

Parametric design-tool to optimize preliminary design of navigation lock chambers

L. van Olst

Master of Science Thesis



Cover picture: Navigation Lock Complex Lith, Rijkswaterstaat (Joop van Houdt)

Parametric design-tool to optimize preliminary design of navigation lock chambers

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Hydraulic Engineering at Delft
University of Technology

L. van Olst

July 3, 2019

Student number	: 4247000	
Thesis committee	: Prof. dr. ir. S.N. Jonkman	TU Delft, committee chair
	Ir. W.F. Molenaar	TU Delft, daily supervisor
	Prof. dr. ir. M. van Koningsveld	TU Delft
	Ir. M.C. Booi	Royal HaskoningDHV

Faculty of Civil Engineering and Geosciences (CiTG) · Delft University of Technology



The work in this thesis was conducted on behalf of, and carried out in cooperation with Royal HaskoningDHV.



Copyright © Department of Hydraulic Engineering
All rights reserved.

Preface

This thesis is written as the final stage in the process of obtaining my degree as Master of Science in the field of Hydraulic Engineering at the faculty of Civil Engineering and Geosciences at the Delft University of Technology. This Graduation Project was realized in a cooperation between TU Delft and Royal HaskoningDHV.

To start off, I would like to thank my professor, Bas Jonkman, for being chair of my graduation committee, leading the committee meetings and helping me with his critical feedback during these meetings and on the report.

Furthermore, I would like to thank my daily supervisor, Wilfred Molenaar, for his support throughout my entire graduation process. Helping me with the structure of the report, his knowledge about the design process and the interesting discussions we had on the digital way of working within Hydraulic Engineering.

In addition, I would like to express my thanks to my third committee member, Mark van Koningsveld, for his knowledge about digital engineering. Despite his late introduction in my thesis committee, he gave me more insight in the digital environment and the way of coding in Python. The joint 'coding labs' with other graduate students were inspiring and informative.

Finally, I would like to thank Maarten Booij and all the other staff from the Maritime and Aviation department of Royal HaskoningDHV for providing me with their knowledge. Maarten helped me with the setup of my graduation topic, guiding the process and the frequent reviewing of my report.

And of course, a thank you for my friends and family for their support. Because of the great time they gave me, I could easily focus on my thesis. A special thanks to my grandfather Bram Gerbers for passing the hydraulic engineering genes.

Bart van Olst

Rotterdam, July 2019

Summary

In the twentieth century a substantial number of navigation locks were built in the Netherlands. The navigation locks built in that era mostly had a life span of 80 to 100 years. This means that in the coming years most of these locks will be in need of either renovation or replacement. Rijkswaterstaat, the manager of most of the locks in the Netherlands, is likely to put out tenders for the renovation and replacement of these locks. Companies can subsequently submit a proposal for design and/or construction, which includes preliminary design of the lock. During this phase of the design multiple possible alternatives will be reviewed to check which of those satisfy the design requirements and how they will be implemented. The time available to create a proposal is limited and the preliminary design phase is time consuming due to the numerous design alternatives available. Therefore, several alternatives are often not elaborated, which already leads to converging in the design process in an early phase. When the time needed to design the alternatives can be reduced, more attention can be paid to out-of-the-box thinking during preliminary design. This time saving can be accomplished by the use of automation and parametric design. With parametric design all the possible alternatives can be calculated in a given range of parameters. After comparison of these generated alternatives a well-educated choice can be made for the direction in preliminary design. This research aims to optimize preliminary lock design in the tendering phase by creating a design-tool that automatically generates all plausible alternatives based on the design criteria and boundary conditions of the client.

The focus of the design tool will be on the lock chamber. Eventually, the complete preliminary lock design can be accomplished with a parametric design tool. However, an advantage of a design tool is that it can be composed of different building blocks which all design a specific part of the navigation lock. The basis of the parametric design tool should be a relevant part of the lock when looking at the total costs of this part, the design costs, the variety of options and the relation with other parts. The lock chamber is chosen as a starting point for the design tool. The lock chamber, like the lock head and the mechanics, has high investment costs and a high percentage of design costs. The lock chamber however, is in contrast to the other parts of the lock, not heavily dependent on other components. This makes it an interesting component for the creation of parametric design tool.

The parametric model, that has been developed as a part of this thesis, includes the commonly used chamber types for inland navigation locks. These common types are retaining wall structures with an under water concrete floor including sheet pile, combi and diaphragm

walls, and a U-basin structure, which can be constructed in an open building pit or in a temporary retaining wall building pit.

When the design tool is finished, research is conducted on the behaviour of the alternatives in different circumstances. The effects of the main design criteria are examined. These include the length of the chamber, width, depth, water head and local soil profile. These values are mostly depended on the vessels that need to be accommodated and the local circumstances. Next to that, the impact of different construction methods, the use of materials and the variation in unit prices are elaborated.

In this thesis the second Juliana lock near Gouda is used as reference case. For this case it has been found that a sheet pile wall structure is economically most attractive, followed by the U-basin with an open building pit, a diaphragm wall structure and least feasible a U-basin structure with a retaining wall building pit. It is concluded that the length of the lock does hardly impact the design of the alternatives. The width of the chamber does influence the dimensions of the structure and the optimal choice. The floor has to be stronger for larger widths and, because of this, the diaphragm wall structure will become more attractive compared to the U-basin structures. The depth and water head mainly impacts the dimensions of the walls. This, combined with the increased soil handling with open building pits, leads to decreasing interest in U-basin structures. When looking at the construction methods and materials, it can be noted that the length of the tension piles is especially important for the pile plan and the floor. The concrete strength for the U-basins is of importance, while the under water concrete strength and the casting method hardly influence the total costs.

When looking at environmental impact with the 'Environmental Cost Indicator', the alternative with the least material use is most attractive, the sheet pile wall structure followed, by the diaphragm wall structure. The type of material does heavily influence the environmental impact. The U-basin structures have more environmental impact due to the construction of a larger temporary building pit. The environmental impact consists of three main components, the used materials, transport of these materials and the processes for the construction of the lock. For the U-basin structures the last two components have a large impact, while for the retaining wall structures the materials used have a large footprint.

The proposed CO₂ taxes of €0,05 and €0,20 per kg CO₂ equivalent will lead to an average cost increase of respectively 4,5% and 16.5%. The impact per alternative differs, but does not influence the order of economical feasibility. As from a CO₂ tax of €0,68 per kg CO₂ equivalent the diaphragm wall structure will become financially more attractive compared to the U-basin structure with open building pit.

Despite the calculated strengths and costs, each structure has its own advantages, disadvantages and limitations in construction, final use and maintenance. Special attention should be paid to these properties, since certain construction methods may be excluded by design requirements and uncertainties can lead to high risk. Further research could focus on steel sheet or combi piles constructed in bentonite excavations, resolving the limitations of vibrations during sheet pile construction and the uncertainties of diaphragm walls.

Contents

Preface	i
Summary	iii
1 Introduction	1
1-1 Report Structure	1
2 Background of Navigation Locks	3
2-1 Components of the lock	4
2-1-1 Gates	5
2-1-2 Lock Head	5
2-1-3 Chamber	6
2-1-4 Filling and Emptying System	12
2-1-5 Salt Water versus Fresh Water	12
2-1-6 Approach Area	13
2-2 Navigation	13
2-2-1 Ships	14
2-2-2 Ship Mooring Loads	15
2-2-3 Future	16
2-3 Availability and Related Costs	17
2-3-1 Climate Change	17

3	Parametric Design	19
3-1	Definition of 'Optimum Design'	20
3-1-1	Environmental Impact	21
3-2	Demand for a Design-Tool	21
3-3	Possibilities for Automation	23
3-3-1	Mechanics	23
3-3-2	Lock Head	23
3-3-3	Lock Chamber	24
3-4	Costs within the Design-tool	24
4	Problem Definition and Research Objectives	25
4-1	Problem Definition	25
4-2	Research Objective	26
4-2-1	Research Questions	27
4-3	Research Method	27
5	Model Description	31
5-1	Model Overview	32
5-2	Model Analysis	33
5-2-1	User Input	33
5-2-2	Default Data	33
5-2-3	General Dimensions	33
5-2-4	General Calculations	34
5-2-5	Retaining Wall	37
5-2-6	U-Basin	39
5-2-7	Output	41
5-2-8	Costs, MKI-Values and CO2	41
5-3	Calculation Methods Model	41
6	Dimensions of the Lock	45
6-1	Horizontal Dimensions	45
6-2	Vertical Dimensions	46
6-2-1	Water Levels	46
6-2-2	Capstone Height	47
6-2-3	Gross Keel Clearance	47
6-3	Lock Area	48
6-3-1	Surcharge Load	48

7	Concrete U-Basin	49
7-1	Construction	49
7-1-1	Open Building Pit	49
7-1-2	Open Building Pit with Cut-off Screens	50
7-1-3	Sheet Pile Building Pit	50
7-1-4	Pneumatic Caisson	51
7-2	Dimensions	51
7-2-1	Walls	51
7-2-2	Floor	52
7-2-3	Concrete	53
7-2-4	Reinforcement	54
7-3	Soil Loads	55
8	Sheet Pile Wall Structure	57
8-1	D-Sheet Piling	57
8-2	Sheet piles	58
8-2-1	Profiles	58
8-2-2	Combi Walls	58
8-2-3	Modification Factor	59
8-2-4	Interlock reduction factor	59
8-2-5	Reduction Factor Moment Capacity	59
8-2-6	Material Factor	59
8-2-7	Delta Friction Angle	59
8-3	Anchors and Struts	60
8-3-1	Grout Anchors	60
8-4	Underwater Concrete	61
8-4-1	CUR 77:2014	61
8-4-2	Concrete	62
8-5	Conservation	62
8-6	Tension Piles	63
8-6-1	Pile Alignment	63
8-6-2	Axial spring stiffness	64
8-7	Capping Beam	65
8-8	Construction	67

9	Diaphragm Wall Structure	69
9-1	Stability	70
9-1-1	Final Stability	70
9-1-2	Stability during construction	70
9-2	Reinforcement	71
9-2-1	Vertical Reinforcement	71
9-2-2	Horizontal Reinforcement	71
9-2-3	Cages	71
9-2-4	Concrete Cover	71
9-2-5	Final Amount of Reinforcement	72
9-3	Resisting Moment	72
9-4	D-Sheet Piling - Diaphragm Wall	73
9-4-1	Diaphragm Walls	73
9-4-2	Anchors and Struts	73
9-4-3	Underwater Concrete and Tension Piles	73
9-5	Capping Beam	73
10	Costs	75
10-1	Cost Structure	75
10-2	Sheet Piles	76
10-2-1	Material	76
10-2-2	Installation	76
10-2-3	Recycling	76
10-2-4	Conservation	77
10-2-5	Mobilization	77
10-3	Diaphragm Wall	77
10-3-1	Materials	77
10-3-2	Installation	78
10-3-3	Mobilization	78
10-4	Concrete	78
10-4-1	Reinforcement Steel	78
10-4-2	Formwork	78
10-5	Soil Handling	78

10-5-1 Digging Work	78
10-5-2 Soil Import	79
10-5-3 Soil Removal	79
10-6 Anchors	79
10-7 Greenfield, Brownfield and Blackfield	79
11 Environmental Impact	81
11-1 Database	81
11-2 Global Warming	82
11-2-1 CO2 Tax	83
11-3 Materials and Processes	85
12 Case Study	87
12-1 Water levels	87
12-2 Vessels	87
12-3 Dimensions	88
12-4 Ground Profile	88
13 Validation	89
13-1 General Dimensions	89
13-1-1 Ship Dimensions	89
13-1-2 Lock Dimensions	90
13-2 General Calculations	90
13-2-1 Tension Piles	90
13-3 Retaining Wall	92
13-3-1 D-Sheet Calculation	92
13-3-2 Underwater Concrete Floor	93
13-4 U-Basin	95
13-4-1 Horizontal Soil Pressure	95
13-4-2 Walls and Floor	96
13-5 Findings	97

14 Results	99
14-1 Basic Case Juliana Lock	99
14-1-1 Sheet Pile Wall Structure	100
14-1-2 Diaphragm Wall Structure	101
14-1-3 U-Basin Open Pit	102
14-1-4 U-Basin Retaining Wall Pit	103
14-1-5 MKI	104
14-1-6 CO2 Tax	106
14-2 Change in Width	106
14-3 Change in Water Head	108
14-4 Change in Water Depth	109
14-5 Change of Soil	110
14-6 Change of Surcharge Load	112
14-7 Change of Piles	112
14-7-1 Pile Width	113
14-7-2 Pile Length	113
14-8 Different Under Water Concrete	114
14-8-1 Casting Method	115
14-9 Change Concrete Strength	115
14-10 Anchor Angle	117
14-11 Unit Prices	118
15 Discussion	119
15-1 Characteristics of the Alternatives	119
15-1-1 Sheet Pile Wall Structure	119
15-1-2 Diaphragm Wall Structure	119
15-1-3 U-Basin Open Pit	120
15-1-4 U-Basin Retaining Wall Pit	120
15-2 Cases	121
15-3 Model Discussion	121
15-3-1 Use of External Programs	122
15-4 User	123
15-5 Application of the Design-tool	123
15-5-1 When to use the Design-tool	124
15-5-2 When to Create a Design-tool	124

16 Conclusion and Recommendations	125
16-1 Conclusion	125
16-1-1 Influence of Main Dimensions	125
16-1-2 Environmental impact	126
16-1-3 CO2 Tax	126
16-1-4 Influence of unit prices	127
16-1-5 Creation of a Parametric Design-tool	127
16-1-6 Parameters within the Design-tool	128
16-2 Recommendations	128
16-2-1 Standardization	129
16-2-2 Sea Locks	129
16-2-3 Combi Walls in Bentonite Excavations	129
16-2-4 Open floors	130
Bibliography	131
List of Figures	137
List of Tables	143
List of Symbols	145
Appendices	I
A CEMT Classes	III
B Anchors	VII
C Environmental Classes	IX
D Lock Gates	XIII
E Profiles Sheetpile	XVII
F Python Code	XIX

Chapter 1

Introduction

In this thesis report the creation of a parametric design tool for navigation locks will be elaborated. A study on navigation locks and parametric design will be conducted to check which component of the navigation lock is most suitable for automation and creation of a parametric design tool. The different alternatives which are implemented in the design tool will be elaborated to make them suitable for automation. With the design tool results will be created to investigate the dependency of different dimensions, requirements, processes or materials on the different design alternatives.

1-1 Report Structure

The structure of the report is visualized in Figure 1-1. After this introduction the background of the problem is given in Chapter 2 and 3. Based on these chapters the Problem Definition and Research Objectives are given in Chapter 4. In Chapter 5 the design model will be explained. In the next part the input of the model will be defined, Chapter 6 gives the overall dimensions of the lock which will be used for the alternatives which are given in Chapter 7, 8 and 9. Since costs are taken into account in the design tool a cost description is given in chapter 10. To link the design alternatives with environmental impact an introduction on this topic is given in Chapter 11. A case study will be introduced in Chapter 12. This case study will be used to check the validation of the design tool in Chapter 13. In Chapter 14 the results from the model will be given which will be discussed in Chapter 15. At last the Conclusion and Recommendations of the thesis are written in Chapter 16.

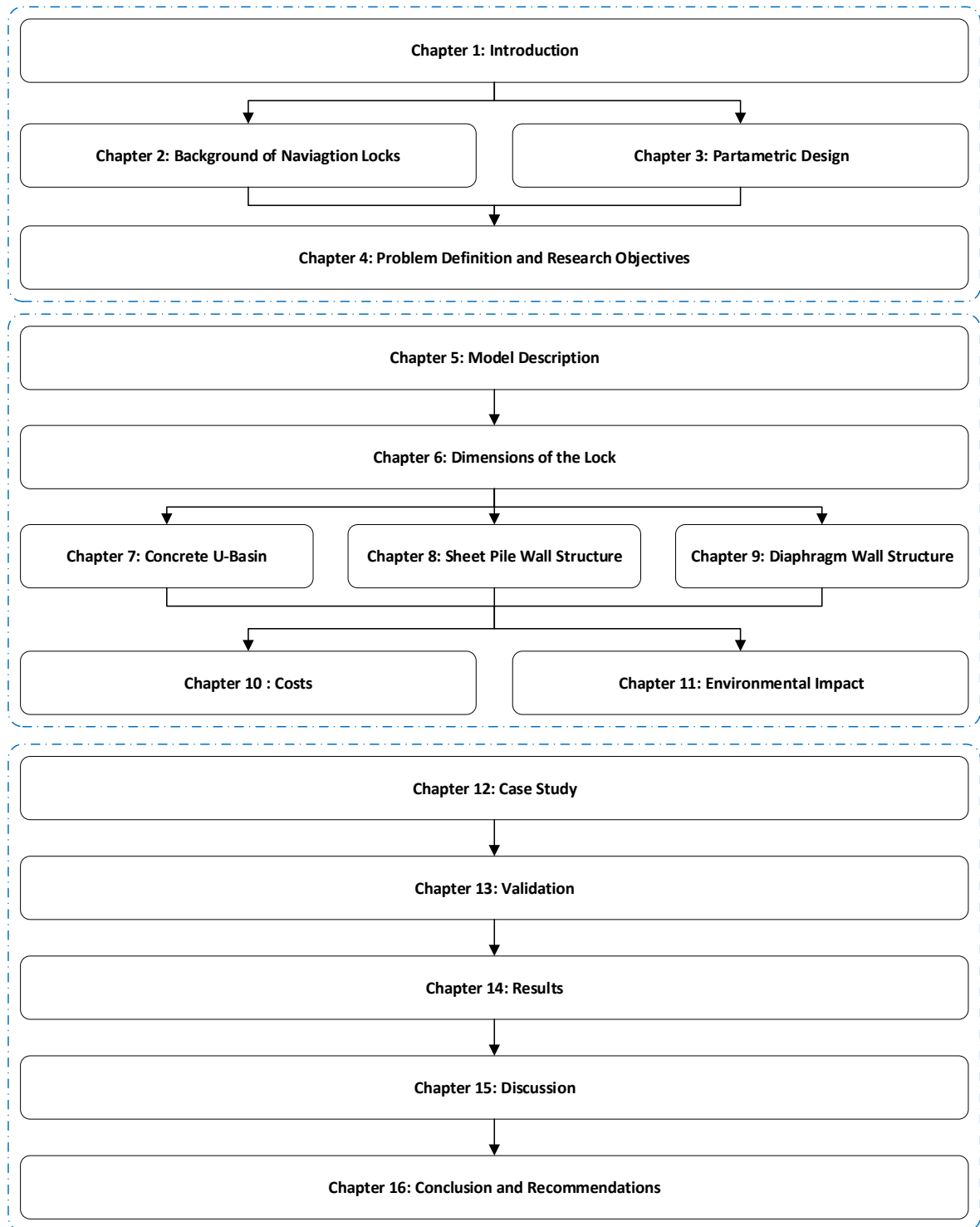


Figure 1-1: Report Structure

Chapter 2

Background of Navigation Locks

Since the year of 983 navigation locks are built to connect waterways with different water levels (Erbisti, 2002). This was first needed when people were in need of larger navigation depths in rivers and they started to build dams and weirs. A larger navigation depth was accomplished but a connection for ships to overcome the height difference became the next problem. Later channels were dug to connect different water reaches, mostly having different water levels to overcome. Also flood protection is a reason to close off rivers and estuaries and locks are needed to enable navigation through this closure.

A main distinction is made between inland navigation locks and maritime navigation locks. Inland navigation locks are connecting waterways and make sure a certain navigation depth in these waterways is preserved. Inland navigation locks are designed to ensure a large amount of ships can pass with as little delay as possible to reduce transport costs. The head difference of inland navigation locks could be high, increasing when going upstream.

Maritime navigation locks are mostly used to connect the sea with a port basin. A lock could be placed to retain a certain water level within the port basin to enable berthing of large vessels. Also locks are placed as part of a storm surge barrier or sea defence. Next to that, locks can be needed to limit the inflow of salt water. Due to the tidal range of the sea maritime navigation locks do often need to retain water from both sides which require a different design.

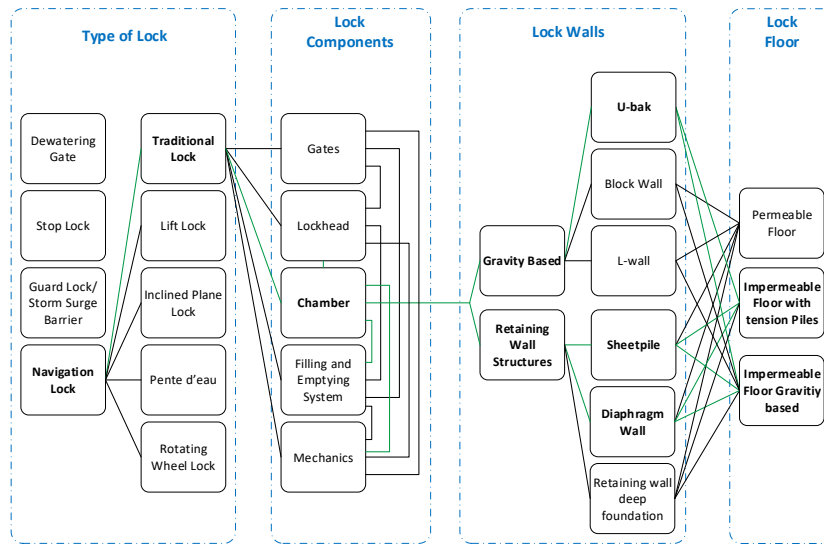


Figure 2-1: Relations in a Lock

2-1 Components of the lock

There are different types of locks as can be seen in Figure 2-1, in this thesis the focus will be on navigation locks. Different types of navigation locks occur all over the world depending on the typical circumstances in the area. Worldwide but also in Europa and the Netherlands the traditional lock is the most widely used navigation lock and the components will be discussed in the following section (Figure 2-2). In Chapter 4 it will be concluded to further elaborate the lock chamber. For this reason, in the following sections, more attention is paid to the lock chamber compared to the other lock components.

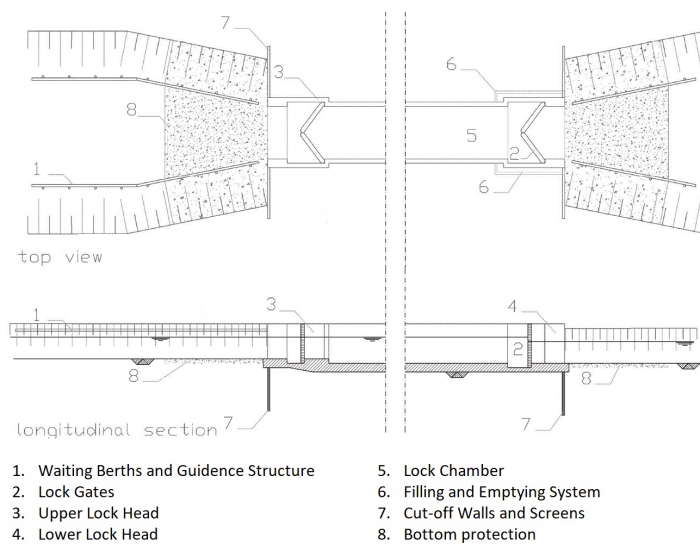


Figure 2-2: Components of a Lock

2-1-1 Gates

There are different types of gates to close the lock. The choice of the gate type for a particular lock depends on the head difference, width of the lock, needed clearance, operation time, available space, type of intake and discharge system, sediment quantities and costs. Multiple studies were performed to decide which type of gate is best for different situations. In Appendix D the common type of gates are discussed in more detail.

2-1-2 Lock Head



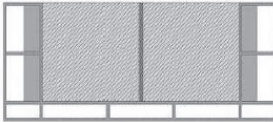


According to the Hydraulic Structures manual on locks (Molenaar, 2011) the lock head fulfills three main functions.

1. Accommodation for the gate and all the accompanying equipment to operate the gates.
2. An important function of a lock is to retain water. A lock head has to be high enough to withstand flooding, ensure a good sealing with the gates and also seepage need to be prevented for example by using cut-off screens.
3. The lock head has to be designed to resist the loads of the gates originating from the water level differences.

Apart from this, there should be recesses for stop-logs to allow maintenance. Also the choice for the right type of filling and emptying system which is used requires some attention, this will be discussed in section 2-1-4.

Several types of lock heads are used to fulfill these requirements, some options are given in Table 2-1 below.

Table 2-1: Different types of Lock Heads (Molenaar, 2011)

	Description		Application area
1	Standard U-shaped lock head build in-situ		This type of lock head is still used frequently, because it is simple to construct and it is a very rigid structure. All the other alternatives are based on this U-shaped structure.
2	Pneumatically immersed lock head		The lock head is constructed at ground level and pneumatically immersed to its operational depth. The construction alternative is used when only little horizontal area is available (no building pit needed).
3	Stripped U-shaped head, floated in		The stripped lock head is built in a dry dock, then transported and immersed on a prepared gravel bed at its final destination. After ballasting with concrete the head will be finished. This alternative will be applied when building activity at the lock location is not desired and a short construction period is required.
4	U-shape lock head with a construction joint in the middle		This lock head is only applied for very large sea locks, thus only when the width of the floor is relatively large compared to the wall height. Apart from lower internal forces the hinge in the floor (only) has disadvantages regarding construction, different rotations at the joint, leakages etc..
5	U-shape lock head with two construction joints		This alternative is a variation of alternative 4, the difference is the use of two joints instead of one. The same (dis)advantages apply.

Construction Method

The type of the lock head chosen also depends on the location where the lock will be constructed and the time available. Is a lot of space available to construct a building pit? Should the lock head be built at a different site? An important part of the costs for the lock head are within the construction method. Type of lock heads which can be constructed and how are given in Table 2-5 and Table 2-6.

2-1-3 Chamber

The main function of the chamber is to retain varying water levels and ships. For the ships safe mooring is required during leveling. The horizontal dimensions are mostly determined by the dimensions of the normative ship. The vertical dimensions are determined by the the maximum head difference and the draught of the normative ship and the water levels. Alternatives for chamber types are given in Table 2-2, Table 2-3 and Table 2-4.

Walls

The main functions of the walls are to retain the soil and water, guidance of the vessels in the lock and mooring of the vessel. The choice of which wall to implement depends on the subsoil present, the type of ships to be accommodated, the water level differences, the need of a water tight floor, the option for emptying the chamber for maintenance and the available space for construction.

Three types of walls will be introduced here. The first one is the gravity structure. As the name suggests, gravity structures rely on the gravity. This gravity force is caused by the own weight of the structure and the vertical force of the soil above. To be stable the structure should be heavy or wide enough to prevent sliding or overturning caused by the horizontal forces of the soil, water and ships. Examples of gravity walls are: L-walls, block walls and caissons.

Secondly a variety of retaining wall structures are commonly used. This category includes sheet pile walls, diaphragm walls and combi walls. These kind of walls are installed deeper in the ground compared to the floor of the lock and rely on flexural resistance to retain the soil. The horizontal forces on the active side of the wall are compensated by passive soil pressure at the deeper end of the wall. When this is not enough a soil anchor could be placed to obtain stability or to lower the moment in the wall. Special attention should be given to the water tightness between the sheet piles, water tightness between the sheet piles and the floor and the quality of the backfill (PIANC, 2009).

A more expensive solution is a wall which makes use of a deep foundation as can be seen in Figure 2-3. This solution is applicable when the bearing layers of the soil are located deep in the ground and when a large chamber depth is required. Most of the horizontal and vertical loads are transferred to the piles while the sheet pile wall particularly has a soil and water retaining function (De Gijt and Broeken, 2013).

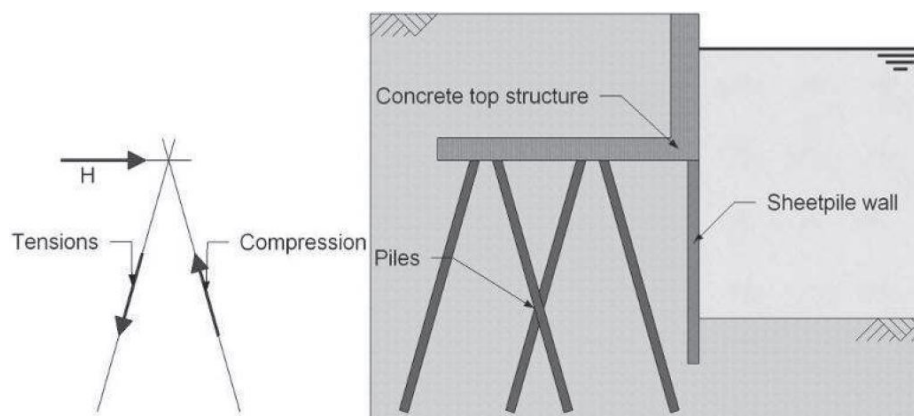


Figure 2-3: Wall with Deep Foundation (Molenaar, 2011)

Floor

A distinction is made between a impermeable floor and a permeable floor. An advantage of a impermeable floor is that it is water retaining, no leakage occurs which results in no spilling

of water or salt intrusion and the possibility of dewatering the chamber. Another advantage is that the mostly stiff floor contributes to the load transfer of the walls. A disadvantage is that water pressure can build up against the floor during water level fluctuations and dewatering of the chamber. The floor should be strong enough to withstand these forces or piles should be applied to reduce the moments and prevent flotation of the structure.

A permeable floor does not have the problem with build-up of water pressures. However, it should be checked when the floor is less permeable than the soil layer underneath. When the floor is more permeable the floor should be checked on stability (Molenaar, 2011). Also the stability should be checked for water motions in the lock caused by ships and filling and emptying.

It is also plausible to combine the advantage of the stiffness of the impermeable floor with the lack of pressure build-up from permeable floor. This can be done by making holes in the stiff impermeable floor and by constructing a filter in these holes to allow drainage.

Construction Method

The applied type of floor depends on the available construction method at the project location. The alternatives concerning the construction method are given in Table 2-5 and Table 2-6.

Table 2-2: Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)


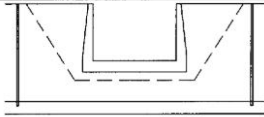
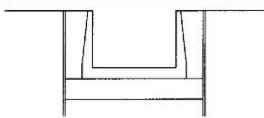
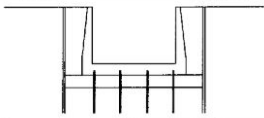
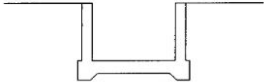
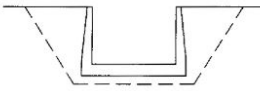
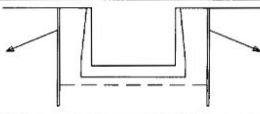
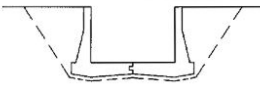
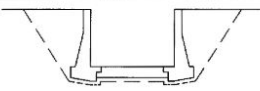
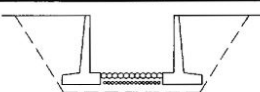
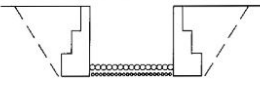
NUMBER OF ALTERNATIVE	CHARACTERISTICS		Suitable for		Suitable for lock size			
	Final construction	Building stage	Chamber	Head	Small	(Medium) large	Very large	
1		Reinforced concrete monolith (possibly with pile foundation)	Construction pit, pumping	X	X	X	X	X
2		As 1	Construction pit, bentonite concrete screens into the impermeable layer	X	X	X	X	X
3		As 1 (no pile foundation)	Sheet pile pit, thick underwater concrete floor	X	X	X	X	X
4		As 1 (always pile foundation)	Sheet pile pit, thick underwater concrete floor, tension piling	X	X	X	X	X
5		As 1	Pneumatic caisson, built on ground level	X	X	X	X	

Table 2-3: Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)

NUMBER OF ALTERNATIVE	CHARACTERISTICS		Suitable for		Suitable for lock size		
	Final construction	Building stage	Chamber	Head	Small	(Medium) large	Very large
6		Final construction Reinforced concrete monolith (on steel foundation)	Prefabricated elsewhere, floating transport, sunk into cunet with slopes	X	X	X	X
7		As 6	As 6 but the cunet is bordered by anchored sheet pile	X	X	X	X
8		Reinforced concrete construction with one hinge	Construction pit, pumping	X	X		X
9		Reinforced concrete construction with two hinges	Construction pit, pumping	X	X		X
10		Reinforced concrete L walls with rubble layer on filter	Construction pit, pumping	X		X	X
11		Terre armee walls, rubble layer on filter	Construction pit, pumping	X		X	X

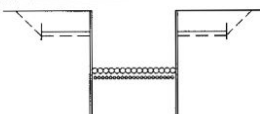
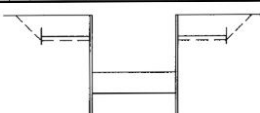
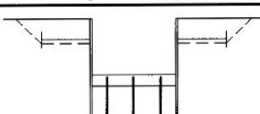
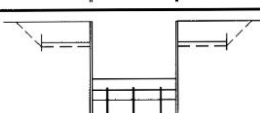
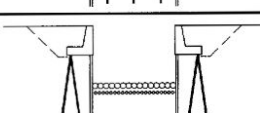
NUMBER OF ALTERNATIVE	CHARACTERISTICS		Suitable for		Suitable for lock size		
	Final construction	Building stage	Chamber	Head	Small	(Medium) large	Very large
12		Anchored dam / deep walls provided with wood fender. Rubble layer on filter	Limited, dry excavation. Rest excavation and filter, rubble layer in the wet	X		X	X
13		Anchored dam / deep walls provided with wood fender. Thick underwater concrete floor	Limited, dry excavation. Rest excavation and underwater concrete floor in the wet	X		X	X
14		Anchored dam / deep walls provided with wood fender. Thin underwater concrete floor (possibly reinforced) anchored to tension piling	Limited, dry excavation. Rest excavation and pile driving and underwater concrete floor in the wet	X		X	X
15		Anchored dam / deep walls provided with wood fender. Constructive reinforced concrete floor	Limited, dry excavation. Rest excavation and pile and underwater concrete floor in the wet. Construction of floor in the dry	X		X	X
16		Reinforced concrete retaining wall (including relieving plate) on pile and deep wall. Rubble layer on filter	Limited dry excavation. Pile and sheet pile driving from bottom dry excavation. Rest excavation and filter /rubble layer in the wet	X		X	X

Table 2-4: Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)


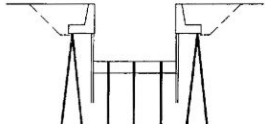

NUMBER OF ALTERNATIVE	CHARACTERISTICS		Suitable for		Suitable for lock size		
	Final construction	Building stage	Chamber	Head	Small	(Medium) large	Very large
17		Reinforced concrete retaining wall (including relieving plate) on piles and deep wall. Thick underwater concrete floor	Limited dry excavation. Pile and sheet pile driving from bottom dry excavation. Rest excavation and floor in the wet	X		X	X
18		Reinforced concrete retaining wall (including relieving plate) on piles and deep wall. Thin underwater concrete floor (possibly reinforced) anchored to tension piling	Limited dry excavation. Pile and sheet pile driving under retaining wall from bottom dry excavation. Rest excavation, tension piling and floor in the wet	X		X	X
19		Green chamber between slopes, provided with bank and bottom lining and guide structures	Wet excavation (upper part possibly in the dry). Underwater lining placed in the wet, dry finishing upper parts of slopes. Piles can be driven while floating. Finishing of guide structure above water line	X		X	X

Table 2-5: Alternatives for lock heads and lock chamber regarding construction method (Vrijburcht et al., 2000)

ALTERNATIVE		2a	2b	2c	3	K	Remarks
NR	Characteristics	Requires pumping	Limited pumping required	Combination pumping with return pumping conceivable	Limited execution width	Chamber can be laid open and floor is closed	
1	Monolith, pumped construction pit	X		X		X	
2	Monolith, screens into the impermeable layer		X	X		X	X (2b) could be sensible in order to make use of a layer that is too high. X (2c) is very conceivable in that case.
3	Monolith, sheet piling pit with unanchored underwater concrete floor		X	X	X	X	X (2b) could be sensible in order to make use of a layer that is too high. X (2c) is very conceivable in that case
4	Monolith, sheet piling pit with underwater concrete floor and tension piling				X	X	
5	Monolith, pneumatic caisson				X	X	
6	Monolith, prefabricated elements sunk in cunet	*				X	* = Probably no pumping, probable pumping at dock
7	Monolith, prefabricated elements sunk in cunet with sheet pile	*			X	X	* = Probably no pumping, probable pumping at dock
8	Construction with 1 concrete hinge, pumped construction pit	X		X		X	

Table 2-6: Alternatives for lock heads and lock chamber regarding construction method (Vrijburcht et al., 2000)

ALTERNATIVE		2a	2b	2c	3	K	Remarks
NR	Characteristics	Requires pumping	Limited pumping required	Combination pumping with return pumping conceivable	Limited execution width	Chamber can be laid open and floor is closed	
17	Reinforced concrete L wall on piling and sheet piling and thick underwater concrete floor		X	X	X	X	As in 16
18	Reinforced concrete L wall on piling and sheet piling and thin underwater concrete floor with tension piling		X	X	X	X	As in 16
19	Green chamber between slopes, provided with bank and bottom lining and guide structures						

ALTERNATIVE		2a	2b	2c	3	K	Remarks
NR	Characteristics	Requires pumping	Limited pumping required	Combination pumping with return pumping conceivable	Limited execution width	Chamber can be laid open and floor is closed	
9	Construction with 2 concrete hinges, pumped construction pit	X		X		X	
10	Reinforced concrete L walls, rubble layer, pumped construction pit	X		X			
11	Terre amee walls, rubble layer, pumped construction pit	X		X			
12	Anchored sheet piling, rubble layer		X	X	X*		X (2b) could be favourable for limitations in sheet piling moments. X (2c) is very conceivable in this case. * = required width depending on anchor walls or grout/ground anchors
13	Anchored sheet piling, thick underwater concrete floor		X	X	X*	X	As in 12
14	Anchored sheet piling, underwater concrete floor with tension piling		X	X	X*	X	As in 12
15	Anchored sheet piling, underwater concrete floor with tension piling, constructive floor		X	X	X*	X	As in 12
16	Reinforced concrete L wall on piling and sheet piling and rubble layer		X	X	X		X (2b) could be favourable to avoid wood fender, obtain shorter sheet piling and limit sheet piling corrosion. X (2c) is highly conceivable in this case.

2-1-4 Filling and Emptying System

For leveling the lock water should leave or enter the lock. The oldest way to accomplish this is by partly opening the gates. This should be done with care to limit the turbulence in the lock chamber. An other widely used option is head filling. This can be leveling by valves in the gates or by short culvert systems around or through the lock head when valves in the gate are not feasible. With this solution water can enter or leave the dock more gently compared to opening of the gates. This option is generally used for locks up to a lift of 6 meters. To reduce turbulence, breaker plates can be installed at the valves or stilling chambers can be introduced where energy dissipation takes place. When even less turbulence is required, or shorter levelling times are required due to large lift, longitudinal culverts can be used. Multiple types of longitudinal culvert systems are used. PIANC (2009) made a distinction of complexity as given in Table 2-7.

For the design tool it is assumed that filling will happen through the lock head or through the gates. Longitudinal culverts are hardly used in the inland waterway locks in the Netherlands due to the limited lifts.

Table 2-7: Longitudinal culvert systems (PIANC, 2009)

Complexity	Hydraulic Systems
1. Simple longitudinal culvert systems	- Wall culvert side port system - In Chamber longitudinal culvert system (ILCS)
2. More complex longitudinal culvert systems	- Wall culvert bottom lateral system - Longitudinal culverts under the lock floor - Wall culvert bottom longitudinal system
3. Very complex longitudinal culvert systems	- Dynamically balanced lock filling system - Pressure chamber under the floor

2-1-5 Salt Water versus Fresh Water

In coastal areas a navigation lock might connect a fresh water body with a salt water body. When the gates open and during leveling, an exchange of water takes place. In some locations, this poses two problems, salt water intrusion or fresh water loss.

Salt Water Intrusion

When the salt water head is higher than the fresh water head, salt water flows into the fresh water body during leveling. Also when opening the gate water from the chamber mixes with the abutting water body. This results in salt water entering the fresh water body. Because salt water has a larger density compared to fresh water the salt water will not mix with the fresh water but will mostly sink to the bottom and could travel, underneath the fresh water, upstream.

This salt water intrusion leads to problems when the water is used for drinking water or for agricultural purposes. An other problem occurs when the fresh ground water is replaced by the salt water, this could also lead to environmental problems (PIANC, 2009).

Fresh Water Loss

In the opposite situation, when the fresh water level is higher than the salt water level, fresh water leaves the body to the salt water body during levelling. Fresh water is often used for agricultural purposes, drinking water and is also needed to retain the water level for ships to navigate. This fresh water can be limited due to a high water extraction or due to a limited supply caused by drought.

2-1-6 Approach Area

To allow for safe navigation and safe waiting for lock availability, the approach area should be designed with care. The layout of the approach depends on the type and amount of ships which pass the lock, the available space, the amount of locks and the wind and waves present in the area. When high waiting times are expected, waiting areas should be present for the ships to moor. The design guide of the approach are written by Molenaar (2011) and Vrijburcht et al. (2000).

2-2 Navigation

The main global type of transport is deep sea transport in between deep sea ports all around the world. If this cargo arrives at a port and is again loaded onto another deep sea ship it is called transshipment. Another option is that the cargo has to be transported to the hinterland, this is called transloading to inland transport. When looking at inland cargo transport a distinction is made between railway transport, road transport, inland waterway transport and transport through pipelines. The way the types of transloading are related to each other in percentages is called the Modal Split. The Modal Split of the Port of Rotterdam from 2017 is given in Figure 2-4. In 2017 the total inland waterway transport in the Netherlands was good for 34% of the total transloading to inland transport (Binnenvaartcijfers, 2018). An explanation of this large market share of inland waterway shipment is the good economic feasibility compared to the other types of transport, due to the large numbers of cargo which can be transported on a ship. Another advantage of inland waterway transport is the relatively low amount of CO² which is produced during transport. Due to the current concern about climate change this is a reason why ports are looking for a shift to more transloading to inland waterway transport.

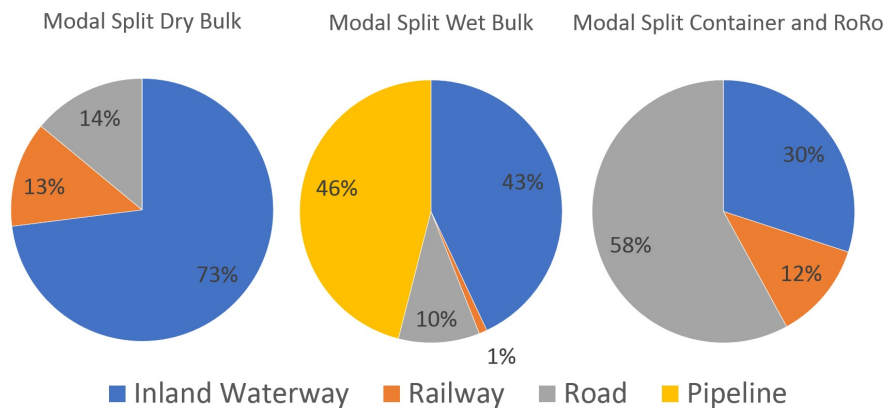


Figure 2-4: Model Split Hinterland Transport Port of Rotterdam (Van Doorn, 2018)

2-2-1 Ships

The design of a lock is mainly determined by the amount and type of ships passing the lock. A rough separation can be made between seagoing vessels and inland waterway vessels. Seagoing vessels are mostly large ships with a considerable depth. Most of their journey is on sea while they only have to pass a lock to enter specific ports, for this case they are designed to sail efficient on open seas while they are not specially designed for lock passages. Large variety in dimensions, the low manoeuvrability and the possible necessity for tugboats influences the dimensions of the lock. This has to be taken into account in the lock design. Seagoing vessels do not have a clear order of classification. However, more small seagoing ships are using inland waterways to reach inland destinations. For this cause PIANC introduced a new classification for River and Seagoing ships which can be seen in Table 2-8.

Table 2-8: River/ Sea going vessels (Rijkswaterstaat, 2017)

Class	Length (m)	Width (m)	Draught(m)	Air Draught (m)
R/S 1	90	13,0	3,5/4,5	7,0/9,1
R/S 2	135	16,0	3,5/4,5	9,1
R/S 3	135	22,8	4,5	9,1

Inland waterway vessels on the other hand are designed for good manoeuvrability and quick passage of locks to reduce time and thus costs. The ships are often equipped with bow thrusters and the crews are experienced with lock passages. Inland waterway vessels are designed in classes which are based on the different waterways with specific waterdepths, locks and bridges. In Europe the CEMT classes are used which can be seen in Table 2-9. A more specific overview of the CEMT classes used in the Netherlands is given in Appendix A. In Appendix A the AVV classes are introduced. Since the CEMT classes were introduced in 1954 some of the ship dimensions are outdated due to modifications on the ships.

Table 2-9: CEMT Classes (Rijkswaterstaat, 2017)

CEMT - Class	Width (m)	Length (m)	Draught (m)		Air Draught (m)	Tonnage (t)
			Loaded	Empty		
I	5,05	38,5	2,5	1,2	4,25	365
II	6,6	50-55	2,6	1,4	5,25	535-615
III	8,2	67-85	2,7	1,5	5,35	910-1250
IV	9,5	80-105	3,0	1,6	5,55	1370-2040
Va	11,4	110-135	3,5	1,8	6,40	2900-3735
VIa	17	135	4,0	2,0	8,75	6000

In addition to the commercial vessels, attention must be paid to recreational vessels. Recreational vessels are often small vulnerable vessels with inexperienced crews. Extra precautions must be taken when a combined passage of recreational and commercial vessels occurs. It should be noticed that the traffic intensity of recreational vessels is heavily depending on the season, local weather conditions and other local circumstances (Vrijburcht et al., 2000). The Economic Commission for Europe (ECE) divided the recreational vessels in different categories for the European inland waterways which can be seen in Table 2-10.

Table 2-10: ECE Recreational Vessel Classes (Rijkswaterstaat, 2017)

Type Vessel	Category	Length (m)	Width (m)	Draught (m)	Air Draught (m)
Open vessel	RA	5,5	2,0	0,50	2,00
Cabin Cruiser	RB	9,5	3,0	1,00	3,25
Motor Yacht	RC	15,0	4,0	1,50	4,00
Large Sailing Yacht	RD	15,0	4,0	2,10	30,00

2-2-2 Ship Mooring Loads

According to Vrijburcht et al. (2000) the loads on the mooring facilities are based on the class of ship which makes use of the lock. The force which the bollards in the chamber has to resist for inland waterway ships are categorized by the CEMT classes which can be seen in Table 2-11. For recreation vessels a maximum 40 kN is used while for sea going vessels the bollard has to withstand 1000 kN . Since it is uncertain how the ships will be moored due to the choice of the skipper it is assumed the whole force will be perpendicular to the chamber wall to be conservative.

Table 2-11: Design Bollard Loads CEMT Classes (Vrijburcht et al., 2000)

CEMT-Class	Bollard load (kN)
I	100
II	100
III	150
IV	150
Va, Vb	200
VI	300

The horizontal distance between the bollards also depends on the ships which pass the lock. Different from the load on the bollards the distance between the bollards is similar for all the inland waterway vessels, namely 15 m . For sea going vessels Vrijburcht et al. (2000) gives 25 m as an indication and for recreational vessels a distance of 5 to 6 m is given. In this thesis the conservative 5 m is used. The bollards for sea going vessels are placed on top of the quay at a distance of approximately 3 m perpendicular to the lock wall. The bollards for inland water way vessels and recreational vessels are constructed in the lock wall. The lock walls will be constructed to satisfy the assumption that the force of the bollards will be transferred over lock wall equally for the calculation of the overall stability of the lock walls.

The vertical distance between the bollards is $1,5\text{ m}$ for inland waterway vessels and $1,25\text{ m}$ for recreational vessels. The maximum distance between the lowest lock level and the bolder is $1,75\text{ m}$ for inland waterway vessels and $1,50\text{ m}$ for recreational vessels.

2-2-3 Future

The type and amount of shipping is changing in the recent years and it will continue to change in the upcoming years. Due to economic growth the total amount of cargo is increasing leading to more ships but also larger ships. Since larger ships are economically more feasible compared to smaller ships there is a shift to larger ships. This has an effect on the efficiency of the current locks (Quist and Verheij, 2010). The expansion of the inland shipping is limited by the present locks due to either maximum ship size or lock capacity. Also a shift in cargo type is occurring. The volume growth of dry bulk transported is decreasing due to the rise of green energy, the amount of coal transported is the mean driver for this decrease. On the other hand the amount of container transport is increasing rapidly, in 2017 a growth of $12,3\%$ was measured in the container transshipment in the Port of Rotterdam (Buijs, 2018), and for inland navigation in Europa a growth of $6,0\%$ was recorded (Centrale Commissie voor de Rijnvaart, 2018). Containers are a relatively light type of cargo which means that a larger amount of volume can be transported on a ship without reaching the maximum tonnage. This, however, means that the containers can be stacked higher which leads to a larger air draught. This is only possible if the bridges and locks on the route have sufficient head clearance. For future lock design a study has to be carried out on the modification of the future fleet to optimize inland waterway transport.

2-3 Availability and Related Costs

Navigation locks are often present in main transport routes. They are designed to serve the amount of ships which are present, or will be present in the future, on the channel or river. When the capacity of the lock is not sufficient to deal with the favourable amount of vessels an extra lock or a lock expansion will be needed. The reason for this is that when the capacity is not sufficient the waiting time will increase. Longer waiting times for the vessels will lead to longer total travel times. How longer the journey takes the higher the costs for the shipping company will be. This can have an influence on the whole economy in the area.

Another cause of waiting times can be downtime of the lock. This downtime can be caused by for instance, maintenance of the lock, a collision with a ship, malfunction of the mechanics or extreme water heights. All these reasons for downtime will have financial consequences. For this reason safety requirements are intensified and locks are designed in a way to limit downtime. This will often lead to higher investment costs which are compensated by the lowering of the downtime.

2-3-1 Climate Change

In the future climate change will lead to more extreme weather events, such as more periods with heavy rainfall, more extreme rainfall events but also longer periods of drought (Van den Hurk et al., 2014). These events have an impact on the rivers and canals. Extreme rainfall events can lead to flash floods while longer periods of drought or heavy rainfall will lead to lower and higher water levels, respectively. High water levels lead to less head clearance under bridges and lift gates. In extreme cases a waterway may be closed due to high water to avoid navigation in high flow velocities. Also locks may be closed if the water level is too high for operation. Low water levels can lead to closure of locks due to insufficient water depths. Another problem of low water concerning locks is the loss of water from the upstream area during a lock cycle. Multiple solutions for this problem are applicable like pumping water back upstream, Water Saving Basins, intermediate gates inside the chamber, twin synchronized lock or the use of a lock ladder (PIANC, 2009).

Low water levels not only lead to a potential closure of locks, the water depth in the canal or river should be sufficient for the normative ships to sail with a considerable load. In the Netherlands there is an agreed minimal water level for large not canalized rivers to ensure sufficient depth which is called "Overeengekomen Laagste Rivierstand" (OLR). Rijkswaterstaat strives to maintain a maximum undercut frequency of the OLR of 5% per year. When the water level of a river is lower than the OLR more than 5% per year, measures should be taken. An option to gain better control over the water level is by canalizing the river by installing locks and weirs to maintain a minimum water level. For canals, canalized rivers and the IJsselmeer the undercut frequency of the OLR has a maximum of 1% since it is less difficult to maintain the needed water level (Rense, 2008).

Chapter 3

Parametric Design

The use of computers within the design process has been increasing rapidly in the past years. A widely used method within the building sector is "Building Information Modelling" referred to as BIM. BIM aims to make design visible and improve the communication and information exchange between different stakeholders within a project (Azhar, 2011). Bew and Richards (2008) created a level system to order the complexity of BIM given in Figure 3-1. It started with level 0 where CAD was introduced, CAD stands for Computer-Aided Design. First CAD was known as Computer-Aided Drafting, it was used for the transfer from simple hand drafts to simple drafts made manually on the computer. This expanded to the level 1 BIM where 2D and 3D models could be made with certain computer programs. Level 2 of BIM is the current mostly used level and targets on the information which can be stored within the design to allow different stakeholders to read, understand and add information to and from the model with wide use of standards. In level 2 of BIM parametric design is introduced, instead of changing each component of a model manually every time some changes have to be made, parameters are introduced. The model will be designed such that by changing one of these parameters the whole model adapts to that change and new result will be calculated. This reduces the time which is needed to introduce changes in a design.

The next step in BIM is Level 3 which is an expansion of level 2. Eastman et al. (2011) stated that the level 3 BIM aims to add "intelligent" layers to the conventional CAD and parametric models. In this way a new dimension is added to the parametric design process. Not all the parameters will be exactly given as in a normal parametric design sheet which gives one answer for the given parameters. The user of the model can give ranges within the parameters or leave some parameters even open. The model now generates various solutions within the range of the entered parameters by the user. This gives the designer the opportunity to explore multiple possible designs in a limited amount of time (Chi et al., 2015). Due to intelligent software the model can be programmed to limit the total amount of designs by evaluating them during process. In this way the model only produces a couple of optimum designs.

Still a lot of accidents and failures occur in the construction sector. Most of these failures are caused by human mistakes in the design or construction phase (Linszen, 2018). Due to

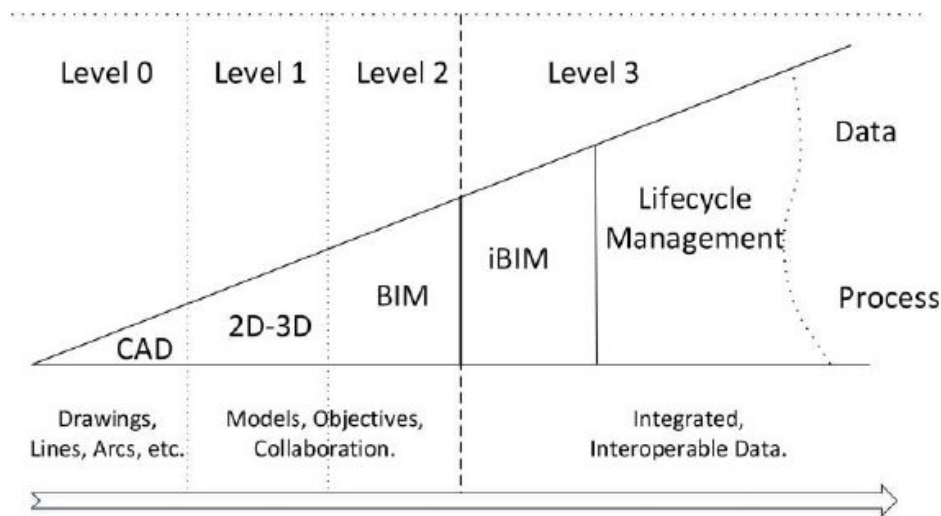


Figure 3-1: BIM Maturity Diagram (Bew and Richards, 2008)

the replication of multiple calculations and the transfer of data after each calculation in a manual procedure the chance of mistakes is high (Chi et al., 2015; Eadie et al., 2013; Sacks, 2004). With automated parametric design the amount of calculation errors can be reduced (van der Ploeg, 2018). When an error is found in the model, the code will be adjusted to make sure that this mistake will not happen again. In this way the models exclude more and more mistakes while a mistake in the manual procedure can be made over and over (Linszen, 2018). However, it will be important to make sure the parameters will be entered correctly into the model. This can be achieved by creating an user friendly user interface which an engineer can understand. During calculations interim results should be presented to verify the correctness of the final answer. It should be noted that a good model does not replace the engineer, the findings need to be checked on validity (Hendrikx, 2018).

3-1 Definition of 'Optimum Design'

It is hard to formulate an 'Optimum Design' valid for the lock building projects in the Netherlands, and nearly impossible for Europe and the whole world. This is because a design should be easy to construct, easy maintenance, no hindrance for navigation during construction, few of nuisance for the surrounding area, environmental friendly, few waiting time for ships and, of course, low in construction and maintenance costs. It is impossible to optimize all these requirements simultaneously, because the available budget will be limited and some of these criteria are conflicting. To make an optimum design it has to be clear which are the most important wishes from the client. However, the client often has wishes which are contradictory. To overcome this problem it is recommended to show multiple design alternatives each with their own benefits and specifics. In this way it is possible to show how different requirements of the clients relate to each other. This allows the client to make an informed decision for a preferred alternative.

3-1-1 Environmental Impact

To evaluate the environmental impact of an alternative the "Milieu Kosten Indicator" (MKI) value is used, which means the Environmental Cost Indicator. This value is developed by Rijkswaterstaat as part of the Dubocalc method. MKI values are addressed to each type of building material and defined by costs per volume or running meter. MKI values are calculated depending its impact on the environment based on the Life Cycle Analysis reported in ISO 14040 standard and Environmental Assessment Method Buildings and Construction (Rijkswaterstaat, 2018). This will be further discussed in Chapter 11.

3-2 Demand for a Design-Tool

As described in this section, parametric tools have many advantages. In the future the ultimate goal will be to have a whole lock complex in one design-tool which can calculate a complete lock for every situation. This, however, is not possible within a Master Thesis. And that, on the other hand, is not a problem because an advantage of parametric design-tools is that different tools can be easily combined. In this way it is possible to develop different design-tools separately which could be connected later, or to develop a specific design-tool which later can be expanded. The question rises which component of the lock should be considered first for a design-tool.

Slijk (2013) analyzed the designed cost for the new lock in Terneuzen. He looked at the total costs of each component of the lock as can be seen in Figure 3-2. It can be seen that a large part of the total costs are going to the lock head, the lock chamber and the mechanics, making these components attractive for further investigation. Apart from the total costs Slijk (2013) studied how these costs are divided into different phases of the life cycle as illustrated in Figure 3-3. The Design-tool can mainly generate savings in the part of Design Costs since the Design-tool will shorten the time needed for a proper preliminary design due to automation. The Design Costs mainly consists of man hours. Another optimization can be accomplished due to the possibility to examine more alternatives during preliminary design. This may lead to efficient designs which in the past were overlooked due to the early converging. These in the past overlooked efficient designs can lead to a reduction in Material Costs and Employment and Overhead Costs due to the application of different building methods and usage of materials. From the most costly components of Figure 3-2 the lock head and the lock chamber are high in Materials, Employment and Overhead Costs. Also the design costs of these components are relative and absolute highest which is highlighted in Figure 3-4.

Total Costs per Component

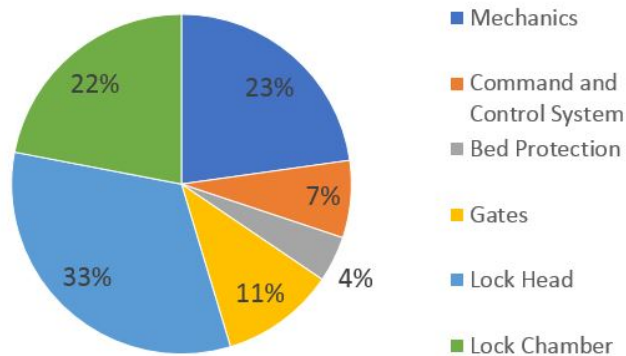


Figure 3-2: Total Cost Distribution Components of a Lock (Slijk, 2013)

Costs Distribution per Component

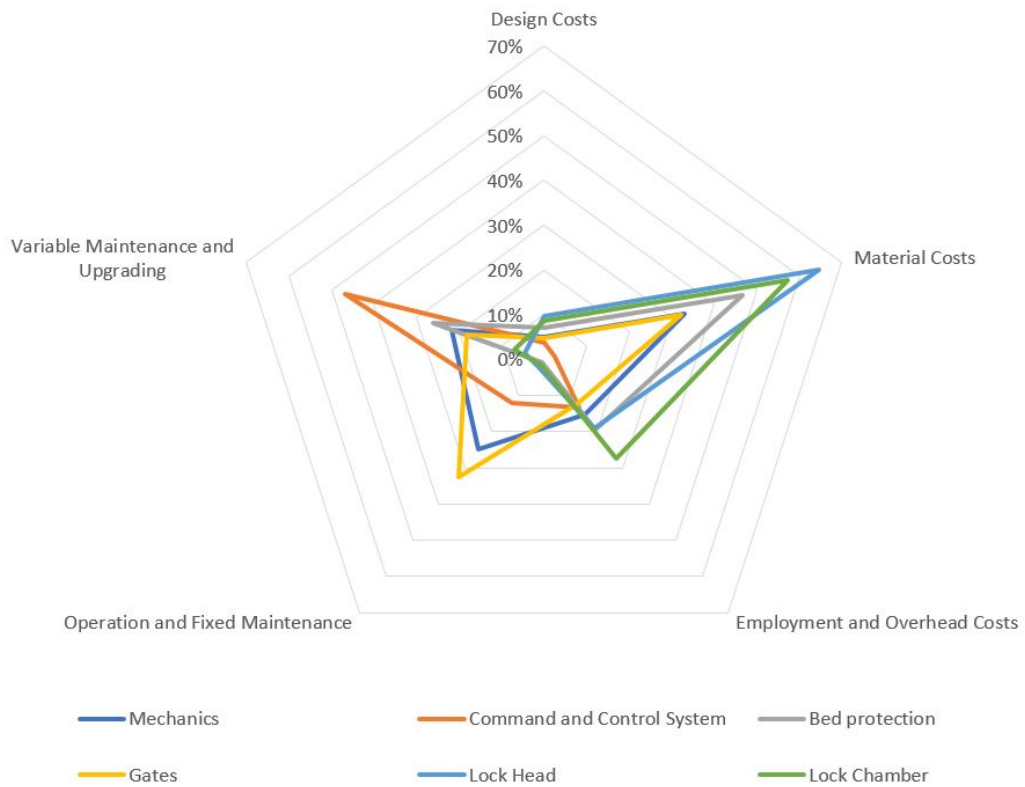


Figure 3-3: Cost Distribution Per Lock Component (Slijk, 2013)

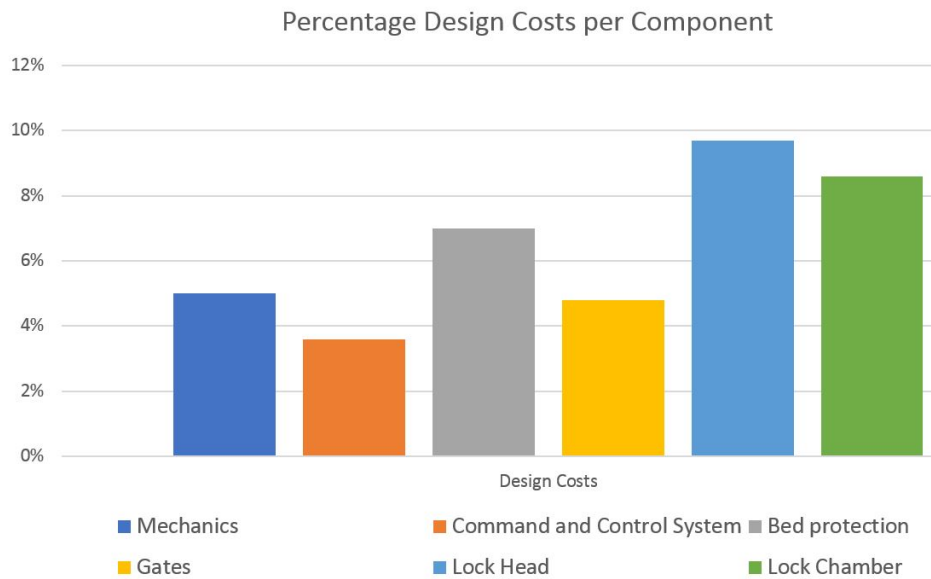


Figure 3-4: Percentage Design Cost per Component (Slijk, 2013)

3-3 Possibilities for Automation

In the previous Section 3-2 the possibilities for cost savings with a Design-tool for different components are discussed and will be evaluated in this section.

3-3-1 Mechanics

23% of the total costs of the lock are being spend on the mechanics which includes for example the movement equipment for the gates and all the electronics. This makes it interesting to check for possibilities for a design-tool. The first motivation would be the reduction in Design Cost, this however, is not effective since the Design Costs of the mechanics are only 5% of the total cost for mechanics. Like the Design Costs, the Material Costs and the Employment and Overhead Costs, of respectively 33,1% and 15,3%, are relatively low compared to the other components. Furthermore, the design of the mechanics heavily depends on the designs of the other components. For example, the costly movement equipment depends strongly on the type of gate which will be used and the filling and emptying systems depends on the type of gate, lock head and lock chamber. These reasons conclude that it is hard and inefficient to start creating the design-tool with the mechanics. When other components are already in the design-tool it will be interesting to add the mechanics.

3-3-2 Lock Head

With 33% of the the total costs, the lock head is the most expensive part of the lock. This makes the lock head an interesting component to look at. The Design Costs, which are interesting when looking at the design-tool, are with 9,7% relatively highest compared to the other components. Also the material costs are the highest with 64,5%. The Employment

and Overhead Costs are also high with 19,1%. However, the type of lock head used strongly depends on the type of gate which will be used in the design. This means that the design of the gates and the lock heads should be combined to be useful for a design-tool. This would be a good opportunity for a design-tool but since the design of both the gates as well as the lock head are complicated, the scope is considered not to be suitable for a single master thesis to create such a complicated design-tool.

3-3-3 Lock Chamber

The lock chamber scores the highest in the field of Employment and Overhead Costs with 25,7% which makes it interesting when looking at multiple alternatives. With a design-tool it will be possible to evaluate more alternatives which could lead to a different perspective on the design resulting in lower Employment and Overhead Costs. Also the Material Cost are high with 57,3%, these also depend on the alternative which is chosen and will be partly related to the Employment and Overhead Costs. The design costs are covering 8,6% of the total which is a little less compared to the Lock head but still a considerable amount. This can be lowered with the use of a design-tool. The lock chamber is hardly depending on the design of other components which makes it a solid base for the Design-tool of a lock. Also lock experts from Royal HaskoningDHV concluded that a lot of under substantiated decisions are made in the preliminary design of the lock chamber due to the lack of time and that the knowledge is only based on experience and thus extrapolating previous designs as stated in Section 3-2.

On the bases of the above mentioned arguments it is concluded that the design of the lock chamber will be the subject of this Design-tool.

3-4 Costs within the Design-tool

In order to make an useful preliminary design a cost estimation should be included. Most building projects have a certain budget to hold on to when fulfilling the design requirements. Which alternative will be chosen is thus often depending on the total costs. The costs in the Design-tool will be based on the unit prices used by Royal HaskoningDHV and will be evaluated by experts in the field of locks. To deal with price fluctuations and preferences specified by particular contractors the unit prices will be manually adaptable for the user. How the prices are chosen and will change over time will not be part of this master thesis.

Chapter 4

Problem Definition and Research Objectives

4-1 Problem Definition

In the twentieth century a lot of navigation locks were built in the Netherlands. In the period between 1930 and 1940 there was a peak caused by, among other things, the construction of the Twenthekanaal. Likewise a peak occurred between 1950 and 1980 caused by the establishment of the Delta Works which led to the closure of many estuaries. With most of the navigation locks having a lifespan of 80 to 100 years a peak of needed renovation and replacement will occur in the coming years. Rijkswaterstaat, the administrator of most of the locks in the Netherlands has made an inventory of all the locks to evaluate their condition in the project called "Risico Inventarisatie Natte Kunstwerken" (RINK). The follow up of this project was "Vervangingsopgave Natte Kunstwerken" (VONK) this included which locks should be renovated or replaced in the upcoming years (Ministerie van Infrastructuur en Milieu and Ministerie van Economische Zaken Landbouw en Innovatie, 2012). An overview of the number of structures which need replacement is given in Figure 4-1.

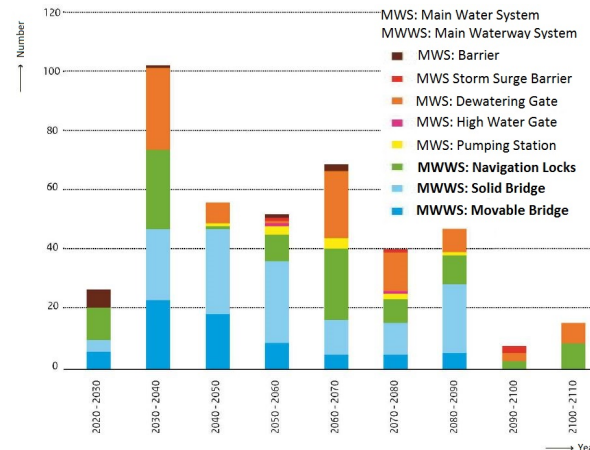


Figure 4-1: Hydraulic Structures which need replacement in the coming years (Ministerie van Infrastructuur en Milieu and Ministerie van Economische Zaken Landbouw en Innovatie, 2012)

Rijkswaterstaat does not replace or renovate the locks themselves. When something has to be done they put it out to tender. Companies now can make an offer to design and/or construct the renovation or replacement of the lock. When the companies make a proposal in the tendering phase, a preliminary design has to be made. The preliminary design has to meet the Design Criteria required by the client. When the companies submit their proposals the client chooses the best design according to a number of criteria which have been discussed in section 3-1.

This tendering phase is time and resource consuming for the companies (van der Ploeg, 2018), they have to make multiple preliminary designs for the multiple alternatives of locks which are available. All these alternatives consist of different components all with multiple types of material and construction methods available. These different parameters lead to a large amount of options which add up in an exponential way with each parameter. To limit the amount of time and resources needed, nowadays, decisions are made in the early phase to exclude multiple combinations and alternatives of different parts of the lock. This leads to early converging in the design process. An expert chooses a few alternatives which will be worked out into a preliminary design based on his experience in the field of lock design. Due to this early excluding of options and the few alternatives which are reviewed due to the lack of time, out of the box thinking is limited and various new innovative solutions will go unexplored. Another problem is that decisions are made based on experience of prior projects, this is partly a good thing because one can learn from previous experiences but this could turn into a problem when solutions found in previous projects are used again in new projects and are extrapolated to the new situation. The lack of numerical substantiation of the solution could lead to wrong estimation of cost and work to be done and increase the risk of potential failure of the design (Riedijk et al., 2018).

4-2 Research Objective

The objective of the research is to optimize preliminary lock design in the tendering phase by creating a design-tool which automatically generates all plausible alternatives based on

the Design Criteria and boundary conditions of the client. This will lead to the opportunity that multiple alternatives can be presented to the client, all with their own advantages and disadvantages. By doing so the expert of the company can discuss the possibilities with the client. Due to relatively fast calculations of preliminary designs for multiple cases this design-tool can discover plausible optima and tipping points in the design of locks which will be studied in a parametric study. The focus of this research will be on the chamber of the lock due to the variety of options, the impact on the construction method and the time needed for the calculations. This choice for the lock chamber is explained in Section 3-2 and Section 3-3.

4-2-1 Research Questions

- Is it possible to create automated design-tool for the preliminary design of a lock chamber?
 - Which variables should be taken into account in the design-tool?
 - Which parameters should be variable and why?
 - Which alternative is most optimum for specific lock lengths, widths, water heads and water depths.
- What is the influence of the building materials used for the lock chamber on the alternative selection?
 - What is the impact of the materials on the environment and climate change?
 - Does a CO₂ tax impact the differences in total costs and to which extend?
 - How does the unit prices of the material influence the total costs?

4-3 Research Method

To reach the final goal as described in section 4-2 the method will be followed as described below. An overview of the Master Thesis can be seen in Figure 4-2.

Phase I

A lock consists of several components, literature research is needed on all the different components of a lock to get a better understanding of the whole system. For these components multiple options are available, all being suitable in different circumstances. Some components relate to each other while others can vary independently.

Phase II

Standardize all the input data to be usable in the design-tool. First develop the design-tool for the sheet pile wall lock chamber for different input parameters with the knowledge gained in the previous phase and input from experts in the field of lock design. After completing this first part of the design-tool it should be tested and validated on different cases to eliminate errors before further expansion of the design-tool.

Phase III

Expansion of the design-tool by adding different types of lock chambers. All using the same input parameters to achieve parallel calculation.

Phase IV

Testing and validation of the design tool on different cases. Check whether the design tool gives comparable options as was decided for in the actual case. Check what the differences are between the design tool and the actual case and explain these differences.

Phase V

Reviewing the design tool. Searching for optima in components of the lock chamber. Where can the process or design be improved? This will be done with the use of a parametric study of the lock chamber. After evaluation of the design-tool improvements can be implemented to make the design-tool more efficient.

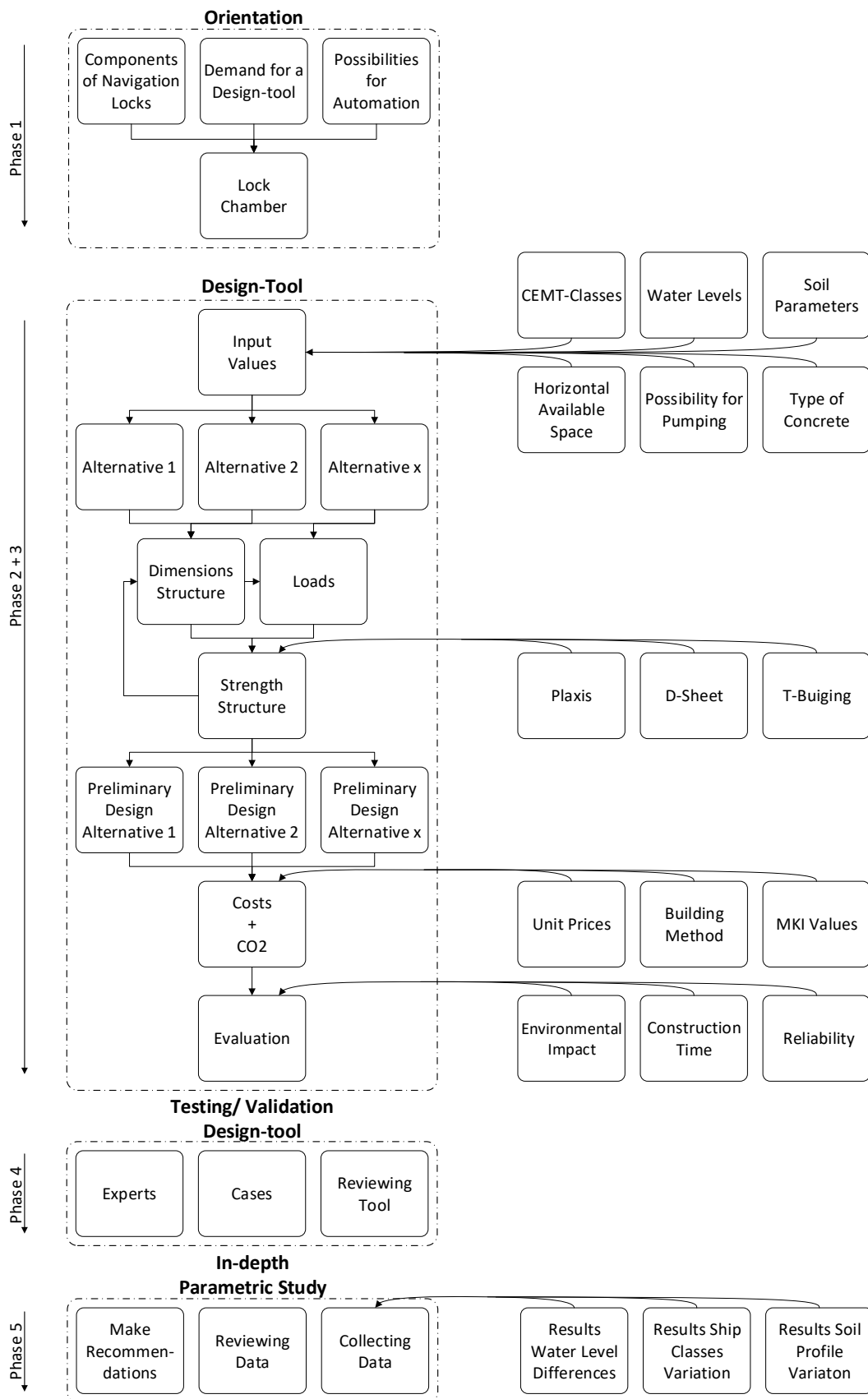


Figure 4-2: Framework Master Thesis

Chapter 5

Model Description

In this chapter the structure of the design tool will be explained. What is calculated, where can it be found in the model and which calculations and inputs relate to each other. The model will be used for the calculation of multiple alternatives for lock chambers (Figure 5-1). As output the model produces the final dimensions, costs and environmental impact of the different alternatives. The alternatives which are checked are a sheet piles wall structure, a diaphragm wall structure and a U-basin, constructed with two different types of construction pits: an open excavation and a construction pit with temporary sheet pile walls. Also Combi-Walls are checked. This will be done within the 'sheet pile wall structure' since the same method applies. First the overview of the model is given. After that the more specific parts of the model are explained. For an even more detailed view the model script itself should be inspected (Appendix F), also more detailed information can be found in Chapter 6 up to Chapter 11.

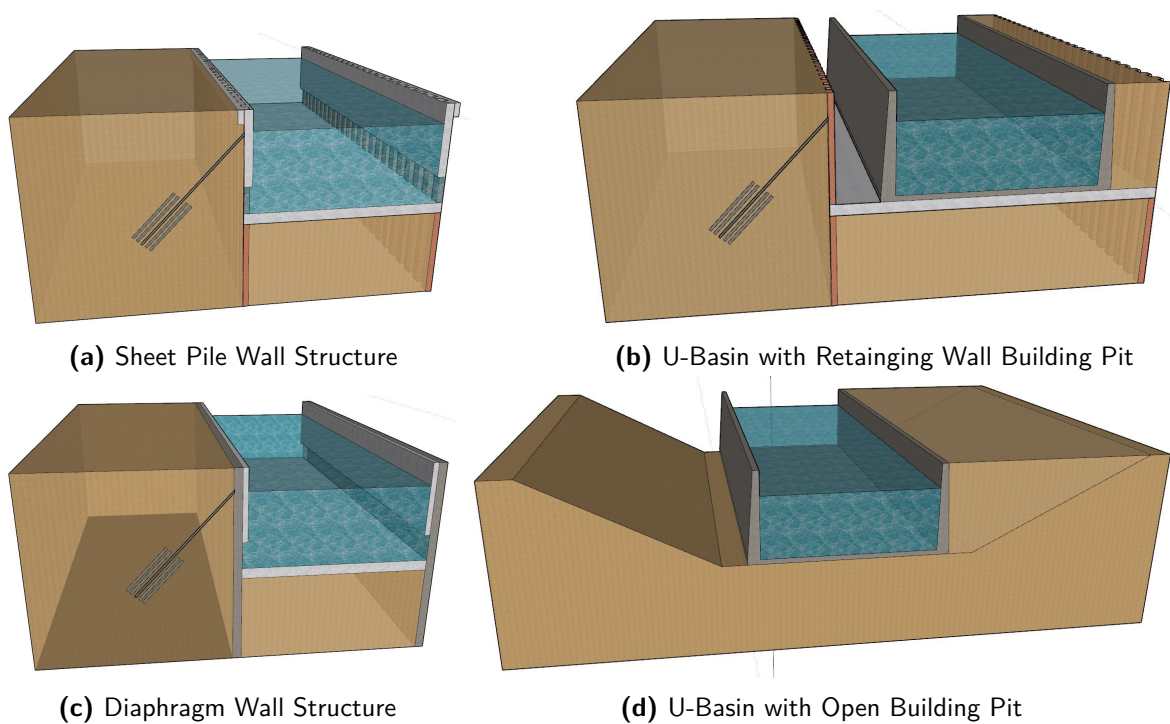


Figure 5-1: Lock Chamber Alternatives

5-1 Model Overview

An overview of the model is given in Figure 5-2. First the User enters a easy accessible Excel file. Here the User Input can be entered and a dictionary with Default Data is present. This Excel file is read by the Python script. Depending on the User Input and the selected Default Data the General Dimensions of the lock chamber are calculated. These dimensions will be used to calculate some general equations, the dimensions of the U-basin structure and the input of the D-Sheet calculation.

The Python script automatically starts the D-Sheet calculation and reads the results. These among the other results for the retaining wall structures and the U-Basin structures are stored in a new Excel file to make them easy accessible.

This Excel file will be used as input for the Python script which calculates the costs, environmental impact and CO₂ tax for the different alternatives. These values will be summarized in a new Excel file.

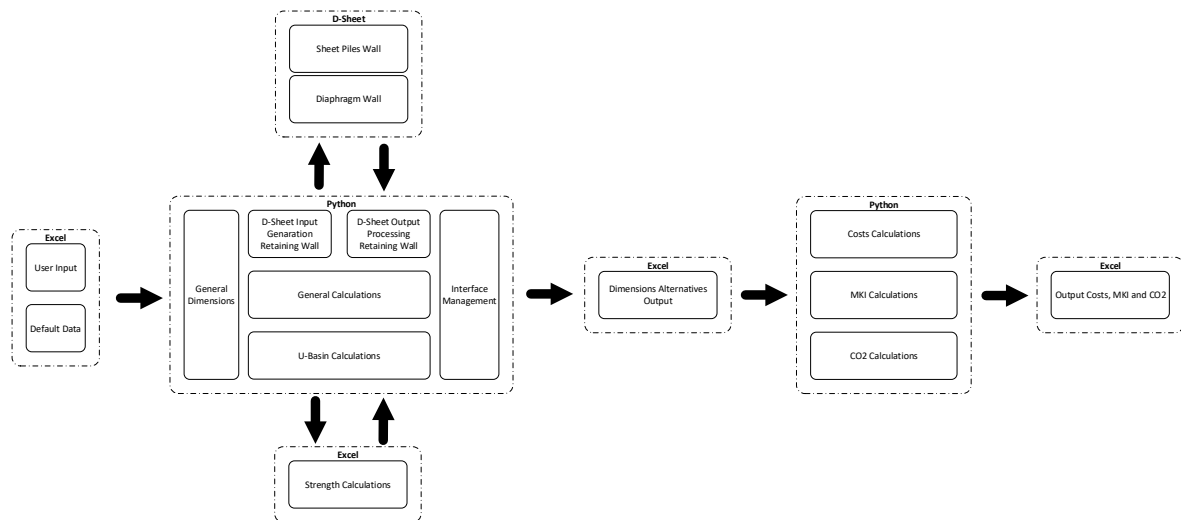


Figure 5-2: Model Overview Design Tool

5-2 Model Analysis

5-2-1 User Input

In the Excel the user of the design tool is asked to specify the values which are unique for the problem. The most important values are the water levels, locking levels, the type of ship using the lock and the soil profile. Some other values are set as a default but can be changed for user preferences. After a sensitivity study these defaults are chosen. To ensure full parametric design the model can be run through multiple times using all the possible combinations.

5-2-2 Default Data

For the different calculations multiple default values are used. All these values are stored together so they can be easily looked into to check whether the data which is used satisfies the users preferences. Also assumptions which are made can be seen in this data. When an other assumption is preferred this can be easily adapted.

5-2-3 General Dimensions

In this section the general dimensions of the lock chamber are calculated. The values which are calculated are the inner dimensions of the lock. These are the dimensions to fulfill navigational task of the lock. The dimensions can also be set manually by the user. In that case, this step will be skipped.

The calculation of the General Dimensions are given in Figure 5-3. A distinction is made between the vertical dimensions and the horizontal dimensions. The horizontal dimensions are mainly based on the ships which make use of the lock. How this is done is described in

Chapter 6. The vertical dimensions are in addition to the ships mainly depending on the water levels and the locking levels.

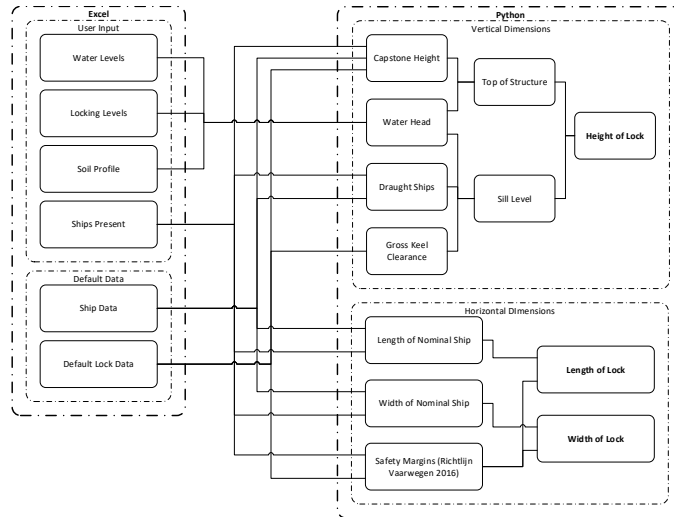


Figure 5-3: Model Analysis General Dimensions

5-2-4 General Calculations

Some equations are needed for both the calculation of the U-Basin structures as well as the retaining wall structures. Also some calculations for the retaining wall structures are described here since they are not present in the D-Sheet calculation.

Tension Piles

The tensions piles are calculated according to the CUR 2001-4 guideline. The work flow of this calculation is given in Figure 5-4. First the Net Vertical Upward Force is calculated to check whether tension piles are needed in the first place. When the tension piles are needed to prevent up drift the soil profile is analyzed to check the capacity of certain piles in a specific alignment. This is an iterative process to come up with the optimal pile alignment.

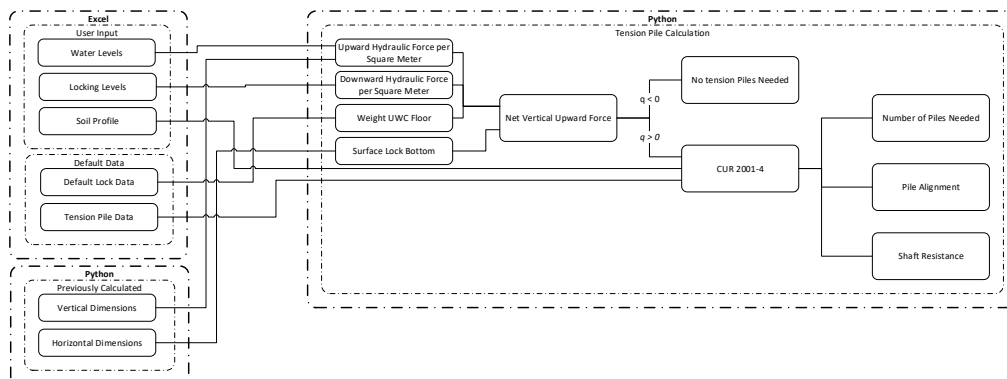


Figure 5-4: Model Analysis Tension Piles

Under Water Concrete Floor

For the retaining wall structures and the temporary sheet piles building pit an under water concrete floor (UWC) will be casted. The overview of the UWC floor model is given in Figure 5-5. The UWC floor is calculated conform the CUR 77 guideline within a VBA sheet developed by RHDHV. The strength needed for the floor mostly depends on the net vertical force and the width of the floor. The normal force in the UWC floor is originating from the ground and hydraulic pressures acting on the retaining wall. A higher normal force leads to more moment resistance. This normal force is calculated in the D-Sheet calculation which will be explained in Section 5-2-5. The strength of the UWC floor is highly influenced by the pile alignment and the shaft resistance of the piles. These are already been calculated in the Tension Pile Part. The casting method of the UWC floor changes the tolerance of the floor and can be changed by the user.

The CUR 77 VBA sheet checks the UWC floor multiple strength criteria. The first check is the moment distribution for the longitudinal direction. The second check is the moment distribution in the width direction of the floor. This check is preformed in a uncracked state, a cracked state with a membrane function and a cracked state without a membrane function. One of these three checks should suffice. To ensure sufficient shear force resistance the UWC floor should satisfy that the shear force resistance at the down side of the UWC floor is smaller than the occurring shear force. When this is not the case, three more specific checks should satisfy, check C2.a, C2.b and C2.c from CUR 77 (Jonker and Hagenaaars, 2015). The design tool takes not into account the connection with the tension piles and the retaining walls.

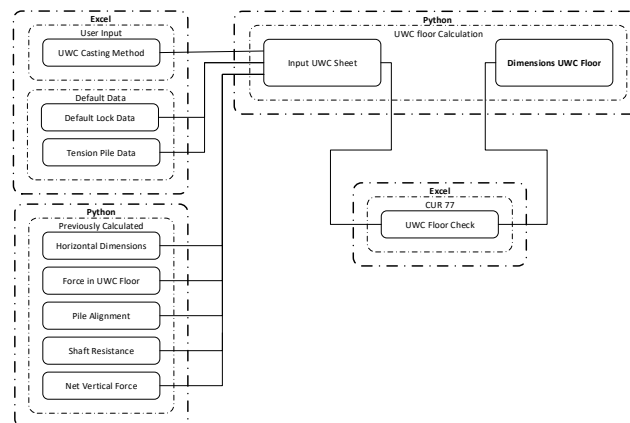


Figure 5-5: Model Analysis Under Water Concrete Floor

Capping Beam

In Figure 5-6 the calculation of the capping beam is displayed. The vertical dimensions of the capping beam depend on the properties of the retaining wall, the locking levels and the capstone height. The horizontal dimensions mainly depend on the thickness of the diaphragm wall or the height of the sheet piles and the required strength of the structure. The required strength depends on the loads which need to be transferred by the structure. The strength of the structure is calculated by the 'T-Normaalkracht en Buiging EC2' VBA sheet developed

within RHDHV. Checks will be performed on moment and shear force resistance. This will lead to the final dimensions of the capping beam including the needed reinforcement.

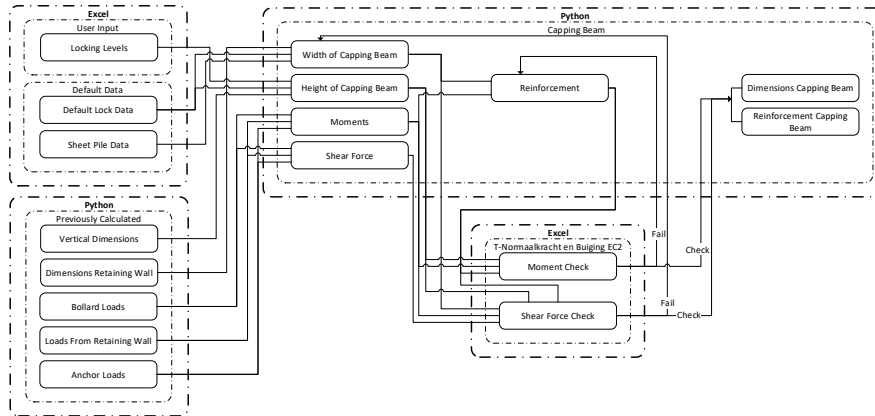


Figure 5-6: Model Analysis Capping Beam

Building Pit

For the construction of the U-Basin a building pit is needed. As can be seen in Figure 5-7 an open building pit and a building pit with temporary retaining walls is calculated. The excavations depend on the soil types en the dimensions of the building pit. The excavation calculation of the retaining wall pit will also be used to calculate the excavation of the retaining wall chambers. For the temporary retaining wall pit the UWC floor can be calculated as explained before and the temporary sheet pile wall will be calculated as described in Section 5-2-5.

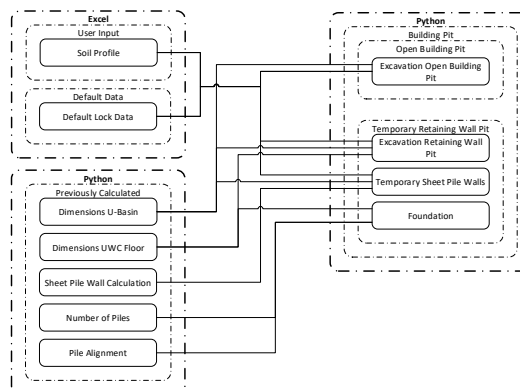


Figure 5-7: Model Analysis Building Pit

5-2-5 Retaining Wall

As can be seen in Figure 5-2 the calculation of the retaining walls starts in Python by creating an input for D-Sheet. D-Sheet checks the structure and sends the output back to Python.

D-Sheet Input

The goal of this step is to translate the user input, default data and the previous calculations into a readable file for D-Sheet. This progress is shown in Figure 5-8. First general input is translated to D-sheet. This is the part that creates the main structure of the retaining wall into D-Sheet. Secondly the bollard loads are calculated in Python and depend on the type of ships and the default data of load transfer through the capping beam. The surcharge load is an input value with a default which is standard for locks. The soil and hydraulic load are not an input for D-Sheet since they are automatically calculated from the general input.

D-Sheet has a different algorithm for both sheet pile and diaphragm wall calculations. The needed algorithm will be defined in the input file. For the calculation with sheet piles a dictionary is present within Python containing the profile specifications. For diaphragm walls first the strength and other properties are calculated in the Python script. These values will serve as input for D-Sheet.

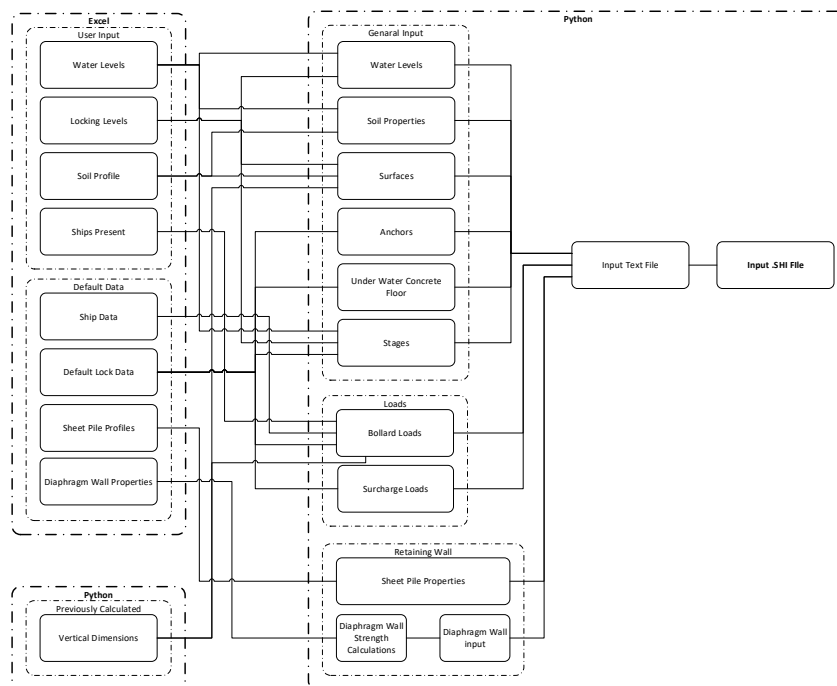


Figure 5-8: Model Analysis D-Sheet Input File Creation

D-Sheet Calculation

A Python script automatically sends all the input files to D-Sheet and creates a batch calculation for all previously created input files. In Figure 5-9 can be seen what happens within D-Sheet. It generally uses all the values from the input files to create an Eurocode 7 and CUR 166 calculation for all the defined stages. For this design tool the most important calculations are the stability check, the strength check and the occurring deformations. D-Sheet creates a .SHD file as output.

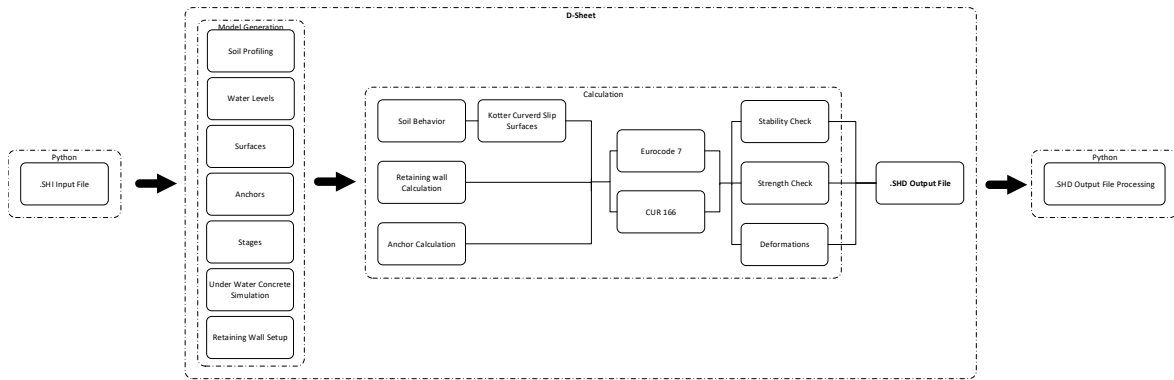


Figure 5-9: Model Analysis D-Sheet Calculation

D-Sheet Output

The D-Sheet .SHD output file will be converted to a .txt file by Python (Figure 5-10). A Python script is created to read this output file. First will be checked whether the retaining wall is stable. When the retaining wall is unstable the examination stops and the next file will be checked. When the structure is stable the moment distribution will be checked with the moment capacity of the sheet pile, combi or diaphragm wall. When the structure is of sufficient strength the deformations and shear forces are analyzed. From the shear force distribution the force in the UWC floor is established.

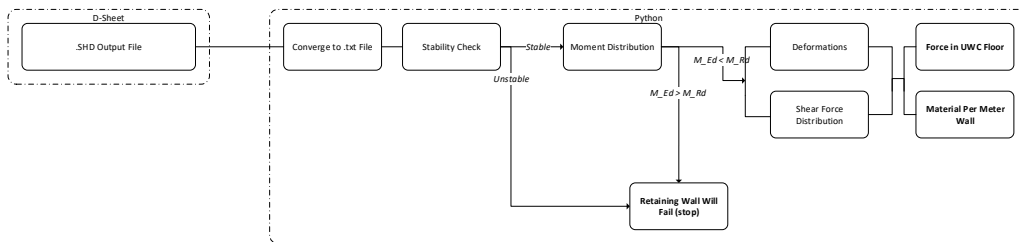


Figure 5-10: Model Analysis D-Sheet Output

Retaining Wall Output

With the use of interface management all the previously explained sub calculations and the default data will be stored in a data frame, see Figure 5-11. This will result in a data frame with multiple options for both diaphragm wall chamber structures as sheet piles wall chamber structures.

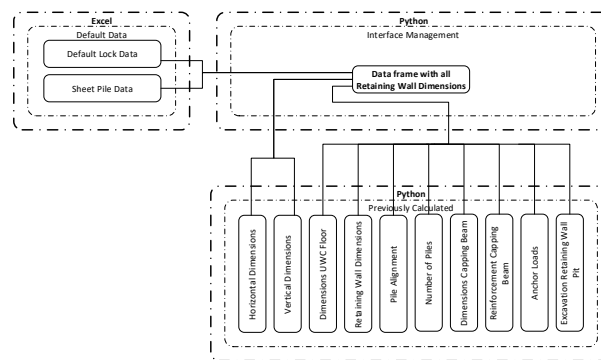


Figure 5-11: Model Analysis Retaining Wall Output

5-2-6 U-Basin

For the calculation of the U-Basin almost the same input is needed as for retaining wall structures. Instead of using D-Sheet the U-Basin is mostly calculated within Python. In Figure 5-12 an overview of the model is given. The soil and hydraulic loads are calculated in Python, with the bollard and surcharge loads this lead to a moment and force distribution which is also calculated in Python. This is done for different stages in its lifetime. Start dimensions and reinforcement values are, together with the moments and forces, exported to the 'T-Normaalkracht en Buiging EC2' VBA sheet. This sheet automatically calculate iterative the final dimensions of the walls and floor. This is done with checks on moment and shear force resistance.

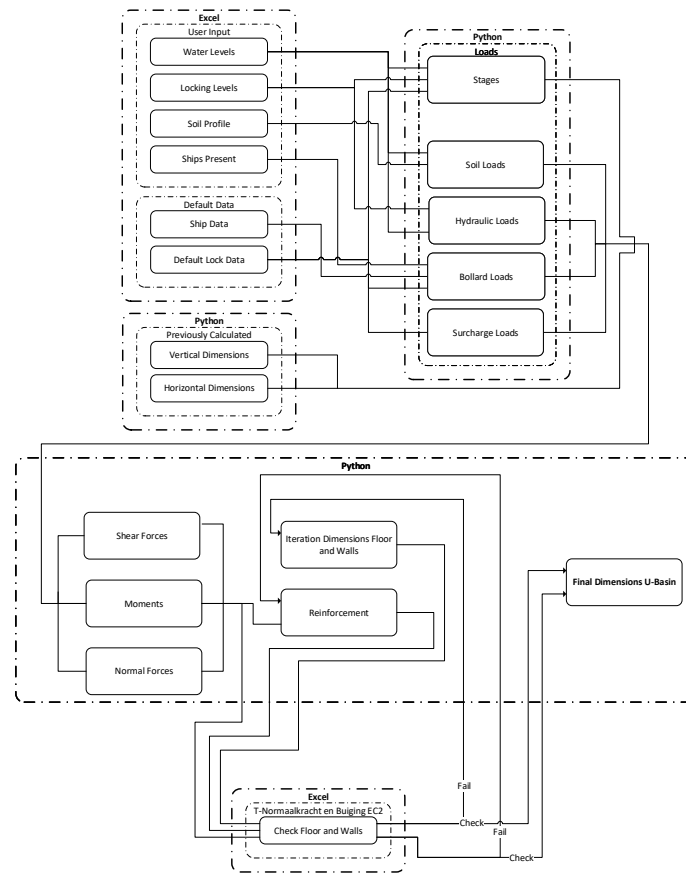


Figure 5-12: Model Analysis U-Basin Calculation

Interface management takes place and the dimensions of the U-Basin are combined with the general dimensions, the building pit specifics and the pile information. A data frame is created with all this information as can be seen in Figure 5-13.

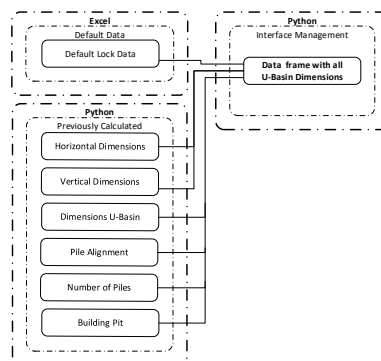


Figure 5-13: Model Analysis U-Basin Output

5-2-7 Output

All the data created before is extracted from the previous calculations. All the data which will be needed for the last part of the model is exported to an Excel file as can be seen in Figure 5-14. This in to easily use the dimensions of the calculated structures for further research and next calculation steps.

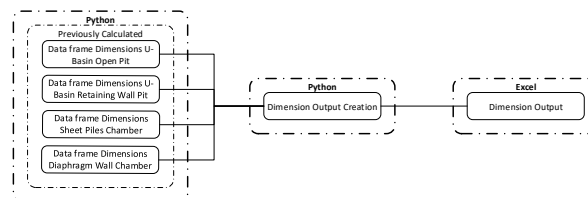


Figure 5-14: Model Analysis Dimensions Output

5-2-8 Costs, MKI-Values and CO2

The dimensions of all the alternatives are extracted from the Excel file. In these data all the quantities of materials used in this alternatives are present. With the default data the MKI and CO2 values can be calculated for each structure. With the user defined unit prices and CO2 taxes the final prices of the structures are calculated. This can be seen in Figure 5-15. All this final data is exported to an Excel file which can be easily read by the user.

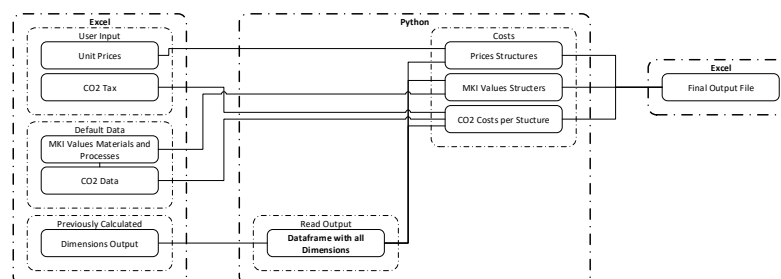


Figure 5-15: Model Analysis Costs Calculation

5-3 Calculation Methods Model

To ensure that all the possible outcomes of the model will be analyzed, the model should strive to be fully parametric. However, to limit calculation time and to filter unrealistic outcomes not all variables will be parametric. Optimization and default calculations will take place within the model. In this section will be explained which part of the model is parametric, where default values and calculations are used and where optimization takes place. The correctness of these methods will be evaluated in a sensitivity study in Chapter 14. Here will be concluded which variables should be parametric and which will contribute to a better results when being parametric.

The type of the structure is calculated parametric, this means that for every calculation a U-basin structure, a sheet pile structure and a diaphragm wall structure will be elaborated. Each part of the structure can be calculated in a parametric way (⌘), by optimization (↻) or with one default calculation (⇒). In Table 5-2, Table 5-3 and Table 5-1 is given with what type of method each part of the structure is calculated. Due to the parametric calculations the number of alternatives increase. The model will calculate all the produced alternatives with output dimensions as result. These dimensions can be used for environmental impact and cost calculations as described in Section 5-2.

Table 5-1: Method of Calculation U-Basin

	⌘	↻	⇒
U-basin	X	X	X
Building Pit	X	X	
Open Pit			X
Retaining Wall Pit	X	X	X
Walls	X		X
Length	X		
Profile	X		
Material			X
Anchor			X
UWC Floor		X	X
Material			X
Thickness		X	
Method			X
Walls		X	X
Thickness		X	
Material			X
Reinforcement		X	
Floor		X	X
Thickness		X	
Material			X
Reinforcement		X	
Tension Piles		X	X
Length			X
Width			X
Type			X
Alignment		X	
Loads			X

Table 5-2: Method of Calculation Sheet Pile Wall Structure







			
Sheet Pile Wall Structure	X	X	X
Walls	X		X
Length	X		
Profile	X		
Material			X
Anchor			X
UWC Floor		X	X
Material			X
Thickness		X	
Method			X
Capping Beam		X	X
Width		X	
Height			X
Reinforcement		X	
Material			X
Tension Piles		X	X
Length			X
Width			X
Type			X
Alignment		X	
Loads			X

Table 5-3: Method of Calculation Diaphragm Wall Structure

			
Diaphragm Wall Structure	X	X	X
Walls	X		X
Length	X		
Thickness	X		
Reinforcement			X
Material			X
Anchor			X
UWC Floor		X	X
Material			X
Thickness		X	
Method			X
Capping Beam		X	X
Width		X	
Height			X
Reinforcement		X	
Material			X
Tension Piles		X	X
Length			X
Width			X
Type			X
Alignment		X	
Loads			X

Chapter 6

Dimensions of the Lock

In this chapter the outer dimensions of the lock will be explained and how they are used within the model.

6-1 Horizontal Dimensions

The horizontal dimensions of the lock chamber are governed by the type and amount of ships which will pass the lock. Most locks in the Netherlands are designed for the dimensions of the normative ship which is present on the waterway. When the amount of traffic is extremely high or in other special cases a locking study is performed to calculate the optimal horizontal dimensions to handle more ships. In this thesis the lock will be designed according to the normative ship of the lock, it will be however possible to manually adapt the input dimensions of the lock within the model.

For maritime locks the length of the lock will be 1,15 times the length of normative vessel and the width will be 1,10 to 1,15 times the width of the normative vessel (Vrijburcht et al., 2000).

For inland navigation locks minimum dimensions are given per CEMT class as can be seen in Table 6-1. For the effective chamber length multiple values are given for CEMT III, IV and Va since extended ships are present in these classes. In the future more and more ships of these classes will be extended, for this reason the maximum values of these classes will be used as standard (Rijkswaterstaat, 2017).

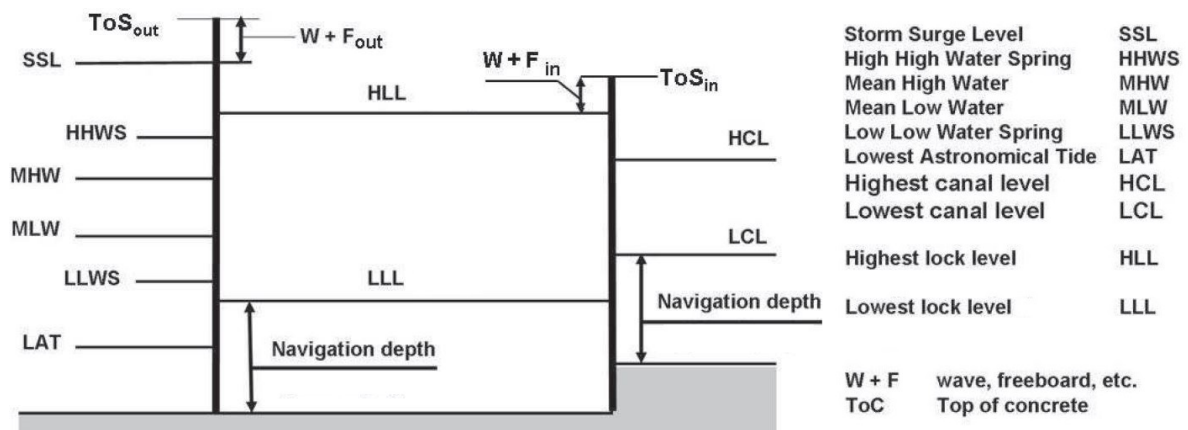
Table 6-1: Minimum Dimensions Inland Navigation Locks (Rijkswaterstaat, 2017)

Class	Effective Length (m)	Chamber Effective Width (m)	Chamber Sill Level (m)
I	43	6,0	2,8 - 3,1
II	60	7,5	3,1 - 3,2
III	80 - 95	9,0	3,1 - 3,3
IV	95 - 115	10,5	3,5 - 3,7
Va	125 - 150	12,5	4,2
Vb	210	12,5	4,7
VIa	160	23,8	5,0
VIb	215	23,8	5,0

6-2 Vertical Dimensions

6-2-1 Water Levels

The main objective of a lock is to transfer ships from one waterway with a certain water level to an other waterway with a different water level. When looking at inland waterway locks there is an upstream and a down stream waterway. At each waterway there is a Highest Canal Level (HCL) and a Lowest Canal Level (LCL). For maritime locks at inland side there is a HCL and a LCL, at the sea side the maximum water level is the Storm Surge Level (SSL) and the minimum water level is the Lowest Astronomical Tide (LAT). Under normal circumstances the water level will change constantly due to the tide. An overview of the water levels of a coastal lock is given in Figure 6-1.

**Figure 6-1:** Water Levels Coastal Lock (Molenaar, 2011)

Lowest Lock Level and Highest Lock Level

Due to extreme water levels like the LAT, SSL, HCL and LCL the range of the lock can be large. To minimize cost a Highest Lock Level (HLL) and a Lowest Lock Level (LLL) are chosen. The HLL is a value to which maximum water level the lock needs to be operational,

the value will be chosen to ensure little downtime for the ships while limiting the costs of the structure. The LLL is the lowest water level for which the lock should be operational and is chosen in the same way as the HLL.

Design Water Level Lock Chamber

For lock chambers the HLL and the LLL are governing since the lock heads are designed to withstand the extreme water levels. The only exception is when the lock is part of a primary flood defence, to ensure flood safety it could be decided to construct the chamber at the same level as the upper lock head. In that case, the lower lock head should be as high as the upper lock head to ensure flood safety. The other option is that extra measures are taken at the upper head to deal with flood safety. This can be done building the upper head for extreme water levels and ensure water safety by installing two lock gates. This is assumed for the Design-tool.

Maintenance Level

A choice has to be made whether the lock chamber should be completely dry during maintenance or if a lower water level is sufficient. This will be an input value for the user of the tool. When the chamber does not need to be completely dry, the maintenance level will be one meter below the Lowest Lock Level to make sure the lock wall and cover are good accessible.

6-2-2 Capstone Height

The capstone height is the height of the chamber wall above the HLL. The chamber wall has to retain the highest lock level and serves as guidance and mooring facilities for the ships. For guidance a distinction is made between visual guidance and physical guidance. Visual guidance helps the skipper to orientate in the lock for good manoeuvring, to ensure this the wall should be higher than the bow of the ship. This however is most of the time economically not feasible. Physical guidance is achieved by sliding and grating against the chamber wall. It is important that the ship as well as the chamber wall do not sustain damage. To avoid damage it is important that the chamber wall is a smooth surface. To reduce damage from the bow of the ship the wall should be as high as possible.

For inland water ways a capstone height of 1,5 m is common for ships up to CEMT Class IV. For class V and higher 2,5 m is used since often higher push vessels are present. Sea going ships often have a high free board. For this reason it is economically not possible to construct high walls. To protect the wall a good fender structure should be in place. In practise the minimum height of the chamber wall for maritime locks should have sufficient height to retain the water levels (Vrijburcht et al., 2000).

6-2-3 Gross Keel Clearance

The level of the sill is determined by the water levels and the vessels to pass. For the water level the lowest lock level is governing. For the vessels to pass, the gross keel clearance is

the important factor. This is the distance between lowest lock level minus the draught of the vessel and the sill (Figure 6-2).

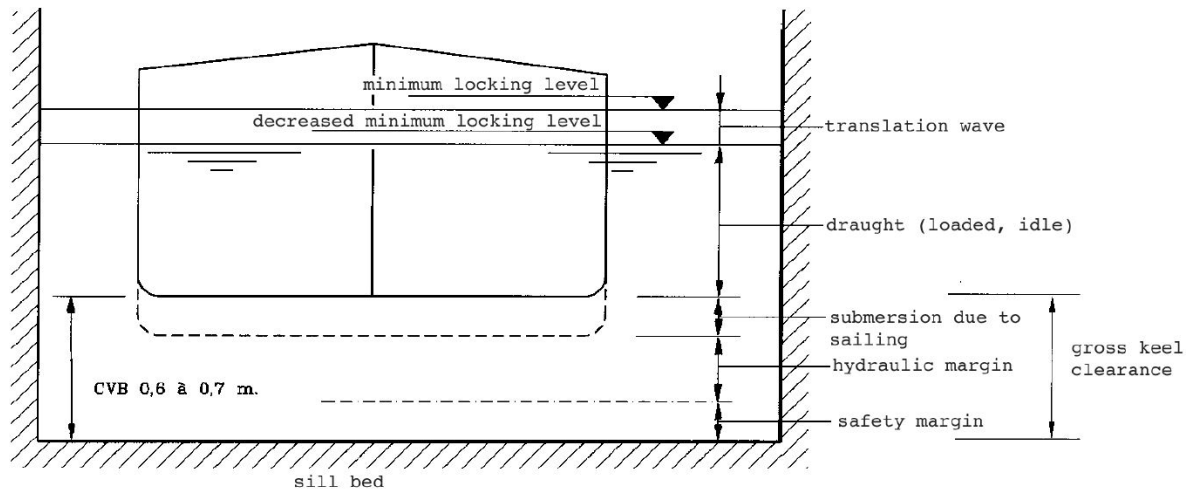


Figure 6-2: Sill Level (Vrijburcht et al., 2000)

The gross keel clearance is a sum of:

- Submersion due to sailing: when a vessel is sailing the draught increases.
- Hydraulic margin: a vessel needs sufficient water to sail, for broad vessels compared to the lock chamber a margin underneath the vessel is desirable.
- Safety margin: to cope with uncertainties in the above factors and the draught of the vessel a safety margin is added.

For inland navigation locks the gross keel clearance depends on the CEMT class. Up to class III $0,60\text{ m}$ is used while for class IV and V $0,70\text{ m}$ is used (Vrijburcht et al., 2000). Rijkswaterstaat (2017) stated that for class VIa and VIb 1.00 m should be used. Leading to a depth per CEMT class as given in Table 6-1.

6-3 Lock Area

Next to the chamber wall a pavement is present. This is used by the lock personnel to walk and to help with mooring of the ships. Also bollards may be present next to the wall as will be discussed in Section 2-2-2. To accommodate these bollards and to ensure safe passage near the chamber the pavement will have a width of $3,0\text{ m}$ at both sides. When the top of the chamber wall is above the surface level the ground needs to be heightened up to the wall height. After the $3,0\text{ m}$ the surface will slope back to the original surface level.

6-3-1 Surcharge Load

According to CUR 166 (2012a) a surcharge load of $10\text{ to }20\text{ kPa}$ is used for the calculation. To be on the conservative side 20 kPa is used in the design-tool.

Chapter 7

Concrete U-Basin

For the construction of locks the concept of a concrete U-basin is widely used. Lock heads are mostly monolithic reinforced concrete structures since a high stiffness is required for the lock gates and different water levels at each end of the head. Lock chambers require less stiffness and can be made of multiple monolith reinforced concrete parts or even with the walls and floors constructed separately and hinge connected.

7-1 Construction

Reinforced concrete U-basins need to be built in a dry building pit, this can be done with different construction methods. Two important limitations for construction are the available space for construction and the possibility for dewatering to lower the ground water level. The groundwater level should be maintained to a level of $0,3\text{ m}$ below the excavation level to ensure a dry and workable construction pit (Normcommissie 351 006, 2017). When the ground water in the building pit is lowered, the ground water in the surrounding area will also be affected. The lowering of the groundwater in the surrounding area may lead to settlements or may harm the agriculture.

7-1-1 Open Building Pit

When dewatering is allowed and sufficient space is available the easiest and economically most feasible construction method is by creating a large open building pit as can be seen in Figure 7-1a. The pit will be pumped to ensure the building pit will stay dry. The slopes of the building pit should not be too steep to ensure stability during construction, this leads to a large amount of soil which has to be excavated and a large area should be available. Depending on the final water levels, the soil parameters, the need for emptying of the chamber for maintenance and the weight of the structure, the U basin can either be built with a shallow foundation, or a deep foundation will be needed. After excavation the pile foundation will be constructed (if applicable) followed by the chamber floor of the U basin. Secondly the walls

will be constructed onto the floor. When the walls are ready a back-fill will be placed up to the favourable height and the pumping will be stopped to restore the original groundwater level. The back-fill material should be sand, when the original soil was sand this material could be re-used. When clay or peat was excavated new material should be imported. The width of the excavation pit should be sufficient to have space for the U-basin, formwork and 80 cm for safe passage during construction. This leads to total excavation width equal to the outer width of the lock plus two extra meters at each side.

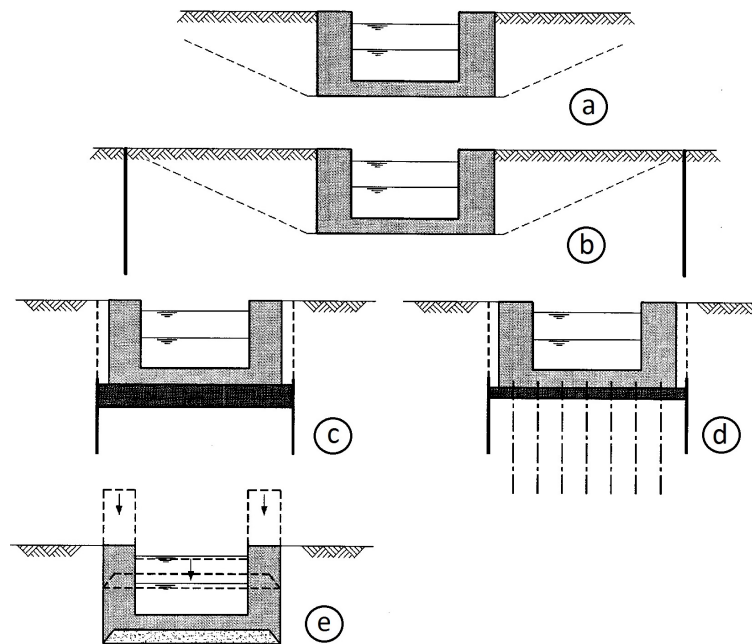


Figure 7-1: Construction Alternatives U Basin(Vrijburcht et al., 2000)

7-1-2 Open Building Pit with Cut-off Screens

When sufficient space is available but pumping is not permitted it is an option to make use of cement-bentonite cut-off screens (Figure 7-1b). These cut-off screens will be constructed down to an impermeable soil layer. In this way the groundwater flow will be stopped and the construction pit can be pumped without heavily influencing the surrounding groundwater levels. The building pit still makes use of sloped earth walls which ensures that the bentonite concrete cut-off walls will not require any structural strength. The impermeable layer should be checked for upbursting. This option is not included in the scope due to the high similarities with a normal open building pit.

7-1-3 Sheet Pile Building Pit

When limited space is available for construction the building pit can be made with sheet piles. In this case the sloped sides of the open building pit will be replaced with vertical sheet piles as can be seen in Figure 7-1c&d. First the sheet piles will be driven into the

ground. The length of the sheet piles is determined by the type of soil, the groundwater levels and the dimensions of the lock. When the sheet piles are in place, the soil in between the sheet piles will be excavated without lowering the groundwater. To ensure water tightness an underwater concrete floor will be constructed. Depending on the circumstances this floor should be anchored or an extra thick floor should be created to prevent uplift. When the underwater concrete floor is finished the building pit can be pumped and the dry construction of the U basin can start. Due to the sheet pile walls and the underwater concrete floor the groundwater levels in the surrounding areas do not change. The underwater concrete floor only has a function during construction. When tension piles are needed for both the building and the user phase, the tension piles should pierce the underwater concrete floor so they can be connected to the floor of the U-basin (Vrijburcht et al., 2000).

7-1-4 Pneumatic Caisson

This type as given in Figure 7-1e is used for lock heads in different locks including the new locks in IJmuiden. This construction method is also possible for lock chambers. The parts of the chamber will be build at ground level. When finished, the soil underneath the caisson will be excavated and the caisson will slowly subside to its final position. This alternative can be a possible solution when very little space is available for construction. Since this alternative is only applicable for special cases and is expensive it is for now not included in the design-tool.

7-2 Dimensions

7-2-1 Walls

The walls should be able to withstand the forces from the soil (Figure 7-2a,d,g), the water (Figure 7-2a,e), the loads from the ships (Figure 7-2b), surface loads (Figure 7-2f,h) and its own weight (Figure 7-2c). The walls will be cast on the floor with a stiff connection. The thickness of the lower part of the wall should be designed on the shear force of which needs to be transferred to the floor since usually no shear force reinforcement (stirrups) is used in lock design (Vrijburcht et al., 2000). With the given thickness the amount of reinforcement can be calculated (Section 7-2-4). The moments and shear force at the upper end of the wall are smaller than near the floor. For this reason the wall thickness will reduce towards the top of the wall. The top of the wall will have a thickness between $\frac{1}{2}$ and $\frac{2}{3}$ of the bottom part (Vrijburcht et al., 2000). In the design-tool $\frac{1}{2}$ of the bottom part will be used with a minimum of 0,5 m to have sufficient space for bollards and recesses for emergency ladders in the wall. The strength of the wall will be checked according to Eurocode 2 (2008).

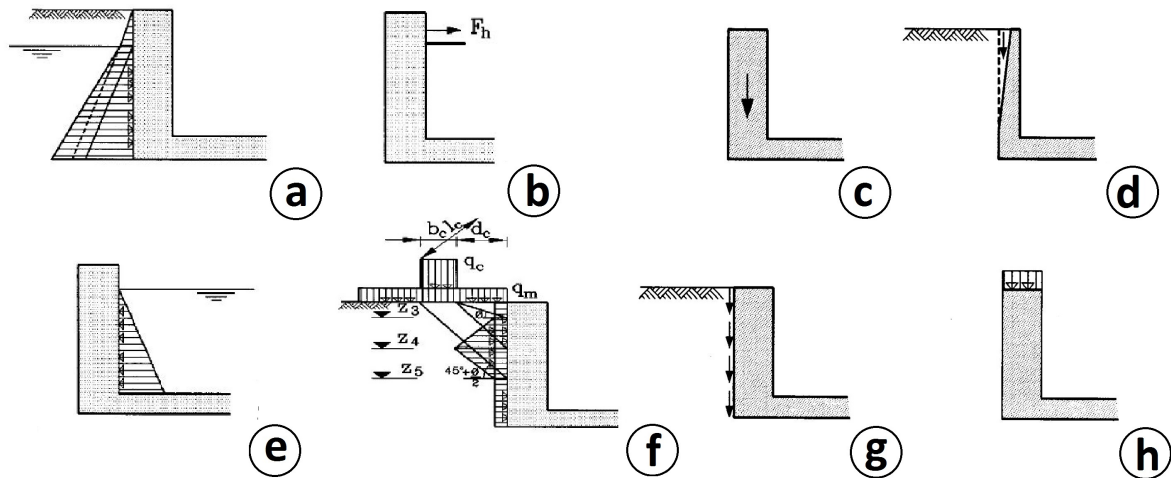


Figure 7-2: Forces Acting on the Wall (Vrijburcht et al., 2000)

7-2-2 Floor

The floor is designed to take the forces and moments coming from the wall (Figure 7-3a), the water pressures from water in the lock (Figure 7-3c), ground water (Figure 7-3d), its own weight (Figure 7-3b) and the impact of a falling anchor.

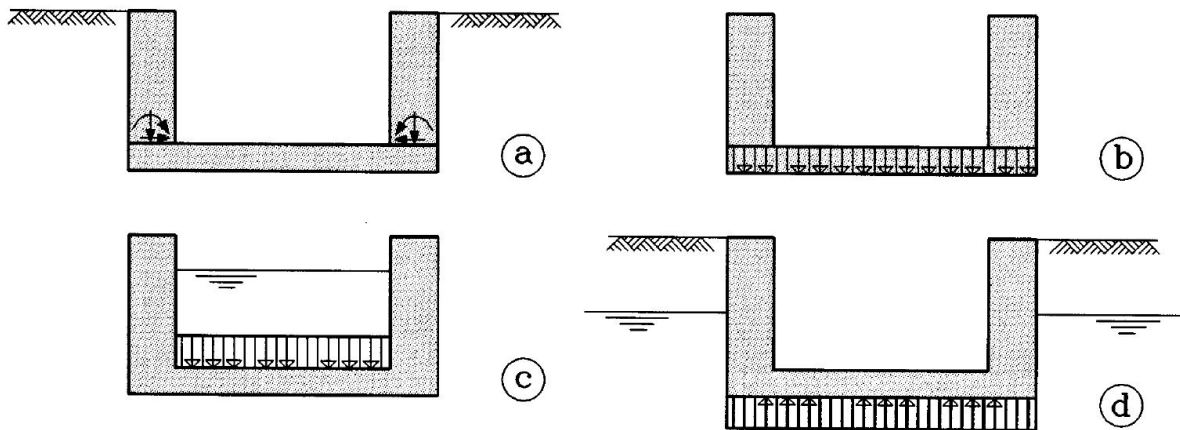


Figure 7-3: Forces Acting on the Floor(Vrijburcht et al., 2000)

Due to the water pressure difference inside and outside the lock the floor will have a moment distribution as is drawn in Figure 7-4a. The moments transferred from the wall have a large influence on the moments in the floor and depending on the magnitude of the water pressure and the moments in the wall the total moment distribution of the floor is as given in Figure 7-4b and Figure 7-4c.

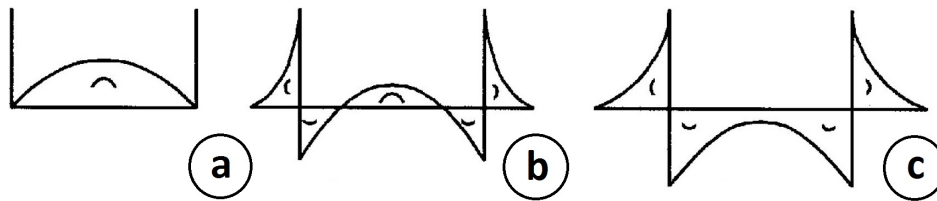


Figure 7-4: Moments Resulting from the Loads (Vrijburcht et al., 2000)

Foundation

When the weight of the chamber is larger than the upward pressure of the water a shallow foundation is preferred due to the simple construction method. With a shallow foundation the floor will be supported by the underlying soil. With a constant load distribution and homogeneous soil the underground acts as a row of constant spring supports as is illustrated in Figure 7-5a. However, due to the moment distribution as given in Figure 7-4b and Figure 7-4c the load transfer to the ground is not constant. The places with high deformations transfer more load compared to the rest of the structure. This will in fact lead to a load transfer as illustrated in Figure 7-5b. This leads to less support in the middle which causes higher stresses in the structure. To predict the exact distribution of the soil support springs geotechnical finite element models are needed. Normcommissie 351 006 (2007) stated in NEN 6744 about 'Shallow Foundation' that the supports springs can be assumed as concentrated under the walls where the highest stresses occur as shown in Figure 7-5c. In this way the floor can be modeled as a beam on two supports, this will lead to a workable calculation which is on the conservative side.

When upward force resulting from the upward water pressures are bigger than the weight of the structure, measures should be taken to prevent uplift. The simplest way is to enlarge the thickness of the concrete floor. In this way the weight of the structure will increase to prevent uplift. Another way to prevent uplift is to make use of tension piles as was described in Section 8-6. This is a more complex solution but when the upward pressure is large this could be the favorable solution. The design-tool will calculate both solutions to check which one is preferred. Usually this is governed by costs but sometimes construction time is decisive.

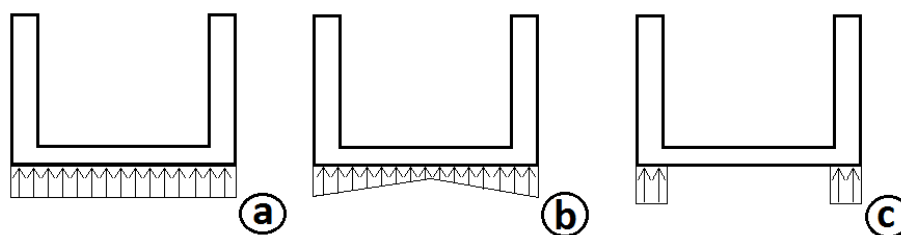


Figure 7-5: Support Springs Soil Shallow Foundation

7-2-3 Concrete

For a good constructability in multiple circumstances concrete of strength class C20/25 up to class C55/67 can be used as concrete input. Extreme high strength concrete is not suitable

for locks since construction should be with constant production circumstances. This is not likely in an outside building pit. The type of concrete can be adapted within the design tool when preferred by the user. The costs of the concrete per cubic meter will not automatically change.

7-2-4 Reinforcement

The amount of reinforcement which is needed in the cross sections will be estimated with the global approximation formula of Braam and Lagendijk (2011) which is given below.

$$A_s = \frac{M_{Ed}}{f_{yd} * 0,9 * d} \quad (7-1)$$

And:

$$d = h - c_{nom} - \varnothing_{stirrups} - \frac{1}{2}\varnothing_{reinforcement} \quad (7-2)$$

In which:

A_s	Needed area of reinforcement per meter of width	$[mm^2]$
M_{Ed}	Design moment at the cross section	$[Nmm]$
f_{yd}	Design yield strength of the reinforcement steel	$[N/mm^2]$
d	Effective depth of the cross section (Figure 7-6)	$[mm]$
\varnothing	Diameter	$[mm]$

The c is the concrete cover and depends on the bond with the reinforcement ($c_{min,b}$), the environmental conditions ($c_{min,dur}$) and the way of constructing (Δc_{def}). The concrete cover which will be used is:

$$c_{nom} = \max(c_{min,b}; c_{min,dur}; 10 \text{ mm}) + \Delta c_{def} \quad (7-3)$$

The $c_{min,dur}$ is based on the environmental conditions in which the lock will be constructed. The class will be determined by Table C-3 in Appendix C. For locks class XC4 and XS3 are common where XC4 only applies when only fresh water is present and when salt water is in the system XS3 should be applied. The $c_{min,dur}$ can be read in Table C-1 of Appendix C. The structural class to be used for the specific location can be found according to Table C-2 from the Appendix C. The structural class for structures with a life time of 50 years start at S4, following the steps of the table will lead to the structural class to use for the concrete cover. Since locks are usually designed for a lifetime of 100 years, and the walls and floor act as a plate the minimum structural class will be S4 (Special Quality Control of the concrete production will not be an input for the tool). This means a minimum $c_{min,dur} = 30 \text{ mm}$. However, within hydraulic engineering usually a minimum cover of 50 mm is used. Especially for the inner sides of the walls and floor due to the high risk of damage. Since it is assumed the diameter of the reinforcement bars will not exceed this value the minimum concrete cover will be:

$$c_{nom} = c_{min,dur} + \Delta c_{def} \quad (7-4)$$

for Δc_{def} a value of 5 mm is used, this is given in Eurocode 2 for casting concrete on even surfaces with the maximum size of aggregate (d_g) not exceeding 32 mm (British Standard Institution, 2008).

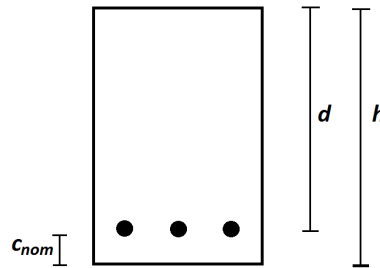


Figure 7-6: Effective Height Concrete

The diameter of the primary reinforcement ($\varnothing_{reinforcement}$) will start in the model with 16 mm . With the needed area of reinforcement the number of bars will be calculated. The model will check whether the spacing of the bars is sufficient, and if needed the diameter of the primary reinforcement will be adapted. The spacing between the bars should be the maximum value of:

$$\max(k_1 * \varnothing_{reinforcement}; d_g + k_2; 20\text{ mm}) \quad (7-5)$$

k_1 and k_2 are respectively 1 and 5 mm . With the maximum aggregate size of 32 mm (d_g) the minimum space should be 37 mm .

Secondary Reinforcement

For slabs the secondary transverse reinforcement should be at least 20% of the principal reinforcement as calculated in the previous section. In the model the value of 20% will be used, in the detailed design phase this value should be checked.

7-3 Soil Loads

Since the U-basin can be seen as a strong stiff structure the horizontal soil pressures are given with a neutral soil pressure coefficient as given below.

$$\sigma'_h = \sigma'_v * K_0 \quad (7-6)$$

In which:

σ'_h	effective horizontal soil pressure	$[N/mm^2]$
σ'_v	effective vertical soil pressure	$[N/mm^2]$
K_0	neutral soil pressure coefficient, $1 - \sin(\phi)$	$[-]$

Sheet Pile Wall Structure

One of the lock types which will be considered is based on walls which are constructed with sheet piles. Multiple types of floors can be used when applying sheet pile walls. Anchors can be introduced to strengthen the structure. In Figure 8-1 an overview of a sheet piles lock chamber is given.

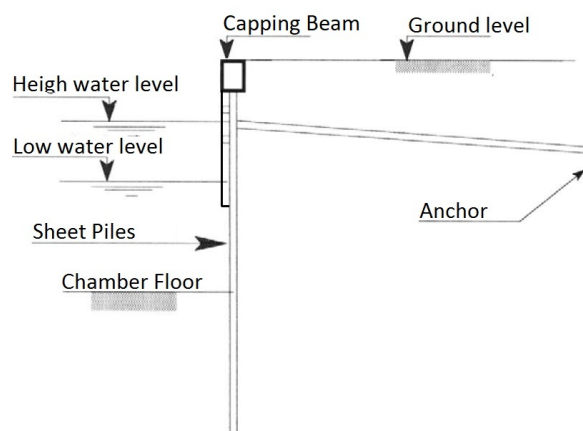


Figure 8-1: Sheet Pile Wall Overview (Vrijburcht et al., 2000) (modified)

8-1 D-Sheet Piling

The calculation of the sheet pile walls will be done in the program D-Sheet Piling from Deltares (2018). D-Sheet Piling is a 2D program which implements the Dutch design code CUR publication 166 and the European design code Eurocode 7 method including the Eurocode annexes for The Netherlands and Belgium. The Python script of the design-tool creates an input file for D-Sheet Piling according to the boundary conditions and the user preference. D-Sheet Piling will be controlled through the Python script to be executed in the background. The result of the calculation will be reviewed and checked by the Python script. When the

design requirements are not met, the input file will be automatically adapted and send back to D-Sheet Piling. When the design requirements are met, the input file will be adapted as well to achieve a more efficient design. This will be an iterative process.

8-2 Sheet piles

The sheet piles in D-Sheet Piling are modeled as an elasto-plastic beam on a soil foundation, represented by uncoupled elasto-plastic springs. Normal forces can be introduced in the sheet pile and will be calculated for additional moments and displacements. D-Sheet piling includes a library with standard sheet piling profiles, but manual entry is possible as well. The sheet piles will be checked on maximum bending moment, maximum shear stress and displacement. The displacement is linked to the total stability of the structure. When the displacements are too large the structure will be unstable and fail.

8-2-1 Profiles

The user can set a preferred sheet pile profile or let the design-tool calculate different profiles to find an optimum. The design tool makes use of AZ profiles. These profiles are widely used in hydraulic engineering. An range of profiles from 'AZ 13-700' up to 'AZ 48' overview of the included profiles is given in Appendix E.

8-2-2 Combi Walls

When the retaining height is high or high forces act on the retaining structure a normal sheet pile wall will not have sufficient strength. To strengthen a sheet pile wall piles can be place in between the sheet pile elements, this makes a combi wall. For the piles HZ profiles from ArcelorMittal are used with a range of 'HZ 880M A' up to 'HZ 1180M A' (Appendix E). For the sheet piles the profile 'AZ 18-700' is commonly used. Larger profiles are not efficient since the sheet piles of the combi wall mainly have a ground retaining and force transfer function, the strength is coming from the HZ profiles. To make sure the piles and the sheet piles act as one structure, not more than four sheet piles may placed in between the piles. Due to practical reasons of installation and layout one or three sheet piles in between the piles are not taken into account. For this reason only two and four sheet piles in between the piles will be calculated in the tool (Figure 8-2). When even higher retaining heights are needed steel round piles can be used instead of the HZ profiles. These are not yet included in the Design-tool but can be entered when needed.

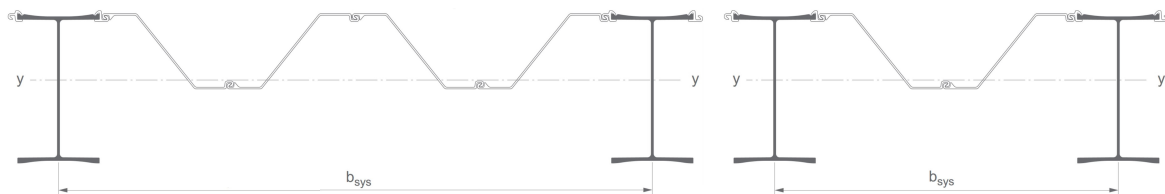


Figure 8-2: HZ-AZ Combi Wall System, 4 and 2 sheet piles in between (ArcelorMittal Sheet Piling, 2014) (modified)

8-2-3 Modification Factor

The Modification factor (k_{mod}) takes into account the life time of the structure. The Dutch code NEN 6702 makes a distinction between short term and long term situations. For synthetic material a modification factor of respective 0.45 and 0.5 is used. For steel the modification factor is 1.0.

8-2-4 Interlock reduction factor

The flexural stiffness of the sheet pile should be reduced when insufficient shear force transmission is present in the interlocks. When using U-piles a reduction factor (β_{D}) should be applied to take into account the lower flexural stiffness. To overcome this problem the U-piles can be welded together to ensure shear force transmission (British Standard Institution, 2007). With Z-piles the connection in the interlocks does not influence the flexural stiffness since the interlocks are situated at the outer fiber of the section. For this reason only Z-piles will be used for the lock chamber design in this design tool.

8-2-5 Reduction Factor Moment Capacity

Due to the possible lack of shear force transmission the maximum moment of the sheet pile should be reduced with β_{B} as with the flexural stiffness. This is only applicable for U-piles.

8-2-6 Material Factor

D-Sheet uses the Eurocode material factors for the profiles within the library. For steel $\gamma_{\text{m}} = 1$ and for concrete $\gamma_{\text{m}} = 1.1$. Since the profiles from the libraries are used no manual input is needed.

8-2-7 Delta Friction Angle

To deal with a large range of soils the Kötter method will be used. The Kötter method makes use of curved slip surfaces and is, in contrast to the Müller-Breslau method applicable for soils with internal friction coefficients over $\varphi' = 30^\circ$.

The Delta Friction Angle depends on the type of wall and is given in Table 8-1, for peat $\delta = 0$. For sheet piles the roughness is 'rough' leading to the delta friction angle with curved slip surfaces given as:

$$\delta = \min(\varphi' - 2, 5^\circ; 27, 5^\circ) \quad (8-1)$$

Table 8-1: Delta Friction Angles (British Standard Institution, 2009)

Wall roughness	Delta Friction Angles (δ)	
	Straight slip curve	Curved slip surface
Toothed	$0,67\varphi'$	$\leq \varphi'$
Rough	$0,67\varphi'$	$\leq \varphi' - 2,5^\circ$ and $\leq 27,5^\circ$
Semi Rough	$0,33\varphi'$	$0,5\varphi'$
Smooth	0	0

8-3 Anchors and Struts

To limit displacement and change the moment distribution, anchors or struts can be introduced. Struts might be used during construction to ensure stability. However, struts are not suitable when the lock is put in to use due to the limitation of head clearance. Anchors and struts are modeled using discrete springs. Anchors are introduced with a given stiffness, cross section of the bar, length, angle, design yield force and the wall height according to the slip surface theory of Kranz (1953). The design-tool will generate anchors automatically to fulfill the safety and stability requirements. The anchors will be connected to the sheet pile. At the level of the anchors normally a girder will be installed to equally transfer the load from the anchors to the sheet piles. In the case of a lock chamber a capping beam will be used as explained later in this section.

8-3-1 Grout Anchors

When a sand layer is present next to the sheet piles, grout anchors can be applied. A thread bar or strand tendon will be drilled from the sheet piles into the surrounding soil. Around the bar or tendon grout will be injected under high pressure, Figure 8-3. At the end of the bar or tendon a mixture of cement and water is injected with a pressure between 0,5 and 1,5 MPa. The grout will be distributed in between the pores of the sand soil, this will lead to compaction of the soil which results in a high friction force which can be developed. To ensure full effectiveness of the anchor the upper part of the grout anchor should be at least between 0,5 m and 1,0 m under the upper level of the sand layer (CUR-commissie C126, 2012b).

In the tool the anchors are standardized and the same for each calculation. A Leeuw Anchor (Appendix B) $\varnothing 101,6 \times 22,2$ will be used with a bar diameter of 101,6 mm and a thickness of 22,2 mm. A force of 2282 kN can resisted by this anchor. This type of anchor is often used in hydraulic projects. The anchor will be installed at 1 meter below the top of the structure. The angle of installation is 30° and the length is 30 m. It is assumed that with these values a sand layer can be reached. A grout body with a diameter of 250 mm and a length of 5 m will be used for the MKI calculation. For more detailed design it should be checked whether a different anchor or a different angle is preferred.

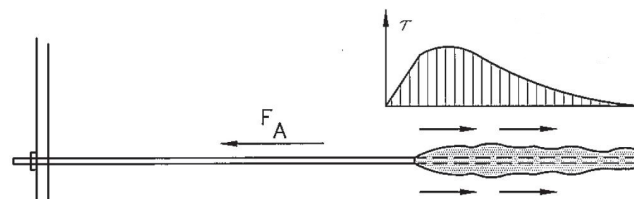


Figure 8-3: Grout Anchor (CUR-commissie C126, 2012b)

For calculation in D-Sheet Piling grout anchors are modeled according to the Method of Kranz (1953) where the anchor length is defined as the sum of the length of the free anchor length (L_{free}) and half of the length of the grout body (L_A) (CUR-commissie C126, 2012b). This is pictured in Figure 8-4.

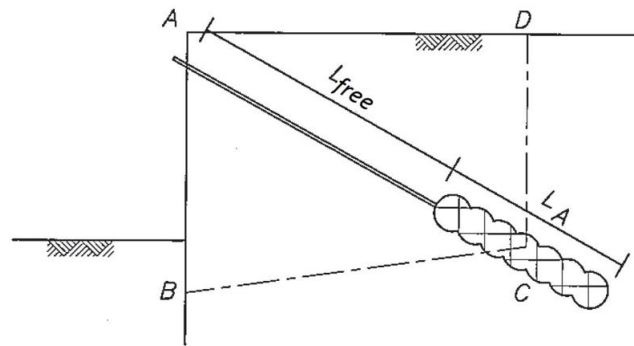


Figure 8-4: Anchor Length Grout Anchor (CUR-commissie C126, 2012b) (modified)

8-4 Underwater Concrete

The floor of the sheet pile wall structure will be made of underwater concrete (UWC). More stiffness will be added to the sheet pile structure since the underwater concrete floor will act as a strut. When dewatering of the chamber is required an UWC floor has to be constructed to ensure water tightness. In D-Sheet the UWC floor can be modeled as a special ground layer according to Deltares systems (2016).

8-4-1 CUR 77:2014

To design the UWC floor the CUR 77:2014 recommendation will be used. It includes the design rules, detail engineering and dimensioning of temporary unreinforced underwater concrete floors in between retaining walls and anchored by tension piles. The CUR 77:2014 is based on the Eurocodes. The floor should have a minimum main thickness of 800 mm. The CUR 77:2014 is only applicable when one type of tension pile is used with constant stiffness and where the floor is modeled as a beam. This is no problem since the design-tool will be used for preliminary design. (Jonker and Hagenaaars, 2015).

CUR 77:2014 calculates two failure mechanism for the UWC floor.

- Failure Mechanism A: When the normal force is small, the failure load of the pressure arc system after cracking is less than the load where cracking starts to occur. In this case failure will occur when the tension strength of the concrete is exceeded and cracks will occur.
- Failure Mechanism B: With large normal forces in the floor the strength of pressure arc system exceeds the crack load. Failure will occur when concrete compressive strength is exceeded.

The loads that will be considered for the underwater concrete floor are: the self weight of the floor ($q_{g,rep}$), the upward water pressure ($q_{w,rep}$), the horizontal normal force caused by the sheet piles ($F_{strut,rep}$) and the upward swell pressure ($q_{z,rep}$). Different load combinations apply the length direction of the lock and the width direction of the lock. For the calculation of the underwater concrete floor it is assumed that the transfer of shear force to the retaining wall is fixed. This is due to the friction resulting from the strut force and inward bending of the retaining walls above the underwater concrete floor.

The width direction will be calculated as a beam on spring supports with free supports at the retaining walls. The entire strut force from the retaining wall is present as a normal forces which causes a moment due to the eccentricity of the support. The length direction will be modeled as a beam on hinged supports without a normal force. An exception has to be made at locks with a width larger than 40 meters. In these cases the width direction should also be checked as if it is a length direction.

The tolerance of the height of the underwater concrete floor depends on both the soil underneath and the method of placing the concrete (Jonker and Hagenaaars, 2015). When a sand layer is present below the floor the tolerance at the bottom side of the floor (tol_{under}) will be 150 mm. When the layer below is not sand an additional filling layer of sand should be constructed to come to a 150 mm tolerance. When clay is present it can be chosen to use a tolerance of 350 mm without using an additional filling layer, it should be checked which is more economical feasible. For the tolerance of the upper side of the floor it depends if the concrete is casted with the Hop-Dobber-Method or with the Tube- or Hydro Valve Method. A tolerance (tol_{upper}) of 75 mm or 150 mm will be applied respectively.

8-4-2 Concrete

To minimize thermal cracking in the under water concrete floor the low strength C20/25 will be used as standard. However, the type of concrete can be adapted within the design tool when preferred by the user. The costs of the concrete per cubic meter will not automatically change.

8-5 Conservation

Since the focus of the design tool will be on inland navigation locks, the environment is not highly aggressive. Extra measures for conservation are not obligated. Beside this the sheet piles are mostly not exposed to the air. From the capstone up to a meter below the lowest

lock level a concrete cover will be present as will be discussed in Section 8-7. However, when preferred in the design measures for conservation are present in the tool. The first measure is additional thickness of the sheet piles to compensate for loss of strength due to corrosion. This extra thickness can be added by the user of the tool and will be present over the whole sheet pile. This will lead to more material and thus a higher price. A second way of conservation is by applying cathodic protection. This will only be needed for the surfaces which are not covered by concrete or soil. Cost will be depending on the surface which needs protection. The last option is by applying special coating to protect the sheet piles.

8-6 Tension Piles

For the calculation of tension piles two CUR regulations are present, CUR 2001-4 (2003) and CUR 236 (2011). CUR 2001-4 describes the design rules for tension piles and includes different types of tension piles. CUR 236 is called "Tension Anchors" and only focus on piles which are produced and formed in the ground. In some specific cases compression can be expected in the piles. To make the design tool applicable for all circumstances solid prefabricated tension piles will be used. When the design will be in a further stadium in the ground formed piles should be added to the options. Since CUR 236 does not take into account solid prefabricated tension piles, CUR 2001-4 will be used in the design-tool for the calculation of tension piles. The user is able to choose the dimensions in of the prefabricated concrete tension pile.

With the given ground parameters, water levels and dimensions of the lock an upward pressure is calculated. The pressure should be compensated by both the weight of the underwater concrete floor and the tension piles. When assuming a height of the underwater concrete floor, the force which has to be taken by the tension piles is known. The CUR 2001-4 is designed to calculate the force which can be taken per pile in a given alignment. This is done with the use of the ground parameters, the type and dimensions of the pile and the method of installation. The design tool will give an optimum for the alignment, type and dimension of the piles to fulfill the needed tension force. The strength of the underwater concrete floor depends on the alignment of the piles which means that the height might have to change due to the calculated pile alignment. This will lead to a different tension force which needs to be adopted by the piles, this will lead to an iterative process.

8-6-1 Pile Alignment

For the strength of the underwater concrete floor it is important that the distance between the piles in the width direction is small to transfer the loads evenly. However CUR 2001-4 stated that the distance between the piles in the length direction can not be larger than the distance between the piles in the width direction.

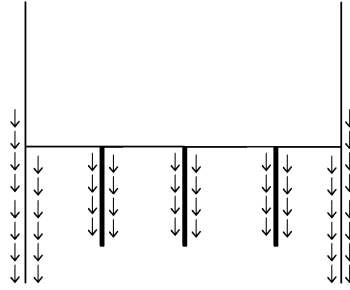


Figure 8-5: Shaft Resistance Tension Piles and Sheet Piles in Tension.

8-6-2 Axial spring stiffness

CUR 236 (2011) defined a method to calculate the axial spring stiffness of anchor piles. This stiffness is needed for the calculation of the underwater concrete floor or an other structure which is on top of the piles. The stiffness of the piles influence the moment distribution of the above structure. CUR 77:2014 (2015) made a summary of the Axial spring stiffness of CUR 236 which is also applicable for prefab concrete piles and based on the use of an underwater concrete floor. Due to the upward pressure of the water the axial spring stiffness is based on the downward shaft resistance and the extension of the piles as can be seen in Figure 8-5. The axial spring stiffness (k_{axial}) in kN/m is calculated in the following way.

$$\frac{1}{k_{axial}} = \frac{1}{k_{shaft}} + \frac{1}{k_{elastic}} + \frac{1}{k_{soil}} \quad (8-2)$$

In which:

k_{axial}	Axial spring stiffness of the pile	$[kN/m]$
k_{shaft}	Spring stiffness coming from the mobilization of the shaft friction	$[kN/m]$
$k_{elastic}$	Elastic extension of the pile	$[kN/m]$
k_{soil}	Lift of the soil under the pile tip caused by unloading	$[kN/m]$

The k_{shaft} follows from the secant value from Normcommissie 351 006 (2017). It uses the maximum shaft resistance of the pile in the extreme condition as calculated in CUR 2001-4 (2003) and depends on the type of pile used. This leads to the following values:

Displacement Piles : $k_{shaft,rep} = 330 * R_{s,cal,max,i}$

Non-displacing Piles : $k_{shaft,rep} = 120 * R_{s,cal,max,i}$

Cast-in-situ Anchor Piles : $k_{shaft,rep} = 120 * R_{s,cal,max,i}$

In which:

$$R_{s,cal,max,i} \quad \text{Shaft friction} \quad [m^{-1}]$$

CUR 77 (2015) state that the elastic spring stiffness of the pile is calculated assuming the load transfer from the pile to the soil is linear in height (the first meter does not contribute due to the driving of the pile). As can be seen below:

$$k_{elastic} = \frac{EA_p}{L_1 + 1/2(L - L_1)} \quad (8-3)$$

In which:

E	Youngs modulus of the pile ($20 \times 10^6 \text{ kN/m}^2$ for concrete prefab piles)	$[\text{kN/m}^2]$
A_p	Pile cross section	$[\text{m}^2]$
L_1	Pile length which contributes to the load transfer	$[\text{m}]$
L	Pile length	$[\text{m}]$

For the calculation of the forces in the underwater concrete floors the k_{soil} is given as infinite (Jonker and Hagenaars, 2015).

At the side ends of the underwater concrete floor retaining walls are present to support the underwater concrete floor. Like tension piles the retaining walls have an axial spring stiffness which influence the moment distribution of the underwater concrete floor and is based on the downward shaft resistance and extension of the retaining walls. However, for the horizontal stability and strength of the retaining walls anchors are placed. To reach strong soil layers these anchors are often placed under a angle downward. This angle results in a downward force in the sheet piles as can be seen in Figure 8-6. When these forces are larger than the upward force from the underwater concrete floor the axial spring stiffness changes sign. For the moment distribution of the underwater concrete floor the axial spring stiffness of the sheet piles will be infinity and thus act as normal hinged supports. Due to this stiffness it is almost certain that additional measures should be taken at the connection between the underwater concrete floor and the sheet piles.

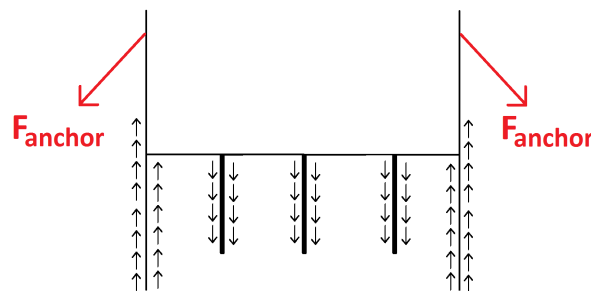


Figure 8-6: Shaft Resistance Tension Piles and Sheet Piles in Tension with Anchors

8-7 Capping Beam

The capping beam is a reinforced concrete top structure of the sheet piles as can be seen in Figure 8-1. On or in the capping beam bollards will be installed and optional anchors will be connected to the capping beam. The capping beam will be designed to equally transfer the

load from the bollards to the sheet piles and the anchors and to transfer the soil load from the sheet piles to the anchors. It has to be a stiff structure which mainly need to resist the shear forces caused by the anchor and bollard loads.

The force in the anchors caused by the horizontal soil and water pressure is given by distance between the anchors times the evenly distributed horizontal pressure. The maximum moment in the capping beam caused by these horizontal pressures is given by:

$$M_{max,soil} = \frac{1}{8} * q_{soil} * l_a^2 \quad (8-4)$$

In which:

$M_{max,soil}$	The maximum moment caused by the horizontal pressures coming from the sheet piles	$[kNm]$
q_{soil}	The horizontal pressures which are anchor forces resulting from the D-Sheet calculation divided by the distance between the anchors	$[kN/m]$
l_a	Distance between the anchors	$[m]$

The bollard forces lead to a shear force in the capping beam with a maximum of the total bollard forces at the place of the bollard. The moment caused by the bollard loads depends on the stiffness of both the capping beam as the stiffness of the supporting anchors. When the capping beam is extreme stiff compared to the anchor stiffness the force will be distributed equally to all the anchors leading to large moments in the capping beam due to the long length of the capping beam distributing the force. When the stiffness of the anchor supports is extreme high compared to the stiffness of the capping beam the anchors next to the bollards absorb major part of the force leading to a short length and thus smaller moments. Since the tool is used for preliminary design and due to the complexity of this calculation a conservative assumption will be done. The capping beam will be calculated as a beam on two supports with a length given by the length between two bollards as given in Section 2-2-2.

$$M_{max,bollard} = \frac{1}{4} * F_{bollard} * l_b \quad (8-5)$$

In which:

$M_{max,bollard}$	The maximum moment caused by the bollard loads	$[kNm]$
$F_{bollard}$	The governing bollard load	$[kN]$
l_b	Distance between the bollards	$[m]$

The capping beam will be designed based on the governing moments, when needed stirrups will be included to resist the shear force.

The structural part of the capping beam with the moment and shear reinforcement will be placed on top of the sheet piles. The width of the capping beam should have a minimum for a good connection with the sheet piles and space should be available for ladder and bollard recesses. For the connection with the sheet piles the width being the height of the sheet pile plus 200 mm on each side should be sufficient to transfer the forces. For the installation of

the bollards and ladder recesses 500 mm is needed on the water side. The recesses can be located in the sinus of the sheet pile, Figure 8-7, this lead to a required additional thickness on the water side of 500 mm minus the height of the sheet pile. This structural part will have a height of 1000 mm to ensure the load will be transferred correctly.

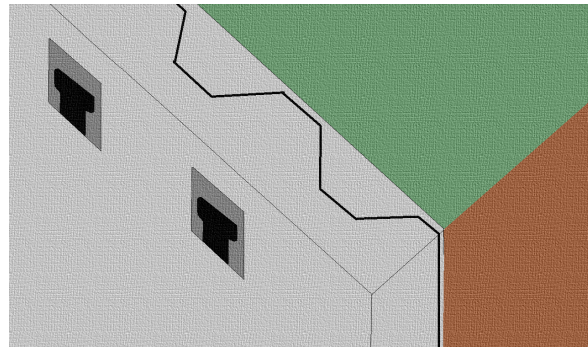


Figure 8-7: Bollard Recesses Capping Beam

To fulfill the requirement of a smooth chamber wall to reduce damage by ship collision the capping beam will continue to the lowest lock level. Here only the water side of the wall needs cover, this reduces the needed amount of concrete.

Because of possible damages due to collisions with ships, the capping beam and the smooth concrete surface at the sheet piles will need a concrete cover of 60 mm on the water side and 50 mm on the ground side (Vrijburcht et al., 2000). The values described above are including the concrete cover.

The reinforcement within the capping beam will be calculated as discussed in Section 7-2-4. The diameter of the reinforcement bars is set as 16 mm . An optimization in the reinforcement should take place when making a detailed design.

8-8 Construction

The purpose of the design-tool is preliminary design, because of this, not the whole construction scheme will be calculated and optimized. The steps of the excavation will not be calculated, this is for more detailed design and since it is possible to place anchors at certain moments of excavation the steps will not be limiting forces. The structure will be checked for three main load combinations. The first is at the end of excavation when the anchors are installed (Figure 8-8b), no underwater concrete floor is present but water is still in the chamber at groundwater level. The second check will be when the underwater concrete floor is in place and the chamber will be dry or at maintenance level, this is assumed the final stage before usage and as the maintenance stage (Figure 8-8c). And the last stage which will be checked is the normative load combination during the usage phase (Figure 8-8d). The water level will be the lowest lock level and the loads caused by the vessels in the lock will be present. When sea going vessels are present in the lock both the loads origin from the sea going vessels as well as the loads normative loads of the inland waterway or recreational vessels are calculated. This means that the tool does not include the construction method where the sheet piles are driven from the water and soil is deposited behind the wall after which the anchors are installed.

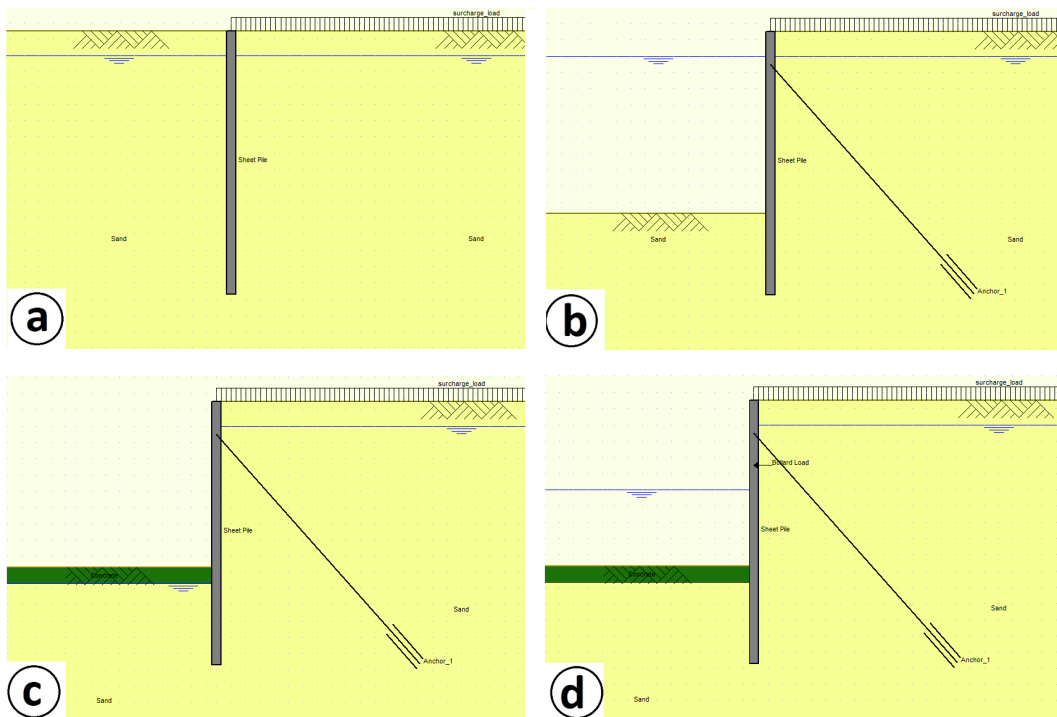


Figure 8-8: Construction Stages Sheet Pile Chamber

Chapter 9

Diaphragm Wall Structure

A diaphragm wall is like sheet piles a soil retaining structure. First a trench is excavated which later will be reinforced and filled with concrete. To make sure the trench does not collapse during construction the trench is filled with the supporting fluid bentonite. When the concrete is casted the bentonite will be pumped away. The construction method is given in Figure 9-1 and the components of the diaphragm wall are given in Figure 9-2. The diaphragm wall consists of multiple panels. The excavator creates one, two or three sleeves which are supported by bentonite, reinforced and casted with concrete. When this process is done the excavator starts with a new panel. For the design tool calculation will be done with three sleeves since this is to most common way and efficient for reinforcement as will be explained in Section 9-2. The thickness of the wall can be made between 600 mm and 1500 mm , due to installation limitations a smaller thickness is not possible (Huybrechts and van Lysebetten, 2014). Larger thicknesses are possible but since the retaining heights for locks are limited this will not be necessary to look into.

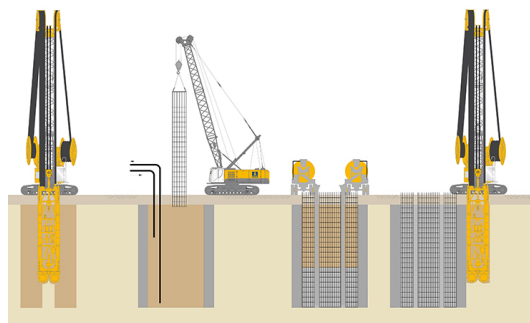


Figure 9-1: Diaphragm Wall Construction (BAUER, 2015)

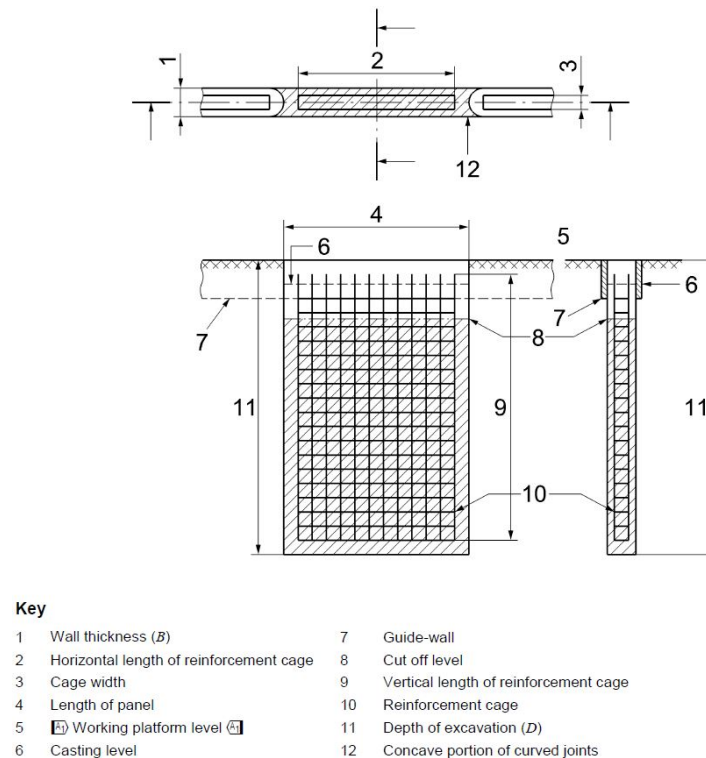


Figure 9-2: Components Diaphragm Wall (Normcommissie 351006, 2015)

9-1 Stability

9-1-1 Final Stability

The stability of the finished diaphragm wall mainly depending on the soil parameters, the retaining height and the external forces acting on the structure. Because a diaphragm wall is considered a stiff structure the natural soil pressures is used instead of the active or passive soil pressure which were used in Section 8. To calculate the stability and the occurring moments, forces and displacements the program D-Sheet will be used. This will be further evaluated in Section 9-4.

9-1-2 Stability during construction

A distinction is made between macro and micro stability. When a trench is excavated the soil pressures change. Due to the difference in horizontal soil pressure the trench is expected to collapse. This is called macro stability. To overcome this problem the trench is filled with bentonite. The length of the panel also has an influence on the macro stability. Another form of instability is when individual particles fall from the wall of the trench into the trench. This is called micro stability. This can also be prevented by using a specific support fluid. For the design tool no further attention is given to this stability during construction. For the panel length will be chosen as 7 meter since this is a widely used length for a three sleeve panel.

To ensure the top of the trench does not collapse due to activity on site a concrete guidance structure is installed. The guidance structure also contributes to the accuracy of the excavator while digging the trench. The guidance structure will be removed when the diaphragm wall is in place.

9-2 Reinforcement

To reinforce the panels reinforcement cages are used. The cages will be lowered into the sleeves. Depending on the length of the panel multiple cages will be placed. For the three sleeve panel of 7 meters which will be used in the design tool two cages will be used. The cages will be designed to ensure good flow of fresh concrete to easily reach all parts of the wall following the design rules of NEN-EN 1538 (Normcommissie 351006, 2015).

9-2-1 Vertical Reinforcement

The vertical reinforcement is needed to resist the moments in the wall induced by the soil. The vertical reinforcement will be placed on both sides of the wall. The minimum diameter of the bars is 12 *mm*. To be consistent with the sheet pile design a diameter of 16 *mm* will be used for preliminary design. Per meter of panel at least 3 bars need to be present on each side. A minimum distance between the vertical bars should be 100 *mm*. Depending on the occurring moment one or more layers will be used with a separating distance of 28 *mm* (COB/CUR-commissie T144/C174, 2010). For this tool 2 layers will be used.

9-2-2 Horizontal Reinforcement

To prevent movement of the vertical reinforcement, horizontal reinforcement is used. The minimal distance between the bars is 200 *mm* to ensure good flow of the fresh concrete. Since the moments are mostly taken by the vertical reinforcement the horizontal reinforcement does not need to be continuous over multiple cages. A value of 20 % of the vertical reinforcement is used as first approximation of the horizontal reinforcement.

9-2-3 Cages

When multiple cages are used in a panel they should be installed with a joint distance of 400 *mm*. The distance between the end of the panel and the cage should be 200 *mm*. This means that at this places no vertical reinforcement is present. When calculating the resisting moment of the diaphragm wall this should be taken into account.

9-2-4 Concrete Cover

For hydraulic structures a minimum cover of 50 *mm* is used. However, for diaphragm walls a minimum of 75 *mm* is used to deal with the uncertainties of the execution.

9-2-5 Final Amount of Reinforcement

Due to the occurring moment in the diaphragm wall vertical reinforcement should be placed on both sides of the wall. However, the resulting moments differ over height as is schematized in Figure 9-3. So when the amount of reinforcement needed for the maximum moment is used over the whole height of the wall the design will be too conservative. For each construction phase it should be analyzed which moments occur and where. For the tool an approximation is made that at the side where tension occurs the reinforcement is placed to resist the maximum moment. At the compression side of the wall the minimum amount of vertical reinforcement will be installed. For this reason, to predict the total amount of reinforcement used one side of maximum reinforcement is taken into account.

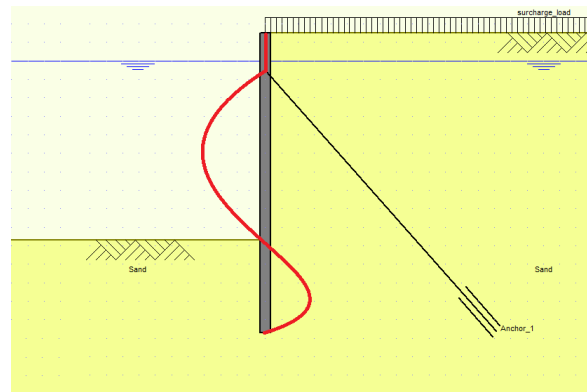


Figure 9-3: Schematized Moment Diagram Diaphragm Wall

9-3 Resisting Moment

To calculate the strength of the construction the method from Eurocode 2 (British Standard Institution, 2008) will be used. First the resisting moment of a panel is calculated. The resisting moment follows from:

$$M_{Rd} = f_{y,d} * A_s * z \quad (9-1)$$

And:

$$z = d * \left(1 - \frac{A_s * f_{y,d}}{2 * b * \eta * f_{c,d}} \right) \quad (9-2)$$

where:

A_s	The amount of vertical reinforcement in a reinforcement cages. Not the whole panel will is equipped with reinforcement due to the clearance between the reinforcement cages and the end of the panel. To deal with this the amount of reinforcement should be calculated per panel and not per meter	$[mm^2]$
b	Overall width cross section	$[mm]$
d	Effective depth of a cross section	$[mm]$
$f_{y,d}$	Design yield strength of steel	$[N/mm^2]$
$f_{c,d}$	Design concrete compressive strength	$[N/mm^2]$
$M_{R,d}$	Design moment resistance	$[Nmm]$
z	Lever arm of internal forces	$[mm]$

9-4 D-Sheet Piling - Diaphragm Wall

9-4-1 Diaphragm Walls

Delta Friction Angle

For diaphragm walls the roughness is 'semi rough' following from Table 8-1 the delta friction angle with curved slip surfaces is given as:

$$\delta = 0,5 \varphi'$$

9-4-2 Anchors and Struts

For the diaphragm wall the same anchors will be used as discussed in Section 8-3-1. The anchors will be connected to the capping beam to transfer the forces.

9-4-3 Underwater Concrete and Tension Piles

Also the UWC floor and the tension piles are calculated in the same for the diaphragm wall structure as the sheet pile wall structure (Section 8-4 and Section 8-6). The UWC floor can differ a bit since the normal force coming from the retaining walls differ per structure. The rest of the values are the same because the water levels and excavation levels do not differ for the retaining wall structures.

9-5 Capping Beam

The capping beam is as explained in Section 8-7 calculated to distribute soil load and ship forces to the anchors. Like the alternative with sheet piles a smooth concrete cover is needed to limit damage from ship collisions. To ensure sufficient space for bollard and ladder recesses a thickness of 650 mm will be used as cover. This is based on the value which was used at the construction of the new lock of Lith (Vrijburcht et al., 2000).

Due to mixture with support fluid the first 0,5 to 1,5 *m* is of insufficient quality. This part should be demolished and replaced by new concrete. This will be combined with the construction of the capping beam. The minimum height of the capping beam thus will be the length of diaphragm wall which will be demolished. For the design tool the average value of 1,0 *m* is used. For this meter the width of the capping beam will be the 650 *mm* for the cover plus the original thickness of the diaphragm wall. The rest of the capping beam, up to the lowest lock level, will only have the width of the cover.

Chapter 10

Costs

To quantify the alternatives of the different lock chambers cost will be taken into account. With certain unit prices and the amount of material which is used in the chambers a cost estimation can be made. The unit prices will be discussed in this chapter and are summarized in Table 10-1.

10-1 Cost Structure

The total costs for the execution of a project like a lock consist of multiple components. For civil projects in the Netherlands the cost estimation is done based on the 'Standard System for Cost Estimates in the Geo, Infrastructure and Water Sector' (CROW, 2002). The main components defined by CROW (2002) are:

- Construction Costs
- Real Estate Costs
- Engineering Costs
- Additional Costs

The components are subdivided in the following cost types:

- Foreseen/ unforeseen
- Direct/ indirect
- Known/ to be elaborated

The direct, indirect, known and to be elaborated costs are within the foreseen costs. The unforeseen costs are separate from this. Rijkswaterstaat (2004) describes the components as given below.

Construction Costs These are all the costs that have to be made for the physical realization of the project. The direct costs are the costs made for materials, equipment and man-hours. The indirect costs consist of non-recurring costs, time-related costs, general costs and risk in construction.

Real Estate Costs These are all the costs related the property where the project will be executed. The direct costs are the purchase of the land or the lease of the area. Indirect costs are all the expenses which are needed to deal with the purchase of lease of the land. Examples are inventory costs, taxation costs and transfer fees.

Engineering Costs These are all the costs for planning, tender, administration, supervision and management.

Additional costs These are all the costs which are not included in the previous components. Examples are costs for archaeology, taxes, permits and possible compensation.

For this Design-tool only the foreseen direct construction costs are taken into account. For a further cost estimation more research should be done for the specific location.

10-2 Sheet Piles

The total cost of the sheet piles consist of multiple components which will be discussed in this section.

10-2-1 Material

The material of the sheet piles is steel. The costs are only based on the weight of the steel. A different sheet pile profile will result in a different weight and thus different costs.

10-2-2 Installation

The sheet piles are vibrated into the ground. This is done by a special machine and the costs for installation can be expressed in costs per square meter of driven pile. The square meters are based on the area where the sheet pile should be installed and not on the surface area of the sheet pile wall. This thus depends on the length of the chamber and the length of the sheet pile. The profile of the sheet pile does not influence the price. The sheet piles can be installed by a water base installation and a land based installation. For this tool only land based solutions are investigated.

10-2-3 Recycling

For temporary building pits sheet piles can be re-used when the project is finished. The sheet piles can be pulled back from the ground. The same costs as installation of the sheet piles will be used for the extraction of the sheet piles.

Since the sheet piles can be used again at other projects the cost will reduce. Assumed is that 60% to 70% of the material price will return due to the recycling. An other option is to use recycled sheet piles at forehand. This lead to a lower purchase price per tonne. Moreover, for temporary sheet pile constructions the costs consist of installation costs, extraction costs and for the material 30% to 40% of the costs should be taken into account. In the model the material costs will be 35% of the normal material costs.

10-2-4 Conservation

To ensure durability of the sheet piles conservation measures could be taken. The first is to apply extra thickness of the sheet piles. This has influence on the weight and thus the costs of the sheet pile. The extra weight can be calculated by multiplying the surface area of the sheet pile wall (the coating area) times the extra thickness applied. The extra weight should be added at the material costs. The second option is to apply conservation methods on the places of the sheet pile where it is needed. This can be done by coating or by cathodic protection. Both apply to the surface area (coating area) of the part which needs protection.

10-2-5 Mobilization

For installation of the sheet piles special equipment is needed and preparation works have to be done. When the project is finished the used equipment has to be dismantled and moved away. To take this into account mobilization costs are introduced.

10-3 Diaphragm Wall

The price of a diaphragm wall highly depends on the thickness of the wall, the depth and the amount of reinforcement in the wall.

10-3-1 Materials

The main material of a diaphragm wall is C25/30 concrete. The amount of concrete needed is the thickness of the wall multiplied by the total length of the diaphragm wall. The steel reinforcement cages within the concrete can be expressed in *kg* per *m* and depends on the reinforcement cages which will be used.

Special attention has to be paid to the bentonite support fluid. To ensure stability during construction the trench is filled with bentonite. This bentonite can be re-used for the next trench. The bentonite can be used three times in total. This means the volume of needed bentonite is 3 times less the amount of needed concrete.

Further more, attention should be paid to the joints between the panels.

10-3-2 Installation

The installation of a diaphragm wall consist of the following: Rent, placement and adjustment of guidance structures and placement of the joints per m . Excavation of the soil, filling with bentonite and casting of concrete per m^3 . the placement of the reinforcement cages per tonne. And extra costs as small equipment, water electricity.

10-3-3 Mobilization

For the bentonite support fluid extra heavy installations need to be present on the location. Next to the heavy excavators this leads to high mobilization costs compared to a sheet pile wall. The costs will depend on the time needed for this process.

10-4 Concrete

The costs of concrete will be expressed in cubic meters of concrete which are needed for the whole project. This includes supply, processing and casting of the concrete. The costs are based on normal strength concrete, C35/45 will be used in this tool as standard concrete. Other types of concrete will not influence the costs.

10-4-1 Reinforcement Steel

The amount of reinforcement which is used in the walls, floor or capping beam is expressed in costs depending on the weight which is used. The costs includes supply, processing and placing of the reinforcement steel.

10-4-2 Formwork

The costs for the formwork are calculated by the square meter of formwork which are needed to cast the concrete. The costs include supply, placing and removal of the formwork. A distinction is made between normal formwork and heavy duty formwork. Heavy duty formwork is used when high elements are casted and the formwork is subject to heavy forces.

10-5 Soil Handling

For soil handling a clear distinction is made between the digging works, the supply of soil and the exportation of soil.

10-5-1 Digging Work

This includes the actual excavation and replacement of the soil. In this thesis it is assumed that for excavation the soil can be stored on site. For replacement the needed soil is present on the project location. The work is expressed in cubic meters of soil which have to be moved.

10-5-2 Soil Import

For the heightening of the ground and for back fill purposes high quality sand is needed. When available, excavated sand from the chamber can be used as back fill sand. When the excavated sand is not sufficient for the back fill demand or no sand is available at all soil has to be imported. The prices of the imported soil strongly depends on the availability of soil near the project location.

10-5-3 Soil Removal

When the excavated soil can not be re-used within the project it needs to be removed from the area. The costs of this work heavily depends on the type of soil and the demand for that soil in the region. Peat is can not be used for constructional purposes and might be expensive to export. Sand on the other hand can be used for construction purposes and often you get money in advance for exporting. When the soil is contaminated the soil has to be cleaned or dumped at special locations, this soil remediation makes it very expensive. This leads to very project related post which will have influence on the alternatives to be chosen.

10-6 Anchors

The costs of grout anchors depend on the diameter of the the rod, the length of the rod, the volume of the grout, the way of installing the anchor and the extra measures on the retaining wall. Because of the many possibilities and different conditions RHDHV uses for preliminary design a standard price per anchor. As described in Section 8-3-1 the calculation is made with one type of anchor. To make a good estimate the resisting force of the anchor is divided by the force per meter on an anchor as calculated by D-Sheet.

10-7 Greenfield, Brownfield and Blackfield

The location of the project can be of high importance in the design of a project. A rough distinction can be made between greenfield, brownfield and blackfield locations. A greenfield location can be described as an area where is very little activity and nothing has yet been build. For locks an example is at a location where a new channel is dug though a grassland where a lock should be present. At a brownfield location there is already some activity and some minor buildings can be present. This will already lead to more attention in design and will maybe effect the building progress. An example of brownfield is when an extra lock should be build in an existing lock area. The hardest location for a building project is at a blackfield location. These area are often highly populated which lead to little space for building and harsh logistics. Often the soil at that locations are polluted which mean that soil remediation should take place. The type of location where the project is executed often influence the costs of the building processes and transport. For each project it should be checked in which extend the the unit prices should be adapted.

Table 10-1: Unit Prices based on RHDHV Values

Discription	Unit	Unit Price
Soil Handling		
Excavation soil	m3	€ x
Replacement soil	m3	€ x
Sand import	m3	€ x
Removal sand	m3	€ x
Removal clay	m3	€ x
Removal peat	m3	€ x
Sheet Piles		
Material	tonne	€ x
Installation, from land	m2	€ x
Installation, from water	m2	€ x
Conservation	m2	€ x
Sheet pile equipment (Mob+Demob)	-	€ x
Diaphragm Walls		
<i>Material</i>		
Concrete C25/30	m3	€ x
Reinforcement cages	tonne	€ x
PVC joints	m1	€ x
Bentonite (3x re use)	m3	€ x
<i>Production</i>		
Guidance structure (placement, translocate, rent)	m1	€ x
Excavation panels (excl. soil removal)	hour	€ x
Placement joints	m1	€ x
Hoist in reinforcement cages	ton	€ x
Casting concrete	m3	€ x
Apply bentonite	m3	€ x
<i>Additional costs</i>		
Diaphragm wall equipment (Mob+Demob)	hour	€ x
Removal excess height	m2	€ x
Electricity and water use	m2	€ x
Auxiliary equipment + small tools	m2	€ x
Drainage installations	m1	€ x
<i>Finish</i>		
Inject wall leaks	m2	€ x
Finishing visable side (outside tolerances)	m2	€ x
Under Water Concrete		
Material and installation	m3	€ x
Concrete		
Material and casting concrete	m3	€ x
Material and placing reinforcement	kg	€ x
Material and placement formwork, normal	m2	€ x
Material and placement formwork, heavy	m2	€ x
Anchors		
Anchors (incl. material and installation)	-	€ x

Chapter 11

Environmental Impact

To make a quantification for environmental impact Rijkswaterstaat introduced the MKI-Value (Milieu Kosten Indicator). This value gives a costs estimation of the environmental impact of the structure based on a life cycle analysis (Figure 11-1). For each phase of the lifetime, production, construction, operational and demolition an estimation has to be made for the impact on the environment. For this tool the operational phase is not taken into account since no high impacts occur during this phase. Energy and water use of the chamber during usage are negligible. Due to the long lifetime of the lock and the long lifetime of the materials used, the impact of maintenance is also negligible. This is especially the case for lock chambers, be aware that this assumption does automatically not apply to the rest of the lock.

For each step in a phase it is analyzed which processes take place and what kind of materials and equipment are needed in that step and how much. Every process and material has a different impact on the environment and on different categories of the environment. In Table 11-2 an overview is given of the categories which are taken into account for the MKI-Value per material or process. For each category and material or process a price per equivalent unit is given for the summation of the phases from Figure 11-1. Multiplication of the amount of hazardous emissions of a process times the weighting factors gives the MKI-Value for that category. A summation of these values give the total MKI-Value for the process.

11-1 Database

To calculate the MKI-Value of a structure Rijkswaterstaat uses the program DuboCalc. This program includes a database of multiple materials, production processes and construction processes. This database will be used for the MKI-Values for the tool. In Section 11-3 a explanation of each material and process is given of what is included in the MKI-Value. However, the construction of diaphragm walls is not included in the DuboCalc database. The MKI-Value for diaphragm walls will be calculated according to the Life Cycle Analysis with

data of the materials and processes of Maas et al. (2011). The Weighting Factors of Stichting Bouwkwaliiteit (2014) are used since these are the most up to date values.

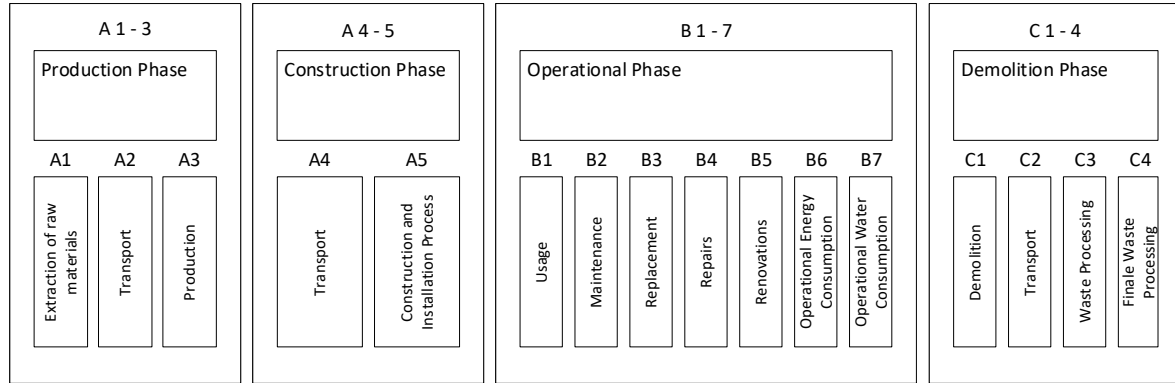


Figure 11-1: Life Cycle Analysis

Table 11-1: Environmental Costs Indicators (Stichting Bouwkwaliiteit, 2014)

Environmental Impact Category	Equivalent Unit	Weighting Factor (€/kg equivalent)
Climate Change (GWP)	CO ₂ eq	€ 0,05
Ozone Layer Depletion	CFK-11 eq	€ 30,00
Photo Chemical Oxidation	C ₂ H ₂ eq	€ 2,00
Acidification	SO ₂ eq	€ 4,00
Eutrophication	PO ₄ eq	€ 9,00
Human Toxicity	1,4-DCB eq	€ 0,09
Fresh Water Aquatic Toxicity	1,4-DCB eq	€ 0,03
Marine Aquatic Toxicity	1,4-DCB eq	€ 0,0001
Terrestrial Toxicity	1,4-DCB eq	€ 0,06
Exhaust Fossil Energy Carriers	Sb eq	€ 0,16
Exhaust Abiotic Raw Materials (Excluding Fossil Energy Carriers)	Sb eq	€ 0,16

11-2 Global Warming

Since the start of the industrial era the temperature on earth is increasing at a higher rate than happened in the past (Figure 11-2). One of the reasons this happens is the global increase of CO₂ emissions Figure 11-3. Large part of these CO₂ emissions originate from the building and construction sector. The construction industry has 6% share of the global final energy consumption and a 11% share of the global energy-related CO₂ emissions (Agency and UN Environment, 2018).

In 2015 the Paris Agreement of the United Nations Convention on Climate Change (UN-FCCC) was signed by 195 countries world wide. The goal is to limit the increase in global temperature below to 2 °C above the average pre-industrial era temperature. One of the

goals is to reduce the CO₂ emissions. The Dutch government has the goal to reduce the CO₂ emissions with 95% in 2050 compared to 1990.

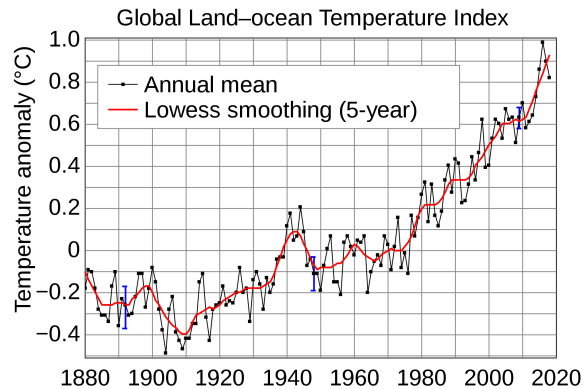


Figure 11-2: Global Land-ocean Temperature Index (Berkeley Earth, 2018)

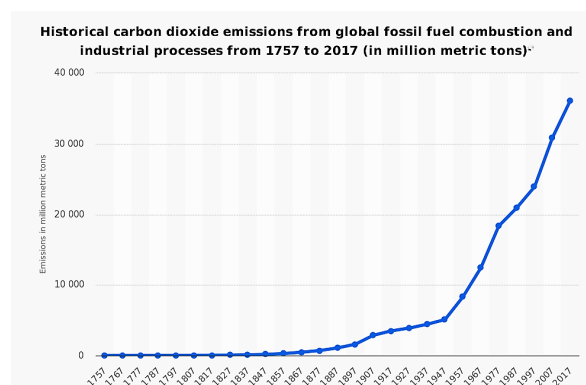


Figure 11-3: Global CO₂ Emissions (Le Quéré et al., 2018)

11-2-1 CO₂ Tax

One of options to reduce CO₂ emissions is by introducing a CO₂ tax. In this way the polluter has to pay a certain amount of money per ton of CO₂ equivalent. The goal is to stimulate people and industries to come with alternatives or innovations to reduce their CO₂ emissions. Moore and Diaz (2015) derived a value of \$220 per ton CO₂ by looking at the economic damages caused by the CO₂ emissions. While Bell and Callan (2011) concluded that the value for the United Kingdom is \$83 per ton CO₂. Stiglitz et al. (2017) concluded that to meet the Paris temperature target a tax of \$40-80 per ton CO₂ is needed by 2020 and \$40-80 per ton CO₂ by 2030 is needed world wide. In the Netherlands Klaver and van der Lee (2018) did a proposal of a €50 per ton CO₂ tax starting in 2021 increasing gradually to €200 per ton CO₂ in 2050.

Table 11-2: Environmental Costs Indicators (Stichting Bouwkwaliiteit, 2014)

Material/Process	Unit	Source	Climate Change (GWP)										Total MKI (per unit)		
			Equivalent Unit Weighting	CO ₂	CFK- 11	C ₂ H ₆	SO ₂	PO ₄	1,4- DCB	1,4- DCB	1,4- DCB	1,4- DCB		Sb	Sb
Concrete C20/25 (CEMIII)	(m ³)	DuboCalc	Factor	196	0	0,105	0,943	0,164	55,2	1,33	9300	0,333	1,00	0	€21,39
Concrete C30/37 (CEMIII)	(m ³)	DuboCalc		197,4	0	0,105	0,948	0,165	55,2	1,33	9400	0,333	1,06	0	€21,48
Concrete C35/45 (CEMIII)	(m ³)	DuboCalc		202	0	0,105	0,965	0,168	56,0	1,33	9500	0,333	1,06	0	€21,91
Concrete C45/55 (CEMIII)	(m ³)			236	0	0,115	1,09	0,186	57,9	1,50	9550	0,333	1,06	0	€24,48
Concrete C55/67 (CEMII-CEMIII)	(m ³)	DuboCalc		271	0	0,125	1,21	0,203	59,8	1,66	9600	0,333	1,06	0	€27,05
UWC 20/25	(m ³)	DuboCalc		183	0	0,100	0,858	0,151	47,1	1,33	12000	0,167	1,00	0	€19,47
UWC 25/30	(m ³)	DuboCalc		210	0	0,125	1,04	0,191	63,4	1,50	10350	0,333	1,16	0	€23,64
UWC 30/35	(m ³)	DuboCalc		237	0	0,150	1,23	0,231	79,7	1,66	8700	0,500	1,31	0	€27,81
Formwork	(m ²)	DuboCalc		53,2	0	0,025	0,320	0,0144	9,33	0,66	2500	0	0,313	0	€4,32
Steel Sheet Piles	(Tonne)	DuboCalc		907	0	0,340	4,18	0,564	132	4,33	18700	0,333	0,850	0,313	€82,53
Temporary Steel Piles	(Tonne)	DuboCalc		233	0	0,135	0,888	0,209	104	4,00	-	0,167	-2,38	0,313	€25,31
Concrete Foundation Pile	(m ³)	DuboCalc		231	0	0,0850	1,12	0,188	53,1	1,66	9600	0,333	0,875	0,250	€23,84
Soil Removal	(m ³)	DuboCalc		12,4	0	0,0100	0,0825	0,189	4,11	0	700	0	0,125	0	€1,60
Import Backfill Sand	(m ³)	DuboCalc		17,0	0	0,0100	0,0875	4,89	4,11	0	1000	0	0,125	0	€1,96
Re use Sand	(m ³)	DuboCalc		18,8	0	0,0150	0,105	0,0244	6,33	0	1000	0	0,125	0	€2,30
Reinforcement Steel	(Tonne)	DuboCalc		868	0	0,375	4,27	0,590	141	3,00	15800	0,333	5,31	0	€106,24
Bentonite	(m ³)	(Maas, 2011)		51,6	6,74E-07	0,0148	0,201	0,0285	5,11	0,0234	321	0,632	26,1	2,57	€9,33

11-3 Materials and Processes

In Table 11-2 the total cost indicators of the materials and processes are given. For comparison with different databases and other research it is important to know the origin of these values. It is important to know which processes and materials are included in the values. For the transport distances average values are taken for the specific transport distance in the Netherlands (Stichting Bouwkwali, 2014). An example of the elaboration of the MKI content of a process is given in Table 11-3. Here the materials and processes which are needed for the concrete life cycle are shown as given by the DuboCalc Database.

Concretes: The complete production in the concrete plant is included with a 20 km transport to the building site with a concrete-mixer lorry. Casting the concrete and compacting is included in the values. The demolishing will be done with an excavator and is with the recycling factor of the material part of the end of life phase. These processes hold for all types of concrete, mainly the material emissions differ.

Under Water Concrete: Like the normal concrete the UWC includes the production and the transport of 20 km. The casting of the concrete will happen from a setup on the land, not from the water. For the demolition it is assumed that it will be executed under controlled circumstances and the values are taken as if it is a floor foundation.

Formwork: Traditional formwork is used to find the MKI values. It includes the material, transport over 25 km and the installation with a crane without small equipment. The demolition will be done with man power while the panels will be removed with a crane. 90% of the wood can be burned and 10% will be deposited.

Steel Sheet Piles: The production of steel with soil, infra and hydraulic engineering quality with a transport distance of 50 km to the building site. The installation is based on the hammering of an AZ36 sheet pile with a length between 10 m and 15 m. For the demolition only the extraction with vibration is taken into account. The sheet piles will be recycled for 95%, the rest will be deposited.

Temporary Steel Sheet Piles: The process is the same as with normal sheet piles. The only difference is that the material and the production of the material is not included for temporary sheet piles. This is because it is assumed that the sheet piles will be re-used 'infinitely'.

Concrete Foundation Piles: The foundation piles are made from concrete and reinforcement steel. The production of these materials are included in the MKI values with a transport of 30 km to the construction site. The installation is based on the vibrating a 400x400 mm pile of 15 m into the ground. The removal of the pile is done in the same way as the installation.

Soil Removal: The soil will be handled in a depot with a height of 6 m to 10 m and a width of 20 m to 50 m. This will be done with an excavator with a load capacity of 1000 liters. The transport will be accomplished by ship and the distance is 75 km. The soil will be recycled for 99%.

Import Backfill Sand: Construction sand has a MKI value for the material. Further more the sand in the depot will be handled with a 2000 liter wheel loader. The transport by ship is 75 km. 99% of the sand can be recycled.

Re-use Sand: Here the MKI for the material is not taken into account since the material which is excavated will be used again. The process starts with excavation of the soil and

transport of 2 km to a depot for temporary storage. When needed the soil will be excavated from the depot, transported back to the construction site (25 km) and placed where needed.

Reinforcement Steel: The environmental impact of the reinforcement steel is mainly caused by the production of the steel. The distance to to construction area is taken as 50 km. The handling of the reinforcement steel at the construction pit is mainly done by man power with help of a crane. The demolition of the reinforcemnt steel is also included in the MKI value. The reinforcement steel is inside the concrete. To make an estimation, the example of the demolishing of a reinforced concrete viaduct is used. 95% of the reinforcement steel can be recycled.

Bentonite: The MKI value for the use of bentonite is a combination between information of Maas et al. (2011) and Stichting Bouwkwaliiteit (2014).It includes the production of the bentonite (Maas et al., 2011), the excavation of the diaphragm shafts (Maas et al., 2011), the transport of the excavated soil by ship over a distance of 75 km (Stichting Bouwkwaliiteit, 2014) and the actual use of the bentonite (Maas et al., 2011). Like with the values from Stichting Bouwkwaliiteit (2014), Maas et al. (2011) uses a full life time analysis.

Table 11-3: MKI Contribution for the use of Concrete (Rijkswaterstaat, 2018)

Type	Name	Amount	Unit	Phase	MKI	Contribution %	MKI Construction	MKI Usage	MKI Maintenance	MKI Demolition
Material	Concrete C35/45 (CEM III)	2.44	tonne	Construction	13.42	64.78	13.42	0	0	0
Process	Hydraulic demolition hammer 600-1900 kg	0.04	h	Demolition	1.16	5.59	0	0	0	1.16
Process	Compacting concrete (internal vibration)	0.4	h	Construction	0	0.02	0	0	0	0
Process	Hydraulic excavator (average)	0.04	h	Construction	0.37	1.78	0.37	0	0	0
Process	Concrete pump including vehicle	9.52E-3	h	Construction	0	0.02	0	0	0	0
Process	Hydraulic excavator (average)	0.06	h	Demolition	0.55	2.67	0	0	0	0.55
Process	Hydraulic excavator (average)	0.4	h	Demolition	3.69	17.82	0	0	0	3.69
Process	Transport bulk (road)	2.44	tonne km	Construction	1.52	7.32	1.52	0	0	0

Chapter 12

Case Study

The second Juliana lock will be used as reference project. When referred to this lock the final design of Volker Infradesign (2011) is used. The case consist of a second navigation lock near Gouda. A new extra lock was needed next to the current lock to ensure sufficient capacity in the future. Inland waterway vessels are increasing in size and more recreational vessels are predicted to use the lock. The lock will be constructed for a life time of 100 years. This case study will be used in Chapter 13 for validation of the design tool and in Chapter 14 as basic case for the results.

12-1 Water levels

The Gouwekanaal has steady water levels varying from $-0,40\text{ m NAP}$ to $-0,80\text{ m NAP}$. The Hollandse IJssel, being a river, has a wider span of water levels. With low discharges the water level of the Hollandse IJssel can drop to $-1,00\text{ m NAP}$ while with high discharges the design water level is $+3,36\text{ m NAP}$. To deal with climate change and other uncertainties in the future the design water level of the Hollandse IJssel is set on $+4,50\text{ m NAP}$.

These high water levels of the Hollandse IJssel only occur in extreme circumstances which hardly occur. When they occur the river may be closed for shipping due to high flow velocities. Because of the low frequency of occurring and the possible closure of the river during high water different water levels will be used as locking levels to limit costs. The minimum locking level is chosen as $-1,00\text{ m NAP}$ which is the lowest Hollandse IJssel Water level but also close to the water levels of the Gouwekanaal. The maximum locking level is chosen as $+2,00\text{ m NAP}$ to ensure an acceptable amount of downtime. These locking levels lead to a maximum water head of $2,8\text{ m}$.

12-2 Vessels

The new lock will be constructed to allow vessels of class CEMT IV-M7 to transfer from the Hollandsche IJssel to the Gouwekanaal. The ships has a width of $9,5\text{ m}$, a loaded draught of

3,0 m and an extended length of 105 m. Apart from the inland navigation vessels, recreational vessels will make use of the lock.

12-3 Dimensions

The length of the lock chamber is chosen as 115,5 m. Which is comparable with the minimum dimensions for inland navigation locks for extended CEMT IV vessels given by Rijkswaterstaat (2017) shown in Table 6-1.

According to Rijkswaterstaat (2017) the width of the lock should have a minimum of 10,5 m. However, due to the large amount of recreation vessels which need to be leveled the designers decided to create a lock chamber width of 14,2 m.

The top level of the UWC floor is designed at $-4,5$ m NAP. This will lead to a minimum water depth of 3,5 m during lowest lock level and a gross keel clearance of only 0,5 m for CEMT IV vessels. For CEMT IV vessels a gross keel clearance of 0,7 m is preferred (Rijkswaterstaat, 2017) for the minimum channel level of $-0,80$ m NAP at the Gouwekanaal this value will be achieved under most circumstances. The capstone level is designed at $+2,80$ m NAP. Leading to a capstone height of 0,8 m during high water.

12-4 Ground Profile

The ground profile at the location consist of first a layer of lose sand, followed by a weak layer of peat and a layer of clay, below the structure a Pleistocene sand layer is situated.

Table 12-1: Ground Profile Juliana Lock Gouda (Volker Infradesign, 2011)

Soil Type	Top of Layer (+ NAP)	γ_{dry} (kN/m^3)	γ_{wet} (kN/m^3)
Lose Sand	+4,0 m	16,0	18,0
Peat	-3,6 m	-	10,5
Clay	-7,8 m	-	15,0
Pleistocene	-12,8 m	18,0	20,0

Chapter 13

Validation

To check whether the results obtained from the design tool are correct the model needs validation. Validation is a research which is done to check the correctness of the Design-tool. Does the Design-tool gives the answers which are expected? To check the correctness of the calculations, the answers of the intermediate calculations will be compared with known and validated answers from literature, hand calculations or a case. The second Juliana lock will be used as reference project. When referred to this lock the final design of Volker Infradesign (2011) is used. The case is discussed in Chapter 12. The validation of the tool will be done in line with the model description.

13-1 General Dimensions

In this part of the tool the vertical and horizontal dimensions of the lock are calculated.

13-1-1 Ship Dimensions

The dimensions of the different CEMT-classes are present in the script. Based on the chosen ships from the input file the dimensions of the normative ships are chosen. The largest values are set as normative width, height, draught and bollard loads of the ship. The dimensions and bollard force of each ship in the script are checked manually. To check whether the right dimensions are chosen as normative dimensions random ship inputs were send to the script. Manually the normative dimensions where checked with the calculated normative dimensions, these were the same.

A random check is made for a lock with the passing ships of Motor Vessel Class CEMT Va, Pushed Convoy Class CEMT II and Coupled Formation Class VIa. When looking at Appendix A and Table 2-11 the representative length width draught and bollard forces are:

Motor Vessel Class: 135 m, 11,4 m, 4 m and 300 kN

Pushed Convoy Class: 70 m, 6,6 m, 2,6 m and 100 kN

Coupled Formation Class: 110 m, 22,8 m, 4 m and 300 kN

This will lead to a normative Length of 135 m, a normative width of 22,8 m, a normative draught of 4 m and a bollard force of 300 kN.

The model gives the output as given in Figure 13-1. It can be seen that this is the same as the manual calculated design.

```
C:\Users\908729\Documents\Pycharm_tool\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool/II_Lock_dimensions.py
Validation Ship Dimensions
Normative length ship: 135
Normative width ship: 22.8
Normative draught ship: 4
Normative bollard force ship: 300
Process finished with exit code 0
```

Figure 13-1: Validation Ship Dimensions

13-1-2 Lock Dimensions

First is checked if the manually inputted dimensions are chosen when the "Manual dimensions input" option is chosen as "Yes", this fulfills.

The height of the lock will be checked for the Juliana Lock. When designing this manually according to Section 6-2 the height should be a summation of the draught of the ship, the gross keel clearance, the capstone height for the ship class and the water head. For the Juliana lock this will be representative 3 m, 0,7 m, 1,5 m and 3 m. Leading to a height of 8,2 m.

The model gives the output as given in Figure 13-2. It can be seen that this is the same.

For the Juliana lock 7,3 m was used as height. Probably since the ultimate locking levels does not often occur the gross keel clearance and the capstone height are reduced. This is something which can be checked for final design.

```
C:\Users\908729\Documents\Pycharm_tool\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool/II_Lock_dimensions.py
The height of the lock is 8.2 m
Process finished with exit code 0
```

Figure 13-2: Validation Lock Dimensions

13-2 General Calculations

13-2-1 Tension Piles

The part tension pile calculation will be validated by using the example of the CUR 2001-4. The results of the model will be compared with the example.

Compaction Factor Calculation

This is the first part of the calculation which is also shown in the example. In Figure 13-3 the answers of the example are shown. In Figure 13-4 the answers from the model are shown. It can be seen that the first four colloms are almost the same. A slide difference in the $q_{c,z,ontgr}$, can be seen, this is due to the fact that the actual soil parameters of the layers above the sand layer are not know. An estimation is made based on given values of effective pressures in the sand layer, this can be seen in CUR 2001-4. The $\Delta R_{e,i}$ values differ from the example. However, after checking the CUR 2001-4 example it became clear that using the formula of the CUR 2001-4 below the $e_{,i}$ values in the example are not correct.

$$\Delta R_{e,i} = 0,34 * \ln\left(\frac{q_{c,z}}{61 * (\sigma'_{v,z})^{0,71}}\right) \tag{13-1}$$

In which:

$q_{c,z}$ Cone penetration pressure [N/mm²]
 σ'_v Effective vertical soil pressure [N/mm²]

Since the values of the compaction factor (f_1) are in the same order the calculation is of sufficient quality.

Diepte ok laag [m NAP]	q_{cz} [MPa]	$q_{cz,ontgr}$ [MPa]	γ'_i [kN/m ³]	$\sigma'_{vz,i}$ tijdens saaiering [kPa]	$R_{e,i}$ [-]	$e_{o,i}$ [-]	Δe_i [-]	$e_{i,i}$ [-]	$\Delta R_{e,i}$ [-]	$f_{1,i}$ [-]
-16	15	8,1	10,0	117,6	0,72	0,41	0,0511	0,361	0,128	1,47
-17	24	13,8	10,0	127,6	0,86	0,36	0,0491	0,307	0,123	1,44
-18	24	14,6	10,0	137,6	0,84	0,36	0,0493	0,314	0,123	1,45
-19	23	14,6	10,0	147,6	0,81	0,38	0,0498	0,326	0,124	1,45
-20	22	14,4	10,0	157,6	0,78	0,39	0,0502	0,338	0,126	1,46
-21	26	17,6	10,0	167,6	0,82	0,37	0,0496	0,321	0,124	1,45
-22	23	16,0	10,0	177,6	0,77	0,39	0,0504	0,343	0,126	1,46
-23	20	14,2	10,0	187,6	0,71	0,42	0,0513	0,366	0,128	1,47
-24	26	18,9	10,0	197,6	0,78	0,39	0,0502	0,337	0,125	1,46

Figure 13-3: Validation Tension Piles f1 Example

```
C:\Users\908729\Documents\Pycharm_tool1\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool1/IV_Sheetpile_1.py
Start
III check
z      q_cz  q_czontgr  sigma_vz      R_e          f1
-16.0  15.0  8.153836984563538  117.44900000000027  -0.7068027424937658  1.4729053709317192
-17.0  24.0  13.90560930254458  127.44900000000027  -0.6459685364606819  1.4827307340064166
-18.0  24.0  14.640019207124128  137.44900000000027  -0.6538877002233234  1.48144800450016
-19.0  23.0  14.638363095036274  147.44900000000027  -0.6675348177842008  1.4792400764970237
-20.0  22.0  14.509892092042524  157.44900000000027  -0.6809780058925279  1.4770683589505076
-21.0  26.0  17.676689618928773  167.44900000000027  -0.6627664551650453  1.4800111619377487
-22.0  23.0  16.052003674295168  177.44900000000027  -0.6869510361383186  1.4761044523371245
-23.0  20.0  14.280577650454271  187.44900000000027  -0.7133359324520802  1.4718540713525972
-24.0  26.0  18.941316491853613  197.44900000000027  -0.6800440277597343  1.4772191379525137
```

Figure 13-4: Validation Tension Piles f1

Maximum Tension Force Calculation

The values of the example can be find in Figure 13-5 and the values from the model in Figure 13-6. It can be seen that the values of $q_{c,z,d,i}$ differ. This is due to the fact that the

example uses values calculated in a previous example which was simplified. The design tool is in this case more precise. Due to the difference in $q_{c,z,d,i}$ the rest of the values also differ a bit. But it can be seen that the final values of the maximum tension force of the piles are almost the same.

Diepte ok laag [m NAP]	$q_{c,z,d,i}$ [Mpa]	M_i [kN/m ²]	$\gamma'_{d,i}$ [kN/m ³]	$\sigma'_{v,d,i,0}$ [kPa]	$f_{1,i}$ [-]	T_{i-1} [kN/m ²]	ΣT_{i-1} [kN/m ²]	$f_{2,i}$ [-]	$F_{r,trek,d}$ [kN]	$F_{r,trek,mand}$ [kN]
-15	-	-	-	-	-	-	0,0	-	-	-
-16	4,6	22,52	8,2	47,7	1,47	18,2	18,2	0,81	69	186
-17	6,9	32,87	8,2	55,9	1,44	19,9	38,0	0,60	144	217
-18	6,9	32,93	8,2	64,1	1,45	15,3	53,4	0,47	203	248
-19	6,9	33,05	8,2	72,3	1,45	12,3	65,7	0,37	250	279
-20	6,9	33,16	8,2	80,5	1,46	10,4	76,2	0,31	289	310
-21	8,6	41,26	8,2	88,6	1,45	10,5	86,7	0,26	329	341
-22	8,6	41,51	8,2	96,8	1,46	9,2	95,9	0,22	364	372
-23	6,9	33,43	8,2	105,0	1,47	7,6	103,5	0,23	393	403
-24	6,9	33,15	8,2	113,2	1,46	7,6	111,1	0,23	422	434

Figure 13-5: Validation Tension Piles Maximum Tension Force Example

```
C:\Users\908729\Documents\Pycharm_tool\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool/IV_Sheetpile_1.py
Start
III check
1
2[i] q_csd[i] M_i[i] gamma_dj[i] sigma_vdj0[i] f1[i] T_di[i-1] T_dn[i-1] f2[i] F_rtrekd
-16.0 4.659335419750594 22.770448472813797 8.18181818181818 52.44545454545454 1.4729053709317192 1.8394460249533515 17.59041195017655 0.7955615670300079 73.67887215160115
-17.0 7.94662458596904 39.091922283969026 8.18181818181818 60.62727272727273 1.4827307340064166 2.1524674132928268 41.7531896988989 0.5309639116918338 166.43997697322261
-18.0 8.365725261213788 41.12091537315658 8.18181818181818 68.80609090909091 1.48144800450016 1.483948840126343 59.409680113409976 0.34790235797290403 231.0409878145482
-19.0 8.364778911448301 41.05498461628631 8.18181818181818 76.99090909090909 1.4792400764970237 1.0567635239463025 71.46881241215246 0.2515680848665513 275.32492016876034
-20.0 8.291366909738587 40.63492771738878 8.18181818181818 85.17272727272727 1.4770663589505076 0.8784905409824662 80.77621590392721 0.2144102836781214 310.0562693826361
-21.0 10.100965496530728 49.60216594693368 8.18181818181818 93.35454545454545 1.4800111619377487 0.8757222720308158 90.03231918606196 0.17476197200325247 345.18999918301644
-22.0 9.17257352816867 44.92427267055292 8.18181818181818 101.53636363636364 1.4761044523371245 0.7938157161156647 97.95461909490977 0.17704500793450298 375.00305252870237
-23.0 8.16033008597887 35.851552274879346 8.18181818181818 109.7181818181818 1.4718540713525972 0.7572230348450367 105.34605521399466 0.19105427359987243 402.9429887840487
```

Figure 13-6: Validation Tension Piles Maximum Tension Force

13-3 Retaining Wall

13-3-1 D-Sheet Calculation

The creation of the D-Sheet calculation will be checked with the Juliana Lock. The model will create an input file based on the Excel user input. At the same time a D-Sheet input file is manually created with the Juliana Lock information. The manually created input file can create a project report. From a summary of that report the maximum and minimum moments, shear forces and displacements are given in Figure 13-7.

Stage nr.	Stage	Displacement [mm]	Moment [kNm]	Shear force [kN]
1	without uwc	-147,3	-510,79	-131,09
2	maintance	-143,0	-528,94	-142,38
3	LLL inland	-139,5	-515,79	-148,49
4	LLL seagoing	-139,4	-518,22	-143,16
Max		-147,3	-528,94	-148,49

Figure 13-7: Validation Manual Input D-Sheet Calculation

The Design-tool automatically calculates the D-Sheet file and reads the output file. When the output of the Design-tool is the same as output from the manually created calculation,

which can be seen in Figure 13-7, the Design-tool works. In Figure 13-8 can be seen that the results are the same and thus the Design tool works for the D-Sheet calculation.

```
C:\Users\908729\Documents\Pycharm_tool1\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool1/IV_Sheetpile_1.py
Start
The minimum moment in stage 1 is -510.79495 kNm
The minimum moment in stage 2 is -528.94155 kNm
The minimum moment in stage 3 is -515.78959 kNm
The minimum moment in stage 4 is -518.21642 kNm
The minimum shear force in stage 1 is -131.09006 kN
The minimum shear force in stage 1 is -142.38304 kN
The minimum shear force in stage 1 is -148.48976 kN
The minimum shear force in stage 1 is -143.15601 kN
The maximum displacement in stage 1 is -147.25484 mm
The maximum displacement in stage 1 is -143.00142 mm
The maximum displacement in stage 1 is -139.53909 mm
The maximum displacement in stage 1 is -139.35254 mm
```

Figure 13-8: Validation Design-tool D-Sheet Calculation

D-Sheet Version Check

The D-Sheet calculation is designed for D-Sheet Piling Version 18.2. The input file is created in such a way that it is read by D-Sheet Piling in a correct way. This is checked in the previous validation. At the other hand is the Design-tool modeled to read the results from D-Sheet and to transfer them to usable values in Python. This is also validated. In the future it is possible that the D-Sheet gets a version update. It is not know if this will influence the input files or the readability of the output files. This should be checked when a new version is on the market. In the output file the version of D-Sheet Piling is written. When this version differ from version 18.2 the user gets a warning and should check whether the results are correct.

13-3-2 Underwater Concrete Floor

The thickness of the UWC floor is automatically calculated based on the lock dimensions and the normal force resulting from the D-Sheet Calculation. In the Juliana Lock no upward force is present so the minimum thickness of 800 mm is applied for the UWC floor. This is also the result of the Design-tool.

To check whether the Design-tool works for a case with tension piles an extra validation is needed. CEMENT (de Winter et al., 2017) published an article about the CUR 77 (Jonker and Hagenars, 2015) with calculation examples. One of the examples will be used to check whether the Design-tool calculation works. Case 3 of the examples will be used since this is for shallow building pits with a low stamp force, this is also the case for most of the locks calculated. In Figure 13-9 the dimensions input values for the case are given. According to the example the results from the CUR 77 should be as given in Figure 13-10.

rekenvoorbeelden		1	2	3	4	5	6
axiale veerstijfheid keerwand	[MN/m/m]	60	90	100	70	150	60
axiale veerstijfheid trekelement	[MN/m]	185 ¹⁾	34 ²⁾	140 ³⁾	48	39	40 voor Ø50 mm / 50 voor Ø63,5 mm
breedte bouwput	[m]	16,7	24,0	39,5	22,0	23,5	31,0
aantal velden	–	6	10	10	10	8	8
interval (L_x)	[m]	3,1	2,35	4,5	2,2	3,45	$5 \times 3,30 +$ $3,5 + 3,6 +$ $3,45$
randveld (R_x)	[m]	2,15	2,6	1,75	2,2	1,4	1,725
lange richting (L_y)	[m]	2,5	2,4	4,5	2,5	2,75	3,582
toleranties b/o	[mm]	150/150	100/150	100/150	75/150	75/150	75/150
stijghoogte water	[m]	6,8	17,6	3,2	20,45	7,3	3,45
bovenkant owb t.o.v. NAP	[m]	-6,0	-16,0	0,0	-17,25	-6,35	5,35
stempelkracht (F_{stempel})	[kN/m]	550	1500	90	1307	130	51
membraanveer (k_{membraan})	[MN/m/m]	8,8	27,8	5,0	–	34,5	45,5

Figure 13-9: Validation Design-tool Example Calculations (de Winter et al., 2017)

case 3	lang		kort			dwarskracht	verbinding zonder voorziening	pons	
Toets	A	σ_{ct}	B1	B2	B3	C1	F1	G2	
dikte h_{gem} [mm]	M [kNm/m]	[MPa]	M_{ed} [kNm/m]	σ_{ed} [MPa]	M_{ed} [kNm/m]	q_{ed} [kN/m ²]	$V_{\text{ed, min}}$ [kN/m]	F_{ed} [kN/m]	$1,25 \cdot V_{\text{ed}}$ [kN/m]
1000	23	0,20	109	0,88	45	17,7	37	52	487
capaciteit		$f_{\text{ctd,pl}} = 0,83$		$f_{\text{ctd,pl}} = 0,83$	$M_{\text{rd}} = 43$	$q_{\text{rd}} = 24,3$	$V_{\text{rd,c}} = 246$	$F_{\text{rd}} = 24$	$V_{\text{rd}} = 1873$
eis		$\sigma_{\text{ct}} \leq 1,25 \cdot f_{\text{ctd,pl}}$		$\sigma_{\text{ed}} \leq f_{\text{ctd,pl}}$	$M_{\text{ed}} \leq M_{\text{rd}}$	$q_{\text{ed}} / 1,2 \geq q_{\text{ed}}$	$V_{\text{ed}} \leq V_{\text{rd,c}}$	$F_{\text{rd}} \geq 1,0 \cdot F_{\text{ed}}$	$1,25 \cdot V_{\text{ed}} \leq V_{\text{rd}}$
UC		0,20		1,06	1,05	0,87	0,15	2,13	0,26

Figure 13-10: Validation Design-tool Output From Example

In Figure 13-11 the results from the Design-tool are given. It can be seen that check A1, B1, B3, C1 and F1 are the same. B2 and G2 differ from the example. This can be explained since these checks depend on the type of tension pile which is used, this is not given in the example and thus can not be checked. The B3 check fulfills in this example. However, it should be for further expansion of the Design-tool it should be checked how check B3 works. This is because it is not calculated what the membrane spring stiffness of the retaining wall should be. Like the example the UWC floor should have a thickness of 1000 mm to be of sufficient strength.

```

UC for check A is: 0.2
UC for check B1 is: 1.06
UC for check B2 is: 1.7
UC for check B3 is: 0.87
UC for check C1 is: 0.15
UC for check F1 is: 2.13
UC for check G2 is: 0.47
The Height of the UWC floor should be: 1000 mm

```

Figure 13-11: Validation Design-tool Output From Design-tool

Bollard Loads

For the Juliana lock a bollard force of 200 kN is asked which lead to a load of $13,3$ per meter. The model gives the output as given in Figure 13-12. It can be seen that this is the same.

```

C:\Users\908729\Documents\Pycharm_tool\venv\Scripts\python.exe C:/Users/908729/Documents/Pycharm_tool/Iia_Bollard_loads.py
The bollard load is 13.333333333333334 kN per meter
Process finished with exit code 0

```

Figure 13-12: Validation Bollard Loads

13-4 U-Basin

The code of the U-basin is checked on mistakes. The model calculates the moments, normal forces and shear forces resulting from the loads. With the this forces and moments the dimensions of the U-basin are calculated.

13-4-1 Horizontal Soil Pressure

To check whether the soil pressure is calculated correctly a example will be used as shown in Figure 13-13. The soil consist of sand with the properties as given in Table 13-2 and water has a weight of 10 kN/m^3 .

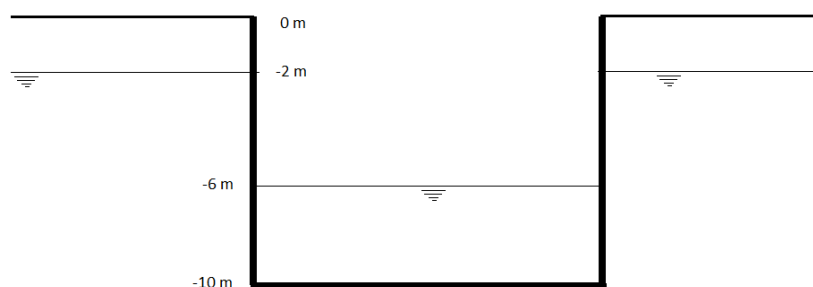


Figure 13-13: Validation Soil Pressures U-Basin

Table 13-1: Soil Properties Validation Example

Sand	
γ_{dry}	18 kN/m^3
γ_{wet}	20 kN/m^3
ϕ'	30^0
OCR	1

By hand is calculated:

Table 13-2: Soil Properties Validation Example

Level	σ_v	p	σ'_v	σ'_h
0	0	0	0	0
-2	36	0	36	18
-6	116	40	76	38
-10	196	80	116	58

```
Effective horizontal ground pressure at -2 m is 18.0 kN/m2
Effective horizontal ground pressure at -6 m is 38.0 kN/m2
Effective horizontal ground pressure at -10 m is 58.0 kN/m2
```

Figure 13-14: Validation Soil Pressures

In Figure 13-14 can be seen that the values of σ'_h are the same as the hand calculations. This will lead to a moment at the bottom of: $(18 - 0) * 0,5 * 2 * (8 + \frac{1}{3} * 2) + 18 * 8 * 4 + (58 - 18) * 0,5 * 8 * (\frac{1}{3} * 8) + 80 * 0,5 * 8 * (\frac{1}{3} * 8) - 40 * 0,5 * 4 * (\frac{1}{3} * 4) = 1905 kNm$

In Figure 13-15 the moments following from the model are shown. It can be seen that the moment resulting from the soil and thus the total moment differ a bit from the hand calculation. This can be explained by the step size which is taken as 10 cm within the model. The difference is 0,8% which is negligible.

```
Moment from soil near floor is 1142.615 kNm
Moment from water outside near floor is 853.333333333333 kNm
Moment from water inside near floor is -106.666666666666 kNm
Total moment near floor is 1889.281666666665 kNm
```

Figure 13-15: Validation Moment in Wall

13-4-2 Walls and Floor

The floor and walls of the U-basin are calculated with verified and validated VBA sheets from RHDHV. Which means that validation of these sheets and calculation is not needed. However, it should be checked if the VBA sheet is coupled correctly with the Python script.

The dimensions resulting from the model should be checked on correctness. This is done by manually using the VBA sheets to calculate the dimensions. The answers of the manually

entered calculations are checked with the automatically generated output from the model. These turned out to be the same.

13-5 Findings

The validation is done by comparing the model with a case, literature and hand calculations. The input values for the water levels in the lock chamber, the ships and the ground profiles are processed in a correct way leading to the dimensions of the lock. When comparing with the case, some differences occur due to different design choices which deviate from the standard prescriptions.

The tension piles are validated with calculations from literature. The values differ hardly and thus validate. The UWC floor is checked both on the case and literature. The cases is correct with a minimum UWC floor thickness while the literature study results in the same results with a more complicated UWC floor leading to validation.

The calculation of the strength and stability of the retaining walls is done by the external validated D-Sheet software. The input and output of the program is checked with the model and satisfies the results.

The calculation for the U-basin is checked by validating the occurring moments and forces and the outputs form the VBA sheets.

It can be concluded that the validation of the Design-tool fulfills the requirements for preliminary design.

Chapter 14

Results

The Design-tool is created to calculate lock chambers for every inland navigation lock within the Netherlands. For this thesis the second Juliana lock Gouda will be used to create results (Chapter 12). The basic calculation will be done based on the design basis from Volker Infradesign (2011). After this different adjustments will be made on this basic design. The Design-tool will calculate the case studies with the adjustments to check what will differ from the original case and to what extent. Two types of adjustments can be distinguished. The first one are the main dimensions of the lock, the lock length, lock width, the water head and the water depths. These values mainly differ for project location and types of ships. The adjustments in main dimensions are treated in Section 14-2 up to Section 14-5. The other adjustments are changes within the alternatives like type of material, changes in building processes or extra requirements for the lock design. These adjustments will be treated in Section 14-6 up to Section 14-11.

14-1 Basic Case Juliana Lock

The costs of the optimum designs of the different alternatives are given in Table 14-1. These are per type of structure the best options. For the Basic Case the CO₂ tax is given as €0,05 per kg CO₂, all other examples and cases will also use this CO₂ tax (except in Section 14-1-6 where the CO₂ tax will be elaborated).

It can be seen that the costs differ significant per alternative. Also the MKI Values differ per alternative. The MKI values are in this case close related to the Potential CO₂ tax while a clear relation between MKI Values and Total Costs are not perceptible.

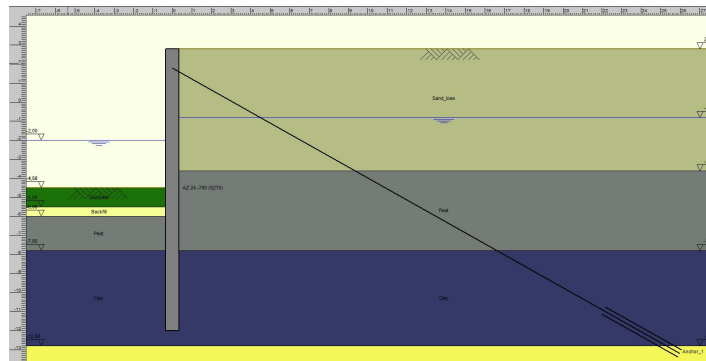
Table 14-1: Overview Results Basic Case Juliana Lock

	Total Costs	MKI Value	Potential CO2 Tax
Sheet Piles Wall Structure	€ 1.257.365	€ 140.838	€ 52.143
Diaphragm Wall Structure	€ 1.796.997	€ 211.814	€ 77.889
U-Basin Structure Open Pit	€ 1.563.937	€ 270.314	€ 93.142
U-Basin Structure Retaining Wall Pit	€ 2.276.377	€ 228.483	€ 81.976

14-1-1 Sheet Pile Wall Structure

For the Basic Case a sheet pile profile AZ24-700 is needed to withstand the forces and moments. To be stable the sheet piles should be installed to a depth of $-12\text{ m} + \text{NAP}$ having a total length of $14,8\text{ m}$. The minimum Thickness of the applied UWC floor is 800 mm . With this thickness the downward pressure from the own weight is more than the upward pressure from the water during the maintenance level in the lock. Because of this the 800 mm thickness is sufficient for the UWC floor. Since there is no net upward force, tension piles are not needed. A schematization of the structure is given in Figure 14-1. Further more, a capping beam of 820 mm by 1000 mm is needed to evenly transfer loads from the ships and anchors. The capping beam is expended one meter below LLL to ensure a smooth surface.

For the conservation of the sheet piles an extra thickness will be used. By looking at table 4-1 and 4-2 of Eurocode 5 (British Standard Institution, 2007) 6 mm is chosen for the loss of thickness, this is on the conservative side.

**Figure 14-1:** Sheet Pile Wall Structure, Basic Case

In Figure 14-2 an overview of the costs of the sheet pile wall structure is given. It can be seen that half of the costs (50%) are going to the sheet pile material. This is logical since the sheet piles form the basic of the chamber, the walls. The other large constructional part of the lock, the floor, is good for 20% of the total costs.

Costs Sheet Pile Wall Chamber

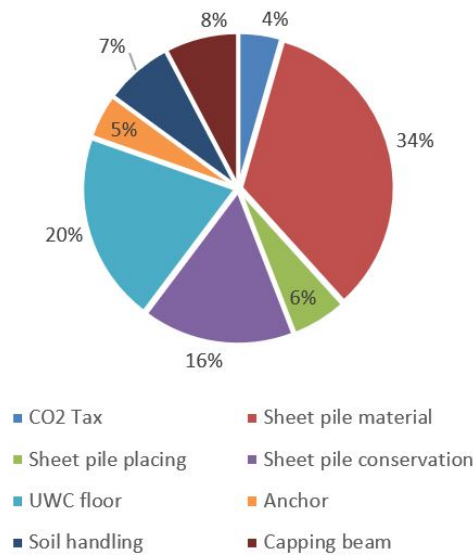


Figure 14-2: Costs Sheet Pile Wall Structure, Basic Case

14-1-2 Diaphragm Wall Structure

In the Basic Case the diaphragm wall should have a thickness of 600 mm. This is the minimum thickness possible for a diaphragm wall. In this case the diaphragm wall is oversized. The UWC floor will like with the sheet pile wall structure have an thickness of 800 mm. The schematization can be seen in Figure 14-3. The capping beam is larger compared to the sheet pile wall structure due to the larger thickness of the wall. The capping beam for the Juliana Lock case will be 1250 mm by 1000 mm.

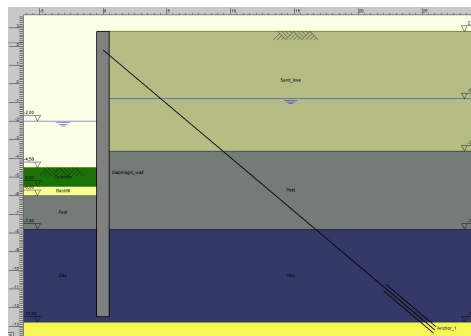


Figure 14-3: Diaphragm Wall Structure, Basic Case

In contrast to the sheet pile wall structure, the costs for the materials are less than the costs for the constructing and placing of the diaphragm wall (Figure 14-4a). This is due the extensive construction method with the excavation, separate reinforcement cages and bentonite. A distribution of the placement costs is given in Figure 14-4b while the distribution of the materials can be found in Figure 14-4c.

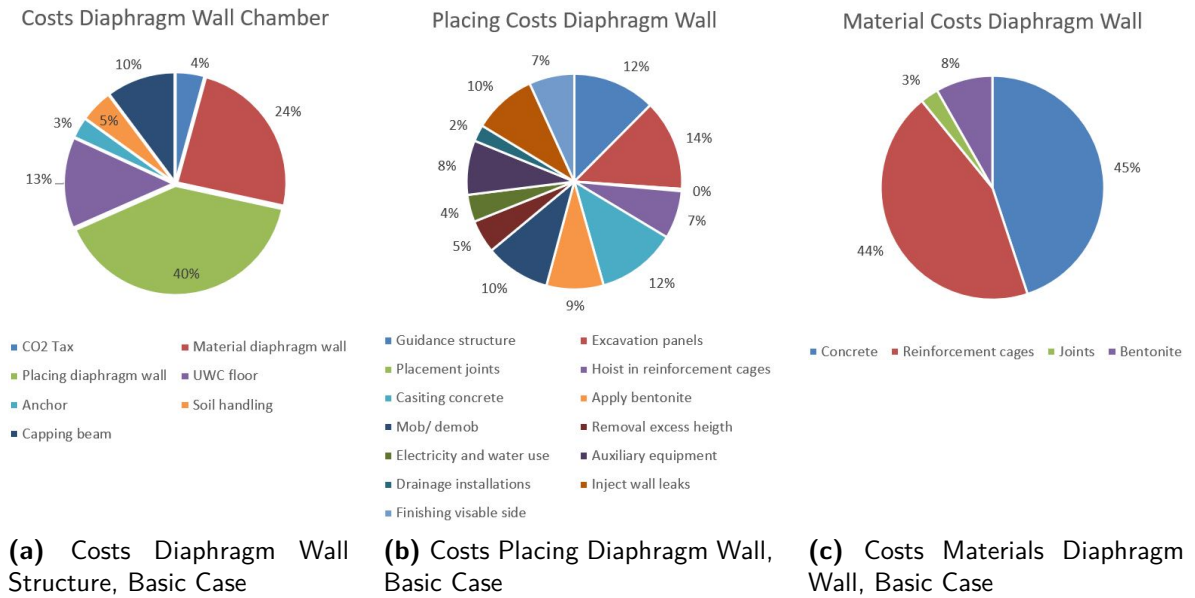


Figure 14-4: Costs Diaphragm Wall Structure

14-1-3 U-Basin Open Pit

In Figure 14-5 the dimensions of the U-basin are given which are needed for the Juliana Lock. This structure will be constructed in a dry open building pit. The concrete is reinforced to deal with moments in the structure. Shear force reinforcement is not present. Because of this, the shear force check has been found governing when calculating the dimensions.

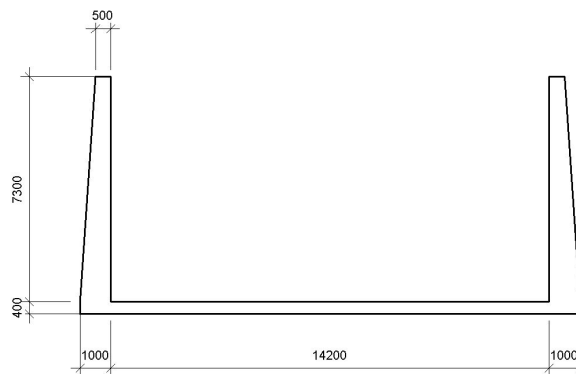


Figure 14-5: U-basin Open Pit, Basic Case

In Figure 14-6 the costs related to the structure are given. It can be seen that, due to the open building pit, the costs for soil handling are a significant part of the total costs (47%). It can be seen that the walls(32%) are more expensive than the floors (15%). This can be explained by looking at the costs accrual. The walls need to resist significantly more shear force compared to the floor. Since no shear force reinforcement is used this force should be

adopted by the concrete. For this reason the wall, especially at the bottom, is thicker than the floor and thus more concrete is needed. The moments in the floor and the wall are of the same magnitude. The difference in formwork costs can be easily explained by the amount of formwork needed. For casting the floor formwork is only needed at the bottom and at the sides. For the walls formwork is needed at both high sides leading to a larger area which needs support. Also, due to the relatively large heights of the wall, heavier formwork is needed which is more expensive.

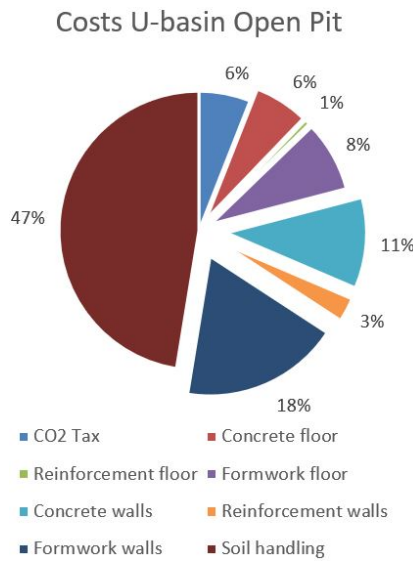


Figure 14-6: Costs U-Basin Open Pit, Basic Case

14-1-4 U-Basin Retaining Wall Pit

For the U-basin which will be build in the retaining wall building pit the same dimensions are needed as with the open pit, Figure 14-5. The different in the cost will thus resulting from the building pit. For the open pit a major part of the total costs originated with soil handling. In this case the building pit needs to be constructed as given in Figure 14-7. When the construction is finished the retaining wall will be pulled out and can be used again in an other project.

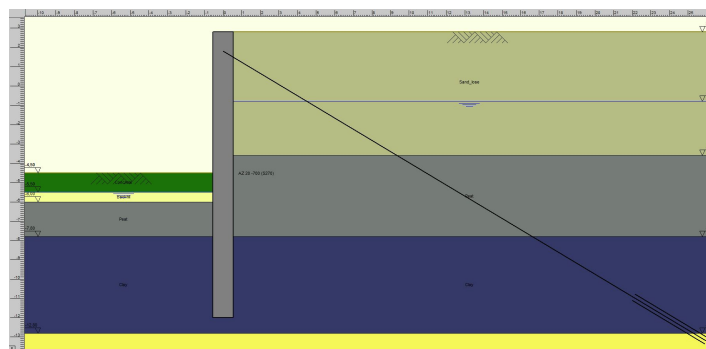


Figure 14-7: U-Basin Retaining Wall Pit, Basic Case

In Figure 14-8 can be seen that the construction of the retaining wall building pit, with the temporary retaining wall, the UWC floor and the tension piles, are responsible for more than half the total costs (58%). In Section 14-1-3 the costs for the building pit, the soil handling, was only 47% which also includes the excavation for the chamber.

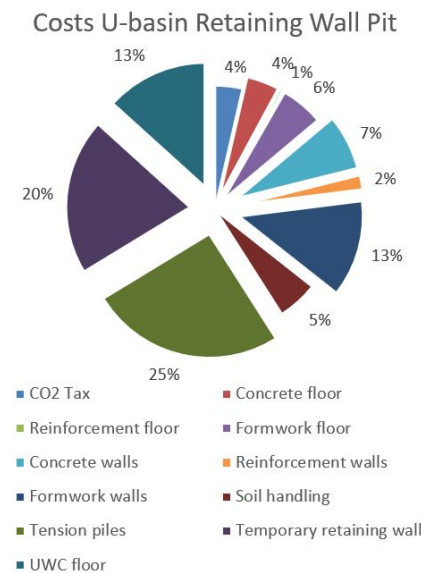


Figure 14-8: Costs U-Basin Retaining Wall Pit, Basic Case

14-1-5 MKI

In Table 14-1 the total MKI values of the structures were given. In this section these values will be further elaborated to get more insight on the environmental impact of the structures. In Figure 14-17 the MKI analysis is given for the different structures.

For the sheet pile wall structure (Figure 14-9a) most of the environmental impact is resulting from the walls (the sheet piles and the capping beam). The floor and the walls have both an impact resulting from the material production and the installation. With the soil handling, which represents 19% of the total MKI, only transport and excavation has an impact on the environment since the soils are natural products.

The diaphragm wall structure (Figure 14-9b) has a larger impact on the environment. In the basic case this can be explained by the minimum thickness of the diaphragm wall which is too heavy for this case. The bentonite which is used to support the shaft only has a limited impact.

Soil handling will play a major part when an open building pit (Figure 14-9c) is used. The large amount of soil which needs to be processed is responsible for 67% of the total environmental impact. This explains why a temporary retaining wall structure (Figure 14-9d) has a lower MKI value. The temporary sheet piles have a low MKI value since they can be reused. For the temporary sheet pile wall the MKI is €18.858. While for the definitive sheet piles the MKI is €50.102.

When looking at the chamber structure alone, the U-basin is only has an MKI of €90.462. This is less than the sheet pile wall structure (€113.821) and the diaphragm wall structure

(€1.193.497). This can be partly explained by the floor, the UWC floor has a minimum thickness of 800 mm while the floor of the U-basins is only 400 mm thick. For the wall the difference can be explained by the fact that steel has a larger environmental impact than concrete. For the diaphragm wall much more reinforcement is needed compared to the U-basin walls.

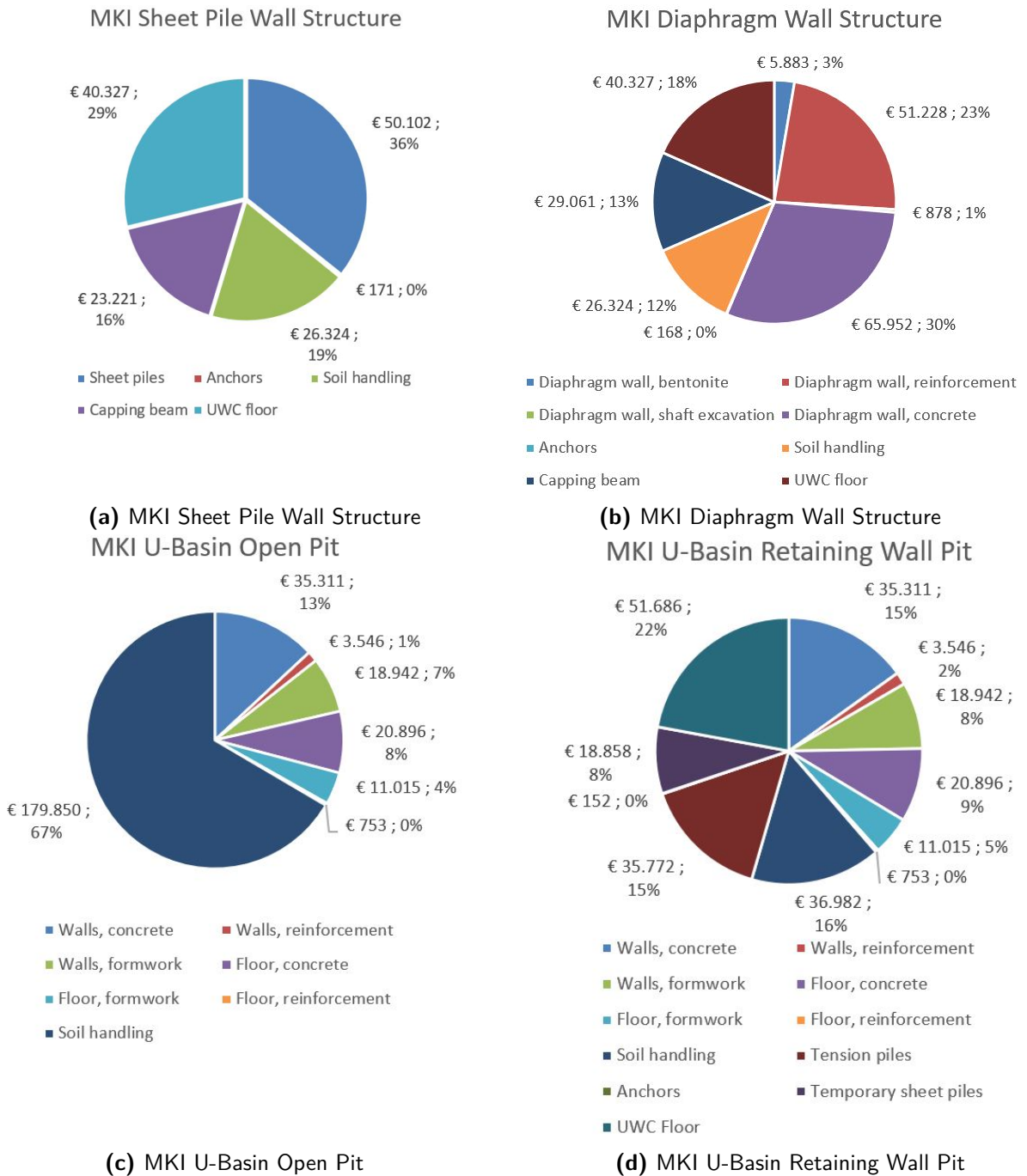


Figure 14-9: MKI Structures

14-1-6 CO2 Tax

Each material and building process has its own MKI value and also its own CO₂ equivalent emissions. The MKI value is an costs expression to give an indication for the environmental impact. The CO₂ equivalent on the other hand can lead to an actual increase in total costs when a CO₂ tax will be introduced.

Each structure has a certain CO₂ equivalent emission. Because of this, the CO₂ tax has different consequences for each alternative. At a specific value of CO₂ tax an alternative can be more attractive than another. The CO₂ tax will at least vary from €0,00 up to €0,20 per kg CO₂ equivalent. This is because at the moment no CO₂ tax is applicable while there are intentions from the environmental organizations to introduce a CO₂ tax of €0,20 per kg CO₂ equivalent. The Dutch government is now talking about a CO₂ tax of €0,05 per kg CO₂ equivalent.

In Figure 14-10 the development of the costs for a increasing CO₂ tax are given. It can be seen that the U-basin structure has a relative high CO₂ emission. This is indicated by the steep course of the costs. The lines diaphragm wall structure and the U-basin with retaining wall pit are more the less parallel which indicates a comparable CO₂ emission. The CO₂ tax has the least influence on the sheet pile wall structure. The vertical lines in Figure 14-10 show the proposed CO₂ taxes of €0,05 and €0,20 per kg CO₂ equivalent. The lines show that these CO₂ taxes do not influence the choices for the alternatives when only looking at costs. This will differ when a CO₂ tax of €0,68 will be introduced, in that case the diaphragm wall structure will be more attractive compared with the U-basin with open pit.

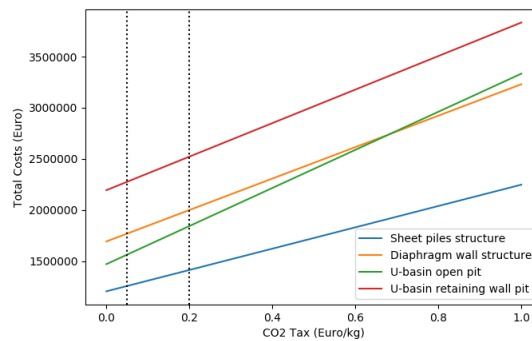


Figure 14-10: Cost Development as a Function of CO₂ Tax

14-2 Change in Width

In this section it will be checked what the influence of the width of the chamber is on the costs of the different structures. When wider ships need to pass the lock or it is chosen to level the ships next to each other the width of the lock increases. For this check the basic Juliana lock case is used as starting point, a width of 14,2 m. Since a larger width will not influence the force distribution on the walls, the profile of the wall structures will not differ from the basic case.

In Figure 14-11a the costs development is plotted as function of the width of the lock chamber. It can be seen that the cost of the sheet pile structure and the diaphragm wall structure increase almost linear. This can be explained by the fact that up to a width of $24,2\text{ m}$ the thickness of the UWC floor remains the minimum thickness of 800 mm . This lead to the case that the extra costs only resulting from the extra width meters of UWC floor and soil excavation, this is linear. The dimensions of the floor of the U-basin are also constant over certain intervals, this is due to the fact that there is almost no vertical pressure acting on the floor. The jump at $19,2\text{ m}$ is shows the increase of thickness in the U-basin floors. When the vertical pressure is of a larger magnitude, the floor thickness will increase with an increasing width caused by the increasing moment in the floor. In this case the thickness of the floor is mainly dependent on the moments resulting from the walls, which are constant for increasing width. Furthermore, the extra step rise of the U-basin with retaining wall pits is explained by the needed tension piles (Figure 14-11b).

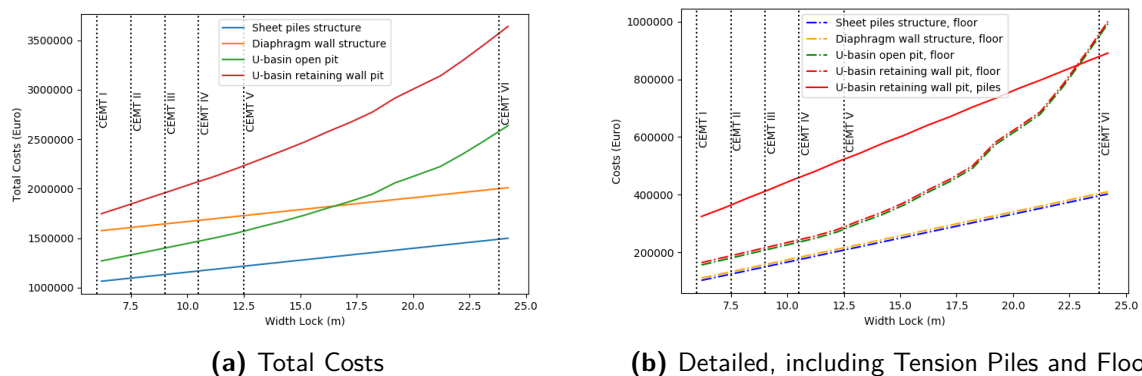


Figure 14-11: Cost Development as a Function of Width

For this case the sheet pile wall structure economically most attractive regardless the width of the chamber. What can be noticed is that as of a width of $19,2\text{ m}$ the U-basin structure with an open pit becomes more attractive compared to a diaphragm wall structure. This is due to the stronger increase in costs of the U-basin.

In Figure 14-12 can be seen that like the costs the MKI values gradually increase with the width. With the costs a tipping point can be seen with the U-basin open pit and the diaphragm wall structure. This is not the case when looking at the MKI value, the diaphragm wall structure is constantly more environmental friendly.

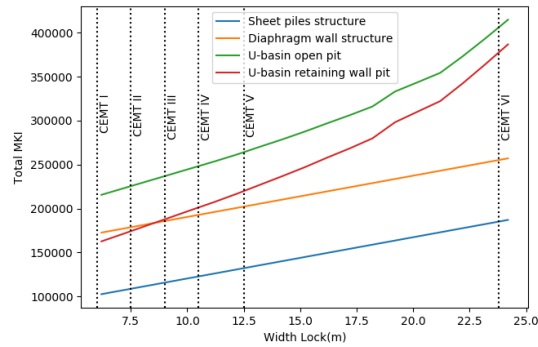


Figure 14-12: MKI Development as a Function of Width

14-3 Change in Water Head

To simulate the changes due to a change in water head the highest lock level will increase or decrease. To make sure the structure will still fulfill the standard requirements the capstone height will change with the same value. It is reasonable to assume that the ground water level will change with the HLL, for this reason the ground water level will also change equally with the water head. For the basic Juliana lock case the water head is 3 m.

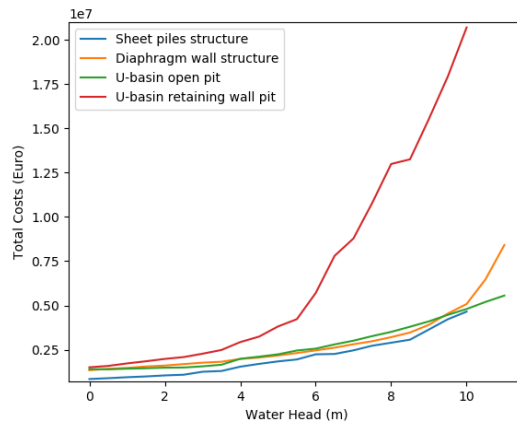


Figure 14-13: Cost Development as a Function of Water Head

In Figure 14-13 the cost development of the chamber structures is given for different water heads. It can be seen that for the initial structures (without water head) the U-basins and the diaphragm wall structure are equally expensive. The sheet pile wall structure is less expensive, this can be explained by the multiple small sheet pile profiles which can be used for the walls. This also explains angulated course of the line. A detailed overview of the costs is given in Figure 14-14. The U-basins and the diaphragm walls are limited with a minimum thickness for the walls. At a water head of 6 m the minimum thickness of the diaphragm wall not long sufficient, from there on the costs of diaphragm wall structure are increasing more rapidly.

The stop of the sheet pile wall structure and the U-basin with retaining wall pit costs at 10,5 m can be explained by the fact that the heaviest combi wall present in the model is not strong enough to resist the forces.

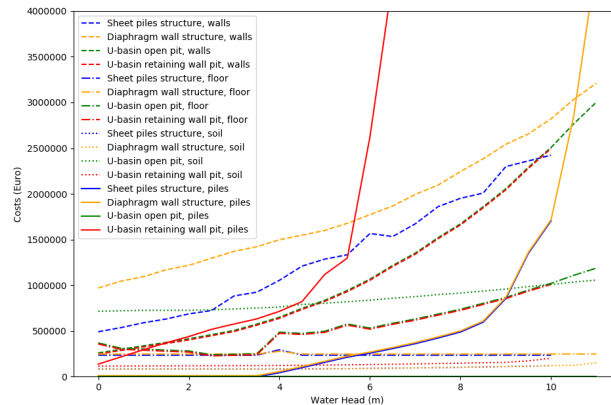


Figure 14-14: Detailed Cost Development as a Function of Water Head

The strong increase in costs of the U-basin with the retaining wall pit starting at a water head of 4,5 m can be explained by the tension piles, Figure 14-14. To ensure a dry construction area for the U-basin an UWC floor is installed. Due to the increasing pressures more and more tension piles are needed to prevent uplift. Also for the retaining wall structures piles can be needed to prevent uplift. For the sheet pile wall and retaining wall structures the first piles are needed with a water head of 3,5 m. After that the number of piles needed are gradually increasing with the increasing water head. At a certain moment the number of piles becomes that large that the mutual piles will influence another. The design tension force decreases and even more piles are needed. This phenomena can be easily seen starting at a water head of 4,5 m for the temporary retaining wall structure. This effect can be reduced by using large piles, in this way less piles are needed and the mutual distance will increase. This also explains the rapid increase at the sheet pile and diaphragm wall structures at 8,5 m. Tensions piles are needed as of 3.5 m from where the price evenly increases up to 8,5 m. This will be further explained in Section 14-7.

It can be seen that the costs for the floor of the U-basin grow when the water head increases. The UWC floor of the retaining walls stay relatively constant. At 4 m it can be seen that the UWC floor thickness increases. However, it drops again after 4 m, this can be explained by the increasing number of tension piles needed. The tension piles increase the moment capacity of the UWC floor.

14-4 Change in Water Depth

When the water depth increases the retaining height increases while the water pressures remain constant. In the Juliana lock the water depth is 3.5 m at LLL, from here on the water depth will increase. In Figure 14-15 the cost development of the structures with increasing

water depth is shown. The U-basin constructed in retaining wall pit rapidly increases after water depth of 8,0 m, this can be explained by the increasing water pressure at the construction stage when the pit is dry. From 8,0 m depth to much piles are needed as explained in Section 14-3.

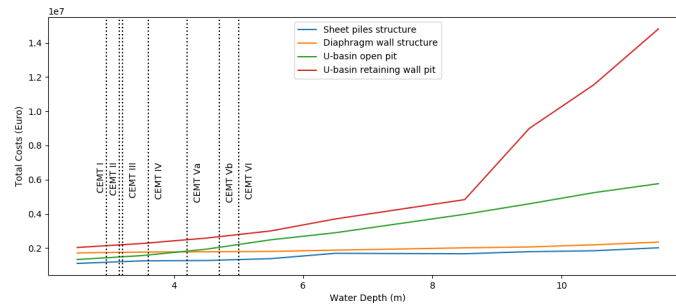


Figure 14-15: Cost Development as a Function of Water Depth

In Figure 14-16 the detailed costs for the structures are given. For the U-basin with open pit the fast increase in costs can be explained by the soil handling costs. Due to the open building pit the volume of excavated soil increases much faster over depth comparing with a retaining wall structure. Further more, the thickness of the floor of the U-basin increases while the the thickness of the UWC floors remain constant. This can be explained by the moments resulting from the wall which should be transferred to the floor in case of a U-basin.

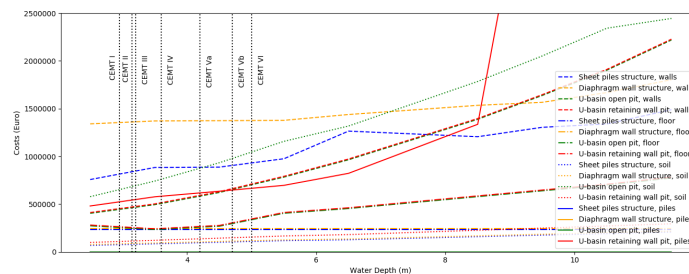


Figure 14-16: Detailed Cost Development as a Function of Water Depth

14-5 Change of Soil

The Juliana lock is constructed in a area with a layered soil profile consisting of sand, peat, clay and again sand. To check the influence of these soils, the Juliana lock will be simulated with homogeneous soil profiles of only sand, only clay and only peat. In Figure 14-17a the costs of the structures for the different soils are given.

It can be seen that sand is the best soil for construction. This is because sand is a strong material. Clay, and especially peat, are weaker materials which make construction more expansive. In Figure 14-17b can be seen that for the retaining wall structures especially the walls become more expansive in weak soils. In peat the retaining walls should be almost

3 times longer than in sand to become stable. Because of this long retaining walls the U-basin with open building pit becomes most attractive than the retaining wall structures. Also soil handling becomes significant more expensive (Figure 14-17c). Sand is a building material which can be easily reused and is thus cheap to handle. Peat and Clay are more costly to process leading the U-basin with open pit to become less attractive compared with the diaphragm wall structure in clay soils. Tension piles are needed for the U-basin with a retaining wall building pit. Tension piles are less efficient in clay and peat. Because of this more piles are needed. The extreme rise in prices can be explained by the mutual influence of the piles (Section 14-7).

It should be noted that for diaphragm walls the implementation for weak soils may differ. Also, extra vertical stabilization for soft soils is not taken into account.



Figure 14-17: Detailed Cost Development for Different Soil Profiles

14-6 Change of Surcharge Load

For the basic case of the Juliana Lock is subject to a surcharge load of 10 kN/m^2 . For locks the surcharge load is mostly taken as 20 kN/m^2 (Vrijburcht et al., 2000). In Figure 14-18 the costs of the different structures are given with a change surcharge load.

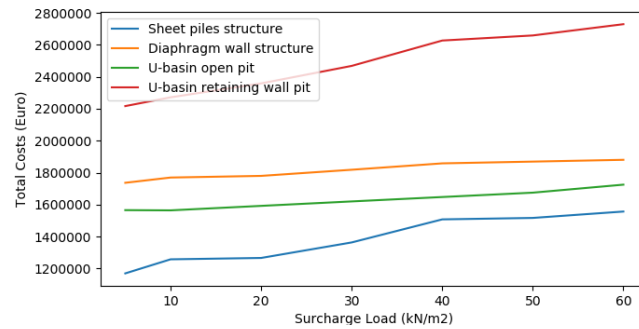


Figure 14-18: Cost Development as a Function of Surcharge Load

It can be noted that The structures with sheet pile walls and combi walls are most sensitive for the increasing surcharge loads. This is because the for the diaphragm walls the minimum thickness of 600 mm is still sufficient for the surcharge load of 50 kN/m^2 while the sheet and combi walls need to scale up for the rising moments (Figure 14-19). The floors are hardly impacted by the increasing surcharge load. Only the reinforcement of the U-basin increases slightly due the increasing moments resulting from the walls. The walls of the U-basin increase less fast due to the stiffer construction.

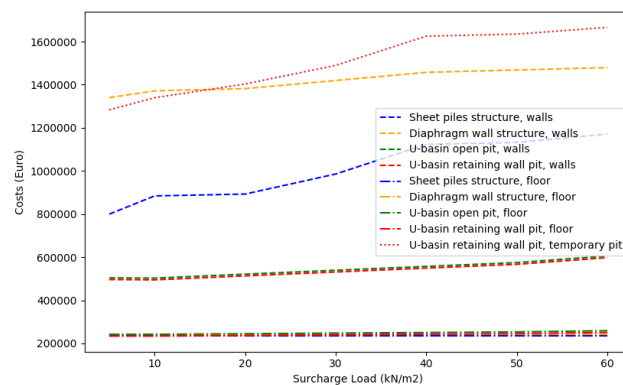


Figure 14-19: Detailed Cost Development as a Function of Surcharge Load, Including Floors and Walls

14-7 Change of Piles

As described in Section 14-3 the tension piles can have a major role in the costs of a structure. To show this the Juliana Case with a water head of 8 m will be calculated for different tension

piles, both in width and length.

14-7-1 Pile Width

Since the unit prices are known for prefabricated concrete piles with a width of 350 mm, 400 mm and 450 mm these piles will be compared. In Figure 14-20 it can be seen that when limited piles are needed (retaining wall structures) the costs for piles decrease when wider piles are used. For the temporary building pit (with many piles) this does not hold. Wider piles are relatively more expensive than smaller piles when looking at extra surface for adherence. When the little more adherence leads to sufficient less piles the centre to centre distance increases which can lead to less piles needed, this can be seen in Figure 14-20 for the retaining wall structures and the 450 mm pile of the temporary retaining wall.

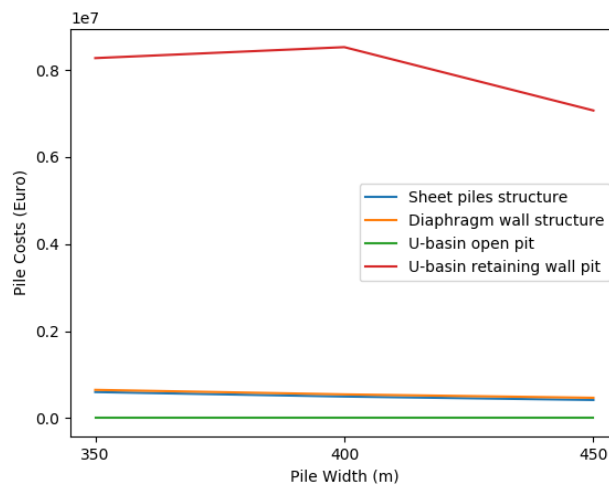


Figure 14-20: Costs for Different Pile Width

14-7-2 Pile Length

The influence of the pile length is shown in Figure 14-21a. The maximum pile length which is considered is 40 m since piles exceeding this length are not common and will differ in price. When a limited amount of piles are needed the costs drop slowly for an increasing pile length. When many piles are needed the drop is steep for larger pile lengths until the centre to centre level is sufficient and the reduction of the tensile force becomes less. In this case the tipping point is at piles between 15 m and 16 m. From there on the pile costs decrease slowly with an increase in length. When the length increases further less piles are needed which might have an influence on the UWC floor. This can be seen in Figure 14-21b with a pile length of 25 m and 35 m the floor has to become thicker. In Figure 14-21c the total costs are given for a changing pile length. It can be seen that the costs saving of the piles is canceled out by the increasing UWC floor at a pile length of first 25 m after which it decreases again due to the constant UWC and the reducing piles. At a pile length of 35 m this happens again.

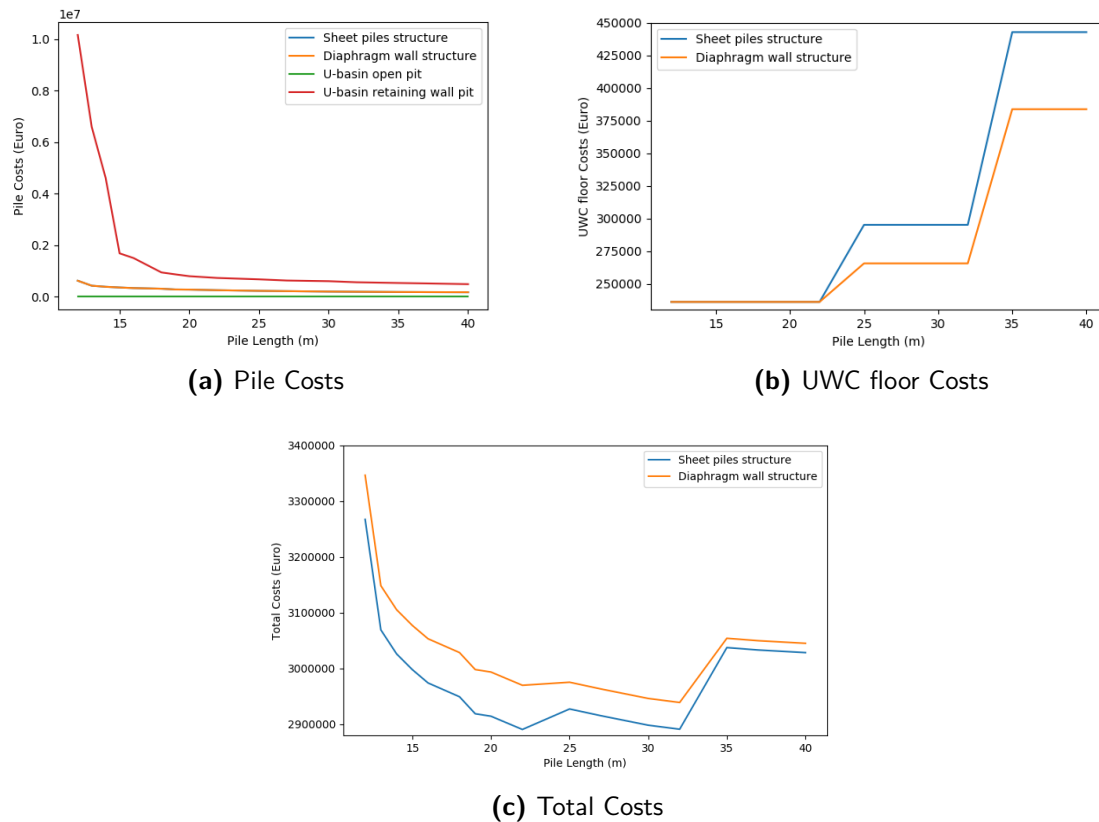


Figure 14-21: Detailed Cost Development as a Function of Pile Length

14-8 Different Under Water Concrete

For the Juliana lock, UWC is used with the strength C25/30. It is possible to use UWC with a different strength. Environmental data is also available for UWC C20/25 and C30/37. In this section will be checked how the type of UWC will influence the costs. To check this the Juliana lock case will be used with a water head of 4 m (3 m is normal). For this check the water head is increased to ensure the UWC floor exceed its medium thickness of 800 m. With this water head the UWC floor has a thickness of 1000 mm when using under water concrete class C25/30.

In Table 14-2 can be seen that the total costs for different under water concrete strengths hardly differ. When comparing the results with Table 14-4 it is noticed that the increase in total costs is completely caused by the higher CO₂ tax. The stronger the UWC the higher the CO₂ emission and the MKI value (Table 14-3). The thickness of the UWC does not differ in this case.

Table 14-2: Different Under Water Concrete, Total Costs

	C20/25	C25/30	C30/37
Sheet Piles Wall Structure	€ 1.541.799	€ 1.544.014	€ 1.546.228
Diaphragm Wall Structure	€ 1.977.878	€ 1.979.871	€ 1.981.863
U-Basin Structure Retaining Wall Pit	€ 2.718.171	€ 2.720.441	€ 2.722.712

Table 14-3: Different Under Water Concrete, MKI Value

	C20/25	C25/30	C30/37
Sheet Piles Wall Structure	€ 165.034	€ 173.931	€ 182.829
Diaphragm Wall Structure	€ 230.256	€ 238.264	€ 246.272
U-Basin Structure Retaining Wall Pit	€ 266.551	€ 275.674	€ 284.797

Table 14-4: Different Under Water Concrete, CO2 Tax (€0,05)

	C20/25	C25/30	C30/37
Sheet Piles Wall Structure	€ 62.698	€ 64.912	€ 67.126
Diaphragm Wall Structure	€ 84.403	€ 86.396	€ 88.389
U-Basin Structure Retaining Wall Pit	€ 96.524	€ 98.794	€ 101.064

14-8-1 Casting Method

Both the Hopper-Dopper Method and the Contractor and Ventiel Method are included in the model for the casting of the UWC floor. The costs of casting the concrete with the different methods are not taken into account. However, the different methods lead to different tolerances for the calculation of the UWC floor. The Hopper-Dopper Method has a tolerance of 75 mm on the top side while the Contractor and Ventiel Method is less accurate which lead to an tolerance of 150 mm.

For the case study no change in costs are obtained. The steps for the UWC floor thickness calculation are 100 mm, because of this a change in tolerance of 75 mm does not necessarily lead to alteration in floor thickness. When smaller steps for the UWC floor are used an change will be visible.

14-9 Change Concrete Strength

To check whether the strength of the concrete influence the costs multiple calculations of the Juliana lock basis case are done with different concrete strengths. In Figure 14-22 the costs development for the different types of concrete are given. It can be seen that mainly the U-basins differ with concrete strength. This is because the main structure consist of normal concrete (not UWC or diaphragm walls) for the U-basin.

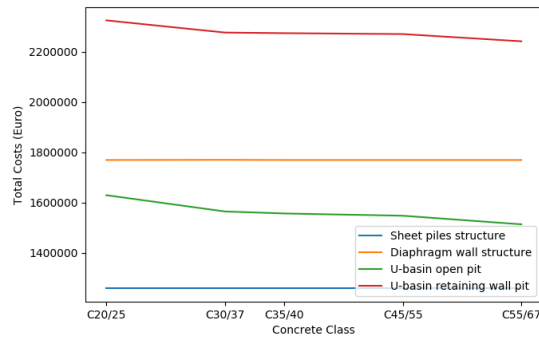


Figure 14-22: Cost Development as a Function of Concrete Strength

In principle, a higher concrete class lead to more strength in the structure. This means that the dimensions of the structure can reduce. In Figure 14-23a the dimensions of the floor and walls are given for the different concrete classes. In Figure 14-23b the costs for the different constructions are given. As expected the costs reduce when using stronger concrete. The only exception are the floors for concrete class C55/67. This is due to the fact that the floor reduces here to 0,7 m. This is sufficient for the shear force but to withstand the moment more reinforcement is needed which lead to more costs.

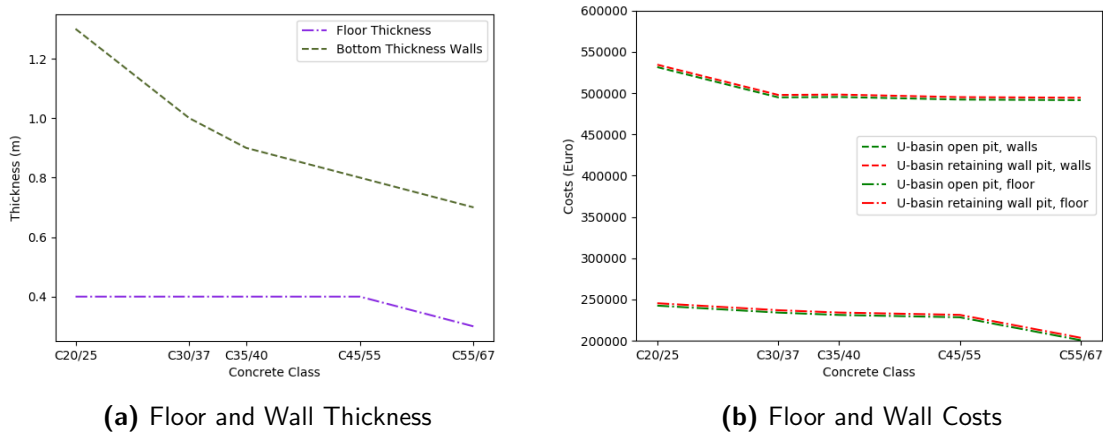


Figure 14-23: Detailed Cost Development as a Function of Concrete Strength

In Table 11-2 The MKI values are given for the different concrete classes. One of the disadvantages of stronger concrete is the environmental impact. However, advantage of strong material is that lesser material is needed to meet sufficient strength. For the environmental impact this leads to an optimum which can be easily seen in Figure 14-24 with a concrete class of C35/40. As from there the MKI value increases while the dimensions decrease.

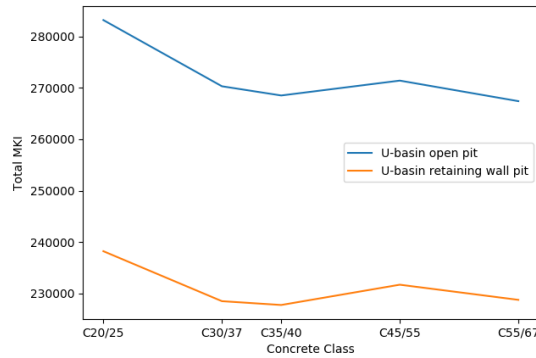


Figure 14-24: MKI Development as a Function of Concrete Strength

When looking at the values of the MKI from the U-basin retaining wall pit the difference between the highest value (C20/25) and the optimum (C35/40) is €10.468 and the difference between C35/40 and the third lowest (C55/67) is only €1.000. Comparing this with the costs it can be found that the difference between C20/25 and C35/40 is €51.039 while the difference between C35/40 and C55/67 is €28.785. From this it can be concluded that even stronger concrete will lead to only 10% increase of environmental impact compared to the accomplished decrease of the impact, while the costs will decrease an extra 56% compared to the accomplished decrease in costs.

14-10 Anchor Angle

The retaining walls make use of grout anchors. For the basic case these anchors haven an angle of 30 degrees into the ground. The angle of the anchor can influence the moment distribution of the retaining wall, the number of anchors needed and the shear force in the capping beam. In Figure 14-25 can be seen that for the Juliana lock case only the amount of anchors differ with the angle. Which lead to a change of about €25.000 which is 1.9% of the total price. It should be noted that in the end it is most important that the anchor reaches a sand layer. This will be the design requirement.

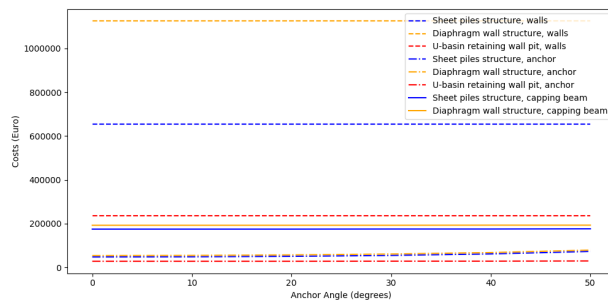


Figure 14-25: Cost Development as a Function of Anchor Angle, including Anchors, Walls and Capping Beams

14-11 Unit Prices

The dimensions of the structures depend on the needed strength, the materials used and the type of structure. From this dimensions follow the amount of material and the process to construct the structures. The total costs will depend on these values. However, the unit prices of the materials heavily determine these total costs. When the unit prices change the cost accrual of the alternatives will change which might lead to different order in financial preference.

It is hard to get useful results since the unit prices from all sections can change depended or independent form each other. In the Netherlands the prices of the materials and processes are relatively constant. When other countries are taken into account prices may differ. This is due to the costs of labour, the availability of curtain materials and the experience with specific processes.

The unit price which is not relatively constant within the Netherlands is the soil removal. The prices which need to be payed for the removal and processing of the excavated soil closely depend on the quality of material. When clean sand is excavated this sand can be re used in other projects or be sold to the highest bidder. In this way profit can be made on the excavation. On the other hand, when the soil is contaminated it will be very costly to process and clean the soil. In general in can be said that the excavation of sand will not effect the costs while the processing of peat and clay will influence the costs. For a costs estimation of a project it is thus important to know the quality of the soil.

Chapter 15

Discussion

15-1 Characteristics of the Alternatives

In the Design-tool only a structural design of the alternatives is included. Assumed is that all the alternatives will be constructed in the way they are designed and no failures during construction will occur. Restrictions imposed by the surrounding areas are also not included in the model. Possible uncertainties, restrictions and advantages of the alternatives are discussed below. An summary of these advantages and disadvantages is given in Table 15-1.

15-1-1 Sheet Pile Wall Structure

Locks are often constructed next to existing structures. During construction these structures may not encounter to much nuisance, or even worse, damage. One of the causes of the nuisance or damage are the driving or vibration of the sheet piles into the ground. Often the driving or vibration of sheet piles is a restriction in a project. To overcome this nuisance sheet piles can be pushed into the ground. However, this have quite a lot of technical restrictions looking at soil type and depth of the piles.

Another problem during the construction of sheet pile walls is the monitoring. Since the sheet piles are connected below ground level it is hard to check whether the interlocks between the mutual sheet piles are rightly executed.

15-1-2 Diaphragm Wall Structure

For the construction of the Diaphragm walls the excavation, placing of the reinforcement cages and the casting of the concrete all have to be executed below the ground level with no visual assistance due to bentonite fluid. As a result it is not possible to check whether the construction is preformed in the correct way. Failures can only be discovered when total excavation is done. To repair these failures can be expansive and time consuming. Frequent failures are:

- Leakage: In between the panels leakages can occur due to bad distribution of the concrete or containment of bentonite fluid. Leakage can lead to surplus water in the structure (this is not a major problem in case of navigation locks) but also to washout of the soil which can lead to subsidence of the surface above.
- Instability of the shaft: During the excavation of the shaft the bentonite could not be sufficient to ensure stability of the shaft. Because of this the shaft can deform which can result in subsidence of the surface above and damage to surrounding buildings. Also deviations of the thickness of the wall can occur.
- Concrete cover: The containment of bentonite fluid, deviation of the placement of a reinforcement cage or bad distribution of the concrete can lead to a lack of concrete cover. This can lead to exposure of the reinforcement steel which can have negative impact on the strength and durability of the diaphragm wall.

These failures lead to higher risk of the project. This have lead to high costs and large delays in recent projects in the Netherlands.

15-1-3 U-Basin Open Pit

The main problem with an open building pit is the lack of space. A open building pit has gentle slopes to ensure stability. These slopes will lead to a lot of extra space needed during construction, this is often not possible. Another problem whit open building pits is that pumping of water is necessary to ensure a dry building pit. Pumping will lead to a lower water table in the surrounding area which can lead to displacement of structures and can have consequences for agricultural lands. Because of these problems pumping of the building pit is often not allowed. Measures can be taken to make sure pumping of the building pit will not influence the surrounding water table. These measures are often complicated and expensive.

A disadvantage of the retaining wall structures and a advantage of the U-basin structures is the possibility for application of a complex filling and emptying system in the floor or in the walls. Because of the dry construction of the U-basin formwork can be easily placed in the right way to construct longitudinal culvert systems. Longitudinal culvert systems are needed when high water heads occur and fast leveling is needed.

15-1-4 U-Basin Retaining Wall Pit

For the U-basin with a retaining wall building bit has the advantages and disadvantages of both a U-basin structure and sheet pile wall structure. The issue with the lowering of the water table can be easily tackled with the construction of water retaining layer like a UWC floor.

Table 15-1: Advantages(+) and Disadvantages(-) Alternatives

	Space During Construction	Risk During Construction	Nuisance During Construction	Monitoring During Construction	Ground water Pumping	Environmental Impact	Total Costs
Sheet Pile Wall Structure	++	-	-	-	+	+	+ +
Diaphragm Wall Structure	++	--	+	--	+	+ -	+
U-Basin Open Building Pit	--	+	+	++	-	-	+
U-Basin Retaining Wall Building Pit	+	+ -	+ -	+	+	+ -	--

15-2 Cases

As case study the Juliana Lock is used for validation and as a basic case for the results from the model. This case is used because the dimensions and the water head are in the mid range of locks in the Netherlands. Also a sufficient amount of clear information is available for this lock.

Due to the lack of official documentation no other locks are used for validation. There are values available to calculate other locks with the model but there are no documents available to check the correctness of the outcome of the model.

However, to check whether the model is applicable for different cases calculations have been executed on several locks constructed in the Netherlands. An overview of the outcome of these calculations is given in Table 15-2. The most attractive alternatives are underlined.

Table 15-2: Results Other Cases

	CEMT Class	Length	Width	Water Head	Sheet Piles Wall Structure	Diaphragm Wall Structure	U-Basin Structure Open Pit	U-Basin Structure Retaining Wall Pit
Eefde	Va	125 m	12,5 m	7,4 m	<u>€2.223.496</u>	€2.577.402	€ 4.046.552	€ 4.897.028
Lith	Va	200 m	17,5 m	5,0 m	<u>€2.860.862</u>	<u>€3.611.382</u>	€ 5.555.735	€ 6.149.142
Maasbracht	Va	142 m	16,0 m	11,8 m	€ -	<u>€3.677.391</u>	€ 9.169.745	€14.301.999
Nieuwe Waterweg	VIb	215 m	23,8 m	6,2 m	<u>€5.320.502</u>	<u>€5.352.972</u>	€11.904.427	€13.082.338
Spoldersluis	Va	142 m	14,0 m	3,3 m	<u>€2.216.004</u>	<u>€2.420.786</u>	€ 2.790.866	€ 3.749.723
Johan Friso Sluis	III	70 m	9,0 m	0,6 m	<u>€ 461.376</u>	€ 789.987	€ 615.079	€ 2.307.071
Sluis II Tilburg	II	65 m	7,5 m	3,0 m	<u>€1.021.331</u>	€ 1.343.676	<u>€ 481.546</u>	€ 3.572.229
Kreekrak Sluis	VIb	320 m	24,0 m	2,2 m	<u>€4.514.191</u>	<u>€5.699.680</u>	€ 7.754.354	€10.249.388

15-3 Model Discussion

The Design-tool is created for a master thesis project which lasted for 9 months. During this process choices were made about the model and knowledge was expanded. In this section the

choices made will be evaluated and improvements are shown.

15-3-1 Use of External Programs

The model is written in Python. However, the Python model uses external programs as input, output and some calculations.

Input Excel

The user interface is an Excel file. This program is chosen for its well-arranged interface. Most people know how to use the program. For the admin it easy to lock certain cells for the user. An other pro is that separation signs adapt automatically to the excel preferences for the user. In this way is more easy for user to correctly fill in the model.

The first thing that happens in the Python model is loading all the input to Python. This is a quick process which only has to happen once.

Output Excel

The output of the dimensions and the final output of the costs will be exported to Excel. This is also done because of the easy workability of Excel. The exportation to Excel is a quick process and makes it easier to use.

Costs Diaphragm Wall

The costs calculation of the diaphragm wall is done in an Excel Sheet developed within RHDHV. These are simple calculations which will not make the design tool significantly slower. The benefit of the Excel Sheet is that when RHDHV changes the costs these values easily adjust.

VBA

The concrete cross sections of the U-basin, the capping beam and UWC floor of the retaining wall structures are calculated with verified Excel Visual Basic for Applications (VBA) sheets from RHDHV. These sheets are used since they where already available and work in a good validated way. A good thing about the validation is that the calculation is checked by experts, this improves the reliability of the Design-tool.

However, VBA is, compared to Python, a slow coding platform. The Python model opens the VBA sheet and enters the needed values in the specific cells. After this, the Python scripts gives the command to run the VBA calculation. This is a slow progress where the model needs to wait for. When this calculation is done the model reads the output from the VBA and when needed the process is repeated with new values.

Since the slow calculation speed of the VBA it is recommended to not use this sheets again. It will take some time to rewrite the calculations in Python and deal with verification. But, this will pay of at the end due to the time reduce during calculation.

D-Sheet Piling

The calculation of the retaining walls is done by the program named D-Sheet Piling. A program of the 'D series' from the independent research institute Deltares. It is a verified tool which is used by many engineering consultants, contractors and universities.

The external program is used for multiple reasons. The first one is that the program includes various complicated calculations which otherwise should be written in Python. Then the calculations would need new verification. The second reason is that the program creates a visualization of the problem, this can give an better inside for the user to check whether the input is correct. The third reason is that many companies and potential clients demand the use of D-Sheet Piling. Since it is a widely used program many companies are confident with results from the program.

A down side of the use of D-Sheet is that since it is an external program it requires extra time for calculation. However, this does not outweigh the benefits of the program. Also it should be checked whether a possible new version of the program influences the results of the model.

15-4 User

The goal of the creation of the Design-tool is to optimize preliminary design executed by a civil engineer. Because of this the tool can be used by a civil engineer which has proper knowledge about navigation locks. The user should enter the input values correctly. When wrong input is used an error will appear or wrong results will be created, in other words, "Garbage in, garbage out" as Army Computer Specialist William D. Mellin already stated in 1957 when he was working with one of the early computers.

Because of this the user should always check there inputted values, and not least the created output. Since it is possible to overlook a mistake the result of the design-tool should be reviewed on errors. The engineer should check if the values are as expected. If not, the input values should be reviewed or the deviation should be reasoned.

The motivation of the Design-tool is to help the engineer, not to replace him. The time saved by the engineer can be used to bring the design to a higher level or think in creative solutions.

15-5 Application of the Design-tool

The purpose of the Design-tool is to optimize preliminary design. This makes the tool optimal to review multiple different alternatives in a short span of time. When the design tool is not used, there mostly is no time available to review all possible designs. An estimation is made which design could be most preferable and that design will be evaluated. That will limit the variety of designs which are looked into which can lead to that possible good designs are overlooked.

15-5-1 When to use the Design-tool

The design tool is most useful for relatively standard circumstances in greenfield areas. For those cases the output of the design is already quite detailed and the costs estimation will be reliable. The output can be easily used for the final design.

When the design becomes more complicated, for instance due to a lack of space, the output of the design tool needs more reviewing. The alternatives need more examination if they fulfill the requirements. If not, the designs may need some adaptations but can be used as a solid starting point for a more detailed design.

For extreme special cases the Design-tool is less useful. However the model can be used to get insight in what the designs would be under less special circumstances. The designs from the model can be elaborated to check what will be the main cost drivers and where the focus should be for the more detailed design.

For scientific manners the Design-tool can be used to quickly evaluate changes in design steps, material properties or unit prices.

15-5-2 When to Create a Design-tool

The urge to create a Design-tool depends on the time which is needed to create the Design-tool and the time which can be saved by using the Design-tool. In general, when the time to create the Design-tool is less than the time which can be saved it is profitable to execute the automation. The time to create a Design-tool is closely linked to the complexity of the design process and the calculations which need to be done. The time which can be saved is time which is needed for the design process without a design tool minus the time which is needed with the design tool multiplied by the repetitions of the calculation. The main drivers are thus the efficiency of the Design-tool and times it will be used.

For scientific purposes the amount of repetitions will easily become large. This is because when research on a specific topic is done lot of data is needed to come with a reliable conclusion. With a design tool small adjustments in parts of the design can be easily implemented and calculated. In this way detailed research can be done on multiple parts of the design process.

For commercial purposes it is harder to estimate the time saving with the design tool. It depends on the real purpose of the tool. When the Design-tool is created to automate a specific process which should be done only one time for each project it is easy to estimate the time saving. However when the design tool is used to try different types of alternatives and play with variables to get a better understanding of the system to optimize design it is harder to find the time saving. Often this extensive examination is only done when a Design-tool is available. This means that in this case a Design-tool does not directly lead to time saving but will lead to quality improvement of the project.

In this thesis the Design-tool is used for scientific research. The model is extensively used to change all the parameters, dimensions and materials to investigate what the influences of all the changes will have on the lock chamber design and its different alternatives. When no model was present all these calculations should have been done by hand which would be too time consuming. After the scientific research the Design-tool can be used for the commercial purposes by RHDHV. This combined makes the Design-tool sufficient time saving.

Conclusion and Recommendations

16-1 Conclusion

In this research, a Design-tool to calculate the chamber of inland navigation locks in the Netherlands is created to optimize preliminary design. A better understanding of the costs accrual is created and the impact of different parameters is evaluated.

16-1-1 Influence of Main Dimensions

It has been found that for the Juliana lock case the sheet pile wall structure is economical most attractive. Followed by the U-basin with an open building pit, the diaphragm wall structure, and, least attractive, the U-basin with a retaining wall building pit. The main dimensions of the lock are mostly based on design requirements, for a specific lock chamber one set of dimensions is used. The main dimension, thus, only differs for different locks in different locations.

First of all, the length of the structure does hardly influence the preference of the different alternatives. This is because all the prices and processes are calculated per running meter of lock, except for the mobilization and demobilization costs for the structures. However, due to the relatively large lengths of the locks in general, and thus the high total costs, these differences are negligible.

With an increasing lock width especially the dimensions of the floor increase, this affects the U-basin structures the most. Due to this effect, from a width of 19.2 m the diaphragm wall structure becomes economically more feasible compared to the U-basin with an open building pit. This means that only for locks with ship class CEMT VI the diaphragm wall structures are favourable. The environmental impact of the sheet pile wall structure is the lowest, regardless the width. Up to 9 m (CEMT III) the U-basin with retaining wall pit has lower environmental impact compared to the diaphragm wall structure.

The water depth mainly impacts the dimensions of the walls. Because of this the U-basin with an open pit is less expensive for low water depths compared to the diaphragm wall structure

which is restricted by a minimum thickness. Due to the rapidly increasing soil handling costs this only holds for a depth up to 4.2 m (CEMT Va).

The location of the lock determines the required water head over the lock. When the right tension piles are used, the development of the price over water head has the same relative increase for all the different structures. The sheet pile wall structure comes out most favorable but its advantage decreases with a larger water head. Up to a water head of 4 m, the U-basin with open pit is more attractive compared to the diaphragm wall. This is within the range of most locks in the Netherlands.

Lock chambers can be constructed in multiple types of soil profiles. The soil profiles mainly depend on the location, but, when there are multiple options, profiles with a high rate of sand are preferable. Strong soils make construction of the lock chamber less complicated.

16-1-2 Environmental impact

The alternative with the lowest environmental impact is the sheet pile wall structure. This can be explained by the limited material that is used for the chamber and the efficient use of the building pit as finite structure. The use of the building pit can be seen when comparing the sheet pile wall structure with the U-basin alternatives. The construction of the walls and floor of the U-basin has lower impact compared to the steel sheet pile wall and the UWC floor, but still the sheet pile wall structure has less impact. The diaphragm wall structure has a large MKI value, mainly because of the minimum thickness of both the UWC floor and the diaphragm wall, but also due to the use of both concrete and a lot of reinforcement for the diaphragm wall.

For the U-basin structures the type of used concrete is of significant influence. Stronger concrete leads to more slender structures, which means less material and thus less environmental impact. However, higher strength concrete has a larger impact per volume of material used. Beside this, due to the slender structures more reinforcement can be needed, which has a high environmental impact.

It is concluded that the use of the building pit in the final structure has the least environmental impact. However, when for a particular reason a U-basin needs to be constructed, soil work needs to be limited. In the results it is shown that the construction of a temporary sheet pile building pit has less environmental impact compared to an open building pit. The large amount of soil work resulting from an open building pit leads to a significant amount of emissions.

16-1-3 CO₂ Tax

The CO₂ tax, proposed by the Dutch Government, of €0,05 per kg CO₂ equivalent will increase the total prices of the alternatives with 4% to 6%. With this proposed CO₂ tax the order of financially most attractive alternatives will not be altered. Multiple environmental organizations claim that the environmental impact of the CO₂ emissions will lead to a damage of €0,20 per kg CO₂. A CO₂ tax of €0,20 per kg CO₂ equivalent would bring the CO₂ tax share of the total costs to 16%, 17%, 20% and 13% of the total costs for respectively the sheet pile wall structure, diaphragm wall structure, U-basin open pit and the U-basin with a

retaining wall pit. A CO₂ tax of €0,20 per kg CO₂ equivalent will still not lead to a change in order of economical feasibility, but it is noticed that for a U-basin with open pit the CO₂ tax will have a significant impact. When a CO₂ tax of €0,68 per kg CO₂ equivalent will be introduced, the diaphragm wall structure will become financially more attractive than the U-basin open pit structure. The CO₂ emissions of the design alternatives are closely related to the MKI values of the structures.

16-1-4 Influence of unit prices

The unit prices heavily influence the total price of the structures. However, the unit prices in the Netherlands are fairly constant and an extensive cost study should be elaborated to be able to tell something substantive about evolution of the unit prices. One cost parameter in particular is subject to large fluctuations, the soil removal. The costs for removal of the soil depends on the quality of the soil, high quality sand can be profitable while heavy contaminated soil can be expensive to process. For heavy contaminated soil an open building pit is not preferable due to the high amount of excavation.

16-1-5 Creation of a Parametric Design-tool

The main goal of this thesis was to create a parametric Design-tool for the preliminary design of a lock chamber. Two main lock chamber types are considered, retaining wall structures and concrete U-basin structures. For the retaining wall structures sheet pile walls, combi walls and diaphragm walls are elaborated. Under water concrete will be used for the floor. A U-basin can be constructed within an open building pit or a temporary retaining wall building bit, both options are included.

The basis of the Design-tool is created in the programming language Python. To optimize ease of use, level of validation and reliability external programs are coupled with the Python script. To calculate the retaining wall structures a connection is made with D-Sheet Piling from Deltares. Structural VBA sheets developed within RHDHV are used to optimize structural calculation for the capping beams, UWC floors and the U-basin. Two outputs will be generated, one with the dimensions of the different structures and one with an overview of the costs and environmental impact per alternative.

Research is being performed on which variables should be taken into account. These variables mainly originate from the basic lock design. Other variables relate to the specific lock chamber structures, which are included in the Design-tool. The main basic lock design variables are the length, width and height of the lock. These values can be introduced directly by the qualified user. The Design-tool is also able to create the length, width and height automatically. In this case the qualified user should enter the ship classes that will be present in the lock and the water levels for which locking needs to be possible.

To become a parametric model multiple alternatives and options should be calculated in the model. In this way the model is parametric since all four alternatives are elaborated and compared. Furthermore, each main structure will have multiple options of materials, dimensions and layout.

16-1-6 Parameters within the Design-tool

To limit the processing time of the Design-tool the tool will not be fully parametric, this means that not all the parameters will be variable. This is because not all the variables will have an impact when they are adjusted. The impact of the variables has been reviewed in the results. The most outstanding variable is the length of the tension piles. The length of the tension pile determines the number of piles needed. An optimum can be found due to the mutual dependency of the piles. Also, the thickness of the UWC floor depends on the number of piles. The length of the tension piles thus should be parametric and variable.

For the tension piles also the width of the piles can be changed. The width of the piles has less impact than the length of the tension piles. However, in some circumstances the tensions piles can be a significant part of the total costs which makes the impact of the pile width also relevant.

The surcharge load next to the lock chamber is often a design criterion. However, it is concluded that a higher surcharge load can lead to a large rise in costs. This holds especially for sheet pile and combi pile walls. Due to the large impact the surcharge should be parametric. The user of the model can then analyze the differences in costs and reconsider the design criteria of the surcharge load.

Different types of concrete for the UWC floor have hardly any impact on the costs of the total structure. Stronger concrete will not lead to significant smaller dimensions of the floor. Stronger concrete will only lead to more environmental impact. Because of this, it is recommended to use concrete class C20/25 as default or to let the user choose its specific preference.

The casting method of the UWC floor has no impact in the Juliana lock case. However, the casting tolerance can have quite an influence on the costs of the UWC floor. The casting method often depends on the preferable equipment of the contractor. Because there are only two options within the model, it is preferred to make the casting method parametric to see whether it is auspicious to use different equipment.

In addition of the under water concrete, also the regular reinforced concrete can differ in strength. This mainly effects the U-basin structures since the whole structure is made of concrete. Higher strength concrete will lead to smaller dimensions and thus less material. However, due to the increase of emissions with stronger concrete also smaller dimensions can lead to more environmental impact. To make both the costs and the environmental impact of the concrete visible, the concrete strength should be made parametric.

The grout anchor that will be used does influence the total structure, both the length, angle and diameter of the anchor. In the model an estimation of the needed anchors is made for a specific anchor. The placement of the anchor is very specific per location since a sand layer should be reached. Because of this the alignment of the anchor should be designed separately for each project. The design tool gives an indication of the anchor costs but the anchor variables should not be parametric.

16-2 Recommendations

The goal of this thesis is to the optimize preliminary design of lock chambers. To fully optimize the development of the tender proposal, the whole lock design process should be optimized. In

this research the lock chamber is chosen as a starting point, because it represents a significant part of the total lock costs, the chamber is responsible for the main dimensions of the lock and the design of the chamber is relatively independent of the other parts of the lock. However, the lock gates and the lock head also contribute to a large part of the total lock costs. With the chamber design resulting from the Design-tool the dimensions can be used as input for the design of the gates and heads. Multiple combinations of gates and heads are available, which makes them time consuming to calculate. Because of this it can be time saving to construct an additional parametric design tool for these parts. It is recommended to create one design tool for both the lock head and the gates due to mutual dependency. Also, a parametric design tool can contribute to evaluate the different alternatives in a structured manner.

16-2-1 Standardization

Rijkswaterstaat is currently looking at standardization within hydraulic engineering. Structures can become economically more feasible when repetition of building processes are involved. Due to the large number of navigation locks which need renovation or replacement it can be interesting to examine the possibilities of standardization for navigation locks and thus lock chambers.

The Design-tool can be used to calculate multiple alternatives for all the locks which need replacement to find similarities in the designs at different locations. Also, it can be checked what the influences of possible over standardization and overdimensioning will be on the costs of the lock chamber.

16-2-2 Sea Locks

The scope of this thesis is inland navigation locks. The Design-tool can be easily expanded to cover sea locks as well. This can be done by including ship classes of marine vessels and the dimensions and forces resulting from these ships. The main differences are the sizes of the lock that become larger. Especially interesting is the increasing depth. When the lock becomes larger other structure types become interesting to take into account for lock design. Concrete L-walls should be looked into and for even higher retaining heights deep foundation walls with combi walls, a L-wall and tension piles should be included. The influence of the tides are already discussed in Section 2. Sea waves which enter the lock when the gates are open should be taken into consideration. Due to the salt sea water additional attention should be paid to the conservation of the structures.

16-2-3 Combi Walls in Bentonite Excavations

A large advantage of diaphragm walls over sheet pile walls is the ability of installation without vibrations. Because of the requirement of driving without vibrations often diaphragm walls are chosen over sheet pile or combi walls despite their higher costs.

In Germany they are testing sheet piles and combi walls with the placement method of diaphragm walls. This means that a shaft will be excavated and filled with bentonite. Instead of lowering the reinforcement cages and casting it with concrete, sheet piles or combi walls

will be lowered into the shaft. When the sheet piles or combi walls are in place the bentonite will be replaced with soil. In this way, a vibration free wall can be placed without the use of concrete. It should be interesting to take this type of structure into account, since it has the benefits of both the sheet pile or combi wall and the diaphragm wall.

16-2-4 Open floors

In the Design-tool only closed floors are included. Closed floors are most common, since there is no loss of water and the floor acts as a strut. However, in some options the water tightness of the floor is not important, which means that holes can be made to relieve upward water pressure. In this way the floor still acts as a strut, but can be designed more slender due to the smaller moments. When no strut is needed for the stability of the walls and water tightness is not important it can be chosen to construct a completely open floor consisting of rocks. This might lead to a decrease in costs.

Bibliography

- Agency, I. E. and UN Environment (2018). Towards a zero-emission, efficient, and resilient buildings and construction sector. Global Status Report 2018.
- ArcelorMittal (2019). General Catalogue - Steel Foundation Solutions.
- ArcelorMittal Sheet Piling (2014). The HZ ® -M Steel Wall System 2014 The development of the HZ ® / AZ ® combined wall system.
- Azhar, S. (2011). Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry. *Leadership and Management in Engineering*, 3(Bazjanac 2006):241–253.
- BAUER (2015). *Diaphragm and Cut-off Walls*. Schrobenhausen.
- Bell, R. G. and Callan, D. (2011). Social Cost of Carbon in U.S. Climate Policy, Plain English. *Environmental Law*, (July).
- Berkeley Earth (2018). Global Temperature Anomaly.
- Bew, M. and Richards, M. (2008). BIM Maturity Diagram. In.
- Binnenvaartcijfers (2018). *Binnenvaart Cijfers*. Bureau Voorlichting Binnenvaart.
- Braam, C. and Lagendijk, P. (2011). *Constructieleer - Gewapend Beton*. Aeneas, seventh edition.
- British Standard Institution (2007). *Eurocode 3, Design of steel structures, Part 5: Piling*, volume 3. British Standard, London.
- British Standard Institution (2008). *Eurocode 2 : Design of concrete structures*. British Standard, London.
- British Standard Institution (2009). *Eurocode 7: Geotechnical design*. British Standard, London.

- Buijs, M. (2018). Visie op Sectoren, Transport & Logistiek. Technical Report April 2018, ABN AMRO.
- Centrale Commissie voor de Rijnvaart (2018). Jaarverslag 2018 Europese Binnenvaart Marktopservatie.
- Chi, H. L., Wang, X., and Jiao, Y. (2015). BIM-Enabled Structural Design: Impacts and Future Developments in Structural Modelling, Analysis and Optimisation Processes. *Archives of Computational Methods in Engineering*, 22(1):135–151.
- COB/CUR-commissie T144/C174 (2010). *CUR-Publicatie 231 Handboek diepwanden - Ontwerp en uitvoering*. Gouda.
- CROW (2002). Standaardsystematiek voor Kostenramingen in de GWW. Technical report.
- CUR-commissie C123 (2003). *CUR 2001-4 Ontwerpregels voor trekpalen*. Gouda, second edition.
- CUR-commissie C126 (2012a). *CUR 166 Damwandconstructies, deel 1*. Gouda.
- CUR-commissie C126 (2012b). *CUR 166 Damwandconstructies, deel 2*. Gouda.
- CUR-commissie C152 (2011). *CUR 236 Ankerpalen*. Gouda, first edition.
- De Gijt, J. G. and Broeken, M. L. (2013). *Quay Walls*. Number December. CRC Press, second edition.
- de Winter, E., Arkesteijn, R., and Barten, P. (2017). Herziening CUR Aanbeveling 77 (4). *CEMENT*, 4:42–52.
- Deltares (2018). D-Sheet Piling.
- Deltares systems (2016). D-Sheet Piling User Manual.
- Doeksen, J. W. (2012). Gate Design For Large , High Head Locks " The development of an innovative lock gate ". Technical report, Delft University of Technology.
- Eadie, R., Browne, M., Odeyinka, H., McKeown, C., and McNiff, S. (2013). BIM implementation throughout the UK construction project lifecycle: An analysis. *Automation in Construction*, 36:145–151.
- Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A guide to Building Information Modeling for owners, managers, designers, engineers and contractors*. Second edi edition.
- Erbisti, P. C. (2002). *Design of Hydraulic Gates*. A.A. Balkema Publishers.
- Hendriks, P. (2018). Cement, revolutiebestendig. *CEMENT*, 6(3):56–61.
- Huybrechts, N. and van Lysebetten, G. (2014). Infofiche 70.02 - Uitvoering van diepwanden. Technical report, WTCB.
- Jonker, F. and Hagenaaars, P. (2015). *CUR-Aanbeveling 77: 2014 Rekenregels voor ongewapende onderwaterbetonvloeren*. CUR, second edition.

- Klaver, J. and van der Lee, T. (2018). Proposal Legislative Change Environmental Taxes.
- Kranz, E. (1953). *Über die Verankerung von Spundwänden*. Verlag von Wilhelm Ernst & Sohn, Berlin, zweite, un edition.
- Le Quéré, Corinne, Andrew, R. M., Friedlingstein, P., Sitch, S., Pongratz, J., Manning, A. C., Korsbakken, J. I., Peters, G. P., Canadell, J. G., Jackson, R. B., Boden, T. A., Tans, P. P., Andrews, O. D., Arora, V., Bakker, D. C. E., Barbero, L., Becker, M., Betts, R. A., Bopp, L., Chevallier, F., Chini, L. P., Ciais, P., Cosca, C., Cross, J., Currie, K., Gasser, T., Harris, I., Hauck, J., Haverd, V., Houghton, R. A., Hunt, C. W., Hurtt, G., Ilyina, T., Jain, A. K., Kato, E., Kautz, M., Keeling, R. F., Goldewijk, K. K., Körtzinger, A., Landschützer, P., Lefèvre, N., Lenton, A., Lienert, S., Lima, I., Lombardozzi, D., Metzl, N., Millero, F., Monteiro, P. M. S., Munro, D. R., Nabel, J. E. M. S., Nakaoka, S.-i., Nojiri, Y., Padin, X. A., Pfeil, B., Pierrot, D., Poulter, B., Rehder, G., Reimer, J., Rödenbeck, C., Schwinger, J., Séférian, R., Skjelvan, I., Stocker, B. D., Tian, H., Tilbrook, B., Tubiello, F. N., van der Laan-Luijkx, I. T., van der Werf, G. R., van Heuven, S. M. A. C., Viovy, N., Vuichard, N., Walker, A. P., Watson, A. J., Wiltshire, A. J., Zaehle, S., and Zhu, D. (2018). The Global Carbon Budget.
- Linssen, J. (2018). De scope van parametrisch ontwerpen. *CEMENT*, 6(3):18–22.
- Maas, G. (2011). *Appendix Comparison of quay wall designs in concrete, steel, wood and composites with regard to the CO₂-emission and the Life Cycle Analysis*. PhD thesis, Delft University of Technology.
- Maas, T., De Gijt, J., and Dudok Van Heel, D. (2011). *Comparison of quay wall designs in concrete, steel, wood and composites with regard to the CO₂-emission and the Life Cycle Analysis*. PhD thesis, Delft University of Technology.
- Ministerie van Infrastructuur en Milieu and Ministerie van Economische Zaken Landbouw en Innovatie (2012). Deltaprogramma 2013, Bijlage H; Vervangingsopgave Natte Kunstwerken. Technical report.
- Molenaar, W. F. (2011). *Hydraulic Structures, Locks*. Delft University of Technology, Delft.
- Moore, F. C. and Diaz, D. B. (2015). Temperature impacts on economic growth warrant stringent mitigation policy. *Nature Climate Change*, 5(2):127–131.
- Normcommissie 351 006 (2007). NEN 6744 Geotechnics - Calculation method for shallow foundations. Technical report, Nederlands Normalisatie-instituut, Delft.
- Normcommissie 351 006 (2017). NEN 9997-1+c2, Geotechnical design of structures, Part 1. Technical report, Nederlands Normalisatie Instituut, Delft.
- Normcommissie 351001 (2016). NEN-EN 1992-1-1 + C2/NB. Technical Report november, Nederlands Normalisatie-instituut.
- Normcommissie 351006 (2015). NEN-EN 1538 + A1 - Execution of special geotechnical work - Diaphragm walls. Technical report, Nederlands Normalisatie-instituut.
- PIANC (1986). Final report of the international commission for the study of locks. Technical report.

- PIANC (2006). Report of WG 26 "Design of Movable Weirs and Storm Surge Barriers". Technical Report January, International Navigation Association.
- PIANC (2009). Innovations in Navigation Lock Design - Final Version 20b.
- Quist, P. and Verheij, H. J. (2010). Staat van de scheepvaart en de binnenvaartwegen in Nederland 2009. Tweejaarlijkse trendrapportage. Technical report, Rijkswaterstaat, TU Delft, Delft.
- Rense, R. (2008). Klimaatverandering en scheepvaart op de grote rivieren. <https://www.helpdeskwater.nl/onderwerpen/water-ruimte/klimaat/onderwerpen/scheepvaart/>.
- Riedijk, W., Weigand, R., Bron, W., and Reurings, R. (2018). Online applicaties gebouwd door ingenieurs. *CEMENT*, 6:36–40.
- Rijkswaterstaat (2004). ProSes Hoordrapport Kostenopstelling t.b.v. MKBA en S-MER. Technical Report april, Ministerie van Verkeer en Waterstaat.
- Rijkswaterstaat (2017). *Richtlijnen Vaarwegen 2017 Kader verkeerskundig vaarwegontwerp*.
- Rijkswaterstaat (2018). DuboCalc. <https://www.dubocalc.nl/meer-informatie/>.
- Sacks, R. (2004). Evaluation of Economic Impact of Three-Dimensional Modeling in Precast Concrete Engineering. *Journal of Computing in Civil Engineering*, 18(4):301–3012.
- Slijk, R. (2013). Standardization in River Locks! Technical report, Delft University of Technology.
- Stichting Bouwkwaliiteit (2014). SBK Bepalingsmethode versie 2.0. Technical Report November, Stichting Bouwkwaliiteit, Rijswijk.
- Stiglitz, J. E., Stern, N., Duan, M., Edenhofer, O., Giraud, G., Heal, G., La Rovere, E. L., Morris, A., Moyer, E., Pangestu, M., Shukla, P. R., Sokona, Y., and Winkler, H. (2017). Report of the High-Level Commission on Carbon Prices. Technical report, Carbon Pricing Leadership Coalition.
- Van den Hurk, B., Siegmund, P., Klein Tank, A., Attema, J., Bakker, A., Beersma, J., Bessembinder, J., Boers, R., Brandsma, T., Brink, H. V. D., Drijfhout, S., Eskes, H., Haarsma, R., Hazeleger, W., Jilderda, R., Katsman, C., Lenderink, G., Loriaux, J., Meijgaard, E. V., Noije, T. V., Oldenborgh, G. J. V., Selten, F., Siebesma, P., Sterl, A., Vries, H. D., van Weele, M., de Winter, R., and van Zadelhoff, G. (2014). KNMI14: Climate Change scenarios for the 21st Century, A Netherlands perspective. *Scientific Report WR2014-01*, (May):115.
- van der Ploeg, C. (2018). De keuze voor parametrisch ontwerpen. *CEMENT*, 7(2):10–16.
- Van Doorn, M. (2018). Port of Rotterdam Hinterland, Masterclass Betuweroute.
- Volker Infradesign (2011). DO Ontwerpbasis civiel en geothechniek 2e Julianasluis Gouda. Technical report.

Vrijburcht, A., Beem, R. C. A., Boogaard, A., Glerum, A., de Graaf, M. A., Henneberque, S. D., Hiddinga, P. H., Kranenburg, D., van der Meer, M. T. J., Nagtegaal, G., Van der Paverd, M., Smink, L. M. C., and Weijers, J. (2000). *Ontwerp van Schutshuizen: Deel 1 + 2*.

List of Figures

1-1	Report Structure	2
2-1	Relations in a Lock	4
2-2	Components of a Lock	4
2-3	Wall with Deep Foundation (Molenaar, 2011)	7
2-4	Model Split Hinterland Transport Port of Rotterdam (Van Doorn, 2018)	14
3-1	BIM Maturity Diagram (Bew and Richards, 2008)	20
3-2	Total Cost Distribution Components of a Lock (Slijk, 2013)	22
3-3	Cost Distribution Per Lock Component (Slijk, 2013)	22
3-4	Percentage Design Cost per Component (Slijk, 2013)	23
4-1	Hydraulic Structures which need replacement in the coming years (Ministerie van Infrastructuur en Milieu and Ministerie van Economische Zaken Landbouw en Innovatie, 2012)	26
4-2	Framework Master Thesis	29
5-1	Lock Chamber Alternatives	32
5-2	Model Overview Design Tool	33
5-3	Model Analysis General Dimensions	34
5-4	Model Analysis Tension Piles	34
5-5	Model Analysis Under Water Concrete Floor	35
5-6	Model Analysis Capping Beam	36

5-7	Model Analysis Building Pit	36
5-8	Model Analysis D-Sheet Input File Creation	37
5-9	Model Analysis D-Sheet Calculation	38
5-10	Model Analysis D-Sheet Output	38
5-11	Model Analysis Retaining Wall Output	39
5-12	Model Analysis U-Basin Calculation	40
5-13	Model Analysis U-Basin Output	40
5-14	Model Analysis Dimensions Output	41
5-15	Model Analysis Costs Calculation	41
6-1	Water Levels Coastal Lock(Molenaar, 2011)	46
6-2	Sill Level (Vrijburcht et al., 2000)	48
7-1	Construction Alternatives U Basin(Vrijburcht et al., 2000)	50
7-2	Forces Acting on the Wall (Vrijburcht et al., 2000)	52
7-3	Forces Acting on the Floor(Vrijburcht et al., 2000)	52
7-4	Moments Resulting from the Loads (Vrijburcht et al., 2000)	53
7-5	Support Springs Soil Shallow Foundation	53
7-6	Effective Height Concrete	55
8-1	Sheet Pile Wall Overview (Vrijburcht et al., 2000) (modified)	57
8-2	HZ-AZ Combi Wall System, 4 and 2 sheet piles in between (ArcelorMittal Sheet Piling, 2014) (modified)	58
8-3	Grout Anchor (CUR-commissie C126, 2012b)	61
8-4	Anchor Length Grout Anchor (CUR-commissie C126, 2012b) (modified)	61
8-5	Shaft Resistance Tension Piles and Sheet Piles in Tension.	64
8-6	Shaft Resistance Tension Piles and Sheet Piles in Tension with Anchors	65
8-7	Bollard Recesses Capping Beam	67
8-8	Construction Stages Sheet Pile Chamber	68
9-1	Diaphragm Wall Construction (BAUER, 2015)	69
9-2	Components Diaphragm Wall (Normcommissie 351006, 2015)	70
9-3	Schematized Moment Diaphragm Wall	72
11-1	Life Cycle Analysis	82
11-2	Global Land-ocean Temperature Index (Berkeley Earth, 2018)	83

11-3 Global CO2 Emissions (Le Quéré et al., 2018)	83
13-1 Validation Ship Dimensions	90
13-2 Validation Lock Dimensions	90
13-3 Validation Tension Piles f1 Example	91
13-4 Validation Tension Piles f1	91
13-5 Validation Tension Piles Maximum Tension Force Example	92
13-6 Validation Tension Piles Maximum Tension Force	92
13-7 Validation Manual Input D-Sheet Calculation	92
13-8 Validation Design-tool D-Sheet Calculation	93
13-9 Validation Design-tool Example Calculations (de Winter et al., 2017)	94
13-10 Validation Design-tool Output From Example	94
13-11 Validation Design-tool Output From Design-tool	95
13-12 Validation Bollard Loads	95
13-13 Validation Soil Pressures U-Basin	95
13-14 Validation Soil Pressures	96
13-15 Validation Moment in Wall	96
14-1 Sheet Pile Wall Structure, Basic Case	100
14-2 Costs Sheet Pile Wall Structure, Basic Case	101
14-3 Diaphragm Wall Structure, Basic Case	101
14-4 Costs Diaphragm Wall Structure	102
14-5 U-basin Open Pit, Basic Case	102
14-6 Costs U-Basin Open Pit, Basic Case	103
14-7 U-Basin Retaining Wall Pit, Basic Case	103
14-8 Costs U-Basin Retaining Wall Pit, Basic Case	104
14-9 MKI Structures	105
14-10 Cost Development as a Function of CO2 Tax	106
14-11 Cost Development as a Function of Width	107
14-12 MKI Development as a Function of Width	108
14-13 Cost Development as a Function of Water Head	108
14-14 Detailed Cost Development as a Function of Water Head	109
14-15 Cost Development as a Function of Water Depth	110
14-16 Detailed Cost Development as a Function of Water Depth	110

14-17	Detailed Cost Development for Different Soil Profiles	111
14-18	Cost Development as a Function of Surcharge Load	112
14-19	Detailed Cost Development as a Function of Surcharge Load, Including Floors and Walls	112
14-20	Costs for Different Pile Width	113
14-21	Detailed Cost Development as a Function of Pile Length	114
14-22	Cost Development as a Function of Concrete Strength	116
14-23	Detailed Cost Development as a Function of Concrete Strength	116
14-24	MKI Development as a Function of Concrete Strength	117
14-25	Cost Development as a Function of Anchor Angle, including Anchors, Walls and Capping Beams	117
B-1	Van Leeuwen Anchor Catalog	VII
D-1	Stop logs (Doeksen, 2012)	XIII
D-2	Lift Gate (Doeksen, 2012)	XIV
D-3	Single Leaf Gate (Doeksen, 2012)	XIV
D-4	Mitre Gates (Doeksen, 2012)	XIV
D-5	Radial Sector Gates - vertical axis (Doeksen, 2012)	XV
D-6	Radial Sector Gates - horizontal axis (Doeksen, 2012)	XV
D-7	Flap Gate (Doeksen, 2012)	XVI
D-8	Rolling or Sliding Gate (Doeksen, 2012)	XVI
F-1	Code Ship Dimensions	XX
F-2	Code Lock Dimensions	XXI
F-3	Code Bollard Loads	XXI
F-4	Code Tension Piles	XXII
F-5	Code UWC Floor	XXIII
F-6	Code Capping Beam Estimation	XXIV
F-7	Code Capping Beam Check	XXV
F-8	Code Excavation Retaining Wall Building Pit	XXVI
F-9	Code Excavation Open Building Pit	XXVI
F-10	Code Sheet Pile Wall Structure Input (1)	XXVII
F-11	Code Sheet Pile Wall Structure Input (2)	XXVIII
F-12	Code Sheet Pile Wall Structure Input (3)	XXIX

F-13 Code Sheet Pile Wall Structure Output (1)	XXX
F-14 Code Sheet Pile Wall Structure Output (2)	XXXI
F-15 Code Resisting Moment Calculation Reinforced Concrete	XXXI
F-16 Code Diaphragm Wall Structure Input (1)	XXXII
F-17 Code Diaphragm Wall Structure Input (2)	XXXIII
F-18 Code Diaphragm Wall Structure Input (3)	XXXIV
F-19 Code Diaphragm Wall Structure Output (1)	XXXV
F-20 Code Diaphragm Wall Structure Output (2)	XXXVI
F-21 Code U-Basin Moments and Forces	XXXVI
F-22 Code U-Basin Floor (1)	XXXVII
F-23 Code U-Basin Floor (2)	XXXVIII
F-24 Code U-Basin Walls	XXXVIII
F-25 Code U-Basin Piles	XXXIX
F-26 Code Output Dimensions (1)	XL
F-27 Code Output Dimensions (2)	XLI
F-28 Code Input Dimensions	XLII
F-29 Code Input Dimensions	XLIII
F-30 Code Output Sheet Pile Wall Structure	XLIV
F-31 Code Output Diaphragm Wall Structure	XLV
F-32 Code U-Basin Structure Open Building Pit	XLVI
F-33 Code U-Basin Structure Retaining Wall Building Pit	XLVII
F-34 Code Excel Output	XLVIII

List of Tables

2-1	Different types of Lock Heads (Molenaar, 2011)	6
2-2	Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)	8
2-3	Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)	9
2-4	Alternatives for lock heads and lock chamber (Vrijburcht et al., 2000)	10
2-5	Alternatives for lock heads and lock chamber regarding construction method (Vrijburcht et al., 2000)	10
2-6	Alternatives for lock heads and lock chamber regarding construction method (Vrijburcht et al., 2000)	11
2-7	Longitudinal culvert systems (PIANC, 2009)	12
2-8	River/ Sea going vessels (Rijkswaterstaat, 2017)	14
2-9	CEMT Classes (Rijkswaterstaat, 2017)	15
2-10	ECE Recreational Vessel Classes (Rijkswaterstaat, 2017)	15
2-11	Design Bollard Loads CEMT Classes (Vrijburcht et al., 2000)	16
5-1	Method of Calculation U-Basin	42
5-2	Method of Calculation Sheet Pile Wall Structure	43
5-3	Method of Calculation Diaphragm Wall Structure	43
6-1	Minimum Dimensions Inland Navigation Locks (Rijkswaterstaat, 2017)	46
8-1	Delta Friction Angles (British Standard Institution, 2009)	60
10-1	Unit Prices based on RHDHV Values	80

11-1 Environmental Costs Indicators (Stichting Bouwkwiteit, 2014)	82
11-2 Environmental Costs Indicators (Stichting Bouwkwiteit, 2014)	84
11-3 MKI Contribution for the use of Concrete (Rijkswaterstaat, 2018)	86
12-1 Ground Profile Juliana Lock Gouda (Volker Infradesign, 2011)	88
13-1 Soil Properties Validation Example	96
13-2 Soil Properties Validation Example	96
14-1 Overview Results Basic Case Juliana Lock	100
14-2 Different Under Water Concrete, Total Costs	115
14-3 Different Under Water Concrete, MKI Value	115
14-4 Different Under Water Concrete, CO2 Tax (€0,05)	115
15-1 Advantages(+) and Disadvantages(-) Alternatives	121
15-2 Results Other Cases	121
A-1 CEMT Classes (Rijkswaterstaat, 2017)	IV
A-2 CEMT Classes (Rijkswaterstaat, 2017)	V
C-1 Environmental concrete cover demands Netherlands (Normcommissie 351001, 2016) IX	
C-2 Structural Exposure Classes Concrete Structures (British Standard Institution, 2008) X	
C-3 Environmental Classes Structures (British Standard Institution, 2008)	XI
E-1 Sheet Pile Profiles ArcelorMittal (ArcelorMittal, 2019)	XVII
E-2 HZ Pile Profiles ArcelorMittal (ArcelorMittal, 2019)	XVII

List of Symbols

A_p	Pile cross section
A_s	Area of reinforcement
b	Overall width cross section
Δc_{def}	Concrete cover in relation with construction
$c_{min,b}$	Minimum concrete cover in relation with bond
$c_{min,dur}$	Minimum concrete cover in relation with environmental conditions
c_{nom}	Nominal concrete cover
d	Effective depth of a cross section
e_0	Initial pore value soil
e_1	Pore value soil after pile installation
E	Youngs modulus
f_1	Compaction factor
f_2	Factor for relaxation for tensile force pile group
$f_{c,d}$	Design concrete compressive strength
$f_{y,d}$	Design yield strength of steel
$F_{bollard}$	Governing bollard load
$F_{tension,d}$	Design value tensile force pile
h	Height of a cross section
k_1	Constant reinforcement spacing
k_2	Constant reinforcement spacing
k_{axial}	Axial spring stiffness of a pile
$k_{elastic}$	Elastic extension of a pile
k_{mod}	Modification factor life time
k_{shaft}	Spring stiffness coming from the mobilization of the shaft friction
k_{soil}	Lift of the soil under a pile tip caused by unloading
K_0	neutral soil pressure coefficient
l_a	Distance between the anchors
l_b	Distance between the bollards
L_1	Pile length which contributes to the load transfer
L_A	Length of grout body anchor

L_{free}	Free anchor length
L	Pile length
$M_{E,d}$	Design moment
M_i	Auxiliary factor soil relaxation
$M_{max,bollard}$	Maximum moment bollard loads
$M_{max,soil}$	Maximum moment by horizontal pressures from sheet piles
$M_{R,d}$	Design moment resistance
$q_{c,z}$	Cone penetration pressure
$q_{c,z,d,i}$	Design cone penetration pressure at layer i
$q_{g,rep}$	Representative self weight
q_{soil}	Horizontal pressures
$q_{w,rep}$	Representative water pressure
$q_{z,rep}$	Representative upward swell pressure
$\Delta R_{e,i}$	Increase in relative density soil after pile installation
$R_{s,cal,max,i}$	Shaft friction pile
tol_{under}	Tolerance bottom UWC floor
tol_{upper}	Tolerance top UWC floor
$T_{d,i}$	Design value of auxiliary factor for tensile bearing capacity
z	Lever arm of internal forces
β_D	Interlock reduction factor
β_B	Capacity moment reduction factor
γ	Load/ material Factor
γ'_i	Effective volumic weight at layer i
$\gamma'_{d,i}$	Design value effective volumic weight at layer i
δ	Delta friction angle
η	Concrete compressive zone factor
σ'_v	Effective vertical soil pressure
σ'_h	Effective horizon soil pressure
φ'	Internal friction angle
\varnothing	Diameter reinforcement

Appendices

Appendix A

CEMT Classes

CEMT-Klasse	Motorvrachtschepen (Motorvessels)							Duwstellen (Barges)				
	RWS Klasse	Karakteristieken maatgevend schip**				Classificatie		RWS Klasse	Karakteristieken maatgevend duwstel**			
		Naam	Breedte	Lengte	Diepgang (geladen)	Laadvermogen	Breedte en lengte		Combinatie	Breedte	Lengte	
												m
	M0	Overig				1-250	B ≤ 5,00 of L ≤ 38,00					
I	M1	Spits	5,05	38,5	2,5	251-400	B = 5,01-5,10 en L >= 38,01	BO1	<input checked="" type="checkbox"/>	5,2	55	
II	M2	Kempenaar	6,6	50-55	2,6	401-650	B = 5,11-6,70 en L >= 38,01	BO2	<input checked="" type="checkbox"/>	6,6	60-70	
III	M3	Hagenaar	7,2	55-70	2,6	651-800	B = 6,71-7,30 en L >= 38,01	BO3	<input checked="" type="checkbox"/>	7,5	80	
	M4	Dortmund Eems (L <= 74 m)	8,2	67-73	2,7	801-1050	B = 7,31-8,30 en L = 38,01-74,00	BO4	<input checked="" type="checkbox"/>	8,2	85	
	M5	Verl. Dortmund Eems (L > 74 m)	8,2	80-85	2,7	1051-1250	B = 7,31-8,30 en L >= 74,01					
IVa	M6	Rijn-Herne Schip (L <= 86 m)	9,5	80-85	2,9	1251-1750	B = 8,31-9,60 en L = 38,01-86,00	BI	Europa I duwstel	9,5	85-105	<input checked="" type="checkbox"/>
	M7	Verl. Rijn-Herne (L > 86 m)	9,5	105	3,0	1751-2050	B = 8,31-9,60 en L >= 86,01					
IVb												
Va	M8	Groot Rijnschip (L <= 111 m)	11,4	110	3,5	2051-3300	B = 9,61-11,50 en L = 38,01-111,00	BII-1	Europa II duwstel	11,4	95-110	<input checked="" type="checkbox"/>
	M9	Verlengd Groot Rijnschip (L > 111 m)	11,4	135	3,5	3301-4000	B = 9,61-11,50 en L >= 111,01	BIIa-1	Europa IIa duwstel	11,4	92-110	<input checked="" type="checkbox"/>
								BIIIL-1	Europa II Lang	11,4	125-135	<input checked="" type="checkbox"/>
Vb								BII-2I	2-baksduwstel lang	11,4	170-190	<input checked="" type="checkbox"/>
Via	M10	Maatg. Schip 13,5 * 110 m	13,50	110	4,0	4001-4300	B = 11,51-14,30 en L = 38,01-111,00	BII-2b	2-baksduwstel breed	22,8	95-145	<input checked="" type="checkbox"/>
	M11	Maatg. Schip 14,2 * 135 m	14,20	135	4,0	4301-5600	B = 11,51-14,30 en L >= 111,01					
	M12	Rijnmax Schip	17,0	135	4,0	>= 5601	B >= 14,31 en L >= 38,01					
Vib								BII-4	4-baksduwstel (incl. 3-baks lang)	22,8	185-195	<input checked="" type="checkbox"/>
Vic								BII-6I	6-baksduwstel lang (incl. 5-baks lang)	22,8	270	<input checked="" type="checkbox"/>
VIIa								BII-6b	6-baksduwstel breed (incl. 5-baks breed)	34,2	195	<input checked="" type="checkbox"/>

Table A-1: CEMT Classes (Rijkswaterstaat, 2017)



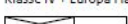




Duwstellen (Barges)			Koppverbanden (Convoys)						Doorvaarthoogte*	
Diepgang (geladen)	Classificatie		RWS Klasse	Karakteristieken maatgevend koppverband**			Classificatie		incl. 30 cm schrikhoogte	
	Laadvermogen	Breedte en lengte		Combinatie	Breedte	Lengte	Diepgang (geladen)	Laadvermogen		Breedte en lengte
m	t	m			m	m	m	t	m	
1,9	0-400	B<=5,20 en L=alle	C1i C1b	2 spitsen lang 	5,05	77-80	2,5	<= 900	B<= 5,1 en L=alle	5,25*
				2 spitsen breed 	10,1	38,5	2,5	<= 900	B=9,61-12,60 en L<=80,00	5,25*
2,6	401-600	B=5,21-6,70 en L=alle								6,1
2,6	601-800	B=6,71-7,60 en L=alle								6,4
2,7	801-1250	B=7,61-8,40 en L=alle								6,6
										6,4
3,0	1251-1800	B=8,41-9,60 en L=alle								7,0*
										7,0*
			C2i	Klasse IV + Europa I lang 	9,5	170-185	3,0	901-3350	B=5,11-9,60 en L=alle	7,0*
3,5	1801-2450	B=9,61-15,10 en L<=111,00								9,1*
4,0	2451-3200	B=9,61-15,10 en L<=111,00								9,1*
4,0	3201-3950	B=9,61-15,10 en L=111,01-146,00								9,1*
3,5-4,0	3951-7050	B=9,61-15,10 en L>=146,01	C3i	Klasse Va + Europa II lang 	11,4	170-190	3,5-4,0	3351-7250	B=9,61-12,60 en L>=80,01	9,1*
3,5-4,0	3951-7050	B=15,11-24,00 en L<=146,00	C2b	Klasse IV + Europa I breed 	19,0	85-105	3,0	901-3350	B=12,61-19,10 en L<=136,00	7,0* alleen voor klasse IV koppverband
			C3b	Klasse Va + Europa II breed 	22,8	95-110	3,5-4,0	3351-7250	B>19,10 en L<=136	9,1*
3,5-4,0	7051-12000 (7051-9000)	B=15,11-24,00 en L=146,01-200	C4	Klasse Va + 3 Europa II 	22,8	185	3,5-4,0	>=7251	B>12,60 en L>=136,01	9,1*
3,5-4,0	12001-18000 (12001-15000)	B=15,11-24,00 en L>=200,01								9,1*
3,5-4,0	12001-18000 (12001-15000)	B>=24,01 en L=alle								9,1*

Table A-2: CEMT Classes (Rijkswaterstaat, 2017)

Appendix B

Anchors



Type ¹⁾		Schroefblad ²⁾	Wanddikte	Staal kwaliteit	A_s / A_s ³⁾	$R_{d, \text{staal}}$	R_d ⁴⁾
diameter	wanddikte		[mm]		[mm]	[kN]	[kN]
[mm]	[mm]						
∅ 32,0	x 5,6	100-150 mm	5,6	E470	464	241	193
∅ 32,0	x 7,2	100-150 mm	7,2	E470	561	291	233
∅ 42,4	x 8,0	125-175 mm	8,0	E470	865	448	359
∅ 42,4	x 11,0	125-175 mm	11,0	E470	1.085	563	450
∅ 51,0	x 10,0	125-200 mm	10,0	E470	1.288	668	534
∅ 51,0	x 12,5	125-200 mm	12,5	E470	1.512	784	627
∅ 60,3	x 12,5	150-225 mm	12,5	E470	1.877	973	778
∅ 60,3	x 16,0	150-225 mm	16,0	E470	2.227	1.154	923
∅ 76,1	x 14,2	175-300 mm	14,2	E470	2.761	1.432	1.145
∅ 76,1	x 17,5	175-300 mm	17,5	E470	3.222	1.670	1.336
∅ 82,5	x 17,5	200-350 mm	17,5	E470	3.574	1.853	1.472
∅ 82,5	x 20,0	200-350 mm	20,0	E470	3.927	2.036	1.618
∅ 101,6	x 17,5	250-400 mm	17,5	E470	4.624	2.397	1.905
∅ 101,6	x 22,2	250-400 mm	22,2	E470	5.538	2.871	2.282
∅ 101,6	x 25,0	250-400 mm	25,0	E470	6.016	3.098	2.479

Figure B-1: Van Leeuwen Anchor Catalog

Appendix C

Environmental Classes

Omgevingseisen voor $c_{min,dur}$ mm							
Constructie- klasse	Milieuklasse volgens tabel 4.1						
	X0	XC1	XC2/XC3	XC4	XD1/XS1	XD2/XS2	XD3/XS3
S1	10	10	10	15	20	25	25
S2	10	10	15	20	25	30	30
S3	10	10	20	25	30	35	35
S4	10	15	25	30	35	40	40
S5	15	20	30	35	40	45	45
S6	20	25	35	40	45	50	50

Table C-1: Environmental concrete cover demands Netherlands (Normcommissie 351001, 2016)

Structural Class							
Criterion	Exposure Class according to Table 4.1						
	X0	XC1	XC2 / XC3	XC4	XD1	XD2 / XS1	XD3 / XS2 / XS3
Design Working Life of 100 years	increase class by 2	increase class by 2	increase class by 2	increase class by 2	increase class by 2	increase class by 2	increase class by 2
Strength Class ^{1) 2)}	≥ C30/37 reduce class by 1	≥ C30/37 reduce class by 1	≥ C35/45 reduce class by 1	≥ C40/50 reduce class by 1	≥ C40/50 reduce class by 1	≥ C40/50 reduce class by 1	≥ C45/55 reduce class by 1
Member with slab geometry (position of reinforcement not affected by construction process)	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1
Special Quality Control of the concrete production ensured	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1	reduce class by 1

Table C-2: Structural Exposure Classes Concrete Structures (British Standard Institution, 2008)

Class designation	Description of the environment	Informative examples where exposure classes may occur
1 No risk of corrosion or attack		
X0	For concrete without reinforcement or embedded metal: all exposures except where there is freeze/thaw, abrasion or chemical attack For concrete with reinforcement or embedded metal: very dry	Concrete inside buildings with very low air humidity
2 Corrosion induced by carbonation		
XC1	Dry or permanently wet	Concrete inside buildings with low air humidity Concrete permanently submerged in water
XC2	Wet, rarely dry	Concrete surfaces subject to long-term water contact Many foundations
XC3	Moderate humidity	Concrete inside buildings with moderate or high air humidity External concrete sheltered from rain
XC4	Cyclic wet and dry	Concrete surfaces subject to water contact, not within exposure class XC2
3 Corrosion induced by chlorides		
XD1	Moderate humidity	Concrete surfaces exposed to airborne chlorides
XD2	Wet, rarely dry	Swimming pools Concrete components exposed to industrial waters containing chlorides
XD3	Cyclic wet and dry	Parts of bridges exposed to spray containing chlorides Pavements Car park slabs
4 Corrosion induced by chlorides from sea water		
XS1	Exposed to airborne salt but not in direct contact with sea water	Structures near to or on the coast
XS2	Permanently submerged	Parts of marine structures
XS3	Tidal, splash and spray zones	Parts of marine structures
5. Freeze/Thaw Attack		
XF1	Moderate water saturation, without de-icing agent	Vertical concrete surfaces exposed to rain and freezing
XF2	Moderate water saturation, with de-icing agent	Vertical concrete surfaces of road structures exposed to freezing and airborne de-icing agents
XF3	High water saturation, without de-icing agents	Horizontal concrete surfaces exposed to rain and freezing
XF4	High water saturation with de-icing agents or sea water	Road and bridge decks exposed to de-icing agents Concrete surfaces exposed to direct spray containing de-icing agents and freezing Splash zone of marine structures exposed to freezing
6. Chemical attack		
XA1	Slightly aggressive chemical environment according to EN 206-1, Table 2	Natural soils and ground water
XA2	Moderately aggressive chemical environment according to EN 206-1, Table 2	Natural soils and ground water
XA3	Highly aggressive chemical environment according to EN 206-1, Table 2	Natural soils and ground water

Table C-3: Environmental Classes Structures (British Standard Institution, 2008)

Appendix D

Lock Gates

Stop logs The use of stop logs is the simplest type of gate and nowadays widely used to close of the lock during maintenance and construction. Stop logs can be lifted and stacked into recesses in the lock head to close off the lock. Figure D-1.

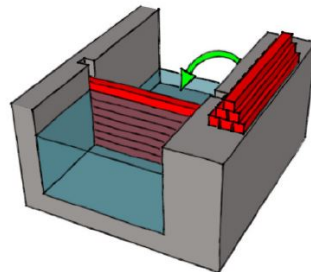


Figure D-1: Stop logs (Doeksen, 2012)

Lift Gate This type of gate is comparable with stop logs. In closed position the gate is in between the shafts in the lock head to which the loads are transmitted. To open, an expensive permanent super structure lifts the gate out of the water allowing ships to pass. This however, leads to a limited clearance for the ships to pass. An advantage is that it can retain negative water level differences. Also a smaller footprint is achieved due to the vertical storage of the gate (Vrijburcht et al., 2000). To level the water in the lock, openings in the gate can be present. Also the gates can be partly lifted to level the water, in this last case the gate is also qualified for controlled discharge (PIANC, 1986). Figure D-2.

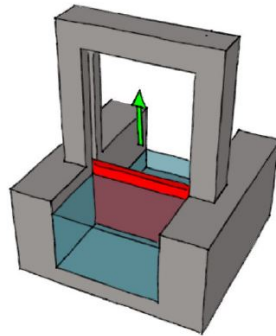


Figure D-2: Lift Gate (Doeksen, 2012)

Single Leaf Gate A Single Leaf Gate is a gate that rotates around the vertical axis. It is a simple gate which is used in comparatively narrow locks. Due to the rotation around the vertical axis it uses a large area of the lock chamber. When a locking device is present a limited negative head can be retained. Figure D-3.

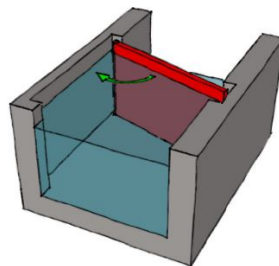


Figure D-3: Single Leaf Gate (Doeksen, 2012)

Mitre Gates Mitre Gates do like Single Leaf Gates rotate around the vertical axis. By using two gates opposite to each other a larger span can be obtained while occupying less room of the lock. Due to the arc shaped closing the forces of the water are better transferred to the lock heads and are able to retain high heads. When applying a locking device, a small negative head can be retained. However, this could lead to leakage. Figure D-4.

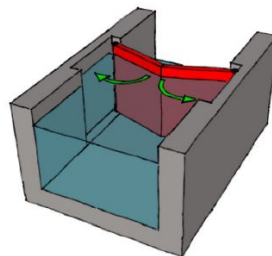


Figure D-4: Mitre Gates (Doeksen, 2012)

Radial Sector Gates - vertical axis These gates operated by regulation the water level differences between the lock and the gate chamber. They use the same kind of bearings as

Mitre Gates. Leveling of the lock takes place by opening the gates and thus opening and closure under free flow is possible. Vertical axis rotated Sector Gates are mostly used with relatively small lifts (PIANC, 1986). Figure D-5.

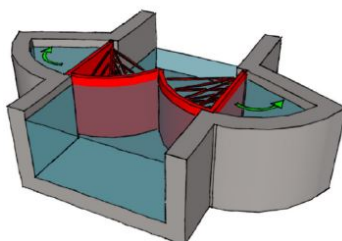


Figure D-5: Radial Sector Gates - vertical axis (Doeksen, 2012)

Radial Sector Gates - horizontal axis Radial Sector Gates can also rotate around the horizontal axis. This type of gate is often used in weirs and spillways and could have relatively large widths. The circular arch shape contributes to an efficient transfer of the loads. In an open condition the gate rotates to a underwater sill. When maintenance is needed the gate can be rotated above the water. Due to these cases no recesses in the lock head are needed (PIANC, 2006). In locks these Sector Gates are mostly used in the upper head (PIANC, 1986). Figure D-6.

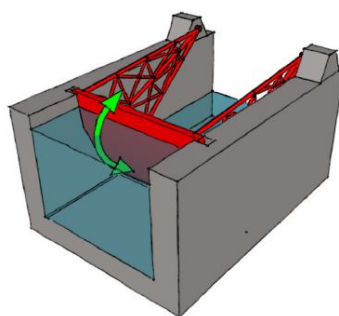


Figure D-6: Radial Sector Gates - horizontal axis (Doeksen, 2012)

Flap Gate Similar to the previous type, Flap Gates rotate along the horizontal axis. In an open position the gate is stored in the sill of the lock head. Seal and bearing designs are simple but the floor mounted flap gates frequently present difficulties due to dirt accumulation (PIANC, 1986). With a bolt lock the flap gate is suitable for bilateral retention (Vrijburcht et al., 2000). Figure D-7.

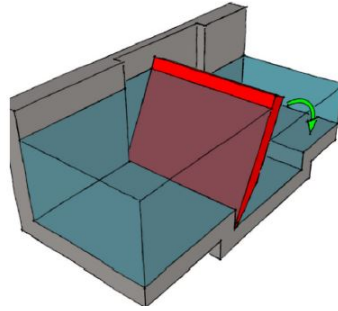


Figure D-7: Flap Gate (Doeksen, 2012)

Rolling Gate The Rolling Gate is a horizontal translation gate. In an open position the gate is stored in a recess in the lock head. To close the lock the gate will roll over a guiding rail into the lock. This type of gate can resist negative water level differences and can be used at locks with large widths. Two major disadvantages occur with the use of this gate. Firstly, a lot of space is needed to store the gate in the open position, and secondly numerous moving parts are present under water due to the movement on the rail track. Figure D-8.

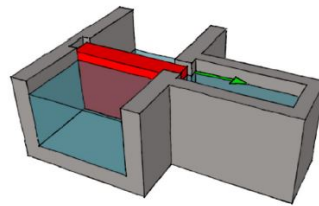


Figure D-8: Rolling or Sliding Gate (Doeksen, 2012)

Sliding Gate To overcome the disadvantage of the moving parts regarding the Rolling Gate a Sliding Gate has been developed. Instead of wheels on a rail track it makes use of hydraulic slide bearings to open or close (Vrijburcht et al., 2000). Figure D-8.

Appendix E

Profiles Sheetpile

Table E-1: Sheet Pile Profiles ArcelorMittal (ArcelorMittal, 2019)

Section	b	h	t	s	A	Gsp	Gw	Iy	Wey,y	Sy	Wpl,y
	mm	mm	mm	mm	cm ² /m	kg/m	kg/m ²	cm ⁴ /m	cm ³ /m	cm ³ /m	cm ³ /m
AZ 13-700	700	315	9.5	9.5	135	74.0	106.0	20540	1305	770	1540
AZ 14-700	700	316	10.5	10.5	146	80.3	115.0	22190	1405	835	1665
AZ 17-700	700	420	8.5	8.5	133	73.1	104.0	36230	1730	1015	2027
AZ 18-700	700	420	9.0	9.0	139	76.5	109.0	37800	1800	1060	2116
AZ 20-700	700	421	10.0	10.0	152	83.5	119.0	40960	1945	1150	2296
AZ 24-700	700	459	11.2	11.2	174	95.7	137.0	55820	2430	1435	2867
AZ 28-700	700	461	13.2	13.2	200	110.0	157.0	63620	2760	1635	3273
AZ 48	580	481	18.0	14.0	291	132.6	229.0	110450	4595	2650	5295

Table E-2: HZ Pile Profiles ArcelorMittal (ArcelorMittal, 2019)

Section	h	h1	b	tmax	t	s	r	Torsional con- stant	Warping con- stant	Sectional area	Mass	Moment of in- ertia	Elastic sec- tion mod- ulus	Coating area
	mm	mm	mm	mm	mm	mm	mm	cm ⁴	10 ³ cm ⁶	cm ²	kg/m	y-y cm ⁴	y-y cm ³	m ² /m
HZ880M A	831.3	803.4	458	29.0	18.9	13.0	20	347.2	58780	292.4	229.5	351350	8650	3.443
HZ1080M	1075.3	1047.4	454	29.0	19.6	16.0	30	525.9	98560	368.7	289.4	690560	13075	3.877
A HZ1180M A	1075.4		458	34.7	31.0	20.0	30	1352.9	124600	494.9	388.5	967270	17865	3.884

Appendix F

Python Code

```
def ship_dimensions():
    # Motor Ship Classes
    MVC = masterfile.Value['Motor Vessel Class']
    FCC = masterfile.Value['Fueled Convoy Class']
    CPC = masterfile.Value['Coupled Formation Class']

    if MVC == 'CGMT 1':
        Wship = 5.05
        Lship = 38.5
        Dloadship = 2.5
        AirDraughtship = 4.25
        Bollard_forcecship = 100
    elif MVC == 'CGMT 1P':
        Wship = 6.6
        Lship = 55
        Dloadship = 2.6
        AirDraughtship = 5.25
        Bollard_forcecship = 100
    elif MVC == 'CGMT 1V':
        Wship = 8.2
        Lship = 85
        Dloadship = 2.7
        AirDraughtship = 5.35
        Bollard_forcecship = 150
    elif MVC == 'CGMT 2V':
        Wship = 9.5
        Lship = 100
        Dloadship = 3
        AirDraughtship = 5.55
        Bollard_forcecship = 150
    elif MVC == 'CGMT 3V':
        Wship = 11.4
        Lship = 110
        Dloadship = 3.5
        AirDraughtship = 6.4
        Bollard_forcecship = 200
    elif MVC == 'CGMT 3VIA':
        Wship = 17
        Lship = 135
        Dloadship = 4
        AirDraughtship = 9.75
        Bollard_forcecship = 300
    elif MVC == 'Unknown':
        Wship = 0
        Lship = 0
        Dloadship = 0
        AirDraughtship = 0
        Bollard_forcecship = 0
    # Fuel Classes
    if FCC == 'CGMT 1':
        Wship = 5.2
        Lship = 55
        Dloadship = 1.9
        AirDraughtship = "not known"
        Bollard_forcecship = 100
    elif FCC == 'CGMT 1P':
        Wship = 6.6
        Lship = 70
        Dloadship = 2.6
        AirDraughtship = "not known"
        Bollard_forcecship = 100
    elif FCC == 'CGMT 1V':
        Wship = 8.2
        Lship = 85
        Dloadship = 2.9
        AirDraughtship = "not known"
        Bollard_forcecship = 150
    elif FCC == 'CGMT 2V':
        Wship = 9.5
        Lship = 100
        Dloadship = 3
        AirDraughtship = "not known"
        Bollard_forcecship = 150
    elif FCC == 'CGMT 3V':
        Wship = 11.4
        Lship = 110
        Dloadship = 4
        AirDraughtship = "not known"
        Bollard_forcecship = 200
    elif FCC == 'CGMT 3VIA':
        Wship = 17
        Lship = 135
        Dloadship = 4
        AirDraughtship = "not known"
        Bollard_forcecship = 300
    elif FCC == 'Unknown':
        Wship = 0
        Lship = 0
        Dloadship = 0
        AirDraughtship = 0
        Bollard_forcecship = 0
    # Coupled Formation Classes
    if CPC == 'CGMT 1(long)':
        Wship = 5.05
        Lship = 80
        Dloadship = 2.5
        AirDraughtship = "not known"
        Bollard_forcecship = 100
    elif CPC == 'CGMT 1(wide)':
        Wship = 18.1
        Lship = 38.5
        Dloadship = 2.5
        AirDraughtship = "not known"
        Bollard_forcecship = 100
    elif CPC == 'CGMT 1V':
        Wship = 11.4
        Lship = 130
        Dloadship = 4
        AirDraughtship = "not known"
        Bollard_forcecship = 200
    elif CPC == 'CGMT 2V':
        Wship = 22.8
        Lship = 185
        Dloadship = 4
        AirDraughtship = "not known"
        Bollard_forcecship = 300
    elif CPC == 'Unknown':
        Wship = 0
        Lship = 0
        Dloadship = 0
        AirDraughtship = "not known"
        Bollard_forcecship = 0

    # Calculating dimensions
    Lship = max(Lship, Lship, Lship, masterfile.Value['Length (m)'])
    Wship = max(Wship, Wship, Wship, masterfile.Value['Width (m)'])
    Dloadship = max(Dloadship, Dloadship, Dloadship, masterfile.Value['Draught (m)'])
    Bollard_forcecship = max(Bollard_forcecship, Bollard_forcecship, Bollard_forcecship,
                             masterfile.Value['Maximum bollard force (kN) (Motor Vessel 180)'])
    AirDraughtship = max(AirDraughtship, AirDraughtship, AirDraughtship)
    print("Normative dimensions ship")
    print("Normative Width is ", Wship, "m")
    print("Normative Length is ", Lship, "m")
    print("Normative Draught of a Loadship is ", Dloadship, "m")
    return (Lship, Wship, Dloadship, Bollard_forcecship, MVC, FCC, CPC)
```

Figure F-1: Code Ship Dimensions


```

def tension_piles(bottom_level_of_structure, netto_vertical_force, width_floor):
    Deq = 1.1 * B_pile / 1000 # equivalent Diameter in m
    begin_pile = Bottom_level_of_structure
    end_pile = begin_pile - Length_tension_pile

    if type_pile == "Ingeheide glasde prefab betonpaal":
        alpha_t_sand = 0.007
    elif type_pile == "Stalen buispaal met gesloten punt":
        alpha_t_sand = 0.007
    elif type_pile == "In de grond gemaakte paal, waarbij de betonkolom direct tegen de grond drukt en de buis terugtrekkend uit de grond is verwijderd":
        alpha_t_sand = 0.012
    elif type_pile == "In de grond gemaakte paal, waarbij de betonkolom direct tegen de grond drukt en de buis trillend is verwijderd":
        alpha_t_sand = 0.010
    elif type_pile == "heesend ingebracht MV paalen":
        alpha_t_sand = 0.012
    elif type_pile == "Lingschroefde paalen met groutinjectie of menging":
        alpha_t_sand = 0.009
    elif type_pile == "Stalen profielen geheel (inclusief open stalen buispaalen en damwanden)":
        alpha_t_sand = 0.004
    elif type_pile == "Massaarmen of avegaarspaalen":
        alpha_t_sand = 0.0045

    # Hier wordt alles geloopt
    depth = -60
    steps = -(depth * 10) + 1
    layer = [number of layers] * steps
    name_soil = np.zeros(steps, dtype=object)
    Gamma_dry_soil = np.zeros(steps)
    Gamma_wet_soil = np.zeros(steps)
    Cohesion_soil = np.zeros(steps)
    Phi_soil = np.zeros(steps)
    cone_resistance = np.zeros(steps)
    OCR = np.zeros(steps)
    gamma_representative = np.zeros(steps)
    alpha_t = np.zeros(steps)

    for i in range(steps): # 40m is assumed to be deep enough (nag niet onderbouwd)
        z = np.round(np.linspace(Top of structure tov NAP, depth + Top of structure tov NAP, steps), 10)

        if z[i] >= (toplevel_layer0): # default van niet gebruikte lagen moeten op -100 worden gezet.
            if z[i] >= (toplevel_layer9):
                if z[i] >= (toplevel_layer8):
                    if z[i] >= (toplevel_layer7):
                        if z[i] >= (toplevel_layer6):
                            if z[i] >= (toplevel_layer5):
                                if z[i] >= (toplevel_layer4):
                                    if z[i] >= (toplevel_layer3):
                                        if z[i] >= (toplevel_layer2):
                                            if z[i] >= (toplevel_layer1):
                                                layer[i] = 0
                                            else:
                                                layer[i] = 1
                                        else:
                                            layer[i] = 2
                                    else:
                                        layer[i] = 3
                                else:
                                    layer[i] = 4
                            else:
                                layer[i] = 5
                        else:
                            layer[i] = 6
                    else:
                        layer[i] = 7
                else:
                    layer[i] = 8
            else:
                layer[i] = 9

        else:
            layer[i] = 10

        if layer[i] < number of layers:
            Gamma_dry_soil[i] = layers.input Gamma Dry soil[layer[i]]
            Gamma_wet_soil[i] = layers.input Gamma Wet soil[layer[i]]
            name_soil[i] = layers.name_layer[layer[i]]
            Cohesion_soil[i] = layers.input Cohesion soil[layer[i]]
            Phi_soil[i] = layers.input Phi soil[layer[i]]
            cone_resistance[i] = layers.cone_resistance[layer[i]]
            OCR[i] = layers.OCR[layer[i]]

        else:
            Gamma_dry_soil[i] = layers.input Gamma Dry soil[layer[i]].iloc[0]
            Gamma_wet_soil[i] = layers.input Gamma Wet soil[layer[i]].iloc[0]
            name_soil[i] = layers.name_layer[layer[i]].iloc[0]
            Cohesion_soil[i] = layers.input Cohesion soil[layer[i]].iloc[0]
            Phi_soil[i] = layers.input Phi soil[layer[i]].iloc[0]
            cone_resistance[i] = layers.cone_resistance[layer[i]].iloc[0]
            OCR[i] = layers.OCR[layer[i]].iloc[0]

        if z[i] >= waterlevel:
            gamma_representative[i] = Gamma_dry_soil[i]
        else:
            gamma_representative[i] = Gamma_wet_soil[i]

        if name_soil[i] == "Sand":
            alpha_t[i] = alpha_t_sand
        elif name_soil[i] == "Peat" or "Clay":
            if cone_resistance[i] > 1:
                alpha_t[i] = 0.025
            else:
                if (z[i] / Deq) > 20:
                    alpha_t[i] = 0.025
                else:
                    alpha_t[i] = 0.020

    Values_per_depth = pd.DataFrame({"z": z, "Layer": layer, "name_soil": name_soil,
    "Gamma dry soil": Gamma_dry_soil, "Gamma wet soil": Gamma_wet_soil, "Cohesion soil": Cohesion_soil,
    "Phi soil": Phi_soil, "cone_resistance": cone_resistance,
    "OCR": OCR, "gamma_representative": gamma_representative, "alpha t": alpha_t})
    Values_per_depth.set_index("z", inplace=True)

    q_cNc = np.zeros(steps)
    q_czd = np.zeros(steps)
    q_cz = np.zeros(steps)
    sigma_v20 = np.zeros(steps)
    sigma_v20tr = np.zeros(steps)
    q_c20tr = np.zeros(steps)

    ##### HERE THE STEPS OF CUR 2001-4 ARE CALCULATED #####
    for i in range(steps):
        # stap 1
        q_c20C = cone_resistance # gemeten conusweerstand in op grond (MPa)
        q_cNc[i] = q_c20C[i] * np.sqrt(1 / OCR[i]) # voor overconsolidatie gereduceerde conusweerstand (MPa)

        # stap 2
        q_cz[i] = q_cNc[i] # conusweerstand op diepte z (MPa)

        if z[i] >= waterlevel:
            sigma_v20[i] = gamma_representative[i] * 0.1 + sigma_v20[
            i - 1] # initiële effectieve spanning op diepte z (kPa)
        else:
            sigma_v20[i] = gamma_representative[i] * 0.1 + sigma_v20[
            i - 1] - unit_weight_of_water * 0.1 # initiële effectieve spanning op diepte z

        delta_e3 = np.zeros(steps)
        delta_e = delta_e1 + delta_e2 + delta_e3
        delta_Re = (delta_e) / (e_max - e_min)
        f1 = np.exp(0 + delta_Re)
        q_cz1 = f1 * q_czd # deze is volgens mij niet nodig

        ## STAP 5
        A = (h_o_h_afstand_lange[k] * h_o_h_afstand_korte[k]) - (
        (B_pile / 1000) * (B_pile / 1000)) # invloedsoppervlak van de paal in m2
        volgens 7.3.4 en 7.3.5
        if A < 0:
            break

        d1 = 0.1 # zie alle lagen als 0,1 m. Dit is eerste plan, moet later veranderd misschien
        veranderd worden om het als een soort laag te laten figureren.
        M1 = np.zeros(steps)
        gamma_dj = np.zeros(steps) # effectief volumiek gewicht van de laag i in kN/m3
        sigma_vdj0 = np.zeros(steps)
        T_d1 = np.zeros(steps)
        T_dn = np.zeros(steps) # dit is de som van de T n van de boven gelegen lagen
        F2 = np.zeros(steps)
        sandpresent = False
        for i in range(int(begin_pile - end_pile) / 10):
            if "Sand" in Values_per_depth.name_soil[round(begin_pile - i / 10, 1)]:
                sandpresent = True
            if sandpresent == True:
                for i in range(steps):
                    if name_soil[i] == "Sand":
                        alpha_t[i] = alpha_t[i]
                    else:
                        alpha_t[i] = 0.5 * alpha_t[i]
                    if name_soil[i] == "Clay": # zie 8.2, voor klei: f1 en f2 = 1
                        f1[i] = 1
                    if name_soil[i] == "Peat":
                        f1[i] = 1
                        O_pi = 4 * B_pile / 1000 # onttrek paal in m
                    if z[i] < Bottom_level_of_structure:
                        gamma_mg = 1.1
                    else:
                        gamma_mg = 1.0
                    if z[i] >= Bottom_level_of_structure:
                        M1[i] = 0
                        gamma_dj[i] = 0
                    else:
                        M1[i] = (f1[i] * 0_pi * alpha_t[i] * q_czd[i] * 1000 * (d1)) / A
                        if z[i] >= waterlevel:
                            gamma_dj[i] = gamma_representative[i] / gamma_mg
                        else:
                            gamma_dj[i] = gamma_representative[
                            i] / gamma_mg - unit_weight_of_water # initiële
                            effectieve spanning op diepte z (kPa)
                    if z[i] >= Bottom_level_of_structure:
                        sigma_vdj0[i] = 0
                    else:
                        sigma_vdj0[i] = gamma_dj[i] * 0.1 + sigma_vdj0[
                        i - 1] # check of deze klopt, snap er geen kut van (formule 94) denk nu
                        dat ie wel klopt signallijk

                    if name_soil[i] == "Clay":
                        f2[i] = 1
                    elif name_soil[i] == "Peat":
                        f2[i] = 1
                    elif sigma_vdj0[i] == 0: # Goefft error als de deler nul is in f2, daarom zet je
                    f2 op 1
                        f2[i] = 1

```

Figure F-4: Code Tension Piles


```

def
underwater_concrete_floor(structure,range_pile_length,range_profile,vertical_force,excavation_wid
th,h_o_h_afstand_lange,h_o_h_afstand_korte):
    first_time_structure = True
    H_underwaterconcretefloor_start = 0.8*1000 #m
    steps_UWC_floor = np.linspace(0,4000,41)
    H_underwaterconcretefloor=[H_underwaterconcretefloor_start]*(len(steps_UWC_floor))
    Inv.range('E6').value = [size_of_DWC]
    Inv.range('F6').value = [unit_weight_of_water]
    if structure == 'u_basin':
        Inv.range('B15').value = [Top_of_structure_tov_NAP - Height - floor_height_U_basin_max]
    else:
        if Dewatering_chamber == 'Yes':
            Inv.range('B15').value = [Top_of_structure_tov_NAP-Height]
            Inv.range('B15').value = [-15]
        else:
            Inv.range('B15').value = [Top_of_structure_tov_NAP - Height + (Lowest_Lock_Level - 1
            - (Top_of_structure_tov_NAP - Height))]
        Inv.range('B16').value = [waterlevel1]
        Inv.range('B19').value = [tolerance_upside_underwater_concretefloor*1000]
        Inv.range('B19').value = [tolerance_downside_underwater_concretefloor*1000]
        Inv.range('F31').value = [B_pile]
        #####
    for j in range(range_pile_length):
        for l in range(range_number_of_anchor):
            for a in range(range_profile):
                if structure == 'sheetpile':
                    run = ('D-Sheetoutput'+ str(j) + '|' + str(l) + '|' + str(s) + '|')
                    uwc_check = (results.Dsheet2.Strength_check_sheetpile(run))
                    uwc_name_of_run = Name_of_run
                    uwc_normal_force =
                    results.Dsheet2.Force_in_Underwaterconcrete_by_sheetpile(run)
                    elif structure == 'diaphragm':
                        run = ('D-SheetoutputD'+ str(j) + '|' + str(l) + '|' + str(s) + '|')
                        uwc_check = (results.Dsheet2.d.Strength_check_diaphragm_wall(run))
                        uwc_name_of_run = Name_of_run_d
                        uwc_normal_force =
                        results.Dsheet2_d.Force_in_Underwaterconcrete_by_diaphragm_wall(run)
                    elif structure == 'u_basin':
                        run = ('D-SheetoutputU basin'+ str(j) + '|' + str(l) + '|' + str(s) +
                        '|')
                        uwc_check = (results.Dsheet2_u.Strength_check_sheetpile_u(run))
                        uwc_name_of_run = Name_of_run_u
                        uwc_normal_force =
                        results.Dsheet2_u.Force_in_Underwaterconcrete_by_sheetpile_u(run)
                if run in uwc_name_of_run:
                    if vertical_force < 0:
                        Inv.range('E23').value = [h_o_h_afstand_lange]
                        stop = False
                    if uwc_check == 'Check':
                        if first_time_structure == True:
                            first_time_structure = False
                        for k in range(25):
                            if stop == True:
                                break
                            Inv.range('G46').value = [h_o_h_afstand_korte / 2]
                            if k < (excavation_width / h_o_h_afstand_korte):
                                Inv.range('G' + str(46 + k)).value = [k_axial_piles]
                                elif k > 24:
                                    Inv.range('B' + str(46 + k)).value = [0]
                            if k > 0:
                                if k < (excavation_width / h_o_h_afstand_korte):
                                    Inv.range('G' + str(46 + k)).value =
                                    [h_o_h_afstand_korte]
                                if k + 1 == int(excavation_width / h_o_h_afstand_korte):
                                    Inv.range('G' + str(46 + k)).value =
                                    [h_o_h_afstand_korte]
                                else:
                                    Inv.range('G' + str(46 + k + 1)).value =
                                    stop = True
                                for i in range(25):
                                    if k + i < 25:
                                        Inv.range('G' + str(46 + k + 1 + i)).value =
                                        else:
                                            break
                                if uwc_normal_force > 0:
                                    Inv.range('F20').value = [uwc_normal_force]
                                else:
                                    plusmin = 'plus'
                                    H_underwaterconcretefloor_end = 0
                                    for i in range(len(steps_UWC_floor)):
                                        if plusmin == 'plus':
                                            H_underwaterconcretefloor[i] =
                                            H_underwaterconcretefloor_start + steps_UWC_floor[i]
                                        else:
                                            H_underwaterconcretefloor[i] =
                                            H_underwaterconcretefloor_start
                                            - steps_UWC_floor[i]
                                        Inv.range('B1').value = [H_underwaterconcretefloor[i]]
                                        if not os.path.exists('CUR_Aanbeveling_77_Underwaterbeton_versie_5.4_19_10_2018.xlsx'):
                                            xl = win32com.client.Dispatch('Excel.Application')
                                            xl.Workbooks.Open(os.path.abspath('CUR_Aanbeveling_77_Underwaterbeton_versie_5.4_19_10_2018.xlsx
                                            '),readOnly=1)
                                            xl.Application.Run("Berekenen.Bereken")
                                            ## xl.Application.Save() # if you want to save then uncomment
                                            this line and change delete the ", ReadOnly=1" part from the open function.
                                            xl.DisplayAlerts = False # werkt nog niet
                                        if (res.range('F8').value) < 1 and (res.range('G8').value) < 1
                                        and (res.range('H8').value) < 1 and (res.range('I8').value) < 1:
                                            print('Longitudinal direction (A) check')
                                        else:
                                            print('Longitudinal direction (A) failed')
                                            break
                                        if (res.range('F14').value) == 'Valdoet' and
                                        (res.range('G14').value) == 'Valdoet' and (res.range('H14').value) == 'Valdoet' and
                                        (res.range('I14').value) == 'Valdoet':
                                            print('min' H_underwaterconcretefloor[i])
                                        if (res.range('F20').value) == 'Valdoet' and
                                        (res.range('G20').value) == 'Valdoet' and (res.range('H20').value) == 'Valdoet' and
                                        (res.range('I20').value) == 'Valdoet':
                                            plusmin = 'min'
                                            if H_underwaterconcretefloor[i] >
                                            H_underwaterconcretefloor[i-1]:
                                                H_underwaterconcretefloor_end =
                                                H_underwaterconcretefloor_start =
                                                H_underwaterconcretefloor_end
                                                break
                                            elif H_underwaterconcretefloor[i] == 800:
                                                H_underwaterconcretefloor_end =
                                                H_underwaterconcretefloor_start =
                                                H_underwaterconcretefloor_end
                                                break
                                            else:
                                                plusmin = 'plus'
                                        else:
                                            if H_underwaterconcretefloor[i] < H_underwaterconcretefloor
                                            [i-1]:
                                                H_underwaterconcretefloor_end = H_underwaterconcretefloor
                                                [i-1]
            H_underwaterconcretefloor_start =
            H_underwaterconcretefloor_end =
            break
            plusmin = 'plus'
            if H_underwaterconcretefloor_end == 800:
                H_underwaterconcretefloor_start =
                break
            if structure == 'sheetpile':
                results.Dsheet2.at(run, 'df_H_underwaterconcretefloor_all') =
                H_underwaterconcretefloor_end
            elif structure == 'diaphragm':
                results.Dsheet2_d.at[
                run, 'df_H_Underwaterconcretefloor_all'] =
                H_underwaterconcretefloor_end
            elif structure == 'u_basin':
                results.Dsheet2_u.at(run, 'df_H_underwaterconcretefloor_all') =
                H_underwaterconcretefloor_end
            else:
                if structure == 'sheetpile':
                    results.Dsheet2.at(run, 'df_H_underwaterconcretefloor_all') = 0
                elif structure == 'diaphragm':
                    results.Dsheet2_d.at(run, 'df_H_underwaterconcretefloor_all') = 0
                elif structure == 'u_basin':
                    results.Dsheet2_u.at(run, 'df_H_underwaterconcretefloor_all') = 0
                print('fail')
            else:
                if structure == 'sheetpile':
                    results.Dsheet2.at(run, 'df_H_underwaterconcretefloor_all') = 800
                elif structure == 'diaphragm':
                    results.Dsheet2_d.at(run, 'df_H_underwaterconcretefloor_all') = 800
                elif structure == 'u_basin':
                    results.Dsheet2_u.at(run, 'df_H_underwaterconcretefloor_all') = 800
            if structure == 'sheetpile':
                output_uwc = results.Dsheet2.df.H_underwaterconcretefloor_all
            elif structure == 'diaphragm':
                output_uwc = results.Dsheet2_d.df.H_underwaterconcretefloor_all
            elif structure == 'u_basin':
                output_uwc = results.Dsheet2_u.df.H_underwaterconcretefloor_all
            return (output_uwc)

```

Figure F-5: Code UWC Floor

```

# Sheet Piles
for j in range(range_length_pile):
    for i in range(range_number_of_anchor):
        for s in range(range_sheetpile):
            run = ('D-Sheetoutput(' + str(j) + '|' + str(i) + '|' + str(s) + '|')
            if run in Name_of_run:
                if (results_Daheet2.Strength_check_sheetpile[run]) == 'Check':
                    # Width
                    width_keep = max(200,500-int(Sheetpile_height[s])) +
                    max(int(Sheetpile_height[s]),float(CombinedWallPilesDiameter[s])/1000) + 200 # (mm)

                    #moments
                    # AA = AA + 1
                    ground_moment_max = 1 / 8 * distance_between_anchors *
                    distance_between_anchors*max(results_Daheet2.anchor_force_1_stage_3[run] +
                    results_Daheet2.anchor_force_2_stage_3[run],results_Daheet2.anchor_force_1_stage_4[run] +
                    results_Daheet2.anchor_force_2_stage_4[run])
                    bollard_moment_max = 1 / 4 * max(Bollard_forceшип, Bollard_load_seagoin) *
                    max(distance_between_bollards_seagoin, distance_between_bollards_inland)
                    Maximum_moment_keep = ground_moment_max + bollard_moment_max

                    height_keep = 1000
                    effective_height_keep = (max(200,500-int(Sheetpile_height[s])) +
                    int(Sheetpile_height[s]) + 200) - 60 - 0.5 * reinforcement_diameter_keep - reinforcement_stirrups
                    A_reinforcement_keep = maximum_moment_keep * (1000000) /
                    steel * 0.9 * effective_height_keep
                    number_of_bars_keep = math.ceil(A_reinforcement_keep /
                    ((reinforcement_diameter_keep ** 2) * 0.25 * np.pi)) + 1

                    print (width_keep,'width sheet keep')

                    #concrete volume
                    volume_concrete_in_keep = (width_keep/1000 * height_keep/1000 +
                    (Top_of_structure_tov_NAP - Lowest_Lock_Level - height_keep/1000) * (max(200,500-
                    int(Sheetpile_height[s])) + 0.5 *
                    max(int(Sheetpile_height[s]),float(CombinedWallPilesDiameter[s])/1000)) /1000) * 2 * Length
                    concrete_costs_in_keep = volume_concrete_in_keep *
                    costs_material_and_casting_concrete
                    MKI_Keep_concrete = volume_concrete_in_keep * MKI_value_future['Concrete
                    C35/45 (CEMIII)']
                    CO2_tax_Keep_concrete = volume_concrete_in_keep * MKI_CO2_future['Concrete
                    C35/45 (CEMIII)']
                    results_Daheet2.at[run, 'df volume concrete in keep'] =
                    volume_concrete_in_keep
                    results_Daheet2.at[run, 'df concrete costs in keep'] = concrete_costs_in_keep
                    results_Daheet2.at[run, 'df MKI_Keep_concrete'] = MKI_Keep_concrete
                    results_Daheet2.at[run, 'df CO2_tax_Keep_concrete'] = CO2_tax_Keep_concrete

                    #Reinforcement volume
                    volume_reinforcement_in_Keep = number_of_bars_Keep *
                    ((reinforcement_diameter_Keep ** 2) * 0.25 * np.pi)/1000000 * 2 * Length
                    costs_reinforcement_in_Keep = volume_reinforcement_in_Keep * 7850 *
                    costs_material_and_placing_reinforcement
                    MKI_Keep_reinforcement = volume_reinforcement_in_Keep * 7850 *
                    MKI_value_future['Reinforcement_steel'] / 1000
                    CO2_tax_Keep_reinforcement = volume_reinforcement_in_Keep * 7850 *
                    MKI_CO2_future['Reinforcement_steel'] / 1000
                    results_Daheet2.at[run, 'df volume reinforcement in keep'] =
                    volume_reinforcement_in_Keep
                    results_Daheet2.at[run, 'df costs_reinforcement_in_Keep'] =
                    costs_reinforcement_in_Keep
                    results_Daheet2.at[run, 'df MKI_Keep_reinforcement'] = MKI_Keep_reinforcement
                    results_Daheet2.at[run, 'df CO2_tax_Keep_reinforcement'] =
                    CO2_tax_Keep_reinforcement
                    #formwork
                    #print (width_Keep)
                    amount_formwork_in_Keep = (width_Keep/1000 + 2 * height_Keep/1000 +
                    (Top_of_structure_tov_NAP - Lowest_Lock_Level - height_Keep/1000) * 2 * Length + Bottom +
                    #structural part_Keep + 1 * cover_Part)

                    costs_formwork_in_Keep = (amount_formwork_in_Keep) *
                    costs_material_and_placing_formwork
                    MKI_Keep_formwork = amount_formwork_in_Keep *
                    MKI_value_future['Traditional_formwork']
                    CO2_tax_Keep_formwork = amount_formwork_in_Keep *
                    MKI_CO2_future['Traditional_formwork']
                    results_Daheet2.d.at[run, 'df amount_formwork_in_Keep'] =
                    amount_formwork_in_Keep
                    results_Daheet2.d.at[run, 'df costs_formwork_in_Keep'] =
                    costs_formwork_in_Keep
                    results_Daheet2.d.at[run, 'df MKI_Keep_formwork_d'] = MKI_Keep_formwork_d
                    results_Daheet2.d.at[run, 'df CO2_tax_Keep_formwork_d'] =
                    CO2_tax_Keep_formwork_d
                    else:
                        # concrete volume
                        results_Daheet2.d.at[run, 'df volume concrete in keep_d'] = 0
                        results_Daheet2.d.at[run, 'df concrete_costs_in_Keep_d'] = 0
                        results_Daheet2.d.at[run, 'df MKI_Keep_concrete_d'] = 0
                        results_Daheet2.d.at[run, 'df CO2_tax_Keep_concrete_d'] = 0
                        # Reinforcement volume
                        results_Daheet2.d.at[run, 'df volume_reinforcement_in_Keep_d'] = 0
                        results_Daheet2.d.at[run, 'df costs_reinforcement_in_Keep_d'] = 0
                        results_Daheet2.d.at[run, 'df MKI_Keep_reinforcement_d'] = 0
                        results_Daheet2.d.at[run, 'df CO2_tax_Keep_reinforcement_d'] = 0
                        # formwork
                        results_Daheet2.d.at[run, 'df amount_formwork_in_Keep_d'] = 0
                        results_Daheet2.d.at[run, 'df costs_formwork_in_Keep_d'] = 0
                        results_Daheet2.d.at[run, 'df MKI_Keep_formwork_d'] = 0
                        results_Daheet2.d.at[run, 'df CO2_tax_Keep_formwork_d'] = 0

# Diaphragm Wall
for dd in range(range_length_diaphragm_wall):
    for ddd in range(range_number_of_anchor_diaphragm_wall):
        for dddd in range(range_thickness_diaphragm_wall):
            run = ('D-SheetoutputD(' + str(dd) + '|' + str(ddd) + '|' + str(ddd) + '|')
            if run in Name_of_run_d:
                if (results_Daheet2.d.Strength_check_diaphragm_wall[run]) == 'Check':
                    # Width
                    width_Keep = 650 + int(thickness_diaphragm_wall(ddd)) # (mm) #misachien nog
                    plus de 200mm

                    #moments
                    # AA = AA + 1
                    ground_moment_max = 1 / 8 * distance_between_anchors*distance_between_anchors
                    *max(results_Daheet2.d.anchor_force_1_stage_3_d[run] +
                    results_Daheet2.d.anchor_force_2_stage_3_d[run],results_Daheet2.d.anchor_force_1_stage_4_d[run] +
                    results_Daheet2.d.anchor_force_2_stage_4_d[run])
                    bollard_moment_max = 1 / 4 * max(Bollard_forceшип, Bollard_load_seagoin) *
                    max(distance_between_bollards_seagoin, distance_between_bollards_inland)
                    Maximum_moment_Keep = ground_moment_max + bollard_moment_max
                    height_Keep = 1000
                    effective_height_Keep = width_Keep - 60 - 0.5 * reinforcement_diameter_Keep *
                    reinforcement_stirrups
                    A_reinforcement_Keep = maximum_moment_Keep * (1000000) /
                    (f_yk_reinforcement_steel * 0.9 * effective_height_Keep)
                    number_of_bars_Keep = math.ceil(A_reinforcement_Keep /
                    ((reinforcement_diameter_Keep ** 2) * 0.25 * np.pi))

                    print (width_Keep, 'width diaphragm ceaps')
                    #concrete volume
                    volume_concrete_in_Keep_d = ((Top_of_structure_tov_NAP - Lowest_Lock_Level -
                    height_Keep/1000) * (width_Keep-int(thickness_diaphragm_wall(ddd)))/1000 + height_Keep/1000 *
                    width_Keep/1000) * (2 * Length)
                    concrete_costs_in_Keep_d = volume_concrete_in_Keep_d *
                    costs_material_and_casting_concrete
                    MKI_Keep_concrete_d = volume_concrete_in_Keep_d * MKI_value_future['Concrete
                    C35/45 (CEMIII)']
                    CO2_tax_Keep_concrete_d = volume_concrete_in_Keep_d *

```

Figure F-6: Code Capping Beam Estimation

```

xl = win32com.client.Dispatch("Excel.Application")
xl.Workbooks.Open(os.path.abspath("T-Normaalrecht_en_Buiging_D02_v2.0.Balk.xlsx"))
wb = xl.Workbook("T-Normaalrecht_en_Buiging_D02_v2.0.Balk.xlsx") # connect to an existing file in
the current working directory
sht = wb.sheets["D02sheet"]
sht.range("D77").value = "type of concrete"
sht.range("H20").value = 0 # no stirrups
sht.range("F76").value = 0 # no normal force
sht.range("F87").value = 0 # no normal force
sht.range("E82").value = "Bollard_force(sheep) Bollard force / shear force"

# Concrete cover according to Vrijbaardt
sht.range("E23").value = 50
sht.range("E33").value = 60
reinforcement_diameter_kestp = 16 # mm
reinforcement_stirrups = 12 # mm
sht.range("E23").value = reinforcement_diameter_kestp
sht.range("E33").value = reinforcement_diameter_kestp
sht.range("E19").value = reinforcement_stirrups
sht.range("E30").value = 37 #spacing between bar layers
sht.range("E28").value = 37 #spacing between bar layers
# yk reinforcement steel = sht.range("G32").value

# Sheet Piles
for j in range(range length pile):
    for i in range(range number of anchor):
        for s in range(range_sheetpile):
            run = (i+sheetpile) + str(j) + '|' + str(i) + '|' + str(s) + '|'
            if run in Name of run:
                if (Outputfile_sheetpile_total_costs_sheetpile_chamber(s)) ==
                    # Width
                    width_kestp = max(200,500-int((Sheetpile_height(s)) +
                    max(int((Sheetpile_height(s)) / float(CombinedWallPilesDiameter(s))*1000) + 200) # (mm)
                    sht.range("E16").value = width_kestp

                    #moments
                    # AA = AA + 1
                    ground moment max = 1 / 8 * distance between anchors *
                    distance between_anchors*max(results_Dsheet2.anchor_force_1_stage_3[run] +
                    results_Dsheet2.anchor_force_2_stage_3[run] + results_Dsheet2.anchor_force_1_stage_4[run] +
                    results_Dsheet2.anchor_force_2_stage_4[run])
                    bollard moment max = 1 / 4 * max(Bollard_force(sheep) Bollard load seagong) *
                    max(distance between bollards seagong, distance between bollards inland)
                    maximum moment_kestp = ground moment_max + bollard moment_max
                    sht.range("F77").value = [maximum moment_kestp] # Maximum moment_kestp
                    sht.range("F83").value = [maximum moment_kestp] # Maximum moment_kestp
                    space needed for reinforcement = False
                    iteration_kestp = False
                    height_kestp = 900
                    effective_height_kestp = (max(200,500-int((Sheetpile_height(s)) +
                    max(int((Sheetpile_height(s)) / float(CombinedWallPilesDiameter(s))*1000) + 200) - 60 - 0.5 *
                    reinforcement_diameter_kestp - reinforcement_stirrups
                    # A reinforcement_kestp = maximum moment_kestp * (1000000) /
                    (if yk_reinforcement_steel * 0.9 * effective_height_kestp)
                    number of bars_kestp = math.ceil(A_reinforcement_kestp /
                    ((reinforcement_diameter_kestp ** 2) * 0.25 * np.pi)) + 1
                    #Shear Force
                    sht.range("F89").value =
                    [max(Bollard_force(sheep) results_Dsheet2.anchor_force_2_stage_3[run])] # bollard force / shear
                    force
                    for i in range(10):
                        maximum_bars_per_height_kestp = int((height_kestp - 2 * 60) / (37 +
                        reinforcement_diameter_kestp))
                        number of bars_kestp = math.ceil(A_reinforcement_kestp /
                        ((reinforcement_diameter_kestp ** 2) * 0.25 * np.pi)) + 1
                        sht.range("E33").value = reinforcement_diameter_kestp

                    sht.range("F77").value = [maximum moment_kestp] # Maximum moment_kestp
                    sht.range("F83").value = [maximum moment_kestp] # Maximum moment_kestp
                    space needed for reinforcement = False
                    iteration_kestp = False
                    height_kestp = 1000
                    effective_height_kestp = width_kestp - 60 - 0.5 * reinforcement_diameter_kestp -
                    reinforcement_stirrups
                    # A reinforcement_kestp = maximum moment_kestp * (1000000) /
                    (if yk_reinforcement_steel * 0.9 * effective_height_kestp)
                    number of bars_kestp = math.ceil(A_reinforcement_kestp /
                    ((reinforcement_diameter_kestp ** 2) * 0.25 * np.pi))
                    for i in range(10):
                        maximum_bars_per_height_kestp = int((height_kestp - 2 * 60) / (37 +
                        reinforcement_diameter_kestp))
                        number of bars_kestp = math.ceil(A_reinforcement_kestp /
                        ((reinforcement_diameter_kestp ** 2) * 0.25 * np.pi)) + 1
                        sht.range("E33").value = reinforcement_diameter_kestp
                        if number of bars_kestp > 3* maximum_bars_per_height_kestp:
                            reinforcement_diameter_kestp = 2
                            space needed for reinforcement = True
                        else:
                            if number of bars_kestp <= maximum_bars_per_height_kestp:
                                sht.range("C31").value = [number of bars_kestp]
                                sht.range("C29").value = 0
                                sht.range("C27").value = 0
                            elif number of bars_kestp <= 2* maximum_bars_per_height_kestp:
                                sht.range("C31").value = [maximum_bars_per_height_kestp]
                                sht.range("C29").value = [maximum_bars_per_height_kestp -
                                maximum_bars_per_height_kestp]
                                sht.range("C27").value = 0
                                number of bars_kestp_second_row = float(number of bars_kestp -
                                maximum_bars_per_height_kestp)
                            elif number of bars_kestp <= 3* maximum_bars_per_height_kestp:
                                sht.range("C31").value = [maximum_bars_per_height_kestp]
                                sht.range("C29").value = maximum_bars_per_height_kestp
                                sht.range("C27").value = number of bars_kestp - 2 *
                                maximum_bars_per_height_kestp
                                sht.range("E27").value = reinforcement_diameter_kestp
                                sht.range("E15").value = height_kestp
                                xl.Application.Run("BerekenD02_Bereken_D02")

                                if (sht.range("G77").value) == '-':
                                    number of bars_kestp += 1
                                elif (sht.range("G77").value) > 1:
                                    number of bars_kestp += 1
                                elif (sht.range("G77").value) <= 1:
                                    #print ("Keep reinforcement check")
                                    if (sht.range("G90").value) == '-':
                                        print ("-")
                                    elif (sht.range("G90").value) > 1:
                                        if iteration_kestp == True:
                                            number of bars_kestp += 1
                                            break
                                    #print ("sht.range('G90').value)
                                    elif (sht.range("G90").value) <= 1:
                                        break

                                # DIAPHRAGM WALL
                                for ddd in range(range length diaphragm wall):
                                    for dddd in range(range thickness diaphragm wall):
                                        run = (i+sheetpile) + str(ddd) + '|' + str(ddd) + '|'
                                        if run in Name of run d:
                                            if (Outputfile_diaphragm_wall_total_costs_diaphragm_wall_chamber(s)) ==
                                                # Width
                                                width_kestp = 650 + int(thickness diaphragm wall(ddd)) # (mm) #misschien nog
                                                plus de 200mm
                                                sht.range("E16").value = width_kestp

                                                #moments
                                                # AA = AA + 1
                                                ground moment max = 1 / 8 * distance between anchors *
                                                distance between_anchors*max(results_Dsheet2_d.anchor_force_1_stage_3_d[run] +
                                                results_Dsheet2_d.anchor_force_2_stage_3_d[run] + results_Dsheet2_d.anchor_force_1_stage_4_d[run] +
                                                results_Dsheet2_d.anchor_force_2_stage_4_d[run])
                                                bollard moment max = 1 / 4 * max(Bollard_force(sheep) Bollard load seagong) *
                                                max(distance between bollards seagong, distance between bollards inland)
                                                maximum moment_kestp = ground moment_max + bollard moment_max

```

Figure F-7: Code Capping Beam Check

```

def excavation_steep pit(excavation_width):
    total volume excavation = excavation width * (toplevel layer1 - (Top_of_structure tov_NAP-
    Height-H Underwaterconcretefloor/1000-extra excavation UWC)) * Length

    Height_excavated_backfill_sand = 0
    Height_excavated_peat = 0
    Height_excavated_clay = 0
    for i in range(1,len(z)): #check whether sand is excavated which can be used to use as
    backfill for UWC
        if Values_per_depth.name_soil[z[i]] == 'Sand':
            Height_excavated_backfill_sand += 0.1
        if Values_per_depth.name_soil[z[i]] == 'Peat':
            Height_excavated_peat += 0.1
        if Values_per_depth.name_soil[z[i]] == 'Clay':
            Height_excavated_clay += 0.1
        if z[i] < Top_of_structure tov_NAP-Height:
            break
    volume_excavated_backfill_sand = Height_excavated_backfill_sand * excavation width * Length
    volume_soil_export_peat = Height_excavated_peat * excavation width * Length
    volume_soil_export_clay = Height_excavated_clay * excavation width * Length

    if toplevel layer1 < Top_of_structure tov_NAP:
        needed_backfill_sand = ((Top_of_structure tov_NAP - toplevel layer1) *
        (extra_backfill_width_next_to_lock**2*(Top_of_structure tov_NAP - toplevel layer1)/
        np.tan(input Phi_soil_backfill*np.pi/180)) + (excavation width*extra_excavation UWC) +
        (excavation width*Width)* (Top_of_structure tov_NAP-bottom_structure)) * Length
    else:
        needed_backfill_sand = (Width*extra_excavation UWC + (excavation width-Width)*
        (Top_of_structure tov_NAP-bottom_structure)) * Length

    if needed_backfill_sand < volume_excavated_backfill_sand:
        volume_soil_export_sand = volume_excavated_backfill_sand - needed_backfill_sand
        volume_soil_import_backfill = 0
        re_used_sand = needed_backfill_sand
    else:
        volume_soil_export_sand = 0
        volume_soil_import_backfill = needed_backfill_sand - volume_excavated_backfill_sand
        re_used_sand = volume_excavated_backfill_sand

    return (needed_backfill_sand,volume_soil_export_sand
    ,volume_soil_import_backfill,total_volume_excavation,volume_soil_export_peat,volume_soil_export_c
    lay,re_used_sand)

```

Figure F-8: Code Excavation Retaining Wall Building Pit

```

def excavation_soil(bottom_excavation):
    first_layer = True
    verticalaal_totaal = 0
    horizontaal_totaal = 0
    square_meters_excavated_sand = 0
    square_meters_excavated_peat = 0
    square_meters_excavated_clay = 0
    excavated_soil = np.zeros(len(toplevel_layer))

    for i in range(len(toplevel_layer)-1):
        if bottom_excavation < toplevel_layer[-(i+1)]:
            if first_layer == True:
                verticalaal = toplevel_layer[-(i+1)] - bottom_excavation
                verticalaal_totaal += verticalaal
                horizontalaal = verticalaal / np.tan(input Phi_soil[-(i+1)] * np.pi / 180)
                excavated_soil[-(i+1)] = (width_excavation_bottom + horizontalaal
                *horizontalaal_totaal) * verticalaal

                horizontalaal_totaal += horizontalaal
                if Name_Layer[-(i+1)] == 'Sand':
                    square_meters_excavated_sand += excavated_soil[-(i+1)]
                elif Name_Layer[-(i+1)] == 'Peat':
                    square_meters_excavated_peat += excavated_soil[-(i+1)]
                elif Name_Layer[-(i+1)] == 'Clay':
                    square_meters_excavated_clay += excavated_soil[-(i+1)]
                first_layer = False

                verticalaal = toplevel_layer[-(i+2)]-toplevel_layer[-(i+1)]
                verticalaal_totaal += verticalaal
                horizontalaal = verticalaal / np.tan(input Phi_soil[-(i+1)] * np.pi / 180)
                excavated_soil[-(i+2)] = (width_excavation_bottom + horizontalaal +
                horizontalaal_totaal) * verticalaal
                horizontalaal_totaal += horizontalaal

                if Name_Layer[-(i+1)] == Name_Layer_backfill:
                    break
                if Name_Layer[-(i+2)] == 'Sand':
                    square_meters_excavated_sand += excavated_soil[-(i+2)]
                elif Name_Layer[-(i+2)] == 'Peat':
                    square_meters_excavated_peat += excavated_soil[-(i+2)]
                elif Name_Layer[-(i+2)] == 'Clay':
                    square_meters_excavated_clay += excavated_soil[-(i+2)]

                if Name_Layer[-(i+2)] == Name_Layer_backfill:
                    break
                elif Name_Layer[-(i+3)] == Name_Layer_backfill:
                    break

            total_excavated_volume = sum(excavated_soil) * Length
            total_excavated_volume_sand = square_meters_excavated_sand * Length
            total_excavated_volume_peat = square_meters_excavated_peat * Length
            total_excavated_volume_clay = square_meters_excavated_clay * Length

    return(total_excavated_volume,total_excavated_volume_sand,total_excavated_volume_peat,total_excav
    ated_volume_clay)

bottom_excavation_with_piles = Top_of_structure tov_NAP - Height - floor_height_U_basin_max
bottom_excavation_without_piles = Top_of_structure tov_NAP - Height - floor_height_U_basin_max -
    extra_thickness_U_basin_floor

excavation_soil(bottom_excavation_with_piles)

```

```

else:
    needed_volume_backfill_sand = total_excavated_volume - (bottom_excavation *
    width_excavation_bottom * length)

    if needed_volume_backfill_sand > total_excavated_volume_sand:
        export_volume_backfill_sand = needed_volume_backfill_sand - total_excavated_volume_sand
        export_volume_soil_sand = 0
        re_used_sand = total_excavated_volume_sand
    else:
        import_volume_backfill_sand = 0
        export_volume_soil_sand = total_excavated_volume_sand - needed_volume_backfill_sand
        re_used_sand = needed_volume_backfill_sand

return(import_volume_backfill_sand,export_volume_soil_sand,needed_volume_backfill_sand,re_used_s
    and)

```

Figure F-9: Code Excavation Open Building Pit

```

for i in range(range_length_pile):
    for l in range(range_number_of_anchor):
        for s in range(range_sheetpile):

            input_top_level_sheetpile = Top of structure tov NAP #tov nul level
            input_bottom_sheetpile = Top of structure tov NAP-lengthvanpile[s]
            input_length_sheetpile = -1* lengthvanpile[s] #sheet met min en plus te maken, alleen
            inputbottomsheetpile = 0

            #####Creating an input
            file#####
            def Input_file_Dsheet():
                with open("Dsheetinput.txt","w") as DS:
                    print("INPUT FILE FOR D-SHEET PILING",file=DS)

print("-----", file=DS)
print("COMPANY : Royal HaskoningDHV",file=DS)
print("DATE : 11/26/2018", file=DS)
print("TIME : 11:12:38 AM", file=DS)
print("FILENAME : ", file=DS)
print("CREATED BY : Dsheet Piling version 18.2.1.20477", file=DS)
print("----- BEGINNING OF DATA -----", file=DS)
print(" ", file=DS)
print("VERSION", file=DS)
print("Soil=1004", file=DS)
print("D-Sheet Piling=1013", file=DS)
print("END OF VERSION", file=DS)
print(" ", file=DS)
print("VERSION EXTERNALS", file=DS)
print("OSGCPipCalc=18.2.1.20463", file=DS)
print("OSGSMoCalc=18.2.1.20455", file=DS)
print("END OF VERSION EXTERNALS", file=DS)
print(" ", file=DS)

print("SOIL COLLECTION", file=DS)
print(l2, " = number of items", file=DS)

#SOIL LAYER 1
for i in range(10):
    # SOIL LAYER 1
    print("SOIL", file=DS)
    print(name_soillayer[i+1], file=DS)
    print("SoilColor", color_soil[i+1], file=DS)
    print("SoilSoilType", file=DS)
    if Name_Layer[i + 1] == "Sand":
        print("SoilGrainType=1", file=DS)
    else:
        print("SoilGrainType=0", file=DS)
    print("SoilGSDry", input_Gamma_Dry_soil[i+1], file=DS)
    print("SoilGSDry", input_Gamma_Dry_soil[i+1], file=DS)
    print("SoilGSDens", input_Gamma_Wet_soil[i+1], file=DS)
    print("SoilCohesion", input_Cohesion_soil[i+1], file=DS)
    print("SoilPhi", input_Phi_soil[i+1], file=DS)
    if Name_Layer[i+1] == "Grout":
        print("SoilDelta", 0, file=DS)
    else:
        print("SoilDelta", min(input_Phi_soil[i+1] - 2.5, 27.5), file=DS)
    print("SoilUseMenard=1", file=DS)
    print("SoilUseBrinch Hansen=1", file=DS)
    print("SoilLamdaType=2", file=DS)
    print("END OF SOIL", file=DS)

# SOIL LAYER CONCRETE
print("SOIL", file=DS)
print(name_soillayer_c, file=DS)
print("SoilColor=55498", file=DS)
print("SoilSoilType", file=DS)
print("SoilGrainType=0", file=DS)
print("SoilGSDry", input_Gamma_Dry_soil_c, file=DS)

print("0 = Sheet without beam = FALSE", file=DS)
print("0 = Tullens calculation = FALSE", file=DS)
print("0 = Diepdaag calculation = FALSE", file=DS)
print("0 = Elastic calculation = TRUE", file=DS)
print("END OF MODEL", file=DS)
print(" ", file=DS)
print("CFR LIST", file=DS)
print("Count", file=DS)
print("END OF CFR LIST", file=DS)
print(" ", file=DS)
print("SHEET PILING", file=DS)
print("0", file=DS)
print(input_top_level_sheetpile, "Level top sheet piling", file=DS)
print(input_length_sheetpile, "Length", file=DS)
print("Number of elements", file=DS)
print("SHEET PILING ELEMENT", file=DS)

print(Sheetpile_name[s], file=DS)
print(SheetPilingElementMaterialType, Sheetpile_material[s], file=DS)
print(SheetPilingElementEI, Sheetpile_EI[s], file=DS)
print(SheetPilingElementMod, Sheetpile_k_mod[s], file=DS)
print(SheetPilingElementLevel, input_bottom_sheetpile, file=DS)
print(SheetPilingElementHeight, Sheetpile_height[s], file=DS)
print(SheetPilingElementArea, Sheetpile_area[s], file=DS)
print(SheetPilingElementWidth, Sheetpile_width[s], file=DS)
print(SheetPilingElementSectionArea, Sheetpile_sectionarea[s], file=DS)
print("SheetPilingElementResistingMoment", Sheetpile_resisting_moment[s],
file=DS)

print("SheetPilingElementReductionFactorEI", Sheetpile_reduction_factor_EI[s], file=DS)
print("SheetPilingElementNote", file=DS)

print("SheetPilingElementMaxCharacteristicMoment", Sheetpile_maximum_characteristic_moment[s],
file=DS)
print("SheetPilingElementMaxPlasticCharacteristicMoment", Sheetpile_plastic_characteristic_moment
[s], file=DS)
print("SheetPilingElementMaterialFactor", Sheetpile_materialfactor[s],
file=DS)
print("SheetPilingElementMaxMoment", Sheetpile_Max_moment[s], file=DS)
print("SheetPilingElementMaxPlasticMoment", Sheetpile_Max_plastic_moment[s],
file=DS)
print("SheetPilingElementReductionFactorMaxMoment", Sheetpile_reduction_factor_max_moment[s],
file=DS)

print("DiaphragmWallSymmetric=0", file=DS)
print("DiaphragmWallNegElastoPlastic=2.00000E+04", file=DS)
print("DiaphragmWallPosElastoPlastic=2.00000E+04", file=DS)
print("DiaphragmWallPosMomElastic=0.00", file=DS)
print("DiaphragmWallPosMomPlastic=0.00", file=DS)
print("DiaphragmWallNegMomPlastic=0.00", file=DS)
print("DiaphragmWallPosElastoPlastic=2.00000E+04", file=DS)
print("DiaphragmWallPosMomElastoPlastic=0.00", file=DS)
print("DiaphragmWallNegElastoPlastic=2.00000E+04", file=DS)
print("DiaphragmWallNegMomElastoPlastic=0.00", file=DS)
print("END OF SHEET PILING", file=DS)
print("END OF SHEET PILING", file=DS)

print(" ", file=DS)
print("COMBINED WALL", file=DS)
print("CombinedWallPileName", CombinedWallPileName[s], file=DS)
print("CombinedWallPileMaterialType", file=DS)
print("CombinedWallPileBottomLevel", input_bottom_sheetpile, file=DS)
print("CombinedWallPileEI", file=DS)
print("CombinedWallPileMod", file=DS)
print("CombinedWallPileArea", file=DS)
print("CombinedWallPileSectionArea", file=DS)
print("CombinedWallPileMaxMoment", file=DS)

print("CombinedWallPileMaxPlasticMoment", CombinedWallPileMaxPlasticMoment[s], file=DS)
print("CombinedWallPileSectionArea", CombinedWallPileSectionArea[s],
file=DS)

print("CombinedWallSheetPileName", CombinedWallSheetPileName[s], file=DS)

```

Figure F-10: Code Sheet Pile Wall Structure Input (1)

```

print("SurchargeLoad", file=DS, magnitude_surcharge_load_start, file=DS)
print("END OF POINT", file=DS)
print("POINT", file=DS) #Hier eindigt de load
print("SurchargeLoadDistance", begin surcharge load end, file=DS)
print("SurchargeLoad", file=DS, magnitude_surcharge_load_end, file=DS)
print("END OF POINT", file=DS)
print("SurchargeLoadPermanent" = 0", file=DS)
print("SurchargeLoadVariable" = 0", file=DS)
print("SurchargeLoadStandardDeviation" = 0.00", file=DS)
print("SurchargeLoadDistribution" = 2", file=DS)
print("END OF LOAD", file=DS)
print("END OF SURCHARGE LOADS", file=DS)
print("HORIZONTAL LINE LOADS", file=DS)
print("2 Number of horizontal line loads", file=DS)
print("Name", file=DS) #Name of the load
print("1", "Lowest Lock Level/Vertical distance load from water, -
Maximum Horizontal Bollard load per meter inland or recreational," 1 0
Bollard load recreational or inland marine", file=DS)
print("2", "input top level sheetpile, -
Bollard load per meter seagoing," 0 0 Bollard load sea going vessel", file=DS)
print("END OF HORIZONTAL LINE LOADS", file=DS)
#Hier begint water
print("WATER", file=DS)
print("Unit weight of water," Unit weight of water, file=DS) #dit kan
sangepast moeten worden i/va zee water
print("END OF WATER", file=DS)
print("EARTH QUAKE", file=DS)
print("0.00", file=DS)
print("END OF EARTH QUAKE", file=DS)
print("Soil profile", file=DS)
print("1 Number of spring characteristics curves", file=DS)
print("0 1/0 : Yes/No Unloading curve", file=DS)
print("1 1/0 : Yes/No Use our values", file=DS)
print("3 Number of soil profiles", file=DS)
print("New Profile", file=DS)
print("0.000 X coordinate", file=DS)
print("0.000 Y coordinate", file=DS)
print("number of layers", "Number of soil layers per soil profile",
file=DS)
print("Nr Level Wosp Top Wosp Bott Material Name", file=DS)
print(Nr_layer1, toplevel_layer1, Wosp_Top_layer1, Wosp_bott_layer1,
name_soillayer_1, file=DS)
if number of layers > 1:
    print(Nr_layer2, toplevel_layer2, Wosp_Top_layer2, Wosp_bott_layer2,
name_soillayer_2, file=DS)
if number of layers > 2:
    print(Nr_layer3, toplevel_layer3, Wosp_Top_layer3, Wosp_bott_layer3,
name_soillayer_3, file=DS)
if number of layers > 3:
    print(Nr_layer4, toplevel_layer4, Wosp_Top_layer4,
Wosp_bott_layer4, name_soillayer_4, file=DS)
if number of layers > 4:
    print(Nr_layer5, toplevel_layer5, Wosp_Top_layer5,
Wosp_bott_layer5, name_soillayer_5, file=DS)
if number of layers > 5:
    print(Nr_layer6, toplevel_layer6, Wosp_Top_layer6,
Wosp_bott_layer6, name_soillayer_6, file=DS)
if number of layers > 6:
    print(Nr_layer7, toplevel_layer7, Wosp_Top_layer7,
Wosp_bott_layer7, name_soillayer_7, file=DS)
if number of layers > 7:
    print(Nr_layer8, toplevel_layer8, Wosp_Top_layer8,
Wosp_bott_layer8, name_soillayer_8, file=DS)
if number of layers > 8:
    print(Nr_layer9, toplevel_layer9, Wosp_Top_layer9,
Wosp_bott_layer9, name_soillayer_9, file=DS)
if number of layers > 9:
    print(Nr_layer10, toplevel_layer10,
Wosp_Top_layer10, Wosp_bott_layer10, name_soillayer_10, file=DS)
print("0.00", "top under floor 9", (bottom structure -
waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -unit weight of water,
Underfloor 7, file=DS)
if number of layers > 7:
    if top under floor 8 >= lowest layer:
        print("8", top under floor 8,
(bottom structure - waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 8, file=DS)
    if number of layers > 8:
        if top under floor 9 >= lowest layer:
            print("9", top under floor 9,
(bottom structure - waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 9, file=DS)
        if number of layers > 9:
            if top under floor 10 >= lowest layer:
                print("10", top under floor 10,
(bottom structure - waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 10, file=DS)
    #soil profile 3
    print("Back fill", file=DS)
    print("0.000 X coordinate", file=DS)
    print("0.000 Y coordinate", file=DS)
    print("number of layers", "Number of soil layers per soil
profile", file=DS)
    else:
        print(" ", number_of_layers, "Number of soil layers per soil profile",
file=DS)
    print("Nr Level Wosp Top Wosp Bott Material Name", file=DS)
    if toplevel layer backfill > toplevel layer1:
        print(1, toplevel layer backfill, Wosp_Top layer backfill,
Wosp_bott_layer backfill, Name Layer backfill, file=DS)
        backfill needed = 1
    else:
        backfill needed = 0
    print(Nr_layer1 + backfill needed, toplevel_layer1, Wosp_Top_layer1,
Wosp_bott_layer1, name_soillayer_1, file=DS)
    if number of layers > 1:
        print(Nr_layer2 + backfill needed, toplevel_layer2, Wosp_Top_layer2,
Wosp_bott_layer2, name_soillayer_2, file=DS)
    if number of layers > 2:
        print(Nr_layer3 + backfill needed, toplevel_layer3, Wosp_Top_layer3,
Wosp_bott_layer3, name_soillayer_3, file=DS)
    if number of layers > 3:
        print(Nr_layer4 + backfill needed, toplevel_layer4,
Wosp_Top_layer4, Wosp_bott_layer4, name_soillayer_4, file=DS)
    if number of layers > 4:
        print(Nr_layer5 + backfill needed, toplevel_layer5,
Wosp_Top_layer5, Wosp_bott_layer5, name_soillayer_5, file=DS)
    if number of layers > 5:
        print(Nr_layer6 + backfill needed, toplevel_layer6,
Wosp_Top_layer6, Wosp_bott_layer6, name_soillayer_6, file=DS)
    if number of layers > 6:
        print(Nr_layer7 + backfill needed, toplevel_layer7,
Wosp_Top_layer7, Wosp_bott_layer7, name_soillayer_7, file=DS)
    if number of layers > 7:
        print(Nr_layer8 + backfill needed, toplevel_layer8,
Wosp_Top_layer8, Wosp_bott_layer8, name_soillayer_8, file=DS)
    if number of layers > 8:
        print(Nr_layer9 + backfill needed, toplevel_layer9,
Wosp_Top_layer9, Wosp_bott_layer9, name_soillayer_9, file=DS)
    if number of layers > 9:
        print(Nr_layer10 + backfill needed, toplevel_layer10, Wosp_Top_layer10,
Wosp_bott_layer10, name_soillayer_10, file=DS)
print("END OF SOIL PROFILES", file=DS)
print("SURFACES", file=DS)
print("4 Number of surfaces", file=DS)
print("1", name groundlevel, file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("1 0.00", "begin groundlevel, file=DS)
print("2 1", name second surface, file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("3 1", "bottom of structure", file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("4 1", "After Back Fill", file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("5 7", "top of structure to w NAP", file=DS)
print("END OF SURFACES", file=DS)
print("WATERLEVELS", file=DS)
print("3 Number of Waterlevels", file=DS)
print("3 Number of data per waterlevel", file=DS)
print(name waterlevel1, file=DS) #Naam van waterlevel1
print("waterlevel1, file=DS) #Niveau van het water
print("0.00", file=DS)
print("2", file=DS)
print(name waterlevel2, file=DS) # naam van waterlevel
print(" ", waterlevel2, file=DS) # level van het water
print("0.00", file=DS)
print("2", file=DS)
print(name waterlevel3, file=DS) # naam van waterlevel
print(" ", waterlevel3, file=DS) # level van het water
print("0.00", file=DS)
print("2", file=DS)
print("END OF WATERLEVELS", file=DS)
print("CONSTRUCTION STAGES", file=DS)
#Hier komen de stages en welke shit er wordt gebruikt per stage
print("1 Method Left: Ka, Ko, Rp Method Right: Ka, Ko,
Rp", file=DS) #welke
method wordt gebruikt voor wat
print("bottom of structure", file=DS) #Naam van grondlevel links
print(name groundlevel, file=DS) # Naam grondlevel Rechts
print(name waterlevel1, file=DS) #Naam Grondlevel links
print(name waterlevel1, file=DS) #Naam waterlevel rechts
print(soilprofile right, file=DS)
print("0.000", "displacement top sheet piling", file=DS)
print("0", "Use top displacement", file=DS)
print(" ", number_of_anchors[1], "Anchors present in
stage", file=DS)
print("1 0.00", "Name anchor", file=DS)
if number of anchors[1] == 2:
    print("2 0.00", "Name anchor", file=DS)
print("0", "Status present in stage", file=DS)
print("0", "Spring supports present in stage", file=DS)
print("0", "Rigid supports present in stage", file=DS)

```

Figure F-11: Code Sheet Pile Wall Structure Input (2)

```
print(" 1 Uniform loads in stage", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 2:
print("New Stage 2", file=DS)
#p", file=DS) # welke method wordt gebruikt voor wat
print(name second surface, file=DS) # Naam van grondlevel links
print(name groundlevel, file=DS) # Naam grondlevel Rechts
print(name_waterlevel2, file=DS) # Naam Grondlevel links
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(soilprofile right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage", file=DS)
print(" 1 0.00", Name anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 0 Struts present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 2 Uniform loads in stage", file=DS)
print(" 1 concrete floor load", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 3:
print("New Stage 3", file=DS)
#p", file=DS) # welke method wordt gebruikt voor wat
print(name second surface, file=DS) # Naam van grondlevel links
print(name groundlevel, file=DS) # Naam grondlevel Rechts
print(name_waterlevel2, file=DS) # Naam Grondlevel links
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(soilprofile right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage", file=DS)
print(" 1 0.00", Name anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 0 Struts present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 2 Uniform loads in stage", file=DS)
print(" 1 concrete floor load", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 4:
print("New Stage 4", file=DS)
#p", file=DS) # welke method wordt gebruikt voor wat
print(name second surface, file=DS) # Naam van grondlevel links
print(name groundlevel, file=DS) # Naam grondlevel Rechts
print(name_waterlevel2, file=DS) # Naam Grondlevel links
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(soilprofile right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage", file=DS)
print(" 1 0.00", Name anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 0 Struts present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 2 Uniform loads in stage", file=DS)
print(" 1 concrete floor load", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 1 Bolland_load_recreational_or_inlandwaterway", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 5:
print("New Stage 5", file=DS)
#p", file=DS) # welke method wordt gebruikt voor wat
print(name second surface, file=DS) # Naam van grondlevel links
print(name groundlevel, file=DS) # Naam grondlevel Rechts
print(name_waterlevel2, file=DS) # Naam Grondlevel links
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(soilprofile right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage", file=DS)
print(" 1 0.00", Name anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 0 Struts present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 2 Uniform loads in stage", file=DS)
print(" 1 concrete floor load", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 1 Bolland_load_sea_going_vessel", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
print("END OF CALCULATION OPTIONS FOR STAGE", file=DS)
print("", file=DS)
# Geen idee nog wat hier gebeurt
print("CONSTRUCTION OPTIONS FOR STAGE", file=DS)
print("2 = number of items", file=DS)
stage 1
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADSD=0", file=DS)
print("StageAnchorFactorECTNADSD=1.000", file=DS)
print("END OF STAGE", file=DS)
stage 2
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADSD=0", file=DS)
print("StageAnchorFactorECTNADSD=1.000", file=DS)
print("END OF STAGE", file=DS)
stage 3
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADSD=0", file=DS)
print("StageAnchorFactorECTNADSD=1.000", file=DS)
print("END OF STAGE", file=DS)
stage 4
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADSD=0", file=DS)
print("StageAnchorFactorECTNADSD=1.000", file=DS)
print("END OF STAGE", file=DS)
stage 5
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADSD=0", file=DS)
print("StageAnchorFactorECTNADSD=1.000", file=DS)
print("END OF STAGE", file=DS)
print("END OF CALCULATION OPTIONS FOR STAGE", file=DS)
print("", file=DS)
thisFile = "Dsheetsinput.txt"
base = os.path.splitext(thisFile)[0]
#os.remove(base + ".str(1)" + ".str(1)" + ".str(1)" + ".SHI")
os.rename(thisFile, base + ".str(1)" + ".str(1)" + ".str(1)" + ".SHI")
#os.rename(os.path.abspath(thisFile),
# "C:\Users\908729\Documents\Pycharm_tool\1D_sheetpile\Dsheetsoutput' + ".str(1)" + ".str(1)" + ".str(1)" +
# ".SHI")
Input file Dsheet ()
```

Figure F-12: Code Sheet Pile Wall Structure Input (3)

```

import subprocess
import os

# Opens and runs all the SHI files in the given map.
Location_DSheetPiling = 'C:\Program Files (x86)\Delaware\DSheetPiling\DSheetPiling.exe'
Location_file = 'C:\Users\908729\Documents\Psycharm_tool1'

D = subprocess.Popen([Location_DSheetPiling, r'/b', Location_file])
D.wait()

for j in range(range_length_pile):
    for ll in range(range_number_of_anchor):
        for ss in range(range_sheetpile):
            AA = AA+1
            Name_of_run[AA] = 'D-Sheetoutput[' + str(j) + ']' + str(ll) + '[' + str(ss) + ']'
            Run_number[AA] = int(AA)
            sheetpile_length[AA] = length_of_pile[j]
            Anchors_present[AA] = number_of_anchors[ll]
            Sheetpile_profile[AA] = Sheetpile_name[ss]
            Sheetpile_couting_area[AA] = Sheetpile_coutingarea[ss]
            if CombinedWallNumberofSheetPiles[ss] == 0:
                sectional_area_sheetpile[AA] = Sheetpile_sectionarea[ss]
                Sheetpile_couting_area[AA] = Sheetpile_coutingarea[ss]
            else:
                sectional_area_sheetpile[AA] = (float(Sheetpile_sectionarea[ss]) *
                int(CombinedWallNumberofSheetPiles[ss]) * 0.70 + float(CombinedWallPilesDiameter[ss])) /
                (int(CombinedWallNumberofSheetPiles[ss]) * 0.70 + float(CombinedWallPilesDiameter[ss])) * 0.7m is
                the width of one A2, Z4 -700 (S270) element
                Sheetpile_couting_area[AA] = (float(Sheetpile_coutingarea[ss]) *
                int(CombinedWallNumberofSheetPiles[ss]) * 0.70 + float(CombinedWallPilesDiameter[ss])) /
                (int(CombinedWallNumberofSheetPiles[ss]) * 0.70 + float(CombinedWallPilesDiameter[ss]))

            try: #this makes sure that the script does not stop with errors. When an error
            occurs, except starts and then the loop continues
            output_file = "Dsheetinput"+str(j)] + str(ll)] + str(ss)]+".shd"
            base = os.path.splitext(output_file)[0]
            os.rename(output_file, base + ".txt")

            lines = []
            with open('Dsheetinput'+str(j)]+str(ll)]+str(ss)]+".txt", 'rb') as in_file:
                for line in in_file:
                    lines.append(line)

            # Stage 1
            for b in range(len(lines)):
                if "moment" in lines[b]:
                    if "shear force" in lines[b + 1]:
                        if "displacements" in lines[b + 2]:
                            if "END OF COLUMN INDICATION" in lines[b + 3]:
                                if "COLUMN DATA" in lines[b + 5]:
                                    begin_stage1 = b + 6
                                    break
            for b in range(len(lines) - begin_stage1):
                if "END OF COLUMN DATA" in lines[b + begin_stage1]:
                    end_stage1 = b + begin_stage1 - 1
                    break

            # Stage 2
            for b in range(len(lines) - end_stage1):
                if "moment" in lines[b + end_stage1]:
                    if "shear force" in lines[b + 1 + end_stage1]:
                        if "displacements" in lines[b + 2 + end_stage1]:
                            if "END OF COLUMN INDICATION" in lines[b + 3 + end_stage1]:
                                if "COLUMN DATA" in lines[b + 5 + end_stage1]:
                                    begin_stage2 = b + 6 + end_stage1
                                    break
            for b in range(len(lines) - begin_stage2):
                if "END OF COLUMN DATA" in lines[b + begin_stage2]:
                    end_stage2 = b + begin_stage2 - 1
                    break

            # Stage 3
            for b in range(len(lines) - end_stage2):
                if "moment" in lines[b + end_stage2]:
                    if "shear force" in lines[b + 1 + end_stage2]:
                        if "displacements" in lines[b + 2 + end_stage2]:
                            if "END OF COLUMN INDICATION" in lines[b + 3 + end_stage2]:
                                if "COLUMN DATA" in lines[b + 5 + end_stage2]:
                                    begin_stage3 = b + 6 + end_stage2
                                    break
            for b in range(len(lines) - begin_stage3):
                if "END OF COLUMN DATA" in lines[b + begin_stage3]:
                    end_stage3 = b + begin_stage3 - 1
                    break

            # Stage 4
            for b in range(len(lines) - end_stage3):
                if "moment" in lines[b + end_stage3]:
                    if "shear force" in lines[b + 1 + end_stage3]:
                        if "displacements" in lines[b + 2 + end_stage3]:
                            if "END OF COLUMN INDICATION" in lines[b + 3 + end_stage3]:
                                if "COLUMN DATA" in lines[b + 5 + end_stage3]:
                                    begin_stage4 = b + 6 + end_stage3
                                    break
            for b in range(len(lines) - begin_stage4):
                if "END OF COLUMN DATA" in lines[b + begin_stage4]:
                    end_stage4 = b + begin_stage4 - 1
                    break

            # Add Results in a DataFrame:
            moment_stage1 = np.zeros(end_stage1 - begin_stage1)
            shear_force_stage1 = np.zeros(end_stage1 - begin_stage1)
            displacement_stage1 = np.zeros(end_stage1 - begin_stage1)
            for b in range(end_stage1 - begin_stage1):
                split_stage1 = lines[b + begin_stage1].split() # split the lines to make
                moment, shearforce and displacements readable
                moment_stage1[b] = float(split_stage1[0])
                shear_force_stage1[b] = float(split_stage1[1])
                displacement_stage1[b] = float(split_stage1[2])

            moment_stage2 = np.zeros(end_stage2 - begin_stage2)
            shear_force_stage2 = np.zeros(end_stage2 - begin_stage2)
            displacement_stage2 = np.zeros(end_stage2 - begin_stage2)
            for b in range(end_stage2 - begin_stage2):
                split_stage2 = lines[b + begin_stage2].split() # split the lines to make
                moment, shearforce and displacements readable
                moment_stage2[b] = float(split_stage2[0])
                shear_force_stage2[b] = float(split_stage2[1])
                displacement_stage2[b] = float(split_stage2[2])

            moment_stage3 = np.zeros(end_stage3 - begin_stage3)
            shear_force_stage3 = np.zeros(end_stage3 - begin_stage3)
            displacement_stage3 = np.zeros(end_stage3 - begin_stage3)
            for b in range(end_stage3 - begin_stage3):
                split_stage3 = lines[b + begin_stage3].split() # split the lines to make
                moment, shearforce and displacements readable
                moment_stage3[b] = float(split_stage3[0])
                shear_force_stage3[b] = float(split_stage3[1])
                displacement_stage3[b] = float(split_stage3[2])

            moment_stage4 = np.zeros(end_stage4 - begin_stage4)
            shear_force_stage4 = np.zeros(end_stage4 - begin_stage4)
            displacement_stage4 = np.zeros(end_stage4 - begin_stage4)
            for b in range(end_stage4 - begin_stage4):
                split_stage4 = lines[b + begin_stage4].split() # split the lines to make
                moment, shearforce and displacements readable
                moment_stage4[b] = float(split_stage4[0])
                shear_force_stage4[b] = float(split_stage4[1])
                displacement_stage4[b] = float(split_stage4[2])

            results_Dsheet = pd.DataFrame({'Moment stage1': moment_stage1,
            'Shear force stage1': shear_force_stage1, 'Displacement stage1':
            displacement_stage1, 'Moment stage2': moment_stage2,
            'Shear force stage2': shear_force_stage2, 'Displacement stage2':
            displacement_stage2, 'Moment stage3': moment_stage3,
            'Shear force stage3': shear_force_stage3, 'Displacement stage3':
            displacement_stage3, 'Moment stage4': moment_stage4,
            'Shear force stage4': shear_force_stage4, 'Displacement stage4':
            displacement_stage4})

            Maximum_moment[AA] =
            max(max(results_Dsheet.Moment_stage1), max(results_Dsheet.Moment_stage2), max(results_Dsheet.Moment
            stage3), max(results_Dsheet.Moment_stage4))
            Minimum_moment[AA] =
            min(min(results_Dsheet.Moment_stage1), min(results_Dsheet.Moment_stage2), min(results_Dsheet.Moment
            stage3), min(results_Dsheet.Moment_stage4))
            Maximum_Shear_Force[AA] =
            max(max(results_Dsheet.Shear_Force_stage1), max(results_Dsheet.Shear_Force_stage2), max(results_Dsh
            eet.Shear_Force_stage3), max(results_Dsheet.Shear_Force_stage4))
            Minimum_Shear_Force[AA] =
            min(min(results_Dsheet.Shear_Force_stage1), min(results_Dsheet.Shear_Force_stage2), min(results_Dsh
            eet.Shear_Force_stage3), min(results_Dsheet.Shear_Force_stage4))
            Maximum_Displacement[AA] =
            max(max(results_Dsheet.Displacement_stage1), max(results_Dsheet.Displacement_stage2), max(results_D
            sheet.Displacement_stage3), max(results_Dsheet.Displacement_stage4))
            Minimum_Displacement[AA] =
            min(min(results_Dsheet.Displacement_stage1), min(results_Dsheet.Displacement_stage2), min(results_D
            sheet.Displacement_stage3), min(results_Dsheet.Displacement_stage4))
            Stable_unstable[AA] = "Stable"

            if Maximum_moment[AA] < float(Sheetpile_Max_moment[ss]):
                if -Minimum_moment[AA] < float(Sheetpile_Max_moment[ss]):
                    Strength_check_sheetpile[AA] = "check"
                else:
                    Strength_check_sheetpile[AA] = "Sheetpile will Fail"
            else:
                Strength_check_sheetpile[AA] = "Sheetpile will Fail"

            if Strength_check_sheetpile[AA] == "Sheetpile will Fail":
                break

            #Find the shear force in the underwater concrete floor
            # Stage
            for d in range(len(lines)):
                if "POINTS ON SHEETPILE" in lines[d]:
                    begin_number_of_nodes_1 = d + 2
                    break
            for d in range(len(lines) - begin_number_of_nodes_1):
                if str(round(floor_level_lock, 1)) in lines[d + begin_number_of_nodes_1]:
                    begin_floor_node_1 = d + 1
                    break
            for d in range(len(lines) - begin_number_of_nodes_1):
                if str(round(bottom_structure, 1)) in lines[d + begin_number_of_nodes_1]:
                    end_floor_node_1 = d
                    break

            for d in range(len(lines) - begin_number_of_nodes_1):
                if "END OF POINTS ON SHEETPILE" in lines[d + begin_number_of_nodes_1]:
                    end_number_of_nodes_1 = d + begin_number_of_nodes_1 - 1
                    break

            # Stage 2
            for d in range(len(lines) - end_number_of_nodes_1):
                if "POINTS ON SHEETPILE" in lines[d + end_number_of_nodes_1]:
                    begin_number_of_nodes_2 = d + 2 + end_number_of_nodes_1
                    break
            for d in range(len(lines) - begin_number_of_nodes_2):
                if str(round(floor_level_lock, 1)) in lines[d + begin_number_of_nodes_2]:
                    begin_floor_node_2 = d + 1
                    break
                if round(float(str(lines[d + begin_number_of_nodes_2].lstrip()), 1)) ==
                round(floor_level_lock, 1):
                    break
            for d in range(len(lines) - begin_number_of_nodes_2):
                if str(round(bottom_structure, 1)) in lines[d + begin_number_of_nodes_2]:
                    end_floor_node_2 = d

            if round(float(str(lines[d + begin_number_of_nodes_2].lstrip()), 2)) ==
            round(bottom_structure, 1):
                break

            for d in range(len(lines) - begin_number_of_nodes_2):
                if "END OF POINTS ON SHEETPILE" in lines[d + begin_number_of_nodes_2]:
                    end_number_of_nodes_2 = d + begin_number_of_nodes_2 - 1
                    break

            minimum_shear = []
            for d in range(end_floor_node_2 - begin_floor_node_2):
                minimum_shear.append(shear_force_stage2[begin_floor_node_2+d])
            minimum_shear = min(minimum_shear)

            Force_in_Underwaterconcrete_by_sheetpile[AA] =
            minimum_shear * (end_floor_node_2 - begin_floor_node_2) +
            shear_force_stage2[begin_floor_node_2] - minimum_shear

            ## ANCHOR FORCE
            count = 0 #for the different stages use count
            for d in range(len(lines)):
                if "ANCHOR DATA" in lines[d]:
                    count += 1
                    begin_anchor_data = d + 13
                    if (lines[d + 13].split()[0]) == "1":
                        anchor_force_1[AA, count-1] = lines[d + 14].split()[2]
                        anchor_force_2[AA, count-1] = lines[d + 15].split()[2]
                    if (lines[d + 13].split()[0]) == "2":
                        anchor_force_1[AA, count-1] = lines[d + 14].split()[2]
                        anchor_force_2[AA, count-1] = lines[d + 15].split()[2]
                    anchor_force_1_stage_1[AA] = anchor_force_1[AA, 0]
                    anchor_force_1_stage_2[AA] = anchor_force_1[AA, 1]
                    anchor_force_1_stage_3[AA] = anchor_force_1[AA, 2]
                    anchor_force_1_stage_4[AA] = anchor_force_1[AA, 3]
                    anchor_force_2_stage_1[AA] = anchor_force_2[AA, 0]
                    anchor_force_2_stage_2[AA] = anchor_force_2[AA, 1]
                    anchor_force_2_stage_3[AA] = anchor_force_2[AA, 2]
                    anchor_force_2_stage_4[AA] = anchor_force_2[AA, 3]

            ## VERSION CHECK
            for d in range(len(lines)):
                if "CREATED BY" in lines[d]:
                    if str(2018077) in lines[d]:
                        version_check = True
                    else:
                        version_check = False
            df_version_check = pd.DataFrame({'Version of D-sheet is changed':
            "The tool is only made for Version
            18.2.1.20477 ",
            'Check input and output if the good
            values are used'})

            except:
                Stable_unstable[AA] = "Unstable"

            os.rename('C:\Users\908729\Documents\Psycharm_tool1\Dsheetinput' + str(j)] +
            str(ll)] + str(ss)] + ".SHI", 'C:\Users\908729\Documents\D sheetpile\Dsheetinput' + str(j)] +
            str(ll)] + str(ss)] + ".SHI")

            if Stable_unstable[AA] == "Unstable":
                print(Name_of_run[AA], Stable_unstable[AA])

            results_Dsheet2 = pd.DataFrame({'Run': Run_number, 'Run Name':
            Name_of_run, 'sheetpile_length': sheetpile_length, 'Anchors_present': Anchors_present,
            'Sheetpile_profile': Sheetpile_profile,
            'sectional_area_sheetpile': sectional_area_sheetpile,
            'Stable_Unstable': Stable_unstable,
            'Maximum_moment': Maximum_moment, 'Minimum_moment': Minimum_moment, 'Maximum_Shear_Force': Maximum_She
            ar_Force,

```

Figure F-13: Code Sheet Pile Wall Structure Output (1)


```

'Minimum_Shear_Force':Minimum_Shear_Force,'Maximum_Displacement':Maximum_Displacement,
'Minimum_Displacement':Minimum_Displacement,'Strength_check_sheetpile':Strength_check_sheetpile,
'Force_in_Underwaterconcrete_by_sheetpile':Force_in_Underwaterconcrete_by_sheetpile,
'anchor_force_1_stage_1':anchor_force_1_stage_1,'anchor_force_2_stage_1':anchor_force_2_stage_1,
'anchor_force_1_stage_2':anchor_force_1_stage_2,'anchor_force_2_stage_2':anchor_force_2_stage_2,
'anchor_force_1_stage_3':anchor_force_1_stage_3,'anchor_force_2_stage_3':anchor_force_2_stage_3,
'anchor_force_1_stage_4':anchor_force_1_stage_4,'anchor_force_2_stage_4':anchor_force_2_stage_4))
results['Sheet2'].set_index('from Sheet', True)

```

Figure F-14: Code Sheet Pile Wall Structure Output (2)

```

def
Resisting_moment_calculation_reinforced_concrete(width_diaphragm_wall,thickness_diaphragm_wall,co
ncrete_cover,f_cd,
number_of_bars,reinforcement_diameter,reinforcement_diameter_stirrups,f_yd=435):
    beta = lambda / 2
    beta = 1 # check EC2 3.1.7 voor siltig, veranderd bij sterker beton
    d = width_diaphragm_wall
    d = thickness_diaphragm_wall - concrete_cover - reinforcement_diameter -
reinforcement_diameter_stirrups - 14 # effective depth #for double layer of bars with distance
of 28 mm
    A_s = (reinforcement_diameter ** 2) * 0.25 * np.pi * number_of_bars
    x = (A_s * f_yd) / (lambda * b * beta * f_cd)
    z = d - (beta * x)
    M_rd_s = f_yd * A_s * z
    M_rd_c = f_cd * b * x * z
    M_rd = min(M_rd_c,M_rd_s)
    return (M_rd)

```

Figure F-15: Code Resisting Moment Calculation Reinforced Concrete


```

print("LOAD", file=DS)
print("nomerite floor_load", file=DS)
print("UniformLoadLeft", (bottom_structure - waterlevel)) * -
unit weight of water, file=DS)
print("UniformLoadRight=0.00", file=DS)
print("END OF LOAD", file=DS)
print("LOAD", file=DS)
print("Surcharge_Load", file=DS)
print("UniformLoadLeft=0", file=DS)
print("UniformLoadRight", surchargeLoad lock, file=DS)
print("END OF LOAD", file=DS)
print("END OF UNIFORM LOADS", file=DS)
print("", file=DS)

print("SURCHARGE LOADS", file=DS)
print("1 - number of items", file=DS)
print("LOAD", file=DS)
print("Surcharge", file=DS)
print("2", file=DS)
print("POINT", file=DS)
print("SurchargeLoadDistance", begin_surcharge_load_start, file=DS)
print("SurchargeLoadValue", magnitude_surcharge_load_start, file=DS)
print("END OF POINT", file=DS)
print("POINT", file=DS)
print("SurchargeLoadDistance", begin_surcharge_load_end, file=DS)
print("SurchargeLoadValue", magnitude_surcharge_load_end, file=DS)
print("END OF POINT", file=DS)
print("SurchargeLoadPermanent = 0", file=DS)
print("SurchargeLoadFavourable = 0", file=DS)
print("SurchargeLoadStandardDeviation = 0.00", file=DS)
print("SurchargeLoadInclination = 27", file=DS)
print("END OF LOAD", file=DS)
print("END OF SURCHARGE LOADS", file=DS)
print("HORIZONTAL LINE LOADS", file=DS)
print("Nr Level Value Permanent Favourable Name", file=DS)
print("1 Lowest Lock Level/Vertical distance load from water,
Maximum Horizontal per meter inland of recreational, 0 0
Bollard load_recreational_of_inlandwaterway, file=DS)
print("2 input_top_level_sheftpile,
Bollard load_per_meter_shefting, 0 0 bollard_load_sea_going_vessel", file=DS)
print("", file=DS)
print("END OF HORIZONTAL LINE LOADS", file=DS)
#Hier begint water
print("WATER", file=DS)
print("unit_weight_of_water," Unit weight of water, file=DS) #dit kan
sangepast moeten worden liv zeewater
print("END OF WATER", file=DS)
print("", file=DS)

print("EARTH QUARE", file=DS)
print("0.00", file=DS)
print("END OF EARTH QUARE", file=DS)
print("", file=DS)
#Hier komt de Soil profile
print("SOIL PROFILES", file=DS)
print("1 Number of spring characteristics curves", file=DS)
print("0 1/0 : Yes/No Inclination curve", file=DS)
print("0 1/0 : Yes/No Use Our values", file=DS)
print("", file=DS)
print("2 Number of soil profiles", file=DS)
#soil profile 1
print("New Profile", file=DS)
print("0.00 X coordinate", file=DS)
print("0.000 Y coordinate", file=DS)
print("", file=DS)
print("number of layers", "Number of soil layers per soil profile",
file=DS)
print("Nr Level Wosp Top Wosp Bott Material Name", file=DS)
print("Nr layer1, toplevel layer1, Wosp Top layer1, Wosp bott layer1,
name soillayer 1, file=DS)
if number of layers > 1:
print("Nr layer2, toplevel layer2, Wosp Top layer2, Wosp bott layer2,

if top under floor 3 >= lowest layer:
print("1", top under floor 3, (bottom structure-waterlevel))-
unit weight of water, (bottom structure-waterlevel)*unit weight of water, Underfloor 3,
file=DS)
if number of layers > 3:
if top under floor 4 >= lowest layer:
print("2", top under floor 4, (bottom structure-
waterlevel))*unit weight of water (bottom structure-waterlevel)*unit weight of water,
Underfloor 4, file=DS)
if number of layers > 4:
if top under floor 5 >= lowest layer:
print("3", top under floor 5, (bottom structure -
waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -unit weight of water,
Underfloor 5, file=DS)
if number of layers > 5:
if top under floor 6 >= lowest layer:
print("4", top under floor 6, (bottom structure -
waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -unit weight of water,
Underfloor 6, file=DS)
if number of layers > 6:
if top under floor 7 >= lowest layer:
print("5", top under floor 7, (bottom structure -
waterlevel) * -unit weight of water, (bottom structure - waterlevel) * -unit weight of water,
Underfloor 7, file=DS)
if number of layers > 7:
if top under floor 8 >= lowest layer:
print("6", top under floor 8,
(bottom structure - waterlevel) * unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 8, file=DS)
if number of layers > 8:
if top under floor 9 >= lowest layer:
print("7", top under floor 9,
(bottom structure - waterlevel) * unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 9, file=DS)
if number of layers > 9:
if top under floor 10 >= lowest layer:
print("8", top under floor 10,
(bottom structure - waterlevel) * unit weight of water, (bottom structure - waterlevel) * -
unit weight of water, Underfloor 10, file=DS)
#soil profile 3
print("Back fill", file=DS)
print("0.00 X coordinate", file=DS)
print("0.000 Y coordinate", file=DS)
print("", file=DS)
print("number of layers", "Number of soil layers per soil profile",
file=DS)
print("Nr Level Wosp Top Wosp Bott Material Name", file=DS)
print("1, toplevel layer backfill, Wosp Top layer backfill,
Wosp bott layer backfill,
Name Layer backfill, file=DS)
print("Nr layer1 + 1, toplevel layer1, Wosp Top layer1, Wosp bott layer1,
name soillayer 1, file=DS)
if number of layers > 1:
print("Nr layer2 + 1, toplevel layer2, Wosp Top layer2, Wosp bott layer2,
file=DS)
if number of layers > 2:
print("Nr layer3 + 1, toplevel layer3, Wosp Top layer3,
Wosp bott layer3, name soillayer 3,
file=DS)
if number of layers > 3:
print("Nr layer4 + 1, toplevel layer4, Wosp Top layer4,
Wosp bott layer4,
name soillayer 4, file=DS)
if number of layers > 4:
print("Nr layer5 + 1, toplevel layer5, Wosp Top layer5,
Wosp bott layer5,
name soillayer 5, file=DS)
if number of layers > 5:
print("Nr layer6 + 1, toplevel layer6, Wosp Top layer6,
Wosp bott layer6,
name soillayer 6, file=DS)
if number of layers > 6:

name soillayer 2, file=DS)
if number of layers > 2:
print("Nr layer3, toplevel layer3, Wosp Top layer3, Wosp bott layer3,
name soillayer 3,
file=DS)
if number of layers > 3:
print("Nr layer4, toplevel layer4, Wosp Top layer4,
Wosp bott layer4,
name soillayer 4, file=DS)
if number of layers > 4:
print("Nr layer5, toplevel layer5, Wosp Top layer5,
Wosp bott layer5,
name soillayer 5, file=DS)
if number of layers > 5:
print("Nr layer6, toplevel layer6, Wosp Top layer6,
Wosp bott layer6,
name soillayer 6, file=DS)
if number of layers > 6:
print("Nr layer7, toplevel layer7, Wosp Top layer7, Wosp bott layer7,
name soillayer 7, file=DS)
if number of layers > 7:
print("Nr layer8, toplevel layer8,
Wosp Top layer8, Wosp bott layer8,
name soillayer 8, file=DS)
if number of layers > 8:
print("Nr layer9 + 1, toplevel layer9,
Wosp Top layer9, Wosp bott layer9,
name soillayer 9, file=DS)
if number of layers > 9:
print("Nr layer10 + 1, toplevel layer10,
Wosp Top layer10, Wosp bott layer10,
name soillayer 10, file=DS)

print("SURFACES", file=DS)
print("4 Number of surfaces", file=DS)
print("1 name groundlevel, file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("No X-coord Value", file=DS)
print("1 0.00 begin groundlevel, file=DS)
print("2 1 name second surface, file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("1 0.00 7, Floor level lock, file=DS)
print("1 bottom of structure", file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("No X-coord Value", file=DS)
print("1 0.00 bottom structure, file=DS)
print("4 1 After Back Fill", file=DS)
print("0.00 Standard deviation", file=DS)
print("2 Distribution type", file=DS)
print("Nr X-coord Value", file=DS)
print("1 0.00 7, Top of structure tow NAP, file=DS)
print("END OF SURFACES", file=DS)
print("", file=DS)
#check voor water levels
print("WATERLEVELS", file=DS)
print("3 Number of Waterlevels", file=DS)
print("3 Number of Data per Waterlevel", file=DS)
print("name waterlevel1, file=DS)
print("waterlevel1, file=DS) #level van het water
print("0.00", file=DS)
print("27", file=DS)
print("name waterlevel2, file=DS) # naam van waterlevel
print("waterlevel2, file=DS) # level van het water
print("0.00", file=DS)
print("24", file=DS)
print("name waterlevel3, file=DS) # naam van waterlevel
print("waterlevel3, file=DS) # level van het water
print("0.00", file=DS)
print("2", file=DS)
print("", file=DS)
print("END OF WATERLEVELS", file=DS)
print("CONSTRUCTION STAGES", file=DS)
print("4 Number of Construction stages", file=DS)
print("New Stage", file=DS)
print("1 1 Method Left: Ka, Ko, Rp Method Right: Ka, Ko,
Rp",
file=DS) # welke method wordt gebruikt voor wat
print("bottom of structure", file=DS) # Naam van groundlevel links

```

Figure F-17: Code Diaphragm Wall Structure Input (2)

```

print(name_groundlevel, file=DS) # Naam grondlevel rechts
print(name_waterlevel1, file=DS) # Naam grondlevel links
print(name_waterlevel2, file=DS) # Naam waterlevel rechts
print(sollprofile_right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage",
file=DS)
print(" 1 0.00", Name_anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 Status present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 1 Uniform loads in stage", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 2:
print("New Stage 2", file=DS)
print(" 1 1 Method Left: Ka, Ko, Kp Method Right: Ka, Ko,
Kp", file=DS)

print(name_second_surface, file=DS) # Naam van grondlevel links
print(name_groundlevel2, file=DS) # Naam grondlevel rechts
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(sollprofile_right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage",
file=DS)
print(" 1 0.00", Name_anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)
print(" 0 Status present in stage", file=DS)
print(" 0 Spring supports present in stage", file=DS)
print(" 0 Rigid supports present in stage", file=DS)
print(" 2 Uniform loads in stage", file=DS)
print(" 1 concrete floor load", file=DS)
print(" 1 surcharge load", file=DS)
print(" 0 Surcharge loads in stage", file=DS)
print(" 0 Horizontal line loads in stage", file=DS)
print(" 0 Moments in stage", file=DS)
print(" 0 Normal forces in stage", file=DS)
print(" 0 Passive side overrule", file=DS)
print("", file=DS)
# hier komt stage 3:
print("New Stage 3", file=DS)
print(" 1 1 Method Left: Ka, Ko, Kp Method Right: Ka, Ko,
Kp", file=DS)

print(name_second_surface, file=DS) # Naam van grondlevel links
print(name_groundlevel3, file=DS) # Naam grondlevel rechts
print(name_waterlevel1, file=DS) # Naam waterlevel rechts
print("Underwater concrete", file=DS)
print(sollprofile_right, file=DS)
print(" 0.000 Displacement top sheet piling", file=DS)
print(" 0 Use top displacement", file=DS)
print(" ", number_of_anchors[1], " Anchors present in
stage",
file=DS)
print(" 1 0.00", Name_anchor1, file=DS)
if number_of_anchors[1] == 2:
    print(" 2 0.00", Name_anchor2, file=DS)

# stage 0
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("END OF STAGE", file=DS)
# stage 1
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("END OF STAGE", file=DS)
# stage 2
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("END OF STAGE", file=DS)
# stage 3
print("STAGE", file=DS)
print("StagePartialFactorSetCUR=0", file=DS)
print("StageVerify=0", file=DS)
print("StageAnchorFactor=1.000", file=DS)
print("StagePartialFactorSetECTNADNL=0", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("StageVerifyECTNADNL=0", file=DS)
print("StageAnchorFactorECTNADNL=1.000", file=DS)
print("END OF STAGE", file=DS)
print("END OF CALCULATION OPTIONS PER STAGE", file=DS)
print("", file=DS)

thisFile = "DaheetinputD.txt"
base = os.path.splitext(thisFile)[0]
# os.remove(base + ".str(ddd) + str(ddd) + ".SHI")
os.rename(thisFile, base + ".str(ddd) + str(ddd) + ".SHI")
input_file Daheet()
    
```

Figure F-18: Code Diaphragm Wall Structure Input (3)


```

        floor_height_negative_moment = floor_height_U_basin[f]

        Upward water pressure = (Height - Top of structure tov NAP + waterlevel +
        floor_height_U_basin[f]) * unit_weight_of_water
        print('The floor height', stage, 'by positive moment is',
        floor height positive moment, ' m',UC is', UC normalforce and shearforce)
        if UC normalforce_and_moment > 1.00:
            print ('Reinforcement floor not sufficient, check manually')
            break

        print (f'f')
        if minimum_moment_floor > 0:
            print ('break')
            break

        spacing_between_bars_concrete_bottom =
        min(spacing_between_bars_concrete_bottom_negative,spacing_between_bars_concrete_bottom_positive)
        spacing_between_bars_concrete_top =
        min(spacing_between_bars_concrete_top_negative,spacing_between_bars_concrete_top_positive)
        number_of_bars_concrete_bottom = 1000/ spacing_between_bars_concrete_bottom
        number_of_bars_concrete_top = 1000/ spacing_between_bars_concrete_top
        A_reinforcement_concrete_floor_bottom = number_of_bars_concrete_bottom *
        ((reinforcement_diameter ** 2) * 0.25 * np.pi)
        A_reinforcement_concrete_floor_top = number_of_bars_concrete_top * ((reinforcement_diameter
        ** 2) * 0.25 * np.pi)

        if minimum_moment_floor < 0:
            floor_height = max(floor height negative moment,floor height positive moment)
        else:
            floor_height = floor_height_positive_moment

        #floor_height_U_basin_start = floor_height
        print ('Floor thickness',stage,floor height)

        #xl = win32com.client.Dispatch("Excel.Application")
        #xl.Application.Quit() # Comment this out if your excel script closes
        # xl.DisplayAlerts = False # werkt nog niet
        # xl.Application.Quit() # Comment this out if your excel script closes

    return(floor_height,spacing_between_bars_concrete_bottom,spacing_between_bars_concrete_top,Upward
    water pressure,A_reinforcement_concrete_floor_bottom,A_reinforcement_concrete_floor_top)

```

Figure F-23: Code U-Basin Floor (2)

```

def wall_thickness_calculation():
    #Max moment in wall
    #moment calculation ubasin('dry chamber')[1][steps],moment calculation ubasin('fill')[1][steps],mo
    #ment calculation ubasin('fill')[1][steps]
    shearforce_bottom_wall =
    #moment calculation ubasin('dry chamber')[0],moment calculation ubasin('fill')[0],moment calculati
    #on ubasin('fill')[0]
    f_yk_reinforcement_steel = sht.range('O12').value #met nog iets worden gedaan met materiaal
    #data's
    steps_wall_thickness = np.linspace(0,3000,81)
    reinforcement_diameter = 16 #mm
    UC_normalforce_and_shearforce = 0
    wall_thickness_bottom=wall_thickness_bottom_start*(len(steps_wall_thickness))

    sht.range('D51').clear_contents()
    sht.range('F51').clear_contents()
    sht.range('D53').clear_contents()
    sht.range('F53').clear_contents()
    sht.range('D55').clear_contents()
    sht.range('F55').clear_contents()
    sht.range('E15').value = [1000] # All the forces are calculated per meter width
    moment_in_wall_near_floor = max(max(moment_in_wall), -min(max(moment_in_wall)))
    sht.range('F77').value = [moment_in_wall_near_floor]
    sht.range('F83').value = [moment_in_wall_near_floor]

    maximum_shearforce_bottom_wall = max(shearforce_bottom_wall)
    sht.range('F82').value = [abs(maximum_shearforce_bottom_wall)]

    for i in range(len(steps_wall_thickness)):
        if UC_normalforce_and_shearforce == '':
            wall_thickness_bottom[i] = wall_thickness_bottom_start + steps_wall_thickness[i] /
1000
        else:
            if UC_normalforce_and_shearforce <= 1.00:
                wall_thickness_bottom[i] = wall_thickness_bottom_start +
steps_wall_thickness[i]/1000
            else:
                wall_thickness_bottom[i] = wall_thickness_bottom_start + steps_wall_thickness[i] /
1000

    wall_thickness_top = wall_thickness_bottom[i] * (1/2)
    if wall_thickness_top < 0.5:
        wall_thickness_top = 0.5

    print (wall_thickness_bottom[i],"process wall bottom")
    # Check thickness wall near floor
    sht.range('E16').value = [wall_thickness_bottom[i]*1000]

    effective_height_concrete = wall_thickness_bottom[i]*1000 - concrete_cover - 0.5 *
reinforcement_diameter
    A_reinforcement_wall = moment_in_wall_near_floor*(1000000) / (f_yk_reinforcement_steel *
0.9 * effective_height_concrete)
    number_of_bars = math.ceil(A_reinforcement_wall /
((reinforcement_diameter**2)*0.25*np.pi))
    spacing_between_bars = 1000 / number_of_bars

    # Normal force wall
    normalforce_wall = np.zeros(steps+1)
    for ii in range(steps+1):
        normalforce_wall[ii] = (z[ii] - Top_of_structure_tov_NAP) * (wall_thickness_top +
((wall_thickness_bottom[ii]-wall_thickness_top) / -Height * (z[ii] - Top_of_structure_tov_NAP))/2)
        # density concrete
        normalforce_wall_above_soil_dry = -(wall_thickness_bottom[ii]-wall_thickness_top)/Height*
(Top_of_structure_tov_NAP-waterlevel)**2 * 0.5 + (Top_of_structure_tov_NAP-waterlevel) *
((wall_thickness_bottom[ii]-wall_thickness_top)-wall_thickness_bottom[i])

```

```

input_gamma_wet_soil_backfill
    normalforce_wall_near_bottom = min(normalforce_wall) + normalforce_wall_above_soil_dry +
normalforce_wall_above_soil_wet

    sht.range('F76').value = [normalforce_wall_near_bottom]
    sht.range('F84').value = [normalforce_wall_near_bottom]

    sht.range('F57').value = [spacing_between_bars]

    if os.path.exists("7-Normalkracht_en_Buiging_EC2_v2.8.xlsx"):
        xl=win32com.client.Dispatch("Excel.Application")
        xl.Workbooks.Open(os.path.abspath("7-Normalkracht_en_Buiging_EC2_v2.8.xlsx"))
        xl.Application.Run("BerekenUGT")
        #xl.Application.Save() # if you want to save then uncomment this line and change
        # delete the # ReadyOnly's part from the open function.
        # xl.DisplayAlerts = False #werkt nog niet
        # xl.Application.Quit() # Comment this out if your excel script closes
        del xl

    UC_normalforce_and_moment = sht.range('O77').value
    UC_normalforce_and_shearforce = sht.range('O56').value
    print(wall_thickness_bottom[i], "process wall bottom", UC_normalforce_and_shearforce)
    if UC_normalforce_and_shearforce == '':
        print(wall_thickness_bottom[i], " ")
        if wall_thickness_bottom[i] < wall_thickness_bottom[i - 1]:
            wall_thickness_bottom = wall_thickness_bottom[i - 1]
            print (wall_thickness_bottom, "kias")
            if UC_normalforce_and_moment > 1.00:
                print ('Reinforcement wall not sufficient, check manually')
                break
    elif UC_normalforce_and_shearforce <= 1.00:
        print (wall_thickness_bottom[i],UC_normalforce_and_shearforce)
        if wall_thickness_bottom[i] == 0.5:
            print (wall_thickness_bottom)
            wall_thickness_bottom = wall_thickness_bottom[i]
            print(wall_thickness_bottom, "kias")
        if UC_normalforce_and_moment > 1.00:
            print ('Reinforcement wall not sufficient, check manually')
            break
    else:
        if wall_thickness_bottom[i] < wall_thickness_bottom[i - 1]:
            wall_thickness_bottom = wall_thickness_bottom[i - 1]
            print (wall_thickness_bottom, "kias")
            if UC_normalforce_and_moment > 1.00:
                print ('Reinforcement wall not sufficient, check manually')
                break
        else:
            print ('Reinforcement wall not sufficient, check manually')
            break

    xl = win32com.client.Dispatch("Excel.Application")
    xl.DisplayAlerts = False #werkt nog niet

return(wall_thickness_bottom,wall_thickness_top,normalforce_wall_near_bottom,A_reinforcement_wall)

```

Figure F-24: Code U-Basin Walls


```
weight_of_U_basin_floor = (floor_height_U_basin_max) * (Width + 2 * wall_thickness_bottom) *  
density_concrete # (kN/m)  
weight_of_U_basin_walls = -normalforce_wall_near_bottom * 2# (kN/m)  
weight_of_U_basin = weight_of_U_basin_floor + weight_of_U_basin_walls  
  
if Dewatering_chamber == "Yes": #including safety factors  
    netto_vertical_force_U_basin = (weight_of_U_basin * 0.9 - Upward_water_pressure * (Width + 2  
    * wall_thickness_bottom)**1.2) * Length  
else:  
    netto_vertical_force_U_basin = (weight_of_U_basin * 0.9 - Upward_water_pressure * (Width + 2  
    * wall_thickness_bottom)**1.2 + (-moment_calculation_ubasin('LLL')[14]-1*unit_weight_of_water) *  
    (Width + 2 * wall_thickness_bottom)**1.2) * Length  
  
if netto_vertical_force_U_basin < 0:  
    extra_thickness_U_basin_floor = -netto_vertical_force_U_basin / Length / density_concrete /  
    (Width + 2 * wall_thickness_bottom)  
    h_o_h_afstand_lange_u_basin = tension_piles(Top_of_structure_tov_NAP-Height-  
    floor_height_U_basin_max,netto_vertical_force_U_basin, (Width + 2 * wall_thickness_bottom))[1]  
    h_o_h_afstand_korte_u_basin = tension_piles(Top_of_structure_tov_NAP-Height-  
    floor_height_U_basin_max,netto_vertical_force_U_basin, (Width + 2 * wall_thickness_bottom))[2]  
    number_of_piles_needed_u_basin = tension_piles(Top_of_structure_tov_NAP-Height-  
    floor_height_U_basin_max,netto_vertical_force_U_basin, (Width + 2 * wall_thickness_bottom))[5]  
else:  
    number_of_piles_needed_u_basin = 0  
    extra_thickness_U_basin_floor = 0
```

Figure F-25: Code U-Basin Piles

```
def output_dimensions():
    ##### GENERAL #####
    output_file_dimensions_general = pd.DataFrame({'Length': [Length], 'Width': [Width], 'Lowest_Lock_Level': [Lowest_Lock_Level], 'Bottom_
    structure': [Bottom_Structure], 'number_of_calculations': [number_of_calculations]})
    output_file_dimensions_general =
    output_file_dimensions_general.assign(range_length_pile=range_length_pile, range_number_of_anchor=
    range_number_of_anchor, range_sheepile=range_sheepile)
    output_file_dimensions_general =
    output_file_dimensions_general.assign(range_length_diaphragm_wall=range_length_diaphragm_wall, range
    number_of_anchor_diaphragm_wall=range_number_of_anchor_diaphragm_wall, range_thickness_diaphrag
    m_wall=range_thickness_diaphragm_wall)
    output_file_dimensions_general =
    output_file_dimensions_general.assign(veasion_check=veasion_check)
    ##### SHEET PILE #####
    mass_sheepile_per_meter = sheepile_length*sectional_area_sheepile/10000*density_steel #
    mass_sheepile = mass_sheepile_per_meter * Length * 2
    surface_sheepile = sheepile_length * Length * 2
    output_file_dimensions_sheepile =
    output_file_dimensions_sheepile.assign(mass_sheepile_per_meter=mass_sheepile_per_meter, mass_she
    epile_total=mass_sheepile_total, surface_sheepile=surface_sheepile)
    volume_placing_backfill = excavation_steep_pit[Width][0]
    volume_import_backfill_soil = excavation_steep_pit[Width][2]
    volume_export_sand = excavation_steep_pit[Width][1]
    volume_export_peat = excavation_steep_pit[Width][4]
    volume_export_clay = excavation_steep_pit[Width][5]
    volume_export_total = volume_export_sand + volume_export_peat + volume_export_clay
    excavation_volume = excavation_steep_pit[Width][3]
    volume_re_used_sand = excavation_steep_pit[Width][6]
    output_file_dimensions_sheepile =
    output_file_dimensions_sheepile.assign(volume_placing_backfill=volume_placing_backfill, volume_in
    port_backfill_soil=volume_import_backfill_soil, volume_export_sand=volume_export_sand, volume_export_
    eat=volume_export_peat, volume_export_sand=volume_export_sand, volume_export_total=volume_export
    _total, excavation_volume=excavation_volume, volume_re_used_sand=volume_re_used_sand)
    volume_underwater_concrete_floor_total =
    results_Daheet2_of_U_underwaterconcretefloor_all/1000 * Width * Length
    output_file_dimensions_underwater_concrete_floor_total =
    output_file_dimensions_underwater_concrete_floor_total.assign(volume_underwater_concrete_floor_total=volume_underwater
    concrete_floor_total)
    total_length_tension_piles = [Length_tension_pile * number_of_piles_needed]
    (range_length_pile*range_number_of_anchor*range_sheepile)
    total_volume_tension_piles = [Length_tension_pile * number_of_piles_needed *
    (B_pile/1000)**2 * (range_length_pile*range_number_of_anchor*range_sheepile)]
    output_file_dimensions_tension_piles =
    output_file_dimensions_tension_piles.assign(total_length_tension_piles=total_length_tension_piles, tot
    al_volume_tension_piles=total_volume_tension_piles)
    distance_between_anchors = maximum_anchor_force /
    np.array(np.max([anchor_force_1_stage_1, anchor_force_1_stage_2, anchor_force_1_stage_3, anchor_fo
    rce_1_stage_4]), axis=0)
    number_of_anchors = Length * 2 / distance_between_anchors
    output_file_dimensions_anchors =
    output_file_dimensions_anchors.assign(number_of_anchors=number_of_anchors)
    output_file_dimensions_sheepile =
    output_file_dimensions_sheepile.assign(sheepile_couting_area=Sheepile_couting_area)
    ##### DIAPHRAGM WALL #####
    output_file_dimensions_diaphragm_wall = results_Daheet2_d
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(reinforcement_per_quad_meter=reinforcement

A_reinforcement_concrete_floor_top/1000000 * (Width + 2 * wall_thickness_bottom) * Length * 1.2
m^3 incl secondary reinforcement)
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(volume_reinforcement_walls=volume_reinforcement_wa
    lls, volume_re_used_sand_U_basin_open_pit=volume_re_used_sand_U_basin_open_pit)
    total_length_tension_piles_U_basin = [Length_tension_pile * number_of_piles_needed_U_basin]
    (B_pile/1000)**2 * 2]
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(total_length_tension_piles_U_basin=total_length_ten
    sion_piles_U_basin, total_volume_tension_piles_U_basin=total_volume_tension_piles_U_basin)
    volume_placing_backfill_U_basin_open_pit_with_piles =
    (backfill_sand(excavation_soil(bottom_excavation_with_piles)[0], excavation_soil(bottom_excavatio
    n_without_piles)[1], bottom_excavation_without_piles)[2])
    volume_import_backfill_U_basin_open_pit_with_piles =
    (backfill_sand(excavation_soil(bottom_excavation_with_piles)[0], excavation_soil(bottom_excavatio
    n_without_piles)[1], bottom_excavation_without_piles)[0])
    volume_export_sand_U_basin_open_pit_with_piles =
    backfill_sand(excavation_soil(bottom_excavation_with_piles)[0], excavation_soil(bottom_excavatio
    n_without_piles)[1], bottom_excavation_without_piles)[1]
    volume_export_peat_U_basin_open_pit_with_piles =
    excavation_soil(bottom_excavation_with_piles)[2]
    volume_export_clay_U_basin_open_pit_with_piles =
    excavation_soil(bottom_excavation_with_piles)[5]
    volume_export_total_U_basin_open_pit_with_piles =
    volume_export_sand_U_basin_open_pit_with_piles + volume_export_peat_U_basin_open_pit_with_piles +
    volume_export_clay_U_basin_open_pit_with_piles
    volume_excavation_costs_U_basin_open_pit_with_piles =
    excavation_soil(bottom_excavation_with_piles)[0], excavation_soil(bottom_excavation
    without_piles)[1], bottom_excavation_without_piles)[3]
    volume_re_used_sand_U_basin_open_pit_with_piles =
    backfill_sand(excavation_soil(bottom_excavation_with_piles)[0], excavation_soil(bottom_excavatio
    n_without_piles)[1], bottom_excavation_without_piles)[3]
    volume_placing_backfill_U_basin_open_pit_without_piles =
    (backfill_sand(excavation_soil(bottom_excavation_without_piles)[0], excavation_soil(bottom_excavati
    on_without_piles)[1], bottom_excavation_without_piles)[0])
    volume_import_backfill_U_basin_open_pit_without_piles =
    (backfill_sand(excavation_soil(bottom_excavation_without_piles)[0], excavation_soil(bottom_excavati
    on_without_piles)[1], bottom_excavation_without_piles)[0])
    volume_export_sand_U_basin_open_pit_without_piles =
    backfill_sand(excavation_soil(bottom_excavation_without_piles)[0], excavation_soil(bottom_excavati
    on_without_piles)[1], bottom_excavation_without_piles)[1]
    volume_export_peat_U_basin_open_pit_without_piles =
    excavation_soil(bottom_excavation_without_piles)[2]
    volume_export_clay_U_basin_open_pit_without_piles =
    excavation_soil(bottom_excavation_without_piles)[5]
    volume_export_total_U_basin_open_pit_without_piles =
    volume_export_sand_U_basin_open_pit_without_piles + volume_export_peat_U_basin_open_pit_without_piles +
    volume_export_clay_U_basin_open_pit_without_piles
    volume_excavation_costs_U_basin_open_pit_without_piles =
    excavation_soil(bottom_excavation_without_piles)[0], excavation_soil(bottom_excavation
    without_piles)[1], bottom_excavation_without_piles)[3]
    volume_re_used_sand_U_basin_open_pit_without_piles =
    backfill_sand(excavation_soil(bottom_excavation_without_piles)[0], excavation_soil(bottom_excavati
    on_without_piles)[1], bottom_excavation_without_piles)[3]
    volume_placing_backfill_U_basin_open_pit =
    (volume_placing_backfill_U_basin_open_pit_with_piles, volume_placing_backfill_U_basin_open_pit_wit
    hout_piles)
    volume_import_backfill_U_basin_open_pit =
    (volume_import_backfill_U_basin_open_pit_with_piles, volume_import_backfill_U_basin_open_pit_wit
    hout_piles)
    volume_export_sand_U_basin_open_pit =
    (volume_export_sand_U_basin_open_pit_with_piles, volume_export_sand_U_basin_open_pit_without_piles)
    volume_export_peat_U_basin_open_pit =
    (volume_export_peat_U_basin_open_pit_with_piles, volume_export_peat_U_basin_open_pit_without_piles)
    volume_export_clay_U_basin_open_pit =
    (volume_export_clay_U_basin_open_pit_with_piles, volume_export_clay_U_basin_open_pit_without_piles)

per_quad_meter_concrete)
    volume_underwater_concrete_floor_total_d =
    results_Daheet2_d_of_U_underwaterconcretefloor_all/1000 * Width * Length
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(volume_underwater_concrete_floor_total_d=volume_und
    erwater_concrete_floor_total_d)
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(number_of_bars=number_of_bars)
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(reinforcement_diameter=reinforcement_diameter)
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(width_diaphragm_wall=width_diaphragm_wall)
    distance_between_anchors_d = maximum_anchor_force /
    np.array(np.max([anchor_force_1_stage_1_d, anchor_force_1_stage_2_d, anchor_force_1_stage_3_d, anch
    or_force_1_stage_4_d]), axis=0)
    number_of_anchors_d = Length * 2 / distance_between_anchors_d
    output_file_dimensions_diaphragm_wall =
    output_file_dimensions_diaphragm_wall.assign(number_of_anchors_d=number_of_anchors_d)
    ##### U-BASIN OPEN PIT #####
    U_basin_option = ['With Tension Piles', 'Without Tension Piles']
    output_file_dimensions_U_basin_open_pit = pd.DataFrame({'U_Basin_option': U_Basin_option})
    floor_height_U_basin = [floor_height, floor_height]
    wall_thickness_bottom_U_basin = [wall_thickness_bottom, wall_thickness_bottom]
    wall_thickness_top_U_basin = [wall_thickness_top, wall_thickness_top]
    extra_thickness_U_basin = [0, extra_thickness_U_basin]
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(floor_height_U_basin=floor_height_U_basin, wall_thi
    ckness_bottom_U_basin=wall_thickness_bottom_U_basin, wall_thickness_top_U_basin=wall_thickness_top
    _U_basin, extra_thickness_U_basin=extra_thickness_U_basin)
    volume_of_U_basin_floor_with_piles = (floor_height) * (Width + 2 * wall_thickness_bottom) *
    Length #m^3
    volume_of_U_basin_walls_with_piles = (wall_thickness_bottom - wall_thickness_top) * Height /
    2 + wall_thickness_top * Height / 2 * Length #m^3
    volume_of_U_basin_floor_without_piles = (floor_height + extra_thickness_U_basin) * (Width + 2 *
    wall_thickness_bottom) * Length #m^3
    volume_of_U_basin_walls_without_piles = (wall_thickness_bottom - wall_thickness_top) *
    Height / 2 + wall_thickness_top * Height / 2 * Length #m^3
    volume_of_U_basin_floor =
    (volume_of_U_basin_floor_with_piles, volume_of_U_basin_floor_without_piles)
    volume_of_U_basin_walls =
    (volume_of_U_basin_walls_with_piles, volume_of_U_basin_walls_without_piles)
    volume_of_U_basin_walls_with_piles = volume_of_U_basin_walls_without_piles +
    volume_of_U_basin_walls_with_piles, volume_of_U_basin_floor_without_piles +
    volume_of_U_basin_walls_without_piles)
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(volume_of_U_basin_floor=volume_of_U_basin_floor, vo
    lume_of_U_basin_walls=volume_of_U_basin_walls, volume_of_concrete_U_basin=volume_of_concrete_U_ba
    sin)
    wall_formwork = (Height + np.sqrt((wall_thickness_bottom -
    wall_thickness_top)**2 + (Height)**2)) * Length #m
    floor_formwork_with_piles = (Width + 2 * wall_thickness_bottom) * (floor_height + 2) *
    Length #m
    floor_formwork_without_piles = (Width + 2 * (wall_thickness_bottom +
    extra_thickness_U_basin_floor + floor_height)) * Length #m
    floor_formwork = (floor_formwork_with_piles, floor_formwork_without_piles)
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(wall_formwork=wall_formwork, floor_formwork=floor_f
    ormwork)
    volume_reinforcement_walls = A_reinforcement_walls/1000000 * (Height + floor_height) * 2 *
    Length * 1.2 #m^3 incl secondary reinforcement
    volume_reinforcement_floor = (A_reinforcement_concrete_floor_bottom +

I
    volume_export_total_U_basin_open_pit =
    (volume_export_total_U_basin_open_pit_with_piles, volume_export_total_U_basin_open_pit_without_pil
    es)
    volume_excavation_U_basin_open_pit =
    (volume_excavation_costs_U_basin_open_pit_with_piles, volume_excavation_costs_U_basin_open_pit_wit
    hout_piles)
    volume_re_used_sand_U_basin_open_pit =
    (volume_re_used_sand_U_basin_open_pit_with_piles, volume_re_used_sand_U_basin_open_pit_without_piles)
    output_file_dimensions_U_basin_open_pit =
    output_file_dimensions_U_basin_open_pit.assign(volume_placing_backfill_U_basin_open_pit=volume_pl
    acing_backfill_U_basin_open_pit, volume_import_backfill_U_basin_open_pit=volume_import_backfill_U
    _basin_open_pit, volume_export_sand_U_basin_open_pit=volume_export_sand_U_basin_open_pit, volume_
    export_peat_U_basin_open_pit=volume_export_peat_U_basin_open_pit, volume_export_clay_U_basin_ope
    n_pit=volume_export_clay_U_basin_open_pit, volume_export_total_U_basin_open_pit=volume_export_tot
    al_U_basin_open_pit, volume_excavation_U_basin_open_pit=volume_excavation_U_basin_open_pit, volum
    e_re_used_sand_U_basin_open_pit=volume_re_used_sand_U_basin_open_pit)
    ##### U-BASIN STEEP PIT #####
    output_file_dimensions_U_basin_steep_pit = results_Daheet2_u
    output_file_dimensions_U_basin_steep_pit =
    output_file_dimensions_U_basin_steep_pit.assign(sheepile_length_u=results_Daheet2_u.sheepile_la
    ngth_u, values)
    output_file_dimensions_U_basin_steep_pit =
    output_file_dimensions_U_basin_steep_pit.assign(sectional_area_sheepile_u=results_Daheet2_u.sect
    onal_area_sheepile_u, values)
    volume_placing_backfill_U_basin_steep_pit = excavation_steep_pit[Width][4][0]
    volume_import_backfill_U_basin_steep_pit = excavation_steep_pit[Width][4][2]
    volume_export_sand_U_basin_steep_pit = excavation_steep_pit[Width][4][1]
    volume_export_peat_U_basin_steep_pit = excavation_steep_pit[Width][4][5]
    volume_export_clay_U_basin_steep_pit = excavation_steep_pit[Width][4][5]
    volume_export_total_U_basin_steep_pit = volume_export_sand_U_basin_steep_pit +
    volume_export_peat_U_basin_steep_pit + volume_export_clay_U_basin_steep_pit
    excavation_volume_U_basin_steep_pit = excavation_steep_pit[Width][3][0]
    volume_re_used_sand_U_basin_steep_pit = excavation_steep_pit[Width][4][6]
    output_file_dimensions_U_basin_steep_pit =
    output_file_dimensions_U_basin_steep_pit.assign(volume_placing_backfill_U_basin_steep_pit=volume_pl
    acing_backfill_U_basin_steep_pit, volume_import_backfill_U_basin_steep_pit=volume_import_ba
    cking_backfill_U_basin_steep_pit, volume_export_sand_U_basin_steep_pit=volume_export_sand_U_basi
    n_steep_pit, volume_export_peat_U_basin_steep_pit=volume_export_peat_U_basin_steep_pit, volum
    e_export_clay_U_basin_steep_pit=volume_export_clay_U_basin_steep_pit, volume_export_total_U_basi
    n_steep_pit=volume_export_total_U_basin_steep_pit, excavation_volume_U_basin_steep_pit=excavation_vo
    lume_U_basin_steep_pit, volume_re_used_sand_U_basin_steep_pit=volume_re_used_sand_U_basin_steep_pit)
    volume_underwater_concrete_floor_total_U =
    results_Daheet2_u_of_U_underwaterconcretefloor_all/1000 * (Width+4) * Length
    output_file_dimensions_U_basin_steep_pit =
    output_file_dimensions_U_basin_steep_pit.assign(volume_underwater_concrete_floor_total_u=volume_u
    nderwater_concrete_floor_total_u)
    number_of_piles_needed_U_basin_steep_pit = tension_piles(bottom_structure-
    results_Daheet2_u_of_U_underwaterconcretefloor_all/values[0]/1000, netto_vertical_force_u_basin_st
    eep_pit, Width+4)[3]
    total_length_tension_piles_U_basin_steep_pit = [Length_tension_pile *
    number_of_piles_needed_U_basin_steep_pit]
    (range_length_pile*range_number_of_anchor*range_sheepile)
    total_volume_tension_piles_U_basin_steep_pit = [Length_tension_pile *
    number_of_piles_needed_U_basin_steep_pit * (B_pile/1000)**2]
    (range_length_pile*range_number_of_anchor*range_sheepile)
    output_file_dimensions_U_basin_steep_pit =
    output_file_dimensions_U_basin_steep_pit.assign(total_length_tension_piles_U_basin_steep_pit=total
    _length_tension_piles_U_basin_steep_pit, total_volume_tension_piles_U_basin_steep_pit=total
    _volume_tension_piles_U_basin_steep_pit)
    distance_between_anchors_u = maximum_anchor_force /
    np.array(np.max([anchor_force_1_stage_1_u, anchor_force_1_stage_2_u, anchor_force_1_stage_3_u, anch
    or_force_1_stage_4_u]), axis=0)
    number_of_anchors_u = Length * 2 / distance_between_anchors_u
    output_file_dimensions_U_basin_steep_pit =
```

Figure F-26: Code Output Dimensions (1)

```
output_file_dimensions_U_basin_steep_pit.assign(number_of_anchors*number_of_anchors_U)

# number_of_anchors = np.array((Length * 2 /
distance_between_anchors)/(cage_length_pile*range_number_of_anchor*range_sheetpile))
# output file dimensions U basin steep pit =
output_file_dimensions_U_basin_steep_pit.assign(number_of_anchors*number_of_anchors)

#####
output_file_dimensions_Values_per_depth = Values_per_depth

with pd.ExcelWriter('output_dimensions.xlsx') as writer: # doctest: +SKIP
    output_file_dimensions_general.to_excel(writer, sheet_name='general')
    output_file_dimensions_sheetpile.to_excel(writer, sheet_name='sheetpile')
    output_file_dimensions_diaphragm_wall.to_excel(writer, sheet_name='diaphragm_wall')
    output_file_dimensions_U_basin_open_pit.to_excel(writer, sheet_name='U_basin_open_pit')
    output_file_dimensions_U_basin_steep_pit.to_excel(writer, sheet_name='U_basin_steep_pit')
    output_file_dimensions_Values_per_depth.to_excel(writer, sheet_name='Values_per_depth')
```

Figure F-27: Code Output Dimensions (2)

```

def def_input_all_excel(main_folder, folder):
    input_general = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'general')
    input_sheetpile = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'sheetpile')
    input_diaphragm_wall = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'diaphragm_wall')
    input_U_basin_open_pit = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'U_basin_open_pit')
    input_U_basin_steep_pit = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'U_basin_steep_pit')
    input_values_per_depth = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'values_per_depth')
    return
(input_general, input_sheetpile, input_diaphragm_wall, input_U_basin_open_pit, input_U_basin_steep_pit,
input_values_per_depth)

##### GENERAL #####
def def_input_general(input_general):
    dict_general = {}
    for col in input_general.columns.tolist():
        dict_general[col] = input_general.at[0, col]
    dict_general['length']
    length = input_general.Length[0]
    width = input_general.Width[0]
    Lowest Lock Level = input_general.Lowest Lock Level[0]
    bottom structure = input_general.bottom structure[0]
    range_length_pile = input_general.range_length_pile[0]
    range_number_of_anchor = input_general.range_number_of_anchor[0]
    range_sheetpile = input_general.range_sheetpile[0]
    range_length_diaphragm_wall = input_general.range_length_diaphragm_wall[0]
    range_number_of_anchor_diaphragm_wall =
input_general.range_number_of_anchor_diaphragm_wall[0]
    range_thickness_diaphragm_wall = input_general.range_thickness_diaphragm_wall[0]
    number_of_calculations = input_general.number_of_calculations[0]
    version_check = input_general.version_check[0]
    return(length,width,Lowest Lock Level,bottom structure,range_length_pile,range_number_of_anchor,
range_sheetpile,range_length_diaphragm_wall,range_number_of_anchor_diaphragm_wall,range_thickness
diaphragm_wall,number_of_calculations,version_check)

##### SHEETPILE #####
def def_input_sheetpile(input_sheetpile):
    sheetpile_length = input_sheetpile.sheetpile_length.values
    sectional_area_sheetpile = input_sheetpile.sectional_area_sheetpile.values
    volume_placing_backfill = input_sheetpile.volume_placing_backfill.values
    volume_import_backfill_soil = input_sheetpile.volume_import_backfill_soil.values
    volume_export_sand = input_sheetpile.volume_export_sand.values
    volume_export_peat = input_sheetpile.volume_export_peat.values
    volume_export_clay = input_sheetpile.volume_export_clay.values
    volume_export_total = input_sheetpile.volume_export_total.values
    excavation_volume = input_sheetpile.excavation_volume.values
    volume_reused_sand = input_sheetpile.volume_reused_sand.values
    volume_underwater_concrete_floor_total =
input_sheetpile.volume_underwater_concrete_floor_total.values
    total_length_tension_piles = input_sheetpile.total_length_tension_piles.values
    total_volume_tension_piles = input_sheetpile.total_volume_tension_piles.values
    number_of_anchors = input_sheetpile.number_of_anchors.values
    Sheetpile_couting_area = input_sheetpile.sheetpile_couting_area.values
    volume_concrete_in_Keap = input_sheetpile.df.volume_concrete_in_Keap.values
    volume_reinforcement_in_Keap = input_sheetpile.df.volume_reinforcement_in_Keap.values
    amount_formwork_in_Keap = input_sheetpile.df.amount_formwork_in_Keap.values

    floor_formwork_volume_reinforcement_walls , volume_reinforcement_floor
    volume_placing_backfill_U_basin_open_pit, volume_import_backfill_U_basin_open_pit,
    volume_export_sand_U_basin_open_pit, volume_export_peat_U_basin_open_pit,
    volume_export_clay_U_basin_open_pit, volume_export_total_U_basin_open_pit,
    volume_excavation_U_basin_open_pit, volume_reused_sand_U_basin_open_pit,
    total_length_tension_piles_u_basin, total_volume_tension_piles_u_basin)

##### U-BASIN STEEP PIT #####
def def_input_U_basin_steep_pit(input_U_basin_steep_pit):
    sheetpile_length_u = input_U_basin_steep_pit.sheetpile_length_u.values
    sectional_area_sheetpile_u = input_U_basin_steep_pit.sectional_area_sheetpile_u.values
    number_of_anchors_u = input_U_basin_steep_pit.number_of_anchors_u.values
    volume_placing_backfill_U_basin_steep_pit =
input_U_basin_steep_pit.volume_placing_backfill_U_basin_steep_pit.values
    volume_import_backfill_soil_U_basin_steep_pit =
input_U_basin_steep_pit.volume_import_backfill_soil_U_basin_steep_pit.values
    volume_export_sand_U_basin_steep_pit =
input_U_basin_steep_pit.volume_export_sand_U_basin_steep_pit.values
    volume_export_peat_U_basin_steep_pit =
input_U_basin_steep_pit.volume_export_peat_U_basin_steep_pit.values
    volume_export_clay_U_basin_steep_pit =
input_U_basin_steep_pit.volume_export_clay_U_basin_steep_pit.values
    volume_export_total_U_basin_steep_pit =
input_U_basin_steep_pit.volume_export_total_U_basin_steep_pit.values
    excavation_volume_U_basin_steep_pit =
input_U_basin_steep_pit.excavation_volume_U_basin_steep_pit.values
    volume_reused_sand_U_basin_steep_pit =
input_U_basin_steep_pit.volume_reused_sand_U_basin_steep_pit.values
    volume_underwater_concrete_floor_total_u =
input_U_basin_steep_pit.volume_underwater_concrete_floor_total_u.values
    total_length_tension_piles_U_basin_steep_pit =
input_U_basin_steep_pit.total_length_tension_piles_U_basin_steep_pit.values
    total_volume_tension_piles_U_basin_steep_pit =
input_U_basin_steep_pit.total_volume_tension_piles_U_basin_steep_pit.values
    return(sheetpile_length_u,sectional_area_sheetpile_u,number_of_anchors_u,volume_placing_backfill_
U_basin_steep_pit,volume_import_backfill_soil_U_basin_steep_pit,volume_export_sand_U_basin_steep
_pit,volume_export_peat_U_basin_steep_pit,volume_export_clay_U_basin_steep_pit,volume_export_total_U_b
asin_steep_pit,volume_reused_sand_U_basin_steep_pit,volume_underwater_concrete_floor_total_u,tot
al_length_tension_piles_U_basin_steep_pit,total_volume_tension_piles_U_basin_steep_pit)

return(sheetpile_length,sectional_area_sheetpile,volume_placing_backfill,volume_import_backfill_u
oil,volume_export_sand,volume_export_peat,volume_export_clay,volume_export_total,
excavation_volume,volume_reused_sand,volume_underwater_concrete_floor_total,
Sheetpile_couting_area,volume_concrete_in_Keap,volume_reinforcement_in_Keap,amount_formwork_in_Keap,
##### U-BASIN OPEN PIT #####
def def_input_U_basin_open_pit(input_U_basin_open_pit):
    U_basin_option = input_U_basin_open_pit.U_basin_option.values
    volume_of_U_basin_floor = input_U_basin_open_pit.volume_of_U_basin_floor.values
    volume_of_U_basin_walls = input_U_basin_open_pit.volume_of_U_basin_walls.values
    volume_of_concrete_U_basin = input_U_basin_open_pit.volume_of_concrete_U_basin.values
    wall_formwork = input_U_basin_open_pit.wall_formwork.values
    floor_formwork = input_U_basin_open_pit.floor_formwork.values
    volume_reinforcement_walls = input_U_basin_open_pit.volume_reinforcement_walls.values
    volume_reinforcement_floor = input_U_basin_open_pit.volume_reinforcement_floor.values
    volume_placing_backfill_U_basin_open_pit =
input_U_basin_open_pit.volume_placing_backfill_U_basin_open_pit.values
    volume_import_backfill_U_basin_open_pit =
input_U_basin_open_pit.volume_import_backfill_U_basin_open_pit.values
    volume_export_sand_U_basin_open_pit =
input_U_basin_open_pit.volume_export_sand_U_basin_open_pit.values
    volume_export_peat_U_basin_open_pit =
input_U_basin_open_pit.volume_export_peat_U_basin_open_pit.values
    volume_export_clay_U_basin_open_pit =
input_U_basin_open_pit.volume_export_clay_U_basin_open_pit.values
    volume_export_total_U_basin_open_pit =
input_U_basin_open_pit.volume_export_total_U_basin_open_pit.values
    volume_reused_sand_U_basin_open_pit =
input_U_basin_open_pit.volume_reused_sand_U_basin_open_pit.values
    total_length_tension_piles_u_basin =
input_U_basin_open_pit.total_length_tension_piles_u_basin.values
    total_volume_tension_piles_u_basin =
input_U_basin_open_pit.total_volume_tension_piles_u_basin.values
    return (U_basin_option,
    volume_of_U_basin_floor,volume_of_U_basin_walls,volume_of_concrete_U_basin, wall_formwork

```

Figure F-28: Code Input Dimensions

```

def def input_all_excel(main_folder, folder):
    input_general = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'general')
    input_sheetpile = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'sheetpile')
    input_diaphragm_wall = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'diaphragm_wall')
    input_U_basin_open_pit = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'U_basin_open_pit')
    input_U_basin_steep_pit = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'U_basin_steep_pit')
    input_values_per_depth = pd.read_excel('C:\Users\908729\Documents\Output-Tool\New_runs'+
    main_folder + '\\') + folder + '\output_dimensions.xlsx', 'values_per_depth')
    return
(input_general, input_sheetpile, input_diaphragm_wall, input_U_basin_open_pit, input_U_basin_steep_pit,
input_values_per_depth)

##### GENERAL #####
def def input_general(input_general):
    dict_general = {}
    for col in input_general.columns.tolist():
        dict_general[col] = input_general.at[0, col]
    dict_general['length']
    length = input_general.Length[0]
    width = input_general.Width[0]
    lowest_lock_level = input_general.Lowest Lock Level[0]
    bottom_structure = input_general.bottom structure[0]
    range_length_pile = input_general.range_length_pile[0]
    range_number_of_anchor = input_general.range_number_of_anchor[0]
    range_sheetpile = input_general.range_sheetpile[0]
    range_length_diaphragm_wall = input_general.range_length_diaphragm_wall[0]
    range_number_of_anchor_diaphragm_wall =
    input_general.range_number_of_anchor_diaphragm_wall[0]
    range_thickness_diaphragm_wall = input_general.range_thickness_diaphragm_wall[0]
    number_of_calculations = input_general.number_of_calculations[0]
    version_check = input_general.version_check[0]
    return(length,width,lowest_lock_level,bottom_structure,range_length_pile,range_number_of_anchor,
range_sheetpile,range_length_diaphragm_wall,range_number_of_anchor_diaphragm_wall,range_thickness
diaphragm_wall,number_of_calculations,version_check)

##### SHEETPILE #####
def def input_sheetpile(input_sheetpile):
    sheetpile_length = input_sheetpile.sheetpile_length.values
    sectional_area_sheetpile = input_sheetpile.sectional_area_sheetpile.values
    volume_placing_backfill = input_sheetpile.volume_placing_backfill.values
    volume_import_backfill_soil = input_sheetpile.volume_import_backfill_soil.values
    volume_export_sand = input_sheetpile.volume_export_sand.values
    volume_export_peat = input_sheetpile.volume_export_peat.values
    volume_export_clay = input_sheetpile.volume_export_clay.values
    volume_export_total = input_sheetpile.volume_export_total.values
    excavation_volume = input_sheetpile.excavation_volume.values
    volume_re_used_sand = input_sheetpile.volume_re_used_sand.values
    volume_underwater_concrete_floor_total =
    input_sheetpile.volume_underwater_concrete_floor_total.values
    total_length_tension_piles = input_sheetpile.total_length_tension_piles.values
    total_volume_tension_piles = input_sheetpile.total_volume_tension_piles.values
    number_of_anchors = input_sheetpile.number_of_anchors.values
    sheetpile_couting_area = input_sheetpile.sheetpile_couting_area.values
    volume_concrete_in_keep = input_sheetpile.df.volume_concrete_in_keep.values
    volume_reinforcement_in_keep = input_sheetpile.df.volume_reinforcement_in_keep.values
    amount_formwork_in_keep = input_sheetpile.df.amount_formwork_in_keep.values

    floor_formwork_volume_reinforcement_walls, volume_reinforcement_floor
    volume_placing_backfill_U_basin_open_pit, volume_import_backfill_U_basin_open_pit,
    volume_export_sand_U_basin_open_pit, volume_export_peat_U_basin_open_pit,
    volume_export_clay_U_basin_open_pit, volume_export_total_U_basin_open_pit,
    volume_excavation_U_basin_open_pit, volume_re_used_sand_U_basin_open_pit,
    total_length_tension_piles_u_basin, total_volume_tension_piles_u_basin)

##### U-BASIN STEEP PIT #####
def def input_U_basin_steep_pit(input_U_basin_steep_pit):
    sheetpile_length_u = input_U_basin_steep_pit.sheetpile_length_u.values
    sectional_area_sheetpile_u = input_U_basin_steep_pit.sectional_area_sheetpile_u.values
    number_of_anchors_u = input_U_basin_steep_pit.number_of_anchors_u.values
    volume_placing_backfill_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_placing_backfill_U_basin_steep_pit.values
    volume_import_backfill_soil_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_import_backfill_soil_U_basin_steep_pit.values
    volume_export_sand_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_export_sand_U_basin_steep_pit.values
    volume_export_peat_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_export_peat_U_basin_steep_pit.values
    volume_export_clay_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_export_clay_U_basin_steep_pit.values
    volume_export_total_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_export_total_U_basin_steep_pit.values
    excavation_volume_U_basin_steep_pit =
    input_U_basin_steep_pit.excavation_volume_U_basin_steep_pit.values
    volume_re_used_sand_U_basin_steep_pit =
    input_U_basin_steep_pit.volume_re_used_sand_U_basin_steep_pit.values
    volume_underwater_concrete_floor_total_u =
    input_U_basin_steep_pit.volume_underwater_concrete_floor_total_u.values
    total_length_tension_piles_U_basin_steep_pit =
    input_U_basin_steep_pit.total_length_tension_piles_U_basin_steep_pit.values
    total_volume_tension_piles_U_basin_steep_pit =
    input_U_basin_steep_pit.total_volume_tension_piles_U_basin_steep_pit.values
    return(sheetpile_length_u, sectional_area_sheetpile_u, number_of_anchors_u, volume_placing_backfill_
U_basin_steep_pit, volume_import_backfill_soil_U_basin_steep_pit, volume_export_sand_U_basin_steep
_pit, volume_export_peat_U_basin_steep_pit, volume_export_clay_U_basin_steep_pit, volume_export_total_U_b
asin_steep_pit, volume_re_used_sand_U_basin_steep_pit, volume_underwater_concrete_floor_total_u, tot
al_length_tension_piles_U_basin_steep_pit, total_volume_tension_piles_U_basin_steep_pit)

return(sheetpile_length, sectional_area_sheetpile, volume_placing_backfill, volume_import_backfill_u
oil, volume_export_sand, volume_export_peat, volume_export_clay, volume_export_total,
excavation_volume, volume_re_used_sand, volume_underwater_concrete_floor_total,
sheetpile_couting_area, volume_concrete_in_keep, volume_reinforcement_in_keep, number_of_anchors,
sheetpile_couting_area, volume_concrete_in_keep, volume_reinforcement_in_keep, amount_formwork_in_keep,
amount_formwork_in_keep)

return(thickness_diaphragm_wall, length_diaphragm_wall, width_diaphragm_wall, number_of_bars, reinfor
cement_diameter, reinforcement_per_cubic_meter_concrete, volume_underwater_concrete_floor_total_d, n
umber_of_anchors_d, volume_concrete_in_keep_d, volume_reinforcement_in_keep_d, amount_formwork_in_ka
sp_d)

##### U-BASIN OPEN PIT #####
def def input_U_basin_open_pit(input_U_basin_open_pit):
    U_basin_option = input_U_basin_open_pit.U_basin_option.values
    volume_of_U_basin_floor = input_U_basin_open_pit.volume_of_U_basin_floor.values
    volume_of_U_basin_walls = input_U_basin_open_pit.volume_of_U_basin_walls.values
    volume_of_concrete_U_basin = input_U_basin_open_pit.volume_of_concrete_U_basin.values
    wall_formwork = input_U_basin_open_pit.wall_formwork.values
    floor_formwork = input_U_basin_open_pit.floor_formwork.values
    volume_reinforcement_walls = input_U_basin_open_pit.volume_reinforcement_walls.values
    volume_reinforcement_floor = input_U_basin_open_pit.volume_reinforcement_floor.values
    volume_placing_backfill_U_basin_open_pit =
    input_U_basin_open_pit.volume_placing_backfill_U_basin_open_pit.values
    volume_import_backfill_U_basin_open_pit =
    input_U_basin_open_pit.volume_import_backfill_U_basin_open_pit.values
    volume_export_sand_U_basin_open_pit =
    input_U_basin_open_pit.volume_export_sand_U_basin_open_pit.values
    volume_export_peat_U_basin_open_pit =
    input_U_basin_open_pit.volume_export_peat_U_basin_open_pit.values
    volume_export_clay_U_basin_open_pit =
    input_U_basin_open_pit.volume_export_clay_U_basin_open_pit.values
    volume_export_total_U_basin_open_pit =
    input_U_basin_open_pit.volume_export_total_U_basin_open_pit.values
    volume_excavation_U_basin_open_pit =
    input_U_basin_open_pit.volume_excavation_U_basin_open_pit.values
    volume_re_used_sand_U_basin_open_pit =
    input_U_basin_open_pit.volume_re_used_sand_U_basin_open_pit.values
    total_length_tension_piles_u_basin =
    input_U_basin_open_pit.total_length_tension_piles_u_basin.values
    total_volume_tension_piles_u_basin =
    input_U_basin_open_pit.total_volume_tension_piles_u_basin.values
    return (U_basin_option,
    volume_of_U_basin_floor, volume_of_U_basin_walls, volume_of_concrete_U_basin, wall_formwork

```

Figure F-29: Code Input Dimensions

```

def sheetpile_costs(input_sheetpile_length, lowest_lock_level, bottom_structure,
                    sheetpile_length, sectional_area_sheetpile, volume_placing_backfill,
                    volume_import_backfill_soil,
                    volume_export_sand, volume_export_peat, volume_export_clay,
                    volume_import_sand, volume_underwater_concrete_floor_total,
                    total_length_tension_piles,
                    total_volume_tension_piles, number_of_anchors, sheetpile_couting_area,
                    volume_concrete_in_keep,
                    volume_reinforcement_in_keep, amount_formwork_in_keep):
    mass_sheetpile_per_meter = sheetpile_length * sectional_area_sheetpile / 10000 * 7850 # mass steel
    # 7850 kg/m3 kg/m3
    mass_sheetpile_total = mass_sheetpile_per_meter * Length * 2
    surface_sheetpile = sheetpile_length * Length * 2

    costs_sheetpile_material_total = mass_sheetpile_total * costs_sheetpile_material / 1000
    costs_sheetpile_placing = surface_sheetpile * placing_sheetpile_from_land

    MKI_sheet_pile = mass_sheetpile_total * MKI_value_future['Steel sheet pile'] / 1000
    CO2_tax_sheet_pile = mass_sheetpile_total * MKI_CO2_future['Steel sheet pile'] / 1000

    costs_placing_backfill = volume_placing_backfill * costs_replacement_soil
    costs_import_backfill_soil = volume_import_backfill_soil * costs_deliver_sand
    costs_export_sand = volume_export_sand * costs_processing_sand
    costs_export_peat = volume_export_peat * costs_processing_peat
    costs_export_clay = volume_export_clay * costs_processing_clay
    costs_export_total = costs_export_sand + costs_export_peat + costs_export_clay
    excavation_costs = excavation_volume * costs_excavation
    soil_handling_costs = excavation_costs + costs_placing_backfill + costs_import_backfill_soil + costs_export_total

    MKI_export_soil = volume_export_total * MKI_value_future['Soil (ship)']
    MKI_import_sand = volume_import_backfill_soil * MKI_value_future['Land sand (ship)']
    MKI_re_used_sand = volume_re_used_sand * MKI_value_future['Work with work sand']
    MKI_soil_handling = MKI_export_soil + MKI_import_sand + MKI_re_used_sand

    CO2_tax_export_soil = volume_export_total * MKI_CO2_future['Soil (ship)']
    CO2_tax_import_sand = volume_import_backfill_soil * MKI_CO2_future['Land sand (ship)']
    CO2_tax_re_used_sand = volume_re_used_sand * MKI_CO2_future['Work with work sand']
    CO2_tax_soil_handling = CO2_tax_export_soil + CO2_tax_import_sand + CO2_tax_re_used_sand

    costs_underwater_concrete_floor_total = volume_underwater_concrete_floor_total *
    costs_underwater_concrete_floor

    if type_of_UWC == 'C30/37':
        print('C30/37')
        MKI_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_value_future['Underwaterconcrete C30/37']
        CO2_tax_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_CO2_future['Underwaterconcrete C30/37']
    elif type_of_UWC == 'C30/37R':
        MKI_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_value_future['Underwaterconcrete C30/37R']
        CO2_tax_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_CO2_future['Underwaterconcrete C30/37R']
    else:
        print('<25')
        MKI_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_value_future['Underwaterconcrete C20/25'] / 2
        CO2_tax_underwaterconcretefloor = volume_underwater_concrete_floor_total *
        MKI_CO2_future['Underwaterconcrete C20/25'] / 2

    costs_tension_piles = costs_prefab_tension_pile_per_meter * total_length_tension_piles
    MKI_tension_piles = (total_volume_tension_piles) * MKI_value_future['Concrete foundation
    pile']
    CO2_tax_tension_piles = (total_volume_tension_piles) * MKI_CO2_future['Concrete foundation
    pile']

    anchor_cost_sheetpile = number_of_anchors * 3000.0
    MKI_anchors = number_of_anchors * (0.25 * np.pi * 0.25 ** 2) * 5 * MKI_value_future['Grout
    anchors']
    #grout body of 0,25m diameter and 5m length
    CO2_tax_anchors = number_of_anchors * (0.25 * np.pi * 0.25 ** 2) * 5 * MKI_CO2_future['Grout
    anchors']
    # grout body of 0,25m diameter and 5m length

    # concrete costs conservation
    if type_conservation == 'None':
        conservation_costs = np.array(list(map(float, Sheetpile_couting_area))) * 0
        test = np.array(list(map(float, Sheetpile_couting_area))) * 0
    elif type_conservation == 'Cathodic Protection':
        conservation_costs = np.array(list(map(float, Sheetpile_couting_area))) *
        sheetpile_length * extra_thickness_conservation / 1000 * 7850 * costs_sheetpile_material / 1000 *
        length * 2
    elif type_conservation == 'Cathodic Protection':
        conservation_costs = np.array(list(map(float, Sheetpile_couting_area)))
        * ((lowest_lock_level - bottom_structure) * 2 * Length * 2 * 7850 * costs_sheetpile_material / 1000 *
        length * 2

    # REEP
    concrete_costs_in_keep = volume_concrete_in_keep * costs_material_and_casting_concrete
    MKI_keep_concrete = volume_concrete_in_keep * MKI_value_future['Concrete C35/45 (CMIII)']
    CO2_tax_keep_concrete = volume_concrete_in_keep * MKI_CO2_future['Concrete C35/45 (CMIII)']

    costs_reinforcement_in_keep = volume_reinforcement_in_keep * 7850 *
    costs_material_and_placing_reinforcement
    MKI_keep_reinforcement = volume_reinforcement_in_keep * 7850 *
    MKI_value_future['Reinforcement steel'] / 1000
    CO2_tax_keep_reinforcement = volume_reinforcement_in_keep * 7850 *
    MKI_CO2_future['Reinforcement steel'] / 1000

    costs_formwork_in_keep = (amount_formwork_in_keep) * costs_material_and_placing_formwork
    MKI_keep_formwork = amount_formwork_in_keep * MKI_value_future['Traditional formwork']
    CO2_tax_keep_formwork = amount_formwork_in_keep * MKI_CO2_future['Traditional formwork']

    MKI_value = MKI_soil_handling + MKI_tension_piles + MKI_anchors + MKI_sheet_pile +
    MKI_keep_concrete + MKI_keep_reinforcement + MKI_keep_formwork + MKI_underwaterconcretefloor
    CO2_tax_value = CO2_tax_soil_handling + CO2_tax_tension_piles + CO2_tax_anchors +
    CO2_tax_sheet_pile + CO2_tax_keep_concrete + CO2_tax_keep_reinforcement + CO2_tax_keep_formwork +
    CO2_tax_underwaterconcretefloor

    total_costs_sheetpile_chamber = soil_handling_costs +
    costs_sheetpile_material_total + costs_sheetpile_placing + costs_underwater_concrete_floor_total +
    costs_tension_piles + costs_formwork_in_keep + costs_reinforcement_in_keep + concrete_costs_in_keep +
    concrete_costs_in_keep + anchor_cost_sheetpile + conservation_costs + CO2_tax_value

    total_costs_sheetpile_chamber =
    pd.DataFrame({'Total_costs_sheetpile_chamber': total_costs_sheetpile_chamber})

    outputfile_sheetpiles = pd.DataFrame({'Run Name': input_sheetpile.Run Name.values})
    outputfile_sheetpiles =
    outputfile_sheetpiles.assign(sheetpile_length=sheetpile_length, Stable_Unstable=input_sheetpile.St
    able_Unstable.values, strength_check_sheetpile=input_sheetpile.Strength_check_sheetpile.values, Sh
    eetpile_profile=input_sheetpile.Sheetpile_profile.values)

    outputfile_sheetpiles =
    outputfile_sheetpiles.assign(total_costs_sheetpile_chamber=total_costs_sheetpile_chamber, soil_ha
    ndling_costs=soil_handling_costs, costs_sheetpile_material_total=costs_sheetpile_material_total, c
    osts_sheetpile_placing=costs_sheetpile_placing,
    conservation_costs=conservation_costs, costs_underwater_concrete_floor_total=costs_underwater_co
    ncrete_floor_total, costs_tension_piles=costs_tension_piles, anchor_cost_sheetpile=anchor_cost_she
    etpile, costs_formwork_in_keep=costs_formwork_in_keep, costs_reinforcement_in_keep=costs_reinforce
    ment_in_keep, concrete_costs_in_keep=concrete_costs_in_keep)
    #outputfile_sheetpiles['total_costs_without_anchors'] =
    outputfile_sheetpiles['df_costs_sheetpile_material_total', 'soil_handling_costs', 'costs_underwa
    te_concrete_floor_total', 'costs_tension_piles'].sum(axis=1)
    #outputfile_sheetpiles =
    outputfile_sheetpiles.drop(['Run', 'sectional_area_sheetpile', 'anchor_force_1_stage_1', 'anchor_for
    ce_1_stage_2', 'anchor_force_1_stage_3', 'anchor_force_1_stage_4', 'anchor_force_2_stage_1', 'anchor
    force_2_stage_2', 'anchor_force_2_stage_3', 'anchor_force_2_stage_4', 'df_R_underwaterconcreteflo
    or_all'], axis=1)

    outputfile_sheetpiles =
    outputfile_sheetpiles.assign(MKI_value=MKI_value, CO2_tax_value=CO2_tax_value)
    outputfile_sheetpiles =
    outputfile_sheetpiles[['sheetpile_length', 'Sheetpile_profile', 'Stable_Unstable', 'Strength_check_s
    heetpile', 'total_costs_sheetpile_chamber', 'MKI_value', 'CO2_tax_value', 'costs_sheetpile_material
    + steel', 'costs_sheetpile_placing', 'conservation_costs', 'costs_underwater_concrete_floor_total', 'co
    sts_tension_piles', 'anchor_cost_sheetpile', 'soil_handling_costs', 'costs_formwork_in_keep', 'costs
    reinforcement_in_keep', 'concrete_costs_in_keep']]

    outputfile_sheetpiles.sort values(['Stable_Unstable', 'Strength_check_sheetpile', 'total_costs_she
    etpile_chamber'])
    order = outputfile_sheetpiles.index.values
    tel = np.linspace(0, len(order)-1, len(order))
    outputfile_MKI_sheetpile =
    pd.DataFrame({'tel':tel, 'MKI_value':MKI_value, 'MKI_sheet_pile':MKI_sheet_pile, 'MKI_anchors':MKI_a
    nchors, 'MKI_tension_piles':MKI_tension_piles, 'MKI_import_sand':MKI_import_sand, 'MKI_export_soil':
    MKI_export_soil, 'MKI_re_used_sand':MKI_re_used_sand, 'MKI_soil_handling':MKI_soil_handling, 'MKI_re
    inforcement':MKI_keep_reinforcement, 'MKI_keep_formwork':MKI_keep_formwork, 'MKI_keep_concrete':
    MKI_keep_concrete, 'MKI_underwaterconcretefloor':MKI_underwaterconcretefloor})
    outputfile_CO2_tax_sheetpile =
    pd.DataFrame({'tel':tel, 'CO2_tax_value':CO2_tax_value, 'CO2_tax_sheet_pile':CO2_tax_sheet_pile, 'CO
    2_tax_anchors':CO2_tax_anchors, 'CO2_tax_tension_piles':CO2_tax_tension_piles, 'CO2_tax_import_sand
    ':CO2_tax_import_sand, 'CO2_tax_export_soil':CO2_tax_export_soil, 'CO2_tax_re_used_sand':CO2_tax_re
    used_sand, 'CO2_tax_soil_handling':CO2_tax_soil_handling, 'CO2_tax_keep_concrete':CO2_tax_keep_con
    crete, 'CO2_tax_keep_reinforcement':CO2_tax_keep_reinforcement, 'CO2_tax_keep_formwork':CO2_tax_ka
    eep_formwork, 'CO2_tax_underwaterconcretefloor':CO2_tax_underwaterconcretefloor})
    outputfile_MKI_sheetpile['tel'] = pd.Categorical(outputfile_MKI_sheetpile.tel,
    categories=order, ordered=True)
    outputfile_MKI_sheetpile.sort values('tel', inplace=True)
    outputfile_CO2_tax_sheetpile['tel'] = pd.Categorical(outputfile_CO2_tax_sheetpile.tel,
    categories=order, ordered=True)
    outputfile_CO2_tax_sheetpile.sort values('tel', inplace=True)
    with pd.ExcelWriter('output_sheetpile.xlsx') as writer: # docstest: <SKIP
        outputfile_sheetpiles.to_excel(writer, sheet_name='Output')
        outputfile_MKI_sheetpile.to_excel(writer, sheet_name='MKI')
        outputfile_CO2_tax_sheetpile.to_excel(writer, sheet_name='CO2 tax')

    #outputfile_sheetpiles.to_excel('Output_sheetpiles.xlsx')

    return
    (outputfile_MKI_sheetpile, outputfile_CO2_tax_sheetpile, outputfile_sheetpiles, soil_handling_costs,
    MKI_soil_handling,
    MKI_tension_piles, MKI_export_soil, MKI_import_sand, MKI_re_used_sand,
    CO2_tax_tension_piles, CO2_tax_export_soil,
    CO2_tax_import_sand, CO2_tax_re_used_sand, CO2_tax_soil_handling)

```

Figure F-30: Code Output Sheet Pile Wall Structure

```

def
diaphragm_wall_costs(length,range,length,diaphragm_wall,range,number_of_anchor,diaphragm_wall,range,
thickness,diaphragm_wall,length,diaphragm_wall,width,diaphragm_wall,
number_of_bars,
reinforcement_diameter,reinforcement_per_cubic_meter_concrete,
volume_underwater_concrete_floor_total_d,number_of_anchors_d,
volume_concrete_in_keeping_d,
volume_reinforcement_in_keeping_d,amount_formwork_in_keeping_d,
total_length_tension_piles,soil_handling_costs,
MKI_tension_piles,MKI_export_soil,MKI_import_sand,MKI_re_used_sand,
MKI_soil_handling,
CO2_tax_reinforcement_piles,CO2_tax_export_soil,
CO2_tax_import_sand,CO2_tax_re_used_sand,CO2_tax_soil_handling,input_diaphragm_wall):
reinforcement_cost_diaphragm_wall = np.zeros(number_of_calculations)
concrete_cost_diaphragm_wall = np.zeros(number_of_calculations)
bentonite_cost_diaphragm_wall = np.zeros(number_of_calculations)
production_and_extra_cost_diaphragm_wall = np.zeros(number_of_calculations)
anchor_cost_diaphragm_wall = np.zeros(number_of_calculations)
soil_handling_costs_diaphragm_wall = np.zeros(number_of_calculations)
costs_tension_piles_diaphragm_wall = np.zeros(number_of_calculations)
total_costs_diaphragm_wall = np.zeros(number_of_calculations)
MKI_reinforcement_diaphragm_wall = np.zeros(number_of_calculations)
MKI_concrete_diaphragm_wall = np.zeros(number_of_calculations)
MKI_bentonite_diaphragm_wall = np.zeros(number_of_calculations)
MKI_shaft_excavation_diaphragm_wall = np.zeros(number_of_calculations)
MKI_anchors_diaphragm_wall = np.zeros(number_of_calculations)
MKI_tension_piles_diaphragm_wall = np.zeros(number_of_calculations)
MKI_export_soil_diaphragm_wall = np.zeros(number_of_calculations)
MKI_import_sand_diaphragm_wall = np.zeros(number_of_calculations)
MKI_re_used_sand_diaphragm_wall = np.zeros(number_of_calculations)
MKI_soil_handling_diaphragm_wall = np.zeros(number_of_calculations)
MKI_underwaterconcretefloor_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_reinforcement_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_bentonite_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_shaft_excavation_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_anchors_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_tension_piles_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_export_soil_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_import_sand_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_re_used_sand_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_soil_handling_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_underwaterconcretefloor_diaphragm_wall = np.zeros(number_of_calculations)
CO2_tax_value_diaphragm_wall = np.zeros(number_of_calculations)

##### COSTS
#####
wb_diaphragm = wb.Book('Rekeningen diaphragm_20181120.xlsx') # connect to an existing file in
the current working directory
diaphragm_costs = wb.diaphragm.sheets['Sheet1']
diaphragm_costs.range('F13').value = length * 2
FF = -1
for dd in range(range.length.diaphragm_wall):
    for ddd in range(range.number_of_anchor.diaphragm_wall):
        for dddd in range(range.thickness.diaphragm_wall):
            FF = 1
            diaphragm_costs.range('F10').value = thickness.diaphragm_wall/1000
            diaphragm_costs.range('F11').value = length.diaphragm_wall(FF)
            diaphragm_costs.range('F14').value = reinforcement_per_cubic_meter_concrete [FF]
            concrete_cost_diaphragm_wall(FF) = diaphragm_costs.range('H8').value
            bentonite_cost_diaphragm_wall(FF) = diaphragm_costs.range('H9').value
            production_and_extra_cost_diaphragm_wall(FF) = diaphragm_costs.range('I60').value
            MKI_reinforcement_diaphragm_wall(FF) = diaphragm_costs.range('M2').value
            MKI_concrete_cost_diaphragm_wall(FF) = diaphragm_costs.range('M3').value
            MKI_bentonite_cost_diaphragm_wall(FF) = diaphragm_costs.range('M4').value
            MKI_shaft_excavation_diaphragm_wall(FF) = diaphragm_costs.range('M5').value
            MKI_anchors_diaphragm_wall(FF) = diaphragm_costs.range('M6').value
            MKI_tension_piles_diaphragm_wall(FF) = diaphragm_costs.range('M7').value
            MKI_export_soil_diaphragm_wall(FF) = diaphragm_costs.range('M8').value
            MKI_import_sand_diaphragm_wall(FF) = diaphragm_costs.range('M9').value
            MKI_re_used_sand_diaphragm_wall(FF) = diaphragm_costs.range('M10').value
            MKI_soil_handling_diaphragm_wall(FF) = diaphragm_costs.range('M11').value
            MKI_underwaterconcretefloor_diaphragm_wall(FF) = diaphragm_costs.range('M12').value
            CO2_tax_reinforcement_diaphragm_wall(FF) = diaphragm_costs.range('N2').value
            CO2_tax_bentonite_diaphragm_wall(FF) = diaphragm_costs.range('N3').value
            CO2_tax_shaft_excavation_diaphragm_wall(FF) = diaphragm_costs.range('N4').value
            CO2_tax_anchors_diaphragm_wall(FF) = diaphragm_costs.range('N5').value
            CO2_tax_tension_piles_diaphragm_wall(FF) = diaphragm_costs.range('N6').value
            CO2_tax_export_soil_diaphragm_wall(FF) = diaphragm_costs.range('N7').value
            CO2_tax_import_sand_diaphragm_wall(FF) = diaphragm_costs.range('N8').value
            CO2_tax_re_used_sand_diaphragm_wall(FF) = diaphragm_costs.range('N9').value
            CO2_tax_soil_handling_diaphragm_wall(FF) = diaphragm_costs.range('N10').value
            CO2_tax_underwaterconcretefloor_diaphragm_wall(FF) = diaphragm_costs.range('N11').value
            CO2_tax_value_diaphragm_wall(FF) = diaphragm_costs.range('N12').value

costs_reinforcement_in_keeping_d = volume_reinforcement_in_keeping_d * 7850 *
costs_material_and_placing_reinforcement
MKI_keep_reinforcement_d = volume_reinforcement_in_keeping_d * 7850 *
MKI_value_future_reinforcement_steel_d / 1000
CO2_tax_keep_reinforcement_d = volume_reinforcement_in_keeping_d * 7850 *
MKI_CO2_future_reinforcement_steel_d / 1000

costs_formwork_in_keeping_d = (amount_formwork_in_keeping_d) * costs_material_and_placing_formwork
MKI_keep_formwork_d = amount_formwork_in_keeping_d * MKI_value_future_traditional_formwork
CO2_tax_keep_formwork_d = amount_formwork_in_keeping_d * MKI_CO2_future_traditional_formwork

MKI_value_diaphragm_wall = MKI_soil_handling_diaphragm_wall +
MKI_tension_piles_diaphragm_wall + MKI_anchors_diaphragm_wall +
MKI_shaft_excavation_diaphragm_wall + MKI_concrete_diaphragm_wall +
MKI_reinforcement_diaphragm_wall + MKI_keep_concrete_d + MKI_keep_reinforcement_d +
MKI_keep_formwork_d + MKI_underwaterconcretefloor_diaphragm_wall +
CO2_tax_value_diaphragm_wall + CO2_tax_soil_handling_diaphragm_wall +
CO2_tax_tension_piles_diaphragm_wall + CO2_tax_anchors_diaphragm_wall +
CO2_tax_shaft_excavation_diaphragm_wall + CO2_tax_bentonite_diaphragm_wall +
CO2_tax_export_soil_diaphragm_wall + CO2_tax_import_sand_diaphragm_wall + CO2_tax_re_used_sand_d +
CO2_tax_soil_handling_diaphragm_wall + CO2_tax_keep_reinforcement_d + CO2_tax_keep_formwork_d +
CO2_tax_underwaterconcretefloor_diaphragm_wall

total_costs_diaphragm_wall_chamber = anchor_cost_diaphragm_wall +
soil_handling_costs_diaphragm_wall + total_costs_diaphragm_wall + total_costs_d +
costs_tension_piles_diaphragm_wall + costs_formwork_in_keeping_d + costs_reinforcement_in_keeping_d +
concrete_costs_in_keeping_d + CO2_tax_value_diaphragm_wall

outputfile_diaphragm_wall =
pd.DataFrame({'Run Name': d,'input_diaphragm_wall.Run Name d.values'})
outputfile_diaphragm_wall =
outputfile_diaphragm_wall.assign(diaphragm_wall_length=length.diaphragm_wall,diaphragm_wall_thickness=thickness,diaphragm_wall_width=width,diaphragm_wall_number_of_bars=number_of_bars,
diaphragm_wall_reinforcement_diameter=reinforcement_diameter,diaphragm_wall_reinforcement_per_cubic_meter_concrete=reinforcement_per_cubic_meter_concrete,diaphragm_wall_volume_underwater_concrete_floor_total_d=volume_underwater_concrete_floor_total_d,diaphragm_wall_number_of_anchors_d=number_of_anchors_d,diaphragm_wall_volume_concrete_in_keeping_d=volume_concrete_in_keeping_d,diaphragm_wall_volume_reinforcement_in_keeping_d=volume_reinforcement_in_keeping_d,diaphragm_wall_amount_formwork_in_keeping_d=amount_formwork_in_keeping_d,diaphragm_wall_total_length_tension_piles=length_tension_piles,diaphragm_wall_soil_handling_costs=soil_handling_costs,diaphragm_wall_MKI_tension_piles=MKI_tension_piles,diaphragm_wall_MKI_export_soil=MKI_export_soil,diaphragm_wall_MKI_import_sand=MKI_import_sand,diaphragm_wall_MKI_re_used_sand=MKI_re_used_sand,diaphragm_wall_MKI_soil_handling=MKI_soil_handling,diaphragm_wall_MKI_underwaterconcretefloor=MKI_underwaterconcretefloor,diaphragm_wall_CO2_tax_reinforcement_piles=CO2_tax_reinforcement_piles,diaphragm_wall_CO2_tax_export_soil=CO2_tax_export_soil,diaphragm_wall_CO2_tax_import_sand=CO2_tax_import_sand,diaphragm_wall_CO2_tax_re_used_sand=CO2_tax_re_used_sand,diaphragm_wall_CO2_tax_soil_handling=CO2_tax_soil_handling,diaphragm_wall_CO2_tax_underwaterconcretefloor=CO2_tax_underwaterconcretefloor,diaphragm_wall_CO2_tax_value=CO2_tax_value)
outputfile_diaphragm_wall =
pd.DataFrame({'Run Name': d,'input_diaphragm_wall.Run Name d.values'})
outputfile_diaphragm_wall =
outputfile_diaphragm_wall.assign(diaphragm_wall_length=length.diaphragm_wall,diaphragm_wall_thickness=thickness,diaphragm_wall_width=width,diaphragm_wall_number_of_bars=number_of_bars,
diaphragm_wall_reinforcement_diameter=reinforcement_diameter,diaphragm_wall_reinforcement_per_cubic_meter_concrete=reinforcement_per_cubic_meter_concrete,diaphragm_wall_volume_underwater_concrete_floor_total_d=volume_underwater_concrete_floor_total_d,diaphragm_wall_number_of_anchors_d=number_of_anchors_d,diaphragm_wall_volume_concrete_in_keeping_d=volume_concrete_in_keeping_d,diaphragm_wall_volume_reinforcement_in_keeping_d=volume_reinforcement_in_keeping_d,diaphragm_wall_amount_formwork_in_keeping_d=amount_formwork_in_keeping_d,diaphragm_wall_total_length_tension_piles=length_tension_piles,diaphragm_wall_soil_handling_costs=soil_handling_costs,diaphragm_wall_MKI_tension_piles=MKI_tension_piles,diaphragm_wall_MKI_export_soil=MKI_export_soil,diaphragm_wall_MKI_import_sand=MKI_import_sand,diaphragm_wall_MKI_re_used_sand=MKI_re_used_sand,diaphragm_wall_MKI_soil_handling=MKI_soil_handling,diaphragm_wall_MKI_underwaterconcretefloor=MKI_underwaterconcretefloor,diaphragm_wall_CO2_tax_reinforcement_piles=CO2_tax_reinforcement_piles,diaphragm_wall_CO2_tax_export_soil=CO2_tax_export_soil,diaphragm_wall_CO2_tax_import_sand=CO2_tax_import_sand,diaphragm_wall_CO2_tax_re_used_sand=CO2_tax_re_used_sand,diaphragm_wall_CO2_tax_soil_handling=CO2_tax_soil_handling,diaphragm_wall_CO2_tax_underwaterconcretefloor=CO2_tax_underwaterconcretefloor,diaphragm_wall_CO2_tax_value=CO2_tax_value)
outputfile_MKI_diaphragm_wall = pd.DataFrame()
return(outputfile_diaphragm_wall,outputfile_MKI_diaphragm_wall,outputfile_CO2_tax_diaphragm_wall)

```

Figure F-31: Code Output Diaphragm Wall Structure

```

MKI_U_Basin_open_pit_costs(U_Basin_option,
volume_of_U_basin_floor,volume_of_U_basin_walls,volume_of_concrete_U_basin_wall_formwork
,floor_formwork ,volume_reinforcement_walls ,volume_reinforcement_floor
,volume_placing_backfill_U_basin_open_pit, volume_import_backfill_U_basin_open_pit,
volume_export_sand_U_basin_open_pit ,volume_export_peas_U_basin_open_pit
,volume_export_clay_U_basin_open_pit ,volume_export_total_U_basin_open_pit,
volume_excavation_U_basin_open_pit ,volume_re_used_sand_U_basin_open_pit,
total_length_tension_piles_U_basin,total_volume_tension_piles_U_basin,width,length):
    #Concrete
    if type of concrete == 'C30/35':
        MKI_value_concrete = MKI_value_future[Concrete C30/35 (CEMIII)]
        MKI_CO2_concrete = MKI_CO2_future[Concrete C30/35 (CEMIII)]
    elif type of concrete == 'C30/37':
        MKI_value_concrete = MKI_value_future[Concrete C30/37 (CEMIII)]
        MKI_CO2_concrete = MKI_CO2_future[Concrete C30/37 (CEMIII)]
    elif type of concrete == 'C35/45':
        MKI_value_concrete = MKI_value_future[Concrete C35/45 (CEMIII)]
        MKI_CO2_concrete = MKI_CO2_future[Concrete C35/45 (CEMIII)]
    elif type of concrete == 'C35/47':
        MKI_value_concrete = MKI_value_future[Concrete C35/47 (CEMII-CEMIII)]
        MKI_CO2_concrete = MKI_CO2_future[Concrete C35/47 (CEMII-CEMIII)]
    elif type of concrete == 'C35/50':
        MKI_value_concrete = (MKI_value_future[Concrete C35/50 (CEMII-CEMIII)] +
        MKI_value_future[Concrete C35/45 (CEMIII)]) / 2
        MKI_CO2_concrete = (MKI_CO2_future[Concrete C35/50 (CEMII-CEMIII)] +
        MKI_CO2_future[Concrete C35/45 (CEMIII)]) / 2

    Costs of Concrete U_basin = (volume_of_U_basin_floor + volume_of_U_basin_walls) *
    costs_material_and_casting_concrete
    MKI_concrete_U_basin = (volume_of_U_basin_floor+ volume_of_U_basin_walls) *
    MKI_value_concrete
    CO2_tax_concrete_U_basin = (volume_of_U_basin_floor+ volume_of_U_basin_walls) *
    MKI_CO2_concrete

    mki_concrete_u_basin_floor = volume_of_U_basin_floor * MKI_value_concrete
    mki_concrete_u_basin_walls = volume_of_U_basin_walls * MKI_value_concrete
    cost_concrete_floor = volume_of_U_basin_floor*costs_material_and_casting_concrete
    cost_concrete_walls = volume_of_U_basin_walls * costs_material_and_casting_concrete

    # Formwork
    Costs of Formwork U_basin = wall_formwork * costs_material_and_placing_heavy_formwork +
    (floor_formwork * costs_material_and_placing_formwork + Length*Width_aisle_banden_bevonden
    subrikt
    MKI_formwork_U_basin = (wall_formwork + floor_formwork) * MKI_value_future[Traditional
    formwork]
    CO2_tax_formwork_U_basin= (wall_formwork + floor_formwork) * MKI_CO2_future[Traditional
    formwork]
    mki_formwork_u_basin_floor = floor_formwork * MKI_value_future[Traditional formwork]
    mki_formwork_u_basin_walls = wall_formwork * MKI_value_future[Traditional formwork]
    costs_formwork_floor = floor_formwork *costs_material_and_placing_formwork
    costs_formwork_wall = wall_formwork* costs_material_and_placing_heavy_formwork

    # Reinforcement
    Costs of Reinforcement U_basin = (volume_reinforcement_walls+volume_reinforcement_floor) *
    7850 * costs_material_and_placing_reinforcement
    MKI_reinforcement_U_basin = (volume_reinforcement_walls+volume_reinforcement_floor) * 7850 *
    MKI_value_future[Reinforcement steel] / 1000
    CO2_tax_reinforcement_U_basin = (volume_reinforcement_walls+volume_reinforcement_floor) *
    7850 * MKI_CO2_future[Reinforcement steel] / 1000

    mki_reinforcement_floor = volume_reinforcement_floor * 7850 * MKI_value_future[Reinforcement
    steel] / 1000
    mki_reinforcement_walls = volume_reinforcement_walls * 7850 * MKI_value_future[Reinforcement
    steel] / 1000
    costs_reinforcement_floor = volume_reinforcement_floor * 7850 *
    costs_material_and_placing_reinforcement
    costs_reinforcement_walls = volume_reinforcement_walls*7850 *
    costs_material_and_placing_reinforcement

    # Tension Piles
    Costs of Tension Piles U_basin = (costs_prefab_tension_pile_per_meter *
    total_length_tension_piles_U_basin)

    mki_walls, mki_reinforcement_floor,mki_reinforcement_floor))
    outputfile_CO2_tax_U_basin_open_pit = pd.DataFrame(
        {'CO2_tax_value_U_basin': CO2_tax_value_U_basin, 'CO2_tax_concrete_U_basin':
        CO2_tax_concrete_U_basin,
        'CO2_tax_reinforcement_U_basin': CO2_tax_reinforcement_U_basin,
        'CO2_tax_formwork_U_basin': CO2_tax_formwork_U_basin,
        'CO2_tax_tension_piles_U_basin': CO2_tax_tension_piles_U_basin,
        'CO2_tax_import_sand_U_basin_open_pit': CO2_tax_import_sand_U_basin_open_pit,
        'CO2_tax_export_sand_U_basin_open_pit': CO2_tax_export_sand_U_basin_open_pit,
        'CO2_tax_re_used_sand_U_basin_open_pit': CO2_tax_re_used_sand_U_basin_open_pit,
        'CO2_tax_soil_handling_U_basin_open_pit': CO2_tax_soil_handling_U_basin_open_pit})

    with pd.ExcelWriter('output_U_basin_open_pit.xlsx') as writer: # doctest: +SKIP
        outputfile_U_basin_open_pit.to_excel(writer, sheet_name='Output')
        outputfile_MKI_U_basin_open_pit.to_excel(writer, sheet_name='MKI_Output')

    return
(outputfile_U_basin_open_pit,outputfile_MKI_U_basin_open_pit,outputfile_CO2_tax_U_basin_open_pit,
Costs of Concrete U_basin,Costs of Reinforcement U_basin,Costs of Formwork U_basin,MKI_formwork_U
_basin,MKI_concrete_U_basin,MKI_reinforcement_U_basin,CO2_tax_formwork_U_basin,CO2_tax_concrete_U
_basin,CO2_tax_reinforcement_U_basin)

```

Figure F-32: Code U-Basin Structure Open Building Pit


```

def
    u_basin_steep_pit_costs(input_u_basin_steep_pit, length, range_length_pile, range_number_of_anchor,
    range_sheetpile_sheepile_length_u, sectional_area_sheetpile_u, number_of_anchors_u, volume_placing_b
    ackfill_u_basin_steep_pit, volume_import_backfill_soil_u_basin_steep_pit, volume_export_sand_u_basi
    n_steep_pit, volume_export_peat_u_basin_steep_pit, volume_export_clay_u_basin_steep_pit, volume_export_total_u_basin_steep_pit, excavation_volume_u_b
    asin_steep_pit, volume_reused_sand_u_basin_steep_pit, volume_underwater_concrete_floor_total_u_tot
    al_length_tension_piles_u_basin_steep_pit, total_volume_tension_piles_u_basin_steep_pit, costs_of_c
    oncrete_u_basin, costs_of_reinforcement_u_basin, costs_of_formwork_u_basin, MKI_formwork_u_basin, MKI_co
    ncrete_u_basin, MKI_reinforcement_u_basin, CO2_tax_formwork_u_basin, CO2_tax_concrete_u_basin, CO2
    _tax_reinforcement_u_basin):
    mass_sheetpile_per_meter_u = sheetpile_length_u * sectional_area_sheetpile_u / 10000 * 7850 # mass
    steel = "7850 kg/m3 #kg/m"
    mass_sheetpile_total_u = mass_sheetpile_per_meter_u * length * 2
    surface_sheetpile_u = sheetpile_length_u * length * 2
    costs_sheetpile_material_total_u = mass_sheetpile_total_u * costs_sheetpile_material / 1000
    costs_sheetpile_placing_u = surface_sheetpile_u * placing_sheetpile_from_land
    MKI_sheet_pile_u_basin_steep_pit = mass_sheetpile_total_u * MKI_value_future["Steel temporary
    sheet_pile"] / 1000
    CO2_tax_sheet_pile_u_basin_steep_pit = mass_sheetpile_total_u * MKI_CO2_future["Steel
    temporary sheet_pile"] / 1000
    costs_placing_backfill_u_basin_steep_pit = volume_placing_backfill_u_basin_steep_pit *
    costs_replacement_soil
    costs_import_backfill_u_basin_steep_pit = volume_import_backfill_soil_u_basin_steep_pit *
    costs_deliver_sand
    costs_export_sand_u_basin_steep_pit = volume_export_sand_u_basin_steep_pit *
    costs_processing_sand
    costs_export_peat_u_basin_steep_pit = volume_export_peat_u_basin_steep_pit *
    costs_processing_peat
    costs_export_clay_u_basin_steep_pit = volume_export_clay_u_basin_steep_pit *
    costs_processing_clay
    costs_export_total_u_basin_steep_pit = costs_export_sand_u_basin_steep_pit +
    costs_export_peat_u_basin_steep_pit + costs_export_clay_u_basin_steep_pit
    excavation_costs_u_basin_steep_pit = excavation_volume_u_basin_steep_pit * costs_excavation
    soil_handling_costs_u_basin_steep_pit = excavation_costs_u_basin_steep_pit +
    costs_placing_backfill_u_basin_steep_pit + costs_import_backfill_u_basin_steep_pit +
    costs_export_total_u_basin_steep_pit
    MKI_export_soil_u_basin_steep_pit = volume_export_total_u_basin_steep_pit *
    MKI_value_future["Soil (ship)"]
    MKI_import_sand_u_basin_steep_pit = volume_import_backfill_soil_u_basin_steep_pit *
    MKI_value_future["Land sand (ship)"]
    MKI_reused_sand_u_basin_steep_pit = volume_reused_sand_u_basin_steep_pit *
    MKI_value_future["Wash water sand"]
    MKI_soil_handling_u_basin_steep_pit = MKI_export_soil_u_basin_steep_pit +
    MKI_import_sand_u_basin_steep_pit + MKI_reused_sand_u_basin_steep_pit *
    (range_length_pile / range_number_of_anchor / range_sheetpile)
    CO2_tax_export_soil_u_basin_steep_pit = volume_export_total_u_basin_steep_pit *
    MKI_CO2_future["Soil (ship)"]
    CO2_tax_import_sand_u_basin_steep_pit = volume_import_backfill_soil_u_basin_steep_pit *
    MKI_CO2_future["Land sand (ship)"]
    CO2_tax_reused_sand_u_basin_steep_pit = volume_reused_sand_u_basin_steep_pit *
    MKI_CO2_future["Wash water sand"]
    CO2_tax_soil_handling_u_basin_steep_pit = CO2_tax_export_soil_u_basin_steep_pit +
    CO2_tax_import_sand_u_basin_steep_pit + CO2_tax_reused_sand_u_basin_steep_pit *
    (range_length_pile / range_number_of_anchor / range_sheetpile)
    costs_underwater_concrete_floor_total_u = volume_underwater_concrete_floor_total_u *
    costs_underwater_concrete_floor
    if type of UNC == 'CO2/25':
        MKI_underwaterconcretefloor_u_basin_steep_pit = volume_underwater_concrete_floor_total_u *
        MKI_value_future["Underwaterconcrete CO2/25"]
        CO2_tax_underwaterconcretefloor_u_basin_steep_pit =
        volume_underwater_concrete_floor_total_u * MKI_CO2_future["Underwaterconcrete CO2/25"]
    elif type of UNC == 'CO2/37':
        MKI_underwaterconcretefloor_u_basin_steep_pit = volume_underwater_concrete_floor_total_u *
        MKI_value_future["Underwaterconcrete CO2/37"]
    outputfile_u_basin_steep_pit =
    outputfile_u_basin_steep_pit.assign(total_costs_sheetpile_chamber_u, total_costs_sheetpile_chamber
    _u, MKI_value_u_basin_steep_pit, MKI_value_u_basin_steep_pit, CO2_tax_value_u_basin_steep_pit, CO2_ta
    x_value_u_basin_steep_pit, costs_soil_handling_costs_u_basin_steep_pit, costs_sheet
    pile_material_total_u, costs_sheetpile_placing_u_basin, costs_sheetpile_pl
    acing_u, conservation_costs_u, conservation_costs_u, costs_underwater_concrete_floor_total_u, costs_under
    water_concrete_floor_total_u, costs_tension_piles_u, costs_tension_piles_u_basin_steep_pit, anchor_co
    st_u, sheetpile_anchor_costs_u_basin_steep_pit)
    #outputfile_sheepiles["total_costs_without_anchors"] =
    outputfile_sheepiles["d_costs_sheetpile_material_total", "soil_handling_costs", "costs_underwat
    er_concrete_floor_total", "costs_tension_piles"].sum(axis=1)
    outputfile_u_basin_steep_pit =
    outputfile_u_basin_steep_pit.assign(costs_of_concrete_u_basin_steep_pit=costs_of_concrete_u_basin
    _steep_pit, costs_of_reinforcement_u_basin_steep_pit=costs_of_reinforcement_u_basin_steep_pit, cost
    s_of_formwork_u_basin_steep_pit=costs_of_formwork_u_basin_steep_pit)
    #outputfile_u_basin_steep_pit =
    outputfile_u_basin_steep_pit.drop(["Run_u", "sectional_area_sheetpile_u", "anchor_force_1_stage_u",
    "anchor_force_2_stage_u", "anchor_force_2_stage_2_u", "d_f_U_underwat
    erconcretefloor_all"], axis=1)
    outputfile_u_basin_steep_pit =
    outputfile_u_basin_steep_pit[["sheetpile_length_u", "sheetpile_profile_u", "Stable_Unstable_u", "Sta
    length_check_sheetpile_u", "total_costs_sheetpile_chamber_u", "MKI_value_u_basin_steep_pit", "CO2_tax
    _value_u_basin_steep_pit", "costs_of_concrete_u_basin_steep_pit", "costs_of_reinforcement_u_basin_
    _steep_pit", "costs_of_formwork_u_basin_steep_pit", "costs_sheetpile_material_total", "costs_sheetpile
    _placing_u", "conservation_costs", "costs_underwater_concrete_floor_total", "costs_tension_piles", "a
    nchor_cost_sheetpile", "soil_handling_costs"]]
    outputfile_u_basin_steep_pit.sort_values(["Stable_Unstable_u", "Strength_check_sheetpile_u", "total
    _costs_sheetpile_chamber_u"], inplace=True)
    MKI_outputfile_u_basin_steep_pit =
    pd.DataFrame(["MKI_value_u_basin_steep_pit", "MKI_value_u_basin_steep_pit", "MKI_export_soil_u_basi
    n_steep_pit", "MKI_export_soil_u_basin_steep_pit", "MKI_import_sand_u_basin_steep_pit", "MKI_import_sand
    _u_basin_steep_pit", "MKI_reused_sand_u_basin_steep_pit", "MKI_reused_sand_u_basin_steep_pit",
    "MKI_concrete_u_basin_steep_pit", "MKI_concrete_u_basin_steep_pit",
    "MKI_formwork_u_basin_steep_pit", "MKI_formwork_u_basin_steep_pit",
    "MKI_reinforcement_u_basin_steep_pit", "MKI_reinforcement_u_basin_steep_pit",
    "MKI_tension_piles_u_basin_steep_pit", "MKI_tension_piles_u_basin_steep_pit",
    "MKI_anchors_u_basin_steep_pit", "MKI_anchors_u_basin_steep_pit",
    "MKI_sheet_pile_u_basin_steep_pit", "MKI_sheet_pile_u_basin_steep_pit", "MKI_underwaterconcretefloor
    _u_basin_steep_pit", "MKI_underwaterconcretefloor_u_basin_steep_pit"])
    CO2_tax_outputfile_u_basin_steep_pit = pd.DataFrame(["CO2_tax_value_u_basin_steep_pit",
    CO2_tax_value_u_basin_steep_pit,
    CO2_tax_export_soil_u_basin_steep_pit,
    CO2_tax_export_soil_u_basin_steep_pit,
    CO2_tax_import_sand_u_basin_steep_pit,
    CO2_tax_import_sand_u_basin_steep_pit,
    CO2_tax_reused_sand_u_basin_steep_pit,
    CO2_tax_reused_sand_u_basin_steep_pit,
    CO2_tax_concrete_u_basin_steep_pit,
    CO2_tax_concrete_u_basin_steep_pit,
    CO2_tax_formwork_u_basin_steep_pit,
    CO2_tax_formwork_u_basin_steep_pit,
    CO2_tax_reinforcement_u_basin_steep_pit,
    CO2_tax_reinforcement_u_basin_steep_pit,
    CO2_tax_tension_piles_u_basin_steep_pit,
    CO2_tax_tension_piles_u_basin_steep_pit,
    CO2_tax_anchors_u_basin_steep_pit,
    CO2_tax_anchors_u_basin_steep_pit,
    CO2_tax_sheet_pile_u_basin_steep_pit,
    CO2_tax_sheet_pile_u_basin_steep_pit"])
    with pd.ExcelWriter("output_u_basin_steep_pit.xlsx") as writer: # docctest: +SKIP
        outputfile_u_basin_steep_pit.to_excel(writer, sheet_name="Output_u_basin_steep_pit")
    #outputfile_sheepiles.to_excel("output_sheepiles.xlsx")
    return(outputfile_u_basin_steep_pit, MKI_outputfile_u_basin_steep_pit, CO2_tax_outputfile_u_basin_
    _steep_pit)

```

Figure F-33: Code U-Basin Structure Retaining Wall Building Pit

```

main_folder =
'Output_Basin_Soil'#'Output_Basin_Angle_Anchor'#'Output_Basin_Depth'#'Output_Basin_Length'
#'(Concrete'#'MC_Floor'#'File_Length'#'File_Width'#'Width_Run'#'Depth'#'Drop_Run'#'Results
#(Bulks_Sea')
folder = os.listdir(r'C:\Users\908729\Documents\Output-Tool\New_runs'+ main_folder)
for i in range(len(folder)):

input_general,input_sheetpile,input_diaphragm_wall,input_U_basin_open_pit,input_U_basin_steep_pit
(input_values_per_depth = def input_all(excel_main_folder, folder[i])

length,width,lowest_lock_level,bottom_structure,range_length_pile,range_number_of_anchor,range_sh
sheetpile,range_length_diaphragm_wall,range_number_of_anchor_diaphragm_wall,range_thickness_diaphra
gm_wall,number_of_calculations,version_check = def input_general(input_general)

sheetpile_length,sectional_area_sheetpile,volume_placing_backfill,volume_import_backfill_soil,vol
ume_export_sand,volume_export_peat,volume_export_clay,volume_export_total,excavation_volume
(volume_re_used_sand,volume_underwater_concrete_floor_total,total_length_tension_piles
,total_volume_tension_piles,number_of_anchors,sheetpile_coating_area,volume_concrete_in_keap,
volume_reinforcement_in_keap,amount_formwork_in_keap = def input_sheetpile(input_sheetpile)
thickness_diaphragm_wall,length_diaphragm_wall,width_diaphragm_wall,number_of_bars,
reinforcement_diameter,reinforcement_per_cubic_meter_concrete,
volume_underwater_concrete_floor_total_d,number_of_anchors_d,volume_concrete_in_keap_d,
volume_reinforcement_in_keap_d,amount_formwork_in_keap_d =
def input_diaphragm_wall(input_diaphragm_wall)
U_basin_option,volume_of_U_basin_floor,volume_of_U_basin_walls,volume_of_concrete_U_basin,
wall_formwork,floor_formwork,volume_reinforcement_walls,volume_reinforcement_floor
(volume_placing_backfill_U_basin_open_pit,volume_import_backfill_U_basin_open_pit,
volume_export_sand_U_basin_open_pit,volume_export_peat_U_basin_open_pit
,volume_export_clay_U_basin_open_pit,volume_export_total_U_basin_open_pit,
volume_excavation_U_basin_open_pit,volume_re_used_sand_U_basin_open_pit
total_length_tension_piles_U_basin,total_volume_tension_piles_U_basin,def input_u_basin_open_pit(
input_U_basin_open_pit)

sheetpile_length_u,sectional_area_sheetpile_u,number_of_anchors_u,volume_placing_backfill_U_basin
_steep_pit,volume_import_backfill_soil_U_basin_steep_pit,volume_export_sand_U_basin_steep_pit,vol
ume_export_peat_U_basin_steep_pit
(volume_export_clay_U_basin_steep_pit,volume_export_total_U_basin_steep_pit,excavation_volume_U_b
asin_steep_pit,volume_re_used_sand_U_basin_steep_pit,volume_underwater_concrete_floor_total_u,tot
al_length_tension_piles_U_basin_steep_pit,total_volume_tension_piles_U_basin_steep_pit =
def input_u_basin_steep_pit(input_U_basin_steep_pit)

outputfile_MKI_sheetpile,outputfile_CO2_tax_sheetpile,outputfile_sheetpiles,soil_handling_costs,
MKI_tension_piles,MKI_export_soil,MKI_import_sand,MKI_re_used_sand,
MKI_soil_handling,CO2_tax_tension_piles,CO2_tax_export_soil,
CO2_tax_import_sand,CO2_tax_re_used_sand,CO2_tax_soil_handling =
sheetpile_length,sectional_area_sheetpile,volume_placing_backfill,
volume_import_backfill_soil,
volume_export_sand,volume_export_peat,volume_export_clay,
volume_export_total,excavation_volume,
volume_re_used_sand,volume_underwater_concrete_floor_total,
total_length_tension_piles,
total_volume_tension_piles,number_of_anchors,sheetpile_coating_area,
volume_concrete_in_keap,
volume_reinforcement_in_keap,amount_formwork_in_keap)
outputfile_diaphragm_wall,outputfile_MKI_diaphragm_wall,outputfile_CO2_tax_diaphragm_wall =
diaphragm_wall_costs(length,range_length_diaphragm_wall,range_number_of_anchor_diaphragm_wall,ran
ge_thickness_diaphragm_wall,number_of_calculations,
thickness_diaphragm_wall,length_diaphragm_wall,width_diaphragm_wall,
number_of_bars,
reinforcement_diameter,reinforcement_per_cubic_meter_concrete,
volume_underwater_concrete_floor_total_d,number_of_anchors_d,
volume_concrete_in_keap_d,
volume_reinforcement_in_keap_d,amount_formwork_in_keap_d,

outputfile_U_basin_open_pit,outputfile_MKI_U_basin_open_pit,outputfile_CO2_tax_U_basin_open_pit,
costs_of_Concrete_U_basin,Costs_of_Reinforcement_U_basin,Costs_of_Formwork_U_basin,MKI_formwork_U
_basin,MKI_concrete_U_basin,MKI_reinforcement_U_basin,CO2_tax_formwork_U_basin,CO2_tax_concrete_U
_basin,CO2_tax_reinforcement_U_basin = U_basin_open_pit_costs(U_basin_option,
volume_of_U_basin_floor,volume_of_U_basin_walls,volume_of_concrete_U_basin,wall_formwork
floor_formwork,volume_reinforcement_walls,volume_reinforcement_floor
floor_formwork,volume_reinforcement_walls,volume_reinforcement_floor
volume_placing_backfill_U_basin_open_pit,volume_import_backfill_U_basin_open_pit,
volume_export_sand_U_basin_open_pit,volume_export_peat_U_basin_open_pit
,volume_export_clay_U_basin_open_pit,volume_export_total_U_basin_open_pit,
volume_excavation_U_basin_open_pit,volume_re_used_sand_U_basin_open_pit,
total_length_tension_piles_U_basin,total_volume_tension_piles_U_basin,Width,Length)

outputfile_U_basin_steep_pit,MKI_outputfile_U_basin_steep_pit,CO2_tax_outputfile_U_basin_steep_pi
t =
U_basin_steep_pit_costs(input_U_basin_steep_pit,length,range_length_pile,range_number_of_anchor,
range_sheetpile,sheetpile_length_u,sectional_area_sheetpile_u,number_of_anchors_u,volume_placing_b
ackfill_U_basin_steep_pit,volume_import_backfill_soil_U_basin_steep_pit,volume_export_sand_U_basi
n_steep_pit,volume_export_peat_U_basin_steep_pit,
volume_export_clay_U_basin_steep_pit,volume_export_total_U_basin_steep_pit,excavation_volume_U_b
asin_steep_pit,volume_re_used_sand_U_basin_steep_pit,volume_underwater_concrete_floor_total_u,tot
al_length_tension_piles_U_basin_steep_pit,total_volume_tension_piles_U_basin_steep_pit,Costs_of_C
oncrete_U_basin,Costs_of_Reinforcement_U_basin,Costs_of_Formwork_U_basin,MKI_formwork_U_basin,MKI
concrete_U_basin,MKI_reinforcement_U_basin,CO2_tax_formwork_U_basin,CO2_tax_concrete_U_basin,CO2
tax_reinforcement_U_basin)

with pd.ExcelWriter(r'C:\Users\908729\Documents\Output-Tool\New_runs'+ main_folder +'('))
+ folder[i] + '(costs_soil_adaption_CO2_U.CO.xlsx)' as writer: # doctest: +SKIP
if version_check == False:
version_check.to_excel(writer,sheet_name='VERSION_WARNING')
outputfile_sheetpiles.to_excel(writer,sheet_name='Output_Sheet_Piles')
outputfile_diaphragm_wall.to_excel(writer,sheet_name='Output_Diaphragm_Wall')
outputfile_U_basin_open_pit.to_excel(writer,sheet_name='Output_U_basin')
outputfile_U_basin_steep_pit.to_excel(writer,sheet_name='Output_U_basin_steep_pit')
outputfile_MKI_sheetpile.to_excel(writer,sheet_name='MKI_sheetpile')
outputfile_MKI_diaphragm_wall.to_excel(writer,sheet_name='MKI_diaphragm_wall')
outputfile_MKI_U_basin_open_pit.to_excel(writer,sheet_name='MKI_U_basin_open_pit')
MKI_outputfile_U_basin_steep_pit.to_excel(writer,sheet_name='MKI_U_basin_steep_pit')
outputfile_CO2_tax_sheetpile.to_excel(writer,sheet_name='CO2_tax_sheetpile')
outputfile_CO2_tax_U_basin_open_pit.to_excel(writer,sheet_name='CO2_tax_U_basin_open_pit')
outputfile_CO2_tax_U_basin_steep_pit.to_excel(writer,
sheet_name='CO2_tax_U_basin_steep_pit')
CO2_tax_outputfile_U_basin_steep_pit.to_excel(writer,
sheet_name='CO2_tax_U_basin_steep_pit')

input_values_per_depth.to_excel(writer,sheet_name='Soil_Profile')
data = pd.read_excel(r'C:\Users\908729\Documents\Output-Tool\New_runs'+ main_folder
+'(')) + folder[i] + '(Masterfile_input.xlsx',sheet_name='Master_Sheet')
data1 = pd.read_excel(r'C:\Users\908729\Documents\Output-Tool\New_runs'+ main_folder
+'(')) + folder[i] + '(Masterfile_input.xlsx',sheet_name='Input_soil_layers')
data2 = pd.read_excel(r'C:\Users\908729\Documents\Pycharm_tool\Masterfile_input.xlsx',
sheet_name='Input_Soil_Costs')
data.to_excel(writer,sheet_name='Input_master_sheet')
data1.to_excel(writer,sheet_name='Input_soil_layers')
data2.to_excel(writer,sheet_name='Input_Soil_Costs')

```

Figure F-34: Code Excel Output