NOCONV", "TERMIN" )
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "EXECUT(3)/
        ITERAT/CONVER/SIMULT", False )




def define_output(config):
    setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        SELTYP","USER")
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/DISPLA(1)/TOTAL" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/FORCE(1)/REACTI" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/FORCE(2)/EXTERN" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/STRAIN(1)/TOTAL/GREEN/PRINCI" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/STRAIN(2)/TOTAL/GREEN/GLOBAL" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/STRAIN(3)/CRACK" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/STRESS(1)/TOTAL/CAUCHY/PRINCI" )
    addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
        USER/STRESS(2)/TOTAL/CAUCHY/GLOBAL" )
```

```
        addAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
            USER/STRESS(3)/CRACK" )

        setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
            USER/STRAIN(1)/TOTAL/GREEN/PRINCI/LOCATI", "INTPNT" )
        setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
            USER/STRESS(1)/TOTAL/CAUCHY/PRINCI/LOCATI", "INTPNT" )
        setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
            USER/STRAIN(2)/TOTAL/GREEN/GLOBAL/LOCATI", "INTPNT" )
        setAnalysisCommandDetail( config.analysis, "Structural␣nonlinear", "OUTPUT(1)/
            USER/STRESS(2)/TOTAL/CAUCHY/GLOBAL/LOCATI", "INTPNT" )


def run_analysis(config):
        runSolver([config.analysis])




############################## Output
    ######################################################
def viewsettings(config):
        setViewSettingValue( "view␣setting", "RESULT/DEFORM/MODE", "ABSOLU" )
        setViewSettingValue( "view␣setting", "RESULT/DEFORM/ABSOLU/FACTOR", 100 )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LEGEND", "DISCRE" )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/AUTRNG", "LIMITS" )


def viewsettings_xdeform(config):
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
            boundaryvalues_deform_xmin )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
            boundaryvalues_deform_xmax )


def viewsettings_ydeform(config):
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
            boundaryvalues_deform_ymin )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
            boundaryvalues_deform_ymax )


def viewsettings_xstress(config):
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
            boundaryvalues_stress_xmin )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
            boundaryvalues_stress_xmax )


def viewsettings_ystress(config):
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
            boundaryvalues_stress_ymin )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
            boundaryvalues_stress_ymax )


def viewsettings_xstrain(config):
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
            boundaryvalues_strain_xmin )
        setViewSettingValue( "view␣setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
            boundaryvalues_strain_xmax )
```

```python
def viewsettings_ystrain(config):
    setViewSettingValue( "view setting", "RESULT/CONTOU/LIMITS/MINVAL", config.
        boundaryvalues_strain_ymin )
    setViewSettingValue( "view setting", "RESULT/CONTOU/LIMITS/MAXVAL", config.
        boundaryvalues_strain_ymax )


def export_displacements(config):
    cases = resultCases(config.analysis,'Output')    #output 1 or 2
    nodes  = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [ config.analysis, 'Output', 'Displacements', 'TDtY']
    exportResultsToCSV('Displacementy_sigma%s.csv'%(int(config.function_sigma)),
        table, cases, [int(nodes[0])])

    def plot_displacements(name,contours,output,direction):
        setResultPlot(contours,output,direction)
        writeToPng( "img_%s_step_%s.png" % (name,i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis,cases[i]))
        viewsettings_xdeform(config)
        plot_displacements('Displacement_x', 'contours', 'Displacements/node', '
            TDtX')

    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis,cases[i]))
        viewsettings_ydeform(config)
        plot_displacements('Displacement_y', 'contours', 'Displacements/node', '
            TDtY')


def export_E1(config):
    cases = resultCases(config.analysis,'Output')    #output 1 or 2
    nodes  = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [ config.analysis, 'Output 2', 'Total Strains', 'E1']
    exportResultsToCSV('E1_%s_%s.csv'%(int(config.mu), int(config.sigma)), table,
        cases, [int(nodes[0])])
    viewsettings_xstrain(config)

    def plot_E1(name,contours,output,direction):
        setResultPlot(contours,output,direction)
    writeToPng( "img_%s_step_%s.png" % (name,i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
```

```python
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output_2/%s' % (config.analysis, cases[i]))
        viewsettings_xstrain(config)
        plot_E1('Principal_Strain_E1', 'contours', 'Total_Strains/mappedintpnt', 'E1'
            )

def export_Exx(config):
    cases = resultCases(config.analysis, 'Output')      #output 1 or 2
    nodes = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [config.analysis, 'Output', 'Total_Strains', 'EXX']
    exportResultsToCSV('Exx_down_%s_%s.csv'%(int(config.mu), int(config.sigma)),
        table, cases, [int(nodes[0])])
    viewsettings_xstrain(config)

    def plot_Exx(name, contours, output, direction):
        setResultPlot(contours, output, direction)
        writeToPng("img_%s_step_%s.png" % (name, i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis, cases[i]))
        viewsettings_xstrain(config)
        plot_Exx('Total_Strain_Exx', 'contours', 'Total_Strains/mappedintpnt', 'EXX')

def export_Eyy(config):
    cases = resultCases(config.analysis, 'Output')      #output 1 or 2
    nodes = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [config.analysis, 'Output', 'Total_Strains', 'EYY']
    exportResultsToCSV('Eyy_down_%s_%s.csv'%(int(config.mu), int(config.sigma)),
        table, cases, [int(nodes[0])])
    viewsettings_ystrain(config)

    def plot_Eyy(name, contours, output, direction):
        setResultPlot(contours, output, direction)
        writeToPng("img_%s_step_%s.png" % (name, i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis, cases[i]))
        viewsettings_ystrain(config)
        plot_Eyy('Total_Strain_Eyy', 'contours', 'Total_Strains/mappedintpnt', 'EYY')
```

```python
def export_S1(config):
    cases = resultCases(config.analysis, 'Output␣2')    #output 1 or 2
    nodes = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [ config.analysis, 'Output␣2', 'Cauchy␣Total␣Stresses', 'S1']
    exportResultsToCSV('S1_%s_%s.csv'%(int(config.mu), int(config.sigma)), table,
        cases, [int(nodes[0])])

    def plot_S1(name, contours, output, direction):
        setResultPlot(contours, output, direction)
        writeToPng( "img_%s_step_%s.png" % (name, i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output␣2/%s' % (config.analysis, cases[i]))
        viewsettings_xstress(config)
        plot_S1('Principal_Stresses_S1', 'contours', 'Cauchy␣Total␣Stresses/
            mappedintpnt', 'S1')


def export_Sxx(config):
    cases = resultCases(config.analysis, 'Output')    #output 1 or 2
    nodes = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
    table = [ config.analysis, 'Output', 'Cauchy␣Total␣Stresses', 'SXX']
    exportResultsToCSV('Sxx_down_%s_%s.csv'%(int(config.mu), int(config.sigma)),
        table, cases, [int(nodes[0])])

    def plot_Sxx(name, contours, output, direction):
        setResultPlot(contours, output, direction)
        writeToPng( "img_%s_step_%s.png" % (name, i), png_W, png_H)

    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H

    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis, cases[i]))
        viewsettings_xstress(config)
        plot_Sxx('Total_Stress_Sxx', 'contours', 'Cauchy␣Total␣Stresses/node', 'SXX')


def export_Syy(config):
    cases = resultCases(config.analysis, 'Output')    #output 1 or 2
    nodes = findNearestNodes(config.nodes)
    print(nodes)
    elements = findNearestElements(config.elements)
    print(elements)
```

```
    table = [ config.analysis, 'Output', 'Cauchy␣Total␣Stresses', 'SYY']
    exportResultsToCSV( 'Syy_down_%s_%s.csv '%(int(config.mu), int(config.sigma)),
        table, cases, [int(nodes[0])])


    def plot_Syy(name,contours,output,direction):
        setResultPlot(contours,output,direction)
        writeToPng( "img_%s_step_%s.png" % (name,i), png_W, png_H)


    viewsettings(config)
    png_W = config.png_W
    png_H = config.png_H


    for i in range(1, len(cases)):
        setResultCase('%s/Output/%s' % (config.analysis,cases[i]))
        viewsettings_ystress(config)
        plot_Syy('Total_Stress_Syy','contours','Cauchy␣Total␣Stresses/node', 'SYY')



###############################         Summary
    ######################################################

def material(config):
    creatematerial_TotalStrain(config)
    creatematerial_Engineering(config)
    creategeometry(config)
    setelementclasstype(config)
    assignmaterial(config)
    assigngeometry(config)
    creatematerial_beam(config)
    creategeometry_beam(config)
    setelementclasstype_beam(config)
    assignmaterial_beam(config)
    assigngeometry_beam(config)
    Iteration_scheme(config)                              #Turn off for 2x2
        scheme



def supports(config):
    addsupportsets(config)
    edgesupport_right(config)
    edgesupport_left(config)
    bottomsupport(config)

def interface(config):
    creatematerial_interface(config)
    creategeometry_interface(config)
    createconnection_interface(config)

def loads(config):
    addset_loads(config)
    createdeadweight(config)
    #externalload_edge_all(config)
    #externalload_edge_single(config)

def prescribe_deformation(config):
    createfunction(config)   #ajust value
```

```
        deformation ( config )

def mesh ( config ) :
        create_mesh ( config )
        generate_mesh ( config )

def analysis ( config ) :
        addanalysis ( config )
        executeblock_deadweight ( config )
        executeblock_deform1 ( config )
        executeblock_deform2 ( config )
        define_output ( config )
        run_analysis ( config )

def model_export_TS ( config ) :
        export_displacements ( config )
        export_E1 ( config )
        export_S1 ( config )

def model_export_EMM ( config ) :
        export_displacements ( config )
        export_Exx ( config )
        export_Eyy ( config )
        export_E1 ( config )
        export_Sxx ( config )
        export_Syy ( config )

######################################################################## Output
        analysis


######## One model ##########

def One_model ( config ) :
        convert ( config )
        createNewModel ( config )
        geometry ( config )
        setViewPoint ( "TOP" )
        material ( config )
        supports ( config )
        interface ( config )
        loads ( config )
        prescribe_deformation ( config )
        mesh ( config )
        analysis ( config )
        model_export_EMM ( config )          #TotalStrain/EMM
        saveproject ( config )


######## 1st Analysis ##########

def Analysis1_TS ( config ) :
        config.home    =        'C:/ Users/NLMUND/Documents/Afstuderen/Resultaten/Analysis1/
            TotalStrain '
        config.input_material_model = 1       # TotalStrain = 1, Engineering = 2
```

```python
    y=0
    for i in [ config.masonry1 , config.masonry2 , config.masonry3 ]:     #material
        config.masonry = i
        config.boundaryvalues_deform = config.boundaryvalues_deform_1
        config.boundaryvalues_stress = [0 ,round(0.2+0.15*y,2) ,0 ,round(0.2+0.15*y
            ,2) ]
        config.boundaryvalues_strain = config.boundaryvalues_strain_1
        y = y+1
        for j in range (1 ,18):       #sigma
            convert ( config )
            x = j*500
            config.function_sigma = x
            config.map = "Material_%s/sigma_%s" %(y,j )
            config.filename = "model_%s.dpf" %(int(j*2))
            createNewModel ( config )
            geometry ( config )
            setViewPoint ("TOP")
            material ( config )
            supports ( config )
            interface ( config )
            loads ( config )
            prescribe_deformation ( config )
            mesh ( config )
            analysis ( config )
            model_export_TS ( config )                          #DEFINE EXPORT
            saveproject ( config )
            closeproject ( config )


def Analysis1_EMM ( config ):
    config.home    =     'C:/ Users/NLMUND/Documents/ Afstuderen/ Resultaten/ Analysis1 '
    config.input_material_model = 2       # TotalStrain = 1, Engineering = 2

    config.masonry = config.masonry1

    config.boundaryvalues_deform = config.boundaryvalues_deform_1
    config.boundaryvalues_stress = config.boundaryvalues_stress_1
    config.boundaryvalues_strain = config.boundaryvalues_strain_1
    for j in range (1 ,14):       #sigma
        convert ( config )
        x = j*167
        config.function_sigma = x
        config.map = "Material_%s" %(1)
        config.filename = "model_%s.dpf" %(int(j))
        createNewModel ( config )
        geometry ( config )
        setViewPoint ("TOP")
        material ( config )
        supports ( config )
        interface ( config )
        loads ( config )
        prescribe_deformation ( config )
        mesh ( config )
        analysis ( config )
        model_export_EMM ( config )                          #DEFINE EXPORT
        saveproject ( config )
```

```python
        closeproject(config)

def Analysis2_EMM(config):
    config.home    =      'C:/Users/NLMUND/Documents/Afstuderen/Resultaten/Analysis2'
    config.input_material_model = 2      # TotalStrain = 1, Engineering = 2
    config.masonry = config.masonry2

    config.boundaryvalues_deform = config.boundaryvalues_deform_2
    config.boundaryvalues_stress = config.boundaryvalues_stress_2
    config.boundaryvalues_strain = config.boundaryvalues_strain_2
    for j in range(1,14):        #sigma
        convert(config)
        x = j*167
        config.function_sigma = x
        config.map = "Material_%s" %(2)
        config.filename = "model_%s.dpf" %(int(j))
        createNewModel(config)
        geometry(config)
        setViewPoint("TOP")
        material(config)
        supports(config)
        interface(config)
        loads(config)
        prescribe_deformation(config)
        mesh(config)
        analysis(config)
        model_export_EMM(config)                                #DEFINE EXPORT
        saveproject(config)
        closeproject(config)

def Analysis3_EMM(config):
    config.home    =      'C:/Users/NLMUND/Documents/Afstuderen/Resultaten/Analysis3'
    config.input_material_model = 2      # TotalStrain = 1, Engineering = 2
    config.masonry = config.masonry3

    config.boundaryvalues_deform = config.boundaryvalues_deform_3
    config.boundaryvalues_stress = config.boundaryvalues_stress_3
    config.boundaryvalues_strain = config.boundaryvalues_strain_3
    for j in range(1,14):        #sigma
        convert(config)
        x = j*167
        config.function_sigma = x
        config.map = "Material_%s/" %(3)
        config.filename = "model_%s.dpf" %(int(j))
        createNewModel(config)
        geometry(config)
        setViewPoint("TOP")
        material(config)
        supports(config)
        interface(config)
        loads(config)
        prescribe_deformation(config)
        mesh(config)
        analysis(config)
        model_export_EMM(config)                                #DEFINE EXPORT
        saveproject(config)
```

```
        closeproject ( config )
```

```
################################    Commands
############################################################################
```

```
config.home   =       'C:/Users/NLMUND/Documents/Afstuderen/Resultaten/Analysis1/
     Engineering'
config.map = "Material_3/sigma_10"
config.filename = "model_10.dpf"
Version = 1

config.analysis = 'Analysis1'



#

    Define properties
#Geometry block [height, width, number of elements, name]    #NAME MUST BE MASONRY1
config.block_1 = [2000, 500, 40, 'Masonry1']
config.block_2 = [2000, 1000, 20, 'Masonry1']
config.block_3 = [2000, 800, 25, 'Masonry1']

#Material  [Exx, Eyy, ftx, fty, Gftx, Gfty]
config.masonry1 = [1250, 2500, 0.0167, 0.05, 0.005, 0.005]
config.masonry2 = [2500, 5000, 0.033, 0.1, 0.01, 0.01]
config.masonry3 = [3750, 7500, 0.05, 0.15, 0.015, 0.015]


#Constant parameters [rho, poisson, fc, Gfc, G, fs, c, Gfs, res_strength]
config.masonry_constant_1 = [1.8e-9, 0, 8.5, 20, 2000, 0.35, 0.3, 0.1, 0.001]

#Steelbeam
config.steelbeam_1 = [100, 210000, 0.3, 7.5e-9] # [height, Young, Poisson, density
     ]

#loadinput [name, value, direction]
config.load_all_1 = ['Distributed␣load', -12, 2]

#Loadinput_single [name, value, direction, position, number of single loads]
config.load_single_1 = ["Single",-100, 2, 3, 4]


#Input function [mu, sigma, stepsize]
config.function_1 = [10000, 1600, 10]
config.function_2 = [10000, 1500, 10]
config.function_3 = [10000, 5000, 10]


#output function [x-min, x-max, y-min, y-max]
config.boundaryvalues_deform_1 = [-4,4,-25, 0]
config.boundaryvalues_deform_2 = [-4,4,-25, 0]
config.boundaryvalues_deform_3 = [-4,4,-25, 0]
```

```
config.boundaryvalues_stress_1 = [0,0.05,0,0.05]
config.boundaryvalues_stress_2 = [0,0.1,0,0.1]
config.boundaryvalues_stress_3 = [0,0.15,0,0.15]
config.boundaryvalues_strain_1 = [0,0.001,0,0.001]
config.boundaryvalues_strain_2 = [0,0.001,0,0.001]
config.boundaryvalues_strain_3 = [0,0.001,0,0.001]

#
    ##############################################################################################
        Choose properties

config.block = config.block_3
config.masonry = config.masonry1

config.masonry_constant = config.masonry_constant_1
config.input_material_model = 2     # TotalStrain = 1, Engineering = 2

config.steelbeam = config.steelbeam_1

config.load_all = config.load_all_1
config.load_single = config.load_single_1

config.function = config.function_3

config.boundaryvalues_deform = config.boundaryvalues_deform_1
config.boundaryvalues_stress = config.boundaryvalues_stress_1
config.boundaryvalues_strain = config.boundaryvalues_strain_1

#Linear
convert(config)


################################################### mesh parameters
config.mesh_elementsize = 100 #mm
config.max_iterations = 500
config.steps_1 = '0.001(200)'
config.steps_2 = '0.001(1)'


#################################################### Output parameters

config.nodes = [(int(config.block_elements*config.block_w/2.0), int(config.block_h
    ), 0)]
config.elements = [(int(config.block_elements*config.block_w/2.0), int(config.
    block_h), 0)]

config.png_W = 700
config.png_H = 300


################################################## PERFORM ANALYSIS
    ##############################################################################################

One_model(config)
Analysis1_TS(config)
```

Analysis1_EMM (config)
Analysis2_EMM (config)
Analysis3_EMM (config)