Moving load on guide works during a brush collision with a vessel

Master thesis *Civil Engineering - Structural Engineering*

W.J.G. van Dommelen

June 2019 – February 2020

Moving load on guide works during a brush collision with a vessel

Master thesis *Civil Engineering - Structural Engineering*

to obtain the degree of Master of Science
at the Delft university of Technology,
to be defended publicly on February 14th, 2020

Student

Willem van Dommelen

Date

June 2019 – February 2020

Education

Master Civil Engineering, Structural Engineering – Delft University of Technology Structural Mechanics

Assessment committee

Dr. ir. M.A.N. Hendriks	Delft University of Technology **(chair)**
Dr. ir. P.C.J. Hoogenboom	Delft University of Technology
Ir. W.F. Molenaar	Delft University of Technology
Ir. P.J. Plooy	Lievense|WSP

Preface

May I present to you, my master thesis for the completion of the master Civil Engineering at Delft University of Technology. Herewith I complete the track of *Structural Engineering* that belongs to this master. Brushing of inland navigation vessels with guide works in the time domain is the topic of the thesis. The research is executed at Lievense|WSP in Breda in the department of Hydraulic Engineering. After completion of this master thesis, I am excited to start my engineering life!

Very grateful I am to have obtained the possibility to do my master thesis at Lievense|WSP, and therefore special gratitude goes out to Maurice de Kroon and Peter Jan Plooy. I would like to thank dr.ir. M.A.N. Hendriks, ir. W.F. Molenaar and dr.ir. P.C.J. Hoogenboom for their efforts as well. Without their feedback and assistance, it would have been impossible to reach the results of this thesis. Finally, I would like to thank my friends and family and wish you all a pleasant reading!

February 2020

Willem van Dommelen

Abstract

In the vicinity of locks and bridge piers, guide works are placed in order to steer inland navigation vessels through a narrowed passage. The design of these structures is generally done in a static way, thus without consideration of time-dependent phenomena.

By treating the contact force during a brush collision as a static force, several phenomena are overlooked. First of all, it is possible that, in case of a brush collision, a second impact occurs due to the yawing motion of the vessel that is generated during the first impact. Considering the time domain in the analysis of brush collisions also enables the investigation of the length that is required for a guide work to fulfil its function. Within this thesis the time aspect is considered for brush collisions between inland navigation vessels and guide works consisting of steel hollow tubular cross-sections.

Objective of this research is to quantify the second impact and to provide a recommendation for the length of guide works, in order to answer the question: “In what ways is the design of guide works affected by taking into account the time domain?”.

A parametric tool that simulates brush collisions is created in Python. By variation of one parameter at a time, the influence of this parameter on the magnitude of the second impact and the required length for a guide work is visualised. The nine parameters that are investigated are: the mass, length and draught of the vessel, the outside diameter of the horizontal berthing beam and the vertical piles, the length of the vertical piles above bed level, the initial velocity of the vessel, the initial angle of contact between the vessel and the structure and the location where the first contact takes place.

For all investigated parameters a second impact occurs within the 90 seconds of performed analysis, except for vessels with a large mass or length, or a small initial velocity. On top of that, the second impact appears to be generally governing over the first impact, while guide works are designed for the first impact. Only for initial contact angles larger than 13 degrees or vessels with a draught of more than 7 meters, the first impact is governing.

In most cases the maximum occurring force during the second impact is 30-40% larger than during the first impact. Exceptions are the initial angle of contact and the draught, where an increase of these two parameters leads to a reduction of the ratio between the maximum force during the second and the first impact. Contrary, an increase of the mass of the colliding vessel, also leads to an increase of the ratio between the maximum contact force during the second and the first impact.

During the first impact, the majority of the original kinetic energy due to the translational velocity of the vessel can be converted into rotational kinetic energy and does not need to be absorbed by the structure. During the second impact, the vessel already contains a rotational movement, which needs to be reverted in order to loose contact with the structure again. This is only possible when the kinetic energy due to rotation is absorbed by the structure as potential energy. Increasing potential energy in the structure, subsequently, leads to more deformation which then results in a larger contact force.

The required length of guide works to facilitate a brush collision consists of the length over which the first contact takes place, plus the distance between the location of the first contact of the first impact and the location of the first contact of the second impact.

Reduction of the overall stiffness of the guide work by decreasing the diameters of the circular tubular sections, or an increase of the pile length above bed level, leads to an increase of the duration of the collision. Furthermore, a larger initial velocity, mass and length of the vessel also result in a longer required length for the guide work, while an increase of the draught and initial contact angle lead to a decrease of this needed length.

Given the assumptions during the analyses, e.g. negligence of human intervention, it is recommended to perform field tests to calibrate the created tool. Nonetheless, it is advised to use the created tool for future projects, since it provides a substantiated indication of the contact force during the second impact, which generally governs the first impact.

Lastly, the model made in Python is converted to the finite element software *SCIA Engineer.*

Contents

List of Figures

List of Tables

List of Symbols

Latin:

A	Frontal area	[m^2]
A	Location of contact of the stern	*n.a.*
A	Area underneath deflection curve	[m^2]
a	Distance in x-direction between V and A	[m]
A_{ij}	Added mass per unit length	[kg/m]
A_R	Reference area	[m^2]
B	Scaling quantity for the kinetic energy of the guide work	[m^2]
b	Distance in y-direction between V and A [m]	[m]
B_{ship}	Width of the vessel	[m]
C	Centre of gravity vessel	*n.a.*
c	Distance in y-direction between C and A	[m]
C_A	Dimension dependent coefficient	[-]
C_{ab}	Abnormal berthing factor	[-]
C_B	Block coefficient	[-]
C_c	Waterfront structure attenuation factor	[-]
C_E	Eccentricity factor	[-]
C_m	Virtual mass factor	[-]
C_s	Ship flexibility factor	[-]
C_w	Drag coefficient	[-]
d	Distance in x-direction between C and A	[m]
d	Draught	[m]
D_{hor}	Outside diameter horizontal berthing beam	[m]
D_r	Depth rotation point Brinch Hansen method	[m]
D_{ver}	Outside diameter vertical piles	[m]
E_abs	Absorbed energy	[Nm]
E_abs 1st impact	Maximum potential energy in system during first impact	[Nm]
E_abs 2nd impact	Maximum potential energy in system during second impact	[Nm]
E_drag_rot	Work done by the drag in the rotational direction	[Nm]
E_drag_trans	Work done by the drag in the translational direction	[Nm]
E_hor_beam	Potential energy horizontal berthing beam	[Nm]
E_kin_str	Kinetic energy structure	[Nm]
E_kin_y_ship	Kinetic energy vessel due to the velocity in y-direction	[Nm]
E_piles+soil	Potential energy soil and vertical piles	[Nm]
E_rot_ship	Kinetic energy vessel due to the rotational velocity	[Nm]
$E_{drag,rot}$	Work done by drag in the rotational direction	[Nm]
$E_{drag, trans}$	Work done by drag in the translational direction	[Nm]
$E_{friction}$	Work done by friction	[Nm]
$E_{initial}$	Total energy in the system at the start of the analysis	[Nm]
E_{kin}	Kinetic energy	[Nm]
$E_{kin;d}$	Design berthing energy	[Nm]
$E_{kin;k}$	Characteristic berthing energy	[Nm]
$E_{kin;rot}$	Kinetic energy due to rotational movement	[Nm]
$E_{kin;structure}$	Kinetic energy structure	[Nm]
$E_{kin;trans}$	Kinetic energy due to translational movement	[Nm]
E_{pot}	Potential energy in a beam subjected to bending	[Nm]
EI	Bending stiffness	[Nm^2]
F	Force	[N]
F1_max	Maximum occurring contact force first impact	[N]
F2_max	Maximum occurring contact force second impact	[N]
F_d	Largest design berthing impact load	[N]
$F_{drag,max}$	Largest drag force per unit of depth	[N/m]
$F_{drag,rot}$	Drag force influencing the rotation of the vessel	[N]
$F_{drag,trans}$	Drag force influencing the translation of the vessel	[N]
F_k	Characteristic berthing impact load	[N]
F_Q	Design berthing impact load	[N]
F_s	Design berthing impact load	[N]

F_v	Impact load	[N]
H	Horizontal load Brinch Hansen method	[N]
H_{ship}	Total height of the vessel	[m]
I_G	Mass moment of inertia vessel	[kgm^2]
I_z	Moment of inertia in around the z-axis	[m^4]
k	Radius of gyration of the vessel	[m]
k_{rot}	Rotational spring stiffness	[Nm/rad]
k_{trans}	Translational spring stiffness	[N/m]
k_{tot}	Total representative stiffness	[N/m]
L	Length	[m]
L_f	Length lead in area	[m]
L_{hor}	Centre to centre distance vertical piles	[m]
L_k	Chamber length	[m]
L_o	Length lay-by area	[m]
L_{out}	Run out length	[m]
L_{pile}	Total length vertical piles	[m]
L_{req}	Required guide work length	[m]
L_{seg}	Length segment	[m]
L_{ship}	Length vessel	[m]
L_{tot}	Total length	[m]
$L_{ver;0}$	Length between bed level and horizontal berthing beam	[m]
L_w	Length waiting area	[m]
M	Mass structure	[kg]
M	Moment	[Nm]
m	Mass vessel	[kg]
N	Number of segments	[-]
N_{seg}	Number of segments	[-]
$q_{f;k}$	Characteristic value live load	[N/m^2]
r	Distance mass centre vessel to contact point	[m]
R_i	Distance mass centre segment to mass centre vessel	[m]
R_3	*Ersatzkraft*	[N]
S	Momentum	kg m/s]
T	Length of one sinusoidal period	[s]
t	Fictitious embedment depth	[m]
t_{hor}	Thickness horizontal berthing beam	[m]
t_{ver}	Thickness vertical piles	[m]
t_0	Start of analysis	[s]
t_0	Theoretical penetration depth	[m]
$t_{0'}$	Time immediately after first exchange of momentum	[s]
t_1	Time of maximum contact force first impact	[s]
t_2	End first impact	[s]
t_3	Start second impact	[s]
t_4	Time of maximum contact force second impact	[s]
t_5	End second impact	[s]
V	Location of contact of the bow	*n.a.*
v	Total translational velocity of the mass centre of the ship	[m/s]
v_k	Characteristic berthing velocity	[m/s]
w_{max}	Maximum occurring deflection within one timestep	[m]

Greek:

γ_Q	Partial safety factor berthing impact load	[-]
γ_{sd}	Partial safety factor load modelling	[-]
γ_v	Partial safety factor berthing velocity	[-]
ΔE_{abs}	Increase/decrease potential energy nonlinear springs	[Nm]
ΔL_1	Length required to facilitate the first impact	[m]
ΔL_2	Length between first contact bow and first contact stern	[m]
Δt	Timestep	[s]
δ	Displacement	[m]
η	Scaling factor kinetic energy guide work	[-]
θ	Angular position of the vessel with respect to the x-axis	[°]

μ	Friction coefficient	[-]
μ	Mean value	*n.a.*
ρ	Density	[kg/m^3]
ρ_{model}	Modelled density	[kg/m^3]
ρ_{water}	Density water	[kg/m^3]
φ	Angle between velocity vector v and r	[°]
ω	Angular velocity of the vessel	[rad/s]
ω	Frequency of the collision	[rad/s]

Abbreviations, accents and subscripts:

COG	Centre of gravity
$[\dot{x}]$	Derivative of x with respect to time
$[\ddot{x}]$	Second derivative of x with respect to time
$[..]_A$	Of the stern
$[..]_C$	Of the mass centre
$[..]_i$	Index i
$[..]_j$	Index j
$[..]_V$	Of the bow
$[..]_x$	In x-direction
$[..]_y$	In y-direction
$[..]_z$	In z-direction

1. Introduction

In the vicinity of bridges and locks, inland navigation vessels are frequently navigating through a smaller shipping lane to pass these structures. In order to let these ships pass safely, guide works are placed on both sides in front of the engineering structure to steer ships in the right direction. In absence of guide works, on December 5th 2019 a tanker collided with a railway bridge pier near Zaltbommel (Omroep Gelderland, 2019). Luckily, no victims and damage on the bridge pier were found, but the consequences could have been disastrous.

In principle, the steering of the vessels should be done visually. However, in practice it is also possible that a brush collision between a vessel and the guide work takes place. An example of a guide work is given in Figure 1.1.

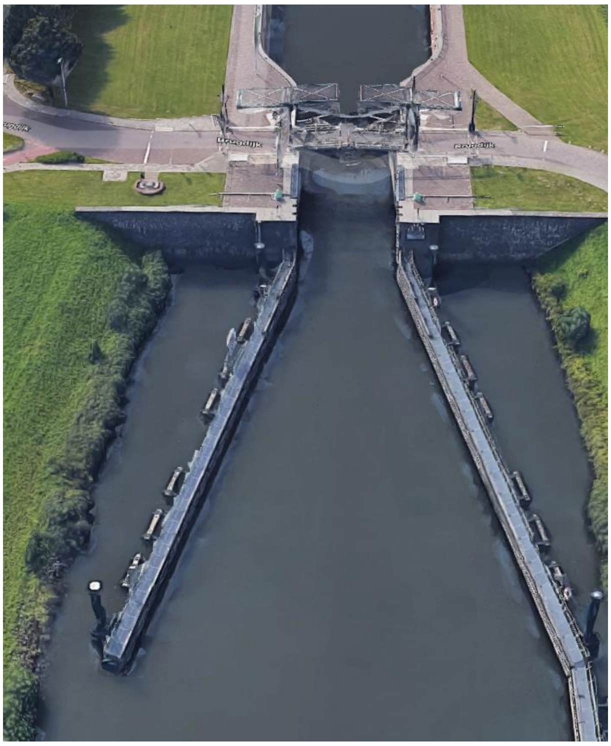


Figure 1.1 Example of a guide work in Vianen (Google Maps, 2018)

In the Netherlands many of the existing bridges and locks were built in the ‘50s and ‘60s of the previous century in the aftermath of the Second World War. Consequently, the end of their intended life span is approaching in the upcoming years and maintenance or renovation is required (Rijkswaterstaat, 2018).

In addition to the enormous maintenance and replacement task that the Directorate-General for Public Works and Water Management, *Rijkswaterstaat,* is facing, there is also an increase in inland navigation transport over the past years, (Rijkwaterstaat, sd), (CBS, 2018). In order to maintain safe navigation conditions, it is necessary to have a proper estimation of the loading on the guide works and to design them accordingly.

From the different loading conditions that exist for guide works, collisions between a vessel and the structure are generally governing (SBRCURnet, 2018). In common design practice, the load that is associated with such a collision is determined by a certain amount of kinetic energy of the colliding vessel that needs to be absorbed by the guide work by means of deformation. Based on this deformation a corresponding design force is determined. The design force, with the appropriate safety factors, is then treated as a static design force.

1.1 Problem definition

By treating the collision force as a static force, and thus not taking into account the time domain, several phenomena are overlooked. Three main issues are discussed below.

First of all, it takes approximately a quarter of a period of the collision, mostly 2 to 10 seconds depending on the design parameters, before the maximum force during a collision occurs (Vrijer, 1978). As a result, the first vertical pile that is encountered by a moving vessel, will not be loaded as heavily as the other vertical piles of the guide work, since the ship also contains a velocity in the longitudinal direction. The dimensions of the first pile could then be reduced to save material costs.

Secondly, the contact point between the vessel and the structure is eccentric with respect to the mass centre of the ship. The eccentricity generates a yawing movement of the vessel and due to this yawing it is possible that the stern collides with the guide work as well. Measurements for the berthing velocity for this so-called *second impact* have been performed for VLCCs[1] (Bratteland, 1988) and for tankers and bulkers (Roubos, Groenewegen, & Peters, 2016). The berthing velocity of the second impact appeared to be in the same order of magnitude as the first impact. However, a computational motivation for the magnitude of the second impact is still lacking.

A final consequence of not taking into account the time domain is related to the length of a guide work. In case of a second impact, there is a certain distance between the first impact location and the second impact location on the guide work. On top of that, the length over which the contact takes place, also needs to be accommodated, see also Figure 1.2. In order to realise a more economic design by avoiding too long structures, knowledge about these two lengths is needed.

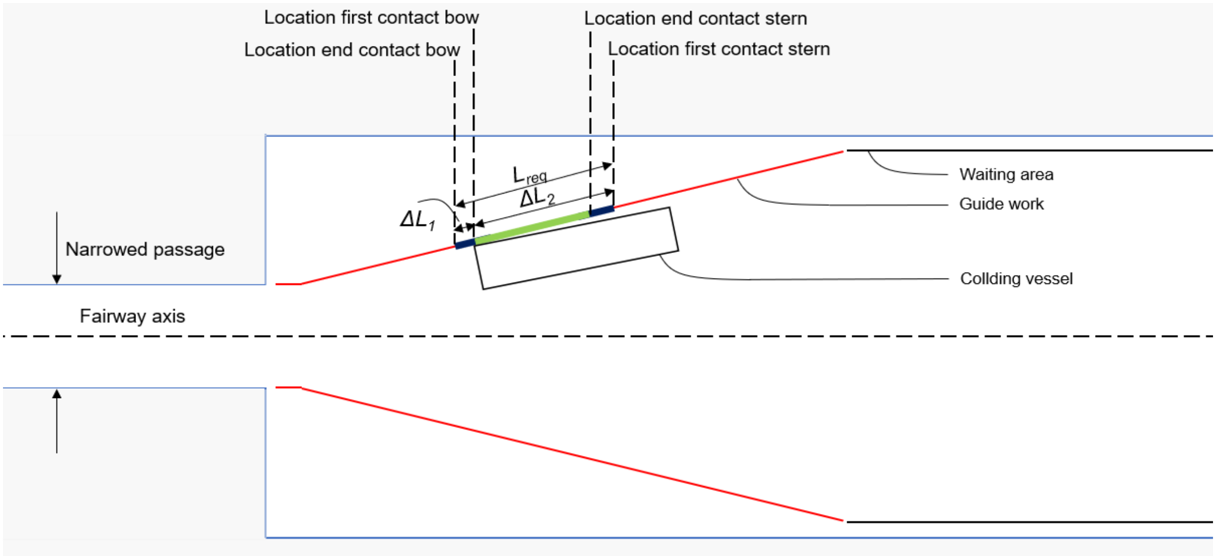


Figure 1.2 Required length of a guide work to facilitate a brush collision with a vessel

1.2 Objectives

Based on the problems discussed in the previous section, the following research question can be formulated:

In which ways is the design of guide works affected by taking into account the time domain?

[1] VLCC stands for *Very Large Crude Carrier*

In order to answer this research question, also three sub questions are formulated. Each sub question focusses on one of the problems discussed in section 1.1. The sub questions are given below:

1) **How do the contact force and the location of the contact vary during a brush collision between an inland navigation vessel and a guide work?**
2) **What is the maximum contact force in case of a second impact due to a brush collision between an inland navigation vessel and a guide work?**
3) **What length of a guide work is required to facilitate a brush collision between a guide work and an inland navigation vessel?**

The three sub questions lead to three objectives as well, which are summarized below:

- Present the development of the contact point and the magnitude of the contact force in the time domain during a brush collision between an inland navigation vessel and a guide work for multiple scenarios.
- Provide a quantification for the magnitude of the second impact due to a brush collision between an inland navigation vessel and a guide work for multiple scenarios.
- Provide a recommendation for the length of a guide work for various design conditions

1.3 Scope and methodology

The focus of this thesis is on brush collisions between inland navigation vessels and steel tubular guide works. The colliding vessel is schematised as a rigid rectangle. The physical nonlinearity of the soil is taken into account, as well as hydrodynamic effects acting on the colliding vessel.

In order to fulfil the objectives, a model is created that describes the collision process in the time domain. On the one hand, this model presents the forces that are generated by the brush collision in the course of time, while on the other hand, the translational and rotational movements of the vessel are described.

The model entails a Python script that simulates the brush collision between a guide work and an inland navigation vessel by means of an explicit time-stepping scheme and a non-linear analysis.

At the start of the script the dimensions of the ship, the initial conditions and the geometry and the cross-sections of the guide work are defined. Since these values are processed in a parametric manner in the model, different design scenarios can easily be compared with each other.

In total nine different input parameters are varied to investigate the influence of these parameters on the duration of a brush collision, the maximum force that occurs during a brush collision and the length over which this collision takes place.

The nine parameters that are varied within this study, are listed below:

1. The mass of the vessel
2. The length of the vessel
3. The draught of the vessel
4. The outside diameter of the horizontal berthing beam
5. The outside diameter of the vertical piles
6. Length of the vertical piles above bed level
7. The initial location of the contact point on the guide work
8. The initial angle of contact
9. The initial velocity of the vessel

As it is expected that the Python model gives a more accurate description of the external loading on a guide work than the conventional design method (see section 2.3), a translation of the Python script to SCIA Engineer is made as extra part of the project scope.

This translation enables the company to use the model for brush collisions in the time domain in future projects, since the Python model has some limitations as will be discussed in section 4.4. Especially the automatic postprocessing of the SCIA Engineer model is a great advantage, as well as the geometrical freedom of the guide work in the performed analyses.

One of the main limitations of the Python model is that the vertical piles are assumed to be prismatic, while in reality a variation in wall thickness is applied on the vertical piles to achieve a more economic design.

1.4 Report structure

The thesis starts with a description of dolphins, with guide works in particular. Then, impact loading on guide works is discussed within chapter 3. Subsequently, in chapter 4 the Python model is described, as well as its translation to SCIA Engineer. In the proceeding chapter, chapter 5, a verification of this Python model is presented. Chapter 6 shows the results that are found during the research. The results are followed by the discussion in chapter 7. Finally, chapter 8 provides the conclusions and recommendations of the thesis. A schematic representation of the report structure is presented in Figure 1.3.

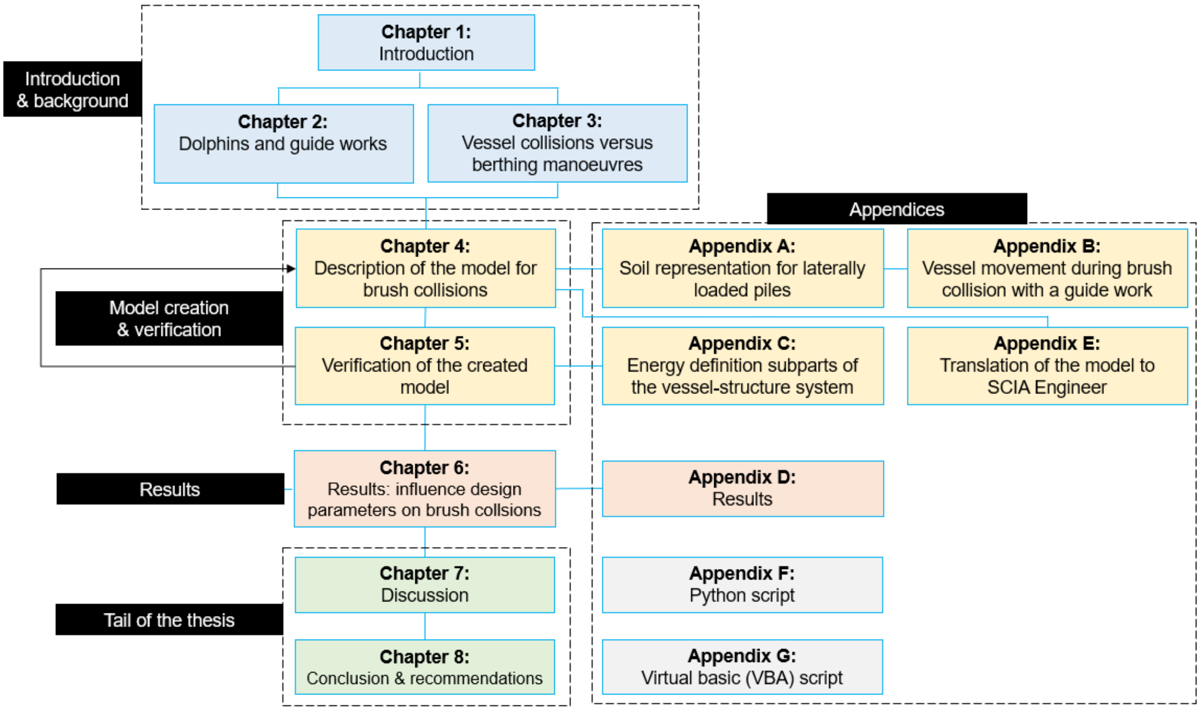


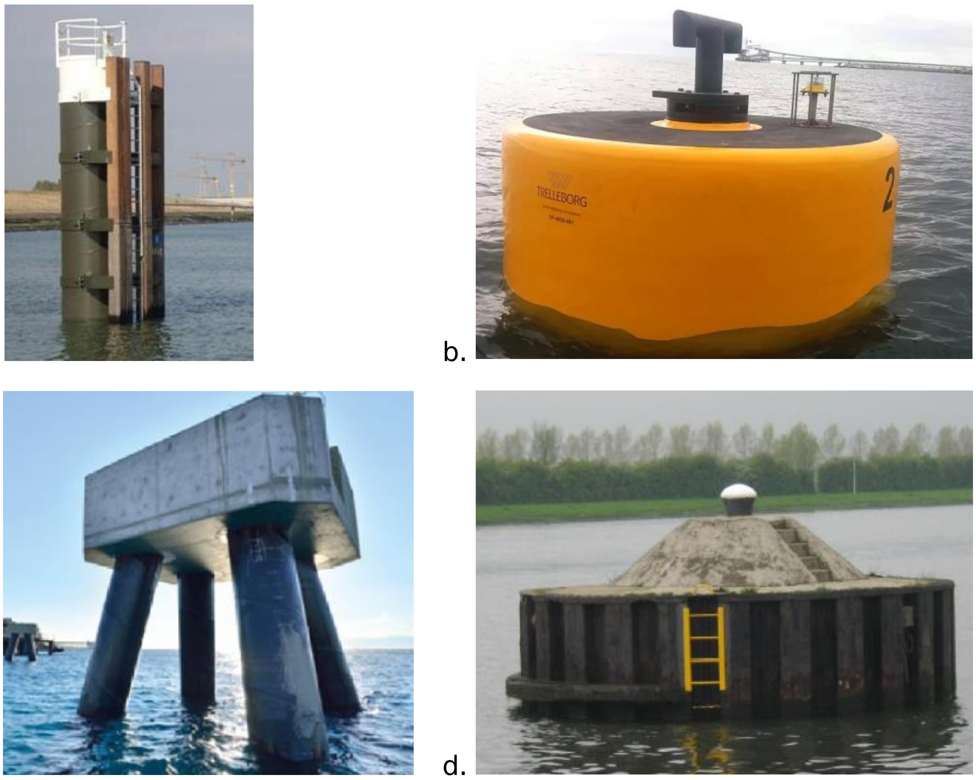
Figure 1.3 Schematic representation report structure

From Figure 1.3 it can be seen that also seven appendices are added to the report to reduce the length of the main report, and improve its readability. In appendix A different soil models are presented for the modelling of laterally loaded piles. In appendix B the movement of a vessel during collision with a guide work is given in more detail than in section 4.2.2. On top of that, the incorporation of the influence of the water surrounding the vessel is described in this appendix. Subsequently, in appendix C the definition of the energy in the different components of the Python model is given.

Then, in appendix D the results of the research are presented. In appendix E the translation of the Python model to SCIA Engineer is provided. Subsequently, in appendix F and appendix G the Python script and the Virtual Basic script are given respectively.

2. Dolphins and guide works

Guide works are a type of so called *dolphin structures*, or *dolphins* in short. Dolphins are man-made structures in the marine environment that extend above water level and they can roughly be divided into four categories, being: flexible piled dolphins, floating dolphins, fixed piled dolphins and gravity-based dolphins. Examples of the four different dolphin types can be found in Figure 2.1a-d respectively.



a. b. c. d.

Figure 2.1 Dolphin types: a) flexible piled dolphin, b) floating dolphin, c) rigid piled dolphin and d) gravity-based dolphin (SBRCURnet, 2018)

Guide works belong to the type of the *flexible piled dolphins* or *flexible dolphins* in short. A characteristic of flexible dolphins, and thus for guide works, is that the loading on the structure is taken up by means of deformation.

This chapter gives insight in the loading conditions for flexible dolphins. Every type of dolphin has its own specific type of loads, i.e. static loads, dynamic loads and impact loads. Obviously, the main focus of this chapter is the loading on guide works, but also loading on different flexible dolphins is discussed to place guide works into perspective.

2.1 Functions of a dolphin

The first step in specifying the loads working on a structure is to determine the function of the structure. In general, dolphins roughly fulfil three functions and have thus multiple loading conditions. The function of a dolphin are given in section 2.1.1, while the function of a guide work are presented in section 2.1.2.

2.1.1 General functions of a dolphin

The first function is to provide safe berthing and mooring possibilities. This entails that the structure should be able to transfer the berthing energy from the vessel to the soil. Also, the contact pressure between the vessel and the dolphin should be limited in order to prevent damage to the vessel's hull. On top of that, line forces of moored ship need to be taken up, also in bad weather conditions with strong winds and waves.

Second function of flexible dolphins is to protect other structures. This can be realised in the form of a sacrificial structure, e.g. a crash barrier. Failing of the sacrificial structure in this case is allowed, since it prevents the protected structure from being damaged. However, not all protecting structure may be regarded as sacrificial structures. Guide works, for example, can also be seen as protecting structures. Nonetheless, they are not allowed to completely deform under the given loading conditions, nor are they allowed to deform plasticly. (Rijkswaterstaat, 2017)

A third function of dolphins is to guide the nautical traffic. Primarily, the guidance is done in a visual manner, but it is also possible for these structures to steer the vessels in the right direction in a physical way, i.e. by means of direct contact. (SBRCURnet, 2018)

2.1.2 Functions of a guide work

In Figure 2.2 a schematic representation of the approach area of a lock is given. In case of narrowed shipping lanes near bridges a similar layout of the approach is present. The guide work in Figure 2.2 is indicated with a red rectangle.

As can be seen from the figure, guide works are placed under an angle to narrow the shipping lane and guide the vessels into the right direction. The angle of a guide work should be at least 1:4 according to *Richtlijn vaarwegen 2017* by Rijkswaterstaat. In extension of the wall of the lock chamber, the guide work also contains a straight part besides the oblique part. In current design guidelines this horizontal length is dependent on the width of the lock chamber, the angle of the guide work and the width of the governing ship, but should be at least 3 meters (Rijkswaterstaat, 2017).

Contrary to most dolphins, guide works are not intended to provide mooring and berthing possibilities, since mooring of vessels to a guide work would obstruct the shipping lane. Therefore, only two of the three functions of a dolphins remain for a guide work, i.e. protection of other structures and guidance of nautical traffic.

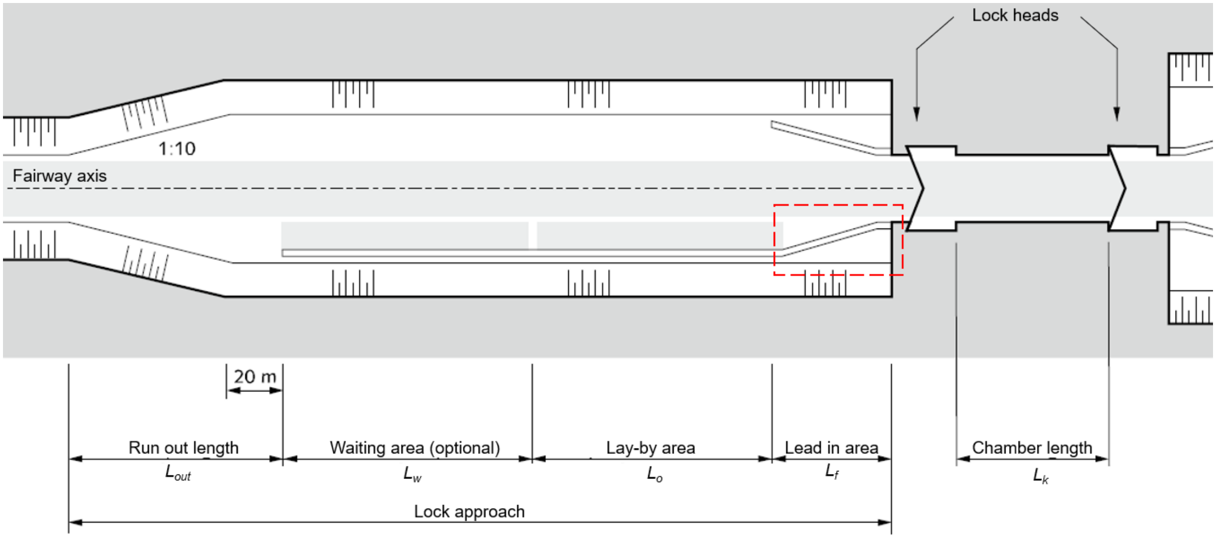


Figure 2.2 Schematic representation lock approach (Rijkswaterstaat, 2017)

To improve the readability of the report, Figure 2.3 is included. Since consensus about all English translations for Dutch terms with respect to this type of structures is not reached, Figure 2.3 gives the definition of terms as they are used within this report.

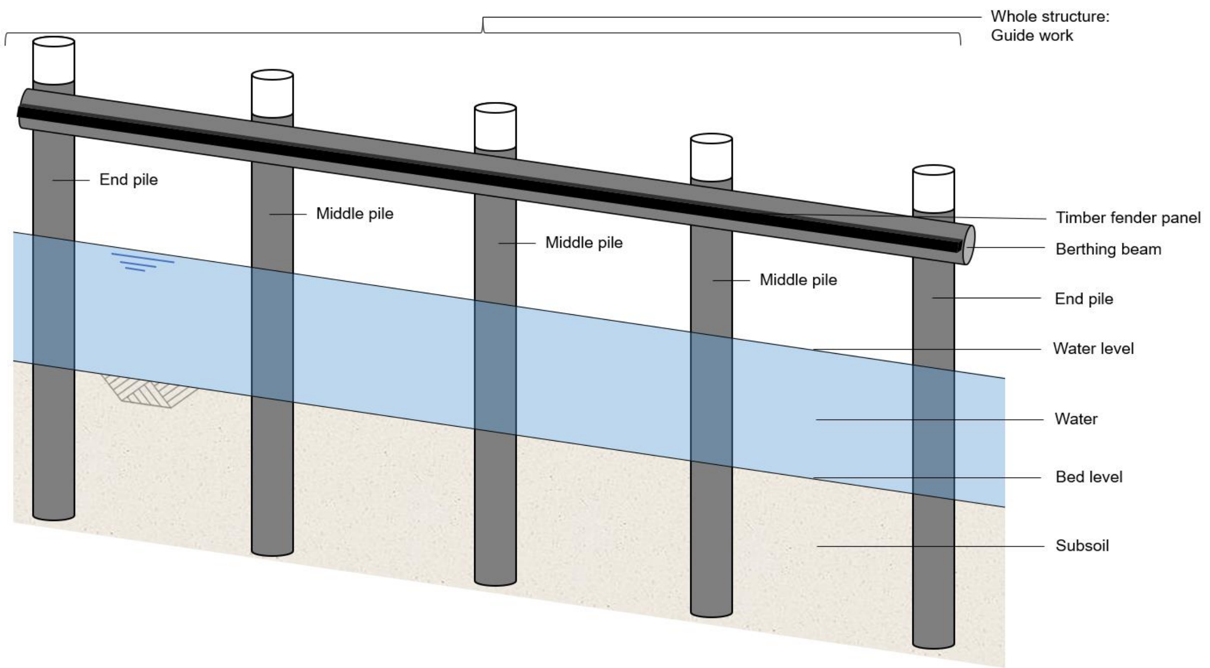


Figure 2.3 Terms and definitions guide work

2.2 Codes and guidelines for the design of guide works

Based on the functions of the structure, several loading conditions can be distinguished. In this section, first an overview of the existing norms and guidelines for the design of flexible dolphins is given. Subsequently, the loading itself is discussed. Many codes and guidelines exist for flexible dolphins, parts of it and the materials that are used. The most important ones are outlined within this subsection.

First of all, the Eurocode needs to be mentioned. The Eurocode can be divided into nine different parts, of which four are directly relevant for steel tubular guide works, being EN 1990, EN 1991, EN 1993 and EN 1997. The EN 1990 describes the structural safety, serviceability and durability, while the EN 1991 discusses actions on structures. EN 1993 entails the design of steel structures and steel connections, and finally, EN 1997 prescribes the geotechnical aspects. (Joint Research Centre Eurocodes, sd)

Based on practical experience and research, several guidelines have been created. One of these guidelines is the EAU (Empfehlungen des Arbeitsausschusses "Ufereinfassungen" Häfen und Wasserstraßen). After introduction of the Eurocodes, the EAU has adopted the safety philosophy of the Eurocodes. (EAU2012, 2012)

In the Netherlands, Rijkswaterstaat has developed the ROK (*Richtlijn Ontwerp Kunstwerken)* in which, among other structures, the design of marine structures is discussed (Rijkswaterstaat, 2017). Apart from the ROK, Rijkswaterstaat has also created the *Richtlijnen Vaarwegen*, that specifically focusses on the design of waterways (Rijkswaterstaat, 2017). On top of that, the BSI (British Standards Institution) has developed their own standards for flexible dolphins. (BSI, sd)

To conclude the list of codes and guidelines that are pointed out in this report, the PIANC needs to be mentioned. The PIANC is the world association for waterborne transport infrastructure and they organise events, e.g. congresses, in which experience, expertise and knowledge is shared (PIANC, sd).

The broad range of different recommendations generated the need for a comparison between the various codes and guidelines. In 2018, the book *Flexible Dolphins* was published under supervision of the SBRCURnet Committee 1720 'Dolphins'. This book compares differences between the various codes and recommendations for flexible dolphins, and aims to provide help in the design process of flexible dolphins. Also various researches have been taken into account in writing this book. (SBRCURnet, 2018)

2.3 Loads on guide works

Since flexible dolphins, and thus guide works, have multiple functions, also more than one loading condition can be distinguished. The following common loads for dolphins can be mentioned:

- Dead weight
- Line forces of moored ships
- Wind load
- Variable vertical loads
- (Corrosion)
- Berthing and collision impacts

2.3.1 Dead weight

Apart from the dead weight of the steel tubes that form the guide work, also timber fender panels may be present. Function of these timber panels is to prevent damage to the steel structure or the vessel's hull.

Furthermore, it is possible that walking facilities are installed on top of the structure, resulting in additional dead weight. Finally, it can be mentioned that dolphins may be equipped with a fender system to increase the energy absorption capacity of the system. However, guide works for inland navigation traffic are not commonly furnished with such systems and are therefore not considered within this thesis. An example of a fender system is given in Figure 2.4.

Figure 2.4 Example of a fender system (Trelleborg, 2016)

2.3.2 Line forces of moored ships

It is common design practice to express mooring loads that are caused by currents and wind as a static loading. Among others, the EAU 2012 and BS 6349-1 have incorporated a method to determine the static forces as a result of the wind and the current action.

A second possibility to obtain the static design force is to use a table provided by the EAU 2012 for characteristic line forces (EAU 2012, 2012), see Table 2.1. As the values in the table are characteristic values, they still need to be multiplied with safety factors depending on the consequence class of the structure in order to obtain the design value of the load, see also section 2.4.

Table 2.1 Characteristic line pull forces (EAU 2012, 2012)

Displacement [t]	Line pull force [kN]
Up to 10 000	300
Up to 20 000	600
Up to 50 000	800
Up to 100 000	1000
Up to 200 000	2000
Up to 250 000	2500
> 250 000	>2500

2.3.3 Wind load

The main influence of the wind is hidden in the mooring line forces. Since the height of the structure is limited, also the wind loading on the structure is limited. Moored vessels, however, may have a more substantial height and catch therefore more wind. On top of that, wind may generate waves resulting in higher line forces.

2.3.4 Variable vertical loads

Variable vertical loads will predominantly arise from a walking facility, when present. Characteristic values for the live loads lie between $q_{f;k}$ = 2.5 kN/m² and $q_{f;k}$ = 5.0 kN/m² (NEN, 2011). Other variable loads may arise from vertical friction of a moored vessel due to tidal movements or waves. Friction loads can be obtained by multiplying the normal force with a friction coefficient μ. Depending on the material of the fender panels, μ varies between a value of 0.2 for polymers and 0.7 for rubbers. The friction coefficient for timber is approximately equal to 0.4. (SBRCURnet, 2018)

2.3.5 Corrosion

Apart from loads in terms of forces, dolphins also have to sustain corrosion effects. Environmental effects may result in serious deterioration of the applied steel. Reduction of the wall thickness leads to a reduction of the load-bearing capacity. Corrosion is strongly dependent on local parameters. It is therefore advised to make use of reference projects to find an appropriate corrosion rate [mm/year]. Also, the use of corrosion protective measures is strongly recommended.

2.3.6 Impact loading

Although the line forces are considerable, impact loadings are mostly governing. Impact loadings are mainly caused by the berthing of ships. Impact loading may also arise from a head-on collision or a brush collision. Different types of impact loading are discussed in chapter 3.

Guide works are commonly built in such a manner that head-on collisions are not possible, or taken up by sacrificial dolphins. In essence, brushing collisions are similar to berthing procedures, but with higher velocities at smaller angles.

Given the high frequency of ship berthing to flexible dolphins, this impact loading is not considered as ‘accidental’ loading. Due to the same reason, brush collisions are neither considered as ‘accidental’ loading. As a consequence of this classification, the structure should be able to withstand the loading conditions without failure and without excessive deformations, thus complying with the SLS (Serviceability Limit State) and ULS (Ultimate Limit State) criteria.

Contrary to common design practice, impact loading due to colliding vessels, in first instance, is not expressed in terms of forces. The loading consists of a certain amount of kinetic energy that needs to be absorbed by the structure. Flexible dolphins, and thus guide works, absorb this energy by means of deformation.

Different codes use comparable expressions for the amount of energy that needs to be absorbed. EAU 2012 makes use of Equation (2.1), while similar definitions are incorporated in PIANC 2002 and BS6349-4. (SBRCURnet, 2018).

$$E_{kin} = \frac{1}{2}\, m\, v^2\, C_m\, C_s\, C_c\, C_E \tag{2.1}$$

where:

E_{kin}	Kinetic energy of the vessel to be absorbed by the dolphin	[Nm]
m	Mass of the vessel	[kg]
v	Total translational velocity of the centre of gravity of the ship	[m/s]
C_m	Virtual mass factor	[-]
C_s	Ship flexibility factor	[-]
C_c	Waterfront structure attenuation factor	[-]
C_E	Eccentricity factor	[-]

Virtual mass factor C_m:
The virtual mass factor accounts for the mass that is added due to the displacement of the water. The added mass is dependent on the keel clearance. In case of a transverse approach (θ>5°), PIANC recommends a value for C_m as: (PIANC, 2002)

C_m = 1.5 for a keel clearance of 0.5 times the draught of the ship
C_m = 1.8 for small keel clearance of 0.1 times the draught of the ship

For keel clearance in the range of 0.1-0.5 times the draught of the ship, C_m can be interpolated. In the case of a longitudinal approach C_m can be taken as 1.1.

Ship flexibility factor C_s:
Purpose of the ship flexibility factor is to account for elastic deformation of the ship's hull during a collision. The recommended value for C_s is 1.0, and only a small reduction can occur in case of more stiff fenders or large ships, leading to 0.9 < C_s <1.0.

Waterfront attenuation factor C_c:
When a vessel approaches a closed structure, e.g. a quay wall, the water that is located between the ship and the structure is trapped. As a result, a part of the kinetic energy of the vessel is absorbed by the water, resulting in a reduction of the berthing energy that the structure needs to absorb.

Contrary to e.g. quay walls, flexible dolphins can be regarded as open structures. As a consequence, there is no reduction of the berthing energy due to the cushion effect of the water. This leads to a value of 1.0 for the C_c factor for flexible dolphins.

Eccentricity factor C_E:
The eccentricity factor, as the name indicates, accounts for the eccentricity of the mass centre of the ship with respect to the point of first impact, *V* in Figure 2.5. C_E is defined by Equation (2.2) (PIANC, 2002). Figure 2.5 is inserted to visualize the symbols used in this expression.

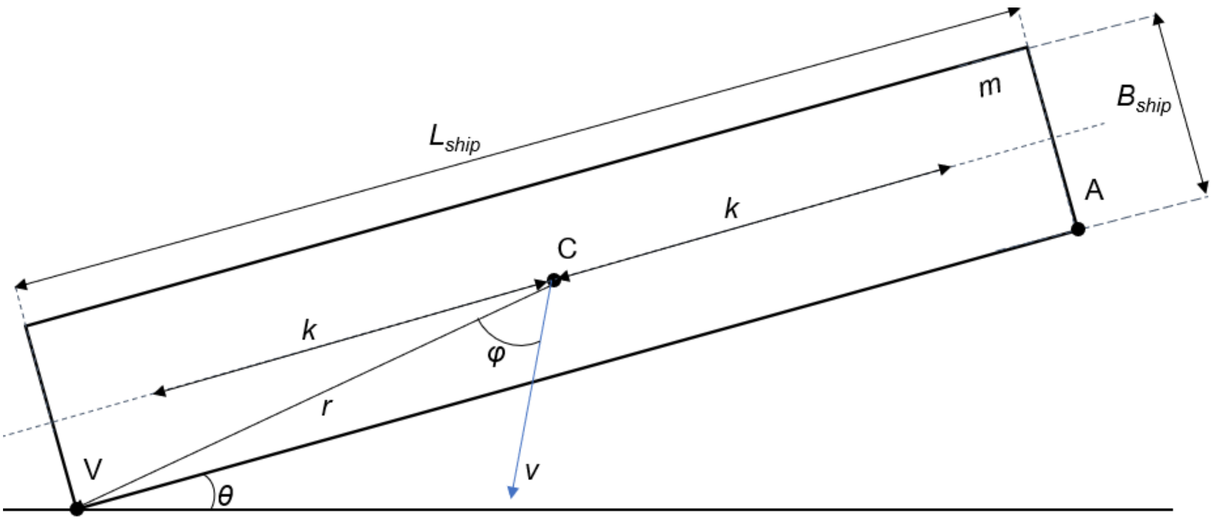


Figure 2.5 Overview of applied variables in energy determination, based on (SBRCURnet, 2018)

$$C_E = \frac{k^2+r^2\cos^2(\phi)}{k^2+r^2} + \frac{\omega r}{v}\frac{2k^2\sin(\phi)}{k^2+r^2} + \frac{\omega^2 r^2}{v^2}\frac{k^2}{k^2+r^2} \text{ [-]} \quad (2.2)$$

where:

k	Radius of gyration of the ship	[m]
r	Distance mass centre ship to contact point	[m]
φ	Angle between velocity vector v and r	[°]
ω	Angular velocity of the ship	[rad/s]
v	Berthing velocity of the mass centre	[m/s]

The radius of gyration is given by Equation (2.3) as:

$$k = L_{ship}(0.19\ C_B + 0.11) \tag{2.3}$$

where:

L_{ship}	Length of the vessel	[m]
C_B	Block coefficient	[-]

Typical values for the block coefficient are in the range of 0.72 – 0.85 for bulk carriers and 0.85 for tankers. Eventually, this leads to an eccentricity factor C_E that varies between approximately 0.4 - 0.6 and 0.6 - 0.8 for quarter and third point[2] berthing respectively.

2.4 Safety philosophy

Various studies have been performed to calibrate Equation (2.1). The dominating factor in the expression is the berthing velocity of the vessel, since this term appears as a square in the energy definition.

The EAU 2012 and PIANC 2002 provide a graph with berthing velocities as a function of vessel size and environmental conditions. This graph is shown in Figure 2.6. It should be noted that the PIANC 2002 use the term “mean design value of berthing velocity”, while the EAU 2012 uses the term “characteristic berthing velocity”. Handling of characteristic and design values is further discussed in section 2.4.1 and section 2.4.2.

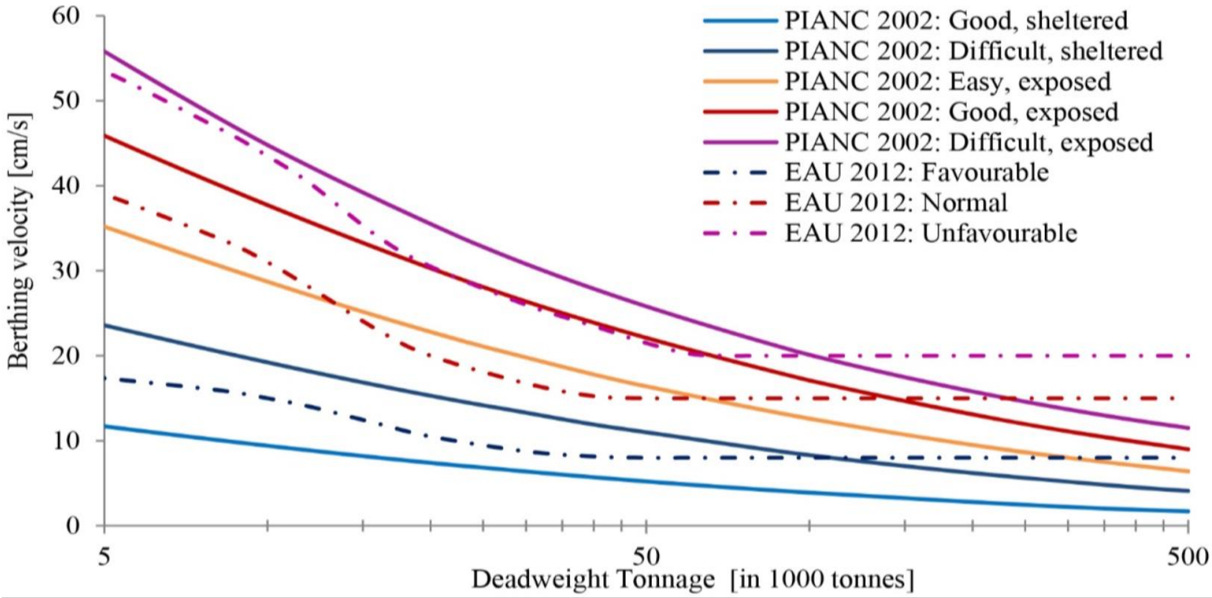


Figure 2.6 Berthing velocities according to the PIANC 2002 and EAU 2012

The difference between a characteristic value and a design value is given in Figure 2.7. A characteristic value of a certain variable, in this case the berthing velocity, is the value that corresponds with a specified exceedance probability (2% in a reference period of 1 year for the Eurocodes) (Jonkman, Steenbergen, Morales-Nápoles, Vrouwenvelder, & Vrijling, 2017). The design value is then obtained by multiplication of the characteristic value with a safety factor.

[2] Quarter and third point berthing refers to the eccentricity of the berthing manoeuvre with respect to the length of the vessel

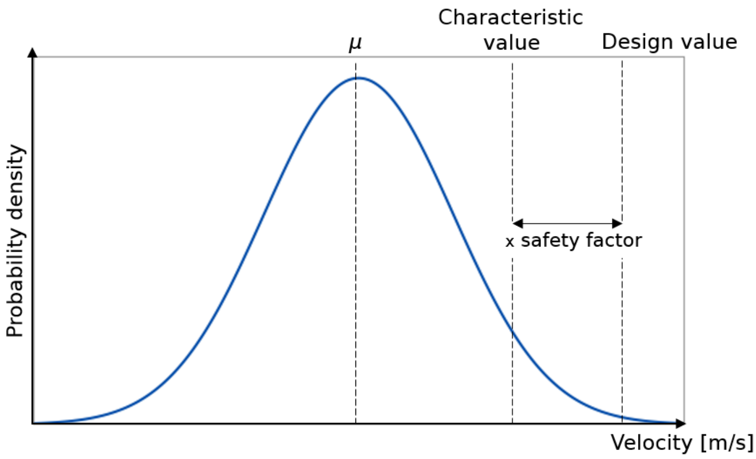


Figure 2.7 Difference characteristic value and design value

2.4.1 Reliability methods

To account for uncertainties in the assessment of loads and material properties, four different levels of reliability methods can be used. Level 0 reliability method implies the usage of deterministic values for every parameter. This means that, in fact, no reliability analysis is performed.

Level I reliability method makes use of semi-probabilistic values that are obtained by applying partial safety factors on characteristic values. Codes predominantly make use of this level of reliability analysis.

Subsequently, the level II reliability analysis can be distinguished. This level uses a probabilistic approach with some approximations, e.g. linearization and translation to normal distributions.

Finally, the third level entails a full probabilistic approach, mostly executed as a Monte Carlo simulation. Elaboration on the second and third level reliability method is omitted, since first level reliability method is used.

Partial safety factors as used in the level I reliability analysis are divided into three different consequence classes, depending on the expected consequences of failing of a structure. Classification of the consequence classes, or equivalently: *reliability classes*, is given in Table 2.2.

Table 2.2 Consequence classes (NEN, 2011)

Consequence Class	**Description**	**Example**
CC3	**High** consequences for loss of human life or economic, social or environ-mental consequences **very great**	Grandstands, public buildings where consequences of failure are high (e.g. a concert hall)
CC2	**Medium** consequences for loss of human life, economic social or environ-mental consequences **considerable**	Residential and office buildings, public buildings where consequences of failure are medium (e.g. an office building)
CC1	**Low** consequences for loss of human life, and economic, social or environ-mental consequences **small or negligible**	Agricultural buildings where people do not normally enter (e.g. storage buildings and greenhouses)

For flexible dolphins, and thus for guide works, the consequence class is equal to the consequence class of the structure that is being protected. It is also possible that the client demands a higher safety level than the minimum required safety level.

2.4.2 Application of level I reliability method in the codes

Using a level I reliability method analogously to the Eurocode gives rise to two ways of applying safety factors on the berthing velocity or energy. (Roubos, Groenewegen, Ollero, Hein, & Wal, 2018)

A first option is to determine a characteristic berthing velocity v_k. Multiplication of v_k with the partial safety factor for the berthing velocity γ_v leads to a design berthing velocity v_d. Now, the design berthing energy $E_{kin;d}$ and the corresponding impact load F_v can be determined. Finally, the impact load needs to be multiplied with γ_{sd} to find the design berthing impact load F_s. γ_{sd} is introduced to account for the modelling uncertainty of the load.

Second option is to make use of the characteristic berthing velocity to obtain the characteristic berthing energy $E_{kin;k}$. Then, based on the characteristic berthing energy, the characteristic berthing impact load F_k can be determined. Finally, to determine the design berthing impact load F_Q, F_k needs to be multiplied with the partial safety factor for berthing impact load γ_Q.

From the two design loads F_s and F_Q the designer has to choose the force that has the biggest influence on the dolphin. This force is denoted as F_d.

Contrary to the Eurocode, the PIANC and EAU make use of the abnormal berthing factor C_{ab} rather than using the partial safety factor on the characteristic berthing velocity; the characteristic berthing energy is computed using the characteristic berthing velocity. Subsequently, the design berthing energy is computed by multiplication of C_{ab} with the characteristic berthing energy. Therefore, C_{ab} is proportional to γ_v^2.

Note that brush collisions are regarded as an exceptional berthing manoeuvres and this classification is taken into account in the abnormal berthing factor. Head-on collisions, however, fall outside the definition of an exceptional berthing manoeuvre and are classified as an accidental limit state. The scope of this thesis is limited to brush collisions, and therefore, head-on collisions fall outside the scope of this thesis.

Table 2.3 gives the safety factors as prescribed by the Eurocode, while Table 2.4 provides the recommended values for the abnormal berthing factor C_{ab}. The last two columns of Table 2.4 are empty, since the OCDI[3] and the Eurocode EN 1990 do not make use of the abnormal berthing factor.

Table 2.3 Partial safety factors for the berthing energy (SBRCURnet, 2018)

Navigation conditions	Pilot assistance	Symbol	Reliability class EN 1990		
			RC1	RC2	RC3
Sheltered and monitored[1)]	Yes	γ_v	1.15	1.20	1.25
		C_{ab}	1.35	1.45	1.55
Sheltered	Yes	γ_v	1.20	1.25	1.30
		C_{ab}	1.45	1.55	1.70
Exposed[2)]	Yes	γ_v	1.30	1.35	1.40
		C_{ab}	1.70	1.80	2.00

1) Pilots are aware of the allowable berthing velocity and use berthing aid systems, such as portable pilot units.
2) Strong tidal currents.

Table 2.4 C_{ab} factors as prescribed in different codes (SBRCURnet, 2018)

Ship type	Size	PIANC (2002)	EAU (2012)	BS 6349-4 (2014)	ROM (1990)	OCDI (2009)	EN 1990 (2011)
Tankers	Largest–Smallest	1.25–2.00	1.25–2.00[1)]	-	2.00	-	-
Bulkers	Largest–smallest	1.25–2.00	1.25–2.00[1)]	-	2.00	-	-
Container	Largest–smallest	1.50–2.00	1.50–2.00[1)]	-	2.00	-	-
General cargo	-	1.75	1.75[1]	1.50[2)]	2.00	-	-
RoRo, ferries	-	≥ 2.00	≥ 2.00[1]	-	2.00	-	-
Tugs, workboats	-	2.00	2.00[1]	-	2.00	-	-
LNG, LPG	-	-	-	2.00	2.00	-	-
Island berth	-	-	-	2.00	2.00	-	-

1) Based on PIANC 2002 [3.21].
2) Continuous quay handling conventional cargo vessels.

3 OCDI is the Overseas Coastal Area Development Institute of Japan that aims to develop ports in developing countries (OCDI, 2019)

3. Vessel collisions versus berthing manoeuvres

Starting with the definition of the different rotational and translational movements of a vessel, this chapter discusses two impact loadings of vessels on flexible dolphins, being: ship collisions and berthing manoeuvres. First, a description of these two processes is given and ultimately, the term 'impact' as used within this report is explained.

3.1 Definitions for vessel movements

In order to facilitate the description of vessel collisions and berthing manoeuvres, this section provides the definitions for the movement of a vessel. The bow and stern of a vessel are indicated in Figure 3.1.

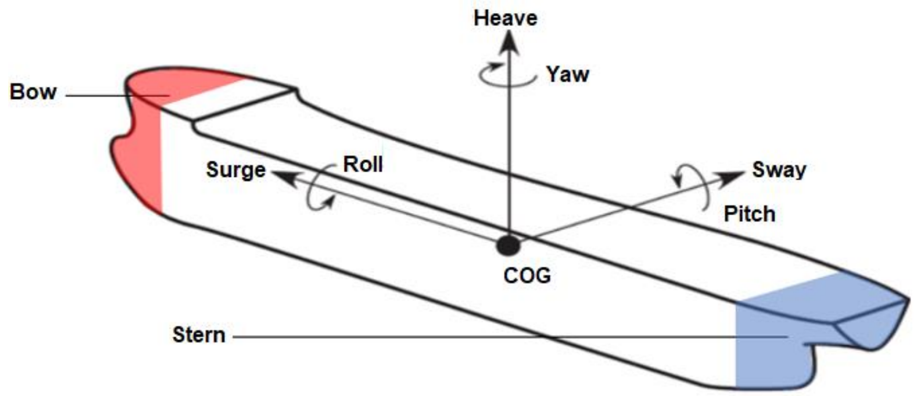


Figure 3.1 Translational and rotational vessel movements (TheNavalArch, sd)

Generally, a vessel has six degrees of freedom; three translational movements and three rotational movements. The three translational movements are surge, sway and heave for a movement in the longitudinal, transverse and vertical direction respectively.

Rotation around the longitudinal axis is defined as *roll*, while rotation around the transverse axis is defined as *pitch*. Finally, rotating around the vertical axis is called *yawing*.

Lastly, the bow and stern are terms for the front and the back of the vessel respectively. (Wärtsilä Encyclopdiea of Marine Technology, sd)

3.2 Berthing manoeuvres

The first impact loading that is discussed within this chapter is berthing of a vessel. The aim of a berthing manoeuvre is to obtain a moored vessel, i.e. a vessel that has come to a standstill. Berthing is particularly important in harbours in order to facilitate the transhipment of goods. Also near locks berthing occurs, in case vessels have to wait before they can enter the lock. Once the berthing manoeuvre is finished, the term moored vessel is used.

In Figure 3.2 a berthing manoeuvre in the waiting area near a lock is schematically shown. Initially, the vessel contains a translational velocity that is dominated by the sway direction. The bow of the vessel starts to make contact with the structure, and the vessel in Figure 3.2 starts yawing in the counterclockwise direction. As a result of the negligible velocity in longitudinal direction the contact point during first impact will remain more or less stationary.

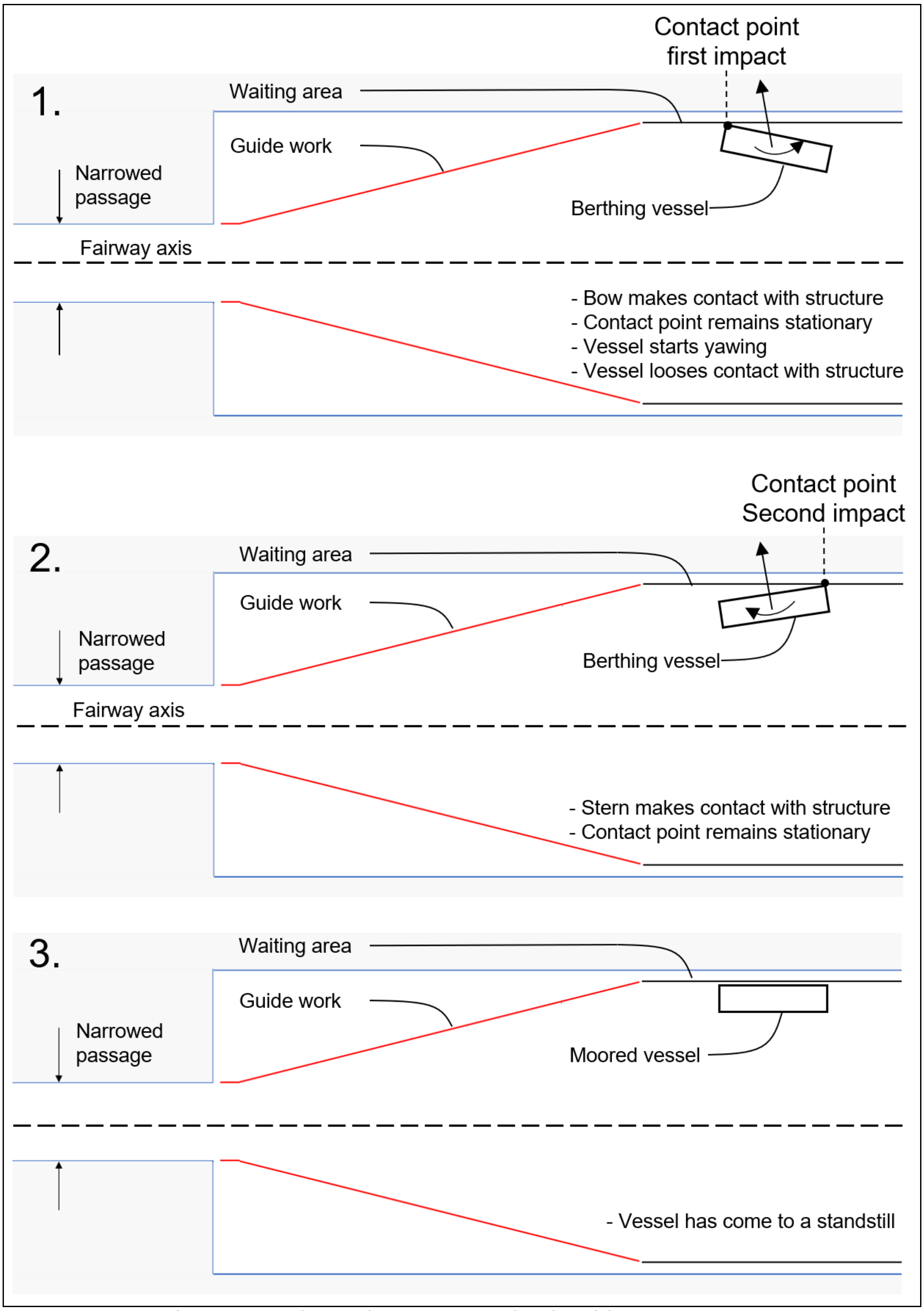


Figure 3.2 Schematic representation berthing manoeuvre

Since the contact between the structure and the vessel is eccentric with respect to the mass centre of the vessel, the vessel starts yawing. It is then possible that the stern also makes contact with the structure and a so-called *second berthing impact* takes place. Due to this second impact the yawing movement is reverted and eventually the vessel comes to a standstill.

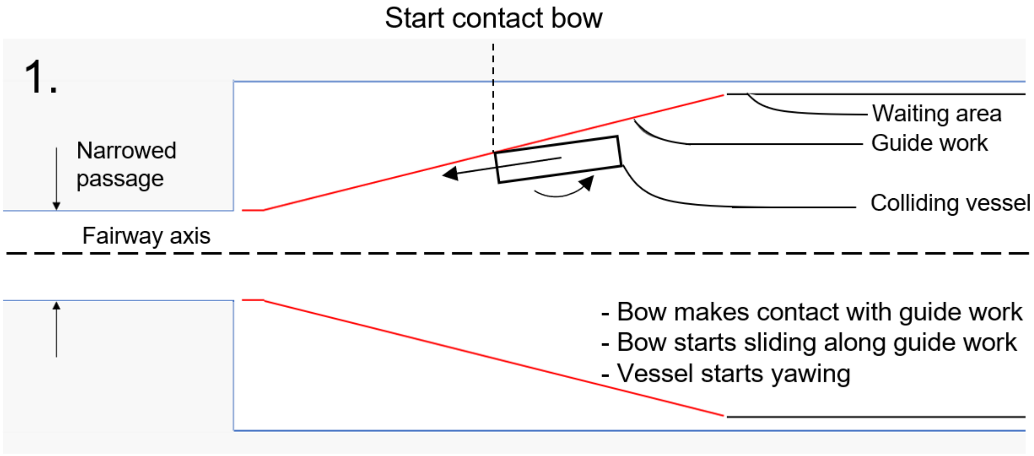
3.3 Vessel collision

Contrary to the intended impact caused by berthing manoeuvres, it is also possible that an impact takes place accidentally. The term ‘collision’ is used for this unintended contact. Collisions can be split into two categories: brush collisions and head-on collisions.

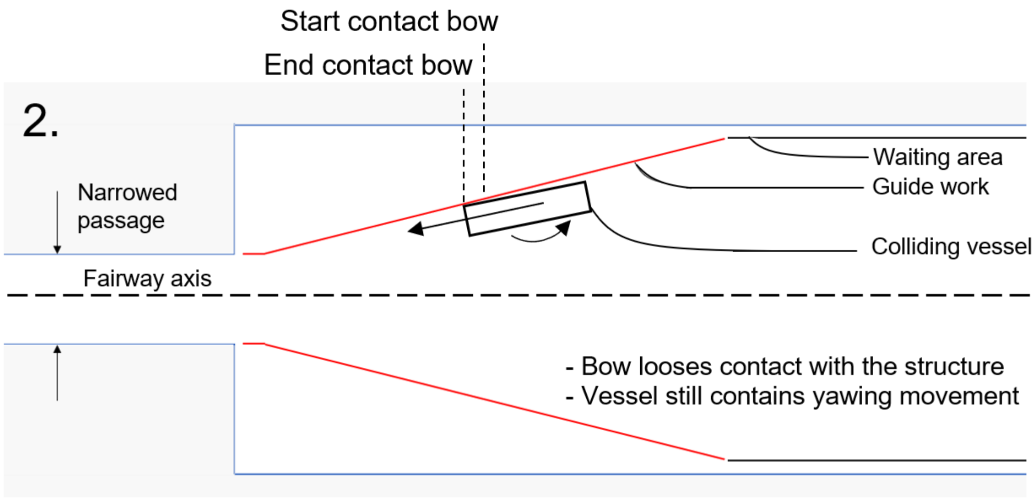
3.3.1 Brush collision

Figure 3.3 gives a schematic representation of a brush collision between an inland navigation vessel and a guide work near a lock entrance, including a textual description.

Initially the vessel contains a translational velocity that is dominated by the velocity in surge direction.

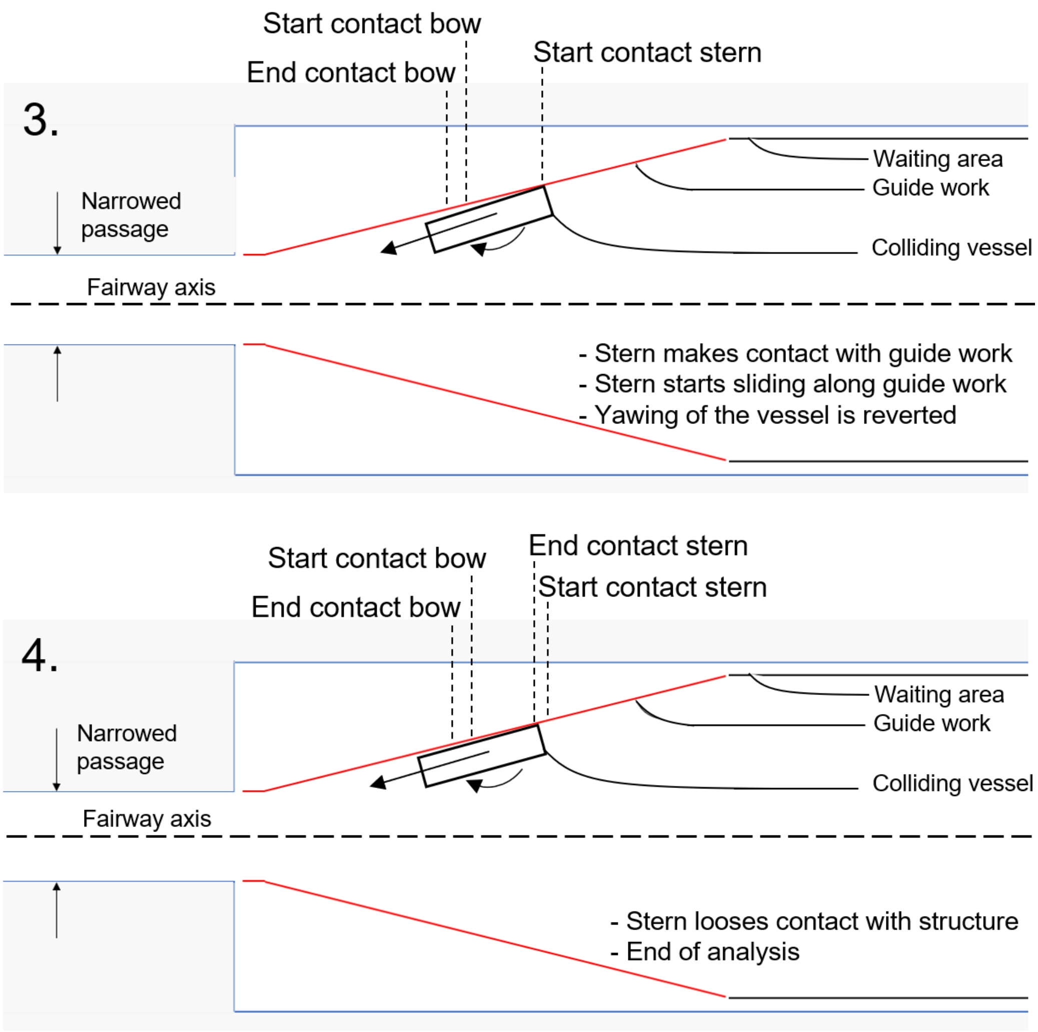


When the bow of the vessel makes contact with the guide work, a similar process as found during a berthing process is observed; a contact force that is eccentric with respect to the mass centre of the vessel, is working on the colliding vessel. As a result, the ship in this figure starts yawing in the counter-clockwise direction and after several seconds the first contact has ended.



A certain amount of time later, it is possible that the stern makes contact with the guide work as well, analogously to a berthing manoeuvre.

Difference, however, between a brush collision and a berthing manoeuvre is that the contact point is not stationary during the process during a brush collision. The contact point of the vessel is sliding along the guide work and therefore the structure needs to have a certain length in order to facilitate this brushing of the vessel.



On top of that, it needs to be mentioned that in case of brush collisions the initial angle of contact between the vessel and structure is limited (< ca. 5°), while for berthing manoeuvres larger angles are not rare.

Figure 3.3 Schematic representation brush collision

3.3.2 Head-on collision

Apart from brush collisions, also head-on collisions exist. In Figure 3.4 an example of a head-on collision of a vessel with a lock door is shown. During a head-on collision of a vessel, the bow of a vessel hits the structure, a lock door in case of Figure 3.4, in an approximately perpendicular way.

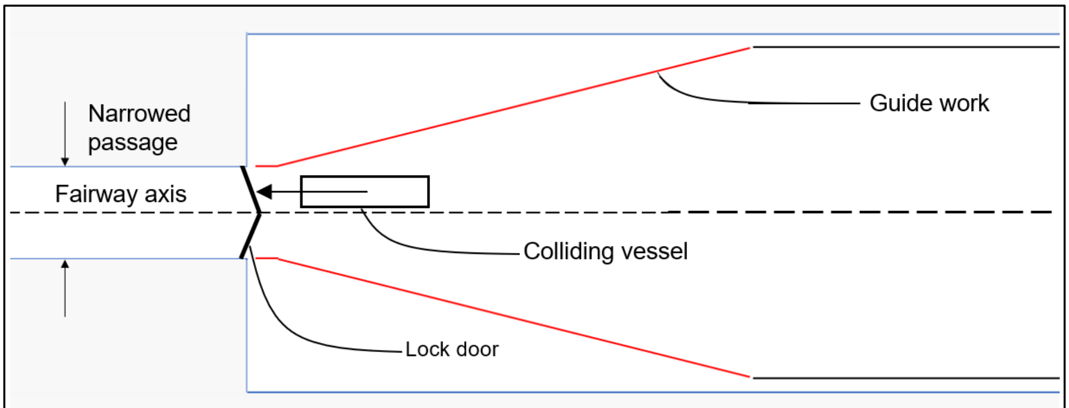


Figure 3.4 Schematic representation head-on collision

Head-on collisions between a vessel and a lock door are thoroughly investigated in the thesis of Edmondson (2017). In the named report, the collision between a vessel and a structure is schematised as a mass-spring system, similar to literature on collisions between vessels and guide works. In addition to that, Jansen (2019) found that reduction of the contact angle is beneficial for the magnitude of the maximum occurring contact force. Head-on collisions, however, fall outside the scope of this thesis, since this research limits itself to brush collisions.

3.4 Definition of ‘impact‘ within this thesis

Within this thesis, brush collisions between inland navigation vessels and guide works are examined. Consequently, ‘impact’ in the upcoming part of this report refers to impact due to brush-collisions. Additionally, ‘first impact’ refers to the period of contact of the bow with the guide work, while ‘second impact’, if applicable, refers to the period of contact of the stern with the structure.

Normally, flexible dolphins are examined by placing the energy-related force on the structure. Different impact locations are investigated in a static manner. The effect of the first impact on the movement of the ship, however, is not fully considered during normal design practice.

Measurements have been performed at different locations in Japan for berthing speeds of very large crude carriers (VLCCs). Not only the berthing velocity of the first impact, but also the berthing velocity of the second berthing impact was monitored. It followed that there were no significant differences between the berthing velocity of the first impact and the second impact for the observed vessels. (Bratteland, 1988).

On top of that, in the port of Rotterdam measurements for the berthing velocity of seagoing vessels have been performed. It followed that for the first impact the translation of the ship was dominant, while for the second impact the yawing of the vessel was of most importance. Also, the second berthing impact could be governing over the first impact (Roubos, Groenewegen, & Peters, 2016).

Please note that the performed measurements are related to berthing velocities, and not to velocities of a brush collision. Due to the similar nature of these two phenomena, however, it is wondered whether the second impact in case of brush collisions is governing the first impact as well, analogously to berthing manoeuvres.

Despite the measurements for the second berthing impact, a proper description of the vessel’s movement to support the performed measurements with a scientific derivation is still lacking. The derivation of the movement of the vessel is elaborated in appendix B.

4

Description model for brush collisions in the time domain

This chapter describes the modelling in Python of brush collisions between inland navigation vessels and guide works in the time domain. First, the modelling of the pile soil interaction is presented, followed by the modelling of the movement of the colliding vessel. Finally, the possibilities and limitations of this model are mentioned.

4.1 Modelling of the pile soil interaction

Based on the studied literature, see appendix A, the laterally loaded vertical piles are modelled based on the modified Blum's method. The vertical piles are supported by a linear translational and linear rotational spring that represent the stiffness of the soil, and the length *L* in Figure 4.1 is the summation of a constant length above bed level and a certain depth *t* underneath bed level. The stiffnesses of both springs, as well as depth *t*, is dependent on the horizontal load *F* working on the pile, see also appendix A.

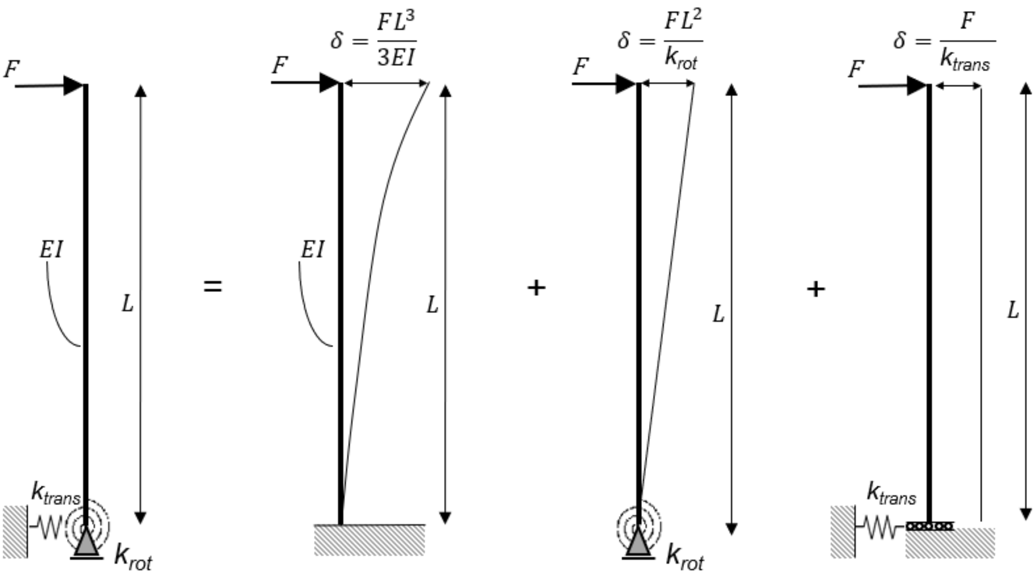


Figure 4.1 Modelling guide work vertical plane

The deflection at the location where the load is applied can be determined by using the superposition principle. This is expressed in Equation (4.1).

$$\delta_{tot} = \sum_{i=1}^{3} \delta_i = \frac{F\,L^3}{3\,EI} + \frac{FL^2}{k_{rot}} + \frac{F}{k_{trans}} \quad (4.1)$$

where:

δ_{tot}	Total displacement at the top of the pile	[m]
F	Lateral load at the top of the pile	[N]
L	Length of the pile	[m]
EI	Bending stiffness of the vertical pile	[Nm^2]
k_{rot}	Rotational spring stiffness	[Nm/rad]
k_{trans}	translational spring stiffness	[N/m]

Equation (4.1) can be rewritten to $\delta_{tot}= F/\ k_{tot}$ The system can be seen as a serial system. Therefore, the total spring stiffness for the top point k_{tot} can be expressed as:

$$\frac{1}{k_{tot}} = \frac{1}{k_{pile}} + \frac{L^2}{k_{rot}} + \frac{1}{k_{trans}} \tag{4.2}$$

where:

k_{tot}	Total representative stiffness	[N/m]
k_{pile}	Stiffness of the vertical pile = $\frac{3EI}{L^3}$	[N/m]

By representing the vertical piles as a spring, it becomes possible to describe the collision process in the horizontal plane only. The guide work can then be represented as shown in Figure 4.2.

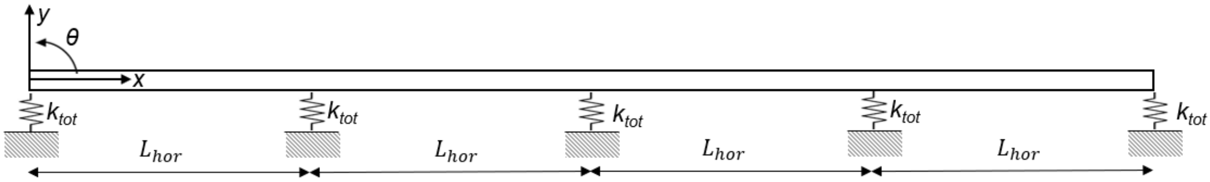


Figure 4.2 Modelling of the guide work in the horizontal plane

When a load is applied on the system, the deflections and the force distribution within the system can be determined either analytically or numerically. Complicating factor, however, is the fact that the spring stiffness is dependent on the force working on it. An option to overcome this problem is to solve the system iteratively. Since an analysis in the time domain is concerned, a second possibility is to use the force in the spring of the previous time step. In the Python model, for the latter solution is chosen. It is important to realise that the modelling of the brush collision is thus nonlinear, while within every timestep a linear elastic calculation is performed.

Note that the force in the spring is dependent on the location of contact and the bending stiffness of the horizontal berthing beam. As a result, the modelling depth of the springs, as well as their stiffnesses, differs per spring, resulting in a different value for k_{tot} for each pile.

4.2 Modelling of the movement of the colliding vessel

This section describes how the modelling of the movement of the colliding vessel is carried out. To explain how the movement is modelled in the Python model, first the geometrical definitions are described. After the introduction of the geometrical definitions, the movement is described in section 4.2.2.

4.2.1 Geometrical definitions

This subsection gives an overview of the different symbols and definitions that are used in describing the horizontal movement of the vessel. First of all, it can be mentioned that the x-axis is aligned with the longitudinal direction of the structure in undeformed condition. The y-axis is perpendicular to the x-axis, as indicated in Figure 4.3, leading to a positive rotation θ for the counterclockwise direction.

The geometry of the ship is simplified to a rigid rectangle with dimensions B_{ship} and L_{ship} for the width and length of the ship respectively. The mass of the ship is denoted with *m* and the centre of gravity is denoted with *C*. Other notable points of the ship are the location where the bow or the stern can make contact with the structure; *V* and *A* are used for these positions respectively.

The x- and y- position of point V, C and A are indicated with the letter of the axis, combined with a subscript that contains the name of the point. The derivative with respect to time, ∂/∂t, is commonly denoted by a ‘over-dot’. Since velocity is defined as the first derivative of the displacement with respect to time, and acceleration is defined as the second derivative of the displacement with respect to time, this leads to e.g. $\dot{x}_C$ and $\ddot{x}_C$ as the notation for the mass centre’s velocity and acceleration in x-direction respectively.

Furthermore, the distances *a* and *b*, being the horizontal and vertical distances from the centre of gravity to contact point V respectively, can be distinguished. Additionally, distances *c* and *d* are introduced as the vertical and horizontal distance respectively, between point A and point C. Figure 4.3 is inserted to present these definitions in a graphic manner.

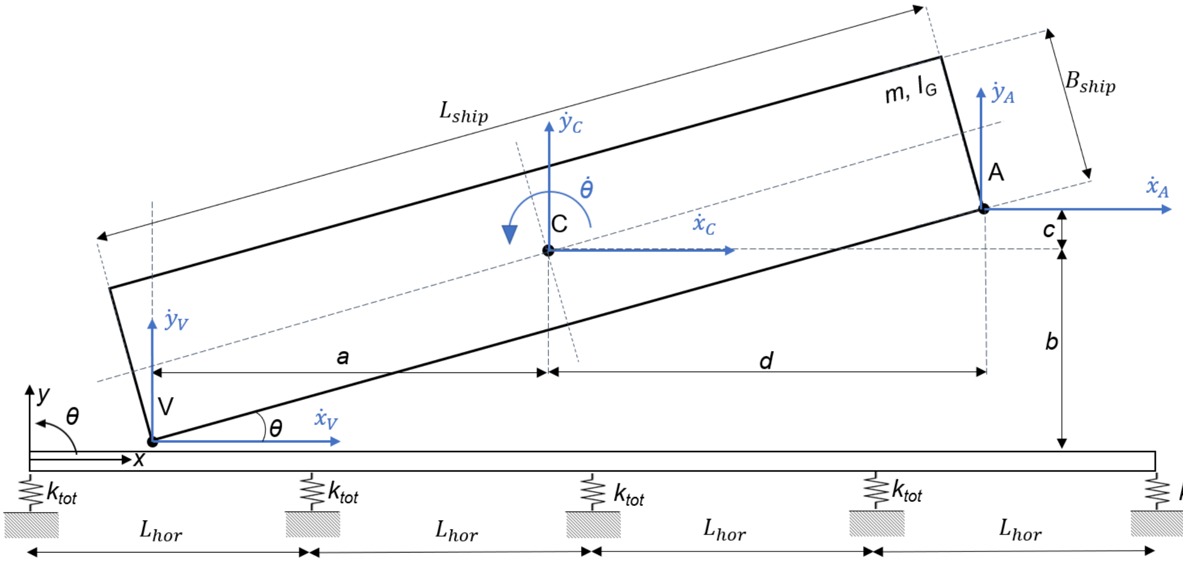


Figure 4.3 Definitions of a moving vessel in the horizontal plane

4.2.2 Vessel's velocities and displacements in the model

In the contact process several distinct moments in time can be observed. The analysis starts at $t=t_0$ at which the very first contact takes place between the bow of the vessel and the guide work. Then, during an infinitesimal small time interval that ends at $t=t_{0'}$, the structure is accelerated and the ship's bow is decelerated. During this time interval an exchange of momentum takes place, resulting in a loss of translational velocity of the vessel, and a gain of a yawing motion (Vrijer, 1978). Elaboration of the exchange of momentum is given in appendix B.

After $t_{0'}$ the collision is discretized in timesteps of 0.1 seconds. During the first time step the bow is moving in the negative y-direction, leading to a deformation of the guide work; a prescribed deformation is imposed on the structure.

As a result of the prescribed deformation, a reaction force by the structure is exerted on the vessel at the location of contact. In the following timestep, the angular and translational velocity and displacement are affected by the reaction force. The new y-position of the bow is now imposed as prescribed deformation, at the new x-position of the bow.

This procedure repeats itself until the contact between the bow and the guide work has ended. After the first contact has ended, the vessel still contains an angular velocity as well as a translational velocity. The position of the stern is updated every timestep as well. As soon as the stern makes contact with the structure, a procedure similar to the first impact is used to describe the second impact; the y-coordinate of the stern is imposed on the structure as prescribed deformation on the location of the x-coordinate of the stern.

Apart from the force that is exerted by the guide work, the water surrounding the vessel also plays an important role. The role of the water is twofold. First of all, there is a virtual increase in the translational mass of the vessel and the mass moment of inertia of the vessel. This increase is due to the water that needs to be pushed away by the vessel to enable the ship to move in a certain direction. The added mass in x- and y-direction is dependent on the angular position of the vessel, and therefore, varies during the analysis.

On top of that, the ship also experiences drag due to the water. The drag damps both the translational and the rotational movement of the vessel. The magnitude of the drag force is dependent on the velocity of the vessel and will therefore change during the contact. Description of the added mass and the damping due to the water is given in appendix B.3 and appendix B.4 respectively.

The workflow for the modelling in the Python model of the brush collision between an inland navigation vessel and a guide work is given in Figure 4.4. In total 900 time-steps of 0.1 seconds are used during every performed analysis, leading to a total analysis time of 90 seconds.

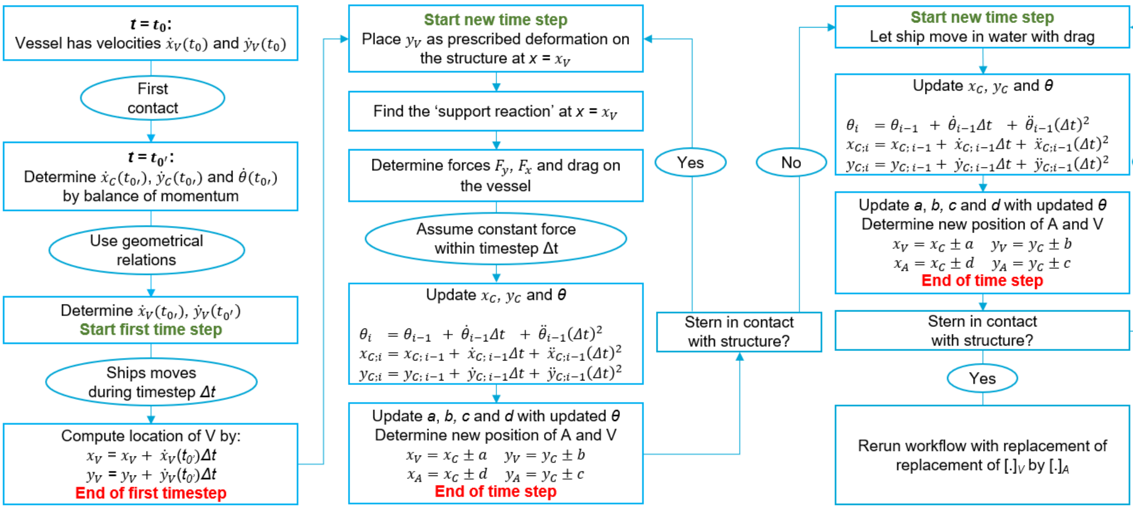


Figure 4.4 Workflow analysis in the time domain

4.3 Capabilities of the created model

The newly created Python model enables the qualitative and quantitative assessment of the influence of the input parameters on the magnitude of the contact force, the duration of a brush collision and the length over which the collision takes place. An overview of the examined parameters is described in the introduction, see also the list below:

- Mass of the colliding vessel
- Length of the colliding vessel
- Draught of the colliding vessel
- Outside diameter of the horizontal berthing beam
- Outside diameter of the vertical piles
- Length of the vertical piles above bed level
- Location of the first contact
- The initial angle of contact
- Initial velocity of the vessel

Since nonlinearities with respect to the soil behaviour are covered by the explicit time-stepping scheme, within every time-step a set of linear equations needs to be solved. Due to this linearity, the created tool is able to run an analysis of 900 time-steps within 0.4 seconds (<0.001 second per time-step), which makes the model a very fast tool to simulate brush collisions.

By using the created tool it is possible to obtain the occurring contact force in each time-step, as well as the displacement and the translational and rotational velocities of the vessel. On top of that, the tool provides insight in the distribution of energy in the system during a brush collision. As a result, the amount of potential energy in the structure and the soil can be found for every time-step, making it possible to adequately design guide works in future projects.

Although the model is originally intended for the simulation of brush collisions between inland navigation vessels and guide works, small adaptations of the model in the future enable the simulation of berthing manoeuvres as well.

4.4 Limitations of the created model

Apart from the capabilities, the Python model also has several limitations with respect to the modelling of brush collisions between guide works and vessels. These limitations are discussed within this subsection.

Most of the limitations of the created tool are related to the geometrical freedom of the guide work. Origin of the restricted freedom is the parametric implementation of geometrical in the Python script within the given time span to create this tool and conduct the research. Restrictions with respect to the geometry are given in the succeeding list.

- The number of vertical piles is fixed
- The vertical piles are assumed to be prismatic, while in practice the wall thickness of the vertical piles varies over its length
- All vertical piles are identical, which means that no difference is made between *end piles* and *middle piles*
- In case of a second impact, the contact point of the vessel should be at the stern and not at a random location along the side of the vessel

Despite the limitations mentioned above, it is still possible to use the created tool to answer the research question and the corresponding sub-questions.

Furthermore, also few limitations are present that are not primarily related to the geometry of a guide work. First of all, the role of human interference is not considered during the analyses; a natural reaction of the captain of a vessel might be to steer the vessel additionally in case of a collision.

Secondly, the assumption of a rectangular shape of the colliding vessel might overestimate the influence of the drag by the water. Also, damping due vibrations of the structure and the soil, which might reduce the effect of an impact, are neglected. It is to be noted, however, that this damping is expected to be small, given the limited transversal velocities in combination with the duration of an impact.

4.5 Translation created model in Python to SCIA Engineer

The model that is created in Python to simulate brush collisions between inland navigation vessels and guide works, as described in this chapter, is translated to the finite element method (FEM) software *SCIA Engineer*, or *SCIA* in short.

The main reason for this translation to SCIA Engineer is to overcome the limitations of the Python model. On top of that, due to the automatic post-processing environment in the FEM software, it is possible to generate e.g. the principle stresses as output. In the case of steel cross-sections, the principle stresses can directly be checked by the von Mises criterion, resulting an acceleration of the design process. Appendix E describes the reason why this translation is made in more detail and provides a description of this tool. Furthermore, the possibilities and limitations of this translation are given in the appendix, as well as a verification of this tool involving SCIA Engineer.

In this section, a short description of the translation to SCIA Engineer is given, as well as the discussion and conclusions regarding the performed translation.

4.5.1 Short description of the translation to SCIA Engineer

The used procedure for the Python model as depicted in Figure 4.4, is the procedure that is also applied in the modelling of brush collisions in SCIA Engineer. However, a difference in mechanical point of view is present in the SCIA Engineer environment. In the Python model, the horizontal berthing beam (indicated with red in Figure 4.5) is supported by springs representing the soil stiffness and the bending stiffness of the vertical piles (indicated with blue in Figure 4.5). In SCIA Engineer, the vertical piles are not replaced by springs, but are present as 1D members in the model. Consequently, the nonlinear springs representing the stiffness of the soil are placed at the bottom of these vertical piles, and the rigid connections between the horizontal beam and the vertical piles are taken into account in the SCIA Engineer model, contrary to the Python model.

As a result of the rigid connection between the vertical piles and the horizontal berthing beam, torsional effects are taken into account, influencing the force distribution in the structure. In Figure 4.5 the mechanical differences between the Python model and the model in SCIA Engineer are presented graphically.

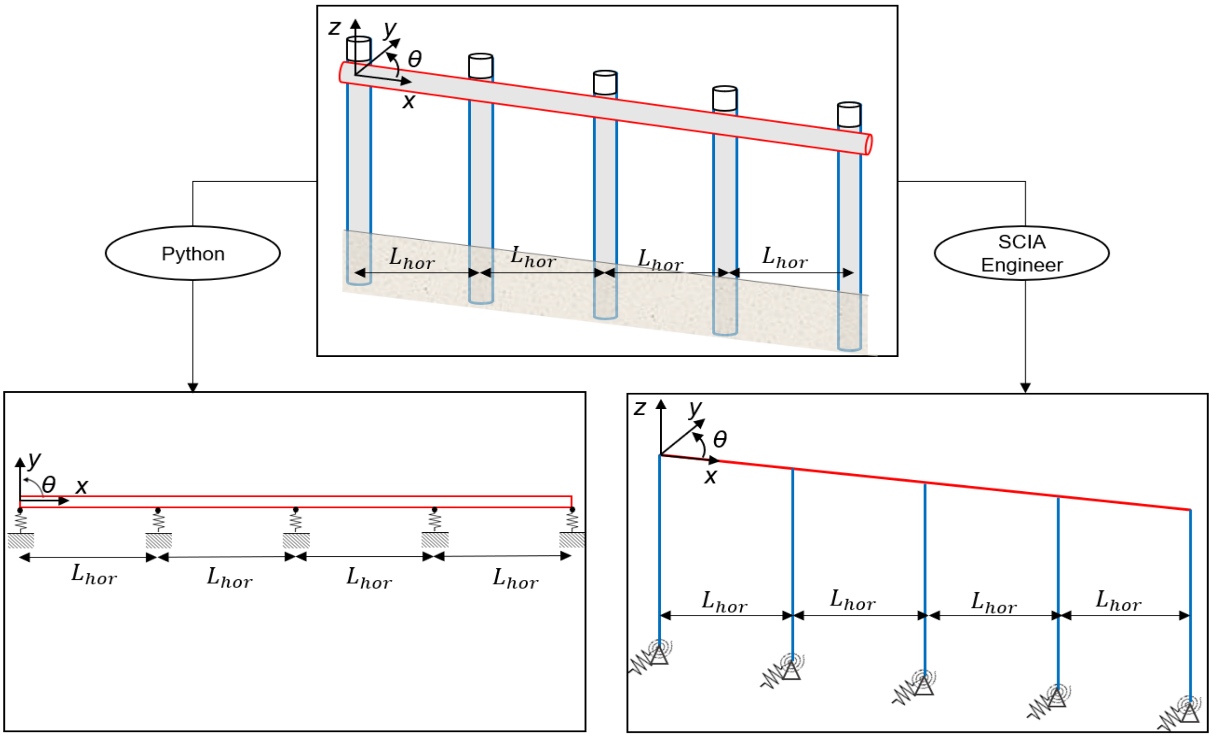


Figure 4.5 Modelling in SCIA Engineer versus modelling in Python

4.5.2 Discussion regarding the translation to SCIA Engineer

From the performed analysis in appendix E, it follows that the results of the model involving SCIA Engineer are in line with the results obtained using the Python model. In chapter 5 a verification of the Python model is performed and the results are assessed to be valid.

The model in SCIA Engineer shows a slightly stiffer behaviour than the Python model. This increased stiffness is probably due to the rigid connections between the horizontal berthing beam and the vertical piles, resulting in the incorporation of torsional effects on the force distribution. Due to the stiffer response the maximum occurring contact force is circa 15% higher and the duration of the impact is approximately 5% shorter.

Furthermore, it is found that the behaviour of the displacements and velocities in the time domain of both models are in accordance with each other.

A significant disadvantage of the translation to SCIA Engineer is the duration of the analysis. In the Python model 900 timesteps of 0.1 seconds are performed in less than 1 second, while the model using SCIA Engineering requires 30 seconds *per timestep* resulting in a total analysis time of 7.5 hours.

Given the automatic post-processing environment of the software, it was expected that the design process of guide works could be accelerated. In its current form, however, the tool involving SCIA is not suitable to use as a design tool, due to its computational time.

Despite the limitations mentioned in the previous paragraph, it is expected that the translation to SCIA can be made suitable for implementation in the future. A possibility to reduce the computational time is to make use of larger time steps. Another option to limit the computational efforts is to 'skip' the linear elastic analysis in SCIA and only update the position of the vessel, in case there is no contact between the vessel and the structure.

Additionally, a proper description of this tool needs to be created in case the tool with SCIA is going to be implemented in the design practice, in order to avoid erroneous input.

4.5.3 Conclusions and recommendations about the translation to SCIA Engineer

To finish the section regarding the translation of the model in Python to SCIA, it can be concluded that it is possible to translate the model to SCIA Engineer, but there are still some practicalities to overcome. ‘Tricks’ as described in the previous subsection are to be implemented to reduce the computation time of the model, and a proper user manual is necessary to enable the usage of this method in future projects.

As long as the practicalities are not overcome yet, it is recommended to use the model that is created in Python. From the Python model, the maximum occurring contact force can be obtained, which can then be treated as static design force for the guide work.

Once the static design contact force is known, it *does* become possible to use SCIA Engineer or other FEM software to verify the design that is proposed, by making use of the software program in the conventional way.

5. Verification of the created model

Before the results for the different parameters are presented, a verification of the Python model is made. For an arbitrary initial condition and vessel, the energy balance of the system is presented in section 5.2. Subsequently, the movement of the vessel during a brush collision is analysed in section 5.3. Finally, in section 5.4 the link with the design guidelines of the EAU and the PIANC is made. Within this chapter the results of the parameters in Table 5.1 are presented. Note that 0.8 at *first contact* means that the first contact takes place on the positive x-axis at 0.8 times the centre to centre distance of the vertical piles.

Table 5.1 Input parameters verification

Variable	Value	Unit		Variable	Value	Unit
m	1E7	kg		D_{hor}	1.82	m
$\theta(t=0)$	5.0	°		t_{hor}	0.013	m
$\dot{x}_c(t=0)$	-1.5	m/s		D_{ver}	0.9	m
L_{ship}	100.3	m		t_{ver}	0.04	m
B_{ship}	22.8	m		$L_{ver;o}$	17+t	m
First contact	0.8	[-]				

The parameters related to the geometry of the guide work are presented in Figure 5.1, while the input parameters related to the dimensions of the vessel and the initial conditions are presented in Figure 4.3.

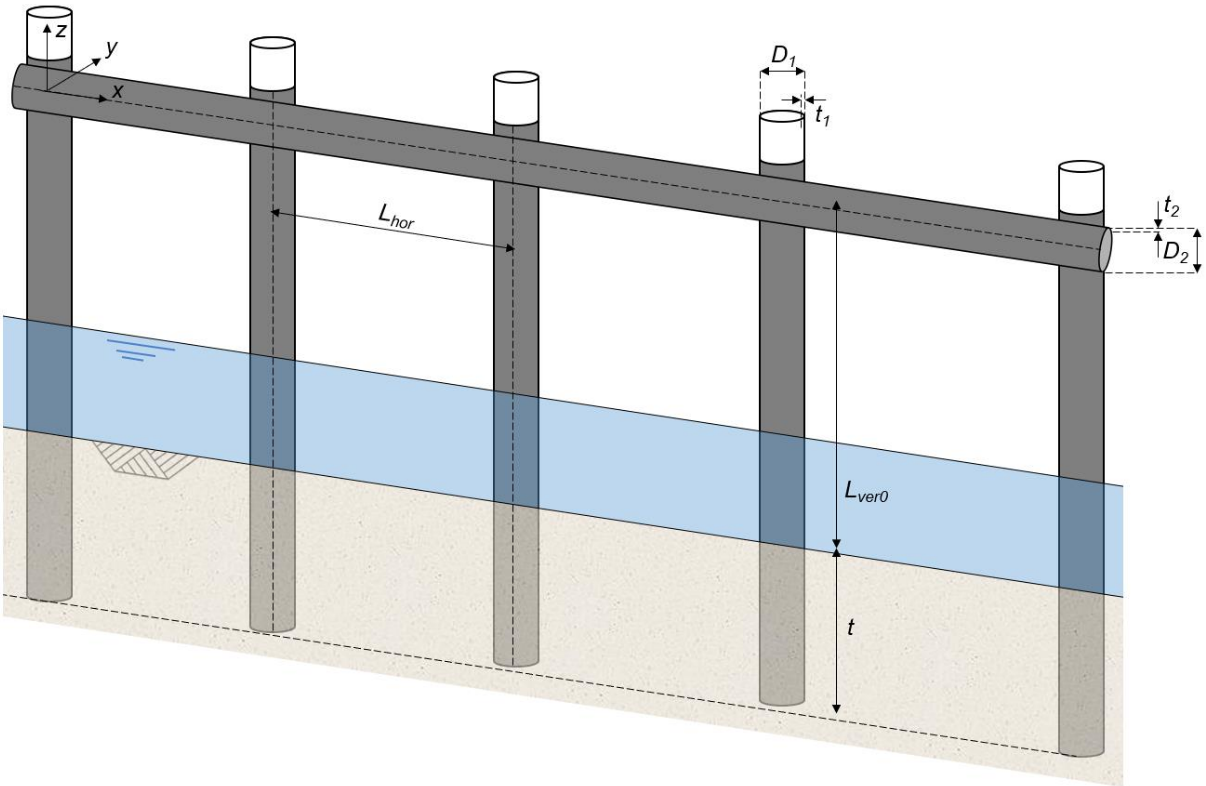


Figure 5.1 Input parameters geometry guide work

5.1 Contact force and deformation contact point

In Figure 5.2 the contact force between the vessel and the guide work is given in the time domain; the black line gives the contact force in the y-direction, while the red line indicates the friction force in x-direction between the vessel and the guide work.

Six distinct moments in time are indicated in Figure 5.2 to Figure 5.4. At $t=t_{0'}$ the exchange of momentum between the guide work and the vessel has just ended and is the starting point of the analysis. Then, at $t=t_1$ the contact force of the first impact reaches its maximum and the vessel is not in contact with the structure anymore at $t=t_2$.

Subsequently, at $t=t_3$ the stern of the vessel starts to make contact with the dolphin. The time at which the maximum force during the second impact is reached, is denoted with $t=t_4$. Finally, at $t=t_5$ the contact of the stern with the guide work has ended again.

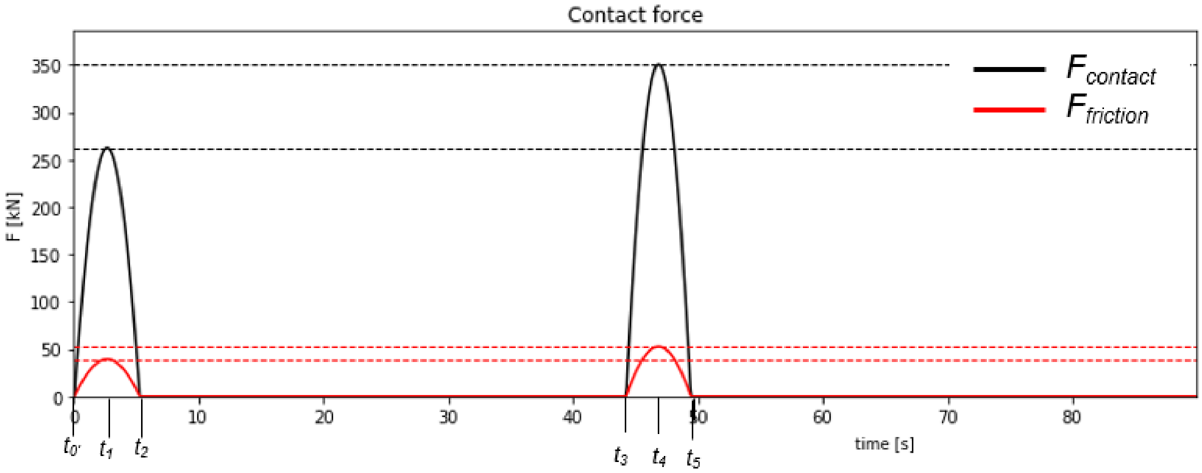


Figure 5.2 Force behaviour in model verification

Figure 5.3 presents the deformation of the contact point in y-direction. It can be seen from Figure 5.2 and Figure 5.3 that, as soon as the deformation of the contact point starts to increase, also the contact force is increasing, which is in accordance with the expectations. On top of that, the ratio between the maximum contact force during the second impact and the first impact is comparable to the ratio between the maximum deformation of the contact point during the second and the first impact.

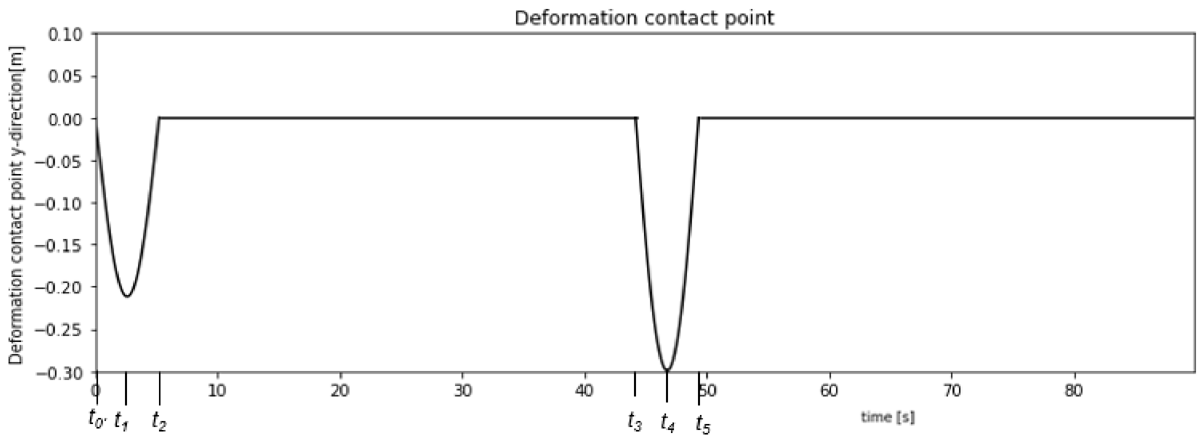


Figure 5.3 Deformation contact point y-direction in model verification

5.2 Verification by means of the energy balance

As a means of verification for the analysis that is performed with the Python model, an energy balance is constructed, see Figure 5.4. Since energy is a preserved quantity, the total energy in the system should be constant. Any deviations from this total energy may imply inconsistencies or errors in the model.

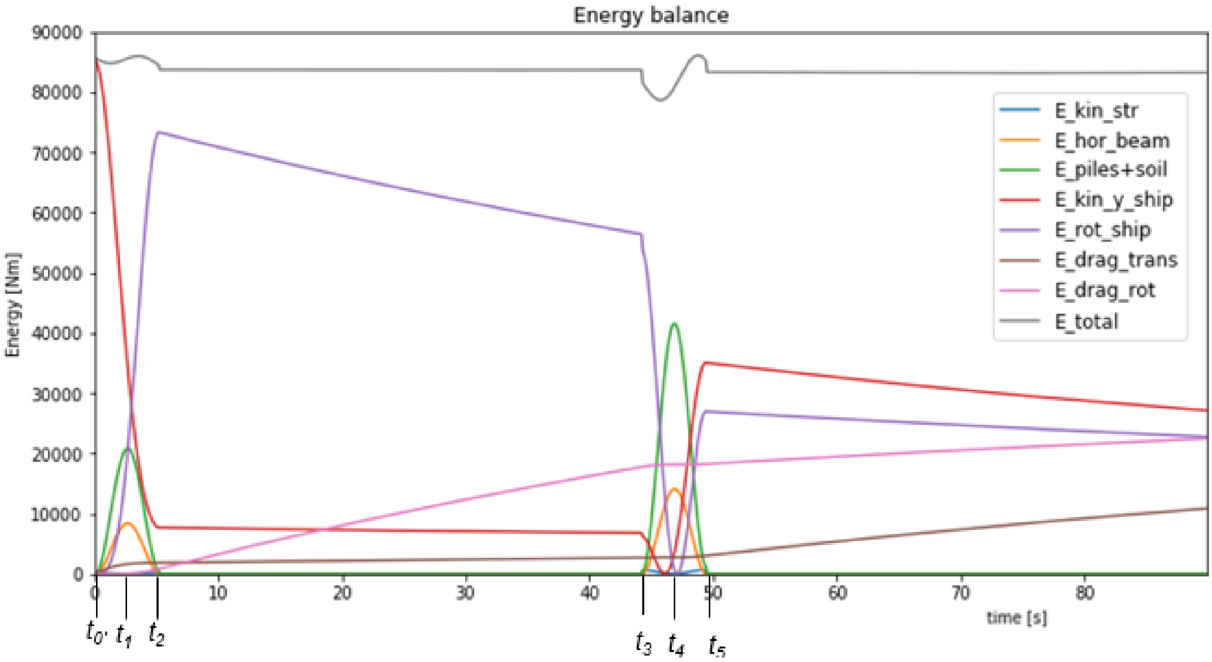


Figure 5.4 Energy balance in model verification

Explanation of the legend in Figure 5.4 is given below. The precise definition of each energy component in the graph is given in appendix C.

E_kin_str	Kinetic energy guide work	[Nm]
E_hor_beam	Potential energy guide work due to bending of berthing beam	[Nm]
E_piles+soil	Potential energy in the vertical piles and soil	[Nm]
E_kin_y_ship	Kinetic energy vessel due to its velocity in y-direction	[Nm]
E_rot_ship	Kinetic energy vessel due to its rotational velocity	[Nm]
E_drag_trans	Work done by drag for movement of the vessel in y-direction	[Nm]
E_drag_rot	Work done by drag for rotational movement of the vessel	[Nm]
E_total	Summation of the seven components mentioned above	[Nm]

In the graph in Figure 5.4 three components are omitted, being: the kinetic energy of the vessel due to its velocity in x-direction, the work done by the friction force and the work done by the engine of the vessel. The velocity in y-direction is proportional with tan(5°) (≈0.09) times the velocity in x-direction. Since the kinetic energy is related to the square of the velocity, the kinetic energy of the vessel in x-direction is significantly larger than the kinetic energy in y-direction. As a result, the kinetic energy in x-direction would dominate the energy balance and the behaviour of the components presented in Figure 5.4 would become indistinguishable.

The work that is done by the friction force and the engine, mainly affects the velocity in x-direction and is therefore also not considered in Figure 5.4.

At the start of the brush collision, at $t=t_0$, all the considered energy in the system consists of kinetic energy due to movement of the vessel in y-direction. After $t_{0'}$ an eccentric force is acting on the vessel, see also Figure 5.2, leading to a continuous increase of the kinetic energy of the vessel due to its rotation until the first contact is finished at $t=t_2$.

The kinetic energy of the vessel due to its translational velocity in y-direction is continuously decreasing during the first impact, implying a decrease of the mass centre's velocity in y-direction. The kinetic energy in y-direction, however, does not become equal to zero during the first impact. This means that the velocity in y-direction of the centre of gravity of the vessel did not change in sign, and thus, that the mass centre of the ship is still moving to towards the guide work.

The potential energy in the vertical piles, as well as the potential energy in the horizontal berthing beam, is at its maximum during the first impact at $t=t_1$. At this moment, also the maximum force of the first impact occurs, indicating a correct correspondence between the moment of maximum contact force and the moment of maximum potential energy in the structure, vertical piles and the soil.

Between the end of the first contact and the start of the second contact, there is no force working on the vessel generated by the structure, see Figure 5.2. Nonetheless, the vessel experiences drag forces by the water for both its translational and its rotational movement. As a result, there is a decreasing line for *E_rot_ship* and *E_kin_y_ship* between t_2 and t_3 in Figure 5.4, and a continuously increasing line for the work done by the drag for the rotational and the translational movement of the vessel.

For the second impact, between t_3 and t_5, a similar behaviour for the potential energy in the vertical piles, soil and the horizontal berthing beam can distinguished as for the first impact. At t_4, when the maximum contact force is reached, also the two potential energy lines reach their maximum.

During the second impact more energy has been absorbed by the structure and the soil than during the first impact, see Figure 5.4. Consequently, the deformation and the contact force are larger for the second impact than for the first impact.

Furthermore, it can be seen from Figure 5.4 that for the first impact the translational movement of the vessel is most important, since this movement contains all the considered initial energy of the system. Not all of this initial kinetic energy is to be converted into potential energy in the structure and soil during the first impact. A significant part of the initial translational kinetic energy is converted into rotational kinetic energy.

For the second impact, the impact is dominated by the rotational movement of the vessel, which is in accordance with measurements that have been performed for large seagoing vessels in the port of Rotterdam (Roubos, Groenewegen, & Peters, 2016). From the energy balance in Figure 5.4, it can be seen that both lines related to the kinetic energy of the vessel become zero during the second impact. This means that both the translational and rotational movement of the vessel changed in sign during the second impact, which is necessary to lose contact with the guide work.

It is then very well explicable that the maximum occurring contact force of the second impact is larger than the maximum force during the first contact. During the first impact a part of the initial translational kinetic energy of the vessel can be transferred to rotational kinetic energy of the vessel, while for the second impact this is not possible.

Finally, a sinusoidal disturbance in the total energy line can be observed during the first and second impact. The origin of this disturbance is not known. It is expected, however, that the distortion is related to the definition of the energy of the subparts of the Python model, and not related to the description of the contact forces or the movement of the vessel. This expectation arises from the fact that after the second impact is finished, the same energy level as before the second impact is present.

A thorough investigation of the potential energy stored in the soil and vertical piles is executed by examining the potential energy of each pile and the surrounding soil separately, instead of using a summation of these contributions. This, however, has not led to an explanation of the disturbance in the energy balance.

Also, other mechanisms exist that can contribute to the dissipation of energy. Examples that are applicable for collisions between vessels and guide works are given in the list on the next page. The aspects mentioned in the list, however, are not incorporated in the created Python model. Therefore, the distortion in the energy balance cannot be explained by these phenomena.

Energy dissipative contributions:

- Plastic deformation of the vessel's hull
- Plastic deformation of the structure
- Plastic deformation of the soil
- Radiation damping by the soil
- Radiation damping by the water
- Rolling movement of the vessel

5.3 Verification by analysis of the movement

Within this section a qualitative assessment of the movement of the vessel is made in order to check the correctness of the Python model. First, a top view of the movement is given, followed by an investigating of the movement in y-direction, its rotation, and finally its movement in the x-direction.

Figure 5.5 presents the movement of the vessel in the time domain. Using time intervals of ten seconds, the position of the vessel is shown in this figure. The blue line in Figure 5.5 indicates the location of the mass centre in the course of time. The used coordinate system is also indicated in this figure.

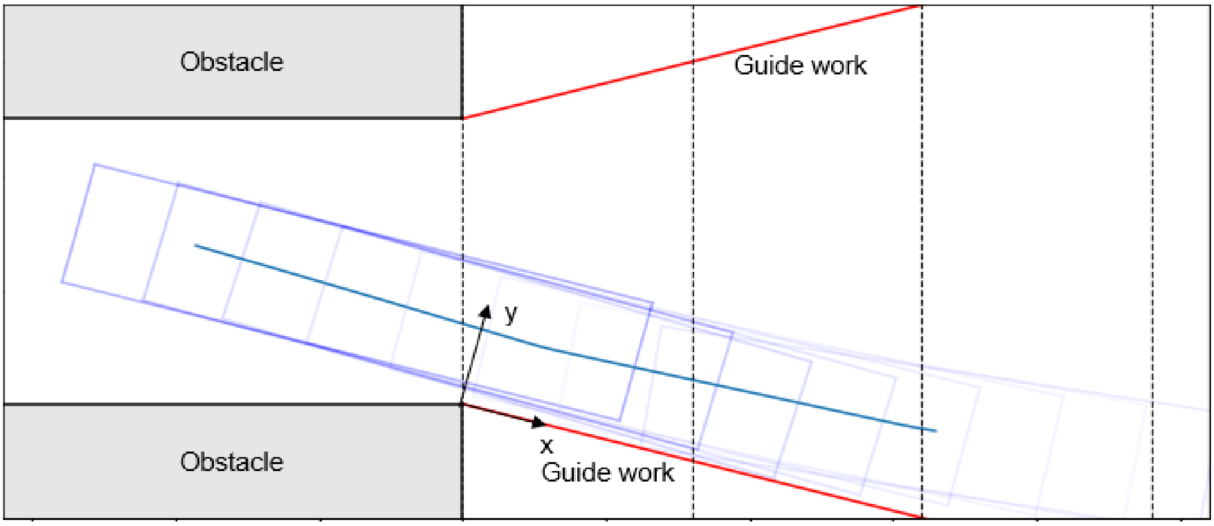


Figure 5.5 Movement vessel in model verification

Figure 5.6 and Figure 5.7 present the coordinates and the velocities of the vessel in y-direction respectively. For the notation of coordinates and velocities, the reader is referred to section 4.2.1 and Figure 4.3.

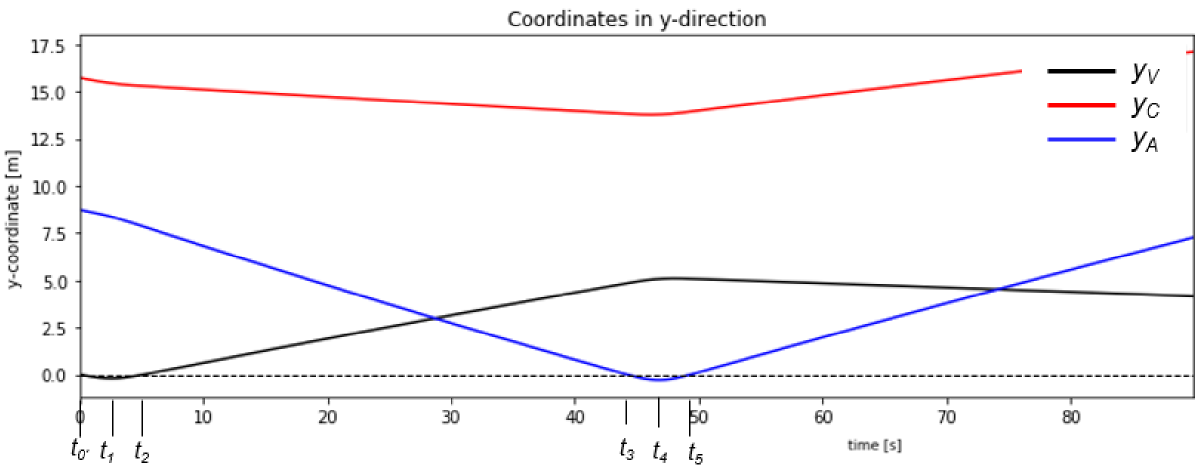


Figure 5.6 Coordinates y-direction in model verification

It can be seen from Figure 5.6 that from $t_{0'}$ until t_1 the bow of the vessel is moving in the negative y-direction, corresponding with an increasing contact force, see Figure 5.2. Then, from t_1 onwards to t_2 the bow starts to move in the positive y-direction, resulting in a decrease of the contact force. Subsequently, from t_2 until t_3 all y-coordinates of the vessel are larger than zero, which means that there is no contact between the guide work and the vessel.

Between t_3 and t_4 the stern of the vessel starts to move into the negative y-direction. At t_4 the maximum deformation is reached, and the deformation starts to decrease again until t_5. From t_5 onwards, no contact between the vessel and the guide work takes place anymore.

With respect to the considered velocities in y-direction it can be seen from Figure 5.7 that the starting velocity of the three considered points of the vessel is equal. Between $t_{0'}$ and t_2 the velocity in y-direction of the centre of gravity of the vessel starts to decrease due to the contact force that is present.

Also, the gain of the yawing motion results in the fact that the velocities in y-direction of the stern and the bow start to differ. Figure 5.7 also clearly shows that after the first contact has ended, the mass centre of the vessel is still has a negative velocity in y-direction and is thus moving towards the guide work. During the second impact, subsequently, the mass centre of the vessel gains a positive velocity and starts moving away from the guide work. Furthermore it can be observed that, in absolute terms, the velocities in y-direction are decreasing due to the drag of the water.

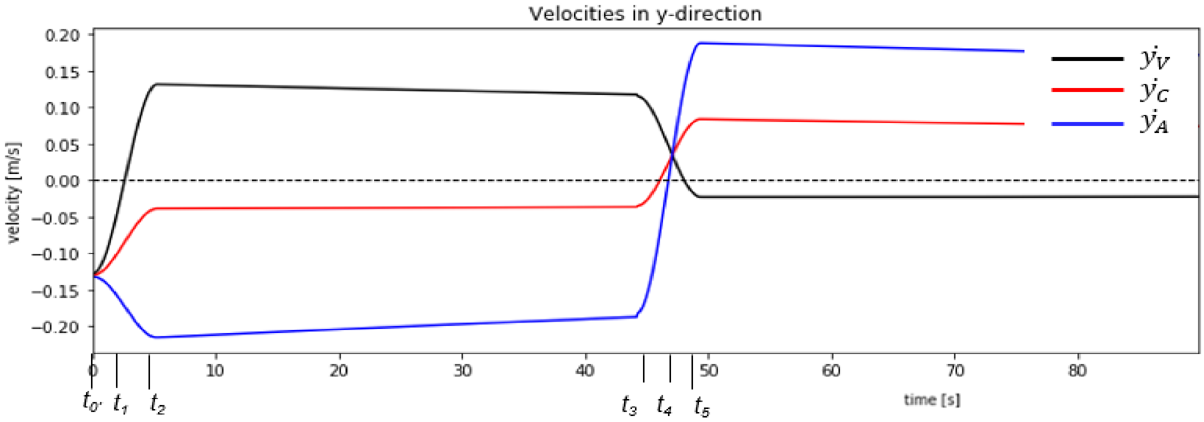


Figure 5.7 Velocities y-direction in model verification

The initial angle of contact between the vessel and the guide work is equal to 5 degrees. As a result of the eccentric contact force during the second impact, a negative yawing movement of the vessel is induced, leading to a decrease of the angle θ, see Figure 5.8. After t_4, angle θ starts to increase again due to the eccentric contact with respect to the centre of gravity at the stern of the vessel, inducing a positive yawing movement.

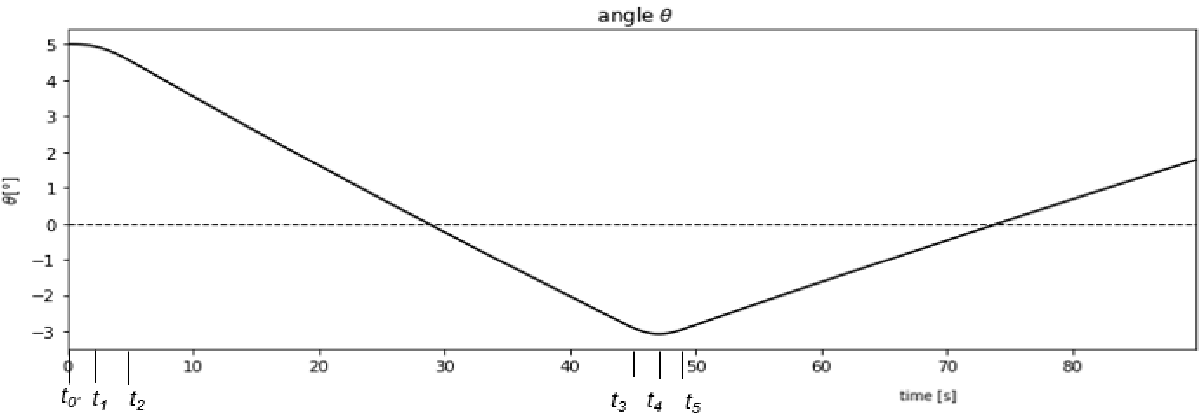


Figure 5.8 Angle θ verification

In Figure 5.9 the angular velocity of the mass centre of the colliding vessel is presented. It is clearly visible that the angular velocity is increasing during the first impact. After t_2 the vessel is not in contact anymore with the guide work until t_3 and only experiences drag of the water, hence the decrease of the angular velocity in this time interval. During the second impact a positive yawing movement is generated by the contact of the stern with the guide work. Finally, after t_5 the contact between the vessel and the guide work is finished and a decrease of the rotational velocity is visible due to the drag by the water.

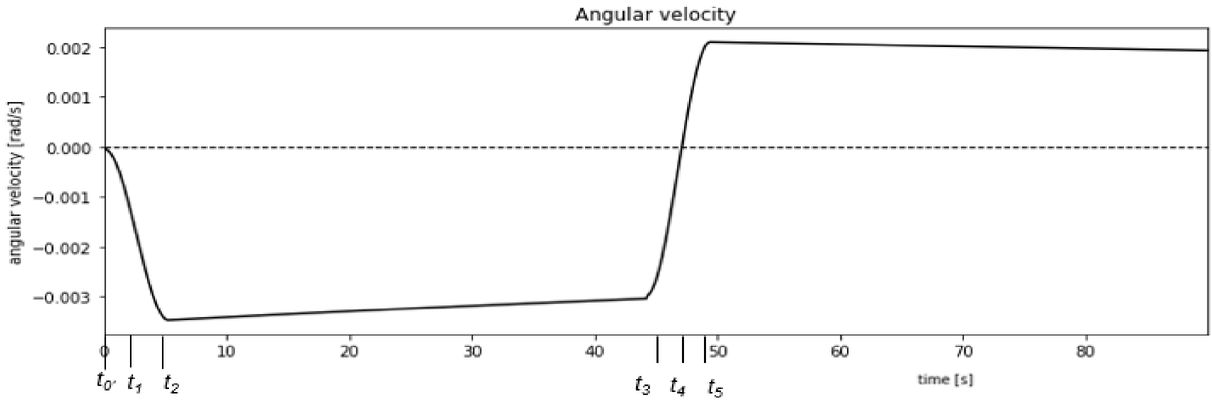


Figure 5.9 Angular velocity verification

To finish this subsection, the movement of the colliding vessel in x-direction is presented and discussed. The x-coordinates of the bow, mass centre and the stern vary linearly, see Figure 5.10, implying a constant velocity in the x-direction.

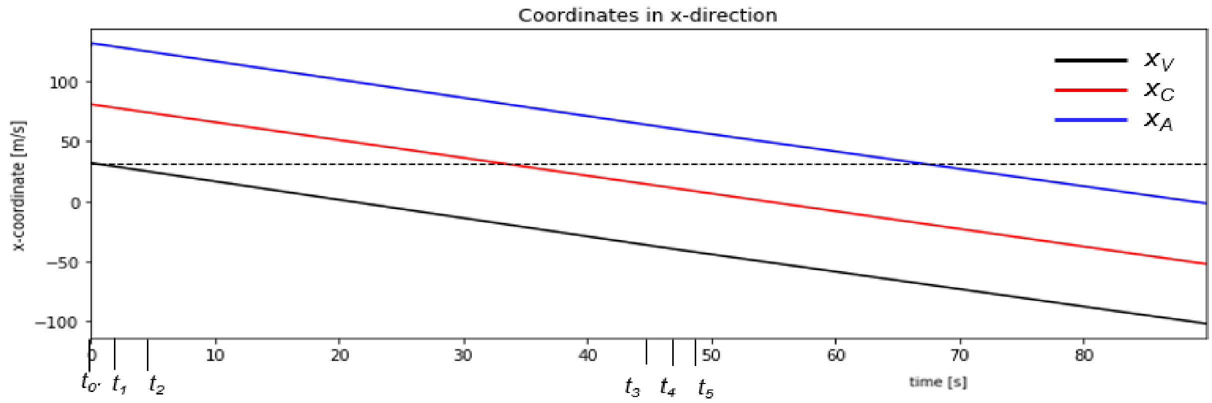


Figure 5.10 Coordinates x-direction verification

The velocity in x-direction is almost constant during the whole movement. Only a small decrease in the velocity in x-direction of the mass centre can be observed as a result of the friction forces during the first and second impact. The jump in the line for $\dot{x}_A$ is caused by the exchange of momentum at the very first contact of the second impact.

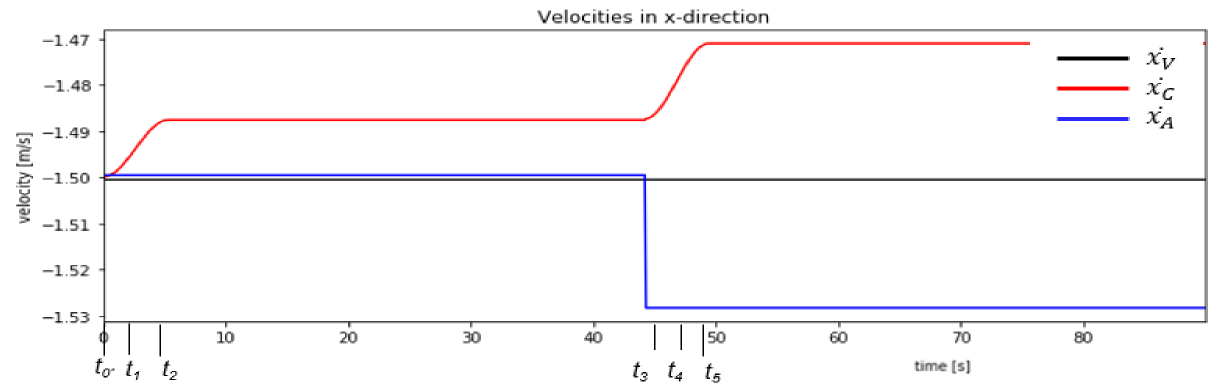


Figure 5.11 Velocities x-direction verification

5.4 Relation to design guidelines

Finally, the link is made with the design guidelines of the EAU and the PIANC for this verification. Based on the input of Table 5.1 and Equation 5.1 it is possible to compute the amount of kinetic energy of the vessel that needs to be absorbed by the structure according to the EAU.

$$E_{kin} = \frac{1}{2} m v^2 C_m C_s C_c C_E \tag{5.1}$$

where:

E_{kin}	Kinetic energy of the vessel to be absorbed by the dolphin	[Nm]
m	Mass of the vessel $=10 \cdot 10^6$ kg	[kg]
v	Total translational velocity of the centre of gravity of the ship $=1.5\tan(5°) = 0.13$ m/s	[m/s]
C_m	Virtual mass factor =1.1	[-]
C_s	Ship flexibility factor =1.0	[-]
C_c	Waterfront structure attenuation factor =1.0	[-]
C_E	Eccentricity factor $= \frac{k^2 + r^2\cos^2(\phi)}{k^2+r^2} + \frac{\omega r}{v}\frac{2k^2 \sin(\phi)}{k^2+r^2} + \frac{\omega^2 r^2}{v^2}\frac{k^2}{k^2+r^2} = 0.31$	[-]

with:

k	Radius of gyration of the vessel $= 0.29\, L_{ship} = 29.09$ m	[m]
r	Distance mass centre vessel to contact point $= 0.5\sqrt{L_{ship}^2 + B_{ship}^2} = 51.43$ m	[m]
ω	angular velocity =0.0 rad/s	[rad/s]
ϕ	angle between velocity vector v and r =72.2°	[°]
v	berthing velocity = 0.13 m/s	[m/s]

With the values that are used within this verification, the kinetic energy that is to be absorbed by the structure according to the EAU for the first impact, see Equation (5.1), is equal to:

$$E_{kin} = \frac{10 \cdot 10^6 \cdot 0.13^2}{2} \cdot 1.1 \cdot 1.0 \cdot 1.0 \cdot 0.31 = 29\,667 \text{ Nm} \tag{5.2}$$

From the created model it follows that the energy that is absorbed by the structure and the soil during the first impact is equal to 29 238 Nm, see Figure 5.4. This verification shows that for the first impact the amount of kinetic energy that is actually absorbed by the structure, is very well in agreement with the expected amount of energy to be absorbed by the structure according to the EAU guidelines.

However, it also shows that guide works tend to be under-dimensioned in case of a second impact, since the amount of kinetic energy that needs to be converted into potential energy during the second impact is significantly higher than the result of Equation (5.2).

Based on the verification that is performed within this chapter and the logical explanations for the found results, the created model is found to be sufficiently accurate to use for a variational study to investigate the influence of the input parameters of the model, see Table 5.1.

6. Results: influence of design parameters on brush collisions

Within this chapter the results of the Python model are presented. First, the results for the magnitude of the contact force and the energy absorption are discussed, followed by an evaluation of the required length of a guide work in different design situations.

6.1 Description of the parameters

The Python model starts with the definition of the parameters in the system, as depicted in Table 6.1. In order to avoid confusion, all parameters are given in SI units, except for the initial contact angle which is given in degrees. All other quantities, for example the moment of inertia and the mass of the structure, are derived from this set of defined parameters. Within this research, the nine parameters that are highlighted in Table 6.1 are varied to investigate their influence on the second impact and the duration of the brush collision.

Table 6.1 Parameter definition

Parameter	Unit	Explanation
Guide work		
• `D1`	[m]	Outside diameter vertical piles
• `t1`	[m]	Wall thickness vertical piles
• `D2`	[m]	Outside diameter horizontal tube
• `t2`	[m]	Wall thickness horizontal tube
• `L_hor`	[m]	Centre to centre distance vertical piles
• `L_ver`	[m]	Distance horizontal tube to bed level
• `N_piles`	[-]	Number of vertical piles
• `N_beams`	[-]	Number of horizontal tubes
Soil		
• `k_trans(F)`	[N/m]	Trend line for translational spring stiffness
• `k_rot(F)`	[Nm/rad]	Trend line for rotational spring stiffness
• `depth(F)`	[m]	Trend line for depth *t*
Vessel		
• `m`	[kg]	Mass ship without added water mass
• `draught`	[m]	Draught of the vessel
• `L_ship`	[m]	Length of the vessel
• `B_ship`	[m]	Width of the vessel
• `H_ship`	[m]	Total height of the vessel
• `Theta_degree`	[°]	Initial angle of contact
• `v_x_C_0`	[m/s]	Initial velocity in x-direction mass centre
Other		
• `E`	[N/m^2]	Young's modulus steel
• `mu`	[-]	Friction coefficient
• `rho_steel`	[kg/m^3]	Density steel
• `time_step_0`	[s]	Initial time step
• `time_step`	[s]	Time step
• `rho_water`	[kg/m^3]	Density water
• `C_w`	[-]	Drag coefficient
• `C_m`	[-]	Virtual mass factor
• `F_init`	[N]	Dummy force
• `x_V`	[m]	x-coordinate first contact

After the input of the model is defined, the derived properties are computed. A list of this derived properties is given below:

- Mass of the structure [kg]
- Moment of inertia of the vessel including added mass [kg m^2]
- Bending stiffness horizontal tube [N m^2]
- Bending stiffness vertical piles [N m^2]
- θ in radians in order to comply with the programming environment [rad]
- *a, b, c* and *d* as defined in Figure 4.3 [m]

6.2 Influence input parameters on energy absorption

This subsection discusses the influence of the investigated input parameters on the amount of kinetic energy that is absorbed during the first impact and second impact. Within this chapter, the term *energy absorption* refers to the maximum amount of kinetic energy during an impact that is taken up by the soil and the structure as potential energy. The amount of converted kinetic energy into potential energy can then be related to the force that is being exerted on the guide work.

In order to preserve the readability of the report, the graphs presenting the contact force between the vessel and the structure during the first and second impact are reduced in size in the main report. The enlarged versions are added in appendix D.

The graphs within this subsection are constructed in the following manner: on the horizontal axis the varied parameter is plotted. The left vertical axis gives the amount of energy that is absorbed by the structure and the soil, scaled by the total initial kinetic energy that is present in the system. Simultaneously, the right vertical axis gives the eccentricity factor as defined in the EAU. The total initial energy for scaling purposes is, see Equation (6.1), defined as:

$$E_{initial} = 0.5\, m\, v^2\, C_m C_s C_c \quad \text{[Nm]} \tag{6.1}$$

Since the C_m factor is already included in the created Python model and the C_s and C_c factor are equal to 1.0, the reduction of the to be absorbed kinetic energy due to the eccentricity becomes visible. By presenting the results in this way, it can clearly be seen whether or not the EAU recommendation for the amount kinetic energy of kinetic energy that needs to be absorbed is in line with the model. The results are compactly presented in Table 6.2 to Table 6.10, while in appendix D the enlarged version of the graphs in these tables are presented.

In the graphs below, short notations are used in the legends. *E_abs 1st impact* and *E_abs 2nd impact* denote the maximum of amount of potential energy in the structure and soil during the first and second impact respectively. *Eccentricity factor* refers to the eccentricity factor as defined by the EAU, see also Equation (2.2). Finally, *F1_max* and *F2_max* are used for the denoting the maximum occurring contact force respectively the first and second impact.

Table 6.2 Influence mass vessel on absorbed energy and contact force

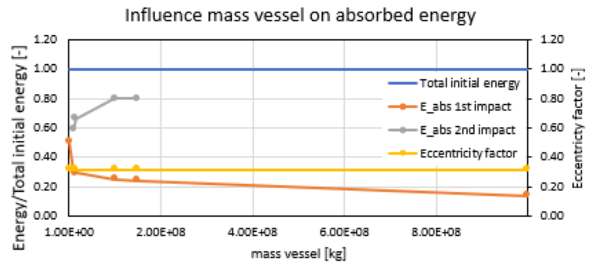


Figure 6.1 Influence mass vessel on absorbed energy during first and second impact

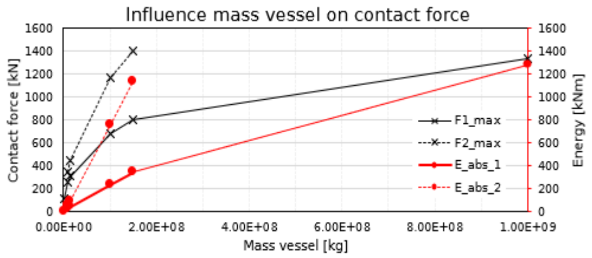


Figure 6.2 Influence mass vessel on contact force during first and second impact

It also needs to be mentioned that 90 seconds of analysis are performed for each situation. Consequently, for some of the investigated cases no second impact is recorded, since this second impact did not occur at all, or did not occur within the elapsed time.

First of all, the influence of the mass of the vessel on the energy absorption during the first and second impact is investigated, see Figure 6.1. It can be observed from these figures that for all the presented results, more energy is absorbed during the second impact than during the first impact. This means that for every investigated value of the mass of the vessel, the maximum force associated with the second impact is larger than the force that occurs during the first impact, see Figure 6.2.

On top of that, Figure 6.1 shows that application of the eccentricity factor as recommended by the EAU gives an overestimation of the amount of kinetic energy that needs to be absorbed during the first impact, except for the vessel with a mass of 1*E*6 kg. For the second impact, however, the amount of kinetic energy that needs to be absorbed is larger than one would expect based on the recommendation of the EAU. Making use of the traditional design method, will then lead to an underestimation of the loading on the guide work.

Table 6.3 Influence length vessel on absorbed energy and contact force

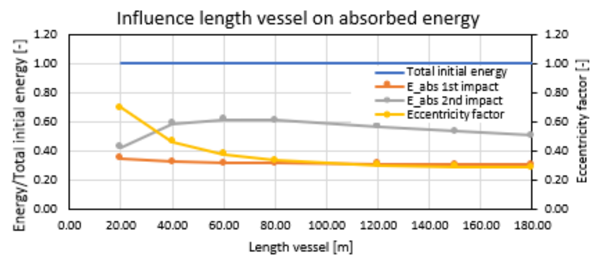


Figure 6.3 Influence length vessel on absorbed energy during first and second impact

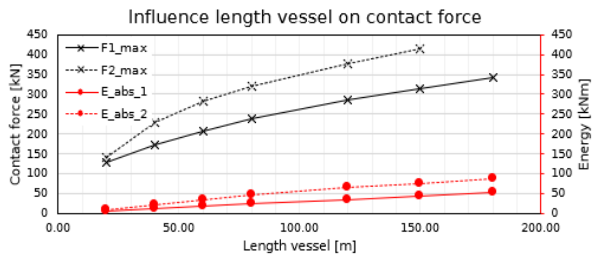


Figure 6.4 Influence length vessel on contact force during first and second impact

Secondly, the influence of the length of the colliding vessel is investigated. In order to maintain realistic scenarios, the original mass of the colliding vessel is scaled accordingly to the variation in length. For example: the reference length and mass of the vessel are 100 m and 10*E*7 kg, respectively. In case a vessel with a length of 80 m is investigated, the reference mass is scaled by 80m/100m = 0.8 [-].

Similar to the influence of the mass of the vessel, it can be seen that for every investigated situation, the second impact is governing the first impact, see Figure 6.4. On top of that, the figure shows that application of the eccentricity factor according to the EAU leads to an overestimation of the amount of kinetic energy that needs to be absorbed during the first impact for vessels up to eighty meters. Nonetheless, the effects of the second impact are still underestimated in case the EAU recommendation is used.

Table 6.4 Influence draught vessel on absorbed energy and contact force

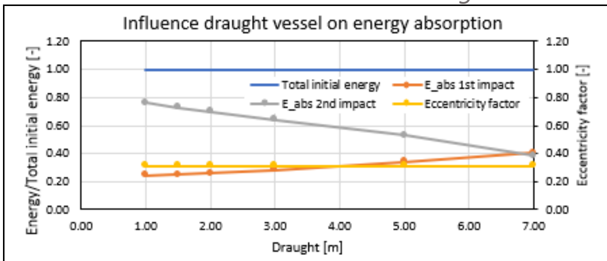


Figure 6.5 Influence draught vessel on absorbed energy during first and second impact

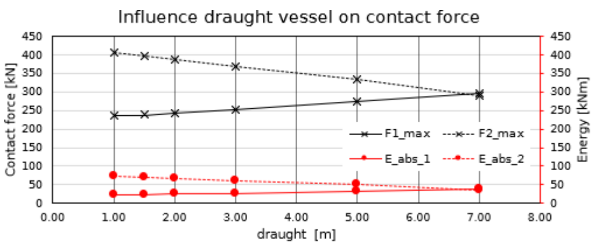


Figure 6.6 Influence draught vessel on contact force during first and second impact

From Figure 6.5 and Figure 6.6 it can be seen that an increase of the draught leads to a more severe impact force during the first impact. Contrary to the first impact, the magnitude of the contact force during the second impact decreases for an increasing draught of the colliding vessel.

Explanation for this latter phenomenon can be found in the drag that the vessel experiences. An increase of the draught leads to an increase of the drag force as well. As a consequence, kinetic energy is dissipated before the second impact takes place and thus does not need to be absorbed by the structure anymore.

For vessels with a draught up to 7 meters, the second impact dominates over the first impact. On top of that, application of the EAU recommendation again leads to an under-estimation of the energy, and thus force, that is exerted on the guide work during the second impact.

Table 6.5 Influence diameter berthing beam on absorbed energy and contact force

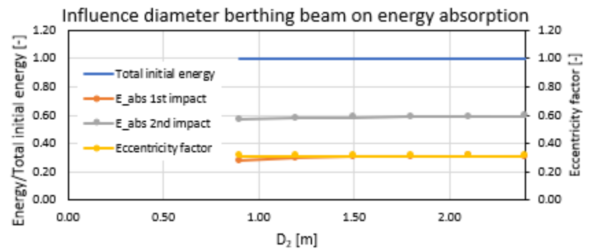


Figure 6.7 Influence diameter berthing beam on absorbed energy during first and second impact

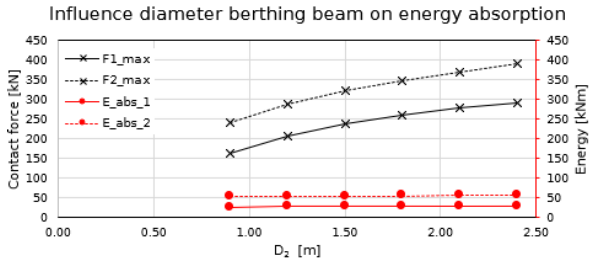


Figure 6.8 Influence diameter berthing beam on contact force during first and second impact

Subsequently the diameter of the horizontal berthing beam and the vertical piles is varied to examine the effect on the energy absorption and the contact force between the vessel and the guide work. The influence of these two parameters are treated simultaneously, because of the resemblance of the results, see Figure 6.7, Figure 6.8, Figure 6.10 and Figure 6.11.

Figure 6.8 and Figure 6.11 clearly show that the amount of energy that is absorbed remains approximately equal for all investigated diameters. The associated contact force, however, is not constant and increases for an increasing tube diameter. Since a larger diameter generates a greater moment of inertia I_z of the cross-section (I_z is proportional with the diameter to the power 4), which then leads to an increased bending stiffness. From Figure 6.9 it can be see that a higher contact force and smaller deformation occur for an increasing stiffness to absorb the same amount of energy.

Figure 6.9 Energy absorption for different stiffnesses

Moreover, it follows from Figure 6.7 and Figure 6.10 that the ratio between the magnitude of the contact force during the second impact and the first impact remains more or less equal for the various investigated diameters.

It can also be observed that, when the diameters of the cross-sections are varied, application of the C_E factor according to the EAU leads to a correct amount of energy that needs to be absorbed during the first impact, but not for the second impact. The contact force of the second impact is approximately 35% larger than the contact force of first impact, while the EAU recommendation is intended for the design of the first impact. As a result, guide works tend to be under-dimensioned when it comes to a second impact due to a brush collision.

Table 6.6 Influence diameter vertical piles on absorbed energy and contact force

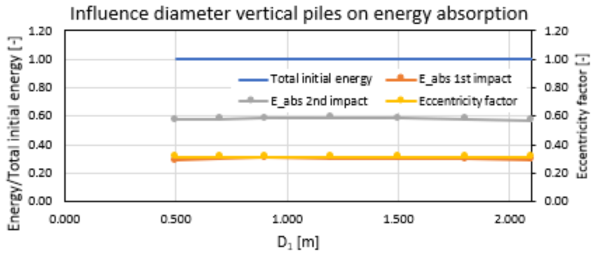


Figure 6.10 Influence diameter vertical piles on absorbed energy during first and second impact

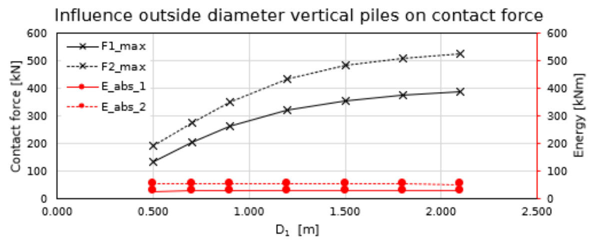


Figure 6.11 Influence diameter vertical piles on contact force during first and second impact

In addition to the influence of the outside diameter of the tubes, the vertical length of the piles above bed level can be mentioned. The effect of the variation of the pile length is presented in Figure 6.12 and Figure 6.13. An increase of the pile length above bed level results in a decrease of the overall stiffness of a guide work. Following the same line of reasoning as for the influence of diameter of the tubular cross-sections, a smaller contact force is needed to absorb the same amount of kinetic energy by means of deflection.

The current recommendation by the EAU for the amount of energy that needs to be absorbed shows great resemblance with the results for the first impact in Figure 6.12. Nonetheless, the contact force during the second impact governs the impact during the first contact.

Table 6.7 Influence pile length above bed level on absorbed energy and contact force

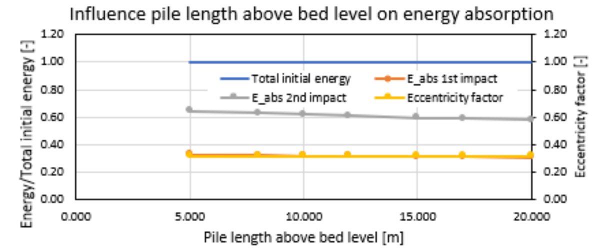


Figure 6.12 Influence pile length above bed level on abs. energy during first and second impact

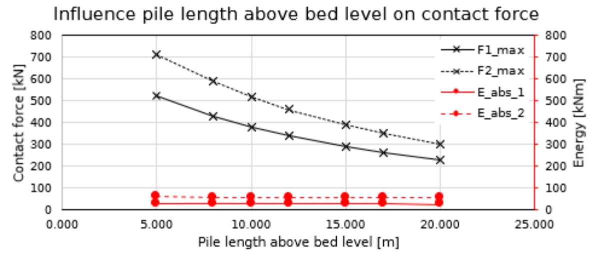


Figure 6.13 Influence pile length above bed level on contact force during first and second impact

Table 6.8 Influence location first contact on absorbed energy and contact force

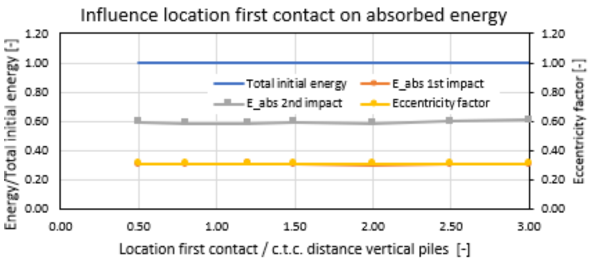


Figure 6.14 Influence location of first contact on absorbed energy during first and second impact

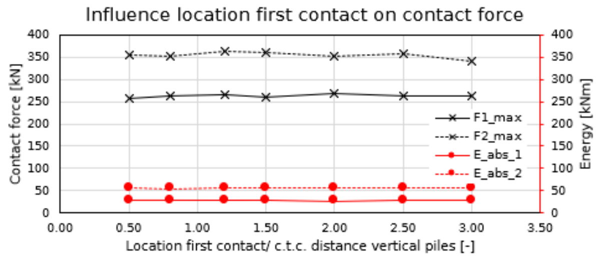


Figure 6.15 Influence location of first contact on the contact force during first and second impact

The seventh parameter that is investigated is the location of the first contact. Results of this analysis are presented in Figure 6.14 and Figure 6.15. The two figures clearly show that the amount of kinetic energy that is absorbed during the first and second collision is approximately equal for the investigated cases. More variation, however, is found in the contact force that is present between the vessel and the guide work. Explanation for this spread in the results can be found in the fact that the reaction of the guide work is stiffer near the vertical piles.

Irrespective of the location of the first contact, the magnitude of the maximum contact force of the second impact is 35-40% larger than the maximum occurring force during the first impact.

Table 6.9 Influence initial contact angle on absorbed energy and contact force

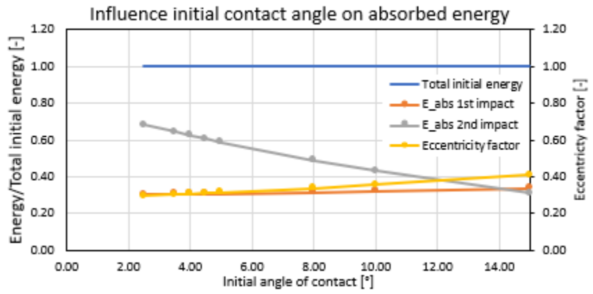


Figure 6.16 Influence initial contact angle on absorbed energy during first and second impact

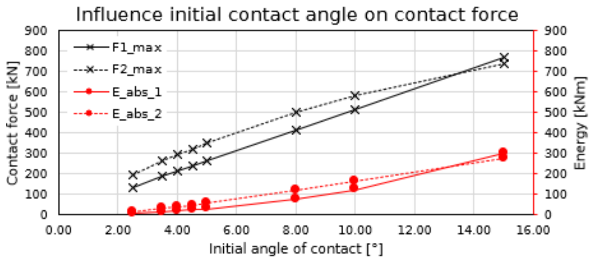


Figure 6.17 Influence initial contact angle on the contact force during first and second impact

The before last parameter that is examined within this study is the angle of contact θ in the initial situation. The initial velocity in x-direction is remained unchanged during the analysis and the initial velocity in y-direction is related to the initial velocity in x-direction by $\tan(\theta)$. Consequently, an increase of the initial contact angle, also leads to an increase of the initial velocity in y-direction. With that being said, the influence of the initial contact angle on the absorbed energy and contact force during the first and second impact is presented in Figure 6.16 and Figure 6.17.

From Figure 6.16 it follows that application of the eccentricity factor according to the EAU leads to a correct amount of kinetic energy that is absorbed by the structure for initial contact angle up to 6 degrees. From initial contact angles larger than 6 degrees an overestimation of kinetic energy is found by application of the eccentricity factor.

The amount of potential energy in the structure and soil during the second impact is larger than during the first impact for initial contact angles smaller than 12 degrees, as can be observed from Figure 6.16 and Figure 6.17. Therefore, also a larger contact force is found for the second impact than for the first impact for initial contact angles smaller than 13 degrees.

A possible explanation for the fact that the first impact starts dominating the second impact for initial contact angles larger than 13 degrees is the rotational drag; the larger the contact angle of the first contact, the more time is required for the vessel to achieve a rotation in the other direction. As the interval between the end of the first impact and the start of the second impact increases, more kinetic energy is dissipated before the second impact takes place.

Table 6.10 Influence initial velocity on absorbed energy and contact force

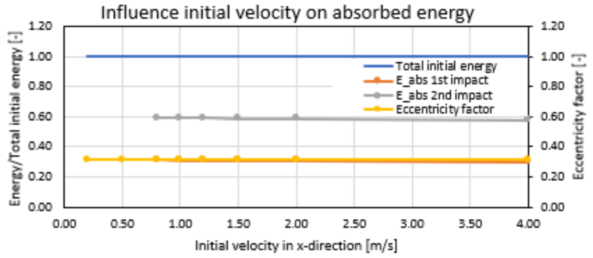


Figure 6.18 Influence initial velocity on absorbed energy during first and second impact

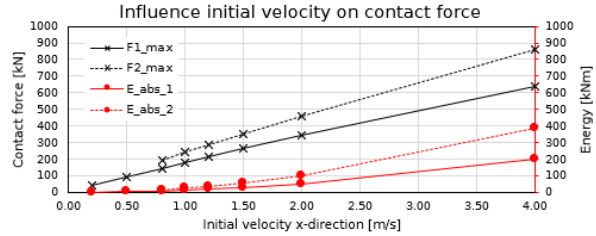


Figure 6.19 Influence initial velocity on the contact force during the first and second impact

Finally, the initial velocity of the vessel in x-direction is varied to check its effect on the second impact. Figure 6.18 demonstrates that the energy absorbed during the second impact is larger than the absorbed energy during the first impact and that the ratio between the contact force of the first and second impact is constant for an increasing initial velocity.

In absolute terms the amount of kinetic energy that is converted into potential energy, as well as the contact force, increases for an increasing initial velocity. This is simply due to the fact that more kinetic energy is present in the system.

6.3 Influence input parameters on required length guide work

This section presents the influence of the input parameter on the length of the guide work that is needed to facilitate the brush collision. The required length L_{req} consists of two distances being: ΔL_1 and ΔL_2, or in formula form:

$$L_{req} = \Delta L_1 + \Delta L_2 \tag{6.2}$$

where:

L_{req}	Required length for the guide work to facilitate the brush collision	[m]
ΔL_1	Length over which the contact with the bow takes place	[m]
ΔL_2	Length between location of the first contact of the bow and first contact of the stern with the guide work	[m]

where:

ΔL_1	=duration first contact [s] · velocity in x-direction [m/s]	[m]
ΔL_2	=x-position first contact bow – x-position first contact stern	[m]

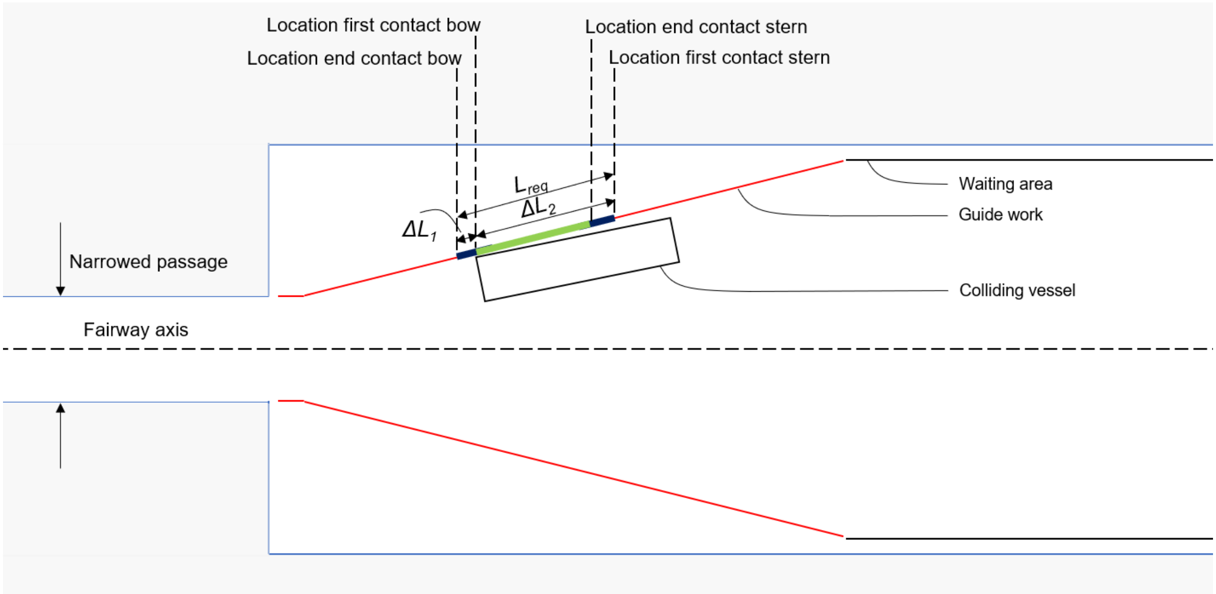


Figure 6.20 Clarification required length guide work

Similar to section 6.2, the graphs for the varied input parameters are presented in small scale, while the enlarged versions of these graph are given in appendix D.3.

From Figure 5.2 it follows that during both moments of contact, the contact force displays a sinusoidal behaviour. Consequently, the sinusoidal behaviour of the contact force between the vessel and the guide work can be characterised by means of a frequency.

When the analogy is made with an undamped mass-spring system, in which the natural frequency ω is equal to $\sqrt{k/m}$, the graphs in this section can be better understood. In this expression k refers to the stiffness of the system, while m refers to the mass that is in motion. Furthermore, it holds that:

$$\omega = \frac{2\pi}{T} \leftrightarrow T = \frac{2\pi}{\omega} \tag{6.3}$$

where:

ω	Frequency of the brush collision	[rad/s]
T	Length of one sinusoidal period	[s]

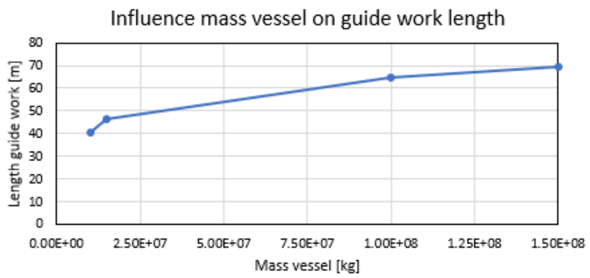


Figure 6.21 Influence mass vessel on the required guide work length

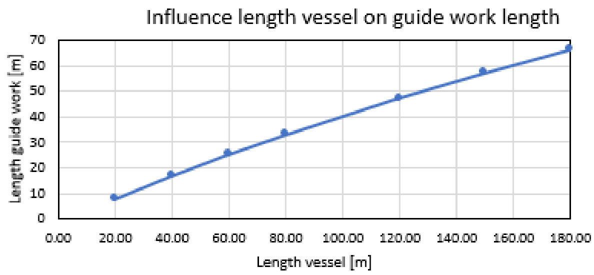


Figure 6.22 Influence length vessel on the required guide work length

Figure 6.21 shows that in increase of the mass of the brushing vessel, leads to an increase of the required length of a guide work as well. Since the frequency of the impact decreases as a result of an increasing mass of the vessel, the duration of an impact becomes longer. Consequently, ΔL_1 increases and more length is required to facilitate the brush collision.

Subsequently, the influence of the vessel's length on the required length of a guide work is investigated, see Figure 6.22. Similar to section 6.2, the mass of the vessel is scaled accordingly with the length of the vessel to avoid unrealistic scenarios. According to the results presented in Figure 6.22, a longer vessel generates the need for a bigger length of the guide work.

The reason for this increase of required length is twofold. First of all, the mass of the vessel increases for an increasing length of the vessel, resulting in a larger ΔL_1. Secondly, it is self-evident that ΔL_2 increases for longer vessels, because of the simple fact that a longer vessel is present.

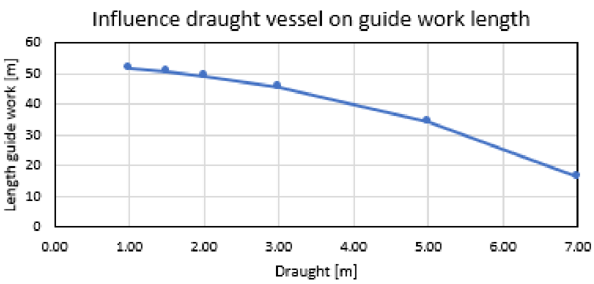


Figure 6.23 Influence draught vessel on the required guide work length

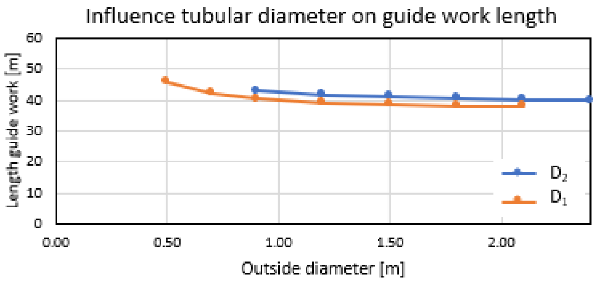


Figure 6.24 Influence diameter horizontal tube on the required guide work length

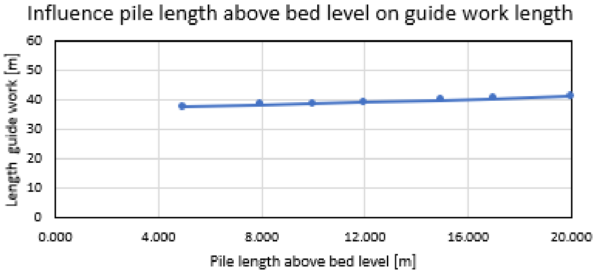


Figure 6.25 Influence pile length above bed level on the required guide work length

Then, Figure 6.23, Figure 6.24 and Figure 6.25 are treated simultaneously, because of the comparable explanation for the required guide work length. In Figure 6.23 and Figure 6.24 the draught of the vessel and the outside diameter of the berthing beam and vertical piles are discussed respectively. An increase of these three parameters, leads to a stiffer response of the system. Consequently, the frequency of the impact increases, and the duration of the impact thus decreases. Decrease of the impact duration leads to a decrease of ΔL_1 and therefore, the required length of the guide work reduces.

Conversely, a larger pile length above bed level of the vertical piles lead to a less stiff behaviour of the guide work. Application of the same line of reasoning as done for the influence of the outside diameter of the cross-sections and the draught, this leads to a larger required length for the guide work.

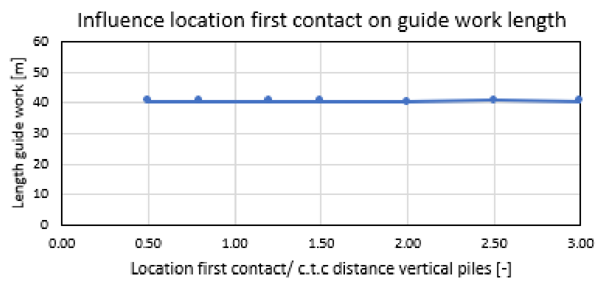


Figure 6.26 Influence location of first contact on the required guide work length

In Figure 6.26 the influence of the location of the first contact on the required guide work length is presented. No notable variation of the required length of a guide work can be found for the different locations of the first impact.

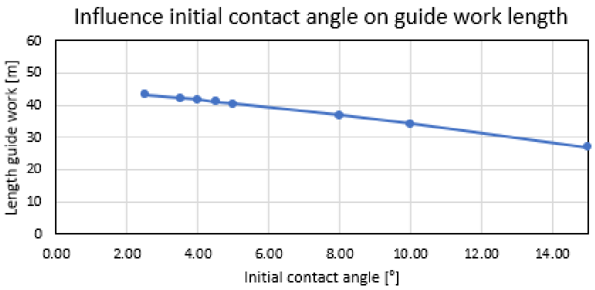


Figure 6.27 Influence initial angle of contact on the required guide work length

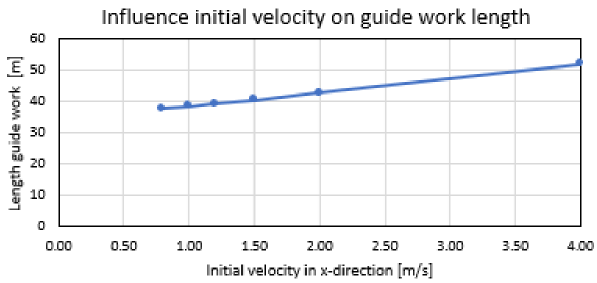


Figure 6.28 Influence initial velocity x-direction on the required guide work length

Furthermore, the initial angle of contact between the vessel and the guide work is varied to investigate the role of this contact angle on the required length of the structure. Figure 6.27 shows that larger contact angle result in a smaller length of the guide work that is necessary to facilitate the brush collision.

The larger the initial contact angle, the more angular displacement needs to be overcome for the stern to hit the structure as well. Consequently, the time interval between the end of the first contact and the start of the second contact increases and the stern has more time to move in the forward direction. A decrease of ΔL_2 and, with that, a decrease of L_{req} is the result.

Finally, the role of the initial velocity on the required length of for guide work is examined, see Figure 6.28. A larger initial velocity leads to an increase of the contact force, as found in Figure 6.19. An increase of the contact force gives rise to an increase of the lateral load on the vertical piles of the structure, resulting in a less stiff behaviour of the soil, which is also depicted in Figure A.9.

A less stiff behaviour of the soil, then, results in a less stiff response of the guide work, causing a smaller frequency of the impact. A lowered frequency of the impact implies an increase of its duration, leading to a larger ΔL_1 and thus a larger required length.

7. Discussion

This chapter presents the discussion of the research. First, the validity of the results is considered, followed by the limitations of the performed analyses. To finish this chapter, the obtained results are interpreted and possible consequences are discussed.

7.1 Validity of the results

With the aid of the performed verification in chapter 5, the validity of the results is shown. Since the sinusoidal behaviour of the contact forces during the first and second impact and the movement of the vessel is in line with existing literature, the results are found to be sufficiently accurate. The only remark is the small inconsistency that appeared in the energy balance during the verification. All found results, however, can be explained in a logical and consistent manner, which was decisive in determining whether the results are valid or not.

Furthermore, even if a small error would exist within the model, the second impact is probably still governing the first impact, since the contact force is more than 30% higher during the second impact than during the first impact in most of the investigated cases.

7.2 Limitations of the performed analyses

Given the complexity of the interplay of various factors during affecting a brush collision, several side notes need to be placed. First of all, potential intervention of the captain is not considered, nor are environmental circumstances taken into account.

In case a (brush)collision occurs, a natural reaction of the captain of the vessel could be to adjust the course of the ship. Also the current in the waterway will influence the movement of the vessel, as well as the wind. The unpredictability of these factors have led to the fact that they are not considered within the model. Field test data or probabilistic computations can be included in the future to adequately incorporate the uncertainties in the model. Further recommendations are made in section 8.2.

Additionally, it is worth mentioning that damping effects are not fully considered. Damping, i.e. energy loss, due to the water is taken into account, but damping due to vibrations in the structure or soil is not. Simultaneously, van der Vorm (1993) defends the choice of initially treating the impact load as a static load because of the relative long duration of an impact (5-30 seconds).

Finally, the representation of the soil by Blum's method is defendable for this research; a quick but coarse approximation for the behaviour of the soil. Nonetheless, usage of more advanced software, e.g. Plaxis 3D or D-Pile Group, will probably generate a more accurate description of the soil behaviour, while at the same time the computational efforts increase.

7.3 Interpretation of the results

The most important finding of this research is the magnitude of the contact force during the second impact. Although current design guidelines as the EAU 2012 and the PIANC focus on the first impact during the design of flexible dolphins (and thus guide works), the second impact is mostly dominant in reality according to the obtained results.

Consequently, existing guide works tend to be under designed. There are, however, several possible explanations why existing guide works are not failing.

First of all, guide works are designed in a linear elastic manner. In case of exceedance of the linear elastic trajectory, (elasto-)plastic behaviour will occur before the dolphin is totally deformed.

On top of that, a safety margin is created by application of safety factors during the design of these structures, see also section 2.4.2. The steel frame of the guide works is then still able to fulfil its function in a linear elastic manner.

The required length for a guide work to facilitate a brush collision is very well obtainable using the created tool. Determining for the total length of a guide work, however, is also dependent on the location where the first contact takes place. In order to provide a useful recommendation for the total length of a guide work, data about the spread in the location of first contact is needed.

Finally, it might be beneficial for a contractor to execute the first vertical pile in smaller dimensions than the other vertical piles. Since it takes a certain time and distance before the contact force reaches its maximum, the first vertical pile will not be as heavily laterally loaded as the other piles. The limited forward velocity of the vessel, however, makes that the contact point also moves relatively slowly. Consequently, the benefit is expected to be marginal, especially if also executional aspects are taken into account.

8. Conclusions and recommendations

In this chapter the conclusions and recommendations of this thesis are presented. Section 8.1 presents the conclusions, while section 8.2 presents the recommendations for both the design practice and further research.

8.1 Conclusions

This section provides the conclusions of the performed research. Firstly, the conclusions with respect to the performed time-dependent analyses are given, followed by conclusions about the translation of the Python model to SCIA Engineer.

8.1.1 Conclusions about brush collision in the time domain

In order to draw conclusions about the results of this research, first the sub questions of this research are answered. After the sub questions are answered the research question is addressed.

1) How do the location of the contact point and contact force vary during a brush collision between an inland navigation vessel and a guide work?

In case of an impact due to a brush collision between a vessel and a guide work, the contact force displays a sinusoidal behaviour, in accordance with Vrijer (1983) and Van der Vorm (1993), during both the first and second impact. After a quarter of a period, which is after 2 to 7 seconds depending on the design parameters, the maximum contact force within one impact is reached. Given the limited forward velocity of a vessel during the approach to a lock, the movement of the contact point along the structure during the first and during the second impact is limited as well.

2) What is the maximum contact force in case of a second impact due to a brush collision between an inland navigation vessel and a guide work?

With respect to the magnitude of the contact force of the second impact during a brush collision, it can be concluded that this second impact force is generally larger than the force associated with the first impact for the investigated scenarios. These results are in line with the findings of the research of Roubos et al. (2016) and Bratteland (1998) for berthing manoeuvres.

For a variation of the outside diameter of the horizontal berthing beam and the vertical piles, as well as for a variation of the pile length above bed level, the contact force of the second impact is 35-40% greater than the maximum force during the first impact. Also, a circa 35% larger contact force during the second impact than during the first impact is found for a variation of the initial velocity, length of the vessel and the location where the first contact takes place.

An increase in the draught of the vessel leads to a decrease of the maximum contact force during the second impact in absolute sense. For a draught of more than 7 meters, the second impact starts to become smaller than the first impact with the used dimensions of the vessel.

In relation to the initial contact angle it is found that an increase of the initial angle of contact leads to decrease of the ratio between the maximum contact force during the second and first impact. For an initial contact angle of 13 degrees and larger, the second impact does not govern the first impact anymore.

Finally, for an increasing mass of the vessel it is observed that the maximum occurring force during the second impact is larger than during the first impact for all investigated cases. In addition to that, the ratio between the force during the second impact and the first impact is continuously increasing for increases vessel masses.

3) What length of a guide work is required to facilitate a brush collision between a guide work and an inland navigation vessel?

The total required length for a guide work is found to be not strictly determined by the length that is needed to facilitate a brush collision. In order to determine an appropriate length for a guide work, knowledge about the spread of the location of the first impact is more important.

The length to facilitate a brush collision is circa 40 meters for the original situation with a 1E7 kg vessel of 100 meters and an initial angle of contact of 5 degrees and an initial velocity in x-direction of 1.5 m/s. Variations of the original situation that lead to a stiffer response of the structure, i.e. increased diameters of the tubular cross-sections or a smaller pile length above bed level, lead to a decrease of this length. Also, an increasing draught and initial contact angle result in a decrease of the length to facilitate a brush collision.

Conversely, a larger mass and length of the colliding vessel, as well as an increased initial velocity, generate the need for longer guide works. Lastly, the location of the first impact appeared to be of negligible influence for the required length of the structure.

Furthermore, it is found that a decrease of the angle of a guide work leads to a smaller contact force and smaller pressure on vessels' hulls, since the initial angle of contact will be reduced.

In which ways is the design of guide works affected by taking into account the time domain?

By taking into account the time domain during the design of guide works, it becomes possible to adequately design these structures, since it follows from the second sub question that the second impact is generally governing the first impact, while in current design practice the first impact is taken as design criterion. Therefore, in future projects or during reassessment of existing guide works the second impact is to be considered as well, instead of only the first impact.

On top of that, based on the answer of the third sub question, a more substantiated choice can be made in the design process for the length of a guide work and the slope under which this structure is placed.

8.1.2 Conclusions about the translation to SCIA Engineer

Regarding the translation of the Python model to SCIA Engineer it can be concluded that it is possible to, in the first place, realise this translation. In addition to that, the automatic post-processing environment enables the automation of design checks and may eventually become beneficial over using the model created in Python.

In its current form, however, the tool using SCIA Engineer requires 7.5 hours of computation time to run the same analysis which takes less than 1 second in the Python model. With some improvements, as described in section 4.5.2, the computation time can be reduced, making the translation to SCIA Engineer a more attractive alternative.

8.2 Recommendations

This section provides recommendations based on the findings within this research. First, recommendations with respect to the design practice are given in section 8.2.1, followed by recommendations for future research in section 8.2.2.

8.2.1 Recommendations for the design practice

Based on the findings in this report, a set of recommendations is made for the design of guide works that are exposed to inland navigation traffic.

- *Introduce a speed limit for inland navigation vessels in the vicinity of narrowed passages where guide works are present*

According to the findings of this study the second impact in case of a brush collision generally governs the first impact, while guide works are normally designed for the first impact. The owner of guide works, mostly *Rijkswaterstaat* in the Netherlands, may consider the introduction of a speed limit of circa 35% such that the contact force during the second impact is equal to the impact for which the structure is originally designed.

- *Use the created tool for future projects*

By using the created tool in Python for future projects, it is possible to determine the contact force and the amount of energy that needs to be absorbed during a second impact, which is generally governing. Using the governing amount of energy that needs to be absorbed, the traditional design approach can then be used.

- *Place guide work under a smaller angle*

Another recommendation for the design of guide works is to place the structures under a smaller angle than the current recommended angle of 1:4; application of a angle of 1:5 already leads to a 2.7 degrees smaller contact angle, resulting in a reduction of circa 135 kN for the maximum contact force. A less oblique guide work is also beneficial to fulfil the criterion with respect to the maximum allowable hull pressure on vessels.

- *Increase stiffness in case of limited space*

In order to reduce the total length of a guide work, the designer might choose for a guide with a high stiffness. This can be realised by application of large diameters in the tubular sections, or by adding extra vertical piles. It has to be mentioned though, that increasing the stiffness also leads to higher contact forces and pressure on the vessel's hull.

8.2.2 Recommendations for future research

Recommendations for future research are given in this subsection.

- *Investigate the role of human intervention:*

Within this research the movement of the vessel is assumed to be totally dependent on the exerted contact force between the guide work and the vessel, combined with the influence of the water. In reality, however, it is expected that the captain of a colliding vessel also tries to steer the vessel in the right direction. The influence of this human interference on brush collisions is therefore an interesting topic for future research.

- *Calibration of the created model by field test data*

The tool created during this thesis contains several simplifications, e.g. the assumption for the rectangular shape of the vessel and the negligence of damping by the structure and the soil. In case fields test are performed, the model can be calibrated, resulting in better predictions for the effects of impacts due to collisions. On top of that, it is possible to visualise the spread in the location where the first impact takes place, which is important in determining an appropriate length of a guide work.

- *Second impact during berthing manoeuvres*

This research focuses on brush collisions between inland navigation vessels and guide works. Given the similar nature and design procedure of flexible dolphins exposed to (brush) collisions and berthing manoeuvres, it is advised to perform research on the magnitude of a second impact during berthing manoeuvres as well.

- *Energy distribution during a brush collision*

During the verification of the created model a minor disturbance was found in the energy balance. Despite the extensive analysis of this inconsistency, a satisfactory explanation is not found yet.

- *Influence of the keel clearance on brush collisions*

In addition to the parameters that are varied within this study, it is advised to investigate the effect of the keel clearance on the movement of a vessel during a brush collision with a guide work. A limited keel clearance might generate higher drag forces, since there is less space for the water to flow underneath the colliding vessel, possibly resulting in a decrease of the contact forces during an impact.

References

Bratteland, E. (1988). *Advances in berthing and mooring of ships and offshore structures.* Trondheim: Kluwer Academic Publishers.

BSI. (n.d.). *About BSI*. Retrieved from www.bsigroup.com: https://www.bsigroup.com/en-GB/about-bsi/

CBS. (2018, January 16). *Binnenvaartschepen vervoerden 368 miljoen ton in 2017*. Retrieved from CBS.nl: https://www.cbs.nl/nl-nl/nieuws/2018/03/binnenvaartschepen-vervoerden-368-miljoen-ton-in-2017

De Boer & De Groot. (2017). *Sloop en herinrichting Zwettebrug*. Retrieved from www.deboerendegroot.nl: https://www.deboerendegroot.nl/projecten/sloop-en-herinrichting-zwettebrug

Delft hydraulics laboratory. (1978). *Belastingen op remmingwerken door schepen, varend onder een hoek tegen het remmingewerk.*

DET Norske Veritas. (2011). *Modelling and analsysis of marine operations.*

EAU 2012. (2012). *Recommendations of the Committee for Waterfront Structures Harbours and Waterways.* Ernst & Sohn.

EAU2012. (2012). *Recommendations of the Committee for Waterfront Strucutres Harbours and Waterways EAU 2012.* Hamburg: Ernst & Sohn.

Edmondson, B. (2017). *Collision Resistance of Fibre Reinforced Polymer Lock Gates.* Master thesis, Delft University of Technology, Faculty of Civil Engineering, Delft.

Findapile.com. (n.d.). *P-Y Curves*. Retrieved from findapile.com: http://www.findapile.com/p-y-curves/definition

Google Maps. (2018, July). *Google Maps*. Retrieved from Google.com: https://www.google.nl/maps/@53.185477,5.7750099,3a,75y,232.02h,77.81t/data=!3m6!1e1!3m4!1sD5jVrkjaXS88BMkbhC5nzQ!2e0!7i13312!8i6656

Jansen, J. (2019). *Ship collision on temporary structures.* Master thesis, Delft University of Technology, Faculty of Civil Engineering, Delft.

Joint Research Centre Eurocodes. (n.d.). *EN Eurocode Parts*. Retrieved from Joint research centre Eurocodes: https://eurocodes.jrc.ec.europa.eu/showpage.php?id=13

Jonkman, Steenbergen, Morales-Nápoles, Vrouwenvelder, & Vrijling. (2017). *Probabilistic Design: Risk and Reliability Analysis in Civil Engineering.* Delft.

Korevaar. (2018). *Berekeningsrapport DO Fuik buitenzijde Sluis 3 + Opstelplaats + Kegelwachtsplaats.*

NEN. (2011). *NEN-EN 1990+A1+A1/C2: Bais of structural design.* Delft.

NEN. (2011). *NEN-EN 1991-2+C1/NB.*

OCDI. (2019). *What is OCDI*. Retrieved from OCID.or.jp: http://ocdi.or.jp/en/about/whats-ocdi

Omroep Gelderland. (2019, December 5). *Onderzoek naar aanvaring Duitse tanker bij Zaltbommel*. Retrieved from Omroepgelderland.nl: https://www.omroepgelderland.nl/nieuws/2431814/Onderzoek-naar-aanvaring-Duitse-tanker-bij-Zaltbommel

PIANC. (2002). *Guidelines for the Design of Fender Systems.*

PIANC. (n.d.). *PIANC: the world association for waterborne infrastructure*. Retrieved from PIANC.nl: http://www.pianc.nl/

Rijkswaterstaat. (2017, April). Richtlijnen Ontwerp Kunstwerken (ROK) 1.4.

Rijkswaterstaat. (2017). *Richtlijnen Vaarwegen 2017.*

Rijkswaterstaat. (2018, October). *Vervanging en renovatie*. Retrieved from Magazinesrijkswaterstaat.nl: https://www.magazinesrijkswaterstaat.nl/zakelijkeninnovatie/2018/04/workshop-vervanging-en-renovatie

Rijkwaterstaat. (n.d.). *About us*. Retrieved from Rijkswaterstaat.nl: https://www.rijkswaterstaat.nl/english/about-us

Roubos, Groenewegen, & Peters. (2016). *Berthing velocity of large seagoing vessels in the port of Rotterdam.* Marine Structures.

Roubos, Groenewegen, Ollero, Hein, & Wal, v. d. (2018). Design values for berthing velocity of large seagoing vessels. *PIANC World Congress.* Panama: PIANC.

Ruigrok, J. (2010). *Laterally loaded piles.* Master thesis, Delft University of Technology, Faculty of Civil Engineering, Delft.

SBRCURnet. (2018). *Flexible Dolphins.* Delft.

SCIA. (2019). *SCIA Engineer Academische Software*. Retrieved from SCIA.net: https://www.scia.net/nl/education/scia-engineer-academische-software

SCIA Engineer. (n.d.). *Import, Export Handboek.*

Suribabu, C., Sabarish, R., Narasimhan, R., & Chandhru, A. (2011). *Backwater Rise and Drag Characteristics of Bridge Piers under Subcritical Flow Conditions.* European Water.

TheNavalArch. (n.d.). *DNV Ship Motion Calculator*. Retrieved from TheNavalArch.com: https://thenavalarch.com/software/marine-transportation/dnv-ship-motion-calculator/

Trelleborg. (2016, July 13). *News*. Retrieved from Trelleborg.com: https://www.trelleborg.com/en/marine-and-infrastructure/news--and--events/news/jul2016_4

Trelleborg. (2016, July 13). *Trelleborg launches extended fender system warranties in North America*. Retrieved from Trelleborg.com: https://www.trelleborg.com/en/marine-and-infrastructure/news--and--events/news/jul2016_4

Voorendt, & Molenaar. (2019). *Manual hydraulic structures.* Delft.

Vorm, v. d. (1993). *Vergelijking van Berekeningsmethoden voor Dukdalven.* Master thesis, Delft University of Technology, Faculty of Civil Engineering, Rotterdam.

Vrijer. (1978). *Belastingen op remmingwerken door schepen, varend onder een hoek tegen het remmingwerk.* Delft.

Vrijer, A. (1983). *Belasting op remmingwerken en dukdalven door varende schpen, deel III: rekenmethodes en procedure voor ontwerp.*

Wärtsilä Encyclopdiea of Marine Technology. (n.d.). *Ship motions*. Retrieved from Wartsila.com: https://www.wartsila.com/encyclopedia/term/ship-motions

Welleman. (2016). *Work, energy methods and influence lines.* Bouwen met Staal.

Welleman, H. (2019). *Lecture sheets: Work and Energy Methods*.

Wendel, K. (1956). *Hydrodynamic masses and hydrodynamic moments of inertia.* Hamburg: Jahrbuch der Schiffsbautechnischer Gesellschaft.

Appendix A: Soil representation for laterally loaded piles

The interaction of the soil with the vertical piles that are embedded in it has significant influence on the deformation of the structure, and thus its loadbearing capacity. Several methods exist for the description of laterally loaded piles embedded in soil. Four methods will be introduced in this appendix, being: Blum's method, P-y curves, Brinch Hansen & Ménard and Plaxis 3D. The models will be compared and finally, the modelling of the soil in the *Python model* is described.

A.1 Soil models

This subsection describes the four soil models that will be compared in section A.2. First Blum's method will be discussed, followed by P-y curves, Brinch Hansen & Ménard and Plaxis 3D respectively.

A.1.1 Blum's method

Blum's method is one of the oldest methods, and is still widely used because of its simplicity. The method is based on some crude simplifications, but still gives reasonable results. In order to clearly describe Blum's method, Figure A.1 is inserted in which the schematization is presented graphically.

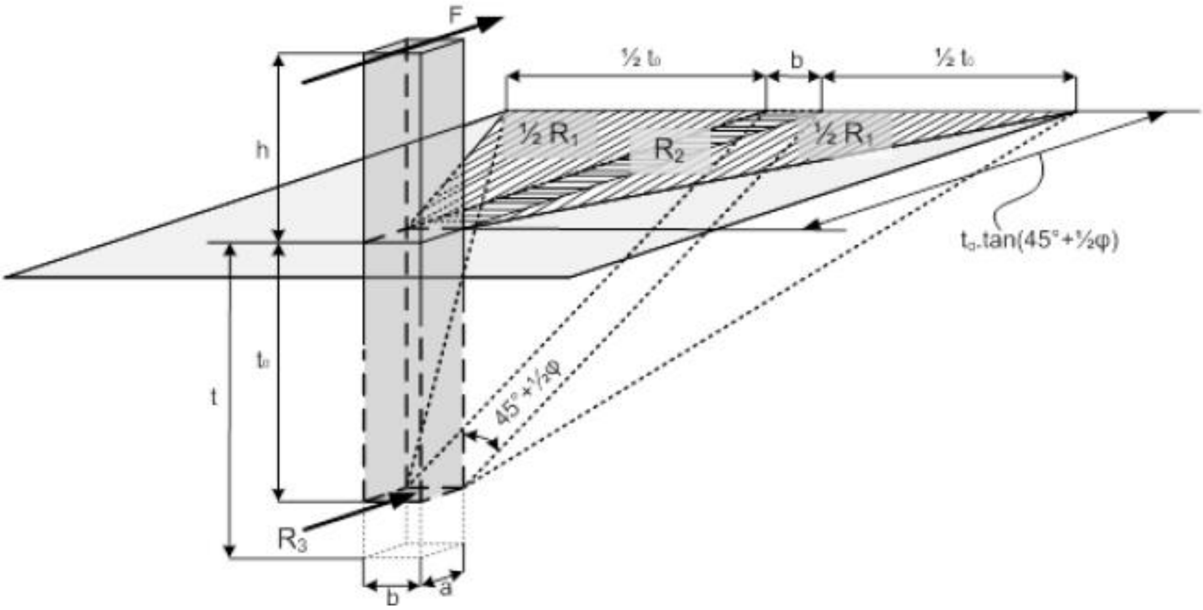


Figure A.1 Pile schematization Blum's method (Voorendt & Molenaar, 2019)

Blum's method assumes the vertical pile to be continuously supported over its embedded part and prevented from lateral deflections at a theoretical penetration depth t_0. At this depth the bending moment is assumed to be zero. Since the real length of the pile is equal to $1.2t_0$, it is possible that a shear force is present at the theoretical embedment depth.

As a result of the actual lateral loading force F, a soil wedge is mobilised. The shape of this wedge is depicted in Figure A.1. Response of the mobilised soil is assumed to be perfectly plastic. In order to maintain horizontal equilibrium of forces an additional force is introduced at the fictitious embedment depth. This force is considered as a resultant of the soil below t_0. In Figure A.1 this force is denoted as R_3 and the common name for this force is *Ersatzkraft*, which is the German word for 'substitute force'.

Based on simple mechanics it is possible to compute the deflections of the pile as a result of the forces working on it, given the zero deformation at depth t_0. Originally, the Blum method only allows for homogeneous soil and no cohesion. However, over the years several adaptions have been made. For example, Lackner has developed a more sophisticated method to determine the required embedment depth than the simple 1.2t_0. Engineering companies, among others Lievense|WSP, make use of internal spreadsheets that enable the user to input e.g. layered soils and sleeve friction. Also adaptions have been made in which the bending stiffness of the pile varies over its length.

A.1.2 P-y curves

A second method of determining the soil behaviour is with the aid of so-called p-y curves. In a p-y analysis the soil is represented by a set of discrete nonlinear springs. The nonlinear behaviour of the different springs is defined based on full-scale tests. The schematisation of this method is depicted in Figure A.2.

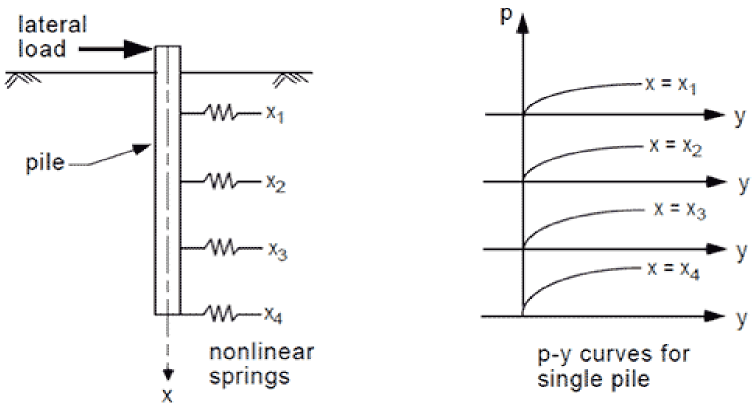


Figure A.2 Schematisation p-y curves (Findapile.com, sd)

Nowadays, the p-y curves are fully implemented in software programs as, e.g. *D-Pile Group* (previously called *MPile*). Alike Blum's method, also layered soils can be modelled with this method, as well as a varying bending stiffness over the height of the vertical piles.

A.1.3 Brinch Hansen & Ménard

The third method that is discussed within this thesis to describe the soil behaviour of a laterally loaded pile, is a combined method of Brinch Hansen and Ménard. In the Brinch Hansen & Ménard method the soil is also assumed to be continuously supported over its embedded depth, similar to Blum's method and p-y curves.

Distinction is made between the resistance of the soil and the stiffness of the soil. While the method of Ménard is used to describe the stiffness of the soil, Brinch Hansen is used to describe the failure behaviour of the soil.

Contrary to Blum, different soil layers can be used and it is also possible to include cohesion in the Brinch Hansen method. In the Brinch Hansen method a rotation point at depth D_r is defined by trial and error. The resultant force of the soil pressures above and underneath this rotation point must make moment equilibrium around the point of load application. The maximum horizontal load *H* is then equal to the difference in the resultant forces of the pressure areas.

The soil pressure is dependent on various coefficients, that take into account the type of soil, the depth and whether the soil pressure is active or passive. Schematisation of the Brinch Hansen method is given in Figure A.3.

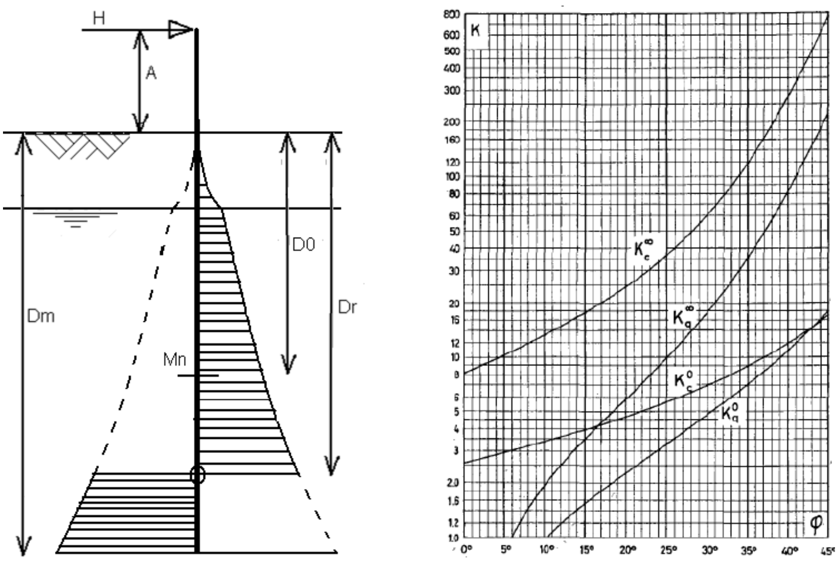


Figure A.3 Brinch Hansen method (Ruigrok, 2010)

Since the Brinch Hanson method itself is unable to compute deflections of a pile, this method is commonly accompanied with Ménard.

The stiffness of the soil is determined by the so-called *Ménard pressure meter test*, also called *Roctest*. Layout of this test is depicted in Figure A.4a and typical results of such test is given in Figure A.4b.

The Roctest consists of a cylindrical probe that is pushed into the soil. Once the cylinder is in the soil, a gas or fluid is being pressed into the cylinder. As a result, the cylinder tries to expand laterally, pushing the surrounding soil away. By means of plotting the volume of the cylindrical probe versus its inside pressure, it is possible to determine the stiffness of the soil.

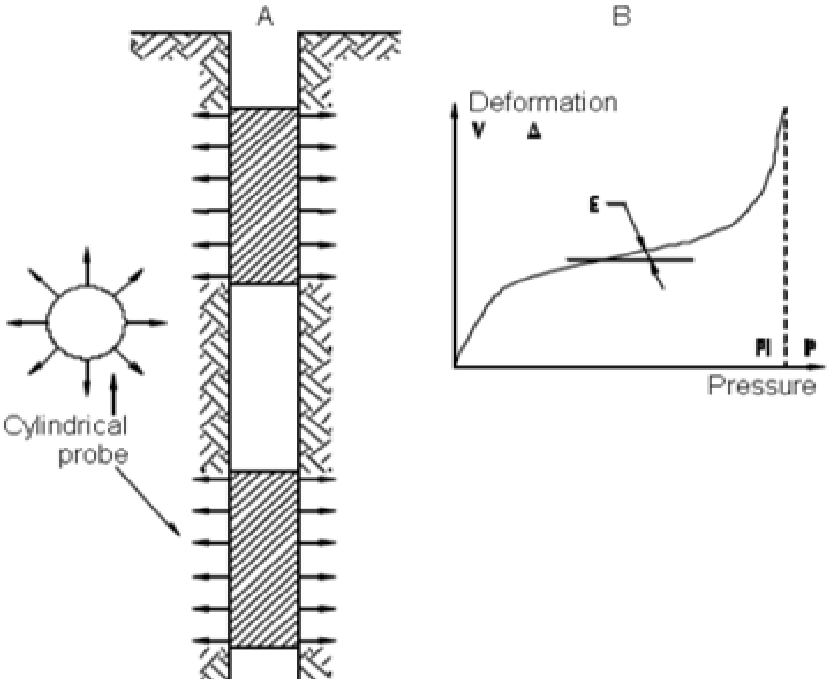


Figure A.4 Ménard pressure meter test: a) test setup b) example of test results (Ruigrok, 2010)

In, e.g. the computer software *D-Sheet Piling* (previously called *MSheet*), the results of a Roctest can be used as input. Since pressure meter test are not commonly executed in the Netherlands, contrary to cone penetration tests (CPT), conversion tables have been created to convert CPT results to pressure meter test results. Within this method the soil is modelled with bilinear behaviour. Then, based on the stiffness by Ménard and the strength by Brinch Hansen, a deflection curve for the pile is created, as well as a force distribution. Variation of the moment of inertia over the length of the pile is in principle not possible.

A.1.4 Plaxis 3D

Finally, Plaxis can be mentioned as a tool to model the soil behaviour. Plaxis 3D is a finite element method (FEM) software. The possibilities of Plaxis for the modelling of soil are enormous. Drawback of the wide variety of possibilities is that it requires the user to determine a very large number of input parameters. Setting up a model can then become rather time consuming and, depending on the model, the calculation time can become a troublesome factor, since the computation may take up to several hours.

A.2 Comparison soil models

Among others, J.A.T. Ruigrok (2010) and the CUR (2018) have performed a study to compare different methods for soil behaviour around laterally loaded piles. Comparison is done on the accuracy, the applicability and the speed of the methods. In both studies full scale test results have been used to compare the used soil model with results in practice.

Due to the automation of Blum's method, P-y curves and Brinch Hansen & Ménard by means of spreadsheets and software programs, they have become very widely applicable. On top of that, the calculation time is limited due to the simplicity of these methods. With respect to Plaxis it can be mentioned that the time consuming input procedure starts to pay off as the complexity of the soil situation increases.

It is difficult to unambiguously express the accuracy of the different methods. Local circumstances have significant influence on the results, e.g. drained soils versus undrained soils and cohesive soils versus cohesionless soils. Nonetheless some general recommendations can be made. (Ruigrok, 2010)

Blum's method is very suitable for the early design stage. It is then advised to use the adapted Blum method as described in the current version of the German *Spundwand Handbuch*, in order to include layered soils and cohesive soils in the model.

D-Pile Group (p-y curves) is a suitable method in a later design stage. After the first coarse design choices have been made, this program can be used to optimize the design. It is also possible to investigate the behaviour of pile groups or the influence of perturbations in the soil.

D-Sheet Piling (Ménard) is preferred as a design tool in the early stages. A rough first design can be made using this program. Advantage of this method is that even with limited soil data, this software can still give useful results.

For the soil model that is used in the new Python model, it is chosen to use Blum's method for the soil behaviour. The most important reason for this choice is the simplicity of this procedure. Since the model makes use of a large number of time steps in order to describe the problem in the time domain, it would require extensive computational capacity and calculation time to perform a fully nonlinear analysis in every time step. In the next subsection the modelling of the soil in the new method is described more elaborately.

A.3 Used soil model in the *Python model*

Based on the theory of Blum and recommendations in the *Spundwand Handbuch* an Excel spreadsheet has been created at Lievense. This spreadsheet computes the deflection and internal forces for a mooring pole. As can be seen from Figure A.5 and Figure A.6, all input boxes are made yellow. First, a soil profile up to eight layers can be defined. Each level is defined by its top and bottom level, volumetric weight, friction angle and cohesion. Also the undrained shear strength f_{undr} and the shaft friction f_s can be given as input. Based on the applicable reliability class, the corresponding partial safety factor for the soil parameters can be used. It is also possible to use soil-filled piles (not depicted in Figure A.5) with a user-defined volumetric weight.

GROND en BODEM		Grenstoestand / Reliability Class:	RC1			partiële veiligheden:			op tan ϕ =	1,15
									op C' =	1,15
Bodem:	nivo tpv. paal		m. tov. NAP						op f_{undr} =	1,50
helling rechts	1 :		tot		m.NAP				op f_s =	1,15
helling links	1 :		tot		m.NAP				op γ' =	1,00
Grond:	nivo [m. tov. NAP]	γ' [kN/m³]	φ [°]	C' [kN/m²]	f_{undr} [kN/m²]	f_s [kN/m²]	δ/φ	δ_0/φ tpv. teen	$K_{\gamma ph}$	$K'_{\gamma ph}$ t.p.v. C:
LAAG 1.									0,00	1,00
onderkant							+ = ↑	+ = ↓		
LAAG 2.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 3.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 4.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 5.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 6.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 7.	-21,00								0,00	0,00
onderkant							+ = ↑	+ = ↓		
LAAG 8.									0,00	0,00
onderkant	-21,00						+ = ↑	+ = ↓		

Figure A.5 Input soil properties model

Subsequently, the properties of a single pile need to be determined. Since the adaption of Blum's method allows for a varying bending stiffness over the length of the pile, the spreadsheet also allows for a different wall thickness for different sections. Furthermore, the steel quality and the yield stress needs to be inputted, as well as the corrosion rate. The input sheet for the pile properties is given in Figure A.6.

GEGEVENS MEERPAAL OF -STOEL						
diameter	D =		inch =	0	mm	
aantal palen	n =		stuks	extra dikte boorpaal:	0	
effektieve breedte	B =		m.	0,00	m.	
top meerpaal	h =		m. NAP	paalkop ingeklemd?	nee	
inheidiepte	t =		m. NAP	G_{tot} =	0,00	ton

SEKTIES: nr.	nivo t.o.v. NAP [m]	staalkwaliteit [API]		vloeigrens [N/mm²]	wanddikte [mm]	corrosie [mm/jaar]	vulling?
	0,00						
1. onderkant		X		0			empty
2. ######		X		0			empty
3. ######		X		0			empty
4. ######		X		0			empty
5. ######	-5,00	X		0			empty
6. ######	0,00	X		0			empty
7. ######	0,00	X		0			empty
8. ######	0,00	X		0			sand
9. ######	-17,00	X		0			sand
10.	0,00	X		0			loose sand

Figure A.6 Input pile properties model

Finally, there is also a worksheet in which the loads can be specified (see Figure A.7). In this sheet the lifetime of the structure is specified. Also the mooring line forces and the kinetic berthing energy need to be specified here, as well as their point of application.

BELASTINGEN

Leeftijd (corrosietijd):	T =		jaar
Vereiste veiligheden:			
op vloeispanning:	γ_{vl} =		
op plooispanning:	γ_{pl} =	0,00	$=\gamma_{vl}$
op stootenergie:	γ_E =		
op troskracht:	γ_F =		

Stootenergie:	E =		kNm =>	E_d =	0 kNm/meerpaal
aangrijpingspunt:	NAP		m.		
wrijving op schort:	f =		(omhoog)		

Troskracht:	F =		kN =>	F_d =	0 kN/paal
aangrijpingspunt:	NAP		m.		
	hoek =	0	graden	(omhoog)	

Capaciteit: aangrijpingspunt	NAP	0,00	m.	
	hoek =	0	graden	(omhoog)
belastingrichting:		naar links		

Kopmoment:	M =	0	kNm	*Inklemming paalkop?*	
aangrijpingspunt:		2,50	m. NAP	*Gewenste hoekverdraaiing =*	*mrad*

Figure A.7 Input of the load

As soon as the input properties have been defined, an iteration takes place within the spreadsheet. The total amount of berthing energy needs to be converted to potential energy in the structure and the soil, leading to a corresponding design force. Since the spreadsheet is originally created to determine the behaviour of a mooring pole, an implementation has been made that enables the use of the spreadsheet for guide works as well.

Based on the deflections that a single mooring pole undergoes, a configuration of a linear elastic translational spring and a linear elastic rotational spring that represent the behaviour of the soil is made. The set of springs is modelled at a fictitious depth *t* underneath the bed level and the stiffnesses of the springs are chosen such that the deflection of the point of application of the load has the same value as the deflection curve of the pile. The schematization of the vertical pile with linear spring supports is given in Figure A.8.

Also Table A.1 is included in which the spring properties and the corresponding modelling depth of the soil supporting the guide works in front of the Beatrixsluis are presented, in order to give an indication of the order of magnitude for the stiffness of both springs. On top of that, values for the design force for a corresponding berthing energy are given as an output of the spreadsheet and included in the table.

Table A.1 Soil spring properties Beatrixsluis (Korevaar, 2018)

Energy [kNm]	**Force [kN]**	**Rotational spring stiffness [MNm/rad]**	**Translational spring stiffness [MN/m]**	**Depth below bed level [m]**	**Level relative to N.A.P [m]**
300	1442	870	45	5.4	-12.7
200	1237	930	52	5.1	-12.4
180	1188	940	54	5.0	-12.3
150	1108	940	55	4.8	-12.1
120	1016	980	61	4.7	-12.0
70	825	1050	71	4.3	-11.6
30	590	1130	86	3.7	-11.0
10	384	1310	123	3.1	-10.4
2	198	1580	191	2.4	-9.7
1.0	148	1670	220	2.1	-9.4
0.5	110	1840	272	1.9	-9.2
0.1	55	2140	385	1.4	-8.7

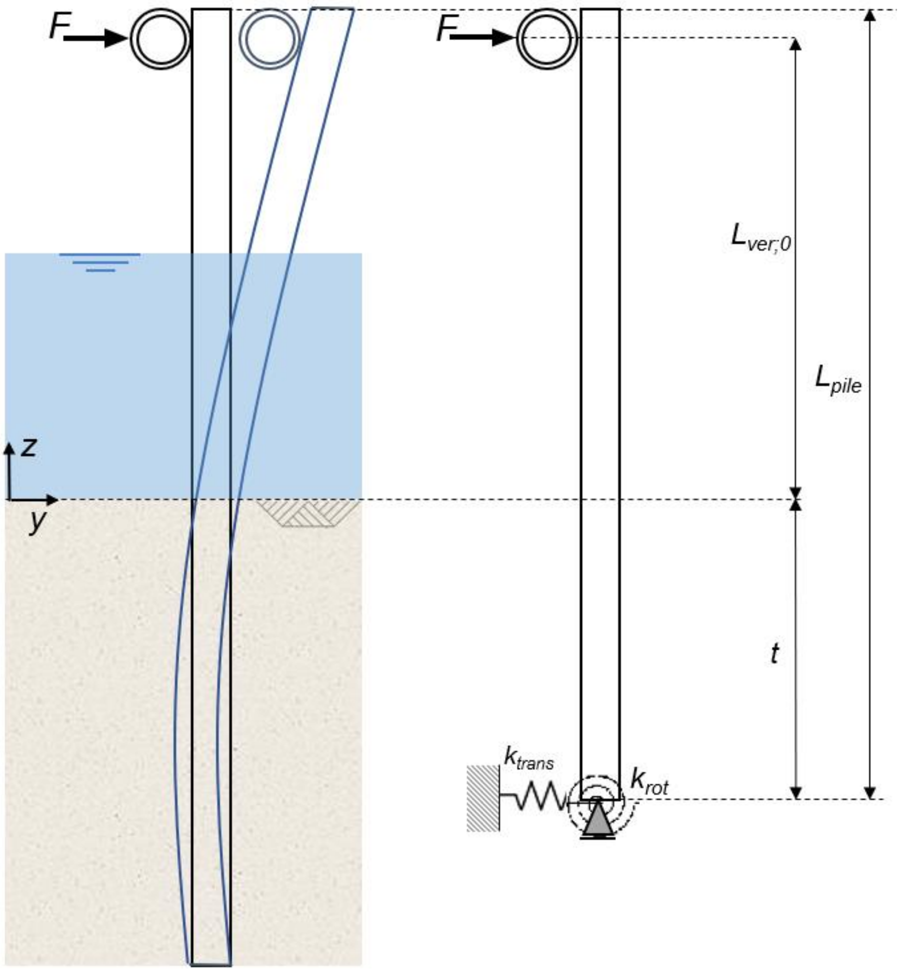


Figure A.8 Modelling of the soil as linear elastic springs

With the aid of the two linear elastic springs it is possible to model a framework, i.e. the guide work, in a FEM software with its complete geometry and the soil interaction.

With the aid of the data given in Table A.1, the graphs in Figure A.9 and Figure A.10 can be constructed. Figure A.9 and Figure A.10 give the behaviour of the springs as a function of the applied force *F* and the depth at which these springs should be modelled. It is important to emphasize that the springs itself are linear elastic. The value of the spring stiffness, however, is nonlinearly dependent on the applied force *F*. Plotting a trendline through the data, as done in Figure A.9 and Figure A.10, makes it possible to find the correct spring stiffness and depth for arbitrary values of *F*.

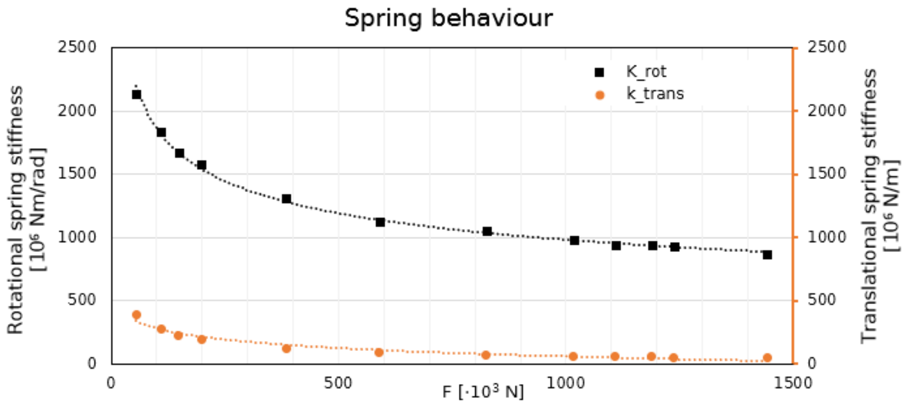


Figure A.9 Spring stiffness as a function of the applied force

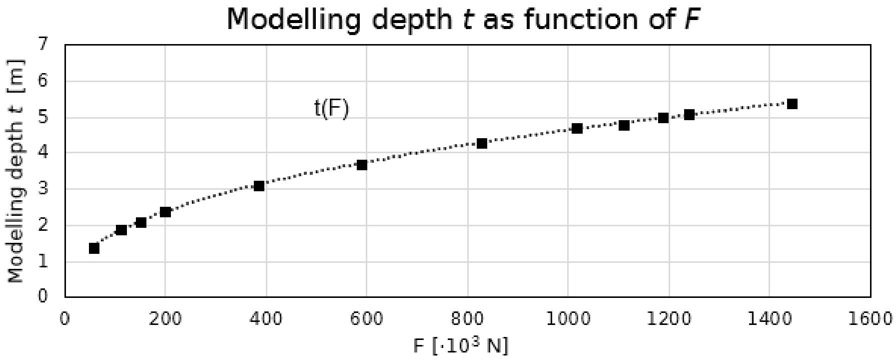


Figure A.10 Fictitious modelling depth t as a function of the applied force

Appendix B: Movement of a vessel during a brush collision

The new model describes the movement of the ship (and thus the contact point) in the time domain and gives a derivation for possible second berthing impacts. Definitions with respect to the movement of the vessel are discussed within this chapter, as well as the derivation itself and the influence of the water on the movement. Finally, a verification method for this newly developed model is introduced.

B.1 Definitions

This subsection gives an overview of the different symbols and definitions that are used in describing the horizontal movement of the vessel. First of all, it can be mentioned that the x-axis is aligned with the longitudinal direction of the structure in undeformed condition. The y-axis is perpendicular to the x-axis, as indicated in Figure B.1, leading to a positive rotation θ for the counterclockwise direction.

The geometry of the ship is simplified to a rigid rectangle with dimensions B_{ship} and L_{ship} for the width and length of the ship respectively. The mass of the ship is denoted with *m* and the centre of gravity is denoted with *C*. Other notable points of the ship are the location where the bow or the stern can make contact with the structure; *V* and *A* are used for these positions respectively.

The x- and y- position of point V, C and A are indicated with the letter of the axis, combined with a subscript that contains the name of the point. The derivative with respect to time, ∂/∂t, is commonly denoted by a ‘over-dot’. Since velocity is defined as the first derivative of the displacement with respect to time, and acceleration is defined as the second derivative of the displacement with respect to time, this leads to e.g. $\dot{x}_C$ and $\ddot{x}_C$ as the notation for the mass centre’s velocity and acceleration respectively.

Furthermore, the distances *a* and *b*, being the horizontal and vertical distances from the centre of gravity to contact point V respectively can be distinguished. Additionally, distances *c* and *d* are introduced as the vertical and horizontal distance respectively, between point A and point C. Figure B.1 is inserted to present the definitions graphically.

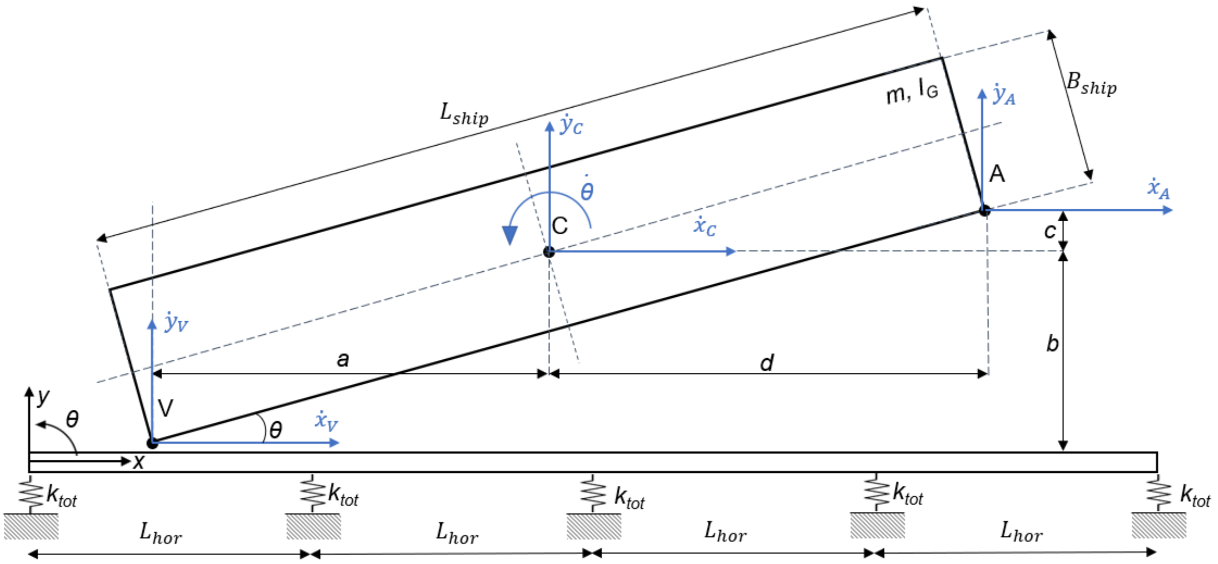


Figure B.1 Definitions of a moving vessel in the horizontal plane

B.2 Analytical derivation

In this subsection the description of the movement of the vessel is given. The derivation starts with the exchange of momentum between the vessel and the guide work as presented in the report by A. Vrijer for the Delft Hydraulics Laboratory (Delft hydraulics laboratory, 1978). Subsequently, the derivation in this thesis is discretised, contrary to the derivation by A. Vrijer.

In the contact process several distinct moments in time can be observed. The analysis starts at $t=t_0$ at which the very first contact takes place. Then, during an infinitesimal small time interval that ends at $t=t_{0'}$, the structure is accelerated and the ship's bow is decelerated. After $t_{0'}$ the bow of the vessel and the structure move simultaneously in y-direction, while the contact point moves along the structure in x-direction. The moment in time where the deflection in y-direction is at its maximum, is denoted with $t=t_1$. Finally, at $t=t_2$ the bow is not in contact anymore with the structure. Please note that the first contact that takes place, is with the bow of the ship.

Since the vessel can contain an angular velocity, the translational velocity of the bow and the stern differ from the translational velocity of the mass centre. For Figure B.1, the velocities of point V and C can be coupled by Equation (B.1) to Equation (B.4).[4]

$$\dot{x}_V = \begin{cases} \dot{x}_c + b\,\dot{\theta} & y_V < y_C \\ \dot{x}_c - b\,\dot{\theta} & y_V > y_C \end{cases} \tag{B.1}$$

$$\dot{y}_V = \dot{y}_C - a\,\dot{\theta} \tag{B.2}$$

$$\dot{x}_A = \begin{cases} \dot{x}_C - c\,\dot{\theta} & y_A > y_C \\ \dot{x}_C + c\,\dot{\theta} & y_A < y_C \end{cases} \tag{B.3}$$

$$\dot{y}_A = \dot{y}_C + d\,\dot{\theta} \tag{B.4}$$

In these relations *a, b, c* and *d* are defined as:, see also Figure B.1.

$$a = \begin{cases} 0.5\big(L_{ship}\cos(|\theta|) - B_{ship}\sin(|\theta|)\big) & \theta > 0 \\ 0.5\big(L_{ship}\cos(|\theta|) + B_{ship}\sin(|\theta|)\big) & \theta < 0 \end{cases} \tag{B.5}$$

$$b = \begin{cases} 0.5\big(L_{ship}\sin(|\theta|) + B_{ship}\cos(|\theta|)\big) & \theta > 0 \\ 0.5\big(B_{ship}\cos(|\theta|) - L_{ship}\sin(|\theta|)\big) & \theta < 0, y_V < y_C \\ 0.5\big(L_{ship}\sin(|\theta|) - B_{ship}\cos(|\theta|)\big) & \theta < 0, y_V > y_C \end{cases} \tag{B.6}$$

$$c = \begin{cases} 0.5\big(L_{ship}\sin(|\theta|) - B_{ship}\cos(|\theta|)\big) & y_A > y_C \\ 0.5\big(B_{ship}\cos(|\theta|) - L_{ship}\sin(|\theta|)\big) & y_A < y_C \\ 0.5\big(L_{ship}\sin(|\theta|) + B_{ship}\cos(|\theta|)\big) & \theta < 0 \end{cases} \tag{B.7}$$

$$d = \begin{cases} 0.5\big(L_{ship}\cos(|\theta|) + B_{ship}\sin(|\theta|)\big) & \theta > 0 \\ 0.5\big(L_{ship}\cos(|\theta|) - B_{ship}\sin(|\theta|)\big) & \theta < 0 \end{cases} \tag{B.8}$$

The analysis starts at the moment when the ship transfers part of its momentum to the structure. In order to avoid a lengthy derivation of this momentum transfer, it is important to realise that brush collisions are considered. Consequently, the angles of collision are small, resulting in y_V to be smaller than y_C at $t=t_0$ and $t=t_{0'}$. This leads to the fact that during the very first moment of contact Equations (B.2) and Equation (B.1) can be rewritten to:

$$\dot{x}_c = \dot{x}_V - b\,\dot{\theta} \tag{B.9}$$

$$\dot{y}_C = \dot{y}_V + a\,\dot{\theta} \tag{B.10}$$

Also a friction coefficient μ is introduced. Multiplication of the normal force with the friction coefficient gives the longitudinal friction force. In the time from t_0 to $t_{0'}$ the vessel and the structure gain a new joint velocity. Since the momentum before and after the first impact

[4] In all cases it is assumed that θ remains smaller than 90°

should be conserved, Equation (B.11) to Equation (B.13) can be formulated for the vessel, in which *S* is the momentum. It needs to be mentioned that the ship initially does not have an angular velocity since a brush collision is considered.

$$\mu\, S = m\,(\dot{x}_c(t_{0'}) - \dot{x}_c(t_0)) \tag{B.11}$$
$$S = m\,(\dot{y}_C(t_{0'}) - \dot{y}_C(t_0)) \tag{B.12}$$
$$S(\mu\, b - a) = I_G \dot{\theta}(t_{0'}) \tag{B.13}$$

In the y-direction it holds for the structure that:

$$-S = M\, \dot{y}_V(t_{0'}) \tag{B.14}$$

By rewriting Equation (B.11) to Equation (B.13), it is possible to obtain expressions for the velocities of the vessel directly after the first impact. It has to be noted that the mass *m* for the vessel is equal to the mass of the ship plus an added water mass. Similarly, in the inertia of the ship I_G also the dynamic effects of the water are included. Elaboration on the influence of the water is done section B.3.

Since not the whole structure is equally accelerated the mass *M* of the structure is taken as $M = \eta\cdot$(total mass structure). Elaboration on the scaling factor η is given in section C.3.

$$\dot{y}_C(t_{0'}) = \frac{S}{m} + \dot{y}_C(t_0) \tag{B.15}$$
$$\dot{\theta}(t_{0'}) = \frac{S(\mu\, b - a)}{I_G} \tag{B.16}$$
$$\dot{y}_V(t_{0'}) = \frac{-S}{M} \tag{B.17}$$

Equation (B.15), Equation (B.16), Equation (B.17) and Equation (B.10) can be combined, resulting in:

$$\frac{S}{m} + \dot{y}_C(t_0) = -\frac{S}{M} + a\left(\frac{S(\mu\, b - a)}{I_G}\right) \tag{B.18}$$

Rewriting of Equation (B.14) leads to a closed expression for *S*:

$$S = \frac{-\dot{y}_C(t_0)}{\frac{1}{m} + \frac{1}{M} - \frac{a\,(\mu b - a)}{I_G}} \qquad \text{[kg m/s]} \tag{B.19}$$

The translational velocities of the bow and the mass centre of the ship at $t = t_{0'}$ can now be determined, as well as the gained rotational velocity.

$$\dot{\theta}(t_{0'}) = \frac{S(\mu b - a)}{I_G} \tag{B.20}$$

$$\dot{y}_V(t_{0'}) = \frac{-S}{M} \tag{B.21}$$

$$\dot{y}_C(t_{0'}) = \frac{-S}{M} - a\, \dot{\theta}(t_{0'}) \tag{B.22}$$

$$\dot{x}_C(t_{0'}) = \frac{\mu\, S}{m} + \dot{x}_C(t_0) \tag{B.23}$$

$$\dot{x}_V(t_{0'}) = \frac{\mu\, S}{m} + \dot{x}_C(t_0) + b\, \dot{\theta}(t_{0'}) \tag{B.24}$$

In order to model the problem in the time domain after $t = t_{0'}$, a discretisation is made. The analysis starts at time $t = t_{0'}$. At this moment in time the structure is not deflected yet and the vessel has both an angular and a translational velocity, see Figure B.2.

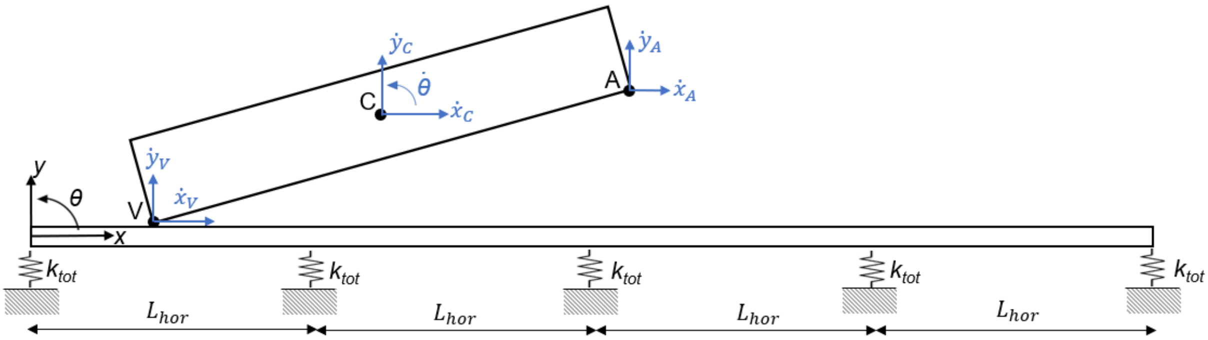


Figure B.2 Situation at t=$t_{0'}$

Within each discretisation step all accelerations are assumed to be constant, and the velocities vary thus linearly. This means that at the end of the first timestep point V is located within the structure. Since this is not possible in reality, the structure needs to deform accordingly, see Figure B.3. The displacement of point V can thus be regarded as a prescribed deformation of the structure.

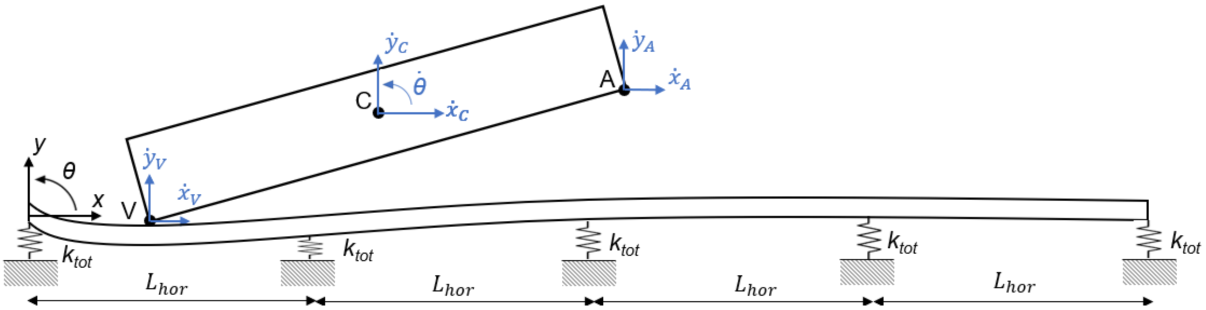


Figure B.3 Situation after the first timestep

A prescribed deformation in the negative y-direction is associated with a corresponding reaction force in the negative y-direction. According to Newton's third law of motion the structure generates an equally large force on the vessel. On top of that a friction force, F_x = μ F_y, is also applied on the vessel, see Figure B.4.

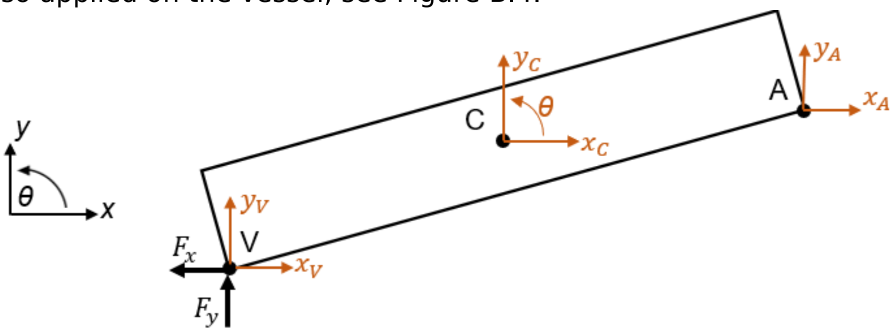


Figure B.4 Forces generated by the structure acting on the vessel

Application of Newton's second law of motion leads to Eq. (B.25) and Eq. (B.26).

$$F_y = m_y \, \ddot{y}_C \tag{B.25}$$

$$F_x = -m_x \, \ddot{x}_C = -\mu \, F_y \tag{B.26}$$

It has to be noted that the mass in x-direction and the mass in y-direction are not the same. Reason for this difference is the virtual added mass that is generated by the movement of the water around the vessel, which is not the same for x- and y-direction. Elaboration of the influence of the water is given in section B.3.

For the rotation of the vessel it holds that:

$$\sum M|_C = I_G \, \ddot{\theta} \tag{B.27}$$

Or equivalently:

$$-F_y\, a - F_x\, b = I_G\, \ddot{\theta} = -F_y(a + \mu\, b) \tag{B.28}$$

The force is assumed to be constant within one timestep, which can be justified by using sufficiently small timesteps. This means that at the end of each time step it holds that:

$$\theta_i \;= \theta_{i-1} \;+ \dot{\theta}_{i-1}\Delta t \;+ \ddot{\theta}_{i-1}(\Delta t)^2 \tag{B.29}$$
$$x_{C;i} = x_{C;\,i-1} + \dot{x}_{C;\,i-1}\Delta t + \ddot{x}_{C;i-1}(\Delta t)^2 \tag{B.30}$$
$$y_{C;i} = y_{C;\,i-1} + \dot{y}_{C;\,i-1}\Delta t + \ddot{y}_{C;i-1}(\Delta t)^2 \tag{B.31}$$

The new value for θ also leads to a new value for the distances *a*, *b*, *c* and *d*. Combining the updated position of the mass centre of the ship with the geometrical relations, gives the position of A and V at the end of the time step.

$$x_V = x_C - a \tag{B.32}$$
$$y_V = \begin{cases} y_C + b & y_C < y_V \\ y_C - b & y_C > y_V \end{cases} \tag{B.33}$$
$$x_A = x_C + d \tag{B.34}$$
$$y_A = \begin{cases} y_C + c & y_C < y_A \\ y_C - c & y_C > y_A \end{cases} \tag{B.35}$$

The value for y_V can now again be used as a prescribed deformation for the next timestep, while the value for x_V is used to obtain the position at which the prescribed deformation is applied. Eventually, when y_V becomes larger than zero again, the first contact has ended. The vessel then moves as a free body in the water, until the stern collides with the structure.

In case of a collision of the stern with the structure, the same procedure is followed as when the bow is in contact with the structure; by means of conservation of momentum the structure is accelerated, and y_A functions as a prescribed deformation at the location x_A. Schematically, the procedure of analysing the collision is given in Figure B.5.

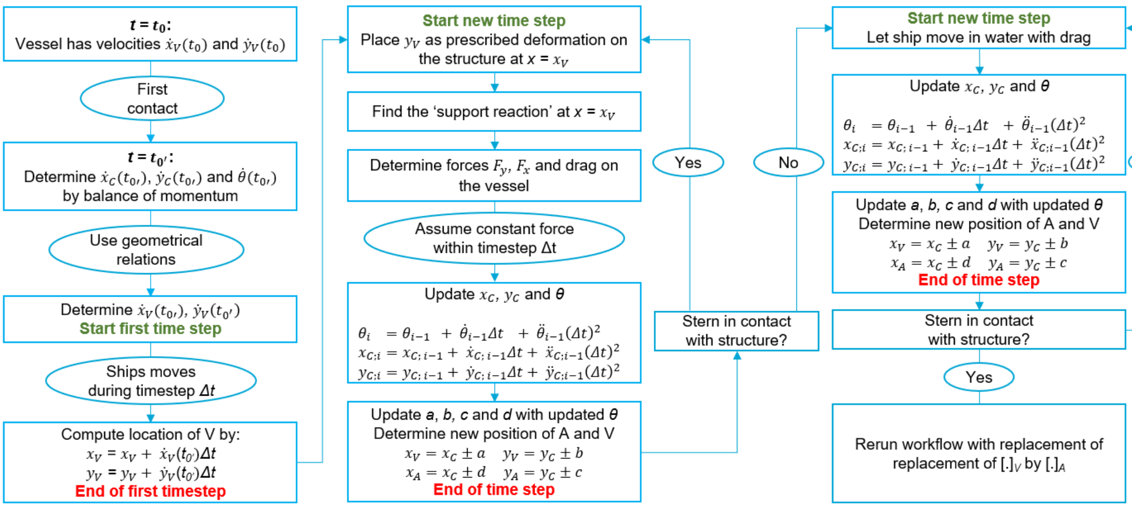


Figure B.5 Workflow analysis in the time domain

B.3 Influence of the water

Since the ship is surrounded by water, also the inertial effects of the water need to be taken into account. Added mass effects for translation and rotation of the ship are treated separately in section B.3.1 and B.3.2 respectively.

B.3.1 Translation

In the traditional approach the C_m factor is introduced for this phenomenon. Elaboration on this factor is given in section 2.3.6. The virtual mass factor applies for the translational movement of the vessel. For longitudinal approaches a C_m factor of 1.1 is recommended. This means an increase of 10% of the total mass of the vessel.

In order to properly decouple the movement in the x- and y-direction and the rotation, the C_m factor is decomposed in two components as well. This leads for the x- and y- direction respectively, to:

$$C_{m,x} = 1 + C_m \cos(|\theta|) \tag{B.36}$$
$$C_{m,y} = 1 + C_m \sin(|\theta|) \tag{B.37}$$

This means that Equation (B.25) and Equation (B.26) and Equation (B.38) and Equation (B.39) are equivalent to each other. It is to be noted that the value of $C_{m,x}$ and $C_{m,y}$ are updated every timestep step since the angular position θ is subjected to change.

$$F_x = -C_{m,x}\, m\, \ddot{x}_C = -\mu\, F_y \tag{B.38}$$

$$F_y = C_{m,y}\, m\, \ddot{y}_C \tag{B.39}$$

B.3.2 Rotation

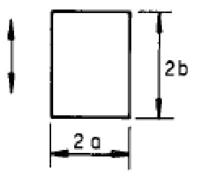
Not only for the translation, but also for the rotation (yawing) of the vessel the water plays a significant role. For the rotation the calculation for the added mass factor is based on the recommended practice for the modelling and analysis of marine operations by the Norske Veritas. (DET Norske Veritas, 2011)

In this report of the Norske Veritas, analytical values for the added mass coefficients are presented for both 2D and 3D objects, for both translation and rotation. In the tables in the appendix of the report the added mass factors for, among others, a rectangle are given, see Table B.1.

Straightforward application of the last column of this table leads to an overestimation of the magnitude of the added mass moment of inertia. This overestimation comes from the fact that the table originally is meant for long cylinders in infinite fluid. Therefore, it is chosen to apply a slight modification of Table B.1 that gives results that are more in accordance with experimental research. (Wendel, 1956)

Table B.1 Added mass factor (DET Norske Veritas, 2011)

Table A-1 Analytical added mass coefficient for two-dimensional bodies, i.e. long cylinders in infinite fluid (far from boundaries). Added mass (per unit length) is $A_{ij} = \rho C_A A_R$ [kg/m] where A_R [m²] is the reference area

Section through body		*Direction of motion*	C_A	A_R	*Added mass moment of inertia* [(kg/m)*m²] $\beta_1 \rho \pi a^4$ or $\beta_2 \rho \pi b^4$		
					a/b	β_1	β_2
	a / b = ∞	Vertical	1.0	πa^2	0.1	-	0.147
	a / b = 10		1.14		0.2	-	0.15
	a / b = 5		1.21		0.5	-	0.15
	a / b = 2		1.36		1.0	0.234	0.234
	a / b = 1		1.51		2.0	0.15	-
	a / b = 0.5		1.70		5.0	0.15	-
	a / b = 0.2		1.98		∞	0.125	-
	a / b = 0.1		2.23				

The vessel is schematised as a rectangle, which is divided in a sufficiently large total number of N segments (N=50), see Figure B.6. When the vessel starts to rotate, every segment is thought to translate in the direction indicated in the figure. Furthermore, it can be mentioned that the length and the mass of a segment are equal to the total length and total mass respectively divided by the number of segments.

For every segment the added mass in terms of translation is determined using Table B.1. It has to be noted that direction of movement is not in the longitudinal direction (x-direction) of the ship anymore, but in its transversal direction (y-direction), see Figure B.6. This means that *a* in Figure B.7 is equal to the draught of the ship, and *b* corresponds with half the width of the ship.

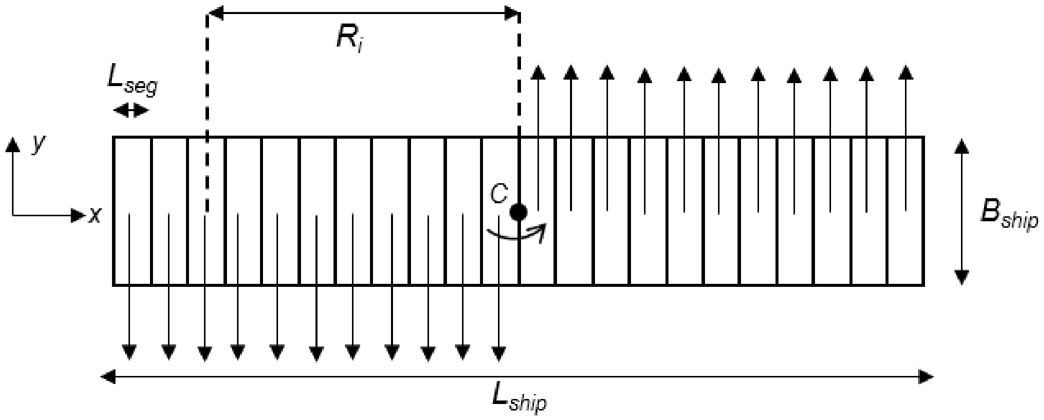


Figure B.6 Added rotational mass ship, top view

Since Table B.1 presumes totally submerged shapes, a modification is necessary to make the table applicable for a vessel. This is done by scaling the density of the water by the actual part of the ship that is underneath the water level.

$$\rho_{model} = \frac{d}{H_{ship}} \rho_{water} \quad \text{(B.40)}$$

where:

ρ_{model}	Modelled density	[kg/m³]
d	Draught	[m]
H_{ship}	Total height of the vessel	[m]
ρ_{water}	Density of the water	[kg/m³]

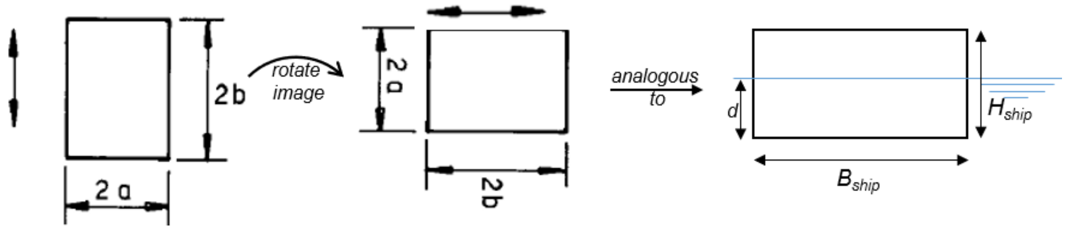


Figure B.7 Determination C_A factor

For the added mass of a segment it holds:

$$A_{ij} = \rho_{model}\, C_A\, A_R \quad \text{(B.41)}$$

where:

A_{ij}	Added mass per unit length	[kg/m]
ρ_{model}	Density in the model	[kg/m³]
C_A	Dimension dependent coefficient (see Table B.1 and Figure B.7)	[-]
A_R	Reference area $= \pi\, d^2$	[m²]

After the mass and the added mass of a segment are known, one can translate the translational movement to a rotational movement by means of the parallel axis theorem. Result of this transformation gives the rotational moment of inertia, including added mass effects, see Equation (B.42)

$$I_G = \sum_{i=1}^{N_{seg}} (m_{seg} + A_{ij}\, L_{seg})\, R_i^2 \tag{B.42}$$

where:

I_G	Inertia vessel included added mass effects	[kg m^2]
N_{seg}	Total number of segments	[-]
m_{seg}	Mass of a segment = (mass vessel) / N_{seg}	[kg]
A_{ij}	Added mass per unit length	[kg/m]
L_{seg}	Length of a segment = L_{ship} / N_{seg}	[m]
R_i	Distance COG[5] segment to COG ship	[m]

B.4 Damping

Implementation of the added mass factors does not automatically lead to incorporation of damping in the model. In order to take the damping into account in the new model, experimental data is required, which is very costly. Nonetheless an estimate can be made based on basic dynamics. Again, distinction is made between rotation and translation. First, the translation is discussed in section B.4.1 and subsequently, the damping for the rotation is elaborated in section B.4.2.

B.4.1 Translation

In case of a translation movement of a vessel, a vessel experiences drag. Drag is a form of resistance and is influenced by numerous factors, including the shape and velocity of an object. The drag force works in the opposite direction of the movement, resulting in a decrease of the velocity, and thus leading in a decrease of energy in the system. Within this report translational drag is only considered in the transversal direction (y-direction) (see Figure B.8), since this is the direction in which the structure needs to absorb the energy of the vessel. The formula for the translational drag force, $F_{drag,trans}$, that is generated by the water, is given in Equation (B.43).

$$F_{drag,trans} = -\frac{1}{2} \rho\, C_W\, \dot{y}\, |\dot{y}|\, A \tag{B.43}$$

where:

$F_{drag,trans}$	Drag force	[N]
ρ	Density of the medium = ρ_{water}	[kg/m^3]
C_w	Drag coefficient	[-]
$\dot{y}$	Translational velocity COG	[m/s]
A	Frontal area = $L_{ship} \cdot draught$	[m^2]

In accordance with Suribabu et al. (2011) the drag coefficient C_w is set equal to 2.0. Furthermore, it needs to be noted that the drag force is in the opposite direction of the movement of the ship, hence the minus sign in Equation (B.43). (Suribabu, Sabarish, Narasimhan, & Chandhru, 2011)

[5] COG stands for *Centre Of Gravity*

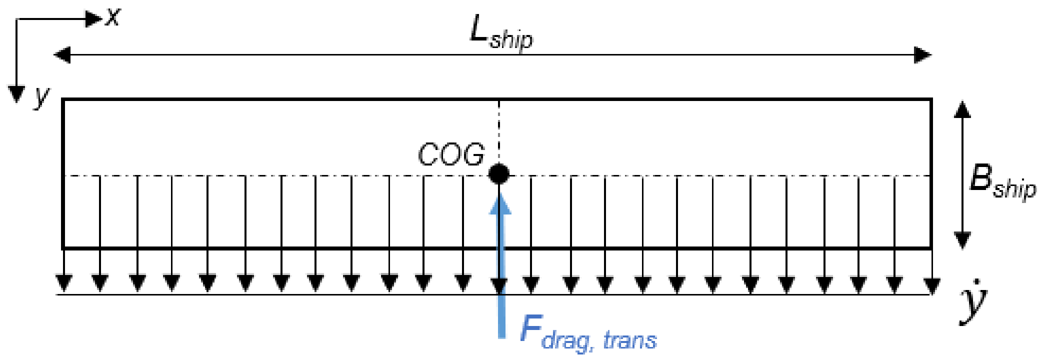


Figure B.8 Drag for translational movements

The drag force needs to be added to Equation (B.25) to obtain the complete equation of motion for the y-direction:

$$F_y + F_{drag,trans} = m_y\, \ddot{y}_C \qquad \text{(B.44)}$$

B.4.2 Rotation

With respect to the rotation of the ship, a story similar to translation of the ship holds. As a result of the angular velocity the vessel possesses, the translational velocity of the ship varies linearly with the distance to the centre of gravity, see Figure B.9. However, the drag force is related to the square of the velocity, resulting in a non-linear behaviour of the drag force for rotation with respect to the distance to the COG, see Figure B.10. Application of the formula in equation (B.43) then leads to the following expression:

$$|F_{drag,rot}| = \int_{\mathrm{COG}}^{\frac{L_{ship}}{2}} \frac{1}{2}\rho\, C_W\, \dot{y}^2\, d\ \mathrm{d}x \qquad \text{(B.45)}$$

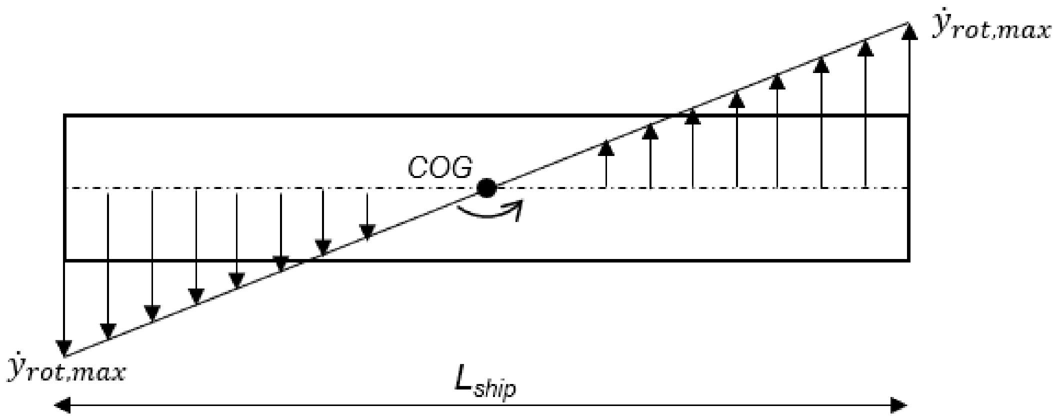


Figure B.9 Translational velocity due to a rotational velocity

Since both the left-hand side and the right-hand side of the ship experience this rotation induced drag, the total drag is equal to two times $F_{drag,rot}$. For the translational movement of the ship, the $F_{drag,rot}$ has no influence. In order to obtain the angular acceleration that is caused by the eccentric forcing, Equation (B.46) can be used. One should keep in mind that drag occurs in the opposite direction as the movement is taken place. Therefore, a plus-minus sign is present in Equation (B.46).

$$\ddot{\theta} = \frac{M}{I_G} = \frac{\pm 2\, F_{drag,rot}}{I_G}\, \frac{3}{8} L_{ship} \qquad \text{(B.46)}$$

The 3/8 L_{ship} in Equation (B.46) and Figure B.10 originates from the distance of the centre of gravity of the parabola to the COG, i.e. ¾ · ½ L_{ship} = ⅜ L_{ship}.

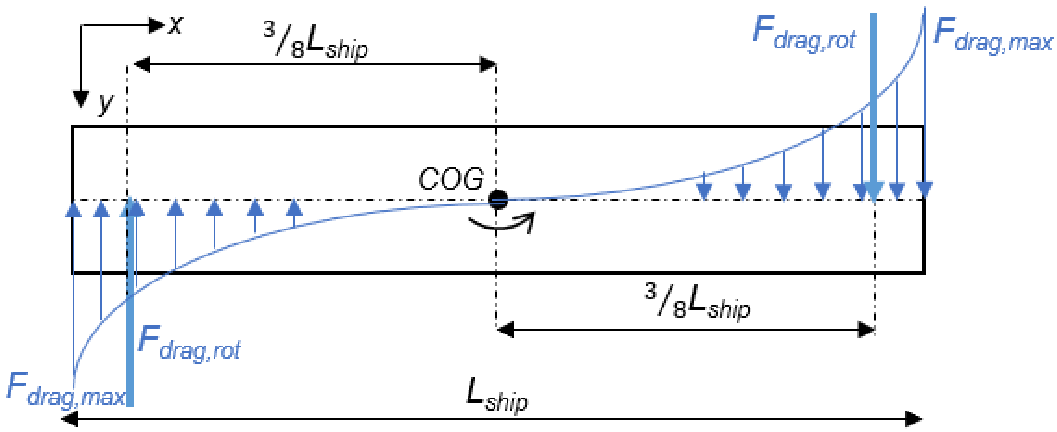


Figure B.10 Drag for rotational movements

Similar to the translation in y-direction, the drag effects need to be added to Equation (B.28). This then leads to Equation (B.47):

$$\sum M|_C = I_G\,\ddot{\theta} = -F_y(a + \mu b) \pm 2F_{drag,rot}\frac{3}{8}L_{ship} \qquad \text{(B.47)}$$

B.5 Verification

Even though the method to describe the movement is not based on the distribution of energy within the system, a verification of the model can be done by means of an energy balance. In an undamped system the total amount of energy in the system should be constant. Any deviations from this amount of energy, may imply errors or inconsistencies in the model. The energy balance is discussed more elaborately in appendix C.

Appendix C: Energy definition in subparts of the vessel-structure system

As a means for verification of the model, the composition of the energy within the system can be used. Since energy is a preserved quantity, the total amount of energy in the system should be constant for an undamped system. Deviations from the total energy in the system may imply an error or inconsistency in the model.

C.1 Kinetic energy vessel

The kinetic energy of the vessel is composed of two components, i.e. the energy due to its translational velocity and its energy due to rotational velocity. These components are discussed in subsection C.1.1 and C.1.2 respectively.

C.1.1 Kinetic energy due to translational velocity

The first component in the total energy composition is the kinetic energy of the vessel due to its translational velocity. This translational energy can be divided into components for the x- and y- direction, as is presented in Equation (C.1) and Equation (C.2).

$$E_{kin,trans.x} = \frac{1}{2} m\, \dot{x}_C^2 \quad [\mathrm{Nm}] \tag{C.1}$$

$$E_{kin,trans.y} = \frac{1}{2} m\, \dot{y}_C^2 \quad [\mathrm{Nm}] \tag{C.2}$$

To account for the water that is moving behind the vessel, the correction factor C_m is introduced, see section 2.3.6. This added mass will also be taken into account in the energy balance and it will be included in the kinetic energy due to translational velocity of the vessel. This leads to the following result:

$$E_{kin,trans.x} = \frac{1}{2} m_x\, \dot{x}_C^2 \quad [\mathrm{Nm}] \tag{C.3}$$

$$E_{kin,trans.y} = \frac{1}{2} m_y\, \dot{y}_C^2 \quad [\mathrm{Nm}] \tag{C.4}$$

$$E_{kin,trans,tot} = \frac{1}{2} m(\dot{x}_C^2 + \dot{y}_C^2)\, C_m \quad [\mathrm{Nm}] \tag{C.5}$$

Since the waterfront attenuation factor C_c and the ship flexibility factor C_s are both equal to one, there is no need to incorporate these factors in the energy composition. The eccentricity factor C_E is merely related to the rotation of the ship and will therefore be discussed in the next subsection.

C.1.2 Kinetic energy due to rotational velocity

The basic expression for the rotational energy reads:

$$E_{kin,rot} = \frac{1}{2} I_G \dot{\theta}^2 \quad [\mathrm{Nm}] \tag{C.6}$$

Similar to the kinetic energy of the translational velocity, correction factors are applied on the kinetic energy due to the rotation of the vessel. However, the waterfront attenuation factor and the ship flexibility factor do not have any influence since they are equal to one for flexible dolphins.

With respect to the added mass of the water in the longitudinal direction it can be said that the added mass pushing the ship in a clockwise rotation cancels out the pushing in counter-clockwise rotation. However, the added mass for the rotation, as described in section B.3.2, is still present and is included in I_G in Equation (C.6).

C.2 Potential energy

The potential energy in the system is composed of the potential energy that is stored in the horizontal berthing beam, the vertical piles and the soil. The general expression for the potential energy in a beam subjected to bending is: (Welleman H. , 2019)

$$E_{pot} = \int_0^L \frac{M(x)^2}{2\,EI}\,\mathrm{d}x \tag{C.7}$$

For the horizontal berthing beams the above formula is applied. The potential energy in the vertical piles and the soil are combined in one term in the energy balance, being $E_{piles+soil}$. Combination of these two potential energies arises from the representation of the stiffness of the soil and the pile by one equivalent spring.

The energy in one nonlinear spring is equal to, see also Figure C.1:

$$E_{abs;i} = E_{abs;i-1} + \Delta E_{abs;i} \tag{C.8}$$

where:

$E_{abs;i}$	Potential energy nonlinear spring current timestep	[Nm]
$E_{abs;i-1}$	Potential energy nonlinear spring previous timestep	[Nm]
$\Delta E_{abs;i}$	Increase/decrease potential energy nonlinear spring $=0.5(F_i+F_{i-1})\cdot(u_i - u_{i-1})$	[Nm]

with:

F_i	Force in nonlinear spring in current timestep	[N]
F_{i-1}	Force in nonlinear spring in previous timestep	[N]
u_i	Displacement nonlinear spring in current timestep	[m]
u_{i-1}	Displacement nonlinear spring in previous timestep	[m]

The total potential energy in the vertical piles and soils is then equal to:

$$E_{piles+soil} = \sum_{j=1}^{5} E_{abs;ij} \tag{C.9}$$

where:

$E_{piles+soil}$	Total potential energy in the vertical piles and the soil	[Nm]
$E_{abs;ij}$	Potential energy nonlinear spring in current timestep, pile j	[Nm]

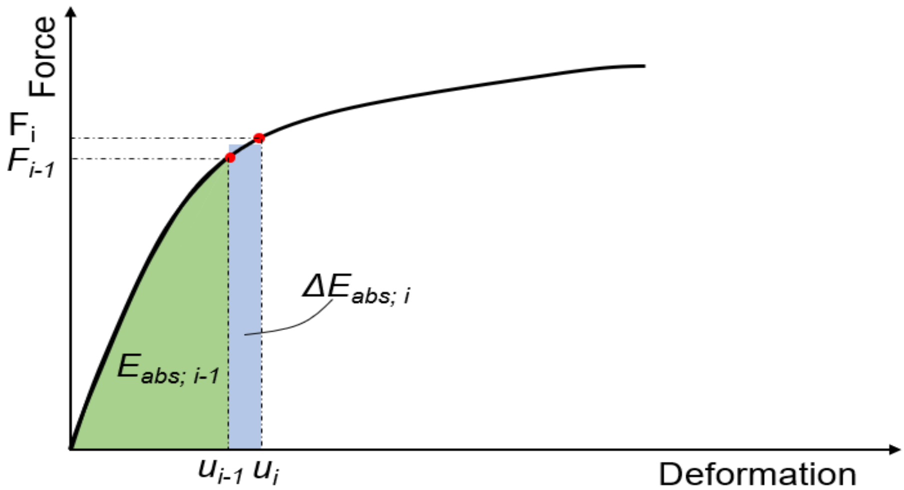


Figure C.1 Potential energy piles and soil

C.3 Kinetic energy structure

Since the structure is moving, it also contains kinetic energy. Complicating factor, however, is that the structure is not considered as a rigid body. This means that every part of the structure may move with a different velocity. In order to make a substantiated assumption for the mass that is in motion, the deflection of the berthing beam is used.

The absolute value of the area underneath the deflection curve is scaled by the area that is created when the horizontal beam would deflect for a value of w_{max} over its whole length. This is schematically depicted in Figure C.2. The result of this division will be the scaling factor η that is applied on the total mass M of the guide work to determine the mass of the structure that is actually in motion.

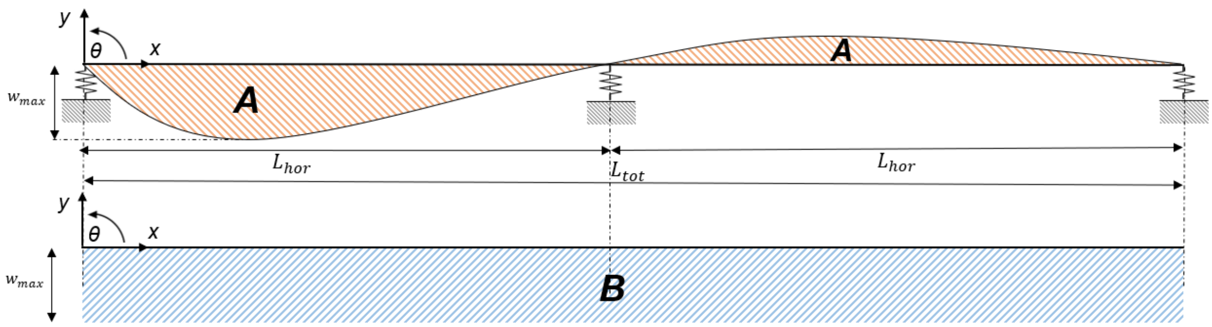


Figure C.2 Background for the scaling factor for the mass of the guide work

In formula form, this yields:

$$\eta = \frac{A}{B} = \frac{\int_0^L |w(x)|\mathrm{d}x}{L_{tot}\, w_{max}} \quad [-] \tag{C.10}$$

With the aid of this scaling the kinetic energy within the structure will be equal to:

$$E_{kin,structure} = \frac{1}{2}\eta\, M\, \dot{y}_V^2 \quad [\mathrm{Nm}] \tag{C.11}$$

The velocity of the structure used in Equation (C.11) is the equal to the velocity of the bow of the vessel, since the structure and ship move simultaneously during the contact.

C.4 Potential energy soil

The potential energy in the soil is composed of the energy stored in the rotational spring and energy stored in the translational spring. This potential energy is incorporated in Equation (C.9).

C.5 Energy dissipation

Various phenomena can be mentioned that contribute to the dissipation of energy in the vessel-structure system. A list of possible sources for energy loss is given below

- Plastic deformation of the vessel’s hull
- Plastic deformation of the structure
- Plastic deformation of the soil
- Radiation damping by the soil
- Radiation damping by the water
- Rolling movement of the ship
- Drag due to the water
- Turbulence of the water
- Friction between ship and structure

From the aspects mentioned above, only the drag by the water is taken into account, as well as friction between ship and structure. The work done due to friction is defined as:

$$E_{friction} = \int_{x_V(t_0)}^{x_V(t)} F_x(t)\ \mathrm{d}x \quad [\mathrm{Nm}] \tag{C.12}$$

Similarly, for the work done by the drag Equation (C.13) and Equation (C.14) can be formulated and can be used for verification purposes in chapter 5.

$$E_{drag,rot} = \int_{\theta(t_0)}^{\theta(t)} 2\, F_{drag,rot}(t)\, \frac{3}{8} L_{ship}\ \mathrm{d}\theta \quad [\mathrm{Nm}] \tag{C.13}$$

$$E_{drag,trans} = \int_{y_C(t_0)}^{y_C(t)} F_{drag,trans}(t)\ \mathrm{d}y \quad [\mathrm{Nm}] \tag{C.14}$$

C.6 Energy balance

With the aid of the previous sections the energy balance in the time domain can be constructed when the analysis is run. In order to obtain a useful graph, the kinetic energy of the vessel in x-direction, the work done by friction and the work done by the engine of the vessel are omitted in the energy balance.

Reason for the omission is given by the fact that the velocity in the y-direction is in the order of 1% (≈ tan(5°)) of the velocity in x-direction. Since the kinetic energy is related to the square of the velocity, the energy in x-direction dominates the energy balance. Work done by friction and by the engine predominantly affects the velocity in x-direction and is therefore also not shown in the energy balance. An example of the energy balance is given in Figure C.3.

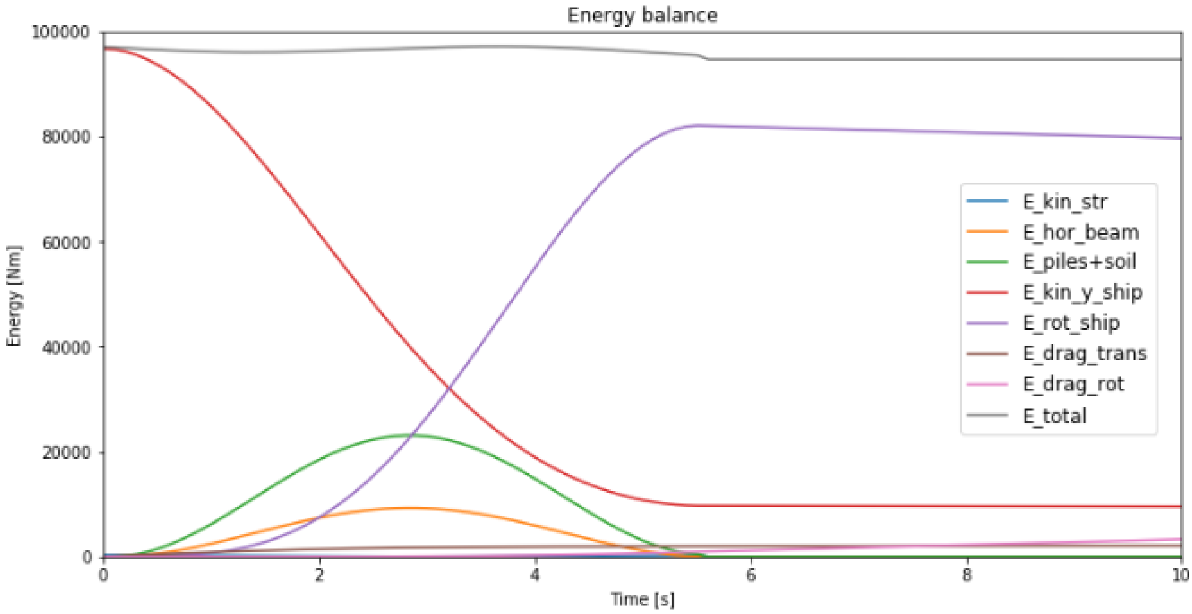


Figure C.3 Energy balance first impact

The dimensions of the vessel and the geometry of the structure that lead to the result in Figure C.3 are discussed in chapter 5. Also, a qualitative and quantitative verification of the performed analysis is given in this chapter.

Appendix D: Results

D.1 Influence input parameters on energy absorption

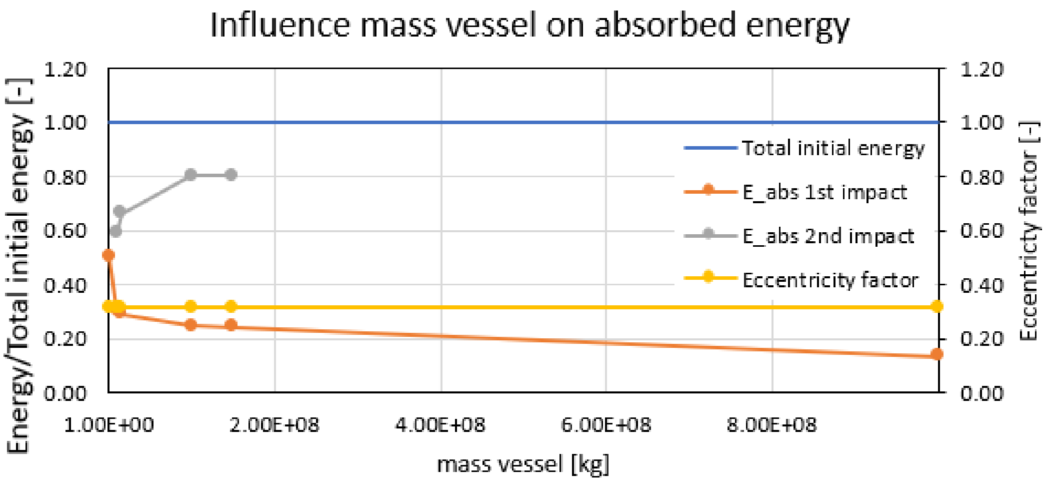


Figure D.1 Absorbed energy as function of the mass of the colliding vessel

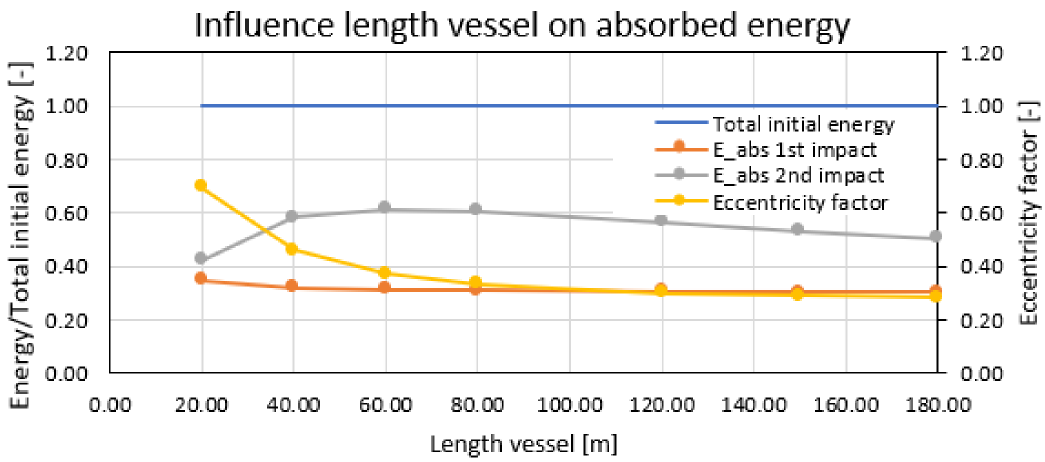


Figure D.2 Absorbed energy as function of the length of the colliding vessel

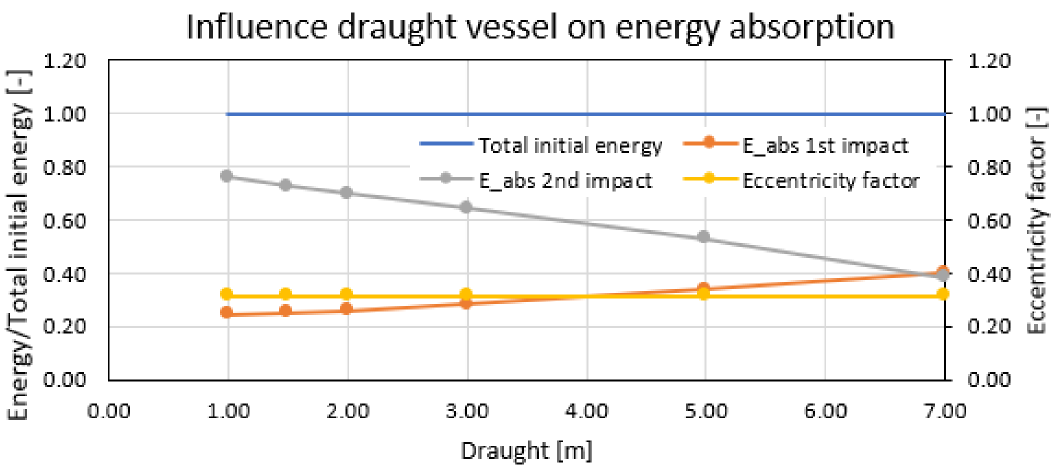


Figure D.3 Absorbed energy as function of the draught of the colliding vessel

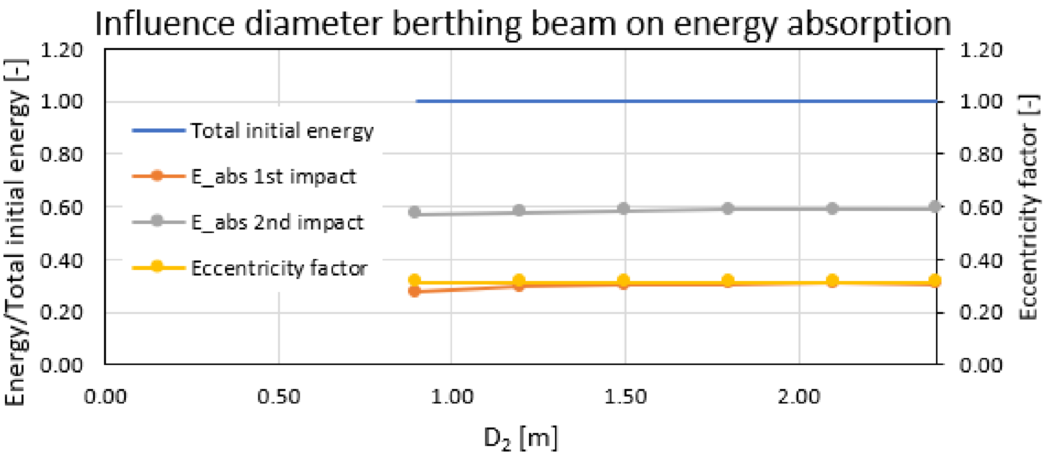


Figure D.4 Absorbed energy as function of the outside diameter horizontal beam

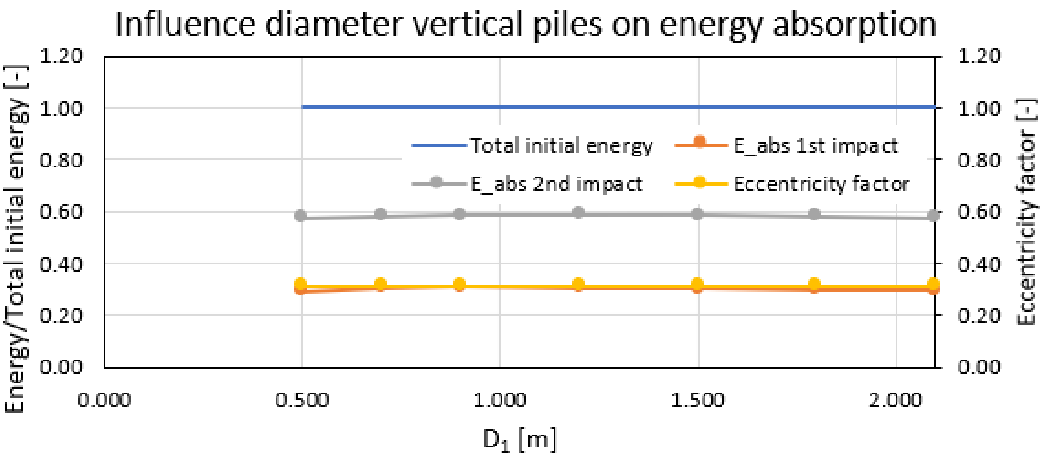


Figure D.5 Absorbed energy as function of the outside diameter of the vertical piles

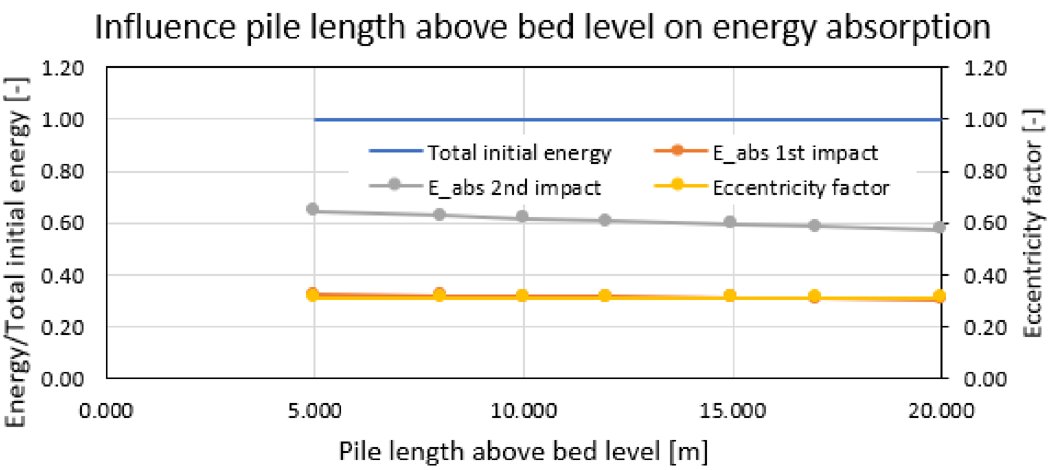


Figure D.6 Absorbed energy as function of the length of the vertical piles above bed level

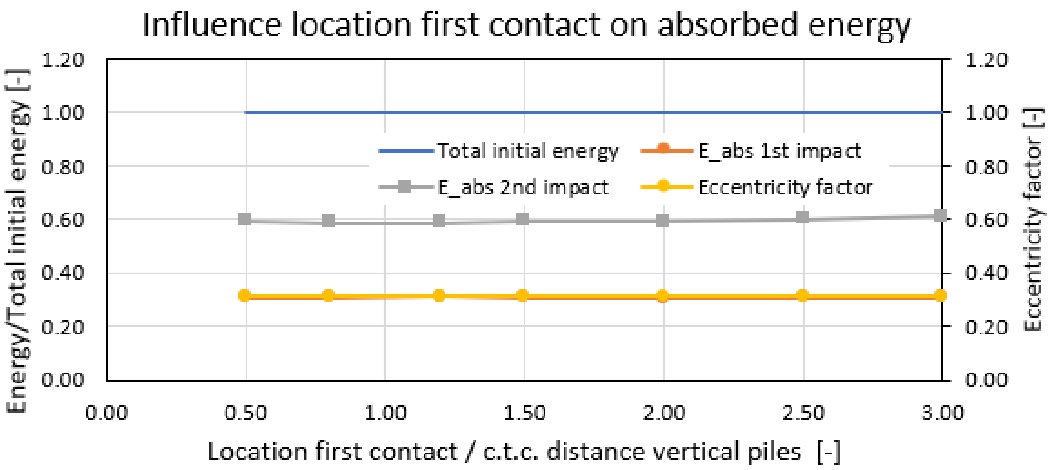


Figure D.7 Absorbed energy as function of the location of the first contact

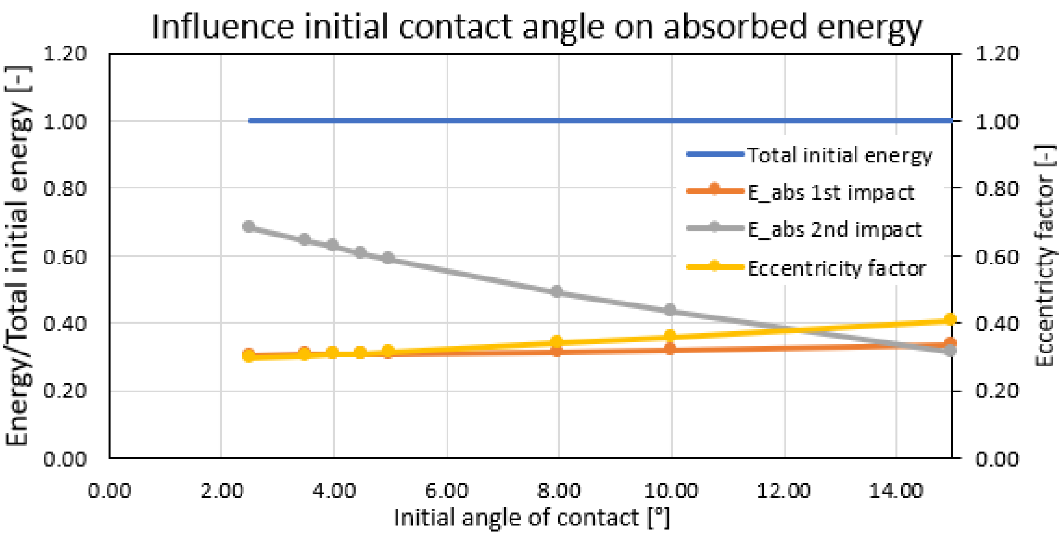


Figure D.8 Absorbed energy as function of the initial contact angle

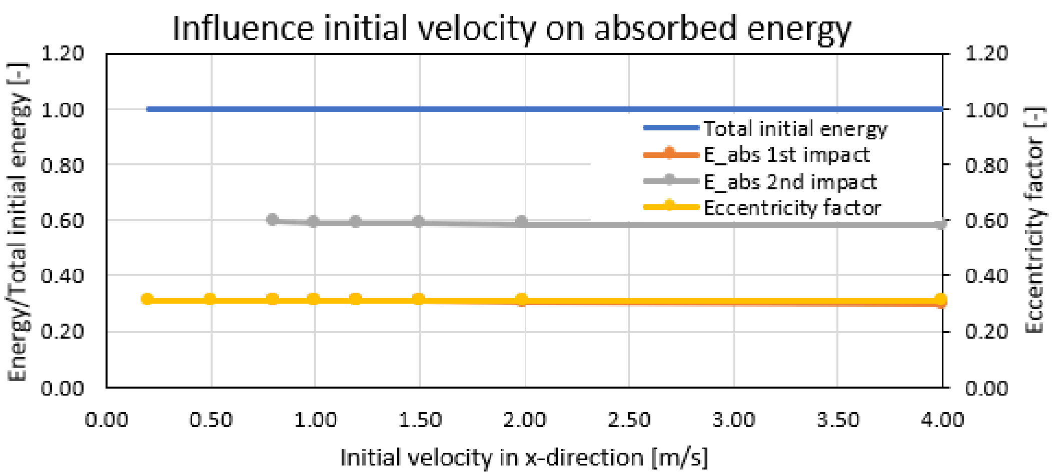


Figure D.9 Absorbed energy as function of the initial velocity in x-direction of the colliding vessel

D.2 Influence input parameters on the contact force

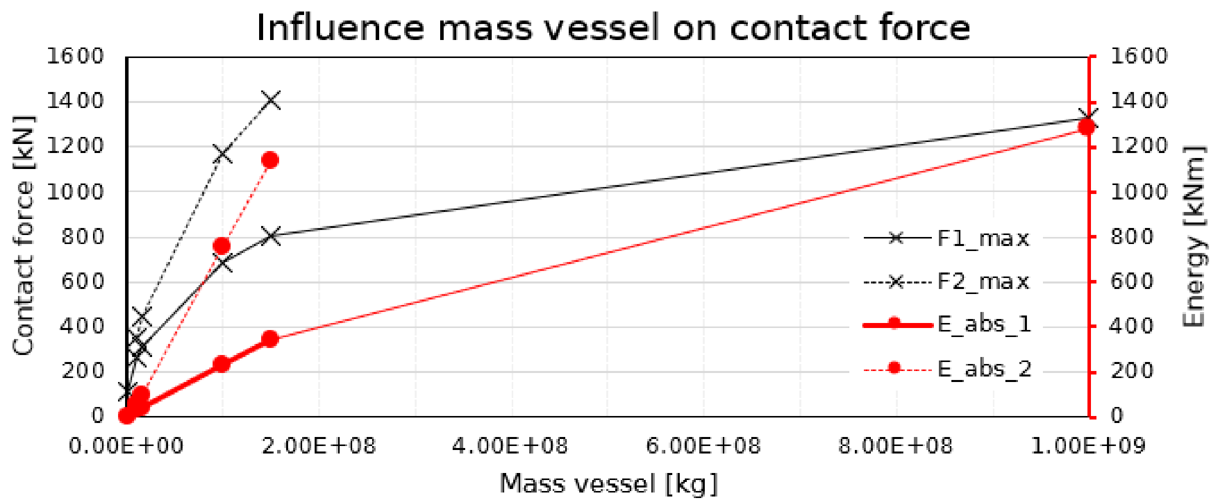


Figure D.10 Influence mass vessel on contact force during first and second impact

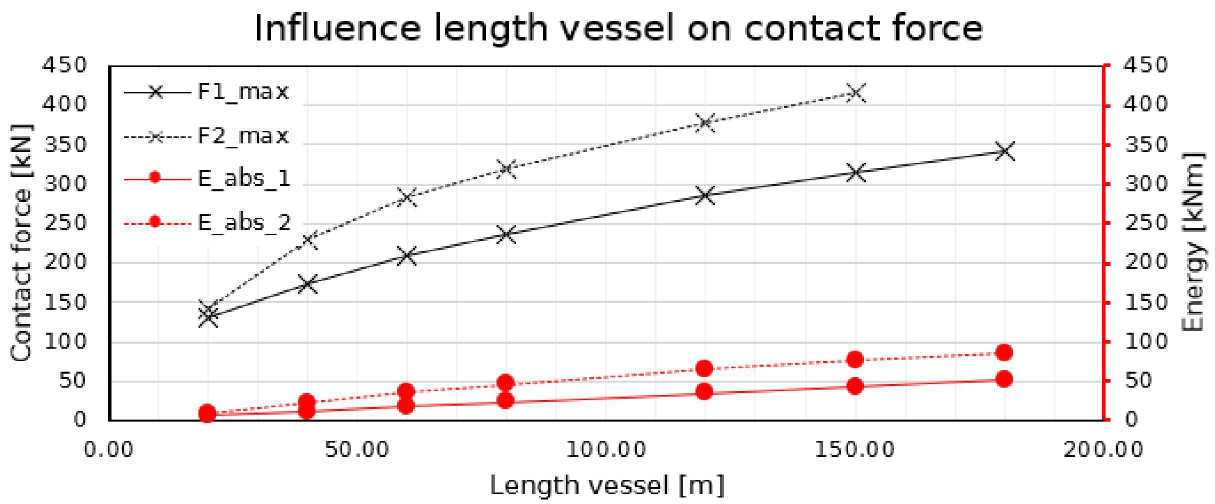


Figure D.11 Influence length vessel on contact force during first and second impact

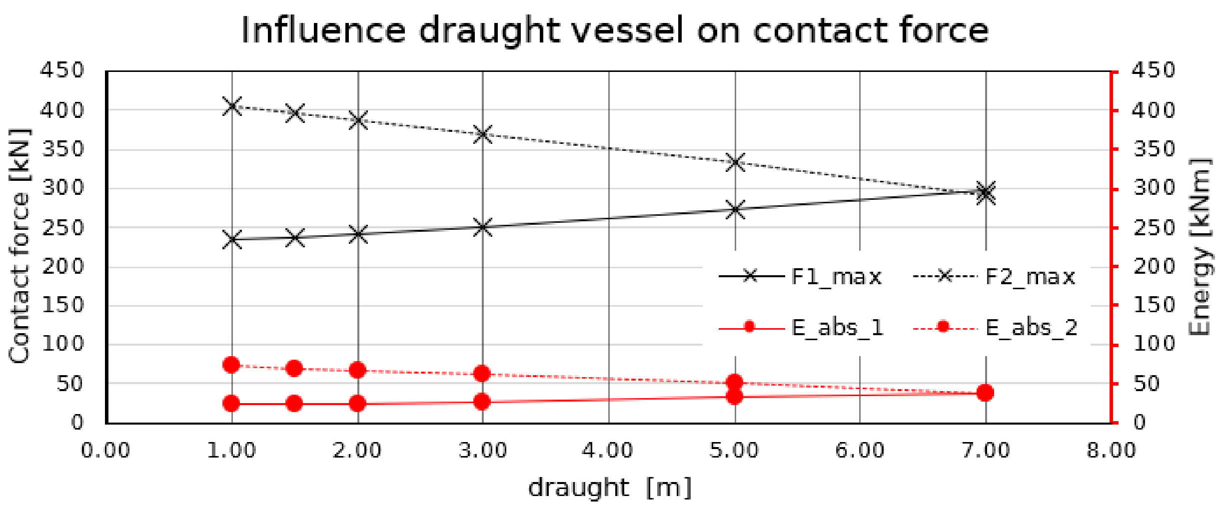


Figure D.12 Influence draught vessel on contact force during first and second impact

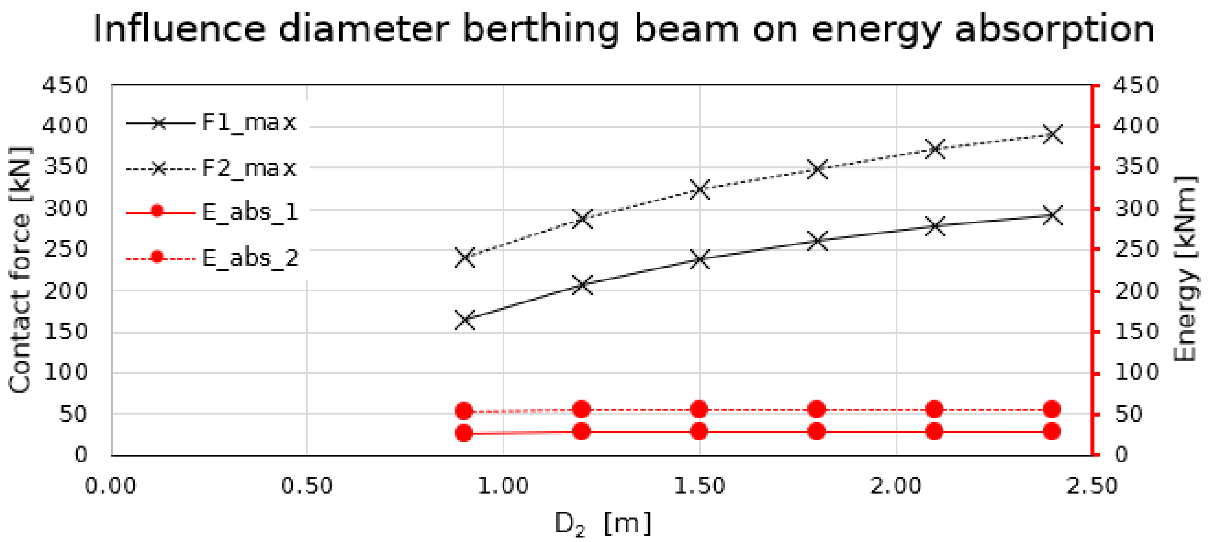


Figure D.13 Influence diameter berthing beam on contact force during first and second impact

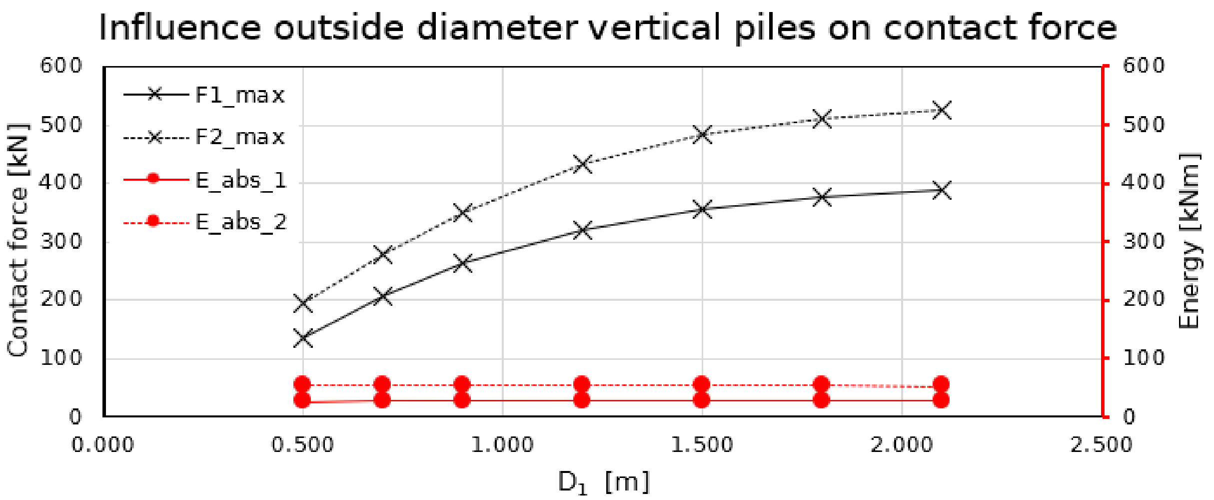


Figure D.14 Influence diameter vertical piles on contact force during first and second impact

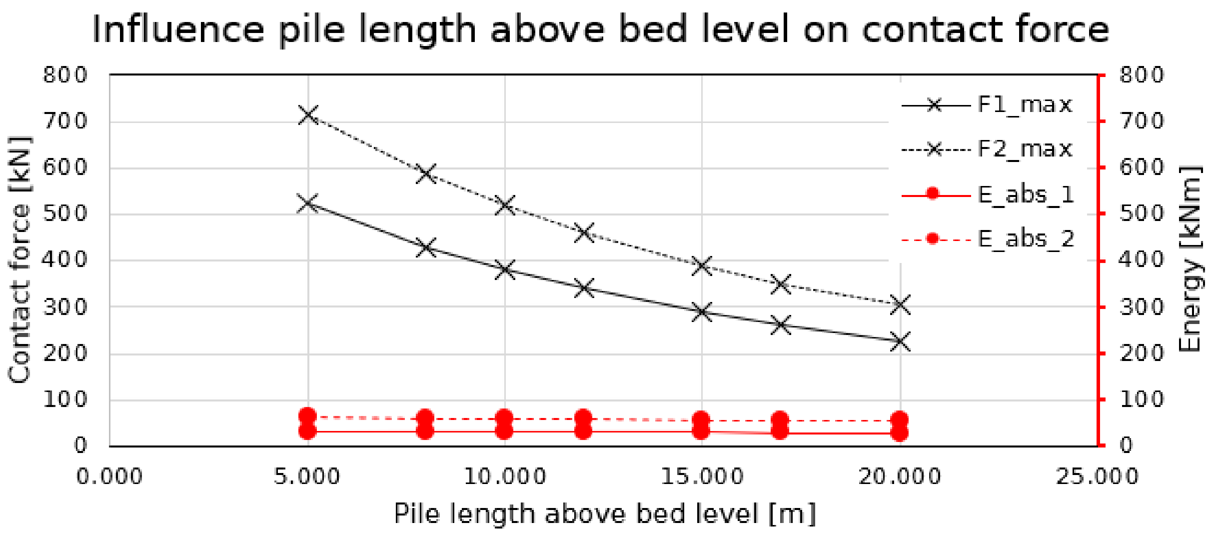


Figure D.15 Influence pile length above bed level on contact force during first and second impact

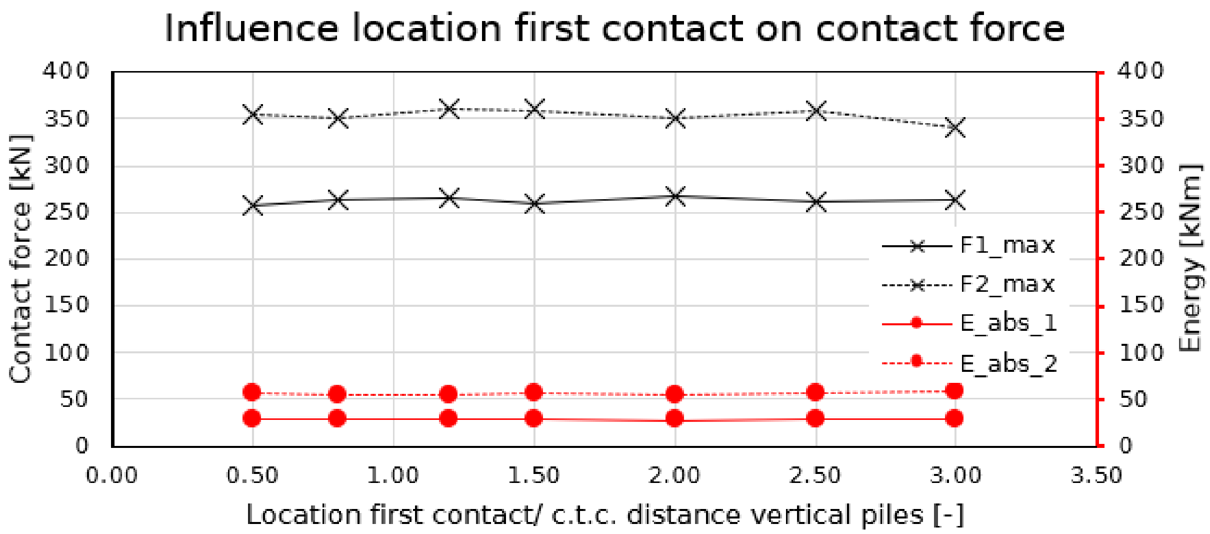


Figure D.16 Influence location first contact on contact force during first and second impact

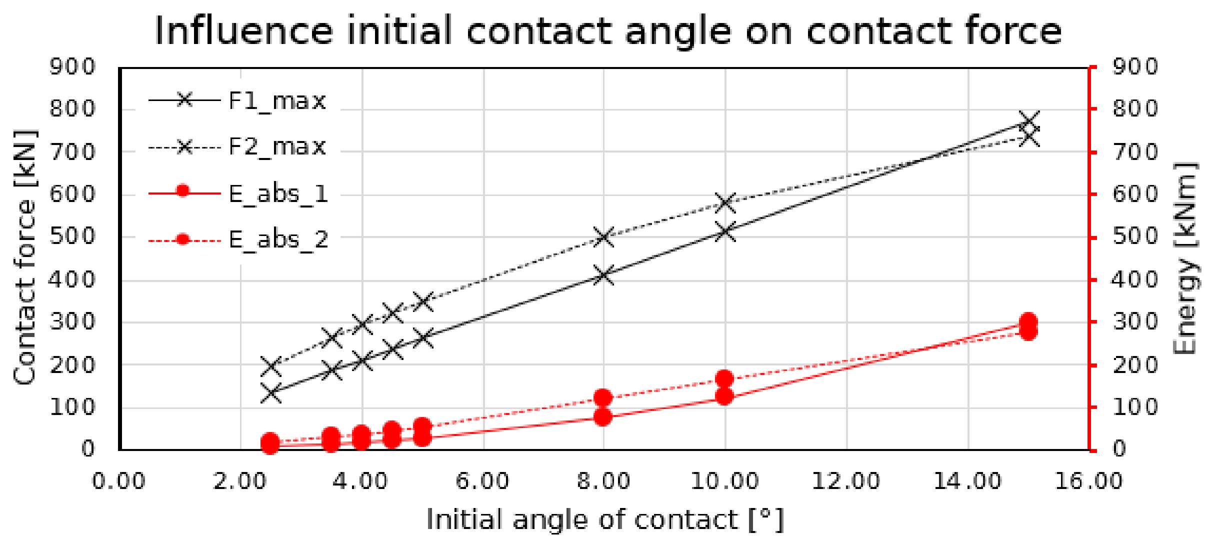


Figure D.17 Influence initial contact angle on contact force during first and second impact

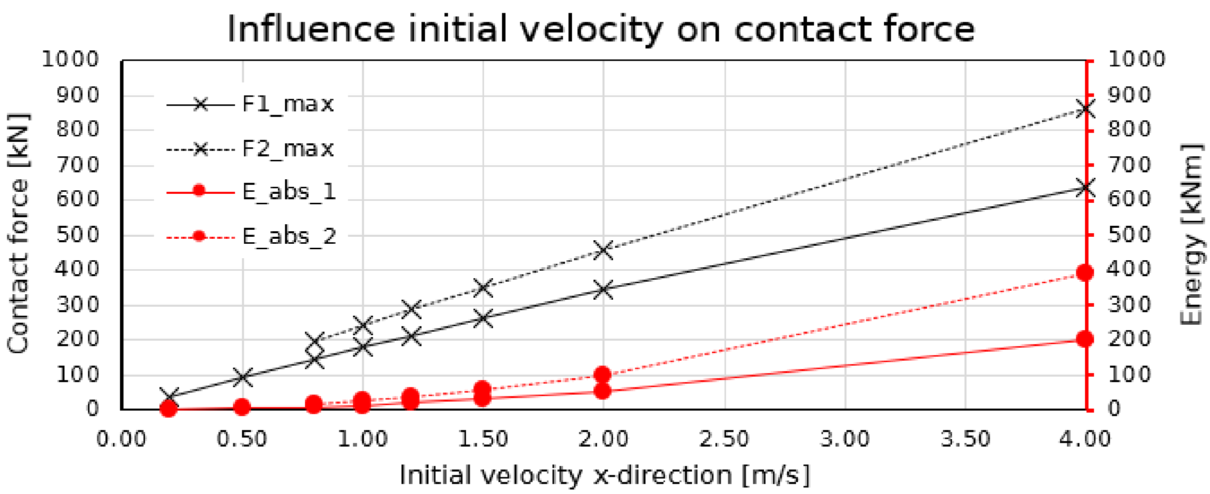


Figure D.18 Influence initial velocity on contact force during first and second impact

D.3 Influence input parameters on the guide work length

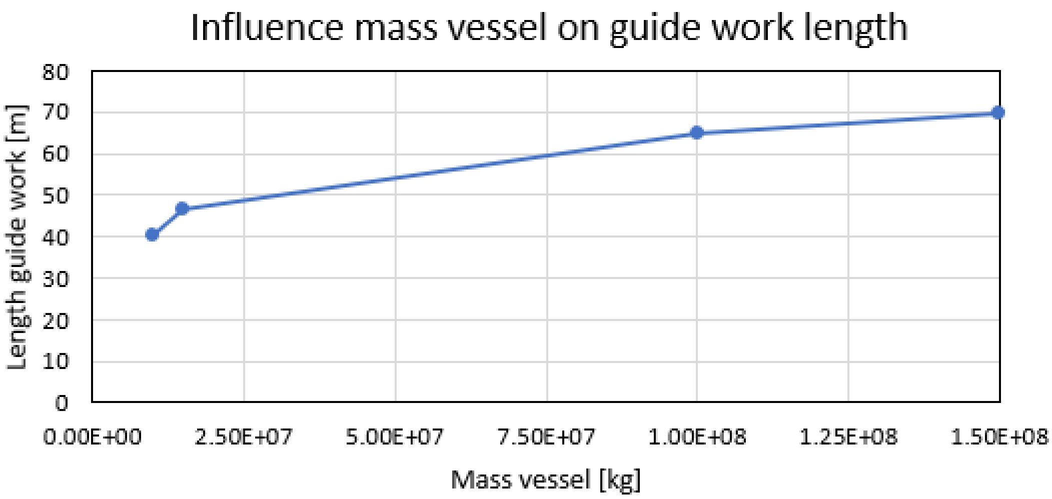


Figure D.19 Required length guide work as function of the mass of the colliding vessel

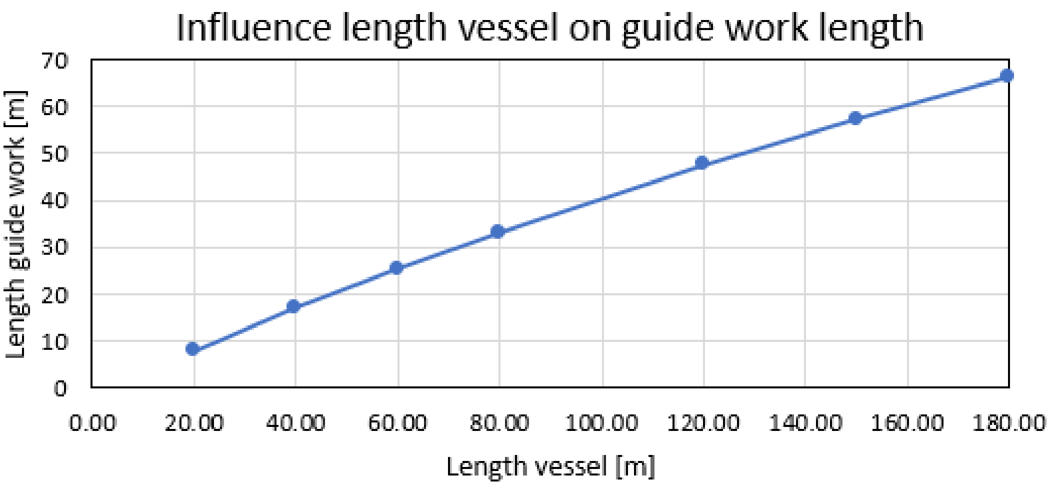


Figure D.20 Required length guide work as function of the length of the colliding vessel

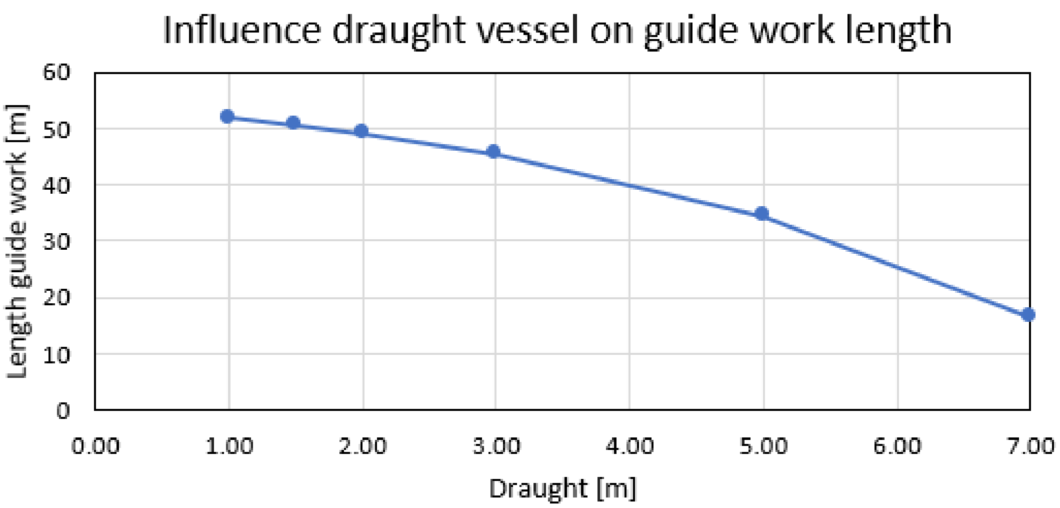


Figure D.21 Required length guide work as function of the draught of the colliding vessel

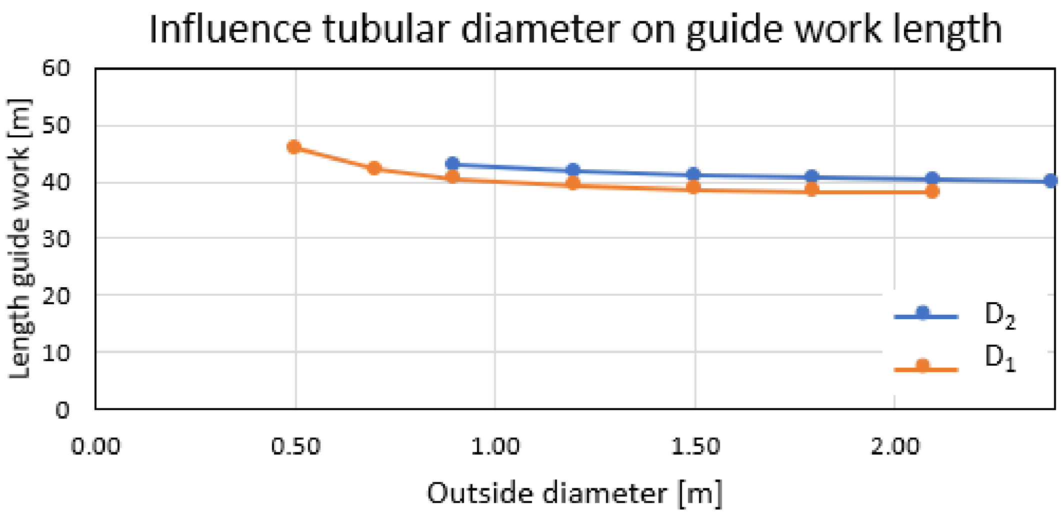


Figure D.22 Required length guide work as function of the diameter of the cross-sections

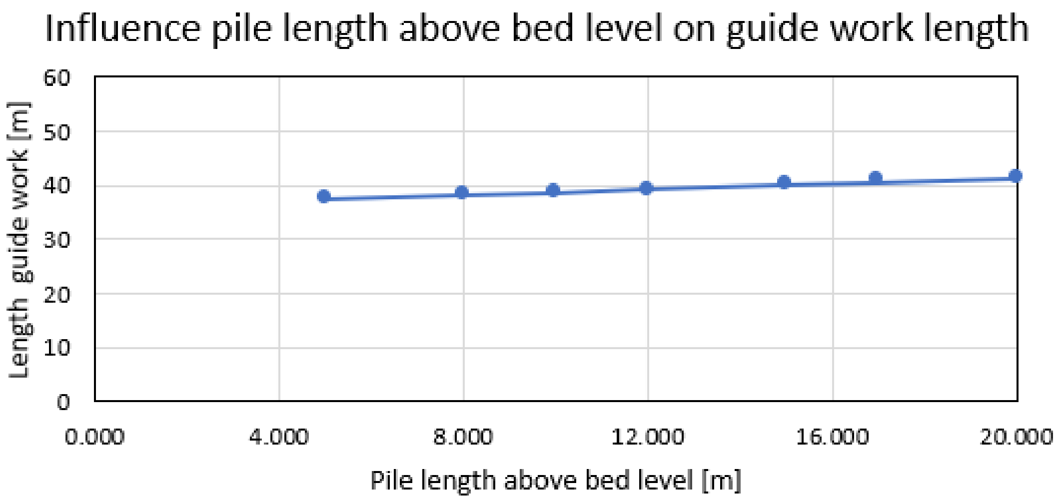


Figure D.23 Required length guide work as function of the vertical pile length above bed level

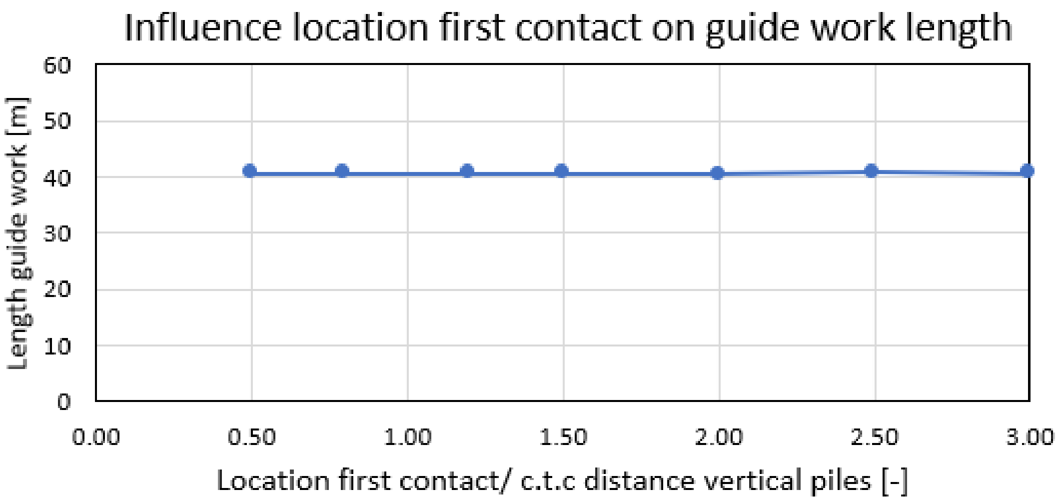


Figure D.24 Required length guide work as function of location of the first contact

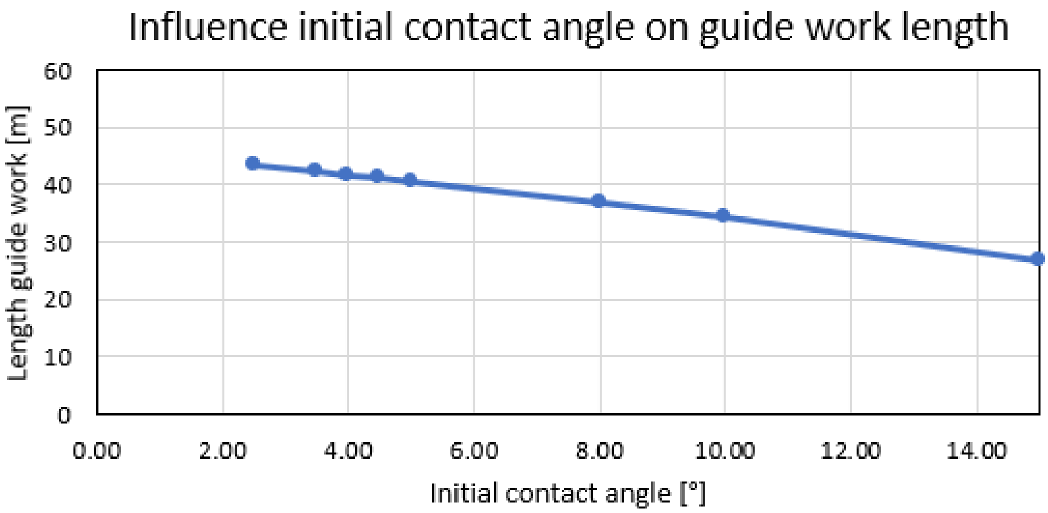


Figure D.25 Required length guide work as function of the initial contact angle

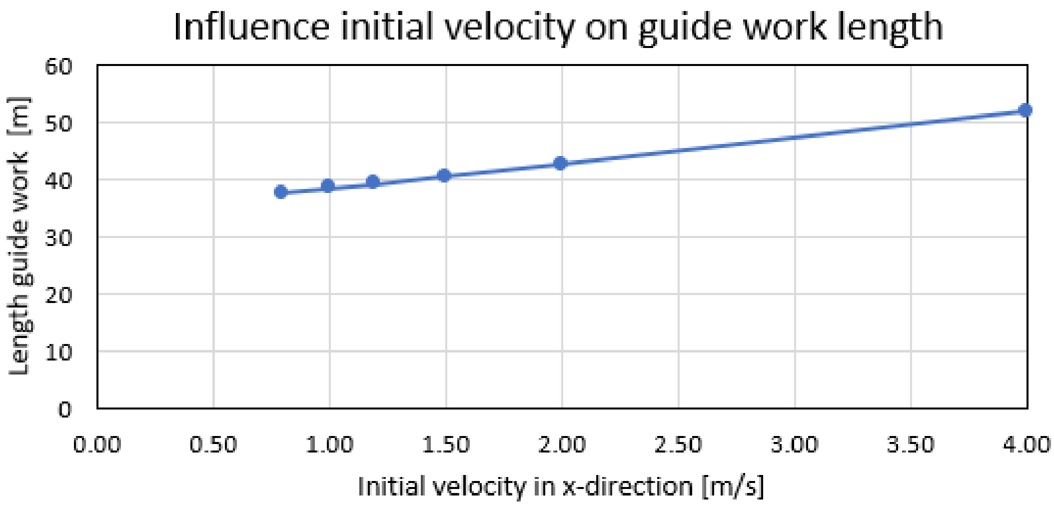


Figure D.26 Required length guide work as function of the initial velocity of the colliding vessel

Appendix E: Translation developed model to SCIA Engineer

The model that is created in Python to simulate brush collisions between inland navigation vessels and guide works, see chapter 4, is translated to the finite element software *SCIA Engineer*, or *SCIA* in short. This appendix describes the reason why this translation is made and provides a description of this tool. Subsequently, the possibilities and limitations of this translation are given. Finally, a verification of this tool involving SCIA Engineer is given.

E.1 Description of the translation to SCIA Engineer

The in Python created tool provides insight in the contact forces during a brush collision between a vessel and a guide work. Because of the limitations of the Python model, see section 4.4, a translation to SCIA is made. In comparison with the Python the model in SCIA has the following advantages:

- It is possible to vary the amount of vertical piles from two to seven
- It is possible to differentiate between 'end piles' and 'middle piles'
- It is possible to divide the 'end piles' and 'middle piles' into ten segments with different wall thicknesses, according to common design practice.
- Rigid connections between the horizontal berthing beam and the vertical piles are taken into account; the influence of torsion is incorporated in the force distribution.
- The model can be used directly for the design of a guide work, since the post-processing environment of SCIA is able to provide the occurring cross-section forces and principle stresses.

In order to model the problem in SCIA Engineer, first a deterministic model needs to be created in the program to form a template. The next step is exporting the model as an .XML file to have the '.XML template' of the model.

The .XML file can then be opened in Microsoft Excel. With the aid of a Virtual Basic (VBA) script it is possible to change different values in the .XML file, e.g. the point of application and the value of the load. Subsequently, exporting the edited .XML file from Excel and importing it in SCIA Engineer again makes it possible to run the model with updated values for the variables. This VBA script is given in appendix G.

In SCIA Engineer it is also possible to generate an Excel file with the results of an analysis. Combining this function with the knowledge about the .XML import and export, a loop can be created that, similar to the Python script, can describe the contact in the time domain. A schematic representation of this routine is given in Figure E.1.

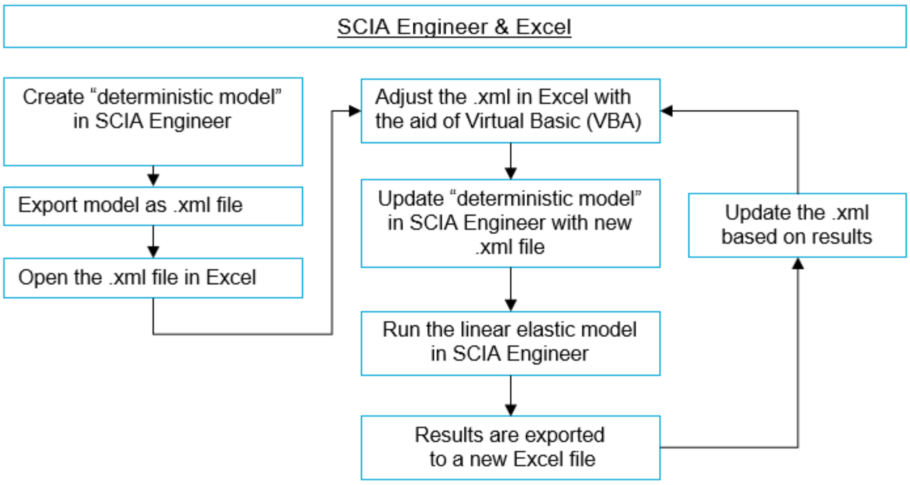


Figure E.1 Looping in SCIA Engineer and Excel

The procedure itself for the analysis of a brush collision in the Python tool, as depicted in Figure 4.4, is equal to the procedure used in the translation to SCIA Engineer and Excel.

The analysis starts with the definition of the used parameters in the form of an Excel sheet, see Figure E.2. In this sheet the diameter, thickness and centre to centre distance of the vertical piles (*L_hor*) need to be specified, as well as the height of its centroidal axis with respect to NAP. Furthermore, the bed level and the found trendline for the rotational and translational stiffness of the soil is inputted.

On top of that, the properties of the colliding vessel, friction coefficient, dummy force and the density of the water is specified. Also, correction factors, identical to the Python model, are introduced here.

Finally, the number of segments and the location of the first contact are defined. The cell with *field of first contact* defines the field where the bow starts the contact with the guide work, while the *point of application of the force* defines the location on this field by a value between 0 and 1, e.g. 0.5 in case of first contact in the middle of a segment.

Horizontal tube		
D2	1820	mm
t2	13	mm
Bed level	-7	[m][NAP]
L_hor	40	m
Height hor_beam N.A.P.	10	m [NAP]
Spring pile 1		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 2		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 3		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 4		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 5		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 6		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]
Spring pile 6		
F_spring1	1000	[N]
K_trans1	5944991669	[N/m]
K_rot1	7327739205	[Nm/rad]
depth	0.27111721	[m]

SHIP PROPERTIES		
Mass	1.00E+07	[kg]
Length	100.3	[m]
Width	22.8	[m]
Total height	10	[m]
draught	4	[m]
Angle of contact	5	[°]
Velocity x-direction	-1.5	[m/s]
PARAMETERS		
Friction coefficient	0.15	[-]
F_initial	1000	[N]
ρ_water	1000	[kg/m^3]
COEFFICIENTS		
drauhgt/total height	0.4	[-]
C_m	1.1	[-]
C_w	2	[-]
STRUCTURE		
Number of segments	4	[-] **(MAX 6)**
OTHER		
Timestep	0.1	s
Field of first contact		
Field	1	[-]
Point of application of the force	0.8	[-]

Figure E.2 Excel input sheet translation to SCIA Engineer (1/2)

After the parameters concerning the soil, design conditions and the horizontal berthing beam are submitted, also the dimensions of the vertical piles need to be defined. Every vertical is divided into ten different segments, that can all have a unique wall thickness. Also, the first pile, middle piles and the end pile are differentiated, see Figure E.3.

Vertical tube								
FIRST PILE			**MIDDLE PILES**			**END PILE**		
Segment 1			**Segment 1**			**Segment 1**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	3.3	m [N.A.P.]	Bottom ends at	3.3	m [N.A.P.]	Bottom ends at	3.3	m [N.A.P.]
Segment 2			**Segment 2**			**Segment 2**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	1	m [N.A.P.]	Bottom ends at	1	m [N.A.P.]	Bottom ends at	1	m [N.A.P.]
Segment 3			**Segment 3**			**Segment 3**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-2	m [N.A.P.]	Bottom ends at	-2	m [N.A.P.]	Bottom ends at	-2	m [N.A.P.]
Segment 4			**Segment 4**			**Segment 4**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-5	m [N.A.P.]	Bottom ends at	-5	m [N.A.P.]	Bottom ends at	-5	m [N.A.P.]
Segment 5			**Segment 5**			**Segment 5**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-6.1	m [N.A.P.]	Bottom ends at	-6.1	m [N.A.P.]	Bottom ends at	-6.1	m [N.A.P.]
Segment 6			**Segment 6**			**Segment 6**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-7	m [N.A.P.]	Bottom ends at	-7	m [N.A.P.]	Bottom ends at	-7	m [N.A.P.]
Segment 7			**Segment 7**			**Segment 7**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-8	m [N.A.P.]	Bottom ends at	-8	m [N.A.P.]	Bottom ends at	-8	m [N.A.P.]
Segment 8			**Segment 8**			**Segment 8**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-13	m [N.A.P.]	Bottom ends at	-13	m [N.A.P.]	Bottom ends at	-13	m [N.A.P.]
Segment 9			**Segment 9**			**Segment 9**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-14	m [N.A.P.]	Bottom ends at	-14	m [N.A.P.]	Bottom ends at	-14	m [N.A.P.]
Segment 10			**Segment 10**			**Segment 10**		
D	900	mm	D	900	mm	D	900	mm
t	40	mm	t	40	mm	t	40	mm
Bottom ends at	-17.3	m [N.A.P.]	Bottom ends at	-17.3	m [N.A.P.]	Bottom ends at	-17.3	m [N.A.P.]

Figure E.3 Excel input sheet translation to SCIA Engineer (2/2)

Once all parameters are set, the created VBA script can be run. The template file for the guide work, see Figure E.4, is then adjusted according to the input parameters.

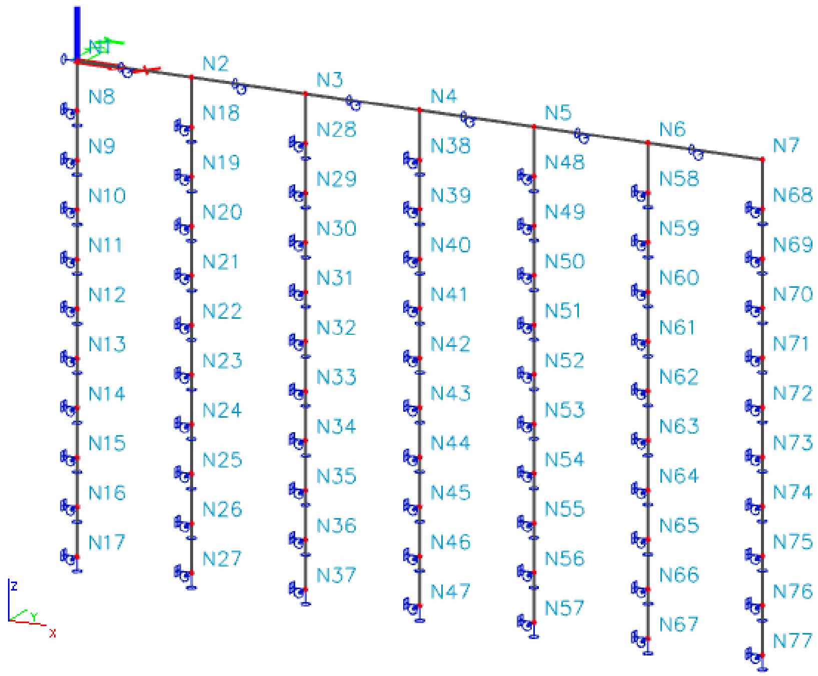


Figure E.4 Template guide work in SCIA Engineer

The updated guide work then has the geometry as defined in the Excel sheet. An example of an updated guide work is given in Figure E.5. While updating the structure, also the prescribed deformation of the corresponding time step (indicated with the green circle) is placed on the structure.

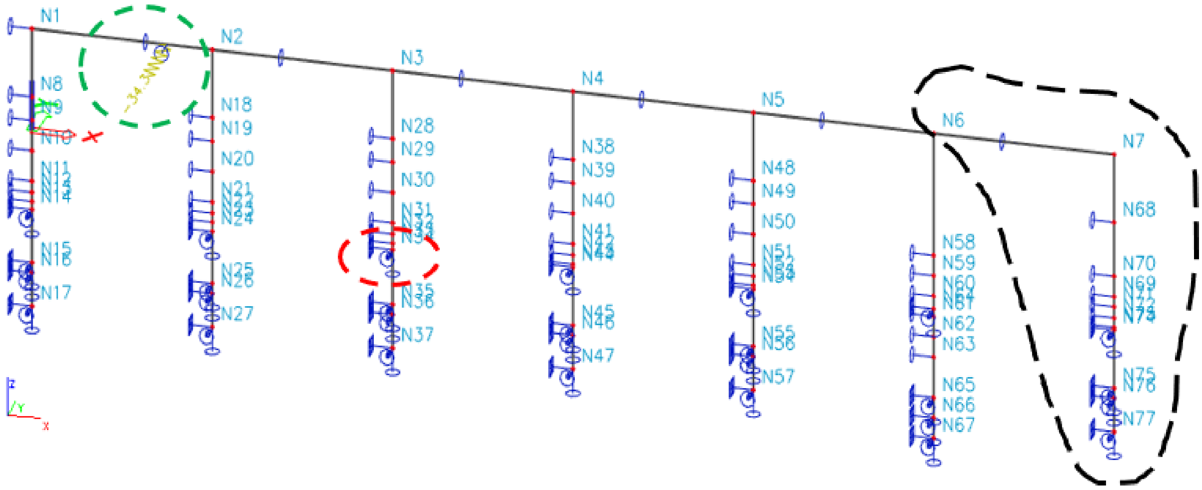


Figure E.5 Updated guide work in SCIA Engineer

In the shown example of Figure E.5, every vertical pile has its own embedment depth *t* and corresponding translational and rotational spring support, depending on the horizontal force at the top of the pile. Figure E.6 shows the properties of the spring support of the third pile (indicated with the red circle in Figure E.5).

Name	Sn28
Type	Standard
Angle [deg]	
Constraint	Custom
X	Rigid
Y	Flexible
Stiffness Y [MN/m]	1.5311e+03
Z	Rigid
Rx	Flexible
Stiffness Rx [MNm/rad]	4.1666e+03
Ry	Free
Rz	Free
Default size [m]	0.200
Node	N34
Geometry	
System	GCS

Figure E.6 Flexible support SCIA Engineer

The created template is constructed in a manner such that it is able to vary the number of fields from one to six. In case that less than six fields are used, all beam elements of the template that are not necessary, are replaced by a dummy cross-section, i.e. a circular cross-section with a diameter of 1 mm, see also Figure E.7.

Dummy cross-sections are also applied on the bottom parts of the vertical piles in case of limited lateral forces. It is, in fact, very well possible that not the full pile length is "activated" when the lateral force is small.

Usage of dummy cross-sections instead of omission of the respective elements, arises from the fact that it is more cumbersome to add or delete an element instead of replacing it by a dummy cross-section.

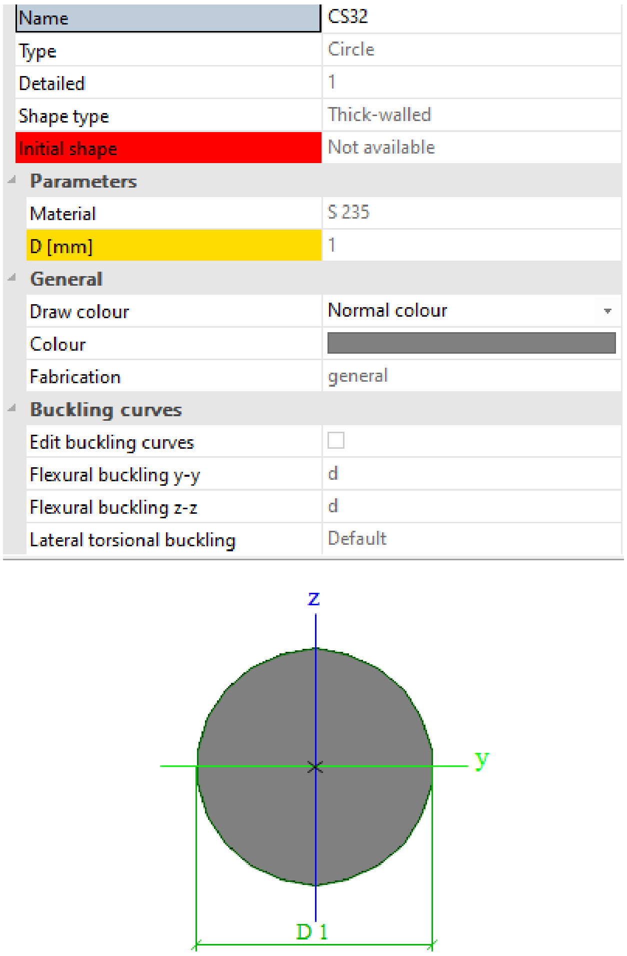
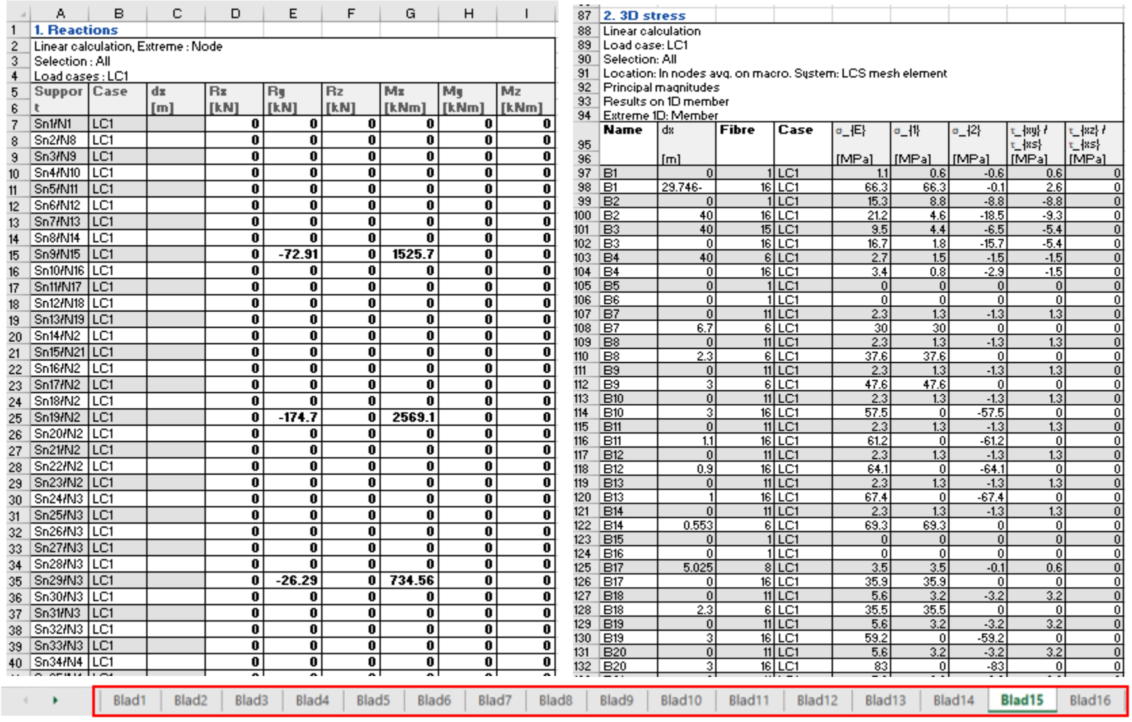


Figure E.7 Dummy cross-section SCIA Engineer

With the known values of the flexible supports and the prescribed deformation, the system can be solved. The support reactions, internal member forces and principle stresses per timestep are exported as an Excel file and can be used to verify the design of the structure.

The ‘support reaction’, i.e. the contact force between the vessel and the structure, is extracted from this result file and used for updating the velocities and displacements of the colliding vessel. Also, the shear forces in all vertical piles are extracted to determine the spring stiffnesses and embedment depth *t* for the next timestep.

Example of the result file is given in Figure E.8. The result document is an Excel file in which the output is stored per timestep. Every timestep has its own worksheet with the results of that step. In the future automated unity checks can be included to the document to speed up the design process. Given the limited available time to conduct the master thesis and the fact that this translation to SCIA Engineer is an additional part of the project scope, this is not implemented yet.



1. Reactions

Linear calculation, Extreme : Node
Selection : All
Load cases : LC1

Support	Case	dx [m]	Rx [kN]	Ry [kN]	Rz [kN]	Mx [kNm]	My [kNm]	Mz [kNm]
Sn1/N1	LC1		0	0	0	0	0	0
Sn2/N8	LC1		0	0	0	0	0	0
Sn3/N9	LC1		0	0	0	0	0	0
Sn4/N10	LC1		0	0	0	0	0	0
Sn5/N11	LC1		0	0	0	0	0	0
Sn6/N12	LC1		0	0	0	0	0	0
Sn7/N13	LC1		0	0	0	0	0	0
Sn8/N14	LC1		0	0	0	0	0	0
Sn9/N15	LC1		0	-72.91	0	1525.7	0	0
Sn10/N16	LC1		0	0	0	0	0	0
Sn11/N17	LC1		0	0	0	0	0	0
Sn12/N18	LC1		0	0	0	0	0	0
Sn13/N19	LC1		0	0	0	0	0	0
Sn14/N2	LC1		0	0	0	0	0	0
Sn15/N21	LC1		0	0	0	0	0	0
Sn16/N2	LC1		0	0	0	0	0	0
Sn17/N2	LC1		0	0	0	0	0	0
Sn18/N2	LC1		0	0	0	0	0	0
Sn19/N2	LC1		0	-174.7	0	2569.1	0	0
Sn20/N2	LC1		0	0	0	0	0	0
Sn21/N2	LC1		0	0	0	0	0	0
Sn22/N2	LC1		0	0	0	0	0	0
Sn23/N2	LC1		0	0	0	0	0	0
Sn24/N3	LC1		0	0	0	0	0	0
Sn25/N3	LC1		0	0	0	0	0	0
Sn26/N3	LC1		0	0	0	0	0	0
Sn27/N3	LC1		0	0	0	0	0	0
Sn28/N3	LC1		0	0	0	0	0	0
Sn29/N3	LC1		0	-26.29	0	734.56	0	0
Sn30/N3	LC1		0	0	0	0	0	0
Sn31/N3	LC1		0	0	0	0	0	0
Sn32/N3	LC1		0	0	0	0	0	0
Sn33/N3	LC1		0	0	0	0	0	0
Sn34/N4	LC1		0	0	0	0	0	0

2. 3D stress

Linear calculation
Load case: LC1
Selection: All
Location: In nodes avg. on macro. System: LCS mesh element
Principal magnitudes
Results on 1D member
Extreme 1D: Member

Name	dx [m]	Fibre	Case	σ_{E} [MPa]	σ_{1} [MPa]	σ_{2} [MPa]	τ_{xy} / τ_{xs} [MPa]	τ_{xz} / τ_{xs} [MPa]
B1	0	1	LC1	1.1	0.6	-0.6	0.6	0
B1	29.746-	16	LC1	66.3	66.3	-0.1	2.6	0
B2	0	1	LC1	15.3	8.8	-8.8	-8.8	0
B2	40	16	LC1	21.2	4.6	-18.5	-9.3	0
B3	40	15	LC1	9.5	4.4	-6.5	-5.4	0
B3	0	16	LC1	16.7	1.8	-15.7	-5.4	0
B4	40	6	LC1	2.7	1.5	-1.5	-1.5	0
B4	0	16	LC1	3.4	0.8	-2.9	-1.5	0
B5	0	1	LC1	0	0	0	0	0
B6	0	1	LC1	0	0	0	0	0
B7	0	11	LC1	2.3	1.3	-1.3	1.3	0
B7	6.7	6	LC1	30	30	0	0	0
B8	0	11	LC1	2.3	1.3	-1.3	1.3	0
B8	2.3	6	LC1	37.6	37.6	0	0	0
B9	0	11	LC1	2.3	1.3	-1.3	1.3	0
B9	3	6	LC1	47.6	47.6	0	0	0
B10	0	11	LC1	2.3	1.3	-1.3	1.3	0
B10	3	16	LC1	57.5	0	-57.5	0	0
B11	0	11	LC1	2.3	1.3	-1.3	1.3	0
B11	1.1	16	LC1	61.2	0	-61.2	0	0
B12	0	11	LC1	2.3	1.3	-1.3	1.3	0
B12	0.9	16	LC1	64.1	0	-64.1	0	0
B13	0	11	LC1	2.3	1.3	-1.3	1.3	0
B13	1	16	LC1	67.4	0	-67.4	0	0
B14	0	11	LC1	2.3	1.3	-1.3	1.3	0
B14	0.553	6	LC1	69.3	69.3	0	0	0
B15	0	1	LC1	0	0	0	0	0
B16	0	1	LC1	0	0	0	0	0
B17	5.025	8	LC1	3.5	3.5	-0.1	0.6	0
B17	0	16	LC1	35.9	35.9	0	0	0
B18	0	11	LC1	5.6	3.2	-3.2	3.2	0
B18	2.3	6	LC1	35.5	35.5	0	0	0
B19	0	11	LC1	5.6	3.2	-3.2	3.2	0
B19	3	16	LC1	59.2	0	-59.2	0	0
B20	0	11	LC1	5.6	3.2	-3.2	3.2	0
B20	3	16	LC1	83	0	-83	0	0

Blad1 Blad2 Blad3 Blad4 Blad5 Blad6 Blad7 Blad8 Blad9 Blad10 Blad11 Blad12 Blad13 Blad14 Blad15 Blad16

Figure E.8 Results file translation to SCIA Engineer

E.2 Possible applications of the translation to SCIA Engineer

The main advantage and reason for the translation to SCIA Engineer is the automatic post-processing of the obtained results. By showing the principal stresses, it is possible to quickly assess the investigated design. The translation to SCIA is thus mainly beneficial for automation design process and staying ahead of competitors as contractor.

On top of that, the translation of the Python model to SCIA Engineer gives rise to more flexibility. First of all, a variation in wall thickness for every vertical pile can be implemented. Secondly, difference can be made between the first pile, end pile and middle piles. Finally, the number of vertical piles can be changed by specifying the number of fields.

E.3 Limitations of the translation to SCIA Engineer

Despite the fact that the translation of the Python model to SCIA overcomes several limitations of the Python model, the tool involving SCIA Engineer also has some short-comings. These shortcomings are discussed within this subsection.

First thing that needs to be remarked is the computational time. Due to the importing and exporting between SCIA and Excel of different .XML files and the computation in SCIA Engineer itself, every timestep takes approximately 30 seconds. To only simulate the first impact of an average brush collision (≈6 seconds) already requires 6 sec/0.1 sec/timestep = 60 timesteps = 1800 seconds = 30 minutes.

Incorporation of the second impact and the interval between these two impacts in the analysis would require even more time. Consequently, only the first impact is incorporated in the translation to SCIA Engineer. Nonetheless, the simulation of only this first impact is still able to demonstrate the capacities of the translation to SCIA Engineer

A second limitation of the translation to SCIA is the lack of transparency of the model; after all input parameters are defined, the ‘user’ clicks on a button that starts the analysis as a ‘black box’. As a result of the limited available time to create this translation, no warnings are built-in in case of erroneous input, e.g. more than six fields. The model is then still able to produce output, but the output is meaningless.

E.4 Verification of the translation to SCIA Engineer

In order to check whether the translation to SCIA Engineer produces correct results, a verification is performed. By using the same timesteps and input parameters as done in chapter 5, the results of the Python model and SCIA Engineer are compared with each other. The investigated output consists of: contact force, deformation of the contact point, the rotational movement and velocity and the translational velocity of the vessel.

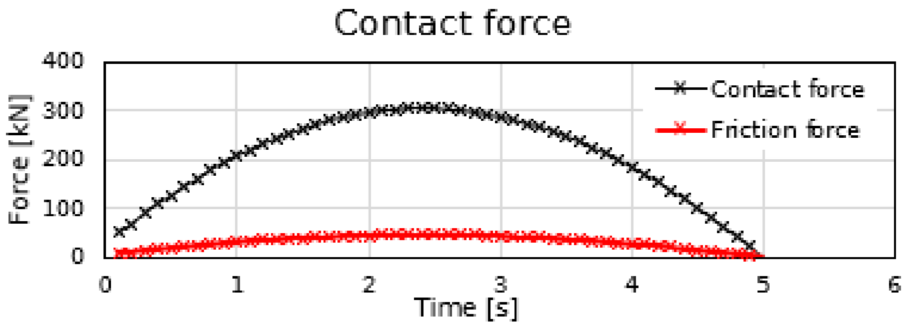


Figure E.9 Contact force translation to SCIA Engineer

Figure E.9 shows the contact force and the friction force during the first impact of a brush collision. Similar to the model in Python, a sinusoidal behaviour is found for the friction and the contact force between the vessel and the structure. The maximum contact force found using the SCIA model is 305.6 kN, which is 16% higher than in the Python model.

Furthermore, it can be seen that the duration of the first impact of the brush collision is circa 5% shorter than in the Python model. This means that the model in SCIA Engineer reacts somewhat stiffer, which simultaneously explains why the contact force is higher in SCIA than in the Python model. This stiffer behaviour might be induced by the rigid connection of the horizontal tube with the vertical piles, leading to torsional effects.

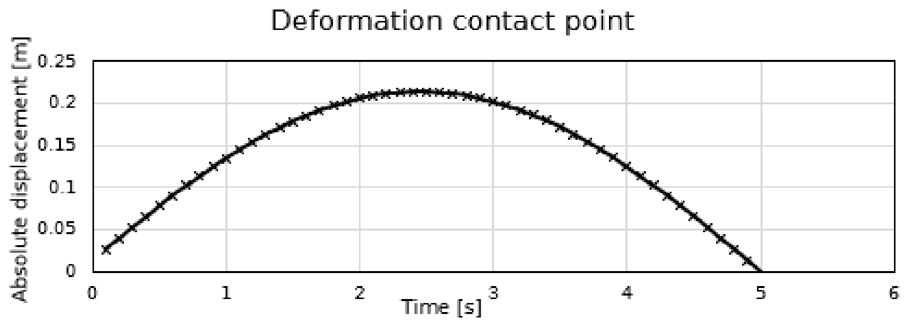


Figure E.10 Deformation contact point translation to SCIA Engineer

For the deformation of the contact point also a sinusoidal is found, in correspondence with the results of the Python model. The maximum deformation of the Python model is 0.223 meters, while the maximum deformation using SCIA Engineer is found to be 0.214 meters, resulting in a difference of circa 4%.

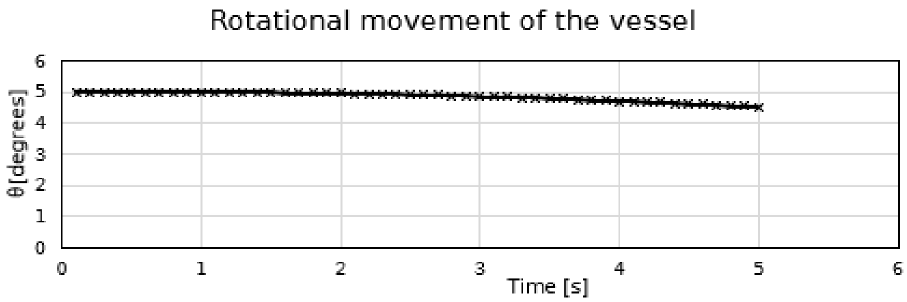


Figure E.11 Rotational movement vessel translation to SCIA Engineer

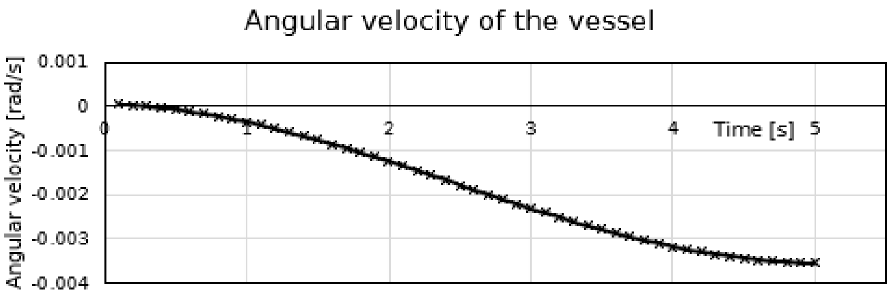


Figure E.12 Rotational velocity vessel translation to SCIA Engineer

Subsequently, the rotational movement and velocity is investigated during the first impact, see Figure E.11 and Figure E.12 respectively. The results in the graphs in Figure E.11 and Figure E.12 show great resemblance with the results of Figure 5.8 and Figure 5.9, both in behaviour and numerical values.

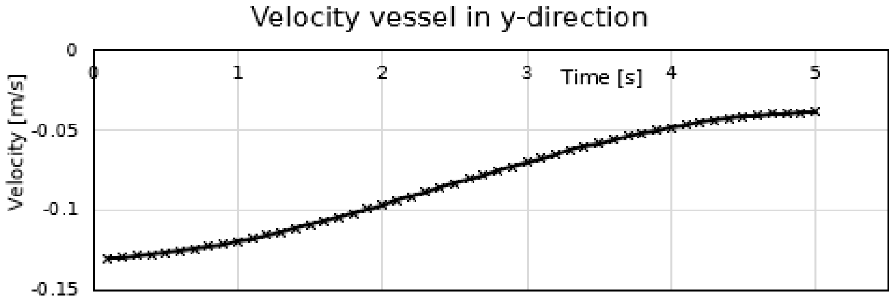


Figure E.13 Velocity vessel y-direction translation to SCIA Engineer

Finally, the velocity in y-direction of the mass centre of the vessel is plotted in Figure E.13. Similar to Figure 5.7, during the first impact the translational velocity is, in absolute terms, continuously decreasing but does not change of sign.

Based on the results in Figure E.9 to Figure E.10, it can be stated that the translation to SCIA Engineer leads to comparable results. Since the Python model is reviewed as sufficiently accurate, this classification also holds for the translation of the Python model to SCIA Engineer.

E.5 Sensitivity analysis of the translation to SCIA Engineer

Important characteristic of an analysis is its sensitivity with respect to the applied time steps. The model involving SCIA is run with timesteps of 0.1, 0.2, 0.4, 0.8 and 1.2 seconds, using the same parameters as in section E.5. In Figure E.14, the obtained contact force is presented in the time domain for the different used timesteps.

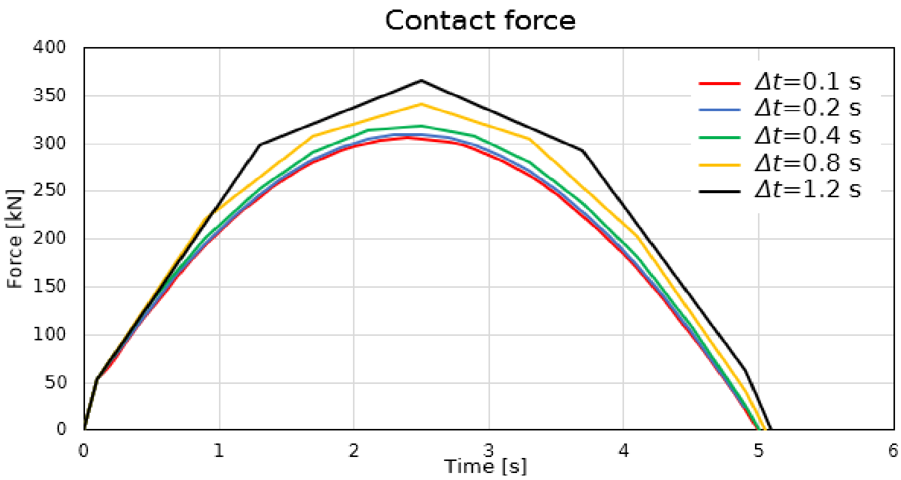


Figure E.14 Sensitivity analysis SCIA model

When the results using timesteps of 0.1 seconds in Figure E.14 are used as a reference to compare the obtained results of the other time steps, it can be seen that an increase of the timestep leads to an increase in the overestimation of the contact force.

Application of timesteps of 0.2 and 0.4 seconds lead to results that are still within a 5% range of the reference results, making these magnitudes of timesteps an attractive alternative for the used timesteps of 0.1 seconds in section E.4; the duration of the analysis can then be reduced by 50%, or even 75% respectively.

Making use of timesteps of 0.8 or 1.2 seconds already leads to an overprediction of the contact force of 12% and 20% respectively. As a result, application of timesteps up to 0.4 seconds is considered appropriate, while application larger timesteps than 0.4 seconds is considered as a too coarse approximation of the ‘correct’ results.

E.6 Discussion, conclusions and recommendations of the translation to SCIA

From the performed analysis in this appendix, it follows that the results of the model involving SCIA are in line with the results obtained using the Python model. In chapter 5, a verification of the Python model is performed and the results are assessed to be valid.

The model in SCIA Engineer shows a slightly stiffer behaviour than the Python model. This increased stiffness is probably due to the rigid connections between the horizontal berthing beam and the vertical piles, resulting in the incorporation of torsional effects on the force distribution. Due to the stiffer response the maximum occurring contact force is circa 15% higher and the duration of the impact is approximately 5% shorter.

Furthermore, the behaviour of the displacements and the velocities in the time domain of both models are in accordance with each other.

A significant disadvantage of the translation to SCIA Engineer is the duration of the analysis. In the Python model 900 timesteps of 0.1 seconds are performed in less than 1 second, while the model using SCIA Engineering requires 30 seconds *per timestep* resulting in a total analysis time of 7.5 hours.

Given the automatic post-processing environment of the software, it was expected that the design process of guide works could be accelerated. In its current form, however, the tool involving SCIA is not suitable to use as a design tool, due to its computational time.

Despite the limitations mentioned in the previous paragraph, it is expected that the translation to SCIA Engineer can be made suitable for implementation. A possibility to reduce the computational time is to make use of larger time steps. Another option to limit the computational efforts is to ‘skip’ the analysis in SCIA and only update the position of the vessel, in case there is no contact between the vessel and the structure. Additionally, a proper description of this tool needs to be created in case the tool with SCIA is going to be implemented in the design practice, in order to avoid erroneous input.

Subsequently, it can be concluded that it is possible to translate the model to SCIA Engineer, but there are still some practicalities to overcome. ‘Tricks’ as described in the previous paragraph are to be implemented in the future to reduce the computation time of the model, and a proper user manual is necessary to enable the usage of this method in future projects.

As long as the practicalities are not overcome yet, it is recommended to use the model that is created in Python. From the Python model, the maximum occurring contact force can be obtained, which can then be treated as static design force for the guide work.

Once the static design contact force is known, it *does* become possible to use SCIA Engineer or other FEM software to verify the design that is proposed, by making use of the software program in a conventional way.

Finally, it can be stated that the model to simulate brush collisions in SCIA is able to simulate berthing manoeuvres as well, when small adaptions are made, similar to the Python model.

Appendix F: Python Script

1. Import of packages

```
import numpy as np
import matplotlib.pyplot as plt
import math
from sympy import *
from scipy import integrate
from scipy.integrate import quad
import ipywidgets as widgets
from ipywidgets import interact
from timeit import default_timer as timer
```

2. Geometry input

```
D1       = 0.9
t1       = 0.04
D2       = 1.82
t2       = 0.013
L_hor    = 40.0
L_ver_0  = 17.0
L_ver    = L_ver_0
N_piles  = 5
N_beams  = 1
C_m      = 1.1
x_V      = 0.8*L_hor
x_V_0    = x_V
z_V      = 0.0
```

3. Soil input

```
def k_trans(F):
    k_trans = 6E+11*F**-0.668
    return k_trans

def k_rot(F):
    k_rot = 5E+10*F**-0.278
    return k_rot

def depth(F):
    t = 0.0158*F**0.4115
    return t
```

4. Ship input

```
m             = 10E6
m0            = m
draught       = 4.0
L_ship        = 100.3
B_ship        = 22.8
H_ship        = 10.0
theta_degree  = 5.0
v_x_C_0       = -1.5
```

5. Other properties

```
E            = 210E9
mu           = 0.15
rho_steel    = 7850
time_step_0  = 0.01
time_step    = 0.1
del_t_after  = 0.1
rho_water    = 1000
C_w          = 2.0

F            = 1E3         #INITIAL VALUE FOR F, TO DETERMINE STIFFNESSES AND DEPTH
F_0          = 1E3         #INITIAL VALUE FOR F, TO DETERMINE STIFFNESSES AND DEPTH
F_x          = 0
```

6. Derived properties

```
M  = (0.25*math.pi*(D1**2-(D1-2*t1)**2)*L_ver_0*N_piles + \
      0.25*math.pi*(D2**2-(D2-2*t2)**2)*L_hor*N_beams)*rho_steel

Inertia = 1/12*m*L_ship**2
N_segments=50
C_a = 1.5476*(draught/(B_ship*0.5))**-0.149;
L_segment=L_ship/N_segments; m_segment = m/N_segments
Inertia_segment=(m_segment+math.pi*draught**2*rho_water/2*C_a*L_segment)
Inertia_sum = 0
for i in range(int(0.5*N_segments)):
    arm=(0.5+i)*L_segment
    Inertia_sum = Inertia_sum+Inertia_segment*arm**2
Ig = Inertia_sum*2

#----------------------------------------------------------------------
I1  = math.pi/64*(D1**4-(D1-2*t1)**4)
I2  = math.pi/64*(D2**4-(D2-2*t2)**4)
EI1 = E*I1
EI2 = E*I2

#----------------------------------------------------------------------
for i in range(5):
    globals()[str('L_ver{}'.format(i+1))] = L_ver_0 + depth(F)
    globals()[str('k_pile{}'.format(i+1))]                                  =
3*EI1/(globals()[str('L_ver{}'.format(i+1))])**3
    globals()[str('k_tot{}'.format(i+1))] = \
                    (1/(globals()[str('k_pile{}'.format(i+1))]) + \
                    (globals()[str('L_ver{}'.format(i+1))])**2/(k_rot(F)) +\
                    1/(k_trans(F)))**(-1)

#----------------------------------------------------------------------
theta      = (theta_degree)/180*math.pi

#----------------------------------------------------------------------
a = 0.5*(L_ship*math.cos(theta) - B_ship*math.sin(theta))
b = 0.5*(L_ship*math.sin(theta) + B_ship*math.cos(theta))
c = 0.5*(-L_ship*math.sin(abs(theta)) + B_ship*math.cos(abs(theta)))
d = 0.5*(L_ship*math.cos(abs(theta)) + B_ship*math.sin(abs(theta)))

#----------------------------------------------------------------------
time = 0
E_friction = 0
E_drag_rot = 0
E_drag_trans = 0
x_C = x_V + a
z_C = z_V - b
x_A = x_C + a
z_A = z_C + c

#----------------------------------------------------------------------
v_x_V_0 = v_x_C_0
v_z_C_0 = -v_x_C_0*math.tan(theta)

v_z_V_0 = v_z_C_0

m_x = m + (C_m-1)*math.cos(theta)*m
m_z = m + (C_m-1)*math.sin(theta)*m
m =m * C_m
```

7. First impact

```
theta_dot_0 = 0.00
S = (a*theta_dot_0-v_z_C_0)/(1/m+1/M-a*(mu*b-a)/Ig)
theta_dot = -S*(mu*b-a)/Ig+theta_dot_0

v_z_V     = -S/M
v_z_C     = S/m+v_z_C_0

v_x_C     = -mu*S/m + v_x_C_0
v_x_V     = v_x_C + b*theta_dot

v_x_A     = v_x_C -c*theta_dot
v_z_A     = v_z_C -a*theta_dot

x_RA = x_A - B_ship*math.sin(theta)
```

```
z_RA = z_A - B_ship*math.cos(theta)
x_RV = x_V - B_ship*math.sin(theta)
z_RV = z_V - B_ship*math.cos(theta)
```

8. Create matrix

```
def make_Matrix_seg1(k1,k2,k3,k4,k5,EI2,L_hor,x_pos):
    for i in range(20):
        if i == 0: #eq1
            Matrix1[i,0]=EI2/k1
            Matrix1[i,3]=1
        if i == 1: #eq2
            Matrix1[i,1]=-EI2
        if i == 2: #eq3
            Matrix1[i,0]=-EI2*x_pos
            Matrix1[i,1]=-EI2
            Matrix1[i,4]=EI2*x_pos
            Matrix1[i,5]=EI2
        if i == 3: #eq4
            Matrix1[i,0]=1.0/6*x_pos**3
            Matrix1[i,1]=0.5*x_pos**2
            Matrix1[i,2]=x_pos
            Matrix1[i,3]=1
            Matrix1[i,4]=-1.0/6*x_pos**3
            Matrix1[i,5]=-0.5*x_pos**2
            Matrix1[i,6]=-x_pos
            Matrix1[i,7]=-1
        if i == 4: #eq5
            Matrix1[i,0]=-0.5*x_pos**2
            Matrix1[i,1]=-x_pos
            Matrix1[i,2]=-1
            Matrix1[i,4]=0.5*x_pos**2
            Matrix1[i,5]=x_pos
            Matrix1[i,6]=1
        if i == 5: #eq6
            Matrix1[i,4]=1/6.0*x_pos**3
            Matrix1[i,5]=0.5*x_pos**2
            Matrix1[i,6]=x_pos
            Matrix1[i,7]=1
        if i == 6: #eq7
            Matrix1[i,4]=-EI2*L_hor
            Matrix1[i,5]=-EI2
            Matrix1[i,8]=EI2*L_hor
            Matrix1[i,9]=EI2
        if i == 7: #eq8
            Matrix1[i,4]=1.0/6*L_hor**3
            Matrix1[i,5]=0.5*L_hor**2
            Matrix1[i,6]=L_hor
            Matrix1[i,7]=1
            Matrix1[i,8]=-1.0/6*L_hor**3
            Matrix1[i,9]=-0.5*L_hor**2
            Matrix1[i,10]=-L_hor
            Matrix1[i,11]=-1
        if i == 8: #eq9
            Matrix1[i,4]=-0.5*L_hor**2
            Matrix1[i,5]=-L_hor
            Matrix1[i,6]=-1
            Matrix1[i,8]=0.5*L_hor**2
            Matrix1[i,9]=L_hor
            Matrix1[i,10]=1
        if i == 9: #eq10
            Matrix1[i,4]=-EI2+k2*1.0/6*L_hor**3
            Matrix1[i,5]=0.5*k2*L_hor**2
            Matrix1[i,6]=L_hor*k2
            Matrix1[i,7]=k2
            Matrix1[i,8]=EI2
        if i == 10: #eq11
            Matrix1[i,8]=-2*L_hor*EI2
            Matrix1[i,9]=-EI2
            Matrix1[i,12]=2*L_hor*EI2
            Matrix1[i,13]=EI2
        if i == 11: #eq12
            Matrix1[i,8]=4.0/3*L_hor**3
            Matrix1[i,9]=2.0*L_hor**2
            Matrix1[i,10]=2*L_hor
            Matrix1[i,11]=1
```

```
            Matrix1[i,12]=-4.0/3*L_hor**3
            Matrix1[i,13]=-2.0*L_hor**2
            Matrix1[i,14]=-2.0*L_hor
            Matrix1[i,15]=-1
        if i == 12: #eq13
            Matrix1[i,8]=-2*L_hor**2
            Matrix1[i,9]=-2*L_hor
            Matrix1[i,10]=-1
            Matrix1[i,12]=2*L_hor**2
            Matrix1[i,13]=2*L_hor
            Matrix1[i,14]=1
        if i == 13: #eq14
            Matrix1[i,8]=-EI2+k3*4.0/3*L_hor**3
            Matrix1[i,9]=2*k3*L_hor**2
            Matrix1[i,10]=2*L_hor*k3
            Matrix1[i,11]=k3
            Matrix1[i,12]=EI2
        if i == 14: #eq15
            Matrix1[i,12]=-EI2*3*L_hor
            Matrix1[i,13]=-EI2
            Matrix1[i,16]=EI2*3*L_hor
            Matrix1[i,17]=EI2
        if i == 15: #eq16
            Matrix1[i,12]=4.5*L_hor**3
            Matrix1[i,13]=4.5*L_hor**2
            Matrix1[i,14]=3*L_hor
            Matrix1[i,15]=1
            Matrix1[i,16]=-4.5*L_hor**3
            Matrix1[i,17]=-4.5*L_hor**2
            Matrix1[i,18]=-3*L_hor
            Matrix1[i,19]=-1
        if i == 16: #eq17
            Matrix1[i,12]=-4.5*L_hor**2
            Matrix1[i,13]=-3*L_hor
            Matrix1[i,14]=-1
            Matrix1[i,16]=4.5*L_hor**2
            Matrix1[i,17]=3*L_hor
            Matrix1[i,18]=1
        if i == 17: #eq18
            Matrix1[i,12]=-EI2+k4*4.5*L_hor**3
            Matrix1[i,13]=k4*4.5*L_hor**2
            Matrix1[i,14]=k4*3*L_hor
            Matrix1[i,15]=k4
            Matrix1[i,16]=EI2
        if i == 18: #eq19
            Matrix1[i,16]=-EI2/k5+32/3.0*L_hor**3
            Matrix1[i,17]=8*L_hor**2
            Matrix1[i,18]=4*L_hor
            Matrix1[i,19]=1
        if i == 19: #eq20
            Matrix1[i,16]=-4*EI2*L_hor
            Matrix1[i,17]=-EI2
#2222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
22222222222
def make_Matrix_seg2(k1,k2,k3,k4,k5,EI2,L_hor,x_pos):
    for i in range(20):
        if i == 0: #eq1
            Matrix2[i,0]=EI2/k1
            Matrix2[i,3]=1
        if i == 1: #eq2
            Matrix2[i,1]=-EI2
        if i == 2: #eq3
            Matrix2[i,0]=-EI2*L_hor
            Matrix2[i,1]=-EI2
            Matrix2[i,4]=EI2*L_hor
            Matrix2[i,5]=EI2
        if i == 3: #eq4
            Matrix2[i,0]=1.0/6*L_hor**3
            Matrix2[i,1]=1/2*L_hor**2
            Matrix2[i,2]=L_hor
            Matrix2[i,3]=1
            Matrix2[i,4]=-1.0/6*L_hor**3
            Matrix2[i,5]=-1/2.0*L_hor**2
            Matrix2[i,6]=-L_hor
            Matrix2[i,7]=-1
        if i == 4: #eq5
            Matrix2[i,0]=-0.5*L_hor**2
```

```
            Matrix2[i,1]=-L_hor
            Matrix2[i,2]=-1
            Matrix2[i,4]=0.5*L_hor**2
            Matrix2[i,5]=L_hor
            Matrix2[i,6]=1
        if i == 5: #eq6
            Matrix2[i,0]=-EI2
            Matrix2[i,4]=k2*1/6.0*L_hor**3+EI2
            Matrix2[i,5]=k2*0.5*L_hor**2
            Matrix2[i,6]=k2*L_hor
            Matrix2[i,7]=k2
        if i == 6: #eq7
            Matrix2[i,4]=-EI2*x_pos
            Matrix2[i,5]=-EI2
            Matrix2[i,8]=EI2*x_pos
            Matrix2[i,9]=EI2
        if i == 7: #eq8
            Matrix2[i,4]=1.0/6*x_pos**3
            Matrix2[i,5]=0.5*x_pos**2
            Matrix2[i,6]=x_pos
            Matrix2[i,7]=1
            Matrix2[i,8]=-1.0/6*x_pos**3
            Matrix2[i,9]=-0.5*x_pos**2
            Matrix2[i,10]=-x_pos
            Matrix2[i,11]=-1
        if i == 8: #eq9
            Matrix2[i,4]=-0.5*x_pos**2
            Matrix2[i,5]=-x_pos
            Matrix2[i,6]=-1
            Matrix2[i,8]=0.5*x_pos**2
            Matrix2[i,9]=x_pos
            Matrix2[i,10]=1
        if i == 9: #eq10
            Matrix2[i,4]=1.0/6*x_pos**3
            Matrix2[i,5]=0.5*x_pos**2
            Matrix2[i,6]=x_pos
            Matrix2[i,7]=1
        if i == 10: #eq11
            Matrix2[i,8]=-2*L_hor*EI2
            Matrix2[i,9]=-EI2
            Matrix2[i,12]=2*L_hor*EI2
            Matrix2[i,13]=EI2
        if i == 11: #eq12
            Matrix2[i,8]=4.0/3*L_hor**3
            Matrix2[i,9]=2.0*L_hor**2
            Matrix2[i,10]=2*L_hor
            Matrix2[i,11]=1
            Matrix2[i,12]=-4.0/3*L_hor**3
            Matrix2[i,13]=-2.0*L_hor**2
            Matrix2[i,14]=-2.0*L_hor
            Matrix2[i,15]=-1
        if i == 12: #eq13
            Matrix2[i,8]=-2*L_hor**2
            Matrix2[i,9]=-2*L_hor
            Matrix2[i,10]=-1
            Matrix2[i,12]=2*L_hor**2
            Matrix2[i,13]=2*L_hor
            Matrix2[i,14]=1
        if i == 13: #eq14
            Matrix2[i,8]=-EI2+k3*4.0/3*L_hor**3
            Matrix2[i,9]=2*k3*L_hor**2
            Matrix2[i,10]=2*L_hor*k3
            Matrix2[i,11]=k3
            Matrix2[i,12]=EI2
        if i == 14: #eq15
            Matrix2[i,12]=-EI2*3*L_hor
            Matrix2[i,13]=-EI2
            Matrix2[i,16]=EI2*3*L_hor
            Matrix2[i,17]=EI2
        if i == 15: #eq16
            Matrix2[i,12]=4.5*L_hor**3
            Matrix2[i,13]=4.5*L_hor**2
            Matrix2[i,14]=3*L_hor
            Matrix2[i,15]=1
            Matrix2[i,16]=-4.5*L_hor**3
            Matrix2[i,17]=-4.5*L_hor**2
            Matrix2[i,18]=-3*L_hor
```

```
            Matrix2[i,19]=-1
        if i == 16: #eq17
            Matrix2[i,12]=-4.5*L_hor**2
            Matrix2[i,13]=-3*L_hor
            Matrix2[i,14]=-1
            Matrix2[i,16]=4.5*L_hor**2
            Matrix2[i,17]=3*L_hor
            Matrix2[i,18]=1
        if i == 17: #eq18
            Matrix2[i,12]=-EI2+k4*4.5*L_hor**3
            Matrix2[i,13]=k4*4.5*L_hor**2
            Matrix2[i,14]=k4*3*L_hor
            Matrix2[i,15]=k4
            Matrix2[i,16]=EI2
        if i == 18: #eq19
            Matrix2[i,16]=-EI2/k5+32/3.0*L_hor**3
            Matrix2[i,17]=8*L_hor**2
            Matrix2[i,18]=4*L_hor
            Matrix2[i,19]=1
        if i == 19: #eq20
            Matrix2[i,16]=-4*EI2*L_hor
            Matrix2[i,17]=-EI2
#333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
def make_Matrix_seg3(k1,k2,k3,k4,k5,EI2,L_hor,x_pos):
    for i in range(20):
        if i == 0: #eq1
            Matrix3[i,0]=EI2/k1
            Matrix3[i,3]=1
        if i == 1: #eq2
            Matrix3[i,1]=-EI2
        if i == 2: #eq3
            Matrix3[i,0]=-EI2*L_hor
            Matrix3[i,1]=-EI2
            Matrix3[i,4]=EI2*L_hor
            Matrix3[i,5]=EI2
        if i == 3: #eq4
            Matrix3[i,0]=1.0/6*L_hor**3
            Matrix3[i,1]=1/2*L_hor**2
            Matrix3[i,2]=L_hor
            Matrix3[i,3]=1
            Matrix3[i,4]=-1.0/6*L_hor**3
            Matrix3[i,5]=-1/2.0*L_hor**2
            Matrix3[i,6]=-L_hor
            Matrix3[i,7]=-1
        if i == 4: #eq5
            Matrix3[i,0]=-0.5*L_hor**2
            Matrix3[i,1]=-L_hor
            Matrix3[i,2]=-1
            Matrix3[i,4]=0.5*L_hor**2
            Matrix3[i,5]=L_hor
            Matrix3[i,6]=1
        if i == 5: #eq6
            Matrix3[i,0]=-EI2
            Matrix3[i,4]=k2*1/6.0*L_hor**3+EI2
            Matrix3[i,5]=k2*0.5*L_hor**2
            Matrix3[i,6]=k2*L_hor
            Matrix3[i,7]=k2
        if i == 6: #eq7
            Matrix3[i,4]=-2*EI2*L_hor
            Matrix3[i,5]=-EI2
            Matrix3[i,8]=2*EI2*L_hor
            Matrix3[i,9]=EI2
        if i == 7: #eq8
            Matrix3[i,4]=4.0/3*L_hor**3
            Matrix3[i,5]=2*L_hor**2
            Matrix3[i,6]=2*L_hor
            Matrix3[i,7]=1
            Matrix3[i,8]=-4.0/3*L_hor**3
            Matrix3[i,9]=-2*L_hor**2
            Matrix3[i,10]=-2*L_hor
            Matrix3[i,11]=-1
        if i == 8: #eq9
            Matrix3[i,4]=-2*L_hor**2
            Matrix3[i,5]=-2*L_hor
            Matrix3[i,6]=-1
            Matrix3[i,8]=2*L_hor**2
            Matrix3[i,9]=2*L_hor
```

```
            Matrix3[i,10]=1
        if i == 9: #eq10
            Matrix3[i,4]=-EI2+k3*4.0/3*L_hor**3
            Matrix3[i,5]=2*k3*L_hor**2
            Matrix3[i,6]=2*L_hor*k3
            Matrix3[i,7]=k3
            Matrix3[i,8]=EI2
        if i == 10: #eq11
            Matrix3[i,8]=-EI2*x_pos
            Matrix3[i,9]=-EI2
            Matrix3[i,12]=EI2*x_pos
            Matrix3[i,13]=EI2
        if i == 11: #eq12
            Matrix3[i,8]=1.0/6*x_pos**3
            Matrix3[i,9]=0.5*x_pos**2
            Matrix3[i,10]=x_pos
            Matrix3[i,11]=1
            Matrix3[i,12]=-1.0/6*x_pos**3
            Matrix3[i,13]=-0.5*x_pos**2
            Matrix3[i,14]=-x_pos
            Matrix3[i,15]=-1
        if i == 12: #eq13
            Matrix3[i,8]=-0.5*x_pos**2
            Matrix3[i,9]=-x_pos
            Matrix3[i,10]=-1
            Matrix3[i,12]=0.5*x_pos**2
            Matrix3[i,13]=x_pos
            Matrix3[i,14]=1
        if i == 13: #eq14
            Matrix3[i,8]=1.0/6*x_pos**3
            Matrix3[i,9]=0.5*x_pos**2
            Matrix3[i,10]=x_pos
            Matrix3[i,11]=1
        if i == 14: #eq15
            Matrix3[i,12]=-EI2*3*L_hor
            Matrix3[i,13]=-EI2
            Matrix3[i,16]=EI2*3*L_hor
            Matrix3[i,17]=EI2
        if i == 15: #eq16
            Matrix3[i,12]=4.5*L_hor**3
            Matrix3[i,13]=4.5*L_hor**2
            Matrix3[i,14]=3*L_hor
            Matrix3[i,15]=1
            Matrix3[i,16]=-4.5*L_hor**3
            Matrix3[i,17]=-4.5*L_hor**2
            Matrix3[i,18]=-3*L_hor
            Matrix3[i,19]=-1
        if i == 16: #eq17
            Matrix3[i,12]=-4.5*L_hor**2
            Matrix3[i,13]=-3*L_hor
            Matrix3[i,14]=-1
            Matrix3[i,16]=4.5*L_hor**2
            Matrix3[i,17]=3*L_hor
            Matrix3[i,18]=1
        if i == 17: #eq18
            Matrix3[i,12]=-EI2+k4*4.5*L_hor**3
            Matrix3[i,13]=k4*4.5*L_hor**2
            Matrix3[i,14]=k4*3*L_hor
            Matrix3[i,15]=k4
            Matrix3[i,16]=EI2
        if i == 18: #eq19
            Matrix3[i,16]=-EI2/k5+32/3.0*L_hor**3
            Matrix3[i,17]=8*L_hor**2
            Matrix3[i,18]=4*L_hor
            Matrix3[i,19]=1
        if i == 19: #eq20
            Matrix3[i,16]=-4*EI2*L_hor
            Matrix3[i,17]=-EI2
#4444444444444444444444444444444444444444444444444444444444444444444444444444
def make_Matrix_seg4(k1,k2,k3,k4,k5,EI2,L_hor,x_pos):
    for i in range(20):
        if i == 0: #eq1
            Matrix4[i,0]=EI2/k1
            Matrix4[i,3]=1
        if i == 1: #eq2
            Matrix4[i,1]=-EI2
        if i == 2: #eq3
```

```
            Matrix4[i,0]=-EI2*L_hor
            Matrix4[i,1]=-EI2
            Matrix4[i,4]=EI2*L_hor
            Matrix4[i,5]=EI2
        if i == 3: #eq4
            Matrix4[i,0]=1.0/6*L_hor**3
            Matrix4[i,1]=1/2*L_hor**2
            Matrix4[i,2]=L_hor
            Matrix4[i,3]=1
            Matrix4[i,4]=-1.0/6*L_hor**3
            Matrix4[i,5]=-1/2.0*L_hor**2
            Matrix4[i,6]=-L_hor
            Matrix4[i,7]=-1
        if i == 4: #eq5
            Matrix4[i,0]=-0.5*L_hor**2
            Matrix4[i,1]=-L_hor
            Matrix4[i,2]=-1
            Matrix4[i,4]=0.5*L_hor**2
            Matrix4[i,5]=L_hor
            Matrix4[i,6]=1
        if i == 5: #eq6
            Matrix4[i,0]=-EI2
            Matrix4[i,4]=k2*1/6.0*L_hor**3+EI2
            Matrix4[i,5]=k2*0.5*L_hor**2
            Matrix4[i,6]=k2*L_hor
            Matrix4[i,7]=k2
        if i == 6: #eq7
            Matrix4[i,4]=-2*EI2*L_hor
            Matrix4[i,5]=-EI2
            Matrix4[i,8]=2*EI2*L_hor
            Matrix4[i,9]=EI2
        if i == 7: #eq8
            Matrix4[i,4]=4.0/3*L_hor**3
            Matrix4[i,5]=2*L_hor**2
            Matrix4[i,6]=2*L_hor
            Matrix4[i,7]=1
            Matrix4[i,8]=-4.0/3*L_hor**3
            Matrix4[i,9]=-2*L_hor**2
            Matrix4[i,10]=-2*L_hor
            Matrix4[i,11]=-1
        if i == 8: #eq9
            Matrix4[i,4]=-2*L_hor**2
            Matrix4[i,5]=-2*L_hor
            Matrix4[i,6]=-1
            Matrix4[i,8]=2*L_hor**2
            Matrix4[i,9]=2*L_hor
            Matrix4[i,10]=1
        if i == 9: #eq10
            Matrix4[i,4]=-EI2+k3*4.0/3*L_hor**3
            Matrix4[i,5]=2*k3*L_hor**2
            Matrix4[i,6]=2*L_hor*k3
            Matrix4[i,7]=k3
            Matrix4[i,8]=EI2
        if i == 10: #eq11
            Matrix4[i,8]=-3*L_hor*EI2
            Matrix4[i,9]=-EI2
            Matrix4[i,12]=3*L_hor*EI2
            Matrix4[i,13]=EI2
        if i == 11: #eq12
            Matrix4[i,8]=4.5*L_hor**3
            Matrix4[i,9]=4.5*L_hor**2
            Matrix4[i,10]=3*L_hor
            Matrix4[i,11]=1
            Matrix4[i,12]=-4.5*L_hor**3
            Matrix4[i,13]=-4.5*L_hor**2
            Matrix4[i,14]=-3*L_hor
            Matrix4[i,15]=-1
        if i == 12: #eq13
            Matrix4[i,8]=-4.5*L_hor**2
            Matrix4[i,9]=-3*L_hor
            Matrix4[i,10]=-1
            Matrix4[i,12]=4.5*L_hor**2
            Matrix4[i,13]=3*L_hor
            Matrix4[i,14]=1
        if i == 13: #eq14
            Matrix4[i,8]=-EI2+k4*4.5*L_hor**3
            Matrix4[i,9]=4.5*k4*L_hor**2
```

```
            Matrix4[i,10]=3*L_hor*k4
            Matrix4[i,11]=k4
            Matrix4[i,12]=EI2
        if i == 14: #eq15
            Matrix4[i,12]=-EI2*x_pos
            Matrix4[i,13]=-EI2
            Matrix4[i,16]=EI2*x_pos
            Matrix4[i,17]=EI2
        if i == 15: #eq16
            Matrix4[i,12]=1.0/6*x_pos**3
            Matrix4[i,13]=0.5*x_pos**2
            Matrix4[i,14]=x_pos
            Matrix4[i,15]=1
            Matrix4[i,16]=-1.0/6*x_pos**3
            Matrix4[i,17]=-0.5*x_pos**2
            Matrix4[i,18]=-x_pos
            Matrix4[i,19]=-1
        if i == 16: #eq17
            Matrix4[i,12]=-0.5*x_pos**2
            Matrix4[i,13]=-x_pos
            Matrix4[i,14]=-1
            Matrix4[i,16]=0.5*x_pos**2
            Matrix4[i,17]=x_pos
            Matrix4[i,18]=1
        if i == 17: #eq18
            Matrix4[i,12]=1.0/6*x_pos**3
            Matrix4[i,13]=0.5*x_pos**2
            Matrix4[i,14]=x_pos
            Matrix4[i,15]=1
        if i == 18: #eq19
            Matrix4[i,16]=-EI2/k5+32/3.0*L_hor**3
            Matrix4[i,17]=8*L_hor**2
            Matrix4[i,18]=4*L_hor
            Matrix4[i,19]=1
        if i == 19: #eq20
            Matrix4[i,16]=-4*EI2*L_hor
            Matrix4[i,17]=-EI2
```

9. Create lists for plots

```
F_list    = [0]
F_x_list  = [0]
time_list = [0]

x_V_list   = [x_V]
z_V_list   = [0]
v_x_V_list = [v_x_V]
v_z_V_list = [v_z_V]

x_C_list   = [x_C]
z_C_list   = [z_C]
v_x_C_list = [v_x_C]
v_z_C_list = [v_z_C]

x_A_list   = [x_A]
z_A_list   = [z_A]
v_x_A_list = [v_x_A]
v_z_A_list = [v_z_A]

theta_list     =[theta]
theta_dot_list =[theta_dot]

E_kin_str_list      = [0]
E_pot_str_list      = [0]
E_kin_x_ship_list   = [0.5*m_x*v_x_C**2]
E_kin_z_ship_list   = [0.5*m_z*v_z_C**2]
E_kin_rot_ship_list = [0]
E_pot_soil_list     = [0]
E_friction_list     = [0]
E_drag_rot_list     = [0]
E_drag_trans_list   = [0]
E_tot_list          = [0.5*m_z*v_z_C**2]     #E_KIN_X BUITEN BESCHOUWING

x_RA_list = [x_RA]
z_RA_list = [z_RA]
x_RV_list = [x_RV]
```

```
z_RV_list = [z_RV]

a_list  =[a]
b_list  =[b]
c_list  =[c]
d_list  =[d]

M_water_rotation_list = [0]
F_water_damping_z_list = [0]
```

10. Mechanical relations

```
def w1(C1,C2,C3,C4,x):
    w1 = 1/6*C1*x**3 + 1/2*C2*x**2 + C3*x + C4
    return w1
def phi1(C1,C2,C3,x):
    phi1 = -(1/2*C1*x**2 + C2*x + C3)
    return phi1
def M1(C1,C2,x):
    M1 = -(EI2*(C1*x+C2))
    return M1
def V1(C1):
    V1=-(EI2*C1)
    return V1

def w2(C5,C6,C7,C8,x):
    w2 = 1/6*C5*x**3 + 1/2*C6*x**2 + C7*x + C8
    return w2
def phi2(C5,C6,C7,x):
    phi2 = -(1/2*C5*x**2 + C6*x + C7)
    return phi2
def M2(C5,C6,x):
    M2 = -(EI2*(C5*x+C6))
    return M2
def V2(C5):
    V2=-(EI2*C5)
    return V2

def w3(C9,C10,C11,C12,x):
    w3 = 1/6*C9*x**3 + 1/2*C10*x**2 + C11*x + C12
    return w3
def phi3(C9,C10,C11,x):
    phi3 = -(1/2*C9*x**2 + C10*x + C11)
    return phi3
def M3(C9,C10,x):
    M3 = -(EI2*(C9*x+C10))
    return M3
def V3(C9):
    V3=-(EI2*C9)
    return V3

def w4(C13,C14,C15,C16,x):
    w4 = 1/6*C13*x**3 + 1/2*C14*x**2 + C15*x + C16
    return w4
def phi4(C13,C14,C15,x):
    phi4 = -(1/2*C13*x**2 + C14*x + C15)
    return phi4
def M4(C13,C14,x):
    M4 = -(EI2*(C13*x+C14))
    return M4
def V4(C13):
    V4=-(EI2*C13)
    return V4

def w5(C17,C18,C19,C20,x):
    w5 = 1/6*C17*x**3 + 1/2*C18*x**2 + C19*x + C20
    return w5
def phi5(C17,C18,C19,x):
    phi5 = -(1/2*C17*x**2 + C18*x + C19)
    return phi5
def M5(C17,C18,x):
    M5 = -(EI2*(C17*x+C18))
    return M5
def V5(C17):
    V5=-(EI2*C17)
    return V5
```

11.	Big loop

```
Collision = 0
start = timer()

E_paal1 = 0
E_paal2 = 0
E_paal3 = 0
E_paal4 = 0
E_paal5 = 0

for i in range(900):

#-------------------SOLVE SYSTEM-------------------
    x_oud = x_V                            #NODIG VOOR FRICTION ENERGY
    theta_oud = theta
    z_oud = z_C
    F_oud = F_x                            #NODIG VOOR FRICTION ENERGY
    F   = Symbol('F')

    if time==0:
        for i in range (20):
            globals()[str('C{}'.format(i+1))] = 0
            globals()[str('F_spring{}'.format(i+1))] = F_0
    w1_oud = w1(C1,C2,C3,C4,0)
    F1_oud = F_spring1
    w2_oud = w2(C5,C6,C7,C8,L_hor)
    F2_oud = F_spring2
    w3_oud = w3(C9,C10,C11,C12,2*L_hor)
    F3_oud = F_spring3
    w4_oud = w4(C13,C14,C15,C16,3*L_hor)
    F4_oud = F_spring4
    w5_oud = w5(C17,C18,C19,C20,4*L_hor)
    F5_oud = F_spring5

#------------------INITIATION---------------------
    if time == 0:
        delta_t = time_step_0
        z_C = z_C + v_z_C * delta_t
        z_V = z_C + b
    else:
        delta_t = time_step
    if time>40:
        delta_t =del_t_after

    for i in range (5):
        globals()[str('k{}'.format(i+1))] = globals()[str('k_tot{}'.format(i+1))]

#-----------------SOLVE SYSTEM------------------------

    if z_V >= 0:
        x_pos = x_V
        PresDef = z_V
    if z_A > 0:
        x_pos = x_A
        PresDef = z_A
        Collision = Collision + 1
        if Collision == 1:
            theta_dot_0 = theta_dot
            v_z_C_0 = v_z_C
            v_x_C_0 = v_x_C
            S = (a*theta_dot_0-v_z_C_0)/(1/m+1/M-a*(mu*b-a)/Ig)
            theta_dot = S*(mu*b-a)/Ig+theta_dot_0

            v_z_A = -S/M
            v_z_C = S/m + v_z_C_0

            v_x_C = -mu*S/m + v_x_C_0
            v_x_A = v_x_C + c*theta_dot
            print('theta at collision', theta)
            print('theta_dot at collision', theta_dot)
            print("velocity COG z", v_z_C)


    if x_pos >= 0 and x_pos < L_hor:
        Matrix1 = np.zeros([20, 20])
        make_Matrix_seg1(k1,k2,k3,k4,k5,EI2,L_hor,x_pos)
```

```
        rhs1 = np.array([0, 0, 0, 0, 0, PresDef, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
        solution = np.linalg.solve(Matrix1,rhs1)
    if x_pos >= L_hor and x_pos < 2*L_hor:
        Matrix2 = np.zeros([20, 20])
        make_Matrix_seg2(k1,k2,k3,k4,k5,EI2,L_hor,x_pos)
        rhs2 = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, PresDef, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
        solution = np.linalg.solve(Matrix2,rhs2)
    if x_pos >= 2*L_hor and x_pos < 3*L_hor:
        Matrix3 = np.zeros([20, 20])
        make_Matrix_seg3(k1,k2,k3,k4,k5,EI2,L_hor,x_pos)
        rhs3 = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, PresDef, 0, 0, 0, 0, 0, 0])
        solution = np.linalg.solve(Matrix3,rhs3)
    if x_pos >= 3*L_hor and x_pos <= 4*L_hor:
        Matrix4 = np.zeros([20, 20])
        make_Matrix_seg4(k1,k2,k3,k4,k5,EI2,L_hor,x_pos)
        rhs4 = np.array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, PresDef, 0, 0])
        solution = np.linalg.solve(Matrix4,rhs4)

#-----------------ASSIGN SOLUTION----------------------
    for i in range (len(solution)):
        globals()[str('C{}'.format(i+1))] = solution[i]


    if z_V < 0 and z_A <0:
        for i in range (len(solution)):
            globals()[str('C{}'.format(i+1))] = 0

#----------------DETERMINE FORCES IN SPRINGS FOR LENGTHS AND STIFFNESSES--------------------
---
    if x_pos >= 0 and x_pos < L_hor:
        F_spring1 = abs(V1(C1))                      #postive spring force means compression
of the spring
        F_spring2 = abs(V3(C9)-V2(C5))
        F_spring3 = abs(V4(C13)-V3(C9))
        F_spring4 = abs(V5(C17)-V4(C13))
        F_spring5 = abs(-V5(C17))
    if x_pos >= L_hor and x_pos < 2*L_hor:
        F_spring1 = abs(V1(C1))                      #postive spring force means compression
of the spring
        F_spring2 = abs(V2(C5)-V1(C1))
        F_spring3 = abs(V4(C13)-V3(C9))
        F_spring4 = abs(V5(C17)-V4(C13))
        F_spring5 = abs(-V5(C17))
    if x_pos >= 2*L_hor and x_pos < 3*L_hor:
        F_spring1 = abs(V1(C1))                      #postive spring force means compression
of the spring
        F_spring2 = abs(V2(C5)-V1(C1))
        F_spring3 = abs(V3(C9)-V2(C5))
        F_spring4 = abs(V5(C17)-V4(C13))
        F_spring5 = abs(-V5(C17))
    if x_pos >= 3*L_hor and x_pos < 4*L_hor:
        F_spring1 = abs(V1(C1))                      #postive spring force means compression
of the spring
        F_spring2 = abs(V2(C5)-V1(C1))
        F_spring3 = abs(V3(C9)-V2(C5))
        F_spring4 = abs(V4(C13)-V3(C9))
        F_spring5 = abs(-V5(C17))

    if F_spring1 <= 0:
        F_spring1 = F_0
    if F_spring2 <= 0:
        F_spring2 = F_0
    if F_spring3 <= 0:
        F_spring3 = F_0
    if F_spring4 <= 0:
        F_spring4 = F_0
    if F_spring5 <= 0:
        F_spring5 = F_0

    F_spring1_list.append(F_spring1)
    F_spring2_list.append(F_spring2)
    F_spring3_list.append(F_spring3)
    F_spring4_list.append(F_spring4)
    F_spring5_list.append(F_spring5)

#------------------UPDATE LENGTHS AND STIFFNESSES-----------------
    for i in range(5):
```

```
        globals()[str('L_ver{}'.format(i+1))]             =             L_ver_0             +
depth(globals()[str('F_spring{}'.format(i+1))])
        globals()[str('k_pile{}'.format(i+1))]                                              =
3*EI1/(globals()[str('L_ver{}'.format(i+1))])**3
        globals()[str('k_tot{}'.format(i+1))] = \
                        (1/(globals()[str('k_pile{}'.format(i+1))]) + \

(globals()[str('L_ver{}'.format(i+1))])**2/(k_rot(globals()[str('F_spring{}'.format(i+1))]))
+\
                        1/(k_trans(globals()[str('F_spring{}'.format(i+1))])))**(-1)
    L_ver1_list.append(L_ver1)
    L_ver2_list.append(L_ver2)
    L_ver3_list.append(L_ver3)
    L_ver4_list.append(L_ver4)
    L_ver5_list.append(L_ver5)

#------------------UPDATE POSITION + VELOCITIES----------------------
    if x_pos >= 0 and x_pos < L_hor:
        F = V1(C1)-V2(C5)
    if x_pos >= L_hor and x_pos < 2*L_hor:
        F = V2(C5)-V3(C9)
    if x_pos >= 2*L_hor and x_pos < 3*L_hor:
        F = V3(C9)-V4(C13)
    if x_pos >= 3*L_hor and x_pos <= 4*L_hor:
        F = V4(C13)-V5(C17)                    #FORCE BY STRUCTURE WORKING ON THE SHIP

#------------------RESISTANCE WATER-----------------------

    v_mean      = theta_dot*0.5*L_ship*0.5
    v_max       = theta_dot*0.5*L_ship
    A_side_ship = draught*L_ship
    F_drag_max  = -0.5*rho_water*C_w*v_max*abs(v_max)*draught
    M_water_rotation = 1.0/3*L_ship/2*F_drag_max*3.0/8*L_ship*2
    F_water_damping_z = -0.5*C_w*rho_water*A_side_ship*v_z_C*abs(v_z_C)

    if F <0:
        F=0
        for i in range (len(solution)):
            globals()[str('C{}'.format(i+1))] = 0

    C1_list.append(C1)
    C2_list.append(C2)
    C3_list.append(C3)
    C4_list.append(C4)
    C5_list.append(C5)
    C6_list.append(C6)
    C7_list.append(C7)
    C8_list.append(C8)
    C9_list.append(C9)
    C10_list.append(C10)
    C11_list.append(C11)
    C12_list.append(C12)
    C13_list.append(C13)
    C14_list.append(C14)
    C15_list.append(C15)
    C16_list.append(C16)
    C17_list.append(C17)
    C18_list.append(C18)
    C19_list.append(C19)
    C20_list.append(C20)

    if z_V > 0:
        M_excentric = -F*a+mu*F*b
    if F == 0:
        M_excentric = 0
    if z_A > 0:
        M_excentric = F*d+mu*F*c

    ang_acc   = (M_excentric+M_water_rotation)/Ig
    theta     = theta + theta_dot*delta_t + 0.5*ang_acc*delta_t**2
    theta_dot = theta_dot + ang_acc*delta_t

    a = 0.5*(L_ship*math.cos(theta) - B_ship*math.sin(theta))
    if theta < 0:
        0.5*(L_ship*math.cos(abs(theta)) + B_ship*math.sin(abs(theta)))

    b = 0.5*(L_ship*math.sin(theta) + B_ship*math.cos(theta))
```

```
    if theta < 0:
        b = 0.5*(B_ship*math.cos(abs(theta))-L_ship*math.sin(abs(theta)))
        if z_V < z_C:
            b = -b
    if z_A < z_C:
        c = 0.5*(L_ship*math.sin(abs(theta)) - B_ship*math.cos(abs(theta)))
    if z_A > z_C:
        c = 0.5*(-L_ship*math.sin(abs(theta)) + B_ship*math.cos(abs(theta)))
    if theta < 0:
        c = 0.5*(L_ship*math.sin(abs(theta)) + B_ship*math.cos(abs(theta)))

    d = 0.5*(L_ship*math.cos(abs(theta)) + B_ship*math.sin(abs(theta)))
    if theta < 0:
        d = 0.5*(L_ship*math.cos(abs(theta)) - B_ship*math.sin(abs(theta)))

    m_x = m0 + (C_m-1)*math.cos(abs(theta))*m0
    m_z = m0 + (C_m-1)*math.sin(abs(theta))*m0


    a_z   = (-F+F_water_damping_z)/m_z
    z_C   = z_C + v_z_C*delta_t + 0.5*a_z*delta_t**2
    v_z_C = v_z_C + a_z*delta_t

    a_x   = F*mu/m_x
    x_C   = x_C + v_x_C*delta_t + 0.5*a_x*delta_t**2
    v_x_C = v_x_C + a_x*delta_t

    F_x = F * mu

    x_V = x_C - a
    if z_V > z_C:
        z_V = z_C + b
    if z_V < z_C:
        z_V = z_C - b
    v_z_V = v_z_C + a*theta_dot
    v_z_A = v_z_C - d*theta_dot

    x_A = x_C + d
    if z_A > z_C:
        z_A = z_C + c
    if z_A < z_C:
        z_A = z_C - c

    x_RA = x_A - B_ship*math.sin(theta) #FOR PLOTTING THE SHIP
    z_RA = z_A - B_ship*math.cos(theta) #FOR PLOTTING THE SHIP
    x_RV = x_V - B_ship*math.sin(theta) #FOR PLOTTING THE SHIP
    z_RV = z_V - B_ship*math.cos(theta) #FOR PLOTTING THE SHIP

    M  = (0.25*math.pi*(D1**2-(D1-2*t1)**2)*L_ver_0*N_piles + \
      0.25*math.pi*(D2**2-(D2-2*t2)**2)*L_hor*N_beams)*rho_steel

#------------------------------------ENERGY-----------------------------
    x   = Symbol('x')
    def w11(x):
        return w1(C1,C2,C3,C4,x)
    def w22(x):
        return w2(C5,C6,C7,C8,x)
    def w33(x):
        return w3(C9,C10,C11,C12,x)
    def w44(x):
        return w4(C13,C14,C15,C16,x)
    def w55(x):
        return w5(C17,C18,C19,C20,x)

    def M11(x):
        return ((M1(C1,C2,x))**2)/EI2*0.5
    def M22(x):
        return ((M2(C5,C6,x))**2)/EI2*0.5
    def M33(x):
        return ((M3(C9,C10,x))**2)/EI2*0.5
    def M44(x):
        return ((M4(C13,C14,x))**2)/EI2*0.5
    def M55(x):
        return ((M5(C17,C18,x))**2)/EI2*0.5

    if x_pos >= 0 and x_pos < L_hor:
        E_pot_M1, err = quad(M11,0,x_pos)
```

```
        E_pot_M2, err = quad(M22,x_pos,L_hor)
        E_pot_M3, err = quad(M33,L_hor,2*L_hor)
        E_pot_M4, err = quad(M44,2*L_hor,3*L_hor)
        E_pot_M5, err = quad(M55,3*L_hor,4*L_hor)
        w1_int,err = quad(w11,0,x_pos)    #QUAD TAKES THE INTEGRAL OF A FUNCTION
        w2_int,err = quad(w22,x_pos,L_hor)
        w3_int,err = quad(w33,L_hor,2*L_hor)
        w4_int,err = quad(w44,2*L_hor,3*L_hor)
        w5_int,err = quad(w55,3*L_hor,4*L_hor)
    if x_pos >= L_hor and x_pos < 2*L_hor:
        E_pot_M1, err = quad(M11,0,L_hor)
        E_pot_M2, err = quad(M22,L_hor,x_pos)
        E_pot_M3, err = quad(M33,x_pos,2*L_hor)
        E_pot_M4, err = quad(M44,2*L_hor,3*L_hor)
        E_pot_M5, err = quad(M55,3*L_hor,4*L_hor)
        w1_int,err = quad(w11,0,L_hor)    #QUAD TAKES THE INTEGRAL OF A FUNCTION
        w2_int,err = quad(w22,L_hor,x_pos)
        w3_int,err = quad(w33,x_pos,2*L_hor)
        w4_int,err = quad(w44,2*L_hor,3*L_hor)
        w5_int,err = quad(w55,3*L_hor,4*L_hor)
    if x_pos >= 2*L_hor and x_pos < 3*L_hor:
        E_pot_M1, err = quad(M11,0,L_hor)
        E_pot_M2, err = quad(M22,L_hor,2*L_hor)
        E_pot_M3, err = quad(M33,2*L_hor,x_pos)
        E_pot_M4, err = quad(M44,x_pos,3*L_hor)
        E_pot_M5, err = quad(M55,3*L_hor,4*L_hor)
        w1_int,err = quad(w11,0,L_hor)    #QUAD TAKES THE INTEGRAL OF A FUNCTION
        w2_int,err = quad(w22,L_hor,2*L_hor)
        w3_int,err = quad(w33,2*L_hor,x_pos)
        w4_int,err = quad(w44,x_pos,3*L_hor)
        w5_int,err = quad(w55,3*L_hor,4*L_hor)
    if x_pos >= 3*L_hor and x_pos < 4*L_hor:
        E_pot_M1, err = quad(M11,0,L_hor)
        E_pot_M2, err = quad(M22,L_hor,2*L_hor)
        E_pot_M3, err = quad(M33,2*L_hor,3*L_hor)
        E_pot_M4, err = quad(M44,3*L_hor, x_pos)
        E_pot_M5, err = quad(M55,x_pos,4*L_hor)
        w1_int,err = quad(w11,0,L_hor)    #QUAD TAKES THE INTEGRAL OF A FUNCTION
        w2_int,err = quad(w22,L_hor,2*L_hor)
        w3_int,err = quad(w33,2*L_hor,3*L_hor)
        w4_int,err = quad(w44,3*L_hor,x_pos)
        w5_int,err = quad(w55,x_pos,4*L_hor)

    A = w1_int + w2_int + w3_int + w4_int + w5_int
    B = PresDef*4*L_hor
    eta = A/B

    if time<20:
        E_kin_str = 0.5*eta*M*v_z_V**2
    if time>20:
        E_kin_str = 0.5*eta*M*v_z_A**2

    for i in range(5):
        globals()[str('E_pot_ver{}'.format(i+1))]                                =
((globals()[str('F_spring{}'.format(i+1))])**2 * \

(globals()[str('L_ver{}'.format(i+1))])**3)/(6*EI2)
    E_pot_pile1 = F_spring1**2*L_ver1**3 / (6*EI2)
    E_pot_pile2 = F_spring2**2*L_ver2**3 / (6*EI2)
    E_pot_pile3 = F_spring3**2*L_ver3**3 / (6*EI2)
    E_pot_pile4 = F_spring4**2*L_ver4**3 / (6*EI2)
    E_pot_pile5 = F_spring5**2*L_ver5**3 / (6*EI2)
    E_pot_pile1_list.append(E_pot_pile1)
    E_pot_pile2_list.append(E_pot_pile2)
    E_pot_pile3_list.append(E_pot_pile3)
    E_pot_pile4_list.append(E_pot_pile4)
    E_pot_pile5_list.append(E_pot_pile5)
    E_pot_piles_tot_list.append(E_pot_pile1+E_pot_pile2+E_pot_pile3+E_pot_pile4+E_pot_pile5)

    E_pot_M1_list.append(E_pot_M1)
    E_pot_M2_list.append(E_pot_M2)
    E_pot_M3_list.append(E_pot_M3)
    E_pot_M4_list.append(E_pot_M4)
    E_pot_M5_list.append(E_pot_M5)

    E_pot_str = (E_pot_M1 + E_pot_M2 + E_pot_M3 + E_pot_M4 + E_pot_M5 + \
                E_pot_ver1 + E_pot_ver2 + E_pot_ver3 + E_pot_ver4 + E_pot_ver5)
```

```
    E_pot_str2 = (E_pot_M1 + E_pot_M2 + E_pot_M3 + E_pot_M4 + E_pot_M5)
    E_pot_str2_list.append(E_pot_str2)

    E_grond_paal1 = 0.5*k_tot1*(w1(C1,C2,C3,C4,0))**2
    E_grond_paal2 = 0.5*k_tot2*(w2(C5,C6,C7,C8,L_hor))**2
    E_grond_paal3 = 0.5*k_tot3*(w3(C9,C10,C11,C12,2*L_hor))**2
    E_grond_paal4 = 0.5*k_tot4*(w4(C13,C14,C15,C16,3*L_hor))**2
    E_grond_paal5 = 0.5*k_tot5*(w5(C17,C18,C19,C20,4*L_hor))**2

    E_grond_paal_tot = E_grond_paal1
+E_grond_paal2+E_grond_paal3+E_grond_paal4+E_grond_paal5
    E_grond_paal_tot_list.append(E_grond_paal_tot)

    E_paal1 = E_paal1 + (F_spring1+F1_oud)*0.5*(w1(C1,C2,C3,C4,0)-w1_oud)
    E_paal2 = E_paal2 + (F_spring2+F2_oud)*0.5*(w2(C5,C6,C7,C8,L_hor)-w2_oud)
    E_paal3 = E_paal3 + (F_spring3+F3_oud)*0.5*(w3(C9,C10,C11,C12,2*L_hor)-w3_oud)
    E_paal4 = E_paal4 + (F_spring4+F4_oud)*0.5*(w4(C13,C14,C15,C16,3*L_hor)-w4_oud)
    E_paal5 = E_paal5 + (F_spring5+F5_oud)*0.5*(w5(C17,C18,C19,C20,4*L_hor)-w5_oud)
    E_paal_totaal = E_paal1+E_paal2+E_paal3+E_paal4+E_paal5

    if F == 0:
        E_pot_str =0
        E_kin_str= 0
        E_paal_totaal = 0

    E_paal_totaal_list.append(E_paal_totaal)

    E_kin_x_ship = 0.5 * m_x * v_x_C**2
    E_kin_z_ship = 0.5 * m_z * v_z_C**2
    E_kin_rot_ship = 0.5 * Ig * theta_dot**2

    E_soil_rot1 = 0.5*F_spring1**2*L_ver1**2/k_rot(F_spring1);
E_soil_rot1_list.append(E_soil_rot1)
    E_soil_rot2 = 0.5*F_spring2**2*L_ver2**2/k_rot(F_spring2);
E_soil_rot2_list.append(E_soil_rot2)
    E_soil_rot3 = 0.5*F_spring3**2*L_ver3**2/k_rot(F_spring3);
E_soil_rot3_list.append(E_soil_rot3)
    E_soil_rot4 = 0.5*F_spring4**2*L_ver4**2/k_rot(F_spring4);
E_soil_rot4_list.append(E_soil_rot4)
    E_soil_rot5 = 0.5*F_spring5**2*L_ver5**2/k_rot(F_spring5);
E_soil_rot5_list.append(E_soil_rot5)

    E_soil_trans1 = 0.5*F_spring1**2/k_trans(F_spring1);
E_soil_trans1_list.append(E_soil_trans1)
    E_soil_trans2 = 0.5*F_spring2**2/k_trans(F_spring2);
E_soil_trans2_list.append(E_soil_trans2)
    E_soil_trans3 = 0.5*F_spring3**2/k_trans(F_spring3);
E_soil_trans3_list.append(E_soil_trans3)
    E_soil_trans4 = 0.5*F_spring4**2/k_trans(F_spring4);
E_soil_trans4_list.append(E_soil_trans4)
    E_soil_trans5 = 0.5*F_spring5**2/k_trans(F_spring5);
E_soil_trans5_list.append(E_soil_trans5)

    E_pot_soil_pile1 = E_soil_rot1 + E_soil_trans1;
E_pot_soil_pile1_list.append(E_pot_soil_pile1)
    E_pot_soil_pile2 = E_soil_rot2 + E_soil_trans2;
E_pot_soil_pile2_list.append(E_pot_soil_pile2)
    E_pot_soil_pile3 = E_soil_rot3 + E_soil_trans3;
E_pot_soil_pile3_list.append(E_pot_soil_pile3)
    E_pot_soil_pile4 = E_soil_rot4 + E_soil_trans4;
E_pot_soil_pile4_list.append(E_pot_soil_pile4)
    E_pot_soil_pile5 = E_soil_rot5 + E_soil_trans5;
E_pot_soil_pile5_list.append(E_pot_soil_pile5)

    E_pot_soil = (E_pot_soil_pile1 + E_pot_soil_pile2 + E_pot_soil_pile3 + E_pot_soil_pile4
+ E_pot_soil_pile5)
    if F == 0:
        E_pot_soil =0

    E_friction = (E_friction + 0.5*(F_oud+F_x)*(abs(x_oud-x_V)))
    E_drag_rot = (E_drag_rot + abs((theta-theta_oud)*(M_water_rotation)))
    E_drag_trans = (E_drag_trans + abs((z_C-z_oud)*(F_water_damping_z)))

    #E_tot = E_kin_str + E_pot_str + E_kin_x_ship + E_kin_z_ship + E_kin_rot_ship + E_pot_soil
+E_friction #MET E_kin_x
    #E_tot = E_kin_str + E_pot_str + E_kin_z_ship + E_kin_rot_ship + E_pot_soil +E_friction
#ZONDER E_kin_x
```

```
    #E_tot = E_kin_str + E_pot_str + E_kin_z_ship + E_kin_rot_ship + E_pot_soil+E_drag_rot
+E_drag_trans
    E_tot = E_kin_str + E_pot_str2 + E_kin_z_ship + E_kin_rot_ship + E_paal_totaal+E_drag_rot
+E_drag_trans
    time = time+delta_t
    #-------------------APPENDING TO LISTS-------------------
    F_list.append(F)
    F_x_list.append(F_x)
    time_list.append(time)

    theta_list.append(theta)
    theta_dot_list.append(theta_dot)

    x_V_list.append(x_V)
    z_V_list.append(z_V)
    v_x_V_list.append(v_x_V)
    v_z_V_list.append(v_z_V)

    x_C_list.append(x_C)
    z_C_list.append(z_C)
    v_x_C_list.append(v_x_C)
    v_z_C_list.append(v_z_C)

    z_A_list.append(z_A)
    x_A_list.append(x_A)
    v_x_A_list.append(v_x_A)
    v_z_A_list.append(v_z_A)

    E_kin_str_list.append(E_kin_str)
    E_pot_str_list.append(E_pot_str)
    E_kin_x_ship_list.append(E_kin_x_ship)
    E_kin_z_ship_list.append(E_kin_z_ship)
    E_kin_rot_ship_list.append(E_kin_rot_ship)
    E_pot_soil_list.append(E_pot_soil)
    E_friction_list.append(E_friction)
    E_drag_rot_list.append(E_drag_rot)
    E_drag_trans_list.append(E_drag_trans)
    E_tot_list.append(E_tot)

    a_list.append(a)
    b_list.append(b)
    c_list.append(c)
    d_list.append(d)
    x_RA_list.append(x_RA)
    z_RA_list.append(z_RA)
    x_RV_list.append(x_RV)
    z_RV_list.append(z_RV)

    x_pos_list.append(x_pos_list)

    M_water_rotation_list.append(M_water_rotation)
    F_water_damping_z_list.append(F_water_damping_z)


end = timer()
print(end - start)
```

12.	Units conversion

```
for i in range(len(F_list)):
    F_list[i]=F_list[i]/1000
    F_x_list[i]=F_x_list[i]/1000

for i in range(len(theta_list)):
    theta_list[i]=theta_list[i]/math.pi*180
```

13.	Plot ship movement

```
%matplotlib inline
#matplotlib notebook
def datapunt(t):
    i=int(t)
    x1 = [x_V_list[i],x_A_list[i]]
    z1 = [z_V_list[i],z_A_list[i]]

    x2 = [x_RV_list[i],x_RA_list[i]]
```

```
    z2 = [z_RV_list[i],z_RA_list[i]]

    x3 = [x_V_list[i],x_RV_list[i]]
    z3 = [z_V_list[i],z_RV_list[i]]

    x4 = [x_A_list[i],x_RA_list[i]]
    z4 = [z_A_list[i],z_RA_list[i]]

    tijdstip = time_list[i]
    return x1, z1, x2, z2, x3, z3, x4, z4, tijdstip

def plot_line(t):
    x1 = datapunt(t)[0]
    z1 = datapunt(t)[1]

    clr='black'
    if z1[0] > 0:
        clr='red'
    else:
        clr = 'black'
    if z1[1] > 0:
        clr='green'

    plt.plot(x1,z1,'-',color=clr)

    x2 = datapunt(t)[2]
    z2 = datapunt(t)[3]
    plt.plot(x2,z2,'-',color=clr)

    x3 = datapunt(t)[4]
    z3 = datapunt(t)[5]
    plt.plot(x3,z3,'-',color=clr)

    x4 = datapunt(t)[6]
    z4 = datapunt(t)[7]
    plt.plot(x4,z4,'-',color=clr)

    plt.xlim(-100,200)
    plt.ylim(0.03,-150)
    plt.title(datapunt(t)[8])#"time =")
    plt.plot([0,4*L_hor],[0,0],color='red')
    plt.plot(x_C_list,z_C_list)
interact(plot_line,
        t    =    widgets.FloatSlider(min=0,    max=(len(time_list)-1),    step=1,    value=1,
description='time (steps)'))
```

Appendix G: Virtual Basic (VBA) script

```
Private Sub CommandButton1_Click()
'--------------------------------------------DIMENSIONS------------------------------------------
Dim mystring As String
Dim myline As String
Dim i As Integer
Dim p As Integer
Dim fs, f
Dim copyResults As Boolean

'--------------------------------------------LISTS-----------------------------------------------
D_list = Array("D_CS1", "D_CS2", "D_CS3", "D_CS4", "D_CS5", "D_CS6", "D_CS7", "D_CS8", _
              "D_CS9", "D_CS10", "D_CS11", "D_CS12", "D_CS13", "D_CS14", "D_CS15", "D_CS16", _
              "D_CS17", "D_CS18", "D_CS19", "D_CS20", "D_CS21", "D_CS22", "D_CS23", "D_CS24", _
              "D_CS25", "D_CS26", "D_CS27", "D_CS28", "D_CS29", "D_CS30", "D_CS31", "D_CS32")
'-----------------------------------------------------------------------------------------------
t_list = Array("t_CS1", "t_CS2", "t_CS3", "t_CS4", "t_CS5", "t_CS6", "t_CS7", "t_CS8", _
              "t_CS9", "t_CS10", "t_CS11", "t_CS12", "t_CS13", "t_CS14", "t_CS15", "t_CS16", _
              "t_CS17", "t_CS18", "t_CS19", "t_CS20", "t_CS21", "t_CS22", "t_CS23", "t_CS24", _
              "t_CS25", "t_CS26", "t_CS27", "t_CS28", "t_CS29", "t_CS30", "t_CS31", "t_CS32")
'-----------------------------------------------------------------------------------------------
x_coordlist = Array("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10", "x11", "x12", _
                   "x13", "x14", "x15", "x16", "x17", "x18", "x19", "x20", "x21", "x22", "x23", _
                   "x24", "x25", "x26", "x27", "x28", "x29", "x30", "x31", "x32", "x33", "x34", _
                   "x35", "x36", "x37", "x38", "x39", "x40", "x41", "x42", "x43", "x44", "x45", _
                   "x46", "x47", "x48", "x49", "x50", "x51", "x52", "x53", "x54", "x55", "x56", _
                   "x57", "x58", "x59", "x60", "x61", "x62", "x63", "x64", "x65", "x66", "x67", _
                   "x68", "x69", "x70", "x71", "x72", "x73", "x74", "x75", "x76", "x77")
'-----------------------------------------------------------------------------------------------
z_coordlist = Array("z1", "z2", "z3", "z4", "z5", "z6", "z7", "z8", "z9", "z10", "z11", "z12", _
                   "z13", "z14", "z15", "z16", "z17", "z18", "z19", "z20", "z21", "z22", "z23", _
                   "z24", "z25", "z26", "z27", "z28", "z29", "z30", "z31", "z32", "z33", "z34", _
                   "z35", "z36", "z37", "z38", "z39", "z40", "z41", "z42", "z43", "z44", "z45", _
                   "z46", "z47", "z48", "z49", "z50", "z51", "z52", "z53", "z54", "z55", "z56", _
                   "z57", "z58", "z59", "z60", "z61", "z62", "z63", "z64", "z65", "z66", "z67", _
                   "z68", "z69", "z70", "z71", "z72", "z73", "z74", "z75", "z76", "z77")
'-----------------------------------------------------------------------------------------------
CSB_list = Array("CSB1", "CSB2", "CSB3", "CSB4", "CSB5", "CSB6", "CSB7", "CSB8", "CSB9", "CSB10", "CSB11",
"CSB12", _
                 "CSB13", "CSB14", "CSB15", "CSB16", "CSB17", "CSB18", "CSB19", "CSB20", "CSB21", "CSB22",
"CSB23", _
                 "CSB24", "CSB25", "CSB26", "CSB27", "CSB28", "CSB29", "CSB30", "CSB31", "CSB32", "CSB33",
"CSB34", _
                 "CSB35", "CSB36", "CSB37", "CSB38", "CSB39", "CSB40", "CSB41", "CSB42", "CSB43", "CSB44",
"CSB45", _
                 "CSB46", "CSB47", "CSB48", "CSB49", "CSB50", "CSB51", "CSB52", "CSB53", "CSB54", "CSB55",
"CSB56", _
                 "CSB57", "CSB58", "CSB59", "CSB60", "CSB61", "CSB62", "CSB63", "CSB64", "CSB65", "CSB66",
"CSB67", _
                 "CSB68", "CSB69", "CSB70", "CSB71", "CSB72", "CSB73", "CSB74", "CSB75", "CSB76")
'-----------------------------------------------------------------------------------------------
Blad_list = Array("Blad1", "Blad2", "Blad3", "Blad4", "Blad5", "Blad6", "Blad7", "Blad8", "Blad9", "Blad10",
"Blad11", "Blad12", _
                   "Blad13",  "Blad14",  "Blad15",  "Blad16",  "Blad17",  "Blad18",  "Blad19",  "Blad20",
"Blad21", "Blad22", "Blad23", _
                   "Blad24",  "Blad25",  "Blad26",  "Blad27",  "Blad28",  "Blad29",  "Blad30",  "Blad31",
"Blad32", "Blad33", "Blad34", _
                   "Blad35",  "Blad36",  "Blad37",  "Blad38",  "Blad39",  "Blad40",  "Blad41",  "Blad42",
"Blad43", "Blad44", "Blad45", _
                   "Blad46",  "Blad47",  "Blad48",  "Blad49",  "Blad50",  "Blad51",  "Blad52",  "Blad53",
"Blad54", "Blad55", "Blad56", _
                   "Blad57",  "Blad58",  "Blad59",  "Blad60",  "Blad61",  "Blad62",  "Blad63",  "Blad64",
"Blad65", "Blad66", "Blad67", _
                   "Blad68",  "Blad69",  "Blad70",  "Blad71",  "Blad72",  "Blad73",  "Blad74",  "Blad75",
"Blad76", "Blad77")
'-----------------------------------------------------------------------------------------------
'-----------------------------------------------IMPORTING DATA FROM TABLE-------------------------
copyResults = True

L_hor = Worksheets("Table").Cells(9, 2).Value
x_V =  (Worksheets("Table").Cells(49, 2).Value - 1) * L_hor + (Worksheets("Table").Cells(59, 2).Value) *
L_hor
mu = Worksheets("Table").Cells(15, 16).Value
delta_t = Worksheets("Table").Cells(27, 16).Value

mass = Worksheets("Table").Cells(5, 16).Value
draught = Worksheets("Table").Cells(9, 16).Value
L_ship = Worksheets("Table").Cells(6, 16).Value
B_ship = Worksheets("Table").Cells(7, 16).Value
H_ship = Worksheets("Table").Cells(8, 16).Value
theta_degree = Worksheets("Table").Cells(10, 16).Value
v_x_C_0 = Worksheets("Table").Cells(11, 16).Value

C_w = Worksheets("Table").Cells(22, 16).Value
```

```
F_init = Worksheets("Table").Cells(16, 16).Value
Force1 = F_init
Force2 = F_init
Force3 = F_init
Force4 = F_init
Force5 = F_init
Force6 = F_init
Force7 = F_init

'-----------------------------------------------DERIVED PROPERTIES----------------------------------
Inertia = 1 / 12 * mass * L_ship ^ 2
Ig = 1 / 12 * mass * L_ship ^ 2 * 1.4  '1.8 FOR ADDED MASS
m = CDec(100000)



theta = theta_degree / 180 * Application.WorksheetFunction.Pi()
a = 0.5 * (L_ship * Cos(theta) - B_ship * Sin(theta))
b = 0.5 * (L_ship * Sin(theta) + B_ship * Cos(theta))

time_t = 0
x_C = x_V + a
z_C = z_V - b

v_x_V_0 = v_x_C_0
v_z_C_0 = -v_x_C_0 * Tan(theta)

m_x = m + 0.1 * Cos(theta) * m
m_z = m + 0.1 * Sin(theta) * m


S = CDec(v_z_C_0 / (1 / mass * (1 - (12 * a) / (L_ship ^ 2) * (mu * b - a)) + 1 / m))

theta_dot = S * (mu * b - a) / Inertia

v_z_V = S / m
v_z_C = v_z_V - a * theta_dot

v_x_C = mu * S / m + v_x_C_0
v_x_V = v_x_C + b * theta_dot


For p = 1 To 40

Set fs = CreateObject("Scripting.FileSystemObject")
Set f = fs.CreateTextFile("C:\Users\wvdommelen\Desktop\Guidework\newlyScripted.xml", True, True)
delta_t = 0.1

'---------------------------UPDATE POSITIONS-------------------------
If time_t = 0 Then
    delta_t = 0.1
    z_C = z_C + v_z_C * delta_t
    z_V = v_z_C * delta_t
End If

v_mean = theta_dot * 0.5 * L_ship * 0.5
v_max = theta_dot * 0.5 * L_ship
A_side_ship = draught * L_ship
F_drag_max = -0.5 * 1000 * C_w * v_max * Abs(v_max) * draught
M_water_rotation = 1# / 3 * L_ship / 2 * F_drag_max * 3# / 8 * L_ship * 2
F_water_damping_z = -0.5 * C_w * 1000 * A_side_ship * v_z_C * Abs(v_z_C)
Debug.Print "M:", M_water_rotation, "F:", F_water_damping_z

M_excentric = -F_result * a + mu * F_result * b

Debug.Print "theta voor:", theta
ang_acc = (M_excentric + M_water_rotation) / Ig
theta = theta + theta_dot * delta_t + 0.5 * ang_acc * delta_t ^ 2
theta_dot = theta_dot + ang_acc * delta_t
Debug.Print "theta na:", theta

a = 0.5 * (L_ship * Cos(theta) - B_ship * Sin(theta))
b = 0.5 * (L_ship * Sin(theta) + B_ship * Cos(theta))
c = L_ship * Sin(theta) - b


m_x = m + 0.1 * Cos(theta)
m_z = m + 0.1 * Sin(theta)

a_z = (-F_result + F_water_damping_z) / m_z
z_C = z_C + v_z_C * delta_t + 0.5 * a_z * delta_t ^ 2
v_z_C = v_z_C + a_z * delta_t

a_x = F_result * mu / m_x
x_C = x_C + v_x_C * delta_t + 0.5 * a_x * delta_t ^ 2
v_x_C = v_x_C + a_x * delta_t
```

```
x_V = x_C - a
z_V = z_C + b
v_z_V = v_z_C + a * theta_dot
v_z_A = v_z_C - a * theta_dot

'---------------------------------------------------------------------
LenXML = Worksheets("XML").Cells(1, 7).Value
For i = 1 To LenXML
    mystring = Worksheets("XML").Cells(i, 2).Value
    Worksheets("XML").Cells(i, 15) = mystring
Next i

For i = 1 To LenXML
    If Worksheets("XML").Cells(i, 1).Value <> "" Then

        For j = 1 To 32                         'Change outside diameter'
            If Worksheets("XML").Cells(i, 1).Value = D_list(j - 1) Then
                If j < 11 Then                  'First pile
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(j *
5, 6).Value / 1000) & """/>"
                End If
                If j > 10 And j < 21 Then       'Middle piles
                    k = j Mod 11 + 1
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(k *
5, 9).Value / 1000) & """/>"
                End If
                If j > 20 And j < 31 Then       'End pile
                    k = j Mod 11 + 1
                    L = k Mod 11 + 1
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(L *
5, 12).Value / 1000) & """/>"
                End If
                If j = 31 Then
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(2,
2).Value / 1000) & """/>"
                End If
            End If

            If Worksheets("XML").Cells(i, 1).Value = t_list(j - 1) Then 'Change wall thickness
                If j < 11 Then                  'First pile
                    k = j Mod 11 - 1
                    L = 6 + k * 5
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(L,
6).Value / 1000) & """/>"
                End If
                If j > 10 And j < 21 Then       'Second Pile
                    k = j Mod 11
                    L = 6 + k * 5
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(L,
9).Value / 1000) & """/>"
                End If
                If j > 20 And j < 31 Then       'End pile
                    k = -99 + 5 * j
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(k,
12).Value / 1000) & """/>"
                End If
                If j = 31 Then
                    Worksheets("XML").Cells(i, 15).Value = " <p4 v=""" & Str(Worksheets("Table").Cells(3,
2).Value / 1000) & """/>"
                End If
            End If
        Next j

        For j = 1 To 77
            If Worksheets("XML").Cells(i, 1).Value = x_coordlist(j - 1) Then    'Update x-coord
                If j > 0 And j < 8 Then         'Horizontal beam
                    k = j - 1
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * k) & """/>"
                End If
                If j > 7 And j < 18 Then        'Pile 1
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 0) & """/>"
                End If
                If j > 17 And j < 28 Then       'Pile 2
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 1) & """/>"
                End If
                If j > 27 And j < 38 Then       'Pile 3
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 2) & """/>"
                End If
                If j > 37 And j < 48 Then       'Pile 4
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 3) & """/>"
                End If
                If j > 47 And j < 58 Then       'Pile 5
```

```
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 4) & """/>"
                End If
                If j > 57 And j < 68 Then       'Pile 6
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 5) & """/>"
                End If
                If j > 67 And j < 78 Then       'Pile 7
                    Worksheets("XML").Cells(i, 15).Value = " <p1 v=""" & Str(Worksheets("Table").Cells(9,
2).Value * 6) & """/>"
                End If
            End If
        Next j

        For j = 1 To 77
            If Worksheets("XML").Cells(i, 1).Value = z_coordlist(j - 1) Then    'Update z-coord
                If j > 0 And j < 8 Then         'horizontal beam
                    k = j - 1
                    Worksheets("XML").Cells(i, 15).Value = " <p3 v=""" & Str(Worksheets("Table").Cells(10,
2).Value) & """/></obj>"
                End If
                If j > 7 And j < 18 Then        'pile 1
                    k = -33 + 5 * j
                    Worksheets("XML").Cells(i, 15).Value = " <p3 v=""" & Str(Worksheets("Table").Cells(k,
6).Value) & """/></obj>"
                End If
                If j > 17 And j < 28 Then        'pile 2
                    If Worksheets("Table").Cells(24, 16).Value > 1 Then
                        k = -83 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 9).Value) & """/></obj>"
                    End If
                    If Worksheets("Table").Cells(24, 16).Value = 1 Then
                        k = -83 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 12).Value) & """/></obj>"
                    End If
                End If
                If j > 27 And j < 38 Then        'pile 3
                    If Worksheets("Table").Cells(24, 16).Value > 2 Then
                        k = -133 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 9).Value) & """/></obj>"
                    End If
                    If Worksheets("Table").Cells(24, 16).Value = 2 Then
                        k = -133 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 12).Value) & """/></obj>"
                    End If
                End If
                If j > 37 And j < 48 Then       'pile 4
                    If Worksheets("Table").Cells(24, 16).Value > 3 Then
                        k = -183 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 9).Value) & """/></obj>"
                    End If
                    If Worksheets("Table").Cells(24, 16).Value = 3 Then
                        k = -183 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 12).Value) & """/></obj>"
                    End If
                End If
                If j > 47 And j < 58 Then        'pile 5
                    If Worksheets("Table").Cells(24, 16).Value > 4 Then
                        k = -233 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 9).Value) & """/></obj>"
                    End If
                    If Worksheets("Table").Cells(24, 16).Value = 4 Then
                        k = -233 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 12).Value) & """/></obj>"
                    End If
                End If
                If j > 57 And j < 68 Then        'pile 6
                    If Worksheets("Table").Cells(24, 16).Value > 5 Then
                        k = -283 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 9).Value) & """/></obj>"
                    End If
                    If Worksheets("Table").Cells(24, 16).Value = 5 Then
                        k = -283 + 5 * j
                        Worksheets("XML").Cells(i,      15).Value      =      "      <p3      v="""       &
Str(Worksheets("Table").Cells(k, 12).Value) & """/></obj>"
                    End If
                End If
                If j > 67 And j < 78 Then        'pile 7
                    k = -333 + 5 * j
```

```
                    Worksheets("XML").Cells(i, 15).Value = " <p3 v=""" & Str(Worksheets("Table").Cells(k,
12).Value) & """/></obj>"
                End If
            End If
        Next j

        Nsegments = Worksheets("Table").Cells(24, 16).Value
        t1 = 0.0158 * Force1 ^ 0.4115
        t2 = 0.0158 * Force2 ^ 0.4115
        t3 = 0.0158 * Force3 ^ 0.4115
        t4 = 0.0158 * Force4 ^ 0.4115
        t5 = 0.0158 * Force5 ^ 0.4115
        t6 = 0.0158 * Force6 ^ 0.4115
        t7 = 0.0158 * Force7 ^ 0.4115

        L1_1 = Worksheets("Table").Cells(72, 6).Value
        L1_2 = Worksheets("Table").Cells(73, 6).Value
        L1_3 = Worksheets("Table").Cells(74, 6).Value
        L1_4 = Worksheets("Table").Cells(75, 6).Value

        L2_1 = Worksheets("Table").Cells(72, 7).Value
        L2_2 = Worksheets("Table").Cells(73, 7).Value
        L2_3 = Worksheets("Table").Cells(74, 7).Value
        L2_4 = Worksheets("Table").Cells(75, 7).Value

        L3_1 = Worksheets("Table").Cells(72, 8).Value
        L3_2 = Worksheets("Table").Cells(73, 8).Value
        L3_3 = Worksheets("Table").Cells(74, 8).Value
        L3_4 = Worksheets("Table").Cells(75, 8).Value

        BedLevel = Worksheets("Table").Cells(5, 2).Value

        StiffnessRx1 = 50000000000# * Force1 ^ (-0.278)
        StiffnessY1 = 600000000000# * Force1 ^ (-0.668)
        StiffnessRx2 = 50000000000# * Force2 ^ (-0.278)
        StiffnessY2 = 600000000000# * Force2 ^ (-0.668)
        StiffnessRx3 = 50000000000# * Force3 ^ (-0.278)
        StiffnessY3 = 600000000000# * Force3 ^ (-0.668)
        StiffnessRx4 = 50000000000# * Force4 ^ (-0.278)
        StiffnessY4 = 600000000000# * Force4 ^ (-0.668)
        StiffnessRx5 = 50000000000# * Force5 ^ (-0.278)
        StiffnessY5 = 600000000000# * Force5 ^ (-0.668)
        StiffnessRx6 = 50000000000# * Force6 ^ (-0.278)
        StiffnessY6 = 600000000000# * Force6 ^ (-0.668)
        StiffnessRx7 = 50000000000# * Force7 ^ (-0.278)
        StiffnessY7 = 600000000000# * Force7 ^ (-0.668)

        PresDef = z_V

        For j = 1 To 76
            If Worksheets("XML").Cells(i, 1).Value = CSB_list(j - 1) Then        'j = beam number    'CS1
has ID=2, C2 has ID=3 etc....
                If j > 0 And j < 7 Then                                         'Horizontal beams
                    If j <= Nsegments Then
                        Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(32) & """/>"
                    End If
                    If j > Nsegments Then
                        Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
                    End If
                End If
                'Pile 1-----------------------------------------------------------------------------
--
                If j > 6 And j < 17 Then
                    Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 5) & """/>"
                    'PILE 1 SEGMENT 7
                    If t1 > L1_1 Then
                        Worksheets("XML").Cells(3399, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                        Worksheets("XML").Cells(3400, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                        Worksheets("XML").Cells(3401, 15).Value = Worksheets("XML").Cells(3375, 15).Value
                    End If
                    If t1 <= L1_1 Then
                        k = BedLevel - t1
                        Worksheets("XML").Cells(1041, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                        Worksheets("XML").Cells(1738, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(1761, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(1784, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(3381, 15).Value = " <p1 v=""" & Str(StiffnessRx1) & """/>"
                        Worksheets("XML").Cells(3382, 15).Value = " <p2 v=""" & Str(StiffnessY1) & """/>"
                    End If
                    'PILE 1 SEGMENT 8
                    If t1 > (L1_1 + L1_2) Then
                        Worksheets("XML").Cells(3425, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                        Worksheets("XML").Cells(3426, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                        Worksheets("XML").Cells(3427, 15).Value = Worksheets("XML").Cells(3375, 15).Value
                    End If
                    If t1 <= (L1_1 + L1_2) And t1 > L1_1 Then
                        k = BedLevel - t1
                        Worksheets("XML").Cells(1047, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
```

```
                Worksheets("XML").Cells(1761, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(1784, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(3407, 15).Value = " <p1 v=""" & Str(StiffnessRx1) & """/>"
                Worksheets("XML").Cells(3408, 15).Value = " <p2 v=""" & Str(StiffnessY1) & """/>"
            End If
            'PILE 1 SEGMENT 9
            If t1 > (L1_1 + L1_2 + L1_3) Then
                Worksheets("XML").Cells(3451, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(3452, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(3453, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t1 <= (L1_1 + L1_2 + L1_3) And t1 > (L1_1 + L1_2) Then
                k = BedLevel - t1
                Worksheets("XML").Cells(1053, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(1784, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(3433, 15).Value = " <p1 v=""" & Str(StiffnessRx1) & """/>"
                Worksheets("XML").Cells(3434, 15).Value = " <p2 v=""" & Str(StiffnessY1) & """/>"
            End If
            'PILE 1 SEGMENT 10
            If t1 <= (L1_1 + L1_2 + L1_3 + L1_4) And t1 > (L1_1 + L1_2 + L1_3) Then
                k = BedLevel - t1
                Worksheets("XML").Cells(1059, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            End If
            Worksheets("XML").Cells(3459, 15).Value = " <p1 v=""" & Str(StiffnessRx1) & """/>"
            Worksheets("XML").Cells(3460, 15).Value = " <p2 v=""" & Str(StiffnessY1) & """/>"
        End If

        'Pile 2-------------------------------------------------------------------------------
--
        If j > 16 And j < 27 Then                                       'Pile 2
            If Nsegments > 1 Then
                Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 5) & """/>"
            End If
            If Nsegments = 1 Then
                Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j + 5) & """/>"
                L2_1 = L3_1
                L2_2 = L3_2
                L2_3 = L3_3
                L2_4 = L3_4
            End If
            'PILE 2 SEGMENT 7
            If t2 > L2_1 Then
                Worksheets("XML").Cells(3659, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(3660, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(3661, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t2 <= L2_1 Then
                k = BedLevel - t2
                Worksheets("XML").Cells(1101, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(1968, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(1991, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2014, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(3641, 15).Value = " <p1 v=""" & Str(StiffnessRx2) & """/>"
                Worksheets("XML").Cells(3642, 15).Value = " <p2 v=""" & Str(StiffnessY2) & """/>"
            End If
            'PILE 2 SEGMENT 8
            If t2 > (L2_1 + L2_2) Then
                Worksheets("XML").Cells(3685, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(3686, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(3687, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t2 <= (L2_1 + L2_2) And t2 > L2_1 Then
                k = BedLevel - t2
                Worksheets("XML").Cells(1107, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(1991, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2014, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(3667, 15).Value = " <p1 v=""" & Str(StiffnessRx2) & """/>"
                Worksheets("XML").Cells(3668, 15).Value = " <p2 v=""" & Str(StiffnessY2) & """/>"
            End If
            'PILE 2 SEGMENT 9
            If t2 > (L2_1 + L2_2 + L2_3) Then
                Worksheets("XML").Cells(3711, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(3712, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(3713, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t2 <= (L2_1 + L2_2 + L2_3) And t2 > (L2_1 + L2_2) Then
                k = BedLevel - t2
                Worksheets("XML").Cells(1113, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2014, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(3693, 15).Value = " <p1 v=""" & Str(StiffnessRx2) & """/>"
                Worksheets("XML").Cells(3694, 15).Value = " <p2 v=""" & Str(StiffnessY2) & """/>"
            End If
            'PILE 2 SEGMENT 10
            If t2 <= (L2_1 + L2_2 + L2_3 + L2_4) And t2 > (L2_1 + L2_2 + L2_3) Then
                k = BedLevel - t2
                Worksheets("XML").Cells(1119, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            End If
            Worksheets("XML").Cells(3719, 15).Value = " <p1 v=""" & Str(StiffnessRx2) & """/>"
```

```
        Worksheets("XML").Cells(3720, 15).Value = " <p2 v=""" & Str(StiffnessY2) & """/>"
    End If

    'Pile 3-----------------------------------------------------------------------------------
--
    If j > 26 And j < 37 Then                                           'Pile 3
        If Nsegments > 2 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 15) & """/>"
        End If
        If Nsegments = 2 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 5) & """/>"
            L2_1 = L3_1
            L2_2 = L3_2
            L2_3 = L3_3
            L2_4 = L3_4
        End If
        If Nsegments < 2 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
        End If
        'PILE 3 SEGMENT 7
        If t3 > L2_1 Then
            Worksheets("XML").Cells(3919, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(3920, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(3921, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t3 <= L2_1 Then
            k = BedLevel - t3
            Worksheets("XML").Cells(1161, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2198, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2221, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2244, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(3901, 15).Value = " <p1 v=""" & Str(StiffnessRx3) & """/>"
            Worksheets("XML").Cells(3902, 15).Value = " <p2 v=""" & Str(StiffnessY3) & """/>"
        End If
        'PILE 3 SEGMENT 8
        If t3 > (L2_1 + L2_2) Then
            Worksheets("XML").Cells(3945, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(3946, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(3947, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t3 <= (L2_1 + L2_2) And t3 > L2_1 Then
            k = BedLevel - t3
            Worksheets("XML").Cells(1167, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2221, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2244, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(3927, 15).Value = " <p1 v=""" & Str(StiffnessRx3) & """/>"
            Worksheets("XML").Cells(3928, 15).Value = " <p2 v=""" & Str(StiffnessY3) & """/>"
        End If
        'PILE 3 SEGMENT 9
        If t3 > (L2_1 + L2_2 + L2_3) Then
            Worksheets("XML").Cells(3971, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(3972, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(3973, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t3 <= (L2_1 + L2_2 + L2_3) And t3 > (L2_1 + L2_2) Then
            k = BedLevel - t3
            Worksheets("XML").Cells(1173, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2244, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(3953, 15).Value = " <p1 v=""" & Str(StiffnessRx3) & """/>"
            Worksheets("XML").Cells(3954, 15).Value = " <p2 v=""" & Str(StiffnessY3) & """/>"
        End If
        'PILE 3 SEGMENT 10
        If t3 <= (L2_1 + L2_2 + L2_3 + L2_4) And t3 > (L2_1 + L2_2 + L2_3) Then
            k = BedLevel - t3
            Worksheets("XML").Cells(1179, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
        End If
        Worksheets("XML").Cells(3979, 15).Value = " <p1 v=""" & Str(StiffnessRx3) & """/>"
        Worksheets("XML").Cells(3980, 15).Value = " <p2 v=""" & Str(StiffnessY3) & """/>"
    End If

    'Pile 4-----------------------------------------------------------------------------------
--
    If j > 36 And j < 47 Then                                           'Pile 4
        If Nsegments > 3 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 25) & """/>"
        End If
        If Nsegments = 3 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 15) & """/>"
            L2_1 = L3_1
            L2_2 = L3_2
            L2_3 = L3_3
            L2_4 = L3_4
        End If
        If Nsegments < 3 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
        End If
        'PILE 4 SEGMENT 7
        If t4 > L2_1 Then
```

```
            Worksheets("XML").Cells(4179, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(4180, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(4181, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t4 <= L2_1 Then
            k = BedLevel - t4
            Worksheets("XML").Cells(1221, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2428, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2451, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2474, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(4161, 15).Value = " <p1 v=""" & Str(StiffnessRx4) & """/>"
            Worksheets("XML").Cells(4162, 15).Value = " <p2 v=""" & Str(StiffnessY4) & """/>"
        End If
        'PILE 4 SEGMENT 8
        If t4 > (L2_1 + L2_2) Then
            Worksheets("XML").Cells(4205, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(4206, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(4207, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t4 <= (L2_1 + L2_2) And t4 > L2_1 Then
            k = BedLevel - t4
            Worksheets("XML").Cells(1227, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2451, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2474, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(4187, 15).Value = " <p1 v=""" & Str(StiffnessRx4) & """/>"
            Worksheets("XML").Cells(4188, 15).Value = " <p2 v=""" & Str(StiffnessY4) & """/>"
        End If
        'PILE 4 SEGMENT 9
        If t4 > (L2_1 + L2_2 + L2_3) Then
            Worksheets("XML").Cells(4231, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(4232, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(4233, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t4 <= (L2_1 + L2_2 + L2_3) And t4 > (L2_1 + L2_2) Then
            k = BedLevel - t4
            Worksheets("XML").Cells(1233, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2474, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(4213, 15).Value = " <p1 v=""" & Str(StiffnessRx4) & """/>"
            Worksheets("XML").Cells(4214, 15).Value = " <p2 v=""" & Str(StiffnessY4) & """/>"
        End If
        'PILE 4 SEGMENT 10
        If t4 <= (L2_1 + L2_2 + L2_3 + L2_4) And t4 > (L2_1 + L2_2 + L2_3) Then
            k = BedLevel - t4
            Worksheets("XML").Cells(1239, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
        End If
        Worksheets("XML").Cells(4239, 15).Value = " <p1 v=""" & Str(StiffnessRx4) & """/>"
        Worksheets("XML").Cells(4240, 15).Value = " <p2 v=""" & Str(StiffnessY4) & """/>"
    End If

    'Pile 5---------------------------------------------------------------------------------
--
    If j > 46 And j < 57 Then                                       'Pile 5
        If Nsegments > 4 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 35) & """/>"
        End If
        If Nsegments = 4 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 25) & """/>"
            L2_1 = L3_1
            L2_2 = L3_2
            L2_3 = L3_3
            L2_4 = L3_4
        End If
        If Nsegments < 4 Then
            Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
        End If
        'PILE 5 SEGMENT 7
        If t5 > L2_1 Then
            Worksheets("XML").Cells(4439, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(4440, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(4441, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t5 <= L2_1 Then
            k = BedLevel - t5
            Worksheets("XML").Cells(1281, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            Worksheets("XML").Cells(2658, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2681, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(2704, 15).Value = " <p3 i=""" & Str(33) & """/>"
            Worksheets("XML").Cells(4421, 15).Value = " <p1 v=""" & Str(StiffnessRx5) & """/>"
            Worksheets("XML").Cells(4422, 15).Value = " <p2 v=""" & Str(StiffnessY5) & """/>"
        End If
        'PILE 5 SEGMENT 8
        If t5 > (L2_1 + L2_2) Then
            Worksheets("XML").Cells(4465, 15).Value = Worksheets("XML").Cells(3373, 15).Value
            Worksheets("XML").Cells(4466, 15).Value = Worksheets("XML").Cells(3374, 15).Value
            Worksheets("XML").Cells(4467, 15).Value = Worksheets("XML").Cells(3375, 15).Value
        End If
        If t5 <= (L2_1 + L2_2) And t5 > L2_1 Then
            k = BedLevel - t5
```

```
                Worksheets("XML").Cells(1287, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2681, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2704, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(4447, 15).Value = " <p1 v=""" & Str(StiffnessRx5) & """/>"
                Worksheets("XML").Cells(4448, 15).Value = " <p2 v=""" & Str(StiffnessY5) & """/>"
            End If
            'PILE 5 SEGMENT 9
            If t5 > (L2_1 + L2_2 + L2_3) Then
                Worksheets("XML").Cells(4491, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(4492, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(4493, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t5 <= (L2_1 + L2_2 + L2_3) And t5 > (L2_1 + L2_2) Then
                k = BedLevel - t5
                Worksheets("XML").Cells(1293, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2704, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(4473, 15).Value = " <p1 v=""" & Str(StiffnessRx5) & """/>"
                Worksheets("XML").Cells(4474, 15).Value = " <p2 v=""" & Str(StiffnessY5) & """/>"
            End If
            'PILE 5 SEGMENT 10
            If t5 <= (L2_1 + L2_2 + L2_3 + L2_4) And t5 > (L2_1 + L2_2 + L2_3) Then
                k = BedLevel - t5
                Worksheets("XML").Cells(1299, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
            End If
            Worksheets("XML").Cells(4499, 15).Value = " <p1 v=""" & Str(StiffnessRx5) & """/>"
            Worksheets("XML").Cells(4500, 15).Value = " <p2 v=""" & Str(StiffnessY5) & """/>"
        End If

        'Pile 6---------------------------------------------------------------------------------
--
        If j > 56 And j < 67 Then                                       'Pile 6
            If Nsegments > 5 Then
                Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 45) & """/>"
            End If
            If Nsegments = 5 Then
                Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 35) & """/>"
                L2_1 = L3_1
                L2_2 = L3_2
                L2_3 = L3_3
                L2_4 = L3_4
            End If
            If Nsegments < 5 Then
                Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
            End If
            'PILE 6 SEGMENT 7
            If t6 > L2_1 Then
                Worksheets("XML").Cells(4699, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(4700, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(4701, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t6 <= L2_1 Then
                k = BedLevel - t6
                Worksheets("XML").Cells(1341, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2888, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2911, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2934, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(4681, 15).Value = " <p1 v=""" & Str(StiffnessRx6) & """/>"
                Worksheets("XML").Cells(4682, 15).Value = " <p2 v=""" & Str(StiffnessY6) & """/>"
            End If
            'PILE 6 SEGMENT 8
            If t6 > (L2_1 + L2_2) Then
                Worksheets("XML").Cells(4725, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(4726, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(4727, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t6 <= (L2_1 + L2_2) And t6 > L2_1 Then
                k = BedLevel - t6
                Worksheets("XML").Cells(1347, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2911, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(2934, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(4707, 15).Value = " <p1 v=""" & Str(StiffnessRx6) & """/>"
                Worksheets("XML").Cells(4708, 15).Value = " <p2 v=""" & Str(StiffnessY6) & """/>"
            End If
            'PILE 6 SEGMENT 9
            If t6 > (L2_1 + L2_2 + L2_3) Then
                Worksheets("XML").Cells(4751, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                Worksheets("XML").Cells(4752, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                Worksheets("XML").Cells(4753, 15).Value = Worksheets("XML").Cells(3375, 15).Value
            End If
            If t6 <= (L2_1 + L2_2 + L2_3) And t6 > (L2_1 + L2_2) Then
                k = BedLevel - t6
                Worksheets("XML").Cells(1353, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                Worksheets("XML").Cells(2934, 15).Value = " <p3 i=""" & Str(33) & """/>"
                Worksheets("XML").Cells(4733, 15).Value = " <p1 v=""" & Str(StiffnessRx6) & """/>"
                Worksheets("XML").Cells(4734, 15).Value = " <p2 v=""" & Str(StiffnessY6) & """/>"
            End If
            'PILE 6 SEGMENT 10
            If t6 <= (L2_1 + L2_2 + L2_3 + L2_4) And t6 > (L2_1 + L2_2 + L2_3) Then
```

```
                        k = BedLevel - t6
                        Worksheets("XML").Cells(1359, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                    End If
                    Worksheets("XML").Cells(4759, 15).Value = " <p1 v=""" & Str(StiffnessRx6) & """/>"
                    Worksheets("XML").Cells(4760, 15).Value = " <p2 v=""" & Str(StiffnessY6) & """/>"
                End If

                'Pile 7-------------------------------------------------------------------------------
--
                If j > 66 And j < 77 Then                                       'Pile 7
                    If Nsegments = 6 Then
                        Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(j - 45) & """/>"
                    End If
                    If Nsegments < 6 Then
                        Worksheets("XML").Cells(i, 15).Value = " <p3 i=""" & Str(33) & """/>"
                    End If
                    'PILE 7 SEGMENT 7
                    If t7 > L3_1 Then
                        Worksheets("XML").Cells(4959, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                        Worksheets("XML").Cells(4960, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                        Worksheets("XML").Cells(4961, 15).Value = Worksheets("XML").Cells(3375, 15).Value
                    End If
                    If t7 <= L3_1 Then
                        k = BedLevel - t7
                        Worksheets("XML").Cells(1401, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                        Worksheets("XML").Cells(3118, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(3141, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(3164, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(4941, 15).Value = " <p1 v=""" & Str(StiffnessRx7) & """/>"
                        Worksheets("XML").Cells(4942, 15).Value = " <p2 v=""" & Str(StiffnessY7) & """/>"
                    End If
                    'PILE 7 SEGMENT 8
                    If t7 > (L3_1 + L3_2) Then
                        Worksheets("XML").Cells(4985, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                        Worksheets("XML").Cells(4986, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                        Worksheets("XML").Cells(4987, 15).Value = Worksheets("XML").Cells(3375, 15).Value
                    End If
                    If t7 <= (L3_1 + L3_2) And t7 > L3_1 Then
                        k = BedLevel - t7
                        Worksheets("XML").Cells(1407, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                        Worksheets("XML").Cells(3141, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(3164, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(4967, 15).Value = " <p1 v=""" & Str(StiffnessRx7) & """/>"
                        Worksheets("XML").Cells(4968, 15).Value = " <p2 v=""" & Str(StiffnessY7) & """/>"
                    End If
                    'PILE 7 SEGMENT 9
                    If t7 > (L3_1 + L3_2 + L3_3) Then
                        Worksheets("XML").Cells(5011, 15).Value = Worksheets("XML").Cells(3373, 15).Value
                        Worksheets("XML").Cells(5012, 15).Value = Worksheets("XML").Cells(3374, 15).Value
                        Worksheets("XML").Cells(5013, 15).Value = Worksheets("XML").Cells(3375, 15).Value
                    End If
                    If t7 <= (L3_1 + L3_2 + L3_3) And t7 > (L3_1 + L3_2) Then
                        k = BedLevel - t7
                        Worksheets("XML").Cells(1413, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                        Worksheets("XML").Cells(3164, 15).Value = " <p3 i=""" & Str(33) & """/>"
                        Worksheets("XML").Cells(4993, 15).Value = " <p1 v=""" & Str(StiffnessRx7) & """/>"
                        Worksheets("XML").Cells(4994, 15).Value = " <p2 v=""" & Str(StiffnessY7) & """/>"
                    End If
                    'PILE 7 SEGMENT 10
                    If t7 <= (L3_1 + L3_2 + L3_3 + L3_4) And t7 > (L3_1 + L3_2 + L3_3) Then
                        k = BedLevel - t7
                        Worksheets("XML").Cells(1419, 15).Value = " <p3 v=""" & Str(k) & """/></obj>"
                    End If
                    Worksheets("XML").Cells(5019, 15).Value = " <p1 v=""" & Str(StiffnessRx7) & """/>"
                    Worksheets("XML").Cells(5020, 15).Value = " <p2 v=""" & Str(StiffnessY7) & """/>"
                End If
            End If
        Next j


    End If
    If i = 5045 Then
        If x_V >= 0 And x_V <= (1 * L_hor) Then              'FIELD 1
            Worksheets("XML").Cells(5237, 15).Value = Worksheets("XML").Cells(5237, 6).Value
            For j = 5056 To 5077:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = x_V / (L_hor)
            Worksheets("XML").Cells(5076, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
            Worksheets("XML").Cells(5246, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
        If x_V > (1 * L_hor) And x_V <= (2 * L_hor) Then            'FIELD 2
            Worksheets("XML").Cells(5270, 15).Value = Worksheets("XML").Cells(5270, 6).Value
            For j = 5079 To 5100:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = (x_V - L_hor) / (L_hor)
            Worksheets("XML").Cells(5099, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
```

```
            Worksheets("XML").Cells(5279, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
        If x_V > (2 * L_hor) And x_V <= (3 * L_hor) Then              'FIELD 3
            Worksheets("XML").Cells(5303, 15).Value = Worksheets("XML").Cells(5303, 6).Value
            For j = 5102 To 5123:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = (x_V - 2 * L_hor) / (L_hor)
            Worksheets("XML").Cells(5122, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
            Worksheets("XML").Cells(5312, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
        If x_V > (3 * L_hor) And x_V <= (4 * L_hor) Then              'FIELD 4
            Worksheets("XML").Cells(5336, 15).Value = Worksheets("XML").Cells(5336, 6).Value
            For j = 5125 To 5146:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = (x_V - 3 * L_hor) / (L_hor)
            Worksheets("XML").Cells(5145, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
            Worksheets("XML").Cells(5345, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
        If x_V > (4 * L_hor) And x_V <= (5 * L_hor) Then            'FIELD 5
            Worksheets("XML").Cells(5369, 15).Value = Worksheets("XML").Cells(5369, 6).Value
            For j = 5148 To 5169:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = (x_V - 4 * L_hor) / (L_hor)
            Worksheets("XML").Cells(5168, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
            Worksheets("XML").Cells(5378, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
        If x_V > (5 * L_hor) And x_V <= (6 * L_hor) Then            'FIELD 6
            Worksheets("XML").Cells(5402, 15).Value = Worksheets("XML").Cells(5402, 6).Value
            For j = 5171 To 5192:
                Worksheets("XML").Cells(j, 15).Value = Worksheets("XML").Cells(j, 6).Value
            Next j
            RelLoc = (x_V - 5 * L_hor) / (L_hor)
            Worksheets("XML").Cells(5191, 15).Value = " <p6 v=""" & Str(PresDef) & """/>"
            Worksheets("XML").Cells(5411, 15).Value = " <p15 v=""" & Str(RelLoc) & """/>"
        End If
    End If
Next i
'-------------------------------------------------------------------------------------------------------
For i = 1 To 5416
    mystring = Worksheets("XML").Cells(i, 15).Value
    f.WriteLine (mystring)
Next i
f.Close

Shell ("C:\Program Files (x86)\SCIA\Engineer18.1\ESA_XML.exe LIN
C:\Users\wvdommelen\Desktop\Guidework\GuideworkFinal.esa
C:\Users\wvdommelen\Desktop\Guidework\newlyScripted.xml /tHTML
/oC:\Users\wvdommelen\Desktop\Guidework\Results.xls")

Application.Wait (Now + TimeValue("0:00:20"))

If copyResults = True Then
    Workbooks.Open Filename:="C:\Users\wvdommelen\Desktop\Guidework\Results.xls"
    For i = 78 To 83
        If Worksheets("Results").Cells(i, 5).Value <> 0 Then
        F_result = Worksheets("Results").Cells(i, 5).Value * 1000
        End If
    Next i
    For i = 200 To 400
        If Worksheets("Results").Cells(i, 1).Value = "B7" Then
            Force1 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force1 = 0 Then
            Force1 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 1).Value = "B17" Then
            Force2 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force2 = 0 Then
            Force2 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 1).Value = "B27" Then
            Force3 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force3 = 0 Then
            Force3 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 1).Value = "B37" Then
            Force4 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force4 = 0 Then
            Force4 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 1).Value = "B47" Then
            Force5 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
```

```
            If Force5 = 0 Then
            Force5 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 1).Value = "B57" Then
            Force6 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force6 = 0 Then
            Force6 = 1000
            End If
        End If
        If Worksheets("Results").Cells(i, 5).Value = "B67" Then
            Force7 = Abs(Worksheets("Results").Cells(i, 5).Value * 1000)
            If Force7 = 0 Then
            Force7 = 1000
            End If
        End If
    Next i
'    MsgBox ("Spring " & Force6)
    'MsgBox ("Contact force is: " & F_result)
    Workbooks.Open Filename:="C:\Users\wvdommelen\Desktop\Guidework\AllResults.xlsx"

Debug.Print "1:", Force1, "2:", Force2, "3:", Force3, "4:", Force4, "5:", Force5, "6:", Force6, "7:",
Force7


    Workbooks("Results.xls").Worksheets("Results").Range("A1:I500").Copy
Workbooks("Allresults.xlsx").Worksheets(Blad_list(p - 1)).Range("A1:I500")

    Workbooks("Results.xls").Close
    Workbooks("AllResults.xlsx").Save
    Workbooks("AllResults.xlsx").Close

'---------------------------------------------------UPDATE RESULTS TABLE----------------------------------
-------------
    time_t = time_t + delta_t

    Worksheets("Computed").Cells(p + 2, 6).Value = (time_t)
    Worksheets("Computed").Cells(p + 2, 7).Value = (F_result)
    Worksheets("Computed").Cells(p + 2, 8).Value = (F_result * mu)
    Worksheets("Computed").Cells(p + 2, 9).Value = (PresDef)
    Worksheets("Computed").Cells(p + 2, 10).Value = (x_V)
    Worksheets("Computed").Cells(p + 2, 11).Value = (x_C)
    Worksheets("Computed").Cells(p + 2, 12).Value = (z_C)
    Worksheets("Computed").Cells(p + 2, 13).Value = (theta)
    Worksheets("Computed").Cells(p + 2, 14).Value = (theta_dot)
    Worksheets("Computed").Cells(p + 2, 15).Value = (v_z_C)
    Worksheets("Computed").Cells(p + 2, 16).Value = (z_A)
    Worksheets("Computed").Cells(p + 2, 17).Value = (x_pos)
End If

Debug.Print time_t
'MsgBox ("EindTest" & p)
Next p


End Sub
```